Dive into
Deep Learning
MATHEMATICS FOR DEEP LEARNING
Alexander J Smola · Rachel Hu · Brent Werness

Dive into Deep Learning（2026.11.27 版）。

Table of contents

深層学習のための数学 **1**

33 線形代数 **3**
33.1 ベクトル、角度、射影 4
33.1.1 ベクトルとその幾何学 4
33.1.2 部分空間と最小二乗法 9
33.1.3 高次元における類似度 14
33.1.4 超平面と決定境界 18
33.1.5 まとめ 22
33.1.6 演習問題 22
33.2 線形変換としての行列 25
33.2.1 線形変換 26
33.2.2 ランクと可逆性 33
33.2.3 行列式 39
33.2.4 まとめ 47
33.2.5 演習問題 48
33.3 固有値分解 50
33.3.1 固有値と対角化 51
33.3.2 対称行列と二次形式 61
33.3.3 まとめ 66
33.3.4 演習問題 67
33.4 実践におけるスペクトル 68
33.4.1 固有値の位置特定と計算 69
33.4.2 反復と安定性 74
33.4.3 まとめ 80
33.4.4 演習問題 80
33.5 特異値分解と低ランク近似 82
33.5.1 特異値分解 83
33.5.2 低ランク近似 91
33.5.3 SVD による連立一次方程式の解法 98
33.5.4 深層学習における SVD 102
33.5.5 まとめ 107
33.5.6 演習問題 107
33.6 関連文献 110
33.6.1 幾何学と線形変換 110
33.6.2 固有値とスペクトル 111
33.6.3 打ち切り SVD を超える低ランク構造 112

33.6.4 書籍、講義、講義ノート 114

34 微分積分と自動微分 115
34.1 1 変数の微分積分 116
34.1.1 導関数 116
34.1.2 導関数の計算 119
34.1.3 平均値の定理 122
34.1.4 1 変数の勾配降下法 124
34.1.5 曲率とテイラー級数 126
34.1.6 微分不可能な点 136
34.1.7 まとめ 140
34.1.8 演習問題 140
34.2 多変数の微分積分 143
34.2.1 偏微分と勾配 144
34.2.2 勾配の幾何学 146
34.2.3 ヘッセ行列 150
34.2.4 制約付き最適化とラグランジュ乗数 157
34.2.5 まとめ 160
34.2.6 演習問題 160
34.3 計算グラフ上の連鎖律 162
34.3.1 経路にわたる和としての連鎖律 162
34.3.2 誤差逆伝播法 165
34.3.3 まとめ 169
34.3.4 演習問題 169
34.4 行列微分 170
34.4.1 ヤコビ行列 171
34.4.2 ヤコビ行列の連鎖律 174
34.4.3 ベクトルと行列に関する導関数 177
34.4.4 まとめ 183
34.4.5 演習問題 183
34.5 自動微分 185
34.5.1 フォワードモードと二重数 186
34.5.2 リバースモードとテープ 190
34.5.3 ヤコビ行列積の合成 198
34.5.4 まとめ 202
34.5.5 演習問題 203
34.6 積分法 205
34.6.1 定積分 206
34.6.2 微分積分学の基本定理 208
34.6.3 重積分 216
34.6.4 まとめ 222
34.6.5 演習問題 222
34.7 期待値とその勾配 223
34.7.1 積分としての期待値 224
34.7.2 積分記号下の微分 225
34.7.3 スコア関数勾配とパスワイズ勾配 227
34.7.4 まとめ 230
34.7.5 演習問題 230
34.8 関連文献 231
34.8.1 折れ点における微分 232

34.8.2 ベクトル写像と行列写像の微分 232
34.8.3 自動微分 233
34.8.4 積分 234
34.8.5 書籍、講義、チュートリアル 235

35 最適化 **237**
記法 238
35.1 勾配降下法と曲率 238
35.1.1 降下方向と滑らかさ 239
35.1.2 二次モデルと条件数 246
35.1.3 モメンタムと加速 251
35.1.4 まとめ 255
35.1.5 演習問題 255
35.2 曲率法：ニュートン法、準ニュートン法、信頼領域法 258
35.2.1 ニュートン法 259
35.2.2 準ニュートン法 264
35.2.3 信頼領域 266
35.2.4 まとめ 269
35.2.5 演習問題 269
35.3 凸性 270
35.3.1 凸集合 271
35.3.2 凸関数 272
35.3.3 凸性の帰結 281
35.3.4 凸性の判定 288
35.3.5 凸性を超えて 290
35.3.6 まとめ 294
35.3.7 演習問題 294
35.4 確率的勾配法 297
35.4.1 ミニバッチ勾配 298
35.4.2 ステップサイズの減衰 303
35.4.3 固定予算における収束率 309
35.4.4 分散低減 314
35.4.5 まとめ 317
35.4.6 演習問題 318
35.5 適応的手法 320
35.5.1 座標ごとのステップサイズ 321
35.5.2 座標スケーリングとその限界 325
35.5.3 分離型重み減衰 331
35.5.4 ウォームアップ 334
35.5.5 構造化前処理行列 335
35.5.6 まとめ 337
35.5.7 演習問題 338
35.6 制約付き最適化 340
35.6.1 等式制約とラグランジュ乗数 341
35.6.2 不等式制約と KKT 条件 344
35.6.3 射影と射影勾配降下法 347
35.6.4 近接作用素と座標降下法 354
35.6.5 まとめ 360
35.6.6 演習問題 360
35.7 双対性 363

35.7.1 双対問題 . . . 364
35.7.2 強双対性 . . . 368
35.7.3 価格としての乗数 . . . 371
35.7.4 凸共役 . . . 374
35.7.5 具体例 . . . 378
35.7.6 まとめ . . . 384
35.7.7 演習問題 . . . 385
35.8 数値安定性と条件性 . . . 388
35.8.1 浮動小数点演算 . . . 388
35.8.2 安定なソフトマックスと交差エントロピー . . . 393
35.8.3 桁落ち . . . 397
35.8.4 条件性 . . . 403
35.8.5 まとめ . . . 410
35.8.6 演習問題 . . . 411
35.9 関連文献 . . . 414
35.9.1 勾配法 . . . 414
35.9.2 確率的手法と適応的手法 . . . 415
35.9.3 制約と双対性 . . . 416
35.9.4 浮動小数点 . . . 417
35.9.5 最適化全般の参考資料 . . . 418

36 確率と統計学習 **419**
36.1 確率変数 . . . 420
36.1.1 密度 . . . 420
36.1.2 平均と分散 . . . 426
36.1.3 分布の裾 . . . 430
36.1.4 まとめ . . . 435
36.1.5 演習問題 . . . 435
36.2 同時分布と変換 . . . 437
36.2.1 同時密度と条件付き密度 . . . 438
36.2.2 共分散と相関 . . . 444
36.2.3 確率変数の変換 . . . 449
36.2.4 まとめ . . . 453
36.2.5 演習問題 . . . 454
36.3 分布 . . . 455
36.3.1 離散分布 . . . 456
36.3.2 連続分布 . . . 463
36.3.3 多変量ガウス分布 . . . 472
36.3.4 まとめ . . . 476
36.3.5 演習問題 . . . 477
36.4 指数分布族 . . . 480
36.4.1 多様な分布を統一する単一の形式 . . . 480
36.4.2 最大エントロピー . . . 484
36.4.3 対数分配関数 . . . 487
36.4.4 ソフトマックスの起源 . . . 491
36.4.5 共役事前分布 . . . 494
36.4.6 まとめ . . . 499
36.4.7 演習問題 . . . 499
36.5 統計学 . . . 501
36.5.1 推定量とその標本分布 . . . 502

36.5.2 仮説検定 . . . 513
36.5.3 信頼区間 . . . 522
36.5.4 ブートストラップ法 . . . 526
36.5.5 まとめ . . . 529
36.5.6 演習問題 . . . 529
36.6 最尤法 . . . 533
36.6.1 最尤法の原理 . . . 534
36.6.2 負の対数尤度としての損失 . . . 537
36.6.3 最尤推定量の精度 . . . 540
36.6.4 MAP 推定と正則化 . . . 549
36.6.5 まとめ . . . 553
36.6.6 演習問題 . . . 553
36.7 潜在変数、EM、ELBO . . . 556
36.7.1 混合モデル . . . 556
36.7.2 エビデンス下界 . . . 558
36.7.3 期待値最大化法 . . . 559
36.7.4 まとめ . . . 564
36.7.5 演習問題 . . . 564
36.8 ベイズ計算 . . . 565
36.8.1 ベイズロジスティック回帰の事後分布 . . . 566
36.8.2 棄却サンプリングと重要度サンプリング . . . 573
36.8.3 マルコフ連鎖モンテカルロ法 . . . 577
36.8.4 変分推論 . . . 584
36.8.5 各手法の比較 . . . 587
36.8.6 まとめ . . . 589
36.8.7 演習問題 . . . 590
36.9 集中不等式 . . . 593
36.9.1 指数型の裾上界 . . . 594
36.9.2 高次元における確率 . . . 604
36.9.3 まとめ . . . 609
36.9.4 演習問題 . . . 610
36.10一様収束と汎化 . . . 612
36.10.1一様収束 . . . 612
36.10.2ラデマッハ複雑度 . . . 616
36.10.3補間と二重降下 . . . 624
36.10.4まとめ . . . 629
36.10.5演習問題 . . . 630
36.11ナイーブベイズ . . . 633
36.11.1分類のためのベイズの定理 . . . 634
36.11.2数え上げによる学習 . . . 636
36.11.3数え上げによる Fashion-MNIST 分類 . . . 639
36.11.4分類器の評価 . . . 644
36.11.5まとめ . . . 650
36.11.6演習問題 . . . 650
36.12関連文献 . . . 653
36.12.1確率モデル . . . 654
36.12.2推定と検定 . . . 655
36.12.3事後分布を用いた計算 . . . 656
36.12.4集中と学習理論 . . . 657
36.12.5カウントに基づく分類器 . . . 658

36.12.6 確率・統計の一般的な参考資料 658

37 情報理論とダイバージェンス 661
37.1 エントロピー、交差エントロピー、KL ダイバージェンス 662
37.1.1 情報量とエントロピー 663
37.1.2 KL ダイバージェンスと交差エントロピー 668
37.1.3 分類損失 672
37.1.4 確率密度に対するエントロピーと KL ダイバージェンス 678
37.1.5 まとめ 682
37.1.6 演習問題 683
37.2 符号、圧縮、パープレキシティ 685
37.2.1 記号符号 686
37.2.2 ストリーム符号と記述長 691
37.2.3 エントロピー率と典型系列 696
37.2.4 パープレキシティ 700
37.2.5 まとめ 702
37.2.6 演習問題 702
37.3 f ダイバージェンス 705
37.3.1 f ダイバージェンス族 707
37.3.2 順方向 KL と逆方向 KL 714
37.3.3 全変動、ピンスカーの不等式、データ処理 717
37.3.4 フェンシェル共役とクリティック下界 722
37.3.5 まとめ 726
37.3.6 演習問題 726
37.4 テスト関数と輸送 728
37.4.1 積分確率指標と MMD 730
37.4.2 最適輸送とワッサースタイン距離 735
37.4.3 固定カーネルと訓練されたクリティック 741
37.4.4 まとめ 742
37.4.5 演習問題 742
37.5 スコアに基づく比較 745
37.5.1 スコアとフィッシャーダイバージェンス 746
37.5.2 スタインの恒等式 749
37.5.3 カーネルスタイン不一致度 752
37.5.4 訓練目的関数と対応するダイバージェンス 755
37.5.5 まとめ 757
37.5.6 演習問題 757
37.6 相互情報量 759
37.6.1 従属性の尺度 760
37.6.2 データ処理と予測 767
37.6.3 通信と圧縮 773
37.6.4 情報ボトルネック 776
37.6.5 まとめ 781
37.6.6 演習問題 781
37.7 相互情報量の推定 784
37.7.1 既知の情報量を持つペア 785
37.7.2 ノンパラメトリック推定 786
37.7.3 変分下界 793
37.7.4 InfoNCE 796
37.7.5 まとめ 805

37.7.6 演習問題 . . . 806
37.8 関連文献 . . . 809
37.8.1 エントロピーと損失 . . . 809
37.8.2 符号と圧縮 . . . 810
37.8.3 分布の比較 . . . 812
37.8.4 相互情報量 . . . 814
37.8.5 相互情報量の推定 . . . 814
37.8.6 情報理論全般の参考資料 . . . 815

38 ダイナミクス：微分方程式と生成フロー **819**
記法 . . . 820
38.1 常微分方程式 . . . 821
38.1.1 ベクトル場とフロー . . . 822
38.1.2 線形 ODE と安定性 . . . 828
38.1.3 まとめ . . . 836
38.1.4 演習問題 . . . 837
38.2 数値ソルバ . . . 839
38.2.1 前進オイラー法 . . . 839
38.2.2 ルンゲ・クッタ法 . . . 841
38.2.3 硬さと陰的手法 . . . 846
38.2.4 ソルバとしての勾配降下法 . . . 848
38.2.5 まとめ . . . 851
38.2.6 演習問題 . . . 852
38.3 ニューラル ODE、随伴法、連続フロー . . . 854
38.3.1 ニューラル ODE . . . 856
38.3.2 随伴法 . . . 861
38.3.3 連続正規化フロー . . . 866
38.3.4 まとめ . . . 872
38.3.5 演習問題 . . . 873
38.4 確率微分方程式 . . . 876
38.4.1 ブラウン運動 . . . 876
38.4.2 伊藤解析 . . . 880
38.4.3 オルンシュタイン・ウーレンベック過程 . . . 889
38.4.4 シミュレーション . . . 893
38.4.5 まとめ . . . 898
38.4.6 演習問題 . . . 898
38.5 フォッカー・プランク方程式と確率フロー . . . 901
38.5.1 経路から密度へ . . . 902
38.5.2 フォッカー・プランク方程式 . . . 904
38.5.3 スコア関数 . . . 911
38.5.4 連続の式と確率フロー ODE . . . 914
38.5.5 時間反転 . . . 921
38.5.6 まとめ . . . 928
38.5.7 演習問題 . . . 928
38.6 スコアマッチングと拡散モデル . . . 932
38.6.1 スコアの学習 . . . 932
38.6.2 スコアを用いたサンプリング . . . 940
38.6.3 複数ノイズレベルにわたるスコア . . . 943
38.6.4 決定論的サンプリングとガイダンス . . . 947
38.6.5 まとめ . . . 954

38.6.6 演習問題 954
38.7 フローマッチングと最適輸送 957
38.7.1 フローマッチング 958
38.7.2 Rectified Flow 965
38.7.3 最適輸送と直線性 974
38.7.4 学習された速度場によるサンプリング 978
38.7.5 まとめ 982
38.7.6 演習問題 982
38.8 関連文献 985
38.8.1 ソルバ 985
38.8.2 連続時間モデル 986
38.8.3 ノイズ 987
38.8.4 学習される生成器 989
38.8.5 ダイナミクスと生成フロー全般の参考資料 990

参考文献 **993**

深層学習のための数学

本書の主要な各章では、数学的ツールを**必要になったその場で**導入しており、勢いを失うことなく実際に動作するモデルを構築・訓練できるよう、それぞれの結果を幾何学的・統計的・物理的な直観とあわせて提示している。しかし、既存のレシピを適用する段階から一歩進んで、微妙な訓練の失敗を診断したり、最前線の研究論文を批判的に読み解いたり、新たな手法を考案したりするには、いずれその背後にある数学というエンジンの内部を覗き込む必要がある。なぜ勾配降下法（gradient descent）の学習率には、ヘッセ行列（Hessian）の最大固有値によって定まる厳格な上限があるのか。なぜ交差エントロピー（cross-entropy）を最小化すると真の条件付き分布が推定でき、なぜ 2 つの分布に重なりがないとイェンセン・シャノン・ダイバージェンスは飽和してしまうのか。そして、画像に日々ガウスノイズを加えていくことが、なぜ純粋なノイズを写真へと戻す逆時間の微分方程式を定めることになるのだろうか。

全 6 章からなる本付録は、これらの問いに余すところなく答える。本付録は、互いに補完し合う次の 2 つの目的を果たすよう設計されている。

- **本編と対になるリファレンス。**「**基礎**」パートや「**発展**」パートの各章で数学の定理や恒等式を用い、その実践的な帰結に焦点を当てている箇所では、完全な導出・証明・境界条件（boundary condition）を扱う本付録の対応する節が参照されている。すでに馴染みのあるトピックについては、ここでの該当する章を読み飛ばし、相互参照を見て興味が湧いたときにだけ必要な箇所を拾い読みすればよい。
- **深層学習の数学を扱う講義のための自己完結した教科書。**最初から最後まで通読すれば、これら 6 つの章は深層学習に特化した一貫性のある現代的な数学カリキュラムとなる。病的な例の追求そのものに深入りすることなく、現代の機械学習で実際に使われる範囲まで、各主題を第一原理から展開している。

本書の他のパートと同様に、本付録のすべての節では数学的な導出に実行可能なコードと計算実験（PyTorch、JAX、TensorFlow、MXNet で実装）を組み合わせており、抽象的な定理を自ら調べ、プロットし、検証できる具体的な数値現象へと落とし込んでいる。

- **線形代数（第 33 章）。**深層ネットワークはデータとパラメータをベクトル、行列、テンソルとして表現し、各層が空間の幾何学構造を変換する。本章では長さ、角度、内積、射影、ベクトルおよび行列のノルムから出発し、部分空間を回転させ、引き伸ばし、潰す能動的な線形変換として行列を捉える。続いて、高次元のダイナミクスを支配するスペクトル分解を展開する。具体的には、漸化式や最適化で繰り返される行列積がいかに固有値に支配されるかを示す**固有値分解**（eigendecomposition）、ゲルシュゴリンの円や条件数からランダム行列理論およびマルチェンコ・パスツール則に至る**実践におけるスペクトル**、そしてパラメータ効率的ファインチューニング（LoRA）やスペクトル正規化の基礎を成す**特異値分解**（singular value decomposition、SVD）、ムーア・ペンローズ擬似逆行列、エッカート・ヤングの低ランク近似を扱う。
- **微分積分（第 34 章）。**数十億個のパラメータを持つニューラルネットワークの訓練は、すべてのパラメータに同時に摂動を与えたときスカラーの損失がどう変化するかを計算でき

るかに全面的に支えられている。本章では 1 変数の導関数、テイラー級数、数値微分の落とし穴から始め、多変数の勾配、方向微分、ヤコビ行列（Jacobian matrix）、ヘッセ行列による曲率へと進む。続いて計算グラフ上で多変数の連鎖律（chain rule）をたどり、**自動微分**（automatic differentiation）の仕組みを解き明かす。そこではフォワードモードとリバースモード（**誤差逆伝播法**、backpropagation）による累積を対比し、なぜリバースモードを使えば 1 回の順伝播のコストの小さな定数倍で数百万個のパラメータに関する厳密な勾配を計算できるのかを示す。

- **最適化（第 35 章）**。本章では、第 9 章で観察した経験的現象の背後にある形式的な証明を与える。凸集合と凸関数の幾何学（イェンセンの不等式、凸性の一次および二次特徴づけ）を確立し、L 平滑性と μ 強凸性が条件数 $\kappa = L/\mu$ を通じて勾配降下法の収束率とステップサイズの安定性をいかに支配するかを証明する。さらに、悪条件の二次モデルにおけるモメンタムやニュートン法・準ニュートン法による前処理（preconditioning）を解析し、ラグランジュ乗数、カルーシュ・キューン・タッカー（KKT）最適性条件、ラグランジュ双対性（Lagrangian duality）を用いた制約付き最適化（constrained optimization）を展開する。
- **確率と統計（第 36 章）**。機械学習は、ノイズを含むデータ生成や確率的なミニバッチサンプリングから標本外汎化に至るまで、あらゆる段階で不確実性のもとで推論する。本章では確率空間、条件付き分布、ベイズ則、期待値、主要なパラメトリック分布族（ベルヌーイ、カテゴリ、ガウス、ラプラス、指数）から出発し、経験平均のまれな逸脱を抑える古典的な**集中不等式**（concentration inequality）である Markov、Chebyshev、Hoeffding の各不等式へと積み上げる。続いて統計的推論と学習理論を展開し、最尤推定（maximum likelihood estimation、MLE）、フィッシャー情報量、クラメール・ラオの下界、ベイズ的な**最大事後確率推定**（maximum a posteriori、MAP）、十分統計量、仮説検定、信頼区間、そして**汎化**（generalization）の理論的基礎（経験リスク最小化、一様収束、ラデマッハ複雑度、VC 次元、バイアス・分散のトレードオフ）を扱う。
- **情報理論（第 37 章）**。深層学習における分類、言語モデリング、生成のほぼすべての目的関数は、情報理論に源流を持つ。本章では Claude Shannon による自己情報量（サプライザル）とエントロピーの公理から出発し、近似的なモデル分布 Q を真のデータ分布 P の代わりに用いることで生じる符号化の非効率性を**交差エントロピー**と**カルバック・ライブラー（KL）ダイバージェンス**がいかに測るかを示したうえで、微分エントロピーと最大エントロピー原理を通じてこれらの概念を連続確率変数へと拡張する。さらに視野を広げ、KL ダイバージェンスが機能しない場面で確率分布を比較する枠組みとして、f ダイバージェンス、GAN の背後にある変分双対表現、標本空間の基礎にある幾何学構造を反映する**ワッサーシュタイン最適輸送**（Wasserstein optimal transport）距離を扱う。章の締めくくりでは、**相互情報量**（mutual information）、データ処理不等式（data processing inequality）、変分情報ボトルネック、そして現代の対照表現学習を駆動する InfoNCE 下界を解説する。
- **ダイナミクス、フロー、拡散（第 38 章）**。連続正規化フローや状態空間モデルからノイズ除去拡散、フローマッチングに至るまで、現代の深層学習における最も強力なアイデアの多くは連続時間で最も自然に理解できる。最終章となる本章では、ダイナミクスのツールキットを基礎から構築する。具体的には、常微分方程式（ordinary differential equation、ODE）系、ベクトル場、相図（phase portrait）、線形安定性、数値積分器（オイラー法、ルンゲ・クッタ法、シンプレクティック・リープフロッグ法）、および連続随伴感度法によって訓練される**ニューラル ODE** を扱う。続いて、**連続の式**（continuity equation）と Hutchinson のトレース推定量を通じて速度場が確率密度全体をどのように輸送するかを追跡したのち、ランダム性を導入する。すなわち、ブラウン運動、**伊藤確率微分方程式**（Itô stochastic differential equation、SDE）、伊藤の補題、オルンシュタイン・ウーレンベック過程、**フォッカー・プランク方程式**、さらに Anderson の逆時間 SDE と**確率フロー ODE** を導き、スコア関数 $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ を学習することでニューラルネットワークが純粋なガウスノイズを任意のターゲット分布からのサンプルへと変換できる理由を明らかにする。

Chapter 33

線形代数

本書に登場するすべてのニューラルネットワークは、行列積から組み立てられている。線形層は入力に重み行列を掛け、アテンションはクエリにキーを掛け、リカレントネットワークはステップごとに隠れ状態に同じ行列を掛ける。1.3 節では、行列が作用するベクトルに対して実際に**何をするのか**を問うことなく、これらの積を計算した。選択章である本章では、証明も含めてこの幾何学的な問いに取り組む。本文の内容は本章に一切依存しないが、行列積の背後にある幾何学が見えてくると、本書に登場するほぼすべてのモデルの読み解き方が変わるだろう。

784 ピクセルの Fashion-MNIST 画像を 256 個の特徴量へ写像することを考えよう。2 枚の画像はどれほど似ているか、この層はどれだけの情報を捨てるか、そして同じ線形変換（linear map）を 50 回続けて適用すると何が起きるだろうか。784 次元の空間に図を描くことはできないため、その答えは平面幾何を任意の次元へと拡張する代数から得られる。

複雑な行列も、適切な方向に沿って見れば見通しがよくなる。1810 年頃、Gauss は未知数より多くの観測値に対して最小二乗法（least squares）で小惑星パラスの軌道を当てはめたが、これは本章の最初の節で解き、最後の節で再び取り上げる問題である。正規直交軸のもとでは長さと角度がドット積に帰着し、固有ベクトル（eigenvector）に沿えば正方行列は各方向を 1 つのスカラー倍にし、1 組の特異基底のもとでは**任意の**行列が独立な伸縮へと分解される。アテンションがなぜスコアを $\sqrt{d}$ で割るのか（10.2 節）、リカレント勾配がいつ爆発するのか（8.5 節）、そして低ランクアダプタが何を表現できるのかは、これらの考え方から説明できる。

本章は 2 つの段階で進む。第 1 段階では、行列を 1 回だけ適用する。33.1 節では長さ、角度、および部分空間（subspace）内で最も近い点を与える直交射影（orthogonal projection）を高次元へ拡張し、33.2 節ではどの方向が 0 に潰れるか、また写像がいつ可逆になるかを追跡する。第 2 段階では、行列に合わせた座標系を選ぶ。対称行列に対する固有ベクトルとスペクトル定理（33.3 節）、固有値（eigenvalue）だけでは捉えきれない非正規な振る舞い（33.4 節）、そして任意の形状に対する特異ベクトル（singular vector）と最適な低ランク打ち切り（33.5 節）を順に扱う。本章全体を通じて、安易な思い込みを戒める 2 つの小さな行列を一貫して用いる。複素固有値を持つ $\mathbf{A} = \left(\begin{smallmatrix} 1 & 2 \\ -1 & 3 \end{smallmatrix}\right)$ と、固有ベクトルの張る空間（span）が平面全体に満たないせん断（shear）$\mathbf{T} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ である。

初読の際は、n 次元の行列式（determinant）や、33.4 節における Gershgorin の円盤、PageRank、擬スペクトル（pseudospectrum）の議論は流し読みしてかまわない。数値的な固有値ソルバと線形ソルバについては 33.6 節で概観する。

33.1 ベクトル、角度、射影

「基礎」パートの各章を通して、あらゆる入力をベクトルとして表してきた。画像はピクセル値の並びになり、単語は埋め込みになった。機械学習モデルがこれらのベクトルに対して行う操作は、突き詰めれば 3 つの幾何学的な操作に集約される。2 つの埋め込みが似ているかを測るときのようにベクトルを**比較**し、目的変数の組に最もよく当てはまる特徴量の組み合わせを求めるときのようにベクトルを**当てはめ**、分類器が入力を境界のどちら側に振り分けるかを決めるときのようにベクトルを**分離**する。2 次元平面では、これらの操作はいずれも長さ、角度、影からなるおなじみの図として描ける。しかし実際の入力は、誰にも図示できない何百、何千もの次元の空間に存在する。そのような高次元空間でも、こうした幾何学的なイメージはなお意味を持つのだろうか。

T シャツとズボンの見分けが具体的なテストになる。28×28 の Fashion-MNIST 画像（3.2 節）はそれぞれ 784 個のピクセル輝度の並びであり、$\mathbb{R}^{784}$ 内の 1 点である。考えうる最も単純な規則を試してみよう。平均的な T シャツと平均的なズボンを計算し、新しい各画像をより近いほうの平均に割り当てるのである。平面上なら「より近い」かどうかは目で見て確かめられるが、$\mathbb{R}^{784}$ では画像 1 枚あたり 1 回のドット積で済む計算になる。これら 2 つのクラスのテスト画像 $2{,}000$ 枚に対して、この規則は約 92% の割合で正しい判定を下す。これほど単純な規則がこれほどうまく機能するという事実から、2 つの点群は十分に離れて位置しており、目に見えない空間でも距離のような幾何学的概念が意味を保っていることがわかる。

平面の幾何を任意の次元へと運ぶのはドット積である。長さはドット積の平方根であり、あるベクトルの別のベクトルへの影はドット積の比で表され、2 つのベクトルのなす角もドット積から直接定義される。本節では、この幾何学を 3 つの段階で展開する。第 1 に、1 組のベクトルを対象として、直線への射影、2 つのベクトルのなす角、およびコーシー・シュワルツの不等式を調べる。この不等式は 1.3 節で代数的に証明したが、ここでは直角三角形から改めて証明する。第 2 に、本章の残りの部分で基盤となる部分空間の言葉（張る空間、線形独立性、基底、次元）を導入し、QR 分解を計算ツールとして最小二乗法を射影として解く。第 3 に、これらの考え方を高次元の画像へと持ち込み、$\mathbb{R}^{d}$ において無関係なベクトル同士がどれほど似ているかを問い、線形分類器を超平面として読み解く。

なかでも特に重要な結果が 3 つある。射影の残差が射影先の直線や部分空間と直交し、それゆえに射影が最近接点になること、任意の次元で角度が矛盾なく定義されることを保証するコーシー・シュワルツの不等式、そして典型的なスケールが $1/\sqrt{d}$ となる $\mathbb{R}^{d}$ 内のランダムな方向の間のコサインである。データセットの読み込みを除けば、本節のコードはすべて素の NumPy で書かれており、どのフレームワークでも同一に動作する。

33.1.1 ベクトルとその幾何学

33.1.1.1 点と方向

ベクトルは、互いに補い合う 2 つの見方で捉えられる。**点**として見れば、$\mathbf{v} = (3, 2)$ は**原点**から右に 3 単位、上に 2 単位進んだ位置を表す。この見方に立つと、データセット全体が 1 つの点群になり、T シャツとズボンを見分けるような分類課題は 2 つの点群を分離する幾何学的な問題になる。一方、**方向**（または変位）として見れば、同じベクトルは右へ 3 歩、上へ 2 歩移動せよという指示になる。変位はどこを始点にしてもよいため、Figure 33.1 に描かれた平行な矢印はすべて同じベクトルを表している。

方向という見方をとれば、ベクトルの加算は直ちに理解できる。1 本目の矢印をたどり、その先端から 2 本目の矢印をたどればよい（Figure 33.2）。

減算も同様である。$\mathbf{u} = \mathbf{v} + (\mathbf{u} - \mathbf{v})$ が成り立つため、差 $\mathbf{u} - \mathbf{v}$ は点 $\mathbf{v}$ から点 $\mathbf{u}$ へと運ぶ変位の矢印にほかならない。

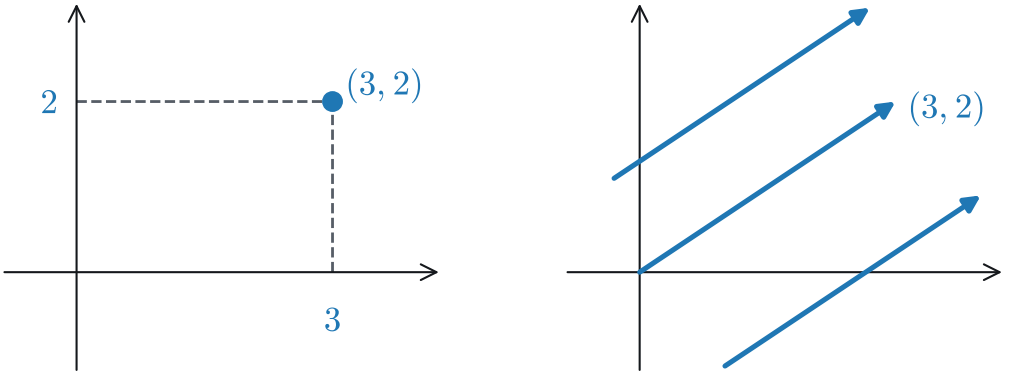


Figure 33.1: ベクトルの 2 通りの読み方：座標 (x, y) を持つ**点**として見るか（左）、どこからでも出発できる矢印としての**方向**と見るかであり、後者では図中のすべての矢印が同じベクトル $(3, 2)$ を表す（右）。

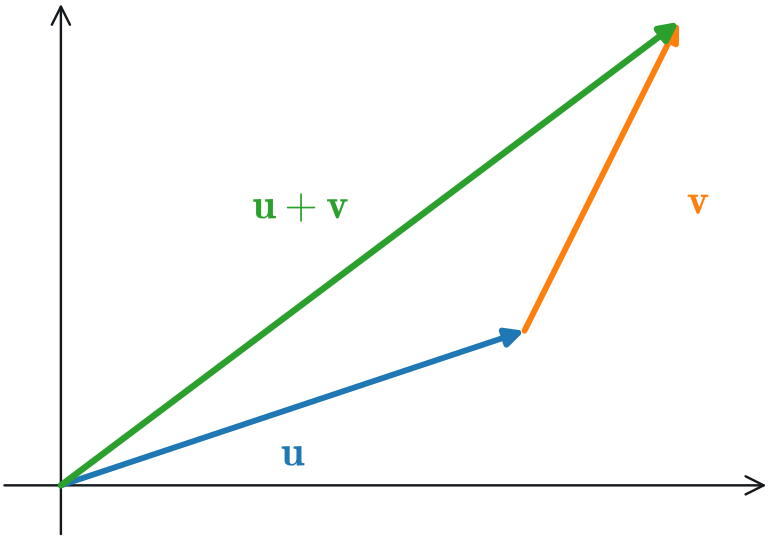


Figure 33.2: 先端と始点をつなぐベクトルの加算：1 本目の矢印をたどり、その終点から 2 本目の矢印をたどる。

本節のこれ以降の内容はすべて、1.3 節で導入した**ドット積**（dot product）というただ 1 つの演算から導かれる。2 つのベクトル $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ に対し、対応する成分同士を掛けてその和をとる演算である。

$$\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^{d} u_i v_i. \tag{33.1}$$

ドット積は $\mathbf{u}^\top \mathbf{v} = \mathbf{v}^\top \mathbf{u}$ と対称であるため、しばしば $\mathbf{u} \cdot \mathbf{v}$ と書く。2 本の矢印をイメージするときはドットを用い、行列の式の中に現れるときは転置を用いる。$\mathbf{u} = (1, 3)$ と $\mathbf{v} = (2, 1)$ に対しては $\mathbf{u} \cdot \mathbf{v} = 2 + 3 = 5$ であり、$\mathbf{u}$ の長さの二乗は $\mathbf{u} \cdot \mathbf{u} = 1 + 9 = 10$ となる。

33.1.1.2 直線への射影

ベクトル $\mathbf{v}$ のうち、別のベクトル $\mathbf{w}$ に沿った成分はどれだけあるだろうか。同じことだが、どのスカラー倍 $t\mathbf{w}$ が $\mathbf{v}$ に最も近くなるだろうか。これは線形代数における最も単純な当てはめ問題であり、単一の方向を使って $\mathbf{v}$ を近似し、最良の係数 t を求めることにあたる。

先ほどの組 $\mathbf{v} = (1, 3)$ と $\mathbf{w} = (2, 1)$ で、t にいくつか値を代入して試してみよう。$t = 1$ では、誤差 $\mathbf{v} - t\mathbf{w} = (-1, 2)$ の長さの二乗は 5 である。$t = 0.5$ では誤差は $(0, 2.5)$ で長さの二乗は 6.25 になり、$t = 1.5$ では $(-2, 1.5)$ でやはり長さの二乗は 6.25 になる。最小値は $t = 1$ で達成され、最近接点は $(2, 1)$ となる。このとき残る誤差を**残差**（residual）と呼ぶ。ここでは残差は $\mathbf{r} = (-1, 2)$ であり、$\mathbf{w}$ とのドット積を計算すると $\mathbf{r} \cdot \mathbf{w} = -2 + 2 = 0$ となることに注目しよう。最短の誤差ベクトルは直線と直角に交わるのである。さらに都合のよいことに、長さの二乗の間には直角三角形の関係が成り立つ。すなわち $\|\mathbf{v}\|^2 = 10$ に対し、最近接点の長さの二乗は 5、残差の長さの二乗も 5 であり、$10 = 5 + 5$ となっている。

この法則はすべての次元で成り立つ。

命題（直交射影）。 $\mathbf{w} \neq \mathbf{0}$ とする。$\mathbf{w}$ のスカラー倍のうち $\mathbf{v}$ に最も近いものは次で与えられる。

$$\operatorname{proj}_{\mathbf{w}} \mathbf{v} = \frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}} \, \mathbf{w}, \tag{33.2}$$

また、残差 $\mathbf{r} = \mathbf{v} - \operatorname{proj}_{\mathbf{w}} \mathbf{v}$ は $\mathbf{r} \cdot \mathbf{w} = 0$ を満たす。

証明。 二乗誤差を t の関数として最小化する。$\mathbf{v} - t\mathbf{w}$ 自身のドット積を展開すると、t の二次関数が得られる。

$$q(t) = \|\mathbf{v} - t\mathbf{w}\|^2 = \|\mathbf{w}\|^2 \, t^2 - 2 \, (\mathbf{v} \cdot \mathbf{w}) \, t + \|\mathbf{v}\|^2.$$

最高次の係数 $\|\mathbf{w}\|^2$ は正であるため、$q(t)$ は導関数 $q'(t) = 2\|\mathbf{w}\|^2 t - 2\,\mathbf{v} \cdot \mathbf{w}$ がゼロになる点、すなわち $t^\star = (\mathbf{v} \cdot \mathbf{w}) / (\mathbf{w} \cdot \mathbf{w})$ で唯一の最小値をとる。$t^\star$ を代入すれば Equation 33.2 が得られる。残差を確認すると、$\mathbf{r} \cdot \mathbf{w} = \mathbf{v} \cdot \mathbf{w} - t^\star \, \mathbf{w} \cdot \mathbf{w} = 0$ となる。■

先ほどの例では $q(t) = 5t^2 - 10t + 10$ となり、最小値 5 を $t^\star = 5/5 = 1$ でとるため、試行した値の結果が裏づけられる。$\operatorname{proj}_{\mathbf{w}} \mathbf{v}$ を、$\mathbf{v}$ の $\mathbf{w}$ への**射影**（projection）、あるいは**影**（shadow）と呼ぶ。射影は $\mathbf{v}$ のうち $\mathbf{w}$ に沿った成分を取り出し、残差はそれに直交する成分を取り出す。なぜ $\mathbf{w} \neq \mathbf{0}$ を仮定したかにも注意しよう。ゼロベクトルは方向を定めず、$\mathbf{w} = \mathbf{0}$ とおくとゼロ除算が生じるからである。

この直角のおかげで、長さの二乗がきれいに足し合わされる。2 つのベクトル $\mathbf{a}$ と $\mathbf{b}$ は、$\mathbf{a}\cdot\mathbf{b}=0$ のとき**直交**（orthogonal）するという。このとき、和の長さの二乗を展開すると**ピタゴラスの定理**（Pythagoras' theorem）が得られる。

$$\|\mathbf{a}+\mathbf{b}\|^2 = \|\mathbf{a}\|^2 + 2\,\mathbf{a}\cdot\mathbf{b} + \|\mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2. \tag{33.3}$$

どの次元でも、直交する成分同士では交差項が消え、長さの二乗がそのまま足し合わされる。これを分解 $\mathbf{v}=\operatorname{proj}_{\mathbf{w}}\mathbf{v}+\mathbf{r}$ に適用すると $\|\mathbf{v}\|^2=\|\operatorname{proj}_{\mathbf{w}}\mathbf{v}\|^2+\|\mathbf{r}\|^2$ が得られ、先ほどの例ではこれが $10=5+5$ に対応する。

また、影が $\mathbf{w}$ と同じ向きか逆向きかを表す符号を含めた影の長さも必要になる。係数 $t^\star$ に $\|\mathbf{w}\|$ を掛けると、この**符号付き長さ**（signed length）が得られる。

$$\frac{\mathbf{v}\cdot\mathbf{w}}{\|\mathbf{w}\|}. \tag{33.4}$$

先ほどの例では $5/\sqrt{5}=\sqrt{5}$ となり、$(2,1)$ の長さに一致する。この符号付き長さは、本節の終盤で点から超平面までの符号付き距離として再び登場する。Figure 33.3 にこの幾何学的な関係全体を示す。

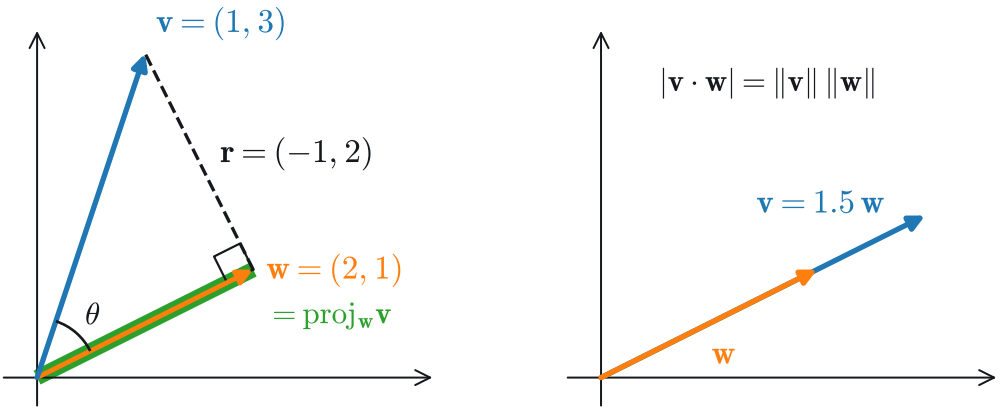


Figure 33.3: (a) $\mathbf{v}=(1,3)$ の $\mathbf{w}=(2,1)$ への射影（影）は $(2,1)$ であり、残差 $(-1,2)$ は $\mathbf{w}$ と直角に交わるため $\|\mathbf{v}\|^2=10=5+5$ が成り立つ。(b) コーシー・シュワルツの不等式の等号成立条件：$\mathbf{v}$ が $\mathbf{w}$ と共線であり、残差がゼロになる。

33.1.1.3 コーシー・シュワルツの不等式と角度

784 次元の空間に分度器はない。それでも「2 枚の画像のなす角」に意味を持たせられるだろうか。意外なことに、その答えははるかに単純な問いから得られる。ベクトルの影が元のベクトル自身より長くなることはありうるだろうか。Figure 33.3 のパネル (a) を見れば、答えは否である。影は $\mathbf{v}$ を斜辺とする直角三角形の 1 辺だからである。ピタゴラスの定理を使えば、この図をあらゆる次元で通用する証明へと変えられる。

命題（コーシー・シュワルツの不等式）。 すべてのベクトル $\mathbf{v},\mathbf{w}$ に対して、次の不等式が成り立つ。

$$|\mathbf{v}\cdot\mathbf{w}| \le \|\mathbf{v}\|\,\|\mathbf{w}\|, \tag{33.5}$$

等号は $\mathbf{v}$ と $\mathbf{w}$ が共線（一方が他方のスカラー倍）のとき、かつそのときに限り成り立つ。

証明。 $\mathbf{w}=\mathbf{0}$ のときは両辺ともゼロになり、2 つのベクトルは共線（collinear）である。そうでないとき、$\mathbf{v}$ を $\mathbf{w}$ への射影とそれに直交する残差 $\mathbf{r}$ に分解する。ピタゴラスの定理 Equation 33.3 より、次が成り立つ。

$$\|\mathbf{v}\|^2 = \|\operatorname{proj}_{\mathbf{w}}\mathbf{v}\|^2 + \|\mathbf{r}\|^2 \ge \|\operatorname{proj}_{\mathbf{w}}\mathbf{v}\|^2 = \frac{(\mathbf{v}\cdot\mathbf{w})^2}{\|\mathbf{w}\|^2}.$$

両辺に $\|\mathbf{w}\|^2$ を掛けて平方根をとれば、求める不等式が得られる。等号が成り立つのは $\mathbf{r}=\mathbf{0}$、すなわち $\mathbf{v}=t^{\star}\mathbf{w}$ のとき、かつそのときに限る。逆に $\mathbf{v}=t\mathbf{w}$ ならば、$|\mathbf{v}\cdot\mathbf{w}| = |t|\,\|\mathbf{w}\|^2 = \|\mathbf{v}\|\,\|\mathbf{w}\|$ となる。$\blacksquare$

平たく言えば、ベクトルの影が元のベクトル自身より長くなることは決してなく、長さが一致するのは残差に何も残らない場合だけである。1.3 節ではコーシー・シュワルツの不等式を代数的に証明したが、ここではパネル (a)（Figure 33.3）から直ちに導かれ、パネル (b) が等号成立の場合を示している。

これで任意の次元において角度を定義できる。平面上では、余弦定理により $\mathbf{v}\cdot\mathbf{w} = \|\mathbf{v}\|\|\mathbf{w}\|\cos\theta$ が成り立つ（1.3 節）。$\mathbb{R}^d$ には実際に測定できる三角形がないため、この恒等式を定義へと転用する。ゼロでないベクトル $\mathbf{v},\mathbf{w}\in\mathbb{R}^d$ に対し、そのなす**角度**を次で定義する。

$$\theta = \arccos\!\left(\frac{\mathbf{v}\cdot\mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}\right) \in [0,\pi]. \tag{33.6}$$

arccos は $[-1,1]$ 上でのみ定義されるため、この式が意味を持つのは比がその区間内に収まる場合に限られるが、コーシー・シュワルツの不等式によってそれは常に保証される。したがって角度はすべての次元で矛盾なく定義され、平面におけるドット積の公式が直ちに任意の次元で成り立つ。

$$\mathbf{v}\cdot\mathbf{w} = \|\mathbf{v}\|\|\mathbf{w}\|\cos\theta. \tag{33.7}$$

コーシー・シュワルツの不等式における 2 つの等号成立条件は、$\theta=0$（同じ向きを指すベクトル）と $\theta=\pi$（反対向きを指すベクトル）に対応する。

ここで自然な疑問が生じる。この代数的な定義は、分度器で実際に測った角度と一致するのだろうか。結論から言えば一致する。任意の 2 つのベクトルは 1 つの平面内（共線であれば 1 つの直線上）に収まり、その平面内では通常の幾何学が成り立つ。射影を使えば、その平面に直交座標を入れられる。$\mathbf{q}_1 = \mathbf{w}/\|\mathbf{w}\|$ とおき、$\mathbf{r}\neq\mathbf{0}$ であれば $\mathbf{q}_2 = \mathbf{r}/\|\mathbf{r}\|$ とおく。これら 2 つのベクトルは単位長を持ち互いに直交するため、それらの線形結合のドット積は $\mathbb{R}^2$ におけるドット積とまったく同じように振る舞う。この座標系では $\mathbf{w} = (\|\mathbf{w}\|, 0)$、$\mathbf{v} = (\mathbf{v}\cdot\mathbf{w}/\|\mathbf{w}\|,\ \|\mathbf{r}\|)$ となる。平面上のこれら 2 本の矢印のなす角を分度器で測ると、そのコサインは $(\mathbf{v}\cdot\mathbf{w}/\|\mathbf{w}\|)/\|\mathbf{v}\|$ となり、Equation 33.6 と一致する。先ほどの例 $\mathbf{v}=(1,3)$ と $\mathbf{w}=(2,1)$ では $\cos\theta = 5/(\sqrt{10}\sqrt{5}) = 1/\sqrt{2}$ となり、$\theta = 45^{\circ}$ が得られる。

ゼロでないベクトルに対して、直交とは $\theta=\pi/2$ を意味する。直交性を $\mathbf{v}\cdot\mathbf{w}=0$ で定義しておくと、ゼロベクトルも含められるというちょっとした利点がある。ゼロベクトルに対して角度は定義されないが、すべてのベクトルと直交することになる。

Equation 33.6 をコードにするには 2 行で足りるが、実用上注目すべき安全策が 1 つある。先ほどの例と、意図的に扱いを難しくした共線な組 $\mathbf{v} = (10^8 + 1, 3, 7)$ と $3\mathbf{v}$（真のコサインは 1、真の角度は 0）で試してみよう。共線な組については、生のコサインから 1 を引いた値と、クリップ前の `arccos` も出力する。

```
def angle(v, w):
    cos = v @ w / (np.linalg.norm(v) * np.linalg.norm(w))
    return np.arccos(np.clip(cos, -1, 1))  # roundoff can push |cos| past 1

v = np.array([1e8 + 1, 3, 7])
cos = v @ (3 * v) / (np.linalg.norm(v) * np.linalg.norm(3 * v))
with np.errstate(invalid='ignore'):        # silence the warning for nan
    print(angle(np.array([1., 3]), np.array([2., 1])), cos - 1,
          np.arccos(cos), angle(v, 3 * v))
```

```
0.7853981633974484 2.220446049250313e-16 nan 0.0
```

最初の数値は $0.785 = \pi/4$ であり、先ほどの例の角度 $45°$ と一致する。残りの 3 つの値は、なぜ `np.clip` が不可欠かを示している。厳密な演算では、コーシー・シュワルツの不等式によって比は $[-1, 1]$ の内部に保たれる。しかし浮動小数点演算では、丸めによって最終桁の 1 単位分だけ境界の外へはみ出すことがある。ここでは共線な組に対して $1 + 2.2 \times 10^{-16}$ が生じ、`np.arccos` が `nan` を返している。$[-1, 1]$ にクリップすることで、正しい答えである 0 が復元される。

コーシー・シュワルツの不等式からは、和の長さが長さの和を決して超えないこともわかる。$\|\mathbf{v}+\mathbf{w}\|^2$ を展開し、交差項 $2\,\mathbf{v}\cdot\mathbf{w}$ を Equation 33.5 によって $2\,\|\mathbf{v}\|\|\mathbf{w}\|$ で上から抑えると、1.3 節と同様に三**角不等式**（triangle inequality）$\|\mathbf{v}+\mathbf{w}\| \le \|\mathbf{v}\| + \|\mathbf{w}\|$ が得られる。

33.1.2 部分空間と最小二乗法

33.1.2.1 張る空間、基底、部分空間

ある量を時刻 $t = 0, 1, 2$ で 3 回測定し、データに最もよく当てはまる直線 $C + Dt$ を求めたいとしよう。測定値を並べてベクトル $\mathbf{y} \in \mathbb{R}^3$ とし、時刻を $\mathbf{t} = (0, 1, 2)$ にまとめ、$\mathbf{1} = (1, 1, 1)$ とおく。3 つの時刻で直線の値を評価すると、ベクトル $C\mathbf{1} + D\mathbf{t}$ が得られる。したがって最良の直線を見つけるとは、$C\mathbf{1} + D\mathbf{t}$ の形をしたベクトルのうち $\mathbf{y}$ に最も近いものを見つけることにほかならない。C と D がすべての実数を動くとき、これらの候補ベクトルは $\mathbb{R}^3$ 内で原点を通る平面を描き出す。直線への射影では単一の方向を扱ったが、平面やその高次元版を扱うには、基本的な用語をいくつか用意する必要がある。張る空間、基底、部分空間にすでに慣れている読者は、直線の当てはめを完了させる次の項へそのまま進んでかまわない。

重み付き和 $a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k$ を、$\mathbf{v}_1, \ldots, \mathbf{v}_k$ の**線形結合**（linear combination）と呼ぶ。そのような線形結合全体の集合が、それらの**張る空間**（span）である。

$$\operatorname{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \{a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k : a_1, \ldots, a_k \in \mathbb{R}\}.$$

$\mathbb{R}^2$ では（Figure 33.4）、1 つのゼロでないベクトルが張る空間はその方向に沿って原点を通る直線であり、原点を通る同一の直線上にない 2 つのベクトルが張る空間は平面全体になる。先ほどの直線当てはめの問題では、候補となる当てはめ全体の集合は $\operatorname{span}(\mathbf{1}, \mathbf{t})$ である。

張る空間は、常に加算とスカラー倍について閉じていることに注目しよう。2 つの線形結合を足しても、あるいは 1 つの線形結合を実数倍しても、やはり線形結合になる。これら 2 つの閉包性を持つ空でないベクトルの集合を**部分空間**（subspace）と呼ぶ。すべての部分空間はゼロベクト

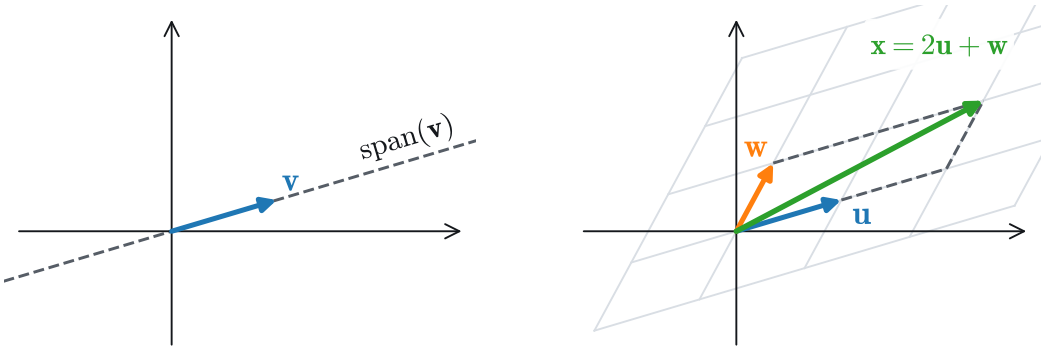


Figure 33.4: 左：1 つのゼロでないベクトル $\mathbf{v}$ が張る空間はそのスカラー倍全体の集合であり、原点を通る直線、すなわち 1 次元の部分空間をなす。右：原点を通る同一の直線上にない 2 つのベクトルは平面全体を張る。破線の平行四辺形は $\mathbf{x}$ を $2\mathbf{u} + \mathbf{w}$ と分解しており、$\mathbf{u}$ と $\mathbf{w}$ が線形独立であるため座標は一意に定まる。

ル $\mathbf{0}$ を含まなければならない（任意の要素に 0 を掛ければよい）。つまり、原点を通らない直線や平面は部分空間ではない。$\mathbb{R}^2$ の部分空間は、原点のみからなる集合、原点を通る直線、そして $\mathbb{R}^2$ 自身に限られる。$\mathbb{R}^3$ では、これに原点を通る平面が加わる。

部分空間を張るベクトルの組には、冗長なものが含まれうる。あるベクトルが他のベクトルの張る空間にすでに含まれているなら、そのベクトルを取り除いても張る空間は縮まない。そのような冗長性が一切ないとき、ベクトルの組は**線形独立**（linearly independent）であるという。これは、線形結合 $a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k$ が $\mathbf{0}$ に等しくなるのが、すべての $a_i = 0$ とした自明な場合に限られることを意味する。先ほどのベクトル $\mathbf{1}$ と $\mathbf{t}$ は線形独立である。実際、$a_1\mathbf{1} + a_2\mathbf{t} = (a_1,\, a_1 + a_2,\, a_1 + 2a_2)$ がゼロになるのは $a_1 = a_2 = 0$ のときに限られる。

部分空間の**基底**（basis）とは、その部分空間を張る線形独立なベクトルの組である。おなじみのベクトル $\mathbf{e}_1 = (1, 0)$ と $\mathbf{e}_2 = (0, 1)$ は $\mathbb{R}^2$ の**標準基底**（standard basis）をなすが、Figure 33.4 の傾いた 2 本の組も同様に立派な基底である。線形代数の基本定理（ここでは証明なしに用いる）によれば、与えられた部分空間のどの基底も同じ個数のベクトルからなり、その共通の個数を部分空間の**次元**と呼ぶ。これにより、「d 次元空間」における d に明確な意味が与えられる。すなわち、$\mathbb{R}^d$ の次元が d であるのは、$\mathbf{e}_1, \ldots, \mathbf{e}_d$ が基底をなすからである。同様に、先ほどの直線当てはめの候補全体は $\mathbb{R}^3$ の 2 次元部分空間をなす。

部分空間内のすべての点に一意な座標を与えるのが線形独立性である。$\mathbf{v}_1, \ldots, \mathbf{v}_k$ が部分空間 S の基底であれば、すべての $\mathbf{x} \in S$ は $\mathbf{x} = a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k$ とただ 1 通りに表せる。基底は S を張るため、少なくとも 1 つの表示が存在する。もし係数 a_i と b_i を持つ 2 通りの表示が存在したとすれば、両者の差をとることで $\sum_i (a_i - b_i)\,\mathbf{v}_i = \mathbf{0}$ が得られ、線形独立性により $a_i = b_i$ がすべての i に対して成り立つ。

いったん基底を固定すれば、k 次元の部分空間は $\mathbb{R}^k$ とまったく同じように振る舞い、各点は係数の組 $(a_1, \ldots, a_k)$ によって一意に指定される。直線の当てはめにおいては、その座標とは単に切片 C と傾き D のことである。

33.1.2.2 部分空間への射影

具体的な測定値 $\mathbf{y} = (6, 0, 0)$ を用いて直線の当てはめに戻ろう。$C\mathbf{1} + D\mathbf{t}$ という形の平面 $\operatorname{span}(\mathbf{1}, \mathbf{t})$ 上の点のうち、$\mathbf{y}$ に最も近いものを求めたい。1 本の直線へ射影したときは、最近接点の残差はその直線と直交していた。平面への射影でも、残差は平面全体と直交し、したがって $\mathbf{1}$

と $\mathbf{t}$ の両方と直交するはずである（この事実はすぐ後で証明する）。両方のドット積をゼロとおくと、2 変数に関する 2 本の線形方程式が得られる。

$$\mathbf{1}\cdot(\mathbf{y}-C\mathbf{1}-D\mathbf{t}) = 6-3C-3D = 0,$$
$$\mathbf{t}\cdot(\mathbf{y}-C\mathbf{1}-D\mathbf{t}) = 0-3C-5D = 0.$$

第 2 式から第 1 式を引くと $2D=-6$ となるので、$D=-3$、$C=5$ が得られる。最もよく当てはまる直線は $5-3t$ である。時刻 $t=0,1,2$ における予測値は $(5,2,-1)$ となり、残差は $\mathbf{r}=\mathbf{y}-(5,2,-1)=(1,-2,1)$ となる。直交性の確認も $\mathbf{r}\cdot\mathbf{1}=1-2+1=0$ および $\mathbf{r}\cdot\mathbf{t}=0-2+2=0$ と両方成り立っている。ピタゴラスの定理により、ここでも $\mathbf{y}$ の長さの二乗が $36=30+6$ ときれいに分解される。比較として、$\mathbf{y}$ を $\mathbf{1}$ のみに射影すると定数による当てはめ $\mathbf{y}\cdot\mathbf{1}/\mathbf{1}\cdot\mathbf{1}=2$（標本平均）が得られ、その残差 $(4,-2,-2)$ の長さの二乗は 24 になる。$\mathbf{t}$ に沿った傾きを許すことで、二乗誤差は 24 から 6 へと減少する。Figure 33.5 に $\mathbb{R}^3$ におけるこの射影を示す。

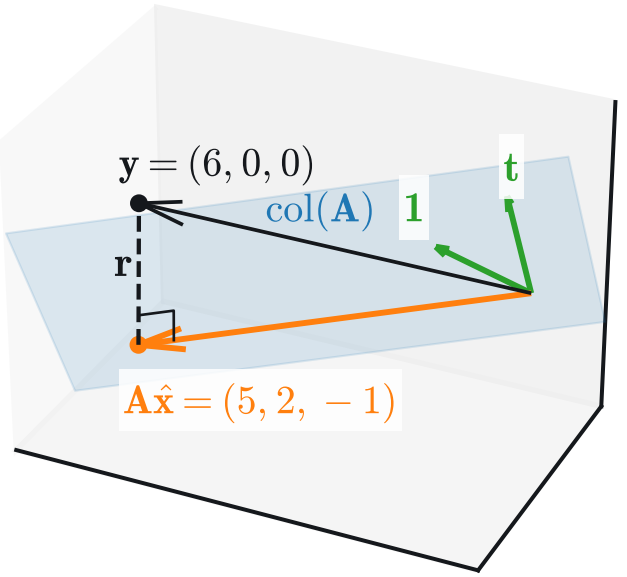


Figure 33.5: 射影としての最小二乗法：測定値 $\mathbf{y}=(6,0,0)$ は候補直線 $C\mathbf{1}+D\mathbf{t}$ のなす平面の外にあり、平面内の最近接点は $(5,2,-1)$（直線 $5-3t$）で、残差 $(1,-2,1)$ は $\mathbf{1}$ と $\mathbf{t}$ の両方に垂直である。

一般の結果を見通しよく述べ、証明するには、互いに直交し単位長を持つベクトルからなる基底を選ぶと都合がよい。ベクトル組 $\mathbf{q}_1,\dots,\mathbf{q}_k$ がなす部分空間 $S\subseteq\mathbb{R}^n$ の基底は、すべてのベクトルが $\|\mathbf{q}_i\|=1$ を満たし、異なるベクトル同士が $\mathbf{q}_i\cdot\mathbf{q}_j=0$ を満たすとき**正規直交**（orthonormal）であるという。有限個のベクトルからなる任意の生成系からは、このような正規直交基底を作ることができる。33.1.2.3 節では、直線への射影だけを使って実際に構成する。これらの基底ベクトルを $n\times k$ 行列 $\mathbf{Q}$ の列として並べると、正規直交性は $\mathbf{Q}^{\top}\mathbf{Q}=\mathbf{I}_k$ という簡潔な形で表せる。なぜなら、成分 (i,j)（$\mathbf{Q}^{\top}\mathbf{Q}$ の各成分）は $\mathbf{q}_i\cdot\mathbf{q}_j$ に等しいからである。本章全体を通じて、文字 $\mathbf{Q}$

は正規直交列を持つ行列を表すために用いる。次の命題は、本章の残りの部分で最も頻繁に用いられる中核的な結果である。

命題（部分空間への射影）。 **Q** が S を張る正規直交列を持つとし、$\mathbf{P} = \mathbf{Q}\mathbf{Q}^{\top}$ とおく。このとき、すべての $\mathbf{x} \in \mathbb{R}^n$ に対して次が成り立つ。 (i) $\mathbf{P}\mathbf{x} \in S$ であり、残差 $\mathbf{x} - \mathbf{P}\mathbf{x}$ は S のすべてのベクトルと直交する。 (ii) $\mathbf{P}\mathbf{x}$ は S 内で $\mathbf{x}$ に最も近い唯一の点である。 (iii) $\mathbf{P}^2 = \mathbf{P}$ かつ $\mathbf{P}^{\top} = \mathbf{P}$ である。

証明。 (i) について、$\mathbf{P}\mathbf{x} = \mathbf{Q}(\mathbf{Q}^{\top}\mathbf{x})$ は **Q** の列の線形結合であるため S に属する。残差に左から $\mathbf{Q}^{\top}$ を掛けると次が得られる。

$$\mathbf{Q}^{\top}(\mathbf{x} - \mathbf{P}\mathbf{x}) = \mathbf{Q}^{\top}\mathbf{x} - (\mathbf{Q}^{\top}\mathbf{Q})\,\mathbf{Q}^{\top}\mathbf{x} = \mathbf{0},$$

したがって残差はすべての列 $\mathbf{q}_i$ と直交し、ひいてはそれらの任意の線形結合とも直交する。(ii) について、任意の $\mathbf{s} \in S$ をとり、$\mathbf{x} - \mathbf{s} = (\mathbf{x} - \mathbf{P}\mathbf{x}) + (\mathbf{P}\mathbf{x} - \mathbf{s})$ と書く。$\mathbf{P}\mathbf{x} - \mathbf{s}$ は S に属する一方、$\mathbf{x} - \mathbf{P}\mathbf{x}$ は S と直交するため、ピタゴラスの定理 Equation 33.3 より次が成り立つ。

$$\|\mathbf{x} - \mathbf{s}\|^2 = \|\mathbf{x} - \mathbf{P}\mathbf{x}\|^2 + \|\mathbf{P}\mathbf{x} - \mathbf{s}\|^2 \geq \|\mathbf{x} - \mathbf{P}\mathbf{x}\|^2,$$

ここで等号が成り立つのは $\mathbf{s} = \mathbf{P}\mathbf{x}$ のときに限られる。(iii) については、$\mathbf{P}^2 = \mathbf{Q}(\mathbf{Q}^{\top}\mathbf{Q})\mathbf{Q}^{\top} = \mathbf{Q}\mathbf{Q}^{\top} = \mathbf{P}$、および $\mathbf{P}^{\top} = (\mathbf{Q}^{\top})^{\top}\mathbf{Q}^{\top} = \mathbf{P}$ である。$\blacksquare$

逆に、(i) の直交条件は最近接点を一意に特徴づける。ある $\mathbf{s} \in S$ に対して残差 $\mathbf{x} - \mathbf{s}$ が S と直交するならば、差 $\mathbf{P}\mathbf{x} - \mathbf{s} = (\mathbf{x} - \mathbf{s}) - (\mathbf{x} - \mathbf{P}\mathbf{x})$ は S に属すると同時に S と直交するため、自分自身と直交することになり、$\mathbf{s} = \mathbf{P}\mathbf{x}$ が導かれる。直線の当てはめで 2 本の直交方程式を解けば必ず最近接点が得られたのは、まさにこのためである。

性質 (iii) の 2 つの式には、それぞれ明確な幾何学的意味がある。恒等式 $\mathbf{P}^2 = \mathbf{P}$ は、1 回目の射影ですでに S 内に到達しているため、2 回目の射影を施しても何も変わらないことを表す。対称性 $\mathbf{P}^{\top} = \mathbf{P}$ こそが、この射影を**直交**射影たらしめている。対称性なしに $\mathbf{P}^2 = \mathbf{P}$ を満たす行列は**斜**射影行列（oblique projector）であり、ベクトルを斜めの方向に沿って列空間へ写すため、一般には最近接点から外れる（3）。1 次元の場合（$k = 1$）には、単一の単位列ベクトル $\mathbf{q} = \mathbf{w}/\|\mathbf{w}\|$ をとることで、$\mathbf{P} = \mathbf{q}\mathbf{q}^{\top}$ は先ほどの直線への射影 Equation 33.2 にそのまま帰着する。

先ほどの直線当てはめは、線形回帰の中核をなす最適化問題である**最小二乗法**（least squares）の 1 例である。行がデータ例、列が特徴量であるデータ行列を $\mathbf{A}$ とし、目的変数のベクトルを $\mathbf{y}$ とする。最小二乗法では、二乗誤差を最小化する係数ベクトル $\mathbf{x}$ を求める。

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2. \tag{33.8}$$

$\mathbf{x}$ がすべての係数ベクトルを動くとき、$\mathbf{A}\mathbf{x}$ は $\mathbf{A}$ の列が張る空間、すなわち**列空間**（column space）$\mathrm{col}(\mathbf{A})$ を描き出す。したがって最小二乗法とは、$\mathrm{col}(\mathbf{A})$ 内で $\mathbf{y}$ に最も近い点を求める問題である。先ほどの射影の命題により、その点は残差 $\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}$ が $\mathbf{A}$ のすべての列と直交する点として定まる。この直交条件を $\mathbf{A}^{\top}(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}) = \mathbf{0}$ と書けば、次の**正規方程式**（normal equations）が得られる。

$$\mathbf{A}^{\top}\mathbf{A}\,\hat{\mathbf{x}} = \mathbf{A}^{\top}\mathbf{y}. \tag{33.9}$$

これは手計算で解いた 2×2 連立方程式の行列形式にほかならない。直線の当てはめでは、$\mathbf{A} = [\mathbf{1}\ \mathbf{t}]$ に対して $\mathbf{A}^{\top}\mathbf{A} = \left(\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right)$ および $\mathbf{A}^{\top}\mathbf{y} = (6, 0)$ となる。$\mathbf{A}$ の列が線形独立であれば、常

に $\mathbf{A}^{\top}\mathbf{A}$ は可逆であり、最適な係数 $\hat{\mathbf{x}}$ は一意に定まる。列が線形従属である場合でも、当てはめベクトル $\mathbf{A}\hat{\mathbf{x}}$ は依然として $\mathrm{col}(\mathbf{A})$ 内の最近接点であるため一意のままだが、それを与える係数ベクトル $\hat{\mathbf{x}}$ は複数存在しうる。

したがってコード上では、当てはめ直線 $5 - 3t$ を計算する方法が 2 通りある。正規方程式を直接解くか、$\mathbf{P} = \mathbf{Q}\mathbf{Q}^{\top}$ を使って射影するかである。$\mathbf{Q}$ を得るには `np.linalg.qr` を呼び出す。この関数は、線形独立な列を持つ行列の列空間に対する正規直交基底を構築する（次の項ではこれを手計算で構成する）。

```
A = np.array([[1., 0], [1, 1], [1, 2]])     # columns 1 and t
y = np.array([6., 0, 0])
x_hat = np.linalg.solve(A.T @ A, A.T @ y)   # normal equations
Q, R = np.linalg.qr(A)                      # orthonormal basis of col(A)
P = Q @ Q.T
r = y - P @ y
print(x_hat.round(6), (P @ y).round(6), r.round(6))
print(np.linalg.norm(P @ P - P), np.linalg.norm(Q.T @ r))
```

```
[ 5. -3.] [ 5.  2. -1.] [ 1. -2.  1.]
1.962615573354719e-16 0.0
```

どちらの方法でも、手計算と同じく係数 $C = 5$ と $D = -3$、射影された値 $(5, 2, -1)$、および残差 $(1, -2, 1)$ が復元される。2 行目にある確認用の 2 つのノルムは浮動小数点の丸め誤差の範囲でゼロになり、$\mathbf{P}^2 = \mathbf{P}$ と $\mathbf{Q}^{\top}\mathbf{r} = \mathbf{0}$ が裏づけられる。

このように小さく性質のよい問題では、2 つの手法は完全に一致する。しかし $\mathbf{A}$ の列がほぼ線形従属である場合、$\mathbf{A}^{\top}\mathbf{A}$ を作ると問題の条件数（丸め誤差に対する感度）が二乗されてしまう（このことは 33.5.3.2 節で示す）。そのため、実用的な最小二乗ソルバは正規方程式を避け、代わりに列空間の正規直交基底を用いて計算する。

33.1.2.3　グラム・シュミット法と QR 分解

実際にはどのようにして正規直交基底を構築するのだろうか。答えは射影の繰り返しである。2 つの線形独立なベクトルが与えられたとき、1 つ目のベクトルの方向はそのまま保ち、2 つ目のベクトルから 1 つ目の方向に沿った影を引き去る。射影の命題により、残った成分は 1 つ目のベクトルと直交する。

これを $\mathbf{a}_1 = (3, 4)$ と $\mathbf{a}_2 = (1, 0)$ に対して実行してみよう。$\mathbf{a}_1$ を正規化すると $\mathbf{q}_1 = \mathbf{a}_1 / 5 = (0.6, 0.8)$ となる。$\mathbf{a}_2$ の $\mathbf{q}_1$ への影は $(\mathbf{q}_1 \cdot \mathbf{a}_2)\,\mathbf{q}_1 = 0.6\,(0.6, 0.8) = (0.36, 0.48)$ である。これを引くと $(1, 0) - (0.36, 0.48) = (0.64, -0.48)$ が残り、これは長さ 0.8 のベクトルで、確かに $\mathbf{q}_1$ と直交している（$0.384 - 0.384 = 0$）。この残りを正規化すると $\mathbf{q}_2 = (0.8, -0.6)$ が得られる。以上の手順を逆にたどれば、元の各ベクトルを新しい正規直交基底で表すことができる。

$$\mathbf{a}_1 = 5\,\mathbf{q}_1, \qquad \mathbf{a}_2 = 0.6\,\mathbf{q}_1 + 0.8\,\mathbf{q}_2 .$$

行列記法で書けば、これは $\mathbf{A} = \mathbf{Q}\mathbf{R}$ である。ここで $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2]$、$\mathbf{Q} = [\mathbf{q}_1\ \mathbf{q}_2]$ であり、$\mathbf{R} = \left(\begin{smallmatrix} 5 & 0.6 \\ 0 & 0.8 \end{smallmatrix}\right)$ はドット積 $\mathbf{q}_i \cdot \mathbf{a}_j$ を集めた行列である。なぜ $\mathbf{R}$ が上三角行列になるかにも注目しよう。構成法から、各 $\mathbf{a}_j$ は $\mathbf{q}_1, \dots, \mathbf{q}_j$ のみに依存するからである。

この考え方を任意の線形独立なベクトル組 $\mathbf{a}_1, \dots, \mathbf{a}_k$ へ拡張したものが**グラム・シュミットの直交化法**（Gram–Schmidt process）である。ベクトルを順に処理し、各 $\mathbf{a}_j$ から、すでに構成した単位ベクトル $\mathbf{q}_1, \dots, \mathbf{q}_{j-1}$ への射影 Equation 33.2 を引き去り、残りを正規化して $\mathbf{q}_j$ を得る。その

結果として、$\mathrm{span}(\mathbf{a}_1, \ldots, \mathbf{a}_k)$ の正規直交基底が得られると同時に、**QR 分解**（QR factorization）$\mathbf{A} = \mathbf{QR}$ も得られる。ここで $\mathbf{Q}$ は正規直交列を持ち、$\mathbf{R}$ は対角成分が正の上三角行列である。この分解には線形独立性が不可欠である。それ以前のベクトルが張る空間にすでに含まれているベクトルからは残りがゼロになり、正規化できないからである。張る空間の正規直交基底だけが必要なら、残りがゼロになったベクトルを単に飛ばして先へ進めばよい。したがって有限個のベクトルからなる任意の生成系から正規直交基底が得られ、先ほど用いた存在の主張が正当化される。

数値計算ライブラリは鏡映に基づく別のアルゴリズムで QR 分解を計算しており、浮動小数点演算においてはるかに良好に振る舞う。その出力は符号の違いを除いてグラム・シュミット法と一致する。$\mathbf{Q}$ のある列とそれに対応する $\mathbf{R}$ の行の符号を同時に反転させても、積 $\mathbf{QR}$ は変わらないからである。$\mathbf{A}$ の列が線形独立であれば、不定性はこの符号反転のみに限られる。2 ステップのグラム・シュミット計算を `np.linalg.qr` と比較し、符号を除いて同じ列が得られることを確かめてみよう。

```
a1, a2 = np.array([3., 4]), np.array([1., 0])
q1 = a1 / np.linalg.norm(a1)
v = a2 - (q1 @ a2) * q1                    # remove the shadow on q1
q2 = v / np.linalg.norm(v)
Q_qr, R_qr = np.linalg.qr(np.stack([a1, a2], axis=1))
print(np.stack([q1, q2], axis=1).round(6))
print(Q_qr.round(6))
print(R_qr.round(6))
```

```
[[ 0.6  0.8]
 [ 0.8 -0.6]]
[[-0.6 -0.8]
 [-0.8  0.6]]
[[-5.  -0.6]
 [ 0.  -0.8]]
```

グラム・シュミット法は $(0.6, 0.8)$ と $(0.8, -0.6)$ を生成する一方、`np.linalg.qr` は両方の列の符号を反転させたものと、対角成分が -5 と -0.8 である $\mathbf{R}$ を返す。これは同じ基底であり、符号を除けば手計算で求めたドット積 5、0.6、0.8 とも一致している。浮動小数点演算では、引き算を実行する順序も重要になる。ほぼ線形従属な列に対しては、**古典的**（classical）グラム・シュミット法と呼ばれる上記の定式化は徐々に直交性を失っていく（5）。

33.1.3 高次元における類似度

ここまでの結果には、すべて $\mathbb{R}^2$ や $\mathbb{R}^3$ の図が伴っていた。ここからは図を描くことのできない空間へと移り、実際の 2 枚の画像がどれほど似ているか、そして無関係なベクトル同士が純粋な偶然によってどの程度の類似度を持つかを考える。Fashion-MNIST データセット（3.2 節）を `d2l.FashionMNIST` 経由で NumPy 配列として読み込み、各画像を 784 ピクセルの行ベクトルへ平坦化し、$[0, 1]$ にスケーリングする。

```
data = d2l.FashionMNIST()
X_train = data.train.data.numpy().reshape(-1, 784) / 255
y_train = data.train.targets.numpy()
X_test = data.val.data.numpy().reshape(-1, 784) / 255
y_test = data.val.targets.numpy()
X_train.shape, X_test.shape
```

```
((60000, 784), (10000, 784))
```

これら 70,000 枚の画像は、いまやそれぞれが $\mathbb{R}^{784}$ 内の 1 点である。

33.1.3.1 コサイン類似度

Equation 33.6 の内部にある比は、1.3 節で導入した**コサイン類似度**（cosine similarity）である。

$$\cos\theta = \frac{\mathbf{v}\cdot\mathbf{w}}{\|\mathbf{v}\|\,\|\mathbf{w}\|} \;\in\; [-1,1]. \tag{33.10}$$

コサイン類似度は長さを無視して方向だけを比較する。ある画像と、その明るさを 10% に落としたコピーはユークリッド距離では大きく離れているが、コサイン類似度は依然として 1 である。

このスケール不変性により、コサイン類似度は埋め込みに対する最近傍検索や、SimCLR (Chen et al. 2020) のような対照学習（contrastive learning）の目的関数における標準的な指標となっている。SimCLR では、同一画像の 2 つのビューから生成された埋め込みの方向を揃える。一方、元の Transformer のアテンション層は異なる選択をしており、$1/\sqrt{d}$ でスケーリングした**正規化なしの**ドット積を用いるため（その理由はまもなく導出する）、そこではベクトルの長さも情報を持つ。後発の多くのモデルでは、先にクエリとキーを正規化する（11.1.2.5 節）ことで、スコアを再びスケール付きコサインへと戻している。

33.1.3.2 ランダムな方向の概直交性

無関係な 2 枚の画像、あるいは無関係な 2 つの埋め込みを選んだとしよう。純粋な偶然によって、それらの間にどの程度のコサイン類似度が期待されるだろうか。比較の基準がなければ、観測されたコサインが意味のある大きさかどうかを判断できない。最も明快な基準は、空間のどの領域にも偏りを持たせず、単位球面上から一様ランダムに選んだ方向である。

命題（概直交性）。 単位ベクトル $\mathbf{u}\in\mathbb{R}^d$ を固定し、$\mathbf{v}$ を $\mathbb{R}^d$ の単位球面上の一様分布から抽出する。このとき $\cos\theta = \mathbf{u}\cdot\mathbf{v}$ は次を満たす。

$$\mathbb{E}[\cos\theta] = 0, \qquad \mathrm{Var}(\cos\theta) = \frac{1}{d}, \qquad \Pr\!\left(|\cos\theta| \ge \varepsilon\right) \le \frac{1}{d\,\varepsilon^2} \quad \text{for every } \varepsilon > 0.$$

証明。 球面上の一様分布は回転に対して不変であるため、$\mathbf{u} = \mathbf{e}_1$ となるように座標系を回転させてよく、このとき $\cos\theta = v_1$ となる。この分布は符号反転 $\mathbf{v}\mapsto -\mathbf{v}$ によっても変わらないため、$\mathbb{E}[v_1] = 0$ である。対称性により、すべての座標は同じ二次モーメント $\mathbb{E}[v_i^2]$ を持つ。$\sum_{i=1}^{d} v_i^2 = \|\mathbf{v}\|^2 = 1$ であるから、期待値をとると $d\,\mathbb{E}[v_1^2] = \mathbb{E}[\sum_{i=1}^{d} v_i^2] = 1$ となり、したがって $\mathrm{Var}(\cos\theta) = \mathbb{E}[v_1^2] = 1/d$ が得られる。裾の評価はチェビシェフの不等式 $\Pr(|X-\mathbb{E}[X]|\ge\varepsilon)\le \mathrm{Var}(X)/\varepsilon^2$（1.6 節）から直ちに従う。■

したがって、ランダムな 2 つの方向のなす角のコサインは典型的に $1/\sqrt{d}$ のオーダーになる。$d = 784$ では標準偏差が $1/28 \approx 0.036$ となり、画像空間内のランダムな 2 つの方向はほとんど常に直交から数度以内に収まる。チェビシェフの不等式は $|\cos\theta| \ge 0.1$ となる確率を $1/(784\cdot 0.01) \approx 0.13$ 以下と評価するが、真の確率は 0.01 未満である。チェビシェフの不等式が緩いのは分散しか使っていないためであり、より鋭い指数型の上界は 36.9 節で扱う。

低次元での様相はまったく異なる（Figure 33.6）。平面（$d = 2$）では、固定した方向とランダムな方向のなす角は $[0,\pi]$ 上で一様であるため、直交から 0.1 ラジアン以内に収まる確率と、共線から 0.1 ラジアン以内に収まる確率は等しい（それぞれ約 6%）。ところが**コサイン**は ± 1 の近くに集中する。$\cos\theta$ が $\theta = 0$ と $\theta = \pi$ の近くで平坦になるためであり、$|\cos\theta| > 0.9$ となる確率が約 0.29 であるのに対し、わずか 0.06 にすぎない（$|\cos\theta| < 0.1$ の場合）。$d = 3$ では

コサインは $[-1, 1]$ 上で一様になり、これは Archimedes にも知られていた古典的な結果である。$d = 4$ からは密度が 0 でピークを持つようになり、$d = 10$ では幅 $1/\sqrt{10} \approx 0.32$ の釣鐘型をなし、$d = 1000$ に至ると幅が約 0.03 の鋭いスパイクへと収縮する。

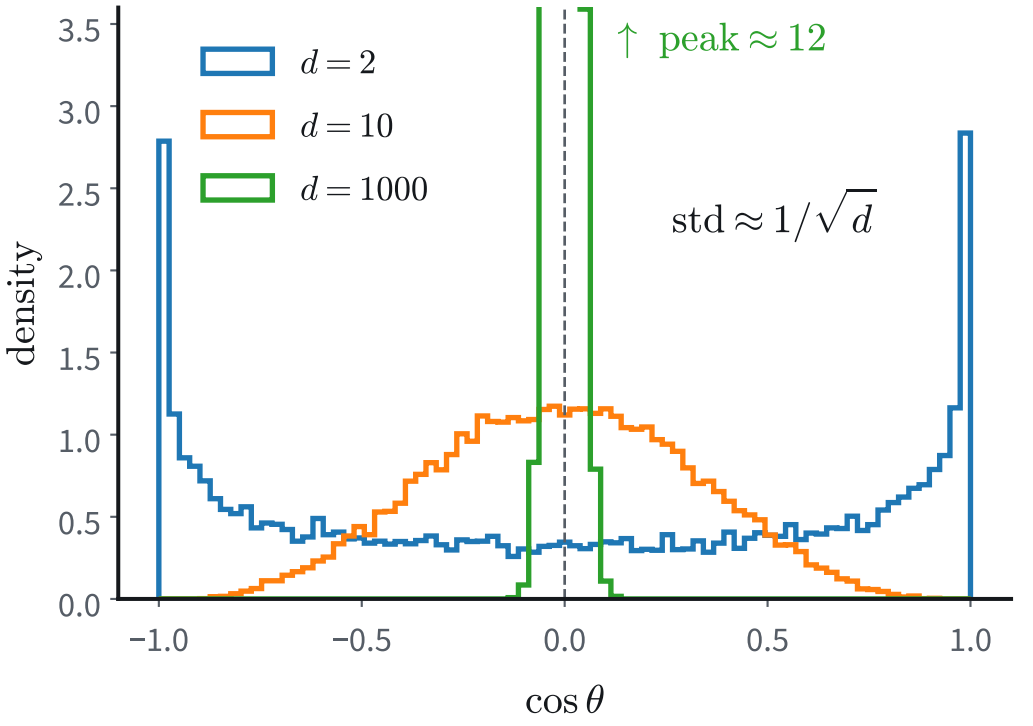


Figure 33.6: 次元 $d = 2$、10、1000 における独立なランダム単位ベクトル間のコサインのヒストグラム。$d = 2$ では、角度は一様であるにもかかわらずコサインが ± 1 付近に集中する。$d = 10$ では 0 の周りに釣鐘型をなし、$d = 1000$ では命題の通り幅 $\approx 1/\sqrt{d}$ の細いスパイクへと集中する。

$d = 784$ におけるこの等方的な基準をシミュレーションで確かめてみよう。標準ガウスベクトルの分布は回転不変であるため、それを正規化すると一様ランダムな単位ベクトルが得られる。同じコードセル内で、すぐ後のアテンションの議論で必要になる分散、すなわち $\mathbf{q} \cdot \mathbf{k}$ の分散（64 次元の独立な標準正規ベクトルの場合）もあわせて推定しておく。

```
rng = np.random.default_rng()
d = 784
U = rng.standard_normal((10000, d))
U /= np.linalg.norm(U, axis=1, keepdims=True)
V = rng.standard_normal((10000, d))
V /= np.linalg.norm(V, axis=1, keepdims=True)
q, k = rng.standard_normal((10000, 64)), rng.standard_normal((10000, 64))
np.std((U * V).sum(axis=1)), 1 / np.sqrt(d), np.var((q * k).sum(axis=1))
```

```
(np.float64(0.03602372011587368),
 np.float64(0.03571428571428571),
 np.float64(65.51162136338216))
```

経験標準偏差は $1/\sqrt{784} \approx 0.036$ と一致し、正規化なしのドット積の分散は次元である 64 に近い値となる。

しかし実データが等方的に見えることはめったにない。無関係な 2 枚の Fashion-MNIST 画像は、ほぼ直交するベクトルのように振る舞うだろうか。テスト画像からランダムなペアを 5,000 組サンプリングし、まず生のピクセルに対して、次に訓練画像の平均を引いた後でコサインを測ってみよう。

```
def cosines(A, B):
    """Cosine similarity between corresponding rows of A and B."""
    return (A * B).sum(axis=1) / (np.linalg.norm(A, axis=1)
                                  * np.linalg.norm(B, axis=1))

i, j = rng.integers(0, len(X_test), (2, 5000))
mean_image = X_train.mean(axis=0)
raw = cosines(X_test[i], X_test[j])
centered = cosines(X_test[i] - mean_image, X_test[j] - mean_image)
raw.mean(), raw.std(), centered.mean(), centered.std()
```

```
(np.float64(0.592208289545085),
 np.float64(0.17608271528452482),
 np.float64(-0.004695197414552555),
 np.float64(0.3390460995081151))
```

結果は似ても似つかない。生の画像の平均コサインは約 0.6、標準偏差は約 0.18 であり、0 ± 0.036 とは程遠い。理由は単純で、ピクセル輝度は非負であるため、すべての画像が $\mathbb{R}^{784}$ の同じ正象限（positive orthant）に属しており、負のドット積が生じえないからである。平均画像を引くとコサインの中心は 0 に移るが、その標準偏差は約 0.34 と依然として等方的な値のほぼ 10 倍もある。先ほどの命題によれば、0.34 という広がりはわずか 8〜9 次元におけるランダムな方向に相当する（$1/0.34^2$ は 8 と 9 の間にある）。コサインから判断する限り、中心化された画像群は 784 次元ではなく 10 次元未満の空間に収まる点群のように振る舞っており、これは 33.5 節で学ぶ低ランク構造の片鱗である。実用上、あるコサインが大きいと言えるのは、**同じ**データセットからとった無関係なペアの広がりよりも際立っている場合に限られる。関連するペアをその背景分布の上へと押し上げることこそが、埋め込みモデルを訓練する役割である。

33.1.3.3 アテンションスコアのスケール

ドット積アテンション（10.2 節）は、クエリ $\mathbf{q} \in \mathbb{R}^d$ とキー $\mathbf{k}_1, \mathbf{k}_2, \ldots$ の適合度を $\mathbf{q} \cdot \mathbf{k}_j / \sqrt{d}$ によってスコア付けし、そのスコアをソフトマックスに通す (Vaswani et al. 2017)。簡単な分散の計算から、$\sqrt{d}$ の由来が説明できる。全 $2d$ 個の成分（$\mathbf{q}$ と $\mathbf{k}$ の成分）が互いに独立で、平均 0、分散 1 を持つと仮定しよう。このとき各積 $q_i k_i$ は平均 0、分散 $\mathbb{E}[q_i^2]\,\mathbb{E}[k_i^2] = 1$ を持ち、異なる項同士は $\mathbb{E}[(q_i k_i)(q_j k_j)] = \mathbb{E}[q_i]\,\mathbb{E}[k_i]\,\mathbb{E}[q_j]\,\mathbb{E}[k_j] = 0$（$i \neq j$）より無相関である。したがって d 個の項の分散はそのまま足し合わされる。

$$\operatorname{Var}(\mathbf{q} \cdot \mathbf{k}) = \sum_{i=1}^{d} \operatorname{Var}(q_i k_i) = d.$$

ゆえに生のドット積の標準偏差は $\sqrt{d}$ となってヘッド次元とともに増大するが、$\sqrt{d}$ で割れば単位分散に戻る。このスケーリングがないと、ソフトマックスは飽和してほぼワンホットに近い重みになってしまう。$d = 64$ の場合、生のスコアの標準偏差は 8 であり、8 だけ離れた 2 つのスコ

アにはソフトマックス重み 0.9997 と 0.0003 が割り当てられる。$\sqrt{d} = 8$ で割ることで、重みは健全な 0.73 と 0.27 に戻る。幾何学からもまったく同じスケーリングが導かれる。各ベクトルの長さは約 $\sqrt{d}$ である一方、無関係な方向のなす角のコサインは約 $1/\sqrt{d}$ であるため、そのドット積の典型的な大きさは $\sqrt{d} \cdot \sqrt{d} \cdot (1/\sqrt{d}) = \sqrt{d}$ となる。

もっとも、無関係なキーが多数ある場合、概直交性があるからといってそれらを無視できるわけではない。無関係な個々のコサインのスケールは $1/\sqrt{d}$ であっても、独立な n 個のキーの中での**最大の**コサインは主要項で $\sqrt{2 \ln n / d}$ のように増大する。大規模な検索インデックスでは、このような背景ノイズによる最悪の一致度が $1/\sqrt{d}$ の数倍に達するため、真に関連するキーははるかに高い水準を超えなければならず、訓練によってまさにそれが実現される（1 でこの効果を具体的な数値で確認する）。

33.1.4 超平面と決定境界

線形分類器は、入力 $\mathbf{x}$ に対して 2 つの幾何学的な問いを投げかける。決定境界（decision boundary）のどちら側にあるか、そしてその境界からどれだけ離れているかである。本節を通して、この境界を $\mathbf{w} \cdot \mathbf{x} + b = 0$ と書く。

33.1.4.1 超平面と符号付き距離

平面上ではこの方程式は直線を定め、簡単な例から $\mathbf{w}$ と b の役割がわかる。$\mathbf{w} = (2, 1)$、$b = -1$ としよう。$2x + y - 1 = 0$ を満たす点は直線 $y = 1 - 2x$ をなす。この直線上のすべての点 $\mathbf{x}$ は $\mathbf{w} \cdot \mathbf{x} = -b$ を満たすため、$\mathbf{w}$ への影はすべて同じ符号付き長さ Equation 33.4 を持つ。

$$\frac{\mathbf{w} \cdot \mathbf{x}}{\|\mathbf{w}\|} = \frac{-b}{\|\mathbf{w}\|} = \frac{1}{\sqrt{5}}.$$

幾何学的には、この直線は $\mathbf{w}$ に垂直に走り、符号付き位置 $-b/\|\mathbf{w}\|$、すなわち原点からの距離 $|b|/\|\mathbf{w}\| = 1/\sqrt{5} \approx 0.45$ の位置でその方向を横切る。

高次元でも事情はまったく変わらない。$\mathbb{R}^d$ における**超平面**（hyperplane）とは、集合 $\{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = 0\}$（ただし $\mathbf{w} \neq \mathbf{0}$）のことであり、$\mathbf{w}$ をその**法線**（normal）ベクトル、b をその**オフセット**と呼ぶ。これは部分空間 $\mathbf{w} \cdot \mathbf{x} = 0$ を平行移動させたものにすぎず、その次元は $d - 1$ である（33.2.2.2 節）。すなわち、$\mathbb{R}^2$ では直線、$\mathbb{R}^3$ では平面、画像空間内では 783 次元の平坦な面となる。超平面は $\mathbb{R}^d$ を 2 つの**半空間**（half-space）$\mathbf{w} \cdot \mathbf{x} + b > 0$ と $\mathbf{w} \cdot \mathbf{x} + b < 0$ に分割する。b を変化させると、超平面は傾きを変えずに $\mathbf{w}$ に沿って平行移動する。

任意の点 $\mathbf{x}$ は超平面からどれだけ離れているだろうか。超平面上の任意の点 $\mathbf{x}_0$ をとると、$\mathbf{w} \cdot \mathbf{x}_0 = -b$ が成り立つ。$\mathbf{x}$ から超平面までの距離は、変位 $\mathbf{x} - \mathbf{x}_0$ の法線 $\mathbf{w}$ への影の符号付き長さであり、次の**符号付き距離**（signed distance）を与える。

$$\frac{\mathbf{w} \cdot (\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{w}\|} = \frac{\mathbf{w} \cdot \mathbf{x} + b}{\|\mathbf{w}\|}. \tag{33.11}$$

$\mathbf{x}_0$ が完全に消えていることに注目しよう。符号付き距離は $\mathbf{x}$、$\mathbf{w}$、b のみに依存する。その値は、$\mathbf{w}$ が向く側で正、反対側で負、超平面上ではゼロになる。先ほどの平面の例では、原点は符号付き距離 $-1/\sqrt{5}$ の位置にあり、点 $(1, 1)$ はスコア $\mathbf{w} \cdot \mathbf{x} + b = 2$ を持つため正の側の符号付き距離 $+2/\sqrt{5} \approx 0.89$ の位置にある。直線上でこの点に最も近い点は $(1, 1) - (2/5)(2, 1) = (0.2, 0.6)$ である。点 $(0, 1)$ のスコアは 0 であり、ちょうど直線上に乗っている（Figure 33.7）。

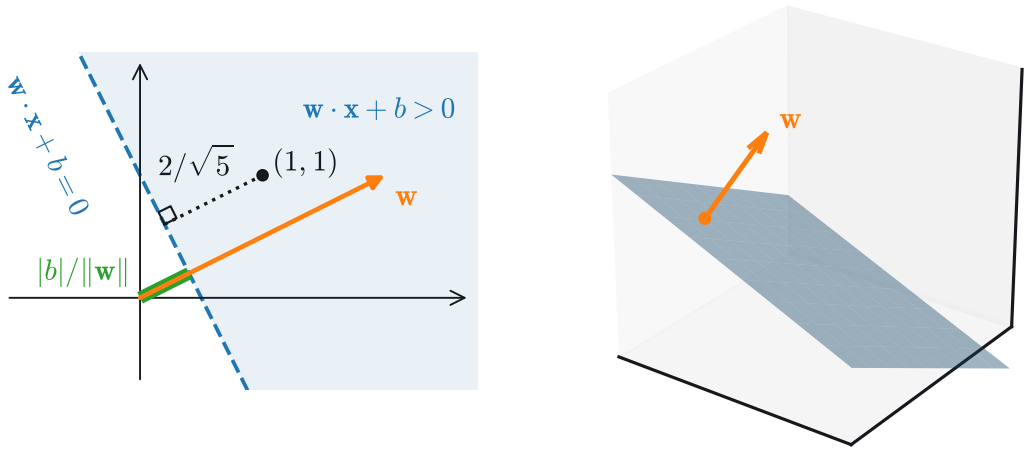


Figure 33.7: 超平面 $\mathbf{w}\cdot\mathbf{x}+b=0$（法線 $\mathbf{w}$）。左：$\mathbf{w}=(2,1)$、$b=-1$ のとき、超平面は直線 $2x+y=1$（原点からの距離 $1/\sqrt{5}$）であり、点 $(1,1)$ は符号付き距離 $2/\sqrt{5}$ の位置（$\mathbf{w}$ が向く側）にあり、色付きの半平面は $\mathbf{w}\cdot\mathbf{x}+b>0$ を表す。右：3 次元では同じ方程式が平面を定める。

ラベル付きデータ例 $(\mathbf{x}, y)$（ただし $y\in\{-1,+1\}$）に対しては、それが**正しい**側にあるかどうかも知りたい。符号付き距離にラベルを掛けたものは**マージン** $y\,(\mathbf{w}\cdot\mathbf{x}+b)/\|\mathbf{w}\|$ を定め、分類が正しいときに正となり、点が本来の半空間の奥深くにあるほど大きくなる。$(\mathbf{w}, b)\to(c\mathbf{w}, cb)$（任意の $c>0$）と再スケーリングしても超平面とマージンは変わらない一方、生のスコア $\mathbf{w}\cdot\mathbf{x}+b$ は c 倍になる。したがって、生のスコアを最大化するような学習目的関数は、いずれも $\mathbf{w}$ のスケールに制約を課さなければならない。

33.1.4.2 最近傍平均分類器

2 つのクラスを分離するには、どの超平面を選ぶべきだろうか。最も単純な規則には反復的な訓練すら不要で、平均がより近いほうのクラスへ $\mathbf{x}$ を割り当てればよい。驚くべきことに、この規則自体がただ 1 つの超平面になる。2 つのクラス平均を $\bar{\mathbf{x}}_0$ と $\bar{\mathbf{x}}_1$ としよう。両方の距離の二乗を展開すると、二次項 $\|\mathbf{x}\|^2$ が両辺で相殺される。

$$\begin{aligned}\|\mathbf{x}-\bar{\mathbf{x}}_1\|^2 < \|\mathbf{x}-\bar{\mathbf{x}}_0\|^2 &\iff -2\,\mathbf{x}\cdot\bar{\mathbf{x}}_1+\|\bar{\mathbf{x}}_1\|^2 < -2\,\mathbf{x}\cdot\bar{\mathbf{x}}_0+\|\bar{\mathbf{x}}_0\|^2\\ &\iff (\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0)\cdot\mathbf{x}-\tfrac{1}{2}\big(\|\bar{\mathbf{x}}_1\|^2-\|\bar{\mathbf{x}}_0\|^2\big)>0.\end{aligned}$$

恒等式 $\|\mathbf{a}\|^2-\|\mathbf{c}\|^2=(\mathbf{a}-\mathbf{c})\cdot(\mathbf{a}+\mathbf{c})$ を使うと、最後の不等式は次の定義のもとで $\mathbf{w}\cdot\mathbf{x}+b>0$ となる。

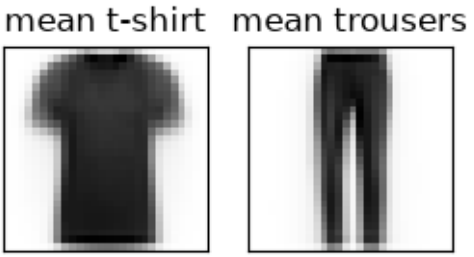
$$\mathbf{w}=\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_0,\qquad b=-\mathbf{w}\cdot\tfrac{1}{2}\left(\bar{\mathbf{x}}_0+\bar{\mathbf{x}}_1\right).$$

つまり「最近傍平均」とは垂直二等分超平面にほかならない。その法線は一方の平均から他方の平均へと向き、2 つの平均の中点をまっすぐ横切る。手動で調整すべきしきい値はなく、この規則はピクセルの一様な再スケーリングに対して不変である。すべてのピクセルに $c>0$ を掛けると $\mathbf{w}\to c\mathbf{w}$、$b\to c^2 b$ となり、すべてのスコアが c^2 倍されるだけで予測は 1 つも変わらない（2 ではこの不変性を回転と平行移動へ拡張し、逆に予測を**変えてしまう**ピクセルごとの再スケーリングを構成する）。

この分類器を T シャツとズボンで試す前に、2 つのクラス平均を観察し、そのなす角を計算してみよう。

```
X0, X1 = X_train[y_train == 0], X_train[y_train == 1]    # t-shirts, trousers
mean_0, mean_1 = X0.mean(axis=0), X1.mean(axis=0)
d2l.show_images([mean_0.reshape(28, 28), mean_1.reshape(28, 28)], 1, 2,
                titles=['mean t-shirt', 'mean trousers'], cmap='Greys')
np.degrees(angle(mean_0, mean_1))
```

```
np.float64(33.20917997376519)
```



どちらの平均画像もぼやけてはいるが、ひと目で判別できる。両者のなす角は約 33°（コサインにして 0.84）にすぎず、直交からは程遠い。どちらの衣類も画像中央の明るい領域を占めているからである。2 つの平均の角度がこれほど近いと悲観的になるかもしれないが、最近傍平均分類器（nearest-mean classifier）は平均同士の直交性を必要とせず、両者の差のみに依拠する。T シャツとズボンのテスト画像全 2,000 枚で評価してみよう。

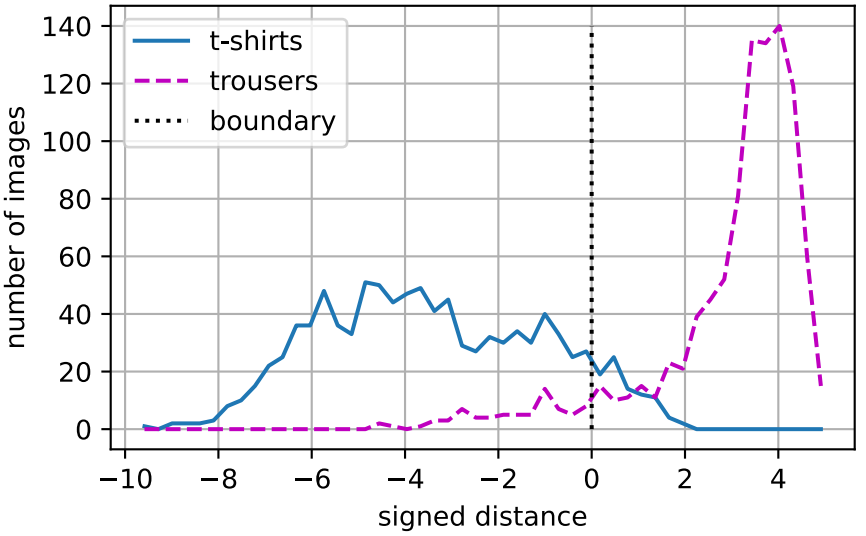
```
w = mean_1 - mean_0
b = -w @ (mean_0 + mean_1) / 2
mask = (y_test == 0) | (y_test == 1)
X2, y2 = X_test[mask], y_test[mask]
predictions = X2 @ w + b > 0                              # True: trousers
np.mean(predictions == (y2 == 1))
```

```
np.float64(0.9155)
```

この規則はテスト画像の約 92% を正しく分類し、誤った側に振り分けられたのはわずか 169 枚（全 2,000 枚中）である。各クラスの平均をとり、1 つの超平面のどちら側に各テスト画像が入るかを調べただけでこの結果が得られた。

残りの誤分類がどこから生じるかを理解するため、すべてのテスト画像を境界までの符号付き距離 Equation 33.11 という 1 つの数値に圧縮し、クラスごとにヒストグラムを描いてみよう。

```
dist = (X2 @ w + b) / np.linalg.norm(w)         # signed distance to boundary
bins = np.linspace(dist.min(), dist.max(), 51)
counts = [np.histogram(dist[y2 == c], bins)[0] for c in (0, 1)]
top = max(counts[0].max(), counts[1].max())
d2l.plot([bins[:-1]] * 2 + [[0, 0]], counts + [[0, top]], 'signed distance',
         'number of images', fmts=('-', 'm--', 'k:'),
         legend=['t-shirts', 'trousers', 'boundary'], figsize=(5, 3))
```



2 つのクラスは明確に分かれた山をなすが、それぞれの裾が境界をまたいでいる。T シャツは -4 付近を中心とする幅広い山をなす一方、ズボンは $+4$ 付近に鋭いピークを持ち、ゼロに向かって長い裾を引いている。平均的には、各クラスはその訓練平均の符号付き距離 $\mp\|\mathbf{w}\|/2 \approx \mp 3.2$ の近くに位置する。$\mathbf{w}$ に沿った各クラスの標準偏差を計算し、誤った側に入るデータ例の個数を数えて、この様子を定量化しよう。

```
(dist[y2 == 0].std(), dist[y2 == 1].std(),
 int((dist[y2 == 0] > 0).sum()), int((dist[y2 == 1] < 0).sum()))
```

```
(np.float64(2.356225995048113), np.float64(1.6627004538100882), 101, 68)
```

$\mathbf{w}$ に沿った標準偏差は T シャツが約 2.4、ズボンが約 1.7 であり、T シャツ 101 枚とズボン 68 枚が誤分類されている。各クラスの中心から境界までの距離が標準偏差の 2 倍未満であるため、数パーセントの誤り率が生じるのは当然の結果である。ここに中点規則の本質的な限界がある。2 つのクラス平均しか見ておらず、どちらの点群の大きさや形状も考慮する術を持たないのである。では、T シャツ対シャツやプルオーバー対コートのように、見た目がはるかに似通った難しいペアに適用するとどうなるだろうか。

```
# t-shirt/trouser, t-shirt/shirt, pullover/coat
for a, c in [(0, 1), (0, 6), (2, 4)]:
    m_a = X_train[y_train == a].mean(axis=0)
    m_c = X_train[y_train == c].mean(axis=0)
    w_ac, b_ac = m_c - m_a, -(m_c - m_a) @ (m_a + m_c) / 2
    pair = (y_test == a) | (y_test == c)
    acc = np.mean((X_test[pair] @ w_ac + b_ac > 0) == (y_test[pair] == c))
    dist_ac = (X_test[pair] @ w_ac + b_ac) / np.linalg.norm(w_ac)
    sd_a, sd_c = (dist_ac[y_test[pair] == k].std() for k in (a, c))
    names = data.text_labels([a, c])
    print(f'{names[0]:>8} vs {names[1]:<9}accuracy {acc:.3f}   '
          f'||w||/2 {np.linalg.norm(w_ac) / 2:.2f}   '
          f'std along w {sd_a:.2f}, {sd_c:.2f}')
```

```
 T-shirt vs trouser  accuracy 0.915   ||w||/2 3.23   std along w 2.36, 1.66
 T-shirt vs shirt    accuracy 0.786   ||w||/2 1.86   std along w 1.77, 2.38
pullover vs coat     accuracy 0.716   ||w||/2 1.24   std along w 1.88, 2.09
```

正解率は急落し、T シャツ対シャツでは約 79%、プルオーバー対コートでは約 72% まで落ち込む。残りの出力列がその理由を物語っている。各訓練平均は境界から距離 $\|\mathbf{w}\|/2$ の位置にあり、重なり具合を左右するのは、この距離を $\mathbf{w}$ に沿った各クラスの標準偏差を単位として測った値である。T シャツとズボンでは、その隔たりは標準偏差の 1.4 倍から 1.9 倍ある。ところが難しい 2 組のペアではわずか 0.6 倍から 1.0 倍に縮まるため、中点に置いた境界が両クラスの中心部をまっぷたつに横切ってしまう。その処方箋は、クラスのばらつきが大きい方向を避けるように $\mathbf{w}$ を傾けることであり、これがフィッシャーの線形判別（Fisher's linear discriminant、4）の考え方である。ソフトマックス回帰はさらに一歩進めて傾きとオフセットの両方をデータから直接学習し、深層ネットワークは入力そのものを変換して、単一の超平面でクラスがほぼ分離可能になる表現を獲得する。

33.1.4.3 超平面の積層としての層

ニューラルネットワークの線形層は、多数の超平面判定を並列に実行している。第 i 成分（行列・ベクトル積 $\mathbf{Ax}$ の成分）は、第 i 行（$\mathbf{A}$ の行）と $\mathbf{x}$ のドット積、すなわちスコア $\mathbf{a}_i \cdot \mathbf{x}$（法線 $\mathbf{a}_i$ に対するスコア）であり、バイアスベクトルを加えると各成分が超平面スコアになる。たとえば、3.1 節における Fashion-MNIST 上の 10 クラスソフトマックス層は、10 個の法線 $\mathbf{w}_1, \dots, \mathbf{w}_{10}$ と 10 個のオフセットを保持している。スコアが最大のクラスを予測するため、クラス i とクラス j を分ける決定境界は、両者のスコアが並ぶ超平面 $(\mathbf{w}_i - \mathbf{w}_j)\cdot\mathbf{x} + (b_i - b_j) = 0$ となる。

具体例として、本章で通しで用いる行列 $\mathbf{A} = \left(\begin{smallmatrix} 1 & 2 \\ -1 & 3 \end{smallmatrix}\right)$ と入力 $\mathbf{x} = (2, -1)$ をとろう。第 1 行からは $1\cdot 2 + 2\cdot(-1) = 0$ が得られ、$\mathbf{x}$ は原点を通る法線 $(1, 2)$ の超平面上にちょうど乗っていることがわかる。第 2 行からは $-2 - 3 = -5$ が得られ、$\mathbf{x}$ は法線 $(-1, 3)$ の超平面の負の側に符号付き距離 $-5/\sqrt{10} \approx -1.58$ だけ離れて位置する。両方の結果を並べれば $\mathbf{Ax} = (0, -5)$ となる。

$\mathbf{Ax}$ を行ごとに眺めれば線形層が超平面判定の積層であることがわかるが、2 つの本質的な問いが未解決のまま残る。この変換はどの入力をゼロに潰すのか、そしてその作用はいつ逆転できるのかという問いである。33.2 節では、まったく同じ積を列ごとに読み解くことでこの両方の問いに答える。

33.1.5 まとめ

ドット積は、平面上の長さ、角度、射影をまったく同じ公式のまま任意の次元へと拡張する。直線や部分空間への射影は、射影先の空間と直交する残差によって特徴づけられる唯一の最近接点を与え、このただ 1 つの幾何学的事実からコーシー・シュワルツの不等式（高次元の角度を矛盾なく定義可能にする）と最小二乗法の正規方程式が導かれる。グラム・シュミットの直交化法は任意の生成系を正規直交基底へと変換し、QR 分解の因子 $\mathbf{Q}$ を与える。$\mathbb{R}^d$ におけるランダムな方向は典型的に $1/\sqrt{d}$ のオーダーのコサインを持ってほぼ直交する一方、実際の画像データセットは等方的とは程遠いため、観測されたコサインは同じ分布からとった無関係なペアと比較して初めて意味を持つ。最後に、最近傍平均分類器は 2 つのクラス平均を二等分する単一の超平面を定め、T シャツ対ズボンでは約 92% の正解率に達するが、境界の法線方向に沿ったクラスの広がりが効いてくる見た目の似たペアでは 72%〜79% に低下する。

33.1.6 演習問題

1. [code] **ランダムベクトル間のコサイン**。

1. 標準ガウスベクトルを正規化することで、独立なランダム単位ベクトルのペアを $1{,}000$ 組（$d = 10{,}000$ 次元）サンプリングし、各ペアのコサインを計算せよ。経験標準偏差を概直交性の命題による予測値 $1/\sqrt{d} = 0.01$ と比較し、観測された最大の $|\cos\theta|$ を報告せよ。
2. ある検索システムは 1 つのクエリを n 個のキーと比較し、最大のコサインを持つキーを返す。クエリとキーを $d = 256$ 次元の独立なランダム単位ベクトルとしてモデル化せよ。n 個のコサインの最大値を $n = 10$、10^3、10^5 に対して計算し、主要項の評価である $\sqrt{2\ln n / d}$（n 個の独立な分散 $1/d$ のガウス乱数の最大値に対する評価）と比較せよ。10^5 個のキーの中で最上位に一致した無関係なキーは、どの程度の大きさのコサインに達するか。2 つの埋め込みが関連していると判定するためのしきい値は、n と d に応じてどのようにスケーリングすべきか。

2. **最近傍平均と不変性。**最近傍平均規則は、$\mathbf{x}$ をクラス 1 に割り当てる条件を $\mathbf{w}\cdot\mathbf{x} + b > 0$（ただし $\mathbf{w} = \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0$、$b = -\mathbf{w}\cdot\tfrac{1}{2}(\bar{\mathbf{x}}_0 + \bar{\mathbf{x}}_1)$）とする。
   1. すべての訓練画像とテスト画像に同じ変換 $\mathbf{x} \mapsto c\mathbf{Q}\mathbf{x} + \mathbf{z}$（ただし $c > 0$、$\mathbf{Q}$ は $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ を満たす正方行列、$\mathbf{z}$ は固定されたベクトル）を施し、クラス平均と b を再計算するとする。予測が一切変わらないことを示し、これを本文で議論した再スケーリング $\mathbf{w} \to c\mathbf{w}$、$b \to c^2 b$ と関連づけよ。
   2. 各ピクセルを単位分散に標準化すると、$\mathbf{x}$ は $\mathbf{D}\mathbf{x}$（$\mathbf{D}$ は正の対角成分を持つ対角行列）に置き換わる。$\mathbb{R}^2$ において、このような $\mathbf{D}$ のもとで予測クラスが反転する 2 つのクラス平均とテスト点を構成せよ。
   3. 最近傍平均規則のもとで q 個のクラスがあるとき、クラス i に割り当てられる領域が $q-1$ 個の半空間 $(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)\cdot\mathbf{x} + b_{ij} > 0$ $(j \neq i)$ の共通部分になることを示し、b_{ij} を求めよ。これをソフトマックス層のペアごとの境界 $(\mathbf{w}_i - \mathbf{w}_j)\cdot\mathbf{x} + (b_i - b_j) = 0$ と比較せよ。重みとバイアスをどのように選べば、ソフトマックス層は最近傍平均分類器を再現するか。

3. **直交射影行列と斜射影行列。**
   1. $\mathbf{P}$ が $\mathbf{P}^2 = \mathbf{P}$ と $\mathbf{P}^\top = \mathbf{P}$ を満たすとする。すべての $\mathbf{x}$ に対して、$\mathbf{P}\mathbf{x}$ が $\mathbf{P}$ の列空間に属し、$\mathbf{x} - \mathbf{P}\mathbf{x}$ がその列空間と直交することを示せ。正規直交基底を用いることなく、ピタゴラスの定理から $\mathbf{P}\mathbf{x}$ が列空間内の最近接点であることを導け。
   2. 行列 $\mathbf{P}_{\mathrm{o}} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ は $\mathbf{P}_{\mathrm{o}}^2 = \mathbf{P}_{\mathrm{o}}$ を満たすが対称ではない。$\mathbf{x} = (1, 1)$ に対して、$\mathbf{P}_{\mathrm{o}}\mathbf{x}$ と、同じ列空間（第 1 座標軸）への $\mathbf{x}$ の直交射影、およびそれぞれと $\mathbf{x}$ との距離を計算せよ。$\mathbf{P}_{\mathrm{o}}$ はどの方向に沿って点を射影するか。
   3. 任意の行列 $\mathbf{P}$（$\mathbf{P}^2 = \mathbf{P}$）が、すべての $\mathbf{x}$ を $\mathbf{P}\mathbf{x} + (\mathbf{x} - \mathbf{P}\mathbf{x})$ と分解し、第 1 項が $\mathbf{P}$ の列空間に、第 2 項がその零空間 $\{\mathbf{z} : \mathbf{P}\mathbf{z} = \mathbf{0}\}$ に属することを示せ。これら 2 つの部分空間が直交するための必要十分条件が $\mathbf{P}^\top = \mathbf{P}$ であることを示せ。

4. [code] **境界を傾ける。**●平均の差をとった法線 $\Delta = \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0$ は、各クラスの画像がその平均の周りにどう広がっているかを無視している。訓練セット上での 2 つのクラス内共分散行列の平均を $\mathbf{C}$ としよう。フィッシャーの線形判別 (Fisher 1936) は、次式を最大化する法線 $\mathbf{w}$ を選ぶ。

$$J(\mathbf{w}) = \frac{(\mathbf{w}\cdot\Delta)^2}{\mathbf{w}^\top\mathbf{C}\mathbf{w}},$$

これは射影されたクラス平均間の距離の二乗を、射影のクラス内分散で割った量である。$\mathbf{C}$ が正定値（$\mathbf{v}^\top\mathbf{C}\mathbf{v} > 0$ がすべての $\mathbf{v} \neq \mathbf{0}$ に対して成り立つ）のとき、逆行列 $\mathbf{C}^{-1}$ が存在し、最大点は $\mathbf{w} \propto \mathbf{C}^{-1}\Delta$ となる。（本問では 33.2.2.3 節の逆行列、33.3.2.2 節の正定値性、および 33.5.3.2 節の条件数を用いる。）

1. $J(c\mathbf{w}) = J(\mathbf{w})$ がすべての $c \neq 0$ に対して成り立つことを示せ。$\mathbf{C} = s^2\mathbf{I}$ のとき、コーシー・シュワルツの不等式を適用して Δ が J を最大化することを示せ。
2. 本文中の T シャツとズボンの配列 `X0` と `X1` を使って $\mathbf{C}$ を計算し、平均ピクセル分散を $\bar{s}^2 = \mathrm{tr}(\mathbf{C})/784$ とおけ。これらの画像では端のピクセルがほぼ一定であるため、$\mathbf{C}$ の条件数は約 10^9 に達する。$\gamma \in \{10^{-4}, 10^{-2}, 1, 10^2, 10^4\}$ に対して正則化した法線 $\mathbf{w}_\gamma = (\mathbf{C} + \gamma \bar{s}^2 \mathbf{I})^{-1}\Delta$ を作り、境界を 2 つの平均の中点に置いて訓練正解率とテスト正解率の両方を評価せよ。次に、各クラスから訓練画像を 1,000 枚ずつホールドアウトし、残りの画像で $\mathbf{C}$ とクラス平均を再計算して、ホールドアウトセット上の検証正解率によって γ を選び、その $\mathbf{w}_\gamma$ を用いて符号付き距離のヒストグラムを再びプロットせよ。
3. なぜ $\mathbf{w}_\gamma$ は Δ の方向へ（$\gamma \to \infty$ の極限で）揃うのか。また、$\bar{s}^2$ で正則化項をスケーリングし、$\mathbf{I}$ の固定定数倍をそのまま足さないのはなぜか。

5. [code] **浮動小数点におけるグラム・シュミット法。**●グラム・シュミットの直交化法は、代数的に同値な 2 通りの順序で実行できる。古典的グラム・シュミット法（CGS）は、2 ベクトルの例と同様に、$\mathbf{q}_j$ を $\mathbf{a}_j - \sum_{i<j} (\mathbf{q}_i \cdot \mathbf{a}_j)\,\mathbf{q}_i$ の正規化によって計算する。修正グラム・シュミット法（MGS）は $\mathbf{v} = \mathbf{a}_j$ と初期化し、$\mathbf{v} \leftarrow \mathbf{v} - (\mathbf{q}_i \cdot \mathbf{v})\,\mathbf{q}_i$ を $i = 1, \dots, j-1$ について順に更新してから $\mathbf{v}$ を正規化する。

   1. 厳密な演算においては CGS と MGS が同一のベクトルを生成することを示せ。
   2. 両方のアルゴリズムを実装し、ほぼ線形従属な列を持つヒルベルト行列 $\mathbf{H}_n$（成分は $1/(i+j-1)$、$n = 4, 6, 8, 10, 12$）に対して実行せよ。フロベニウスノルム $\|\mathbf{Q}^\top\mathbf{Q} - \mathbf{I}\|_F$ を n に対して CGS、MGS、`np.linalg.qr` のそれぞれでプロットせよ。各手法について、どの n で $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top$ が射影行列として機能しなくなるか（$\|\mathbf{P}^2 - \mathbf{P}\|_F$ によって測れ）。
   3. 各列に対して CGS の引き算ステップを 2 回適用せよ（1 回目のパスで残ったベクトルに対して 2 回目のパスを実行する）。得られる列は、1 回のみの CGS や MGS と比べてどれほど正規直交に近いか。1 回目のパスに入るベクトルと 2 回目のパスに入るベクトルそれぞれに含まれる $\mathbf{q}_1, \dots, \mathbf{q}_{j-1}$ 方向の成分の大きさを比較して、精度が改善する理由を説明せよ。

6. [extended] **ほぼ直交する方向のパッキング。**●●$\mathbb{R}^d$ には互いに直交する方向は高々 d 個しか収まらないが、ペアごとのコサインを小さく保ったままなら、それよりはるかに多くの方向を詰め込むことができる。Elhage et al. (2022) はこの幾何学を用いて、幅 d の層が d 個を超える特徴量をほぼ直交する方向として**重ね合わせ**の形で表現できる仕組みを説明している。ここで扱われる特徴量は**疎**であり、各特徴量はごく一部の入力でしか活性化しないため、一度に読み出す必要があるのは少数にとどまる。$\mathbf{u}_1, \dots, \mathbf{u}_n$ を $\mathbb{R}^d$ 内の単位ベクトルとし、それらを行として並べた行列を $\mathbf{U} \in \mathbb{R}^{n \times d}$ とし、ペア間の最悪の干渉を $\mu = \max_{i \neq j} |\mathbf{u}_i \cdot \mathbf{u}_j|$ とおく。

   1. $\sum_{i,j} (\mathbf{u}_i \cdot \mathbf{u}_j)^2$ を座標で展開し、これが $\sum_{k,l} \big(\sum_i u_{ik} u_{il}\big)^2$、すなわち $d \times d$ 行列 $\mathbf{U}^\top\mathbf{U}$ の各成分の二乗和に等しいことを示せ。d 個の対角項と、$\sum_{k=1}^{d} a_k^2 \geq \big(\sum_k a_k\big)^2 / d$ の形のコーシー・シュワルツの不等式を用い、$\sum_k (\mathbf{U}^\top\mathbf{U})_{kk} = n$ に注意してこの和を下から評価せよ。さらに元の和を上から $n + n(n-1)\mu^2$ で抑えることで、次の **Welch の下界** (Welch 1974) を導出せよ。

$$\mu^2 \geq \frac{n - d}{d\,(n-1)}.$$

   $d = 32$ のもとで $n = 64$ と $n = 512$ に対してこの下界の値を求めよ。

2. すべてのペアにわたるユニオンバウンドにより、独立なランダム方向は $\mu \le \varepsilon$ を n が $e^{d\varepsilon^2/4}$ のオーダーになるまで保つことが示される（36.9 節）。$d = 32$、$n \in \{64, 128, 256, 512\}$ に対してランダム単位ベクトルの μ を計算し、Welch の下界と比較せよ。

3. 最適化によって直接 μ を小さくせよ。すなわち、$\sum_{i \neq j} (\mathbf{u}_i \cdot \mathbf{u}_j)^p$ を $p = 10$ のような偶数のべきに対して、$\mathbf{U}$ 上の勾配ステップ（各ステップ後にすべての行を再正規化する）によって最小化せよ。$d \in \{16, 32, 48, 64\}$ と $n \in \{d, 2d, 4d, 8d, 16d\}$ に対し、ランダムなベクトル、最適化したベクトル、および Welch の下界について μ を n に対してプロットせよ。

4. これらの曲線を使って、各 d に対して最大の n（最適化した方向が $\mu \le 0.2$ を保てる上限）を推定せよ。この容量は d の増加に伴って、ランダム方向の増加率 $e^{d\varepsilon^2/4}$ と比べどのように増大するか。もし一部の特徴量が他より重要であったり、多くの特徴量が同時に活性化したりする場合、結論はどのように変わるか。

33.2 線形変換としての行列

33.1 節では、行列とベクトルの積 $\mathbf{A}\mathbf{x}$ を 1 行ずつ読み解いた。第 i 成分は第 i 行と $\mathbf{x}$ のドット積であり、1 回の超平面判定にあたる。この行ごとの見方は、1 つの入力ベクトルに対して行列が何を計算するかは教えてくれるが、入力空間全体に一度に何をするかは見えにくい。とりわけ、その写像が情報を保つのか捨ててしまうのか、またその作用を元に戻せるのかどうかはわからない。

この問いは、ニューラルネットワークの層が入力より狭くなった瞬間に現れる。Fashion-MNIST 画像の 784 ピクセルを 256 個の隠れ特徴量へ写す全結合層（fully connected layer）は、256×784 行列である。出力が保持する数値の個数は入力より少ないため、空間は必然的に潰される。まもなく厳密に数えるように、少なくとも $784 - 256 = 528$ 個の独立な入力方向が 0 に送られるため、非線形性（nonlinearity）を通る前の段階で、異なる画像がまったく同じ特徴量を生み出しうる。

正方行列であっても情報を失うことがある。本章で通しで用いる行列 $\mathbf{A} = \left(\begin{smallmatrix} 1 & 2 \\ -1 & 3 \end{smallmatrix}\right)$ を考えよう。この行列は $(2, -1)$ を $(0, -5)$ に送り、$(0, -5)$ をそのまま $(2, -1)$ に戻す別の行列が存在する。実際、その 2 つ目の行列はあらゆる入力に対して $\mathbf{A}$ の作用を打ち消すため、$\mathbf{A}$ は何も失わない。これを行列 $\mathbf{B} = \left(\begin{smallmatrix} 2 & 1 \\ 4 & 2 \end{smallmatrix}\right)$ と対比してみよう。こちらは $(1, 1)$ と $(0, 3)$ の両方を $(3, 6)$ に送る。出力 $(3, 6)$ だけを与えられても、どちらの入力から生じたのかを判別するすべはないため、どのような行列も $\mathbf{B}$ を逆向きに戻すことはできない。

線形変換（linear map）を元に戻せるかどうかによって、層が情報を捨てるか、連立一次方程式（linear system）が一意な解を持つか、そして変数変換（change of variables）のもとで確率密度がどのようにスケーリングされるか（36.2.3.1 節）が決まる。この問いに明快に答えるため、視点を行から列へと移そう。行列の第 j 列とは、第 j 座標軸の行き先にほかならず、それらの像だけで写像全体が定まる。線形独立（linearly independent）な列がいくつあるかを数えれば入力の何次元が写像で生き残るかがわかり、変換後の単位立方体の符号付き体積を測れば、次元が失われたかどうかを検出する 1 つの数値が得られる。この 2 つの考え方を、まずはすべての主張を図として描ける 2 次元で展開し、続いて n 次元へと拡張する。

本節の軸となる結果は 3 つある。生き残る次元と潰れる次元の釣り合い、正方行列が逆行列を持つための厳密な条件、そして 2 つの写像を合成したときに体積の拡大率が従う規則である。一般の n 次元における行列式（determinant）の構成（33.2.3.2 節）は最も技術的な部分であり、初読の際は流し読みしてかまわない。

```
import numpy as np
```

33.2.1 線形変換

33.2.1.1 座標軸の像としての列

行列は無限に多くの入力ベクトルに作用するが、有限個の数値の表として格納されている。たった 4 つの数値で、なぜ平面上のあらゆるベクトルに対する出力が定まるのだろうか。2×2 行列 $\mathbf{A}$（成分は p, q, r, s）をとり、$\mathbf{x} = (x, y)$ に適用してみよう。x が掛かる項と y が掛かる項をそれぞれまとめると、次の式が得られる。

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} p & q \\ r & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} px + qy \\ rx + sy \end{bmatrix} = x \begin{bmatrix} p \\ r \end{bmatrix} + y \begin{bmatrix} q \\ s \end{bmatrix} = x\,\mathbf{A}\mathbf{e}_1 + y\,\mathbf{A}\mathbf{e}_2.$$

したがって、$\mathbf{A}\mathbf{x}$ とは $\mathbf{A}$ の 2 つの列を、$\mathbf{x}$ の座標を重みとして重み付き和をとったものにすぎない。$\mathbf{A}\mathbf{e}_1 = (p, r)$ かつ $\mathbf{A}\mathbf{e}_2 = (q, s)$ であるから、各列は標準基底ベクトル（standard basis vector）$\mathbf{e}_1 = (1, 0)$ と $\mathbf{e}_2 = (0, 1)$ の像である。この 2 本の軸の行き先さえわかれば、平面上の他のすべてのベクトルの像は完全に定まる。

通しで用いる行列（$p = 1$、$q = 2$、$r = -1$、$s = 3$）では、2 本の軸は $\mathbf{A}\mathbf{e}_1 = (1, -1)$ と $\mathbf{A}\mathbf{e}_2 = (2, 3)$ に移る。入力 $\mathbf{v} = (2, -1)$ は $2\mathbf{e}_1 - \mathbf{e}_2$ と表せるため、その像は列のまったく同じ線形結合になる。

$$\mathbf{A}\mathbf{v} = 2\,(1, -1) - (2, 3) = (0, -5).$$

行ごとに検算しても $(1, 2) \cdot (2, -1) = 0$ および $(-1, 3) \cdot (2, -1) = -5$ となり、結果が一致することが確かめられる。どちらの見方でも同じ積が計算されるが、列ごとの見方に立てば $\mathbf{A}$ が平面全体に一度に何をするかが見て取れる。Figure 33.8 にこの幾何学的意味を示す。$\mathbf{e}_1$ と $\mathbf{e}_2$ が張る空間である単位正方形は 2 つの列が張る平行四辺形へと運ばれ、グリッド上の点 $\mathbf{v}$ は変換後のグリッドの対応する点 $\mathbf{A}\mathbf{v}$ に着地する。

同じ列による描像は任意の次元で成り立つ。$m \times n$ 行列は $\mathbb{R}^n$ のベクトルを $\mathbb{R}^m$ のベクトルへ写し、次の意味で**線形**（linear）である。

$$\mathbf{A}(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\,\mathbf{A}\mathbf{x} + \beta\,\mathbf{A}\mathbf{y} \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \text{ and } \alpha, \beta \in \mathbb{R}.$$

言葉で言えば、重み付き和の像は、それぞれの像の同じ重み付き和に等しい。任意の入力を $\mathbf{x} = \sum_j x_j \mathbf{e}_j$ と書けば $\mathbf{A}\mathbf{x} = \sum_j x_j\,\mathbf{A}\mathbf{e}_j$ となり、ここで $\mathbf{e}_j$ を掛ける操作は第 j 列を取り出すことにほかならない。逆に、この線形性の規則を満たすすべての写像 $f : \mathbb{R}^n \to \mathbb{R}^m$ は、第 j 列を $f(\mathbf{e}_j)$ とする行列として書ける。なぜなら、$f(\mathbf{x}) = f(\sum_j x_j \mathbf{e}_j) = \sum_j x_j f(\mathbf{e}_j)$ が成り立つからである。したがって、$\mathbb{R}^n$ から $\mathbb{R}^m$ への線形変換と $m \times n$ 行列は、同じ対象を記述する 2 つの表現にほかならない。

Figure 33.8 のグリッドが変換後もグリッドの姿を保つのも、線形性のおかげである。あらゆる直線 $\mathbf{p} + t\mathbf{u}$ は $\mathbf{A}\mathbf{p} + t\,\mathbf{A}\mathbf{u}$ に写り、これもまた直線（あるいは $\mathbf{A}\mathbf{u} = \mathbf{0}$ のときは 1 点）になる。さらに、原点は決して動かない（$\mathbf{A}\mathbf{0} = \mathbf{0}$）。

ここまでの議論に行列が正方である必要はまったくない。2×3 行列は $\mathbb{R}^3$ を $\mathbb{R}^2$ 内の 3 つの列が張る空間（span）へと写し、少なくとも 2 つの列が線形独立である限り出力平面全体を埋め尽くす。逆に、3×2 行列は平面を $\mathbb{R}^3$ の中へと写し、その出力は 2 つの列に応じて原点を通る平面、直線、または原点のみをなす。ニューラルネットワークのすべての全結合層はこのような長方形の写像であり、その後にバイアスによるシフトと非線形性が続く。

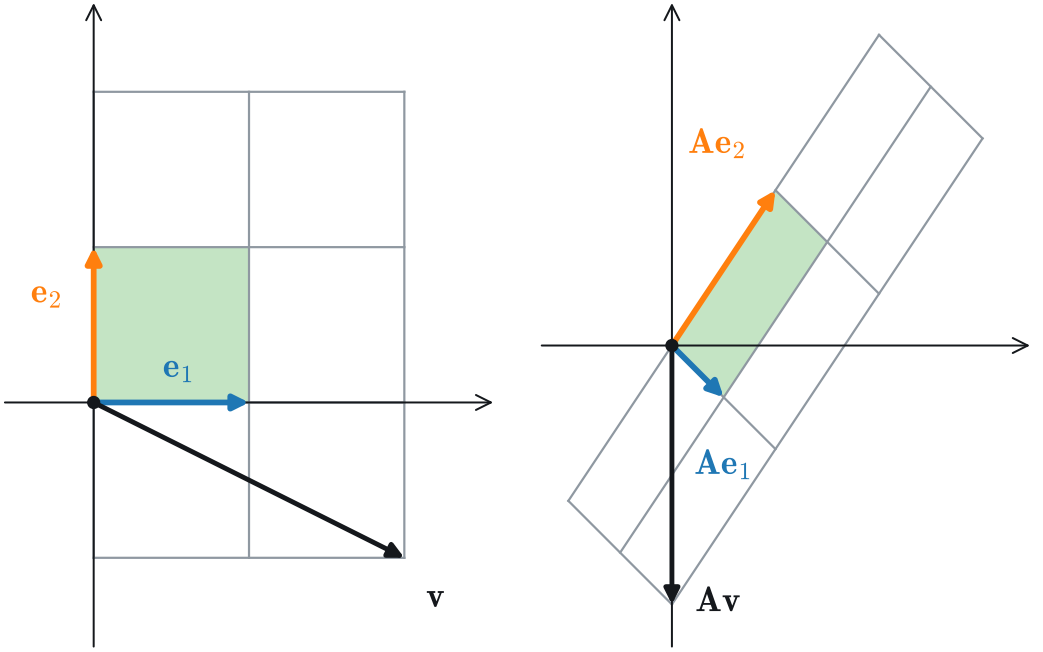


Figure 33.8: 平面に作用する通しで用いる行列 $\mathbf{A} = \left(\begin{smallmatrix} 1 & 2 \\ -1 & 3 \end{smallmatrix}\right)$。(a) 入力グリッド、網掛けの単位正方形、基底ベクトル、および入力 $\mathbf{v} = (2, -1)$。(b) その像。$\mathbf{e}_1$ は $(1, -1)$ に、$\mathbf{e}_2$ は $(2, 3)$ に、$\mathbf{v}$ は $(0, -5)$ に移る。直線は直線のまま保たれ、原点は動かず、等間隔の点は等間隔のまま保たれる。網掛けの正方形は網掛けの平行四辺形に写り、その面積から行列式が導かれる。パネル (b) は (a) の 3 分の 1 の縮尺で描かれている。

視覚的な直感を養うため、Figure 33.9 に 6 つの 2×2 行列を単位正方形への作用によって示す。反時計回りの 90° 回転は $\mathbf{e}_1$ を $\mathbf{e}_2$ に、$\mathbf{e}_2$ を $-\mathbf{e}_1$ に送るため、その列は $(0,1)$ と $(-1,0)$ になる。水平方向のせん断（shear）は $\mathbf{e}_1$ を動かさずに $\mathbf{e}_2$ を $(1,1)$ へと傾け、すべての水平線を高さに比例して横へ滑らせる。最後の 2 つのパネルは質的に異なり、次元を 1 つ失って正方形全体、ひいては平面全体を 1 本の直線へと潰してしまう。

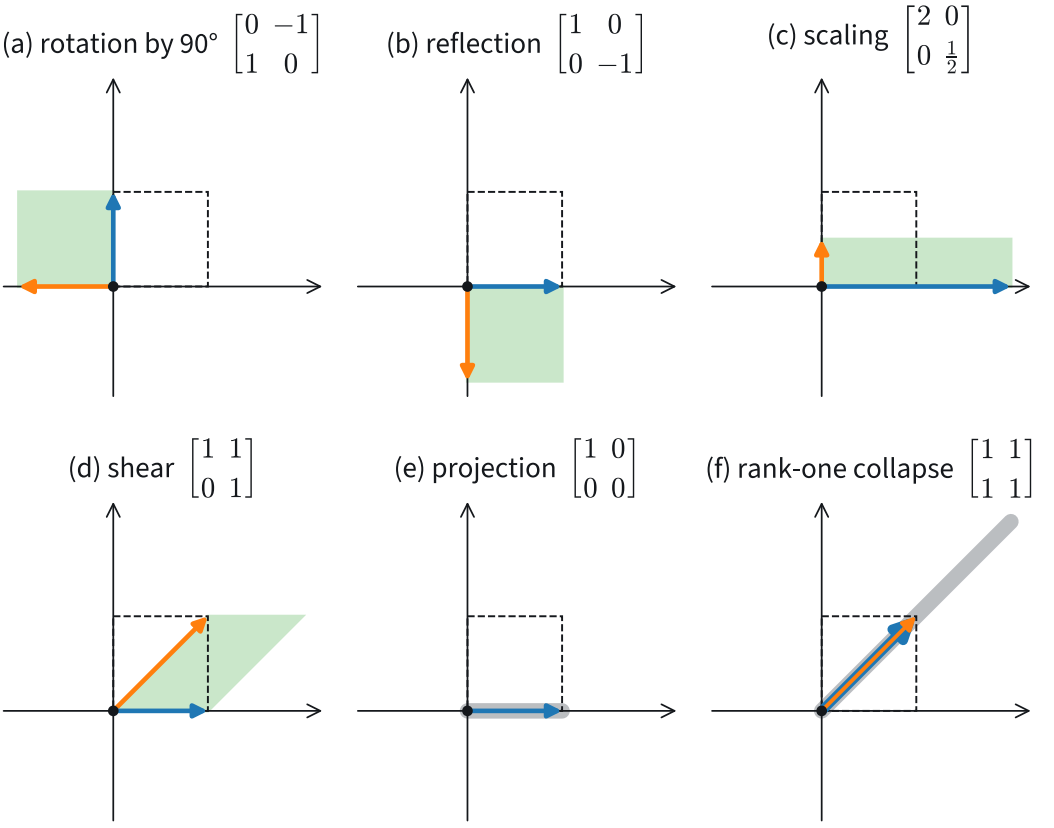


Figure 33.9: 2×2 行列が単位正方形（破線）を変換する 6 通りの様子。各パネルで、矢印は $\mathbf{e}_1$（青）と $\mathbf{e}_2$（オレンジ）の像である各列を表し、網掛け領域は正方形の像を表す。パネル (e) と (f) は平面を直線へと潰しており、(f) では両方の列が $(1,1)$ である。

このギャラリーには、おなじみの操作が 1 つ見当たらないことに気づくだろう。平行移動（translation）$\mathbf{x} \mapsto \mathbf{x} + \mathbf{c}$（ただし $\mathbf{c} \neq \mathbf{0}$）である。原点を動かしてしまうため、これは線形ではない。したがって、ニューラルネットワークにおけるアフィン層 $\mathbf{W}\mathbf{x} + \mathbf{b}$ は、線形変換の後に平行移動を施したものである。

33.2.1.2 合成

非線形性を挟まずに 2 つの線形層を直列に並べると、1 つの線形変換に縮約される。その合成された写像を表すのはどの行列だろうか。Figure 33.9 の 2 つの具体的な写像、すなわち 90° 回転 $\mathbf{R}_{90}$ と（縦軸を傾けることにちなんで名付けた）せん断 $\mathbf{T}$ で確かめてみよう。

$$\mathbf{R}_{90} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{T} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

この 2 つの行列は本章全体を通して繰り返し登場する。それぞれの順序で標準基底ベクトルがどこへ移るかを追跡しよう。先にせん断を施すと、$\mathbf{e}_1$ は $(1,0)$ にとどまり、続く回転によって $(0,1)$ に送られる。一方、$\mathbf{e}_2$ はせん断で $(1,1)$ に移り、回転で $(-1,1)$ に移る。したがって、「せん断してから回転」の列は $(0,1)$ と $(-1,1)$ になる。逆の順序では、回転が $\mathbf{e}_1$ を $(0,1)$ に送り、せん断がそれを $(1,1)$ に動かす。また、回転が $\mathbf{e}_2$ を $(-1,0)$ に送り、せん断はそれを動かさない。ゆえに、2 つの合成写像は次のようになる。

$$\text{shear, then rotate: } \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}, \qquad \text{rotate, then shear: } \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}.$$

Figure 33.10 から明らかなように、この 2 つは同じ写像ではない。

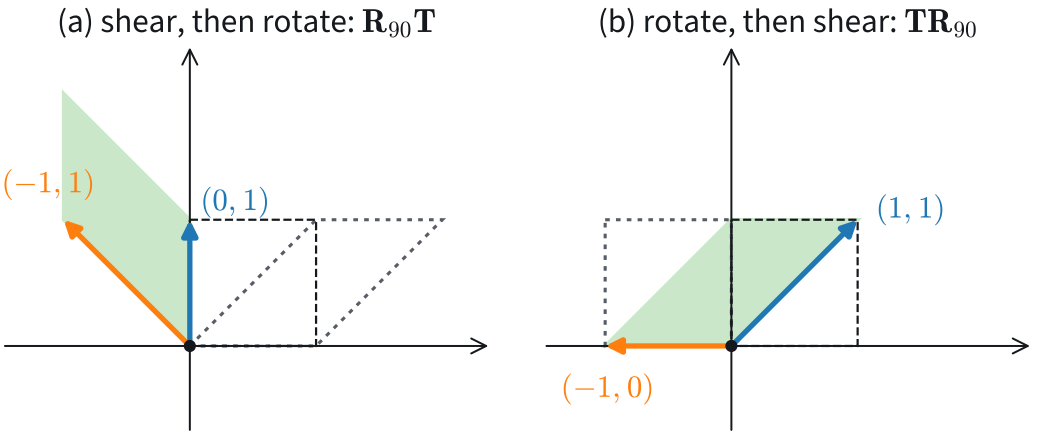


Figure 33.10: 合成の順序は結果を変える。(a) せん断 $\mathbf{T}$（点線の枠）に続けて回転 $\mathbf{R}_{90}$ を適用すると $\mathbf{R}_{90}\mathbf{T}$ になる。(b) 回転（点線の枠）に続けてせん断を適用すると $\mathbf{T}\mathbf{R}_{90}$ になる。ラベルは $\mathbf{e}_1$（青）と $\mathbf{e}_2$（オレンジ）の最終的な像を示す。行列積は常に右から左へと読む。

このような合成を計算する代数規則が行列積（matrix multiplication）である。次元が整合する任意の 2 つの行列 $\mathbf{A}$ と $\mathbf{B}$ について、まず $\mathbf{B}$ を $\mathbf{v}$ に適用し、次にその結果へ $\mathbf{A}$ を適用して、第 i 成分を調べてみよう。

$$\big(\mathbf{A}(\mathbf{B}\mathbf{v})\big)_i = \sum_j a_{ij}(\mathbf{B}\mathbf{v})_j = \sum_j \sum_k a_{ij} b_{jk} v_k = \sum_k \Big(\sum_j a_{ij} b_{jk}\Big) v_k = \big((\mathbf{A}\mathbf{B})\mathbf{v}\big)_i. \tag{33.12}$$

2 つの写像を順に適用することは、単一の行列 $\mathbf{AB}$ を適用することとまったく同じであり、その (i,k) 成分は $\mathbf{A}$ の第 i 行と $\mathbf{B}$ の第 k 列のドット積である。順序に注意しよう。積 $\mathbf{AB}$ は「先に $\mathbf{B}$ を適用し、次に $\mathbf{A}$ を適用する」ことを意味し、関数の合成 $f(g(x))$ と同じく右から左へと読む。また、$\mathbf{v} = \mathbf{e}_j$ を Equation 33.12 に代入すると有用な列の規則も得られる。すなわち、$\mathbf{AB}$ の第 j 列は $\mathbf{A}\mathbf{b}_j$ であり、これは $\mathbf{B}$ の第 j 列の $\mathbf{A}$ による像にほかならない。この事実は行列式を調べる際に活躍する。

積を写像の合成として捉えれば、一見奇妙に見えるその定義にも納得がいく。「行と列の積」という公式は恣意的な約束事ではなく、2 つの線形変換を合成した結果そのものなのである。代数的性質もここから直ちに従う。行列積が結合法則 $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$ を満たす（associative）のは、

関数の合成が結合法則を満たすからである。すなわち、両辺ともまず **C**、次に **B**、最後に **A** を適用する。また、一般に可換（commutative）でないのは、2 つの幾何学的操作を逆の順序で施せば通常は異なる結果になるからである。NumPy では、行列積に `@` 演算子を用いる。回転とせん断（以下のコードでは `R90` と `S` に格納）の 2 通りの合成が再現されることを確かめてみよう。

```
R90 = np.array([[0.0, -1.0], [1.0, 0.0]])  # rotation by 90 degrees
S = np.array([[1.0, 1.0], [0.0, 1.0]])     # shear
print(R90 @ S)
print(S @ R90)
```

```
[[ 0. -1.]
 [ 1.  1.]]
[[ 1. -1.]
 [ 1.  0.]]
```

もちろん、可換になる行列の組も存在する。平面上の 2 つの回転は可換である。角度 ϕ だけ回転してから ψ だけ回転すれば、どちらの順序でも $\phi+\psi$ の回転になるからである。2 つの対角行列も、それぞれ座標軸を独立にスケーリングするだけなので可換である。演習問題 3 では、2×2 行列のうち $\mathbf{R}_{90}$ と可換なものをすべて特徴づける。

33.2.1.3 直交行列

線形変換の中には、あらゆる長さとあらゆる角度を保ったまま空間を剛体的に動かすものがある。どの行列がこの性質を持つだろうか。平面における角度 ϕ の反時計回りの回転を考えよう。この回転は $\mathbf{e}_1$ を $(\cos\phi, \sin\phi)$ に、$\mathbf{e}_2$ を $(-\sin\phi, \cos\phi)$ に送るため、その行列は次のようになる。

$$\mathbf{R}_{\phi} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

その列は単位ベクトルとしての長さ $\sqrt{\cos^2\phi + \sin^2\phi} = 1$ を持ち、ドット積は $-\cos\phi\sin\phi + \sin\phi\cos\phi = 0$ である。すなわち列は正規直交（orthonormal）しており、これは 33.1 節で $\mathbf{R}_{\phi}^{\top}\mathbf{R}_{\phi} = \mathbf{I}$ と書いた条件である。$\phi = 90^\circ$ とおけば $\mathbf{R}_{90}$ が復元され、これは $(3,2)$ を $(-2,3)$ に送るが、どちらのベクトルも長さは $\sqrt{13}$ である。鏡映（reflection）$\mathrm{diag}(1,-1)$（x 軸に関する折り返し）も同様に正規直交する列を持つ。

任意の次元において、正方行列 $\mathbf{Q}$ の列が正規直交しているとき、すなわち次を満たすとき、その行列を**直交行列**（orthogonal matrix）と呼ぶ。

$$\mathbf{Q}^{\top}\mathbf{Q} = \mathbf{I}.$$

この定義には注意すべき微妙な点が 2 つある。第 1 に、列が互いに垂直であるだけでは不十分である。$\mathrm{diag}(1, 10)$ は長さ 1 と 10 の互いに垂直な列を持つが、90° をなす $(1,1)$ と $(1,-1)$ の間の角度を $\arccos(-99/101) \approx 169^\circ$ へと曲げてしまう（1）。第 2 に、「直交」という名称は正方行列に限って用いられる。縦長の $n \times k$ 行列（$k < n$）は、列が正規直交していれば長さを保つものの、$\mathbb{R}^k$ をより大きな空間 $\mathbb{R}^n$ に埋め込むため、両側逆行列を持たない（33.2.2.3 節）。

直交変換はすべてのドット積を保つ。

$$(\mathbf{Q}\mathbf{x})\cdot(\mathbf{Q}\mathbf{y}) = (\mathbf{Q}\mathbf{x})^{\top}(\mathbf{Q}\mathbf{y}) = \mathbf{x}^{\top}\mathbf{Q}^{\top}\mathbf{Q}\,\mathbf{y} = \mathbf{x}^{\top}\mathbf{y} = \mathbf{x}\cdot\mathbf{y}.$$

$\mathbf{y} = \mathbf{x}$ とおけば $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$ となるため、長さは変わらない。さらに、角度はドット積と長さだけで決まるため、角度も変わらない。埋め込みベクトルの集合全体に同じ直交行列を掛けて回転させても、あらゆるペアのコサイン類似度はまったく同じままである。

定義式 $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ は、転置 $\mathbf{Q}^\top$ を $\mathbf{Q}$ の**後に**適用すれば元に戻ることを表している。正方行列 $\mathbf{Q}$ については、実は $\mathbf{Q}^\top$ を $\mathbf{Q}$ の**前に**適用しても打ち消されること、すなわち $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ が成り立ち、したがって $\mathbf{Q}$ の行も正規直交であり $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ となる。これは定義だけからは自明ではなく、33.2.2.3 節で証明する正方行列の一般定理から導かれる。

2 次元では、直交行列は回転と鏡映に限られる。第 1 列は単位ベクトル $(\cos\phi, \sin\phi)$ でなければならず、これに垂直な第 2 列の単位ベクトルは $\pm(-\sin\phi, \cos\phi)$ の 2 通りしか残らない。正の符号をとれば回転 $\mathbf{R}_\phi$ になる。負の符号をとれば $\mathbf{R}_\phi\,\mathrm{diag}(1, -1)$、すなわち x 軸に関する鏡映の後に回転を施した写像になり、これ自身も原点を通るある直線に関する鏡映である。後ほど見るように、行列式の符号を見れば回転と鏡映を一目で区別できる（33.2.3.3 節）。

33.2.1.4 テンソルとアインシュタインの総和規約

Transformer では、自己アテンションがバッチ内のすべての系列とすべてのアテンションヘッドにわたって、あらゆるクエリとあらゆるキーを比較する。バッチを b、ヘッドを h、クエリ位置を i、キー位置を t、ヘッド内の特徴量次元を j とラベル付けすると、アテンションスコアは次の式で与えられる。

$$s_{bhit} = \sum_{j} q_{bhij}\, k_{bhtj},$$

ここで、q_{bhij} と k_{bhtj} は 4 軸のクエリテンソルとキーテンソルの成分である（1.1 節）。各スコアは特徴量軸 j に沿ったドット積であり、残る 4 つの添字のあらゆる組み合わせに対して計算される。通常の行列記法にはこの 4 添字の縮約（contraction）をすっきり書く方法がなく、「最後の 2 軸を転置したバッチ処理版の行列積」と呼んでしまうと、実際にどの軸について和をとっているのかが見えにくくなる。

一歩引いて眺めると、ここまでに出会ったすべての積がまったく同じパターンに従っていることに気づく。ドット積、行列とベクトルの積、行列積、そして正方行列の対角成分の和である**トレース** $\mathrm{tr}(\mathbf{A})$ は、いずれも次のように書ける。

$$\mathbf{v}\cdot\mathbf{w} = \sum_i v_i w_i, \qquad (\mathbf{A}\mathbf{v})_i = \sum_j a_{ij} v_j, \qquad (\mathbf{A}\mathbf{B})_{ik} = \sum_j a_{ij} b_{jk}, \qquad \mathrm{tr}(\mathbf{A}) = \sum_i a_{ii}.$$

トレースはここで初めて登場したが、33.3 節で行列の固有値の和として再び現れる。これら 4 つの式はいずれも、成分どうしを掛け合わせ、2 回現れる添字について和をとっている。**アインシュタイン記法**（Einstein notation）はこのパターンを規約にしたものであり、総和記号を省略し、繰り返されるすべての添字について自動的に和をとる。この規則のもとでは、$\mathbf{v}\cdot\mathbf{w} = v_i w_i$ および $(\mathbf{A}\mathbf{B})_{ik} = a_{ij} b_{jk}$ と書ける。

アテンションスコアの例を見ると、古典的な規約だけでは不十分な場面がわかる。バッチとヘッドの添字 b と h は両方の因子に現れるが、和をとって消すのではなく出力に残したいからである。数値計算ライブラリの `einsum` は、残す添字を明示的に指定させることでこれを解決する。入力の添字を並べた文字列、矢印 `->`、出力の添字を渡すと、右辺で省略された添字について和がとられる。4 つの基本的な積をこの記法で表現し、通しで用いる行列で確かめてみよう。

```
A = np.array([[1.0, 2.0], [-1.0, 3.0]])    # the running matrix
v = np.array([2.0, -1.0])
(np.einsum('i,i->', v, v),         # dot product: v.v
 np.einsum('ij,j->i', A, v),       # matrix-vector product: Av
 np.einsum('ij,jk->ik', R90, S),   # matrix product: R90 S
 np.einsum('ii->', A))             # trace: tr(A)
```

```
(np.float64(5.0),
 array([ 0., -5.]),
 array([[ 0., -1.],
        [ 1.,  1.]]),
 np.float64(4.0))
```

$\operatorname{tr}\mathbf{A} = 1 + 3 = 4$ を含め、すべての出力が先ほどの手計算と一致する。同じ記法を使えば、アテンションスコアもわずか 1 行で書ける。以下では、これを `einsum` で計算し、転置したキーとのバッチ行列積と一致することを確かめ、積和演算の総数を数える。

```
rng = np.random.default_rng()
queries = rng.standard_normal((2, 3, 4, 8))   # (batch, heads, queries, width)
keys = rng.standard_normal((2, 3, 5, 8))      # (batch, heads, keys, width)
scores = np.einsum('bhij,bhtj->bhit', queries, keys)
(scores.shape, np.allclose(scores, queries @ keys.swapaxes(-1, -2)),
 scores.size * queries.shape[-1])             # multiply-adds
```

```
((2, 3, 4, 5), True, 960)
```

出力の形状は $(2, 3, 4, 5)$ であり、系列とヘッドの組ごとに 4×5 のスコア表が 1 つ得られる。2 つの式の計算結果は一致し、この縮約には $2 \cdot 3 \cdot 4 \cdot 5 \cdot 8 = 960$ 回の積和演算を要する。これは 5 つの添字すべてのサイズを掛け合わせた積にほかならない。

コードだけでなく、添字記法を使えば 1 行の簡潔な証明も難なく書ける。任意の $\mathbf{A} \in \mathbb{R}^{m \times n}$ と $\mathbf{B} \in \mathbb{R}^{n \times m}$ に対して、$\mathbf{AB}$ と $\mathbf{BA}$ はどちらも正方行列であり、それらのトレースは次を満たす。

$$\operatorname{tr}(\mathbf{AB}) = a_{ij} b_{ji} = b_{ji} a_{ij} = \operatorname{tr}(\mathbf{BA}). \tag{33.13}$$

中央の変形は、添字には一切触れずに 2 つのスカラー値の積の順序を入れ替えただけである。驚くべきことに、たとえ $\mathbf{AB} \neq \mathbf{BA}$ であっても、さらには 2 つの積のサイズが異なっていても、積のトレースは 2 つの因子の順序によらない。ここから直ちに導かれる帰結の 1 つは、固有値を学ぶ際に重要になる。$\mathbf{V}$ が逆行列 $\mathbf{V}^{-1}$ を持つとき（33.2.2.3 節）、$\operatorname{tr}(\mathbf{V}\mathbf{A}\mathbf{V}^{-1}) = \operatorname{tr}(\mathbf{A}\mathbf{V}^{-1}\mathbf{V}) = \operatorname{tr}(\mathbf{A})$ が成り立つ。$\mathbf{V}\mathbf{A}\mathbf{V}^{-1}$ は同じ線形変換 $\mathbf{A}$ を別の座標系で表したものであるから、トレースは選んだ座標系に依存する見かけの量ではなく、写像そのものに備わった固有の性質である。後で見るように、行列式もこの不変性を共有している（33.2.3.3 節）。

実用上の注意点を 1 つ挙げておく。`einsum` の式で 3 つ以上のテンソルを組み合わせるとき、文字列が指定するのは数学的な結果だけであり、2 つずつの縮約を実行する順序までは指定しないが、その順序によって計算速度に大きな差が生じることがある。たとえば $n \times n$ 行列では、$(\mathbf{AB})\mathbf{v}$ の計算には約 n^3 回の積和演算がかかるのに対し、$\mathbf{A}(\mathbf{B}\mathbf{v})$ ならわずか $2n^2$ 回で済む。

33.2.2 ランクと可逆性

33.2.2.1 列空間と零空間

行列は実際にどのような出力を生み出せ、どのような入力を消し去ってしまうのだろうか。その両方を一度に見るため、特異行列の例を調べてみよう。

$$\mathbf{B} = \begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix}.$$

その列は $\mathbf{b}_1 = (2, 4)$ と $\mathbf{b}_2 = (1, 2)$ であり、一方は他方のちょうど 2 倍になっている（$\mathbf{b}_1 = 2\mathbf{b}_2$）。$\mathbf{B}$ のすべての出力はその列の重み付き和であるから、あらゆる出力は $\mathbf{b}_2$ のスカラー倍へと潰れてしまう。

$$a_1\mathbf{b}_1 + a_2\mathbf{b}_2 = (2a_1 + a_2)\,\mathbf{b}_2.$$

したがって、$\mathbf{B}$ は 2 次元平面全体を、1 本の直線 $y = 2x$（点 $(1, 2)$ を通る直線）の上へと送り込む。必然的に、異なる入力がその直線上で衝突し、$(1, 1)$ と $(0, 3)$ はどちらも $(3, 6)$ に着地する。線形性により、両者の差 $(1, -2)$ はそれぞれの出力の差へと写るはずであり、$\mathbf{B}(1, -2) = \mathbf{0}$ となる。行ごとの見方からも同じ理由が確かめられる。$\mathbf{B}$ の 2 つの行 $(2, 1)$ と $(4, 2)$ はどちらも $(1, -2)$ に垂直であるため、どちらの超平面判定も 0 を返す。Figure 33.11 にこの 2 本の直線を並べて示す。

この例は、すべての $m \times n$ 行列 $\mathbf{A}$ に付随する 2 つの基本部分空間（subspace）を具体的に示している。**列空間**（column space）は $\operatorname{col}\mathbf{A}$ と書き、$\mathbf{A}$ の列が張る空間である。$\mathbf{Ax}$ は $x_1, \dots, x_n$ を重みとする列の重み付き結合であるから、列空間は到達可能なすべての出力の集合にぴったり一致する。**零空間**（null space）は**核**（kernel）とも呼ばれ、$\ker\mathbf{A}$ と書いて、$\mathbf{A}$ が 0 に送るすべての入力の集合を表す。

$$\ker\mathbf{A} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{0}\}.$$

これは $\mathbb{R}^n$ の部分空間である。実際、$\mathbf{Ax} = \mathbf{Ay} = \mathbf{0}$ ならば $\mathbf{A}(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\mathbf{Ax} + \beta\mathbf{Ay} = \mathbf{0}$ が成り立つ。$\mathbf{B}$ においては、列空間は出力平面内の直線 $y = 2x$ であり、零空間は入力平面内の $(1, -2)$ を通る直線である。（どちらも $\mathbb{R}^2$ に属しているのは $\mathbf{B}$ が正方行列であることによる偶然にすぎず、一般には $\ker\mathbf{A} \subseteq \mathbb{R}^n$ かつ $\operatorname{col}\mathbf{A} \subseteq \mathbb{R}^m$ である。）

この例から、次の 3 つの一般的な事実がそのまま導かれる。

1. **零空間は、すべての行に垂直な入力全体からなる。**行ごとの見方では、$(\mathbf{Ax})_i$ は第 i 行と $\mathbf{x}$ のドット積であるため、$\mathbf{Ax}$ のすべての成分が 0 になるのは、$\mathbf{x}$ がすべての行に直交するとき、かつそのときに限る。
2. **列が線形従属（linearly dependent）であることは、零空間が非ゼロベクトルを含むことと同値である。**非自明な従属関係 $\sum_j x_j\mathbf{a}_j = \mathbf{0}$ が列 $\mathbf{a}_j$ の間に成り立つとは、$\mathbf{Ax} = \mathbf{0}$ を満たす非ゼロベクトル $\mathbf{x}$ が存在するということそのものであり、この等式を逆に読めば逆向きも従う。$\mathbf{B}$ では、関係式 $\mathbf{b}_1 - 2\mathbf{b}_2 = \mathbf{0}$ から零空間のベクトル $(1, -2)$ が得られる。
3. **2 つの入力が同じ出力を生むのは、それらの差が零空間のベクトルであるとき、かつそのときに限る。**実際、$\mathbf{Ax} = \mathbf{Ay}$ は $\mathbf{A}(\mathbf{x} - \mathbf{y}) = \mathbf{0}$ と同値である。$\mathbf{B}$ が $(3, 6)$ に写すすべての入力の集合は、平行移動した直線 $(1, 1) + t(1, -2)$ であり、これは点 $(0, 3)$ を $t = -1$ で通る。$\ker\mathbf{A}$ が非ゼロベクトルを含む限り、異なる入力が衝突し、逆行列は存在しえない。

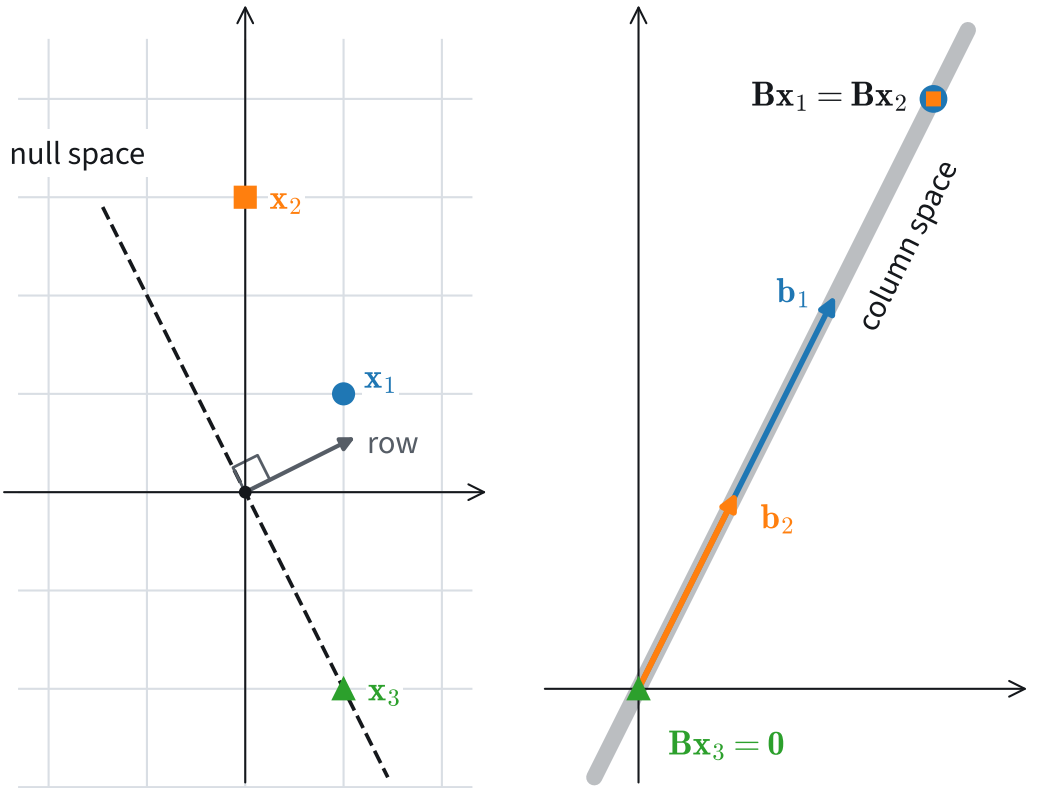


Figure 33.11: 平面を潰す行列 $\mathbf{B} = \left(\begin{smallmatrix} 2 & 1 \\ 4 & 2 \end{smallmatrix}\right)$。(a) 入力平面において、$(1, -2)$ を通る破線の零空間の直線は行の方向 $(2, 1)$ に垂直であり、3 つの入力 $\mathbf{x}_1 = (1, 1)$、$\mathbf{x}_2 = (0, 3)$、および零空間の直線上の $\mathbf{x}_3 = (1, -2)$ が印されている。(b) 出力平面（同じ縮尺）では、あらゆる出力は列空間である直線 $y = 2x$（$\mathbf{b}_1$ と $\mathbf{b}_2$ が張る空間）に着地する。$\mathbf{x}_1$ と $\mathbf{x}_2$ はともに $(3, 6)$ に写り、$\mathbf{x}_3$ は原点に潰れる。異なる入力が同じ像を共有するため、逆行列は存在しえない。

33.2.2.2 ランク

線形変換によって入力の何次元が生き残るだろうか。行列の**ランク**とは、その列空間の次元、言い換えれば線形独立な列の最大本数である。

$$\operatorname{rank}\mathbf{A} = \dim \operatorname{col}\mathbf{A}.$$

行列 $\mathbf{B}$ のランクは 1 である。2 つの列は線形従属であり、どちらもゼロではないからである。もう少し豊かな例として、次の 4×5 行列を考えよう。

$$\mathbf{C} = \begin{bmatrix} 1 & 3 & 0 & -1 & 0 \\ -1 & 0 & 1 & 1 & -1 \\ 0 & 3 & 1 & 0 & -1 \\ 2 & 3 & -1 & -2 & 1 \end{bmatrix},$$

この行列のランクは 2 である。最初の 2 つの列は線形独立であるが、$\binom{5}{3} = 10$ 通りある 3 列の組はどれも線形従属になっている。

注目すべきことに、写像で生き残る次元の数と 0 に潰される次元の数を足し合わせると、常に入力空間の次元に一致する。

命題（ランク・退化次数の定理） $m \times n$ **行列 A に対して、次が成り立つ。**

$$\operatorname{rank}\mathbf{A} + \dim \ker \mathbf{A} = n. \tag{33.14}$$

証明基底 $\mathbf{w}_1, \dots, \mathbf{w}_k$ を $\ker \mathbf{A}$ から選び、それを拡張して基底 $\mathbf{w}_1, \dots, \mathbf{w}_k, \mathbf{u}_1, \dots, \mathbf{u}_{n-k}$（$\mathbb{R}^n$ の基底）を作る。（線形独立なベクトルの組が常に基底へと拡張できること、および部分空間のすべての基底が同じ個数のベクトルを持つことは、標準的な事実として証明なしに用いる。）像 $\mathbf{A}\mathbf{u}_1, \dots, \mathbf{A}\mathbf{u}_{n-k}$ は $\operatorname{col}\mathbf{A}$ の張る空間をなす。なぜなら、任意の入力 $\mathbf{x} \in \mathbb{R}^n$ は基底全体の線形結合で表され、各 $\mathbf{w}_j$ は $\mathbf{A}$ によって 0 に潰れるからである。さらに、これらの像は線形独立でもある。もし $\sum_i c_i \mathbf{A}\mathbf{u}_i = \mathbf{0}$ ならば、$\sum_i c_i \mathbf{u}_i$ は $\ker \mathbf{A}$ に属するため、$\mathbf{w}_1, \dots, \mathbf{w}_k$ の線形結合として書ける。基底の内部では、すべての $c_i = 0$ でない限りそのようなことは起こりえない。ゆえに、$\operatorname{col}\mathbf{A}$ の次元は $n - k$ である。$\blacksquare$

Figure 33.12 に、この証明で構成した基底を示す。n 本のベクトルのそれぞれは、$\mathbf{A}$ によって 0 になるか列空間に 1 次元分寄与するかのいずれかであり、両方に当てはまるベクトルは存在しない。

この次元の保存を具体例で確かめてみよう。$\mathbf{B}$ では $1 + 1 = 2$ となる。冒頭で挙げた Fashion-MNIST の 256×784 層では、ランクは出力座標の個数である 256 を超えられないため、零空間の次元は少なくとも $784 - 256 = 528$ でなければならない。$\mathbf{C}$ では、零空間の次元は $5 - 2 = 3$ となるため、線形独立な 3 本の零空間のベクトルがあれば $\ker \mathbf{C}$ の基底をなす。以下では、$(3, -1, 3, 0, 0)$、$(1, 0, 0, 1, 0)$、$(0, 0, 1, 0, 1)$ がいずれも $\mathbf{C}$ によって 0 に写ることを確かめ、これらを列として並べた行列 `Z` のランクを調べる。

```
B = np.array([[2.0, 1.0], [4.0, 2.0]])
C = np.array([[1, 3, 0, -1, 0], [-1, 0, 1, 1, -1],
              [0, 3, 1, 0, -1], [2, 3, -1, -2, 1]], dtype=float)
Z = np.array([[3, -1, 3, 0, 0], [1, 0, 0, 1, 0],
              [0, 0, 1, 0, 1]], dtype=float).T   # candidate null vectors
rank_C, rank_Z = np.linalg.matrix_rank(C), np.linalg.matrix_rank(Z)
np.linalg.matrix_rank(B), rank_C, np.abs(C @ Z).max(), rank_Z, rank_C + rank_Z
```

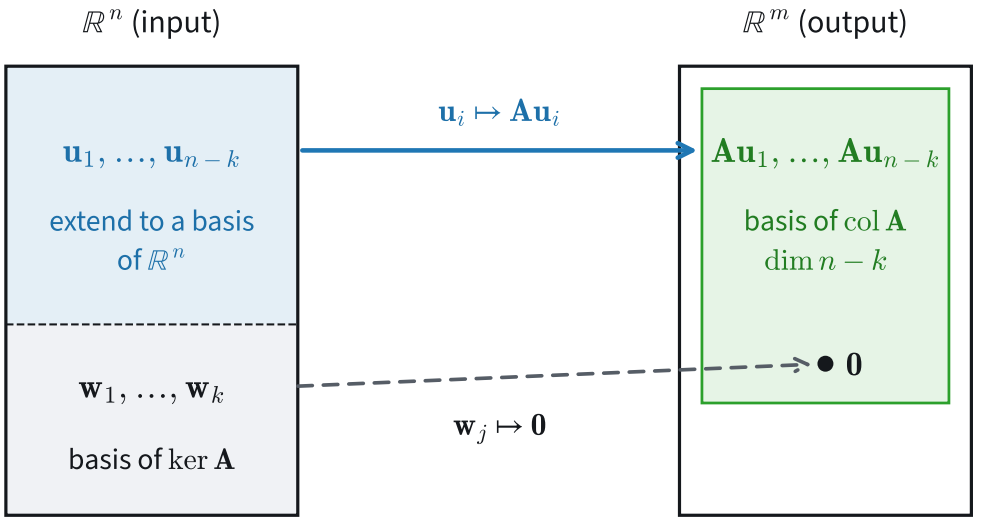


Figure 33.12: ランク・退化次数の定理の証明における基底。k 本のベクトル $\mathbf{w}_1, \dots, \mathbf{w}_k$ は $\ker \mathbf{A}$ を張る空間をなし、$\mathbf{0}$ に送られる。残る $n-k$ 本のベクトル $\mathbf{u}_1, \dots, \mathbf{u}_{n-k}$ はこれらを補って $\mathbb{R}^n$ の基底を完成させ、その像が $\operatorname{col} \mathbf{A}$ の基底をなす。

```
(np.int64(1), np.int64(2), np.float64(0.0), np.int64(3), np.int64(5))
```

$\mathbf{B}$ と $\mathbf{C}$ のランクはそれぞれ 1 と 2 になり、$|\mathbf{CZ}|$ の最大成分は 0、Z の 3 つの列のランクは 3 となって、$2 + 3 = 5$ が確認できる。

行列を列ではなく行ごとに読んだ場合はどうだろうか。線形独立な行の最大本数である**行ランク**（row rank）は、その行列が独立な超平面判定をいくつ実行するかを数えている。$\mathbf{C}$ では、第 3 行は第 1 行と第 2 行の和であり、第 4 行は第 1 行と第 2 行の差であるため、行ランクは 2 となって列ランクと一致する。この一致は常にあらゆる行列で成り立つ。

命題（行ランクと列ランクの一致） すべての行列 $\mathbf{A}$ に対して、線形独立な行の最大本数は、線形独立な列の最大本数に等しい。

証明 $r = \operatorname{rank} \mathbf{A}$ とおき、$\operatorname{col} \mathbf{A}$ の基底を $m \times r$ 行列 $\mathbf{G}$ の各列に並べる。$\mathbf{A}$ の各列は $\mathbf{G}$ の列の線形結合であるため、第 j 列（$\mathbf{A}$ の列）に対する係数を第 j 列（$r \times n$ 行列 $\mathbf{H}$ の列）に格納すれば、次の分解が得られる。

$$\mathbf{A} = \mathbf{G}\mathbf{H}. \tag{33.15}$$

今度は同じ積を行ごとに読もう。第 i 行（$\mathbf{A}$ の行）は $\sum_k g_{ik}\,(\text{row } k \text{ of } \mathbf{H})$ となるため、$\mathbf{A}$ のすべての行は r 本の $\mathbf{H}$ の行が張る空間に含まれる。したがって、$\mathbf{A}$ の行ランクは高々 r である。転置 $\mathbf{A}^\top$（その列は $\mathbf{A}$ の行である）に対して同じ議論を繰り返せば、列ランクも高々行ランクであることがわかり、両者は一致しなければならない。■

分解 $\mathbf{A} = \mathbf{G}\mathbf{H}$ は単なる証明の道具にとどまらない。ここからは、ランク r のあらゆる行列が幅 r のボトルネックに入力を通していることが見えてくる。すなわち、$\mathbf{H}$ が $\mathbf{x} \in \mathbb{R}^n$ を r 個の数値へと圧縮し、$\mathbf{G}$ がその r 個の数値を再び $\mathbb{R}^m$ へと展開する。r が m や n よりはるかに小さいとき、

G と **H** の格納に必要なパラメータ数はわずか $r(m+n)$ 個で済み、mn 個を保持する必要はない。これが 33.5.4.1 節で扱う低ランク更新の中心的な着想となる。

実際には、列の部分集合を片端から調べてランクを求めようとすれば指数関数的な時間がかかってしまう。手計算ではガウスの消去法を用いる（本章では扱わない）。一方、`np.linalg.matrix_rank` のような数値計算ライブラリは、微小な許容誤差を超える**特異値**（singular value）（33.5 節）がいくつあるかを数える。ランクは行列の成分の連続関数ではないため、許容誤差の導入は避けられない。ごくわずかな摂動でも、ランク落ちの状態からフルランクへと不連続に跳ね上がってしまうからである（33.2.3.6 節）。

33.2.2.3 逆行列

出力 **Ax** が与えられたとき、いつ入力 **x** を確実に復元できるだろうか。正方行列 **A** の**逆行列**（inverse）とは、行列 $\mathbf{A}^{-1}$ であって両側から **A** の作用を打ち消すものである。

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I},$$

これにより、$\mathbf{x} = \mathbf{A}^{-1}(\mathbf{A}\mathbf{x})$ がすべての入力 **x** に対して成り立つ。2×2 行列（成分 p, q, r, s が $ps - qr \neq 0$ を満たすもの）には、明示的な公式がある。

$$\mathbf{A}^{-1} = \frac{1}{ps-qr}\begin{bmatrix} s & -q \\ -r & p \end{bmatrix}. \tag{33.16}$$

実際に掛けてみればすぐに確かめられる。$\left(\begin{smallmatrix} s & -q \\ -r & p \end{smallmatrix}\right)$ と **A** をどちらの順序で掛けても $(ps-qr)\,\mathbf{I}$ になり、$ps-qr$ で割れば **I** が残る。条件 $ps-qr \neq 0$ を外すことはできない。もし $ps-qr = 0$ ならば、この積はゼロ行列になるため、仮に逆行列 $\mathbf{A}^{-1}$ が存在して両辺に掛けられたとすると $\left(\begin{smallmatrix} s & -q \\ -r & p \end{smallmatrix}\right) = \mathbf{0}$、ひいては $\mathbf{A} = \mathbf{0}$ となってしまうが、ゼロ行列が逆行列を持たないことは明らかである。量 $ps-qr$ には、33.2.3.1 節で符号付き面積として再び出会う。通しで用いる行列では $ps - qr = 3 + 2 = 5$ であるから、次のようになる。

$$\mathbf{A}^{-1} = \frac{1}{5}\begin{bmatrix} 3 & -2 \\ 1 & 1 \end{bmatrix}, \qquad \mathbf{A}^{-1}(0,-5) = \tfrac{1}{5}\,(10,-5) = (2,-1).$$

確かに、$\mathbf{A}^{-1}$ は 33.2.1.1 節の計算例を逆向きに戻している。

逆行列が存在するとき、それは常に一意である。さらに都合のよいことに、同じ行列の左逆行列と右逆行列は必ず一致する。実際、$\mathbf{L}\mathbf{A} = \mathbf{I}$ かつ $\mathbf{A}\mathbf{X} = \mathbf{I}$ ならば、結合法則により次が成り立つ。

$$\mathbf{L} = \mathbf{L}(\mathbf{A}\mathbf{X}) = (\mathbf{L}\mathbf{A})\mathbf{X} = \mathbf{X}.$$

一般には、いつ逆行列が存在するのだろうか。正方行列の場合、その答えには零空間とランク・退化次数の定理しか必要なく、本章の残りの部分は本節の他のどの結果よりもこの定理に依拠する。

命題（可逆行列） $n \times n$ **行列 A に対して、次の条件はすべて同値である。** (i) **A は可逆である。** (ii) **A の列は線形独立である。** (iii) $\ker \mathbf{A} = \{\mathbf{0}\}$ **である。** (iv) $\operatorname{rank} \mathbf{A} = n$ **である。**

証明 条件 (ii) と (iii) は 33.2.2.1 節の事実 2 により同値であり、(iii) と (iv) はランク・退化次数の定理により同値である。**A** が可逆（invertible）であり $\mathbf{A}\mathbf{x} = \mathbf{0}$ を満たすとき、左から $\mathbf{A}^{-1}$ を掛ければ $\mathbf{x} = \mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{0}$ となるため、(i) から (iii) が従う。

逆向きを示すには、実際に逆行列を構成しなければならないためもう少し手間がかかる。(iv) より、列空間は n 次元の $\mathbb{R}^n$ の部分空間であり、これは $\mathbb{R}^n$ 全体に一致しなければならない。もしその外にあるベクトルを $\mathbf{col}\,\mathbf{A}$ の基底に付け加えれば、$n+1$ 本の線形独立なベクトルが $\mathbb{R}^n$ 内に生じることになり矛盾するからである。ゆえに、$\mathbb{R}^n$ 内のあらゆる目標ベクトルに到達できる。また (iii) と事実 3 より、異なる 2 つの入力が同じ出力に写ることはない。したがって、第 j 列（行列 $\mathbf{X}$ の列）を $\mathbf{A}\mathbf{x} = \mathbf{e}_j$ の一意な解として定義できる。構成法から直ちに $\mathbf{A}\mathbf{X} = \mathbf{I}$ が成り立つ。$\mathbf{X}$ が左からも逆行列として働くことを確かめるため、任意の $\mathbf{x} \in \mathbb{R}^n$ をとろう。$\mathbf{A}(\mathbf{X}\mathbf{A}\mathbf{x}) = (\mathbf{A}\mathbf{X})\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}$ であり、異なる 2 つの入力は衝突しないから、$\mathbf{X}\mathbf{A}\mathbf{x} = \mathbf{x}$ でなければならない。ゆえに $\mathbf{X}\mathbf{A} = \mathbf{I}$ も成り立ち、$\mathbf{X} = \mathbf{A}^{-1}$ となる。■

これにより、33.2.1.3 節で証明を先送りにしていた主張が直ちに解決する。正方の直交行列 $\mathbf{Q}$（$\mathbf{Q}^{\top}\mathbf{Q} = \mathbf{I}$ を満たす）に対して、もし $\mathbf{Q}\mathbf{c} = \mathbf{0}$ ならば $\mathbf{c} = \mathbf{Q}^{\top}\mathbf{Q}\mathbf{c} = \mathbf{0}$ となるため、$\ker \mathbf{Q} = \{\mathbf{0}\}$ である。上の命題より $\mathbf{Q}$ は可逆であり、その左逆行列である転置 $\mathbf{Q}^{\top}$ は $\mathbf{Q}^{-1}$ に一致しなければならず、したがって $\mathbf{Q}\mathbf{Q}^{\top} = \mathbf{I}$ が成り立つ。

逆行列を持たない正方行列は**特異**（singular）であるといい、$\mathbf{B}$ がその典型例である（特異行列とも呼ぶ）。一方、長方形の行列はそもそも両側逆行列を決して持たない。$m \times n$ 行列のランクは高々 $\min(m, n)$ である。$m > n$ のとき、列空間の次元は高々 $n < m$ であるため写像は**全射**（onto）ではなく、$\mathbb{R}^m$ 内に決して到達できない目標が生じる。$n > m$ のとき、ランク・退化次数の定理により少なくとも次元 $n - m > 0$ の零空間が生じるため写像は**単射**（one-to-one）ではなく、異なる入力が衝突する。写像を逆向きにたどるには両方の性質が不可欠であり、上の命題の美しさは、正方行列においてはどちらか一方さえ成り立てばもう一方も保証される点にある。

33.2.2.4 連立一次方程式の実践的な解法

逆行列は代数計算や証明には欠かせない道具であるが、数値計算において逆行列を明示的に作ることはまずない。$\mathbf{A}\mathbf{x} = \mathbf{b}$ を解く必要があるときは、`np.linalg.solve(A, b)` を用い、`np.linalg.inv(A) @ b` は避けるべきである。

理由は大きく 2 つ、計算速度と計算精度にある。第 1 に、`solve` は $\mathbf{A}$ を三角行列の積へと 1 回だけ分解し（**LU 分解**（LU factorization）、コストは約 $n^3/3$ 回の積和演算）、高速に解ける 2 つの三角連立一次方程式を解く。これに対し、$\mathbf{A}^{-1}$ を明示的に求めると約 n^3 回の積和演算がかかり、$\mathbf{b}$ を掛ける前の段階ですでに 3 倍の計算量を要する。第 2 に、`solve` は**後退安定**（backward stable）である。すなわち、計算された解は、$\mathbf{A}$ と $\mathbf{b}$ の成分を浮動小数点の丸め誤差レベルだけわずかに摂動させた連立一次方程式の厳密解になっている。その結果、残差 $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ は計算機精度と $\|\mathbf{A}\|\,\|\mathbf{x}\|$（1.3.4 節のスペクトルノルムを使用）の積のオーダーに収まる。一方、`inv(A) @ b` の計算にはそのような保証がなく、残差は最大で**条件数**（condition number）$\kappa(\mathbf{A})$ 倍まで膨らみうる。条件数とは、$\mathbf{A}$ が特異行列にどれほど近いかを測る指標である（33.5.3.2 節）。

行列が特異性にどれほど近いかは問題そのものの性質であり、アルゴリズムの性質ではない。倍精度（10 進で約 16 桁）では、どちらの手法でも解 $\mathbf{x}$ の精度が最大で約 $\log_{10}\kappa(\mathbf{A})$ 桁失われうる。これは最悪ケースの上界であって個々の右辺に対する厳密な予測ではないため、悪名高い具体例がこの限界にどれほど迫るかを見てみよう。成分 $h_{ij} = 1/(i+j-1)$ を持つ**ヒルベルト行列**（Hilbert matrix）は、特異に極めて近い行列の古典的なテストケースである。$\mathbf{x}_{\text{true}} = \mathbf{1}$ とし、$\mathbf{b} = \mathbf{H}\mathbf{1}$ を 10×10 ヒルベルト行列 $\mathbf{H}$ に対して構成し、両方の方法で $\mathbf{x}$ の復元を試みる。

```
H = 1 / (np.arange(10)[:, None] + np.arange(10) + 1)   # Hilbert matrix
x_true = np.ones(10)
b = H @ x_true
for x in (np.linalg.solve(H, b), np.linalg.inv(H) @ b):
    print(f'residual {np.linalg.norm(H @ x - b):.1e}, '
```

```
            f'error {np.linalg.norm(x - x_true):.1e}')
np.linalg.cond(H)
```

```
residual 5.7e-16, error 2.7e-04
residual 4.8e-05, error 9.0e-03
```

```
np.float64(16024416987428.36)
```

条件数は $\kappa(\mathbf{H}) \approx 1.6 \cdot 10^{13}$ であり、最大で 13 桁が失われうることを警告している。予告どおり、`solve` は残差を計算機精度付近に保つのに対し、`inv` の残差は約 11 桁も大きくなり、因子 κ から予想される結果とよく一致する。ただし、残差が微小だからといって精度の高い解が保証されるわけではない点に注意しよう。どちらの方法でも 16 桁のうち 12 桁以上が失われて最悪ケースの上界に迫り、その中でも `inv` はさらに 1 桁以上悪い誤差を示す。

スパース行列は、決して逆行列を求めてはならない 3 つ目の決定的な理由をもたらす。非ゼロ成分が 500 万個しかない 100 万行 100 万列の行列はメモリに余裕で収まるが、その逆行列は通常完全に密になり、約 10^{12} 個もの成分を要してしまう。

33.2.3 行列式

33.2.3.1 平行四辺形の面積

行列は面積を何倍に拡大・縮小するだろうか。Figure 33.8 を振り返ってみよう。通しで用いる行列 $\mathbf{A}$ は、単位正方形（辺 $\mathbf{e}_1$ と $\mathbf{e}_2$、面積 1）を、辺 $\mathbf{a}_1 = (1, -1)$ と $\mathbf{a}_2 = (2, 3)$、頂点 $(0, 0)$、$(1, -1)$、$(3, 2)$、$(2, 3)$ を持つ平行四辺形へと送る。この平行四辺形は、バウンディングボックス $[0, 3] \times [-1, 3]$（面積 12）の内部にぴったり収まっている。ボックスの余った部分は、直角を挟む 2 辺がそれぞれ 1 と 1、2 と 3、1 と 1、2 と 3 である 4 つの直角三角形からなり、その面積の合計は $\frac{1}{2} + 3 + \frac{1}{2} + 3 = 7$ である。これを差し引くと面積 $12 - 7 = 5$ が残る（Figure 33.13 のパネル (a)）。すなわち、行列 $\mathbf{A}$ は単位正方形の面積を 5 倍にする。

三角形を切り出す方法は 1 つの図に対しては有効だが、頂点の配置が変わるたびに新しい図を描き直す必要がある。一方、「底辺と高さの積」を使えばあらゆる場合を一挙に扱える。平行四辺形の辺を $\mathbf{u} = (p, r)$ と $\mathbf{w} = (q, s)$、すなわち一般の 2×2 行列の 2 つの列としよう。底辺の長さは $\|\mathbf{u}\|$ である。$\mathbf{u}$ を反時計回りに 90° 回転させると $\mathbf{R}_{90}\mathbf{u} = (-r, p)$ になるため、次のベクトルを考える。

$$\mathbf{n} = \frac{(-r, p)}{\|\mathbf{u}\|}$$

これは底辺に垂直な単位ベクトルである。33.1 節で見たとおり、$\mathbf{w}$ の単位ベクトル $\mathbf{n}$ 方向の符号付き成分はドット積 $\mathbf{w} \cdot \mathbf{n}$ で与えられる。幾何学的には、これは底辺から測った平行四辺形の高さに符号を付けたものであり、$\mathbf{w}$ が $\mathbf{u}$ から見て反時計回りの側にあるときは正、時計回りの側にあるときは負になる。底辺に符号付き高さを掛けると、次の式が得られる。

$$\|\mathbf{u}\| \, (\mathbf{w} \cdot \mathbf{n}) = \|\mathbf{u}\| \, \frac{-qr + ps}{\|\mathbf{u}\|} = ps - qr.$$

都合のよいことに長さ $\|\mathbf{u}\|$ が約分されて消え、符号による場合分けを一切必要としない、行列の 4 成分に関する簡潔な多項式だけが残る。通しで用いる行列では $\|\mathbf{u}\| = \sqrt{2}$、$\mathbf{n} = (1, 1)/\sqrt{2}$、$\mathbf{w} \cdot \mathbf{n} = 5/\sqrt{2}$ となり、符号付き面積は 5 となる（パネル (b)）。これが正になるのは、$\mathbf{a}_2$ が $\mathbf{a}_1$ から見て反時計回りの側にあるからである。

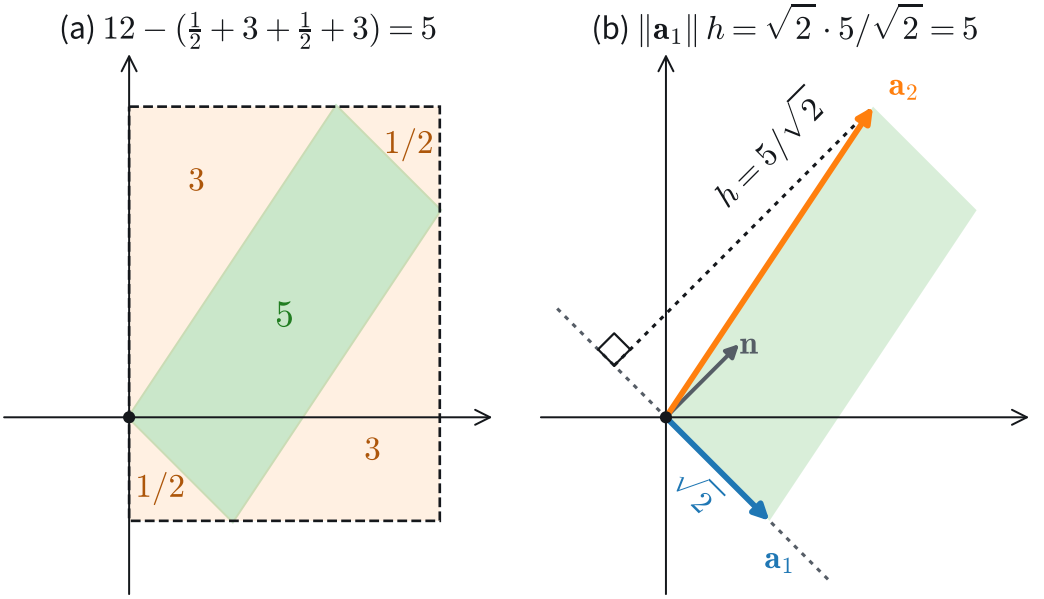


Figure 33.13: $\mathbf{A}$ による単位正方形の像の面積は 5 である。(a) バウンディングボックスによる計算。$[0,3] \times [-1,3]$ のボックス（面積 12）から、合計面積 7 の 4 つの直角三角形を引く。(b) 底辺と高さの積。底辺 $\mathbf{a}_1 = (1,-1)$ の長さは $\sqrt{2}$ であり、単位法線ベクトル $\mathbf{n} = (1,1)/\sqrt{2}$ はこれに垂直で、底辺を含む直線から測った $\mathbf{a}_2 = (2,3)$ の高さは $\mathbf{a}_2 \cdot \mathbf{n} = 5/\sqrt{2}$ である。

2×2 行列の**行列式**を、この単位正方形の像の符号付き面積として定義する。

$$\det \begin{bmatrix} p & q \\ r & s \end{bmatrix} = ps - qr. \tag{33.17}$$

その絶対値 $|\det \mathbf{A}|$ は写像が面積を何倍にするかを表し、その符号は $\mathbf{e}_1$ と $\mathbf{e}_2$ の像が反時計回りの向き（orientation）を保っているかどうかを記録する。$\mathbf{A}$ に対しては $1 \cdot 3 - 2 \cdot (-1) = 5$ となる。$\mathbf{A}$ の 2 つの列を入れ替えると $\left(\begin{smallmatrix} 2 & 1 \\ 3 & -1 \end{smallmatrix}\right)$ が得られ、辺を並べる順序が逆になっただけで張る空間としての平行四辺形はまったく同じだが、行列式の符号は反転して $-2 - 3 = -5$ になる。特異行列 $\mathbf{B}$ に対しては $2 \cdot 2 - 1 \cdot 4 = 0$ となり、単位正方形は $\mathbf{0}$ から $(3,6)$ までの面積 0 の線分へと平坦に潰れる。

Figure 33.14 はこれら 3 つの類型を対比している。正の行列式は向きを保ち、負の行列式は向きを反転させ、0 の行列式は正方形を潰す。

「底辺と高さの積」によって平面の場合は解決した。n 次元へ進むには、n 次元の箱の符号付き体積が必要になる。

33.2.3.2 一般の行列式

n 本の列が $\mathbb{R}^n$ 内で張る箱の符号付き体積は、どのように定義すればよいだろうか。図形を切り分けて高次元の体積を計算しようとする代わりに、ここではより見通しのよい道をとる。符号付き体積が満たすべき 3 つの幾何学的性質を書き下し、それらの性質から公式を一意に決定させるのである。列 $\mathbf{a}_1, \dots, \mathbf{a}_n$（$n \times n$ 行列の各列）の関数として見たとき、行列式は次の性質を持つ。

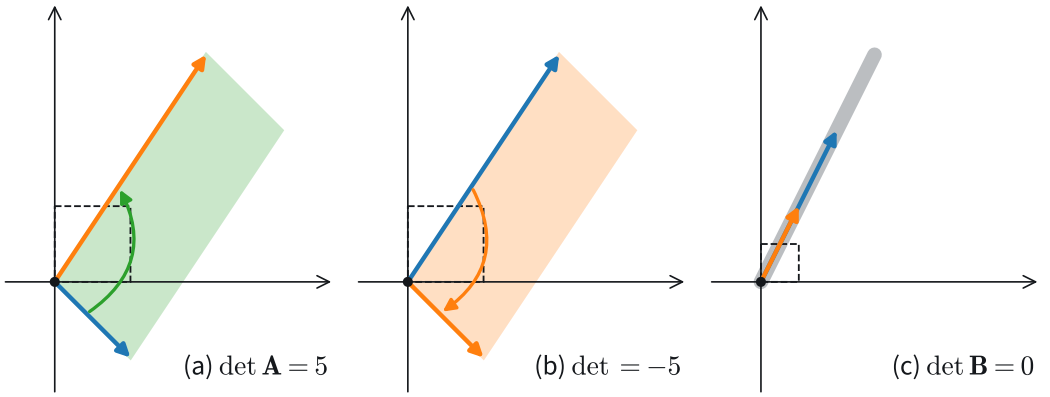


Figure 33.14: 3 つの 2×2 の例における符号付き面積としての行列式。単位正方形（破線）は各列が張る領域へと写る。(a) $\mathbf{A}$（行列式 5）。列は反時計回りの順序（曲線矢印）を保つ。(b) 列を入れ替えた $\mathbf{A}$（行列式 -5）。平行四辺形は同じだが向きが反転している。(c) $\mathbf{B}$（行列式 0）。正方形は直線 $y = 2x$ 上の線分へと潰れる。パネル (c) は (a) と (b) の半分の縮尺で描かれている。

1. **多重線形**（multilinear）：他の列を固定したとき、各列について個別に線形である。（箱の 1 つの辺をスカラー倍すると体積も同じ倍率で変化し、1 つの辺を和として表すと体積もそれに応じて分割される。）
2. **交代的**（alternating）：2 つの列が一致するときは常に 0 に等しい。（2 つの辺が重なると、箱は体積 0 の低次元の板へと潰れる。）
3. **正規化**（normalized）：$\det \mathbf{I} = 1$ である。（単位超立方体の体積は 1 である。）

2×2 の公式 $ps - qr$ がこれら 3 つをすべて満たすことを確かめよう。(p, r) と (q, s) のそれぞれについて線形であり、$(p, r) = (q, s)$ のときは常に 0 になり、さらに値 1 を $\mathbf{I}$ に対してとる。これら 3 つの公理だけから、2 つの強力な帰結が直ちに導かれる。列を入れ替えたときの規則と、3 つの公理をすべて満たす関数が高々 1 つしか存在しないことを示す明示公式である。

第 1 に、**任意の 2 つの列を入れ替えると行列式の符号が反転する**。理由を見るため、2 つの列の位置に $\mathbf{u} + \mathbf{w}$ を置いて多重線形性により展開しよう。

$$\begin{aligned} 0 &= \det(\dots, \mathbf{u} + \mathbf{w}, \dots, \mathbf{u} + \mathbf{w}, \dots) \\ &= \det(\dots, \mathbf{u}, \dots, \mathbf{u}, \dots) + \det(\dots, \mathbf{u}, \dots, \mathbf{w}, \dots) \\ &\quad + \det(\dots, \mathbf{w}, \dots, \mathbf{u}, \dots) + \det(\dots, \mathbf{w}, \dots, \mathbf{w}, \dots). \end{aligned}$$

左辺、第 1 項、および最後の項はいずれも一致する 2 つの列を持つため、交代性により 0 になる。残るのは次の式である。

$$\det(\dots, \mathbf{u}, \dots, \mathbf{w}, \dots) = -\det(\dots, \mathbf{w}, \dots, \mathbf{u}, \dots).$$

箱の 2 つの辺を入れ替えると、Figure 33.14 のパネル (b) で見たのと同様に、箱の向きが反転する。

第 2 に、これら 3 つの性質から公式が完全に定まる。各列を標準基底ベクトルで $\mathbf{a}_j = \sum_i a_{ij}\, \mathbf{e}_i$ と展開し、1 列ずつ多重線形性を適用しよう。すると $\det \mathbf{A}$ は、$a_{i_1 1} a_{i_2 2} \cdots a_{i_n n}\, \det(\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_n})$ という項の大きな和になり、行添字 i_j を各列 j で選ぶすべての組み合わせごとに 1 項ずつ現れ

る。選ばれた行添字のうち 2 つでも等しければ、交代性によりその項は 0 になる。n 個の行添字がすべて異なるとき、並び $\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_n}$ は $\mathbf{I}$ の列を並べ替えたものにすぎないため、列の入れ替えを繰り返すことで $\det(\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_n})$ は ± 1 に帰着する。

2×2 の場合、列 $\mathbf{a}_1 = p\,\mathbf{e}_1 + r\,\mathbf{e}_2$ と $\mathbf{a}_2 = q\,\mathbf{e}_1 + s\,\mathbf{e}_2$ に対して展開には 4 つの項が現れる。同じ基底ベクトルが重複する 2 項は消え、次の式が残る。

$$\det(\mathbf{a}_1, \mathbf{a}_2) = ps\,\det(\mathbf{e}_1, \mathbf{e}_2) + rq\,\det(\mathbf{e}_2, \mathbf{e}_1) = ps - qr,$$

こうして手計算で導いた符号付き面積が再現される。一般に、$\det \mathbf{A}$ はすべての行とすべての列から成分を 1 つずつ選ぶあらゆる方法にわたる符号付き和になる。$\sigma(j)$ で第 j 列において選ばれた行を表せば、次の**置換公式**（permutation formula）が得られる。

$$\det \mathbf{A} = \sum_{\sigma} \operatorname{sgn}(\sigma)\, a_{\sigma(1)1}\, a_{\sigma(2)2} \cdots a_{\sigma(n)n}, \tag{33.18}$$

ここで、σ は $n!$ 通りの $\{1, \dots, n\}$ のすべての置換（permutation）を動き、$\operatorname{sgn}(\sigma) = \det(\mathbf{e}_{\sigma(1)}, \dots, \mathbf{e}_{\sigma(n)})$ は $+1$（σ が偶数回の入れ替えを要するとき）または -1（奇数回の入れ替えを要するとき）をとる。3 つの公理を満たすいかなる関数もこの公式に一致しなければならないため、行列式は高々 1 つしか存在しえない。

ここでは存在の完全な証明には立ち入らない。微妙な点は $\operatorname{sgn}(\sigma)$ が一意に定まること（いかなる置換も偶数回の入れ替えと奇数回の入れ替えの両方では表せないこと）であるが、これは標準的な事実として受け入れる (Horn and Johnson 2012)。この偶奇性の事実さえ認めれば、上の公式は 3 つの公理をすべて満たす。すなわち、各項は各列から成分を 1 つずつ含むため多重線形性が成り立ち、$\mathbf{I}$ に対しては恒等置換のみが生き残るため正規化が成り立ち、交代性も同じ偶奇性の議論から従う（証明は省略する）。

正規化の条件が導出の一番最後にしか使われなかったことに注目しよう。この着眼点から、次の項で用いるより鋭い道具が得られる。もし f が列に関する**任意の**多重線形かつ交代的な関数であれば、同じ列の入れ替え規則と展開がそのまま適用でき、すべての因子 $f(\mathbf{e}_{\sigma(1)}, \dots, \mathbf{e}_{\sigma(n)})$ が $\operatorname{sgn}(\sigma)\, f(\mathbf{I})$ へと変わる。その結果、次の関係が成り立つ。

$$f(\mathbf{A}) = f(\mathbf{I})\,\det \mathbf{A}.$$

要するに、列に関するすべての多重線形かつ交代的な関数は、行列式のスカラー倍にすぎない。

小さな行列の手計算では、Equation 33.18 の各項を第 1 列で選ばれた行ごとにまとめることで**余因子展開**（cofactor expansion）が得られる。すなわち、各成分 a_{i1} に、$(n-1) \times (n-1)$ 小行列（第 i 行と第 1 列を除いて作られる行列）の行列式と交代符号 $(-1)^{i+1}$ を掛け合わせる。次の 3×3 行列に対して、

$$\mathbf{M} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{bmatrix},$$

第 1 列に沿って展開すると次のようになる。

$$\begin{aligned}\det \mathbf{M} &= 2 \det \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} - 1 \det \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} + 0 \det \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \\ &= 2 \cdot 5 - 1 \cdot 2 + 0 = 8.\end{aligned} \tag{33.19}$$

したがって、$\mathbf{M}$ は $\mathbb{R}^3$ 内の単位立方体を体積 8 の箱へと写す。

三角行列では、行列式は劇的に簡単になり、単に対角成分の積になる。行列式を計算するあらゆる規則の中で、真っ先に覚えておくべきなのはこの規則である。$\mathbf{U}$ を上三角行列とし、$u_{ij} = 0$ が $i > j$ のとき常に成り立つとしよう。置換公式において非ゼロの項を得るには、すべての列から成分を 1 つずつ、すべて異なる行から選ばなければならない。第 1 列で非ゼロになりうる成分は第 1 行にしかない。第 2 列では第 1 行がすでに使われており、第 $3, \dots, n$ 行は 0 であるため、必然的に第 2 行を選ぶことになる。同様に列を順に進めていくと、対角成分を選ぶ恒等置換だけが生き残り、入れ替えの回数は 0 回であるため、次が成り立つ。

$$\det \mathbf{U} = u_{11} u_{22} \cdots u_{nn}.$$

第 n 列から逆向きに同じ議論をたどれば、下三角行列の場合も示せる。さらに嬉しいことに、この規則はブロック行列へも拡張できる。**ブロック上三角**（block upper triangular）行列 $\left(\begin{smallmatrix} \mathbf{D}_1 & * \\ \mathbf{0} & \mathbf{D}_2 \end{smallmatrix}\right)$ を考えよう。ただし、正方の対角ブロック $\mathbf{D}_1$ と $\mathbf{D}_2$ 自体は三角行列でなくてもよい。非ゼロとなる置換の項は、$\mathbf{D}_1$ の列における成分を $\mathbf{D}_1$ の行から選ばなければならない。その下にある成分はすべて 0 だからである。これにより $\mathbf{D}_1$ の行がすべて使い果たされるため、残りの列は $\mathbf{D}_2$ の行から成分を選ぶしかなく、非対角ブロック $*$ は一切寄与しない。生き残るすべての置換は 2 つのブロック上の独立な置換の組へと分解され、符号も積になるため、和はきれいに因数分解される。

$$\det \begin{bmatrix} \mathbf{D}_1 & * \\ \mathbf{0} & \mathbf{D}_2 \end{bmatrix} = \det \mathbf{D}_1 \, \det \mathbf{D}_2.$$

Figure 33.15 に、それぞれの場合で生き残る成分を示す。

これら 2 つの規則は手計算だけでなく、現代の機械学習においても不可欠である。正規化フロー（normalizing flow）は単純な確率変数を可逆なニューラルネットワークで変換するモデルであり、各ステップでネットワークのヤコビ行列の対数行列式（log-determinant）を計算しなければならない。変数変換によって確率密度が局所的な体積拡大率で割られるからである（36.2.3.1 節）。そのためフローは、各層が三角またはブロック三角のヤコビ行列を持つように設計される。すると対数行列式は n 個のスカラーの対数の和に帰着し、一般の行列で必要となる約 $n^3/3$ 回の積和演算を回避できる。

もう 1 つの有用な対称性として、転置しても行列式は変わらない（$\det \mathbf{A}^\top = \det \mathbf{A}$）。$\mathbf{A}^\top$ に対する置換公式では、各項は $\operatorname{sgn}(\sigma) \prod_j a_{j\sigma(j)}$ である。ここで $j = \sigma^{-1}(i)$ とおくと積は $\prod_i a_{\sigma^{-1}(i)\, i}$ と書き直せ、これは $\mathbf{A}$ において逆置換 σ^{-1} に対応する項に一致する。一連の入れ替えを元に戻すにはまったく同じ回数の入れ替えを要するため、$\operatorname{sgn}(\sigma^{-1}) = \operatorname{sgn}(\sigma)$ が成り立つ。したがって 2 つの和はまったく同じ項を含み、行列式の列に関するあらゆる命題が行に対してもそのまま成り立つことがわかる。

$n!$ 個もの項を持つ置換公式はあくまで理論上の道具であり、ごく小さな行列を除いて数値計算に使われることは決してない。数値計算ライブラリは（`solve` と同様に）行列を三角行列の因子へと分解し、行の入れ替えに伴う符号を調整したうえで対角成分を掛け合わせることで、約 $n^3/3$ 回の積和演算というコストで行列式を計算する。

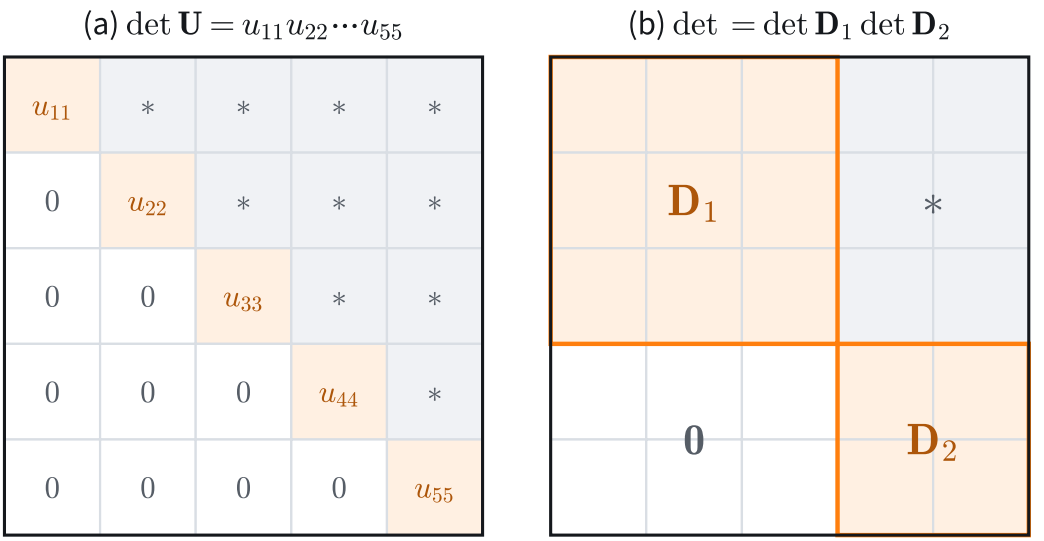


Figure 33.15: 行列式に寄与する成分。(a) 上三角行列では、対角線より下の 0 によって対角成分を選ぶ置換以外がすべて除外されるため、行列式は対角成分の積になる。(b) ブロック上三角行列では、ゼロブロックによって生き残るすべての置換が対角ブロック $\mathbf{D}_1$ と $\mathbf{D}_2$ に閉じ込められる。$*$ と記された成分は一切寄与しない。

33.2.3.3 乗法性

B が面積を $\det \mathbf{B}$ 倍にし、**A** が面積を $\det \mathbf{A}$ 倍にするならば、**B** に続けて **A** を適用すると面積は 2 つの倍率の積倍になるはずである。2 つの例で確かめてみよう。通しで用いる行列では $\mathbf{A}^2 = \left(\begin{smallmatrix} -1 & 8 \\ -4 & 7 \end{smallmatrix}\right)$ であり、$\det \mathbf{A}^2 = -7 + 32 = 25 = 5^2$ となる。また、33.2.1.2 節の合成写像では $\det(\mathbf{R}_{90}\mathbf{T}) = 0 \cdot 1 - (-1) \cdot 1 = 1$ となり、$\det \mathbf{R}_{90} \cdot \det \mathbf{T} = 1 \cdot 1$ と一致する。

なぜこれが一般に成り立つかは、簡単なタイル貼りの図から納得できる。任意の平面領域を微小な正方形の細かいグリッドで覆ってみよう。線形変換は空間全体に一様に作用するため、どの正方形も元の面積の $|\det \mathbf{A}|$ 倍の面積を持つ平行四辺形へと運ばれる。その結果、線形変換は**あらゆる**図形の面積を $|\det \mathbf{A}|$ 倍にする。**B** を面積 V の領域に適用すると面積は $|\det \mathbf{B}|\,V$ になり、続けて **A** を適用すると $|\det \mathbf{A}|\,|\det \mathbf{B}|\,V$ にスケーリングされる。合成 **AB** 自体も 1 つの線形変換であるから、その拡大率はこの積に一致しなければならない。33.2.3.2 節の特徴づけを用いれば、この直感を符号まで含めた 3 行の厳密な証明へと仕立てられる。

命題（行列式の乗法性） 同じサイズの正方行列 A と B に対して、次が成り立つ。

$$\det(\mathbf{AB}) = \det(\mathbf{A})\,\det(\mathbf{B}). \tag{33.20}$$

証明 **A** を固定し、$D(\mathbf{B}) = \det(\mathbf{AB})$ を行列 **B** の列の関数とみなす。33.2.1.2 節より、第 j 列（**AB** の列）は $\mathbf{Ab}_j$ である。**A** は線形であり det は多重線形であるから、D は **B** の列について多重線形である。また、**B** の 2 つの列が一致すれば **AB** の対応する列も一致するため、$D(\mathbf{B}) = 0$ となって D は交代的である。列に関するすべての多重線形かつ交代的な関数は行列式のスカラー倍であるから（33.2.3.2 節）、次が得られる。

$$\det(\mathbf{AB}) = D(\mathbf{B}) = D(\mathbf{I})\,\det(\mathbf{B}) = \det(\mathbf{A})\,\det(\mathbf{B}). \qquad \blacksquare$$

Equation 33.20 から導かれる次の 4 つの帰結は、本章の残りの部分で繰り返し現れる。

1. **行列が可換でなくとも行列式は可換である。** 行列式は通常のスカラーであるため、$\det(\mathbf{AB}) = \det\mathbf{A}\,\det\mathbf{B} = \det\mathbf{B}\,\det\mathbf{A} = \det(\mathbf{BA})$ が成り立つ。回転とせん断の例では $\mathbf{R}_{90}\mathbf{T} \neq \mathbf{T}\mathbf{R}_{90}$ であるが、どちらの積も行列式は 1 である。

2. **相似不変性**（similarity invariance）。任意の可逆行列 $\mathbf{V}$ に対し、帰結 1 を因子 $\mathbf{VA}$ と $\mathbf{V}^{-1}$ に適用すると、$\det(\mathbf{VAV}^{-1}) = \det(\mathbf{V}^{-1}\mathbf{VA}) = \det\mathbf{A}$ が得られる。行列 $\mathbf{VAV}^{-1}$ は、写像 $\mathbf{A}$ を $\mathbf{V}^{-1}$ の各列を座標軸とする座標系で表したものにすぎず、座標系を取り替えても体積の拡大率は変わらない。

3. **逆行列。** $\mathbf{A}$ が可逆ならば、$\mathbf{B} = \mathbf{A}^{-1}$ とおくことで $\det\mathbf{A}\,\det\mathbf{A}^{-1} = \det\mathbf{I} = 1$ が得られる。したがって、すべての可逆行列は $\det\mathbf{A} \neq 0$ かつ $\det\mathbf{A}^{-1} = 1/\det\mathbf{A}$ を満たす（通しで用いる行列では $1/5 = 0.2$）。

4. **直交行列。** 乗法性と $\det\mathbf{Q}^{\top} = \det\mathbf{Q}$ を $\mathbf{Q}^{\top}\mathbf{Q} = \mathbf{I}$ に組み合わせると次が得られる。

$$\det(\mathbf{Q})^2 = \det(\mathbf{Q}^{\top})\det(\mathbf{Q}) = \det(\mathbf{Q}^{\top}\mathbf{Q}) = \det(\mathbf{I}) = 1,$$

   したがって、$\det\mathbf{Q} = \pm 1$ となる。剛体運動は決して体積を変えず、平面においてはその符号によって向きを保つ回転（$\det = +1$）と向きを反転させる鏡映（$\det = -1$）が区別される（33.2.1.3 節）。

33.2.3.4　可逆性と行列式

行列式は、33.2.2.3 節の可逆性定理に 5 つ目の同値条件を加えるものであり、その中でも唯一、単一の数値で表される条件である。2 次元ではこの同値性は直ちにわかる。符号付き面積 $ps - qr$ が 0 になるのは、2 つの列が原点を通る同一の直線上にある（または一方が $\mathbf{0}$ である）とき、すなわち列が線形従属であるとき、かつそのときに限る。同じことがすべての次元で成り立つ。

命題（可逆性と行列式） $n \times n$ **行列 $\mathbf{A}$ が可逆であることは、$\det\mathbf{A} \neq 0$ であることと同値である。言い換えれば、$\det\mathbf{A} = 0$ であることは、$\mathbf{A}$ の列が線形従属であることと同値である。**

証明 $\mathbf{A}$ が可逆ならば、乗法性の帰結 3 より $\det\mathbf{A}\,\det\mathbf{A}^{-1} = 1$ となるため、$\det\mathbf{A} \neq 0$ である。逆に、$\mathbf{A}$ が可逆でないと仮定しよう。33.2.2.3 節の命題よりその列は線形従属であるため、ある列が他の列の線形結合として $\mathbf{a}_j = \sum_{i \neq j} c_i \mathbf{a}_i$ と書ける。第 j 列を多重線形性で展開すると $\det\mathbf{A} = \sum_{i \neq j} c_i \det(\dots, \mathbf{a}_i, \dots)$ となり、各項で $\mathbf{a}_i$ が第 i 列と第 j 列の両方の位置に現れる。したがってすべての項が一致する 2 つの列を持ち、交代性によって 0 になる。$\blacksquare$

通しで用いる行列では $\det\mathbf{A} = 5 \neq 0$ によって可逆性が保証され、$\mathbf{B}$ では $\det\mathbf{B} = 0$ によって特異性が確認される。ただし、この定理が約束**しない**事柄にも留意しておこう。非ゼロの行列式が保証するのは $\mathbf{A}^{-1}$ が数学的に存在することであり、それを数値的に安全に計算できることではない（33.2.3.6 節）。

33.2.3.5　ブロック行列と低ランク更新

行列の一部分（1 つのブロックや、小さな変更が加わる前の行列）の逆行列がすでにわかっている場面は少なくない。これを再利用できないだろうか。答えは、スカラーではなくブロックに対して消去を 1 ステップ施すことで得られる。正方行列 $\mathbf{M}$ を 4 つのブロック $\mathbf{M}_{11}$、$\mathbf{M}_{12}$、$\mathbf{M}_{21}$、$\mathbf{M}_{22}$ に分割し、$\mathbf{M}_{11}$ は正方な可逆行列とする。第 1 ブロック行に左から $\mathbf{M}_{21}\mathbf{M}_{11}^{-1}$ を掛けて第 2 ブロック行から引けば左下のブロックが消去でき、列に対して同じ操作を行えば右上のブロックも消去できる。これを行列の積として書くと次のようになる。

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{M}_{21}\mathbf{M}_{11}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{M}_{11}^{-1}\mathbf{M}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \qquad \mathbf{S} = \mathbf{M}_{22} - \mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{12}, \tag{33.21}$$

これは 3 つの因子を実際に掛け合わせれば確かめられる。行列 $\mathbf{S}$ は $\mathbf{M}_{11}$ に関する**シューア補行列**（Schur complement）であり、最初のブロックを消去した後に $\mathbf{M}_{22}$ に残る部分にほかならない。通しで用いる行列を 1×1 ブロックに分割した場合、$\mathbf{S} = 3 - (-1) \cdot 1^{-1} \cdot 2 = 5$ となる。

外側の 2 つの因子はいずれも対角成分が 1 の三角行列であるため、それぞれの行列式は 1 である。乗法性とブロック三角行列の規則を適用すると $\det \mathbf{M} = \det \mathbf{M}_{11} \, \det \mathbf{S}$（上の例では $1 \cdot 5$）が得られ、$\mathbf{M}$ が可逆であることは $\mathbf{S}$ が可逆であることと同値になる。外側の因子の逆行列は、非対角ブロックの符号を反転させるだけで得られる。したがって、3 つの因子の逆行列を逆順に掛けると次の式が得られる。

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{M}_{11}^{-1}\mathbf{M}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{M}_{21}\mathbf{M}_{11}^{-1} & \mathbf{I} \end{bmatrix}. \tag{33.22}$$

$\mathbf{M}^{-1}$ の右下のブロックは単に $\mathbf{S}^{-1}$ であり、左上のブロックは $\mathbf{M}_{11}^{-1} + \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{S}^{-1}\mathbf{M}_{21}\mathbf{M}_{11}^{-1}$ となる。しかし、左上隅から消去を始めなければならない理由はどこにもない。先に $\mathbf{M}_{22}$ を消去すればブロックの役割が入れ替わり、$\mathbf{M}^{-1}$ の左上のブロックはもう一方のシューア補行列 $\mathbf{M}_{11} - \mathbf{M}_{12}\mathbf{M}_{22}^{-1}\mathbf{M}_{21}$ の逆行列として現れる。この行列は、36.3.3.3 節でガウス分布の条件付き共分散として再び登場する。

左上のブロックに対するこれら 2 つの表現は一致するはずであり、この事実が驚くほど役に立つ。いま $\mathbf{A}^{-1}$ が $n \times n$ 行列 $\mathbf{A}$ に対して既知であり、そこへ $\mathbf{A}$ にランク k の行列（$\mathbf{U}\mathbf{W}^{\top}$、ただし $\mathbf{U}, \mathbf{W} \in \mathbb{R}^{n \times k}$ と書く）だけの変更が加わったとしよう。ブロックを $\mathbf{M}_{11} = \mathbf{A}$、$\mathbf{M}_{12} = -\mathbf{U}$、$\mathbf{M}_{21} = \mathbf{W}^{\top}$、$\mathbf{M}_{22} = \mathbf{I}_k$ と選ぶと、一方のシューア補行列は $\mathbf{A} + \mathbf{U}\mathbf{W}^{\top}$、もう一方は $k \times k$ 行列 $\mathbf{I}_k + \mathbf{W}^{\top}\mathbf{A}^{-1}\mathbf{U}$ となる。左上のブロックの 2 つの表現を等置すれば、次の**ウッドベリーの恒等式**（Woodbury identity）が得られる。

$$(\mathbf{A} + \mathbf{U}\mathbf{W}^{\top})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\left(\mathbf{I}_k + \mathbf{W}^{\top}\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{W}^{\top}\mathbf{A}^{-1}. \tag{33.23}$$

行列に対するランク k の変更は、その逆行列にも高々ランク k の変更しか引き起こさず、その変化分を求めるのに必要なのは $k \times k$ の逆行列だけである。$\mathbf{A}^{-1}$ が手元にあれば、更新コストは $n^2 k$ のオーダーの積和演算で済み、逆行列を最初から求め直す n^3 はかからない。さらに $\mathbf{A}$ が対角行列であれば、$\mathbf{A} + \mathbf{U}\mathbf{W}^{\top}$ を係数とする方程式を解くコストはわずか nk^2 のオーダーで済む。また 2 通りの消去は $\det \mathbf{M}$ に対しても 2 つの表現を与えるため、$\det(\mathbf{A} + \mathbf{U}\mathbf{W}^{\top}) = \det \mathbf{A} \, \det(\mathbf{I}_k + \mathbf{W}^{\top}\mathbf{A}^{-1}\mathbf{U})$ が従う。$k = 1$ のときは内側の逆行列が単なる割り算になり、この恒等式は**シャーマン・モリソンの公式**（Sherman–Morrison formula）として知られる。

$$(\mathbf{A} + \mathbf{u}\mathbf{w}^{\top})^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\,\mathbf{w}^{\top}\mathbf{A}^{-1}}{1 + \mathbf{w}^{\top}\mathbf{A}^{-1}\mathbf{u}}. \tag{33.24}$$

最小二乗法による当てはめでデータ例を 1 つずつ追加・削除できるのは、この公式のおかげである。各データ例は $\mathbf{X}^{\top}\mathbf{X}$ をランク 1 の項だけ変化させるからである。演習問題 6 では、この行列式と逆行列の公式の両方を正規化フローに応用する。

33.2.3.6 行列式の限界

$\det \mathbf{A} \neq 0$ であるとわかれば行列が可逆であることは言えるが、その行列が空間を潰す状態にどれほど**近い**か、あるいはどの方向に沿って潰れかけているかについては何も語らない。両極端の 2 つの鮮烈な例がこの事実を浮き彫りにする。第 1 に、スカラー倍した単位行列 $0.1\,\mathbf{I}_{400}$ は逆行列 $10\,\mathbf{I}_{400}$ を持つこれ以上ないほど性質のよい行列であるが、その行列式 10^{-400} はあまりに小さく、倍精度浮動小数点数で表せる最小の正の数を下回ってアンダーフローしてしまう。第 2 に、極端なせん断 $\left(\begin{smallmatrix} 1 & 10^{8} \\ 0 & 1 \end{smallmatrix}\right)$ は行列式が 1 であるにもかかわらず、左下の成分をわずか 10^{-8} 変えるだけで特異行列になってしまう（5）。

以下では、（1 つ目の例には符号と絶対値の対数を安全に求める `slogdet` を用いて）両方の行列式を、せん断の特異値とともに計算する。セルを実行する前に、**変更を加えていない**元のせん断に対して NumPy がどのランクを返すか予想してみてほしい。

```
sign, logdet = np.linalg.slogdet(0.1 * np.eye(400))
print(np.linalg.det(0.1 * np.eye(400)), sign, logdet)
shear = np.array([[1.0, 1e8], [0.0, 1.0]])
print(np.linalg.det(shear), np.linalg.matrix_rank(shear),
      np.linalg.svd(shear, compute_uv=False))
shear[1, 0] = 1e-8
print(np.linalg.det(shear), np.linalg.matrix_rank(shear))
```

```
0.0 1.0 -921.0340371976151
1.0 1 [1.e+08 1.e-08]
0.0 1
```

予想どおり、`det` を $0.1\,\mathbf{I}_{400}$ に適用すると `0.0` へとアンダーフローするが、`slogdet` は $400 \ln 0.1 \approx -921.03$ を難なく計算する。変更前のせん断の行列式は 1 であり、摂動を加えたせん断の行列式は丸め誤差の範囲で 0 になるが、`matrix_rank` は**どちらに対しても**ランク 1 を返す。厳密な算術では変更前のせん断のランクは 2 であるが、その 2 つの特異値は約 10^{8} と 10^{-8} である。NumPy は、最大特異値に行列サイズと計算機精度を掛けた値（ここでは $10^{8} \cdot 2 \cdot 2.2 \cdot 10^{-16} \approx 4 \cdot 10^{-8}$）を下回る特異値をすべて 0 とみなす。せん断の成分自体は厳密に格納されていても、大きさ 10^{8} の数を含む浮動小数点演算では 10^{-8} 前後の丸め誤差が生じるため、数値計算上このせん断は特異行列と区別できないのである。

行列が異なる方向に沿って空間をどのように引き伸ばし押し縮めるかを、たった 1 つの数値で捉えることは到底できない。それらの方向を見るには、固有値と特異値が必要になる。通しで用いる行列 $\mathbf{A}$ では、固有値は $2 \pm i$ であってその積は $5 = \det \mathbf{A}$ になり（33.3 節）、特異値は約 3.618 と 1.382 であってその積は $5 = \lvert \det \mathbf{A} \rvert$ になる（33.5 節）。実はすべての正方行列において、行列式は（複素共役ペアを含め重複度を込めて数えた）固有値の積に等しく、$\lvert \det \mathbf{A} \rvert$ は特異値の積に等しい。行列が空間を潰す状態にどれほど近いかを測るのは行列式ではなく最大特異値と最小特異値の**比**であり、こうした方向ごとの倍率を求めることが次節以降の課題となる。

33.2.4 まとめ

行列とは、標準座標軸の行き先を記録した各列によって完全に定まる線形変換であり、行列積とは右から左へと読む線形変換の合成にほかならない。アインシュタイン記法と `einsum` は、ドット積、行列積、トレース、そして Transformer のアテンションのような多軸テンソルの縮約を、単一の添字規則のもとで統一する。

すべての $m \times n$ 行列は、到達可能な出力からなる列空間と 0 に送られる入力からなる零空間を定め、両者はランク・退化次数の定理 $\operatorname{rank} \mathbf{A} + \dim \ker \mathbf{A} = n$ によって結ばれている。したがって、256×784 層は少なくとも 528 個の入力次元を必然的に消去する。正方行列においては、

列が線形独立であること、零空間が自明であること、フルランクであること、行列式が非ゼロであること、そして逆行列を持つことはすべて同値である。行列式は写像が体積を拡大・縮小する符号付き倍率を測る量であり、合成のもとで乗法的となり、任意の三角行列に対しては対角成分の積に等しい。ブロック消去を 1 ステップ施すとシューア補行列が現れ、そこからブロック行列の行列式と逆行列が、さらにウッドベリーの恒等式を通じて低ランク変更後の逆行列が導かれる。とはいえ、非ゼロの行列式が保証するのは逆行列の存在だけであり、行列が特異性にどれほど近いかまでは教えてくれない。この問いは固有値と特異値へと直結する。

33.2.5 演習問題

1. **角度を保つ写像**。すべての埋め込みベクトルを同じ直交行列で回転させても、すべてのコサイン類似度は変わらない。本問では、この性質を持つすべての線形変換を求める。
   1. 正方行列 $\mathbf{A}$ の列に関する条件として、$\mathbf{A}\mathbf{e}_1$ と $\mathbf{A}\mathbf{e}_2$ が直交するための条件を求めよ。また、$\mathbf{A} = \mathrm{diag}(1, 10)$ はこの条件を満たすが、$(1,1)$ と $(1,-1)$ のなす角を変えてしまうことを示せ。
   2. 可逆な $\mathbf{A}$ が非ゼロベクトルのあらゆるペアのなす角を保つと仮定する。これをペア $\mathbf{e}_i, \mathbf{e}_j$ および $\mathbf{e}_i + \mathbf{e}_j, \mathbf{e}_i - \mathbf{e}_j$（ただし $i \neq j$）に適用し、$\mathbf{A}^\top \mathbf{A} = c^2 \mathbf{I}$ がある $c > 0$ に対して成り立つこと、すなわち $\mathbf{A} = c\mathbf{Q}$（$\mathbf{Q}$ は直交行列）と書けることを示せ。逆に、この形のすべての行列があらゆるコサイン類似度を保つことを示せ。
   3. 代わりに、$\|\mathbf{A}\mathbf{x}\| = \|\mathbf{x}\|$ がすべての $\mathbf{x}$ に対して成り立つと仮定する。$\|\mathbf{x}+\mathbf{y}\|^2$ と $\|\mathbf{x}-\mathbf{y}\|^2$ を展開することで $\mathbf{A}$ がすべてのドット積を保つことを示し、$\mathbf{A}$ が直交行列であると結論せよ。
2. **積のランク**。線形オートエンコーダは、d 次元の入力を幅 $k < d$ の隠れ層に通し、$\mathbf{x} \mapsto \mathbf{W}_2 \mathbf{W}_1 \mathbf{x}$（ただし $\mathbf{W}_1 \in \mathbb{R}^{k \times d}$、$\mathbf{W}_2 \in \mathbb{R}^{d \times k}$）と変換する。
   1. $\mathbf{A}$ を $m \times n$ 行列、$\mathbf{B}$ を $n \times p$ 行列とする。$\mathbf{AB}$ の列空間が $\mathbf{A}$ の列空間に含まれ、$\mathbf{B}$ の零空間が $\mathbf{AB}$ の零空間に含まれることを示せ。1 つ目の事実、および 2 つ目の事実とランク・退化次数の定理を用いて、$\mathrm{rank}(\mathbf{AB}) \le \min(\mathrm{rank}\,\mathbf{A}, \mathrm{rank}\,\mathbf{B})$ を証明せよ。
   2. $\mathbf{W}_2\mathbf{W}_1$ のランクが高々 k であると結論せよ。その零空間の次元がとりうる最小値はいくらか。また、このオートエンコーダがすべての入力を厳密に再構成できる部分空間の最大次元はいくらか。
   3. ともにランク 2 である 2×3 行列と 3×2 行列であって、その 2×2 の積がランク 1 になるものを構成せよ。
3. **回転と可換な行列**。行列積は一般に可換ではない。本問では $\mathbf{R}_{90}$ に対する例外をすべて求める。
   1. 2×2 行列 $\mathbf{X}$ のうち $\mathbf{X}\mathbf{R}_{90} = \mathbf{R}_{90}\mathbf{X}$ を満たすものをすべて求め、それらが $\left(\begin{smallmatrix} p & -q \\ q & p \end{smallmatrix}\right) = p\mathbf{I} + q\mathbf{R}_{90}$ であることを示せ。
   2. $\mathbf{R}_{90}^2 = -\mathbf{I}$ を用いて $(p\mathbf{I} + q\mathbf{R}_{90})(r\mathbf{I} + s\mathbf{R}_{90}) = (pr - qs)\,\mathbf{I} + (ps + qr)\,\mathbf{R}_{90}$ を示し、これらの行列が複素数 $p + qi$ と同じ規則で掛け合わされることを確かめよ。
   3. $(p, q) \neq (0, 0)$ のとき、$p\mathbf{I} + q\mathbf{R}_{90} = \sqrt{p^2+q^2}\,\mathbf{R}_\phi$（ただし角度 ϕ は $(\cos\phi, \sin\phi) = (p,q)/\sqrt{p^2+q^2}$ を満たす）と書け、スケーリングされた回転になることを示せ。さらにその行列式を計算し、$p = q = 0$ でない限り可逆であると結論せよ。
   4. 通しで用いる行列 $\mathbf{A}$ は $\mathbf{R}_{90}$ と可換か。両方の積を計算せよ。
4. **ハウスホルダー鏡映**。$\mathbf{u}$ を $\mathbb{R}^n$ 内の単位ベクトルとし、$\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ とおく。
   1. $\mathbf{H}$ が対称かつ直交であること、$\mathbf{H}\mathbf{u} = -\mathbf{u}$ であること、および $\mathbf{H}\mathbf{x} = \mathbf{x}$ がすべての $\mathbf{x}$（$\mathbf{u}$ に垂直なベクトル）に対して成り立ち、したがって $\mathbf{H}$ が法線ベクトル $\mathbf{u}$ を持つ超平面に関する鏡映であることを示せ。
   2. $\mathbf{H} = \mathbf{I} - 2\mathbf{P}$（ただし $\mathbf{P} = \mathbf{u}\mathbf{u}^\top$）と書き、$\mathbf{u}$ への射影（33.1 節）を用いてこの式を説明せよ。$\mathbf{x} - 2\mathbf{P}\mathbf{x}$ は、$\mathbf{x}$ の $\mathbf{u}$ 方向の成分に何をするか。
   3. $\det \mathbf{H} = -1$ を証明せよ。その際、$\mathbf{u}$ を拡張して正規直交基底（直交行列 $\mathbf{Q} =$

$[\mathbf{u}\,\mathbf{q}_2 \cdots \mathbf{q}_n]$ の各列）を作り、$\mathbf{Q}^\top \mathbf{H}\mathbf{Q}$ を計算して乗法性を用いよ。

4. 単位ベクトル $\mathbf{u}$ であって、その鏡映が $(3,4)$ を $(5,0)$ に送るものを求め（ヒント：$\mathbf{u}$ は $(3,4)-(5,0)$ に平行である）、$\mathbf{H}$ を書き下して $\det \mathbf{H} = -1$ を確かめよ。この種の鏡映を 1 列ずつ適用していく手法こそ、数値計算ライブラリが行列を直交因子と三角因子に分解する仕組みである。

5. [code] **行列式と特異性への近さ**。正方行列が可逆であることは、その行列式が非ゼロであることと同値である。では、行列式の大きさは、行列が特異である状態にどれほど近いかをどの程度うまく測れるだろうか。
   1. 多重線形性を用いて $\det(c\mathbf{A}) = c^n \det \mathbf{A}$ を $n \times n$ 行列 $\mathbf{A}$ とスカラー c に対して証明し、行列 $\mathbf{M}$（Equation 33.19）と $c = 2$ に対してこの恒等式をコードで確かめよ。
   2. 対角線上およびそれより下に独立な標準ガウス分布に従う成分を持つ $n \times n$ 下三角行列を、$n = 20$、100、1000 に対して生成せよ。`np.linalg.slogdet` が返す対数と、三角行列の規則により丸め誤差を除いて厳密に一致するはずの対角成分の絶対値の対数和とを比較せよ。どの n から両者は食い違い始め、$n = 1000$ ではどれほどずれるか。さらに `np.linalg.cond` を $n = 20$ と $n = 100$ に対して計算し、この破綻の理由を説明せよ。（このルーチンは入力が三角行列であることを知らず、他の一般の行列と同様に行の入れ替えを伴う分解を実行する。）
   3. 本文では、左下の成分を 10^{-8} 変化させるとせん断 $\left(\begin{smallmatrix} 1 & 10^8 \\ 0 & 1 \end{smallmatrix}\right)$ が特異になると述べている。これを確かめ、さらに単一の成分に対するそれより小さな変更では特異にならないことを示せ。

6. **ランク 1 更新の行列式。●プレーナーフロー** (Rezende and Mohamed 2015) は、$\mathbf{z} \in \mathbb{R}^d$ を $f(\mathbf{z}) = \mathbf{z} + \mathbf{u}\, h(\mathbf{w}\cdot\mathbf{z} + b)$ によって変換する（ただし $\mathbf{u}, \mathbf{w}$ はベクトル、b はスカラー、h はスカラー非線形性である）。$f(\mathbf{z})$ の密度を計算するには、ヤコビ行列 $\mathbf{I} + h'(\mathbf{w}\cdot\mathbf{z}+b)\,\mathbf{u}\mathbf{w}^\top$ の行列式が必要になる。
   1. 多重線形性と交代性を用いて、$\det(\mathbf{u}\mathbf{w}^\top) = 0$ が $n \times n$ 行列（ただし $n \geq 2$）に対して成り立つことを示せ。
   2. 次の 2 つの分解を確かめよ。

$$\begin{aligned}\begin{bmatrix} \mathbf{I} & -\mathbf{u} \\ \mathbf{w}^\top & 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{w}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{u} \\ \mathbf{0}^\top & 1 + \mathbf{w}\cdot\mathbf{u} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & -\mathbf{u} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} + \mathbf{u}\mathbf{w}^\top & \mathbf{0} \\ \mathbf{w}^\top & 1 \end{bmatrix}.\end{aligned}$$

      置換公式 Equation 33.18 を用いて、最後の列が $\mathbf{e}_{n+1}$ である行列の行列式が左上の $n \times n$ ブロックの行列式に等しいことを示せ。さらに $\det(\mathbf{I} + \mathbf{u}\mathbf{w}^\top) = 1 + \mathbf{w}\cdot\mathbf{u}$ を Equation 33.20 と三角行列の規則から導け。
   3. プレーナーフローのヤコビ行列式の絶対値の対数を書き下し、その評価コストを一般の $d \times d$ 行列式のコストと比較せよ。また $h = \tanh$ のとき、$\mathbf{u}\cdot\mathbf{w}$ がどの範囲の値をとれば、すべての $\mathbf{z}$ においてヤコビ行列が可逆に保たれるか。
   4. シャーマン・モリソンの公式 Equation 33.24 を用いて、ヤコビ行列 $\mathbf{I} + c\,\mathbf{u}\mathbf{w}^\top$（ただし $c = h'(\mathbf{w}\cdot\mathbf{z}+b)$）の逆行列を書き下せ。この公式が破綻する c の値が行列式が 0 になる値と一致することを確かめ、その逆行列を 1 つのベクトルに適用するのに必要な積和演算の回数を数えよ。なお、ヤコビ行列の逆行列は f 自体の逆写像を与えるわけではない。$\mathbf{z}$ を $f(\mathbf{z})$ から復元する問題が、$\mathbf{w}\cdot\mathbf{z}$ に関する 1 つのスカラー方程式を解くことに帰着することを示せ。

7. [code] **アテンションに対するアインシュタインの総和規約**。クエリ、キー、値の各テンソルがそれぞれ形状 (n_b, n_h, n_q, d_h)、(n_b, n_h, n_k, d_h)、(n_b, n_h, n_k, d_h) を持つとする。
   1. `einsum` の文字列を、スコア $s_{bhit} = \sum_j q_{bhij} k_{bhtj}$ および値の重み付き和 $o_{bhij} = \sum_t p_{bhit}\, v_{bhtj}$（ただし p_{bhit} はアテンション重み）に対してそれぞれ書き下し、`@` と `swapaxes` を用いた計算と一致することを確かめよ。

2. 各縮約の積和演算の回数をサイズの積として数えよ。それらをすべてのクエリに 1 つの $d_h \times d_h$ 線形変換を適用するときの積和演算の回数と比較し、スコア計算のコストの方が大きくなるキー数 n_k の境界を求めよ。
3. $\mathrm{tr}(\mathbf{A}^\top \mathbf{A})$、$\mathrm{tr}(\mathbf{A}\mathbf{A}^\top)$、および $\sum_{ij} a_{ij}^2$ をそれぞれ 1 つの `einsum` としてランダムな 3×5 行列に対して書き、3 つが一致することを確かめよ。2 つの等号のうち、どちらが Equation 33.13 の具体例であり、もう一方はなぜ成り立つのか。

33.3 固有値分解

33.2 節では、行列を 1 回適用したときに空間に何が起きるかを問うた。しかし機械学習における多くの計算では、同じ行列を何度も繰り返し適用する。同一の層を持つ深層線形ネットワークは 50 層を通過すると $\mathbf{A}^{50}\mathbf{x}$ を計算し、二次目的関数に対する勾配降下法はステップごとに誤差へ同じ固定された行列を掛ける。いずれの場合も知りたいのは、信号が増大するか、減衰して消えるか、あるいは一定の値に落ち着くか、そしてそれがどれほど速く進むかである。行列のべき乗を具体的に掛け算すれば数値は得られるが、なぜそうなるのかの説明にはならない。

2 つのウェブページ間のランダムウォークは、この問いの最小の例である。ページ 1 にいるサーファーがそれぞれ確率 $1/2$ でページ 1 に留まるかページ 2 へクリックして移動し、ページ 2 からは確率 $3/4$ でそのまま留まり、確率 $1/4$ でページ 1 に戻るとしよう。$\mathbf{p}_k$ を k 回のクリック後に各ページにいる確率を並べたベクトルとする。1 回クリックするごとに、このベクトルには次の固定された行列が掛けられる。

$$\mathbf{p}_{k+1} = \mathbf{P}\mathbf{p}_k, \qquad \mathbf{P} = \begin{pmatrix} 1/2 & 1/4 \\ 1/2 & 3/4 \end{pmatrix},$$

この行列の各列はそれぞれのページからのクリック確率を記録しているため、列の和は 1 になる。ページ 1 から出発して $\mathbf{p}_0 = (1, 0)$ とすると、最初の数回の分布は $(0.5, 0.5)$、$(0.375, 0.625)$、$(0.344, 0.656)$、$(0.336, 0.664)$ と推移する。クリックを重ねるたびに変化量は小さくなり、ランダムウォークがある一定の分布へ落ち着いていくように見える。50 回クリックした後にサーファーはどこにおり、分布はどれほどの速さで収束するのだろうか。PageRank は、まさにこれと同じ問いをウェブ全体に対して投げかけている。

鍵となるのは、行列が単なる定数倍として作用する特別な方向、すなわち固有ベクトル（eigenvector）を座標軸として初期ベクトルを書き直すことである。その座標系では、各成分がスカラーのべき乗 λ^k として独立に時間発展する。

1.3 節では、行列が伸縮のみを施す方向としてすでに固有ベクトルに出会った。対称行列 $\mathbf{S} = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ は $(1, 1)$ を 3 倍に、$(1, -1)$ を 1 倍に引き伸ばし、任意の対称行列に対して $\mathbf{A}^k\mathbf{x} = \sum_i c_i \lambda_i^k \mathbf{v}_i$ が成り立った。本節では対称性の仮定を外す。任意の正方行列の固有値（eigenvalue）を求める方法を学び、空間全体を張る固有ベクトルの組がいつ存在し、存在しないときには何がそれに代わるのかを確認したうえで、なぜ対称性があるとこれほど豊かな性質が保証されるのかを証明する。

なかでも特に重要な結果は 3 つある。固有ベクトル展開 Equation 33.28（$\mathbf{A}^k\mathbf{x}$ の展開）、スペクトル定理（spectral theorem）、そして二次形式（quadratic form）を重み付き二乗和として表す Equation 33.36 である。先を急ぐ読者は、33.3.1.3 節から直接 33.3.2.1 節へ進んでよい。行列が大きいときに固有値の位置をどう見積もって計算するか、また固有値だけでは判断を誤る状況とはどのようなものかという 2 つの実践的な問いは、33.4 節に譲る。本節全体を通して計算には NumPy を用いる。NumPy の `eig` 関数は、実行列が複素固有値を持つ場合には複素数をそのまま返す。

```
import numpy as np
```

33.3.1 固有値と対角化

33.3.1.1 固有ベクトルと固有値

特定の入力ベクトルに対しては、行列 $\mathbf{P}$ はあたかも 1 つの数であるかのようにスケール倍するだけに働く。$\mathbf{P}$ を $(1,2)$ に適用すると $\mathbf{P}(1,2) = (\frac{1}{2}+\frac{1}{2},\ \frac{1}{2}+\frac{3}{2}) = (1,2)$ となってまったく変化しない一方、$\mathbf{P}$ を $(1,-1)$ に適用すると $(\frac{1}{2}-\frac{1}{4},\ \frac{1}{2}-\frac{3}{4}) = \frac{1}{4}(1,-1)$ となる。第 1 の方向に沿っては $\mathbf{P}$ は 1 倍として作用し、第 2 の方向に沿っては $1/4$ 倍として作用する。

一般に、ゼロでないベクトル $\mathbf{v}$ は、正方行列 $\mathbf{A}$ と対応する**固有値** λ に対して次の関係を満たすとき、その行列の**固有ベクトル**と呼ばれる。

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}. \tag{33.25}$$

幾何学的には、$\mathbf{A}$ は固有ベクトルを倍率 λ で伸縮させるだけであり、$\lambda < 0$ のときはその向きを反転させる。サーファーの行列 $\mathbf{P}$ に対する固有対（eigenpair）$(\lambda, \mathbf{v})$ は、$(1, (1,2))$ と $(\frac{1}{4}, (1,-1))$ である。零ベクトルはすべての λ に対して Equation 33.25 を自明に満たしてしまうため、固有ベクトルからは除外する。しかし、$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ を満たすすべての解（λ は固定）は、$\mathbf{0}$ を含めると線形部分空間をなし、これを λ の**固有空間**（eigenspace）と呼ぶ。

一方、固有値 $\lambda = 0$ は完全に正当な固有値である。これは $\mathbf{A}\mathbf{v} = \mathbf{0}$ がゼロでないある $\mathbf{v}$ に対して成り立つことを意味するから、正方行列が特異（singular）であることと、0 がその固有値の 1 つであることは同値になる。また、実行列が実固有ベクトルを持つとは限らないことにも注意しよう。90° の回転は、平面上のゼロでないあらゆる方向を回転させてしまう。その固有値は複素数になるが、この場合は 33.3.1.6 節で詳しく扱う。

可逆な 2×2 行列が単位円に作用してできる楕円の主軸と、固有ベクトルとを混同しがちであるが、Figure 33.16 に示すとおり両者は別の概念である。対称行列 $\mathbf{S}$ では、確かに固有ベクトルは楕円の軸上に並ぶ。しかし上三角行列 $\left(\begin{smallmatrix}1 & 1\\ 0 & 2\end{smallmatrix}\right)$ ではそうなっていない。その固有ベクトルは $(1,0)$（$\lambda=1$）と $(1,1)$（$\lambda=2$）であるのに対し、楕円の軸は 58° と -32° の方向を向いている（その求め方は 33.5 節で示す）。楕円の軸が最大および最小の伸縮方向を表すのに対し、固有ベクトルはまったく回転しない方向を表す。対称行列ではこれら 2 組の方向が常に一致することが 33.3.2.1 節のスペクトル定理で証明されるが、一般の行列では一致するとは限らない。

33.3.1.2 特性多項式

$\mathbf{P}$ の固有ベクトルは観察によって見つけたが、より大きな行列ではその方法は通用しない。任意の正方行列の固有値を体系的に求めるには、$\lambda\mathbf{v}$ を Equation 33.25 の左辺へ移項し、$\mathbf{v}$ をくくり出す。

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0}. \tag{33.26}$$

ゼロでない解 $\mathbf{v}$ が存在するのは、$\mathbf{A}-\lambda\mathbf{I}$ が非自明な零空間（null space）を持つとき、すなわち特異であって $\det(\mathbf{A}-\lambda\mathbf{I}) = 0$ を満たすとき、かつそのときに限る。この行列式を展開すると n 次の λ の多項式が得られ、これを**特性多項式**（characteristic polynomial）$p(\lambda) = \det(\mathbf{A}-\lambda\mathbf{I})$ と呼ぶ。その根こそが $\mathbf{A}$ の固有値にほかならない。根 λ が求まれば、斉次連立一次方程式 Equation 33.26 を解くことで対応する固有空間が得られる。

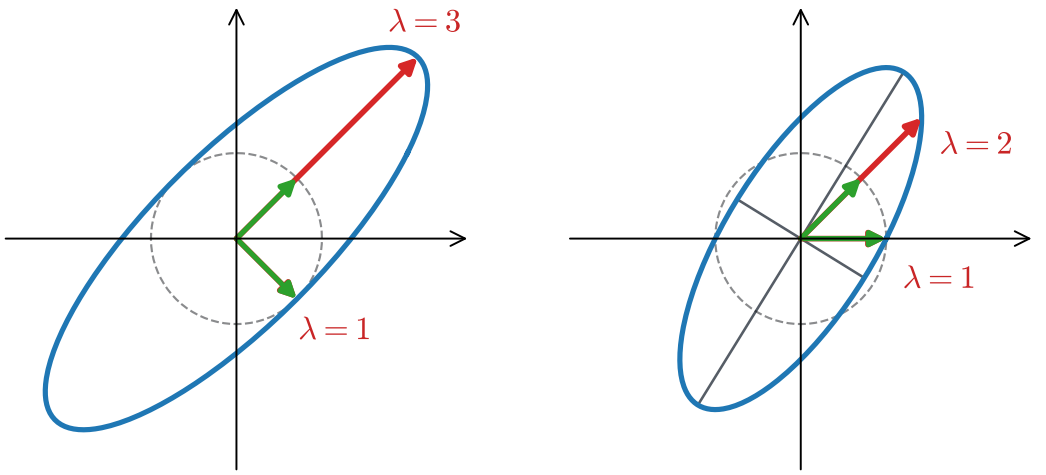


Figure 33.16: 単位円（破線）とその像（青）。左：対称行列 $\left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ では、単位固有ベクトル（緑）が楕円の軸上に乗り、それらの像（赤）はそれぞれ 3 倍と 1 倍にスケールされる（$\lambda = 1$ では 2 本の矢印が重なる）。右：上三角行列 $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 2 \end{smallmatrix}\right)$ では、固有ベクトル $(1, 0)$ と $(1, 1)$ は楕円の軸（灰色）と一致しない。

この手順をサーファーの行列に適用してみよう。

$$\det(\mathbf{P} - \lambda \mathbf{I}) = \left(\tfrac{1}{2} - \lambda\right)\left(\tfrac{3}{4} - \lambda\right) - \tfrac{1}{4} \cdot \tfrac{1}{2} = \lambda^2 - \tfrac{5}{4}\lambda + \tfrac{1}{4} = (\lambda - 1)\left(\lambda - \tfrac{1}{4}\right).$$

根は 1 と $1/4$ であり、先ほど見つけた 2 つの倍率と一致する。各根において $\mathbf{P} - \lambda \mathbf{I}$ は特異であるため、1 つの行を見るだけで固有ベクトルが求まる。$\lambda = 1$ のときは、$\mathbf{P} - \mathbf{I}$ の第 1 行から $-\frac{1}{2} v_1 + \frac{1}{4} v_2 = 0$、すなわち $v_2 = 2 v_1$ となるので、$\mathbf{v} = (1, 2)$ と選べる。$\lambda = \frac{1}{4}$ のときは、$\mathbf{P} - \frac{1}{4}\mathbf{I}$ の第 1 行から $\frac{1}{4} v_1 + \frac{1}{4} v_2 = 0$ となり、$\mathbf{v} = (1, -1)$ が得られる。なお、すべての固有ベクトルはゼロでないスカラー倍の自由度を除いてのみ定まることに留意しておこう。$\mathbf{v}$ が固有ベクトルであれば、$c\mathbf{v}$ も任意の $c \neq 0$ に対して固有ベクトルである。Equation 33.25 の両辺がともに c 倍されるからである。

実際には、この代数計算は数値計算ライブラリに任せる。NumPy の `eig` は、固有値の配列と、対応する固有ベクトルを列に並べた行列を返す。以下のコードの最終行では、すべての固有対に対して Equation 33.25 が成り立つことを一括して確かめている。

```
P = np.array([[0.5, 0.25], [0.5, 0.75]])
lam, V = np.linalg.eig(P)
print('eigenvalues:', lam)
print('eigenvectors (columns):\n', V)
print('max |P V - V diag(lam)|:', np.abs(P @ V - V * lam).max())
```

```
eigenvalues: [0.25 1.  ]
eigenvectors (columns):
 [[-0.70710678 -0.4472136 ]
 [ 0.70710678 -0.89442719]]
max |P V - V diag(lam)|: 0.0
```

順序、スケール、符号はいずれもライブラリの実装次第である。ここでは `eig` は固有値を $1/4$ と 1 の順で返し、各列は $(1,-1)$ および $(1,2)$ に平行な単位長のベクトルとなっている。

数値計算ライブラリは、実際に特性多項式を作って解いているわけではない。$n \geq 5$ では多項式の根をべき根で表す一般の閉じた公式が存在せず、それほど大きくない n でさえ、多項式の係数に生じる微小な丸め誤差（rounding error）に対して根が極端に敏感だからである。その代わりに `eig` は、固有値を変えることなく直交基底変換の列を順次適用し、行列を段階的に三角行列へと近づけていく。三角行列になれば、固有値は対角線上にそのまま現れる。$\lambda\mathbf{I}$ を引いても三角行列のままであり、三角行列の行列式は単に対角成分の積になるからである（33.2.3.2 節）。分解すること自体が不可能なほど巨大な行列に対しては、べき乗法（power iteration）（33.4.1.2 節）が行列・ベクトル積のみを用いて絶対値の大きい固有値を抽出する。

33.3.1.3　対角化

これで冒頭の $\mathbf{P}^{50}\mathbf{p}_0$ に関する問いに答える準備が整った。$\mathbf{P}$ の 2 つの固有ベクトルを行列 $\mathbf{V}$ の列として並べ、対応する固有値を Λ の対角線上に配置する。

$$\mathbf{V} = \begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} 1 & 0 \\ 0 & 1/4 \end{pmatrix}.$$

2 つの固有ベクトル方程式 $\mathbf{P}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ を列として横に並べると、簡潔に $\mathbf{P}\mathbf{V} = \mathbf{V}\Lambda$ と書ける。$\det \mathbf{V} = -1 - 2 = -3 \neq 0$ より行列 $\mathbf{V}$ は可逆であり、$\mathbf{V}^{-1} = \frac{1}{3}\left(\begin{smallmatrix} 1 & 1 \\ 2 & -1 \end{smallmatrix}\right)$ となる。右から $\mathbf{V}^{-1}$ を掛けると次式を得る。

$$\mathbf{P} = \mathbf{V}\Lambda\mathbf{V}^{-1}. \tag{33.27}$$

この分解を右から左へ 3 つのステップとして読もう。まず固有ベクトル座標へ変換し（$\mathbf{V}^{-1}$）、次に各座標をそれぞれの固有値で独立にスケール倍し（Λ）、最後に元の座標へ戻す（$\mathbf{V}$）。正方行列 $\mathbf{A}$ が $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$（ただし $\mathbf{V}$ は可逆行列、Λ は対角行列）と書けるとき、その行列は**対角化可能**（diagonalizable）であるといい、この分解そのものを**固有値分解**（eigendecomposition）と呼ぶ。このような分解が存在するのは、$\mathbf{A}$ が n 個の線形独立な固有ベクトルを持ち、それらで $\mathbf{V}$ の列を埋められるとき、かつそのときに限る。

対角化可能な行列のべき乗は、これで実に簡単に計算できる。k 個の $\mathbf{V}\Lambda\mathbf{V}^{-1}$ の積では、隣り合うすべての $\mathbf{V}^{-1}\mathbf{V}$ の組が打ち消し合う。

$$\mathbf{A}^k = \overbrace{(\mathbf{V}\Lambda\mathbf{V}^{-1})\cdots(\mathbf{V}\Lambda\mathbf{V}^{-1})}^{k\text{ times}} = \mathbf{V}\overbrace{\Lambda\cdots\Lambda}^{k\text{ times}}\mathbf{V}^{-1} = \mathbf{V}\Lambda^k\mathbf{V}^{-1}.$$

したがって、対角化可能な行列のべき乗はスカラーのべき乗に帰着する。$\mathbf{A}^k$ をベクトル $\mathbf{x}$ に適用し、$\mathbf{c} = \mathbf{V}^{-1}\mathbf{x}$ を固有基底（eigenbasis）における $\mathbf{x}$ の座標とすれば（すなわち $\mathbf{x} = \sum_i c_i \mathbf{v}_i$）、次式が得られる。

$$\mathbf{A}^k\mathbf{x} = \mathbf{V}\Lambda^k\mathbf{c} = \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{v}_i. \tag{33.28}$$

各固有座標は、等比数列 $c_i\lambda_i^k$ として独立に推移する。ページ 1 から出発するサーファーの場合、$\mathbf{c} = \mathbf{V}^{-1}\mathbf{p}_0 = \frac{1}{3}(1,2) = (\frac{1}{3}, \frac{2}{3})$ であるから、任意のクリック回数 k の後の分布が閉じた形で次のように求まる。

$$\mathbf{p}_k = \mathbf{P}^k \mathbf{p}_0 = \frac{1}{3}\begin{pmatrix}1\\2\end{pmatrix} + \frac{2}{3}\left(\frac{1}{4}\right)^k \begin{pmatrix}1\\-1\end{pmatrix}. \tag{33.29}$$

これで冒頭の問いは完全に解決した。k が大きくなるにつれて第 2 項は消え、分布は $\pi = (1/3, 2/3)$ に近づく。つまり長期的には、サーファーはページ 1 の 2 倍の時間をページ 2 で過ごすことになる。残る誤差は各成分で $\frac{2}{3}\,4^{-k}$ であり、常に固有ベクトル $(1,-1)$ の方向を向いている。1 回クリックするごとに誤差は 4 で割られるため、10 回のクリック後には 6×10^{-7} まで下がり、50 回のクリック後には 10^{-30} 未満となる。検算として $k = 3$ を代入すると $(\frac{1}{3} + \frac{1}{96},\ \frac{2}{3} - \frac{1}{96}) = (0.34375, 0.65625)$ となり、冒頭で計算した 3 回目の反復値と一致する。Figure 33.17 にその幾何学的な様子を示す。すべての $\mathbf{p}_k$ の成分の和は 1 であるため、反復列は直線 $x + y = 1$ 上を滑りながら、$\lambda = 1$ の固有ベクトルがこの直線と交わる点へ向かって進んでいく。

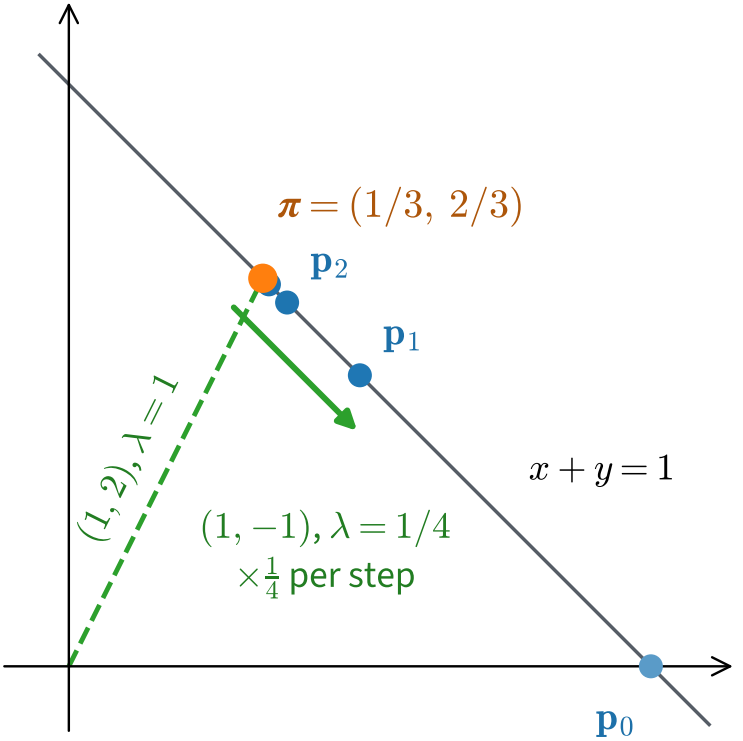


Figure 33.17: サーファーの分布 $\mathbf{p}_k$ は直線 $x + y = 1$ に沿って $\pi = (1/3, 2/3)$（固有ベクトル $(1,2)$（$\lambda = 1$）がこの直線と交わる点）へと移動していく。誤差は固有ベクトル $(1,-1)$（$\lambda = 1/4$）の方向を向き、クリックごとに 4 分の 1 に縮小する。

50 回の行列積と 2 項からなる固有展開が同じ $\mathbf{p}_{50}$ を与えることを検証し、連続する誤差の比も確認してみよう。

```
p0 = np.array([1.0, 0.0])
c = np.linalg.solve(V, p0)              # coordinates of p0 in the eigenbasis
print('P^50 p0            :', np.linalg.matrix_power(P, 50) @ p0)
```

```
    print('sum c_i lam_i^50 v_i:', V @ (c * lam ** 50))
    pi = np.array([1 / 3, 2 / 3])
    errs = np.array([np.abs(np.linalg.matrix_power(P, k) @ p0 - pi).max()
                     for k in range(8)])
    print('error ratios:', np.round(errs[1:] / errs[:-1], 4))
```

```
P^50 p0             : [0.33333333 0.66666667]
sum c_i lam_i^50 v_i: [0.33333333 0.66666667]
error ratios: [0.25 0.25 0.25 0.25 0.25 0.25 0.25]
```

2 つの計算結果は一致し、隣り合う誤差の比はすべて 0.25 となっている。固有値 1 が $\mathbf{p}_0$ のうち生き残る成分を支配し、固有値 $1/4$ が減衰して消える成分を支配している。

同じ手法は、対角化可能な行列に対する他の関数にもそのまま適用できる。どの固有値も 0 でなければ、$\mathbf{A}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{V}^{-1}$ となり、対角線上の各固有値の逆数をとるだけで逆行列が求まる。同様に、行列指数関数（matrix exponential）$e^{\mathbf{A}t} = \mathbf{V}e^{\Lambda t}\mathbf{V}^{-1}$（線形微分方程式 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ の解を与える）も、各固有値の指数関数をとることで構成できる。これについては 38.1 節で詳しく展開する。

相似。固有値分解は、$\mathbf{P}$ と Λ が 2 つの異なる基底で表されたまったく同じ線形変換であることを示している。一般に、$\mathbf{A}' = \mathbf{B}\mathbf{A}\mathbf{B}^{-1}$ という関係（ただし $\mathbf{B}$ はある可逆行列）で結ばれる 2 つの行列は**相似**（similar）であるという。次の命題は、基底を取り替えても行列の固有値は変わらないことを保証する。証明には 33.2 節で学んだ行列式の乗法性を用いる。

$$\det(\mathbf{X}\mathbf{Y}) = \det\mathbf{X}\,\det\mathbf{Y},$$

特にこの性質から、$\det(\mathbf{B}^{-1}) = 1/\det\mathbf{B}$ が $\mathbf{B}\mathbf{B}^{-1} = \mathbf{I}$ より従う。

命題（相似な行列は特性多項式を共有する）。 任意の正方行列 $\mathbf{A}$ と任意の可逆行列 $\mathbf{B}$ に対して、次が成り立つ。

$$\det(\mathbf{B}\mathbf{A}\mathbf{B}^{-1} - \lambda\mathbf{I}) = \det(\mathbf{A} - \lambda\mathbf{I}). \tag{33.30}$$

証明。$\lambda\mathbf{I} = \mathbf{B}(\lambda\mathbf{I})\mathbf{B}^{-1}$ と書き換えて $\mathbf{B}$ と $\mathbf{B}^{-1}$ をくくり出すと、$\det(\mathbf{B}\mathbf{A}\mathbf{B}^{-1} - \lambda\mathbf{I}) = \det\big(\mathbf{B}(\mathbf{A} - \lambda\mathbf{I})\mathbf{B}^{-1}\big) = \det\mathbf{B}\,\det(\mathbf{A} - \lambda\mathbf{I})\,\det\mathbf{B}^{-1} = \det(\mathbf{A} - \lambda\mathbf{I})$ となる。$\blacksquare$

したがって、相似な行列は根としての重複度も含めて同じ固有値を持ち、特性多項式によって決まる量はすべて**相似不変量**（similarity invariant）となる。最も有用な 2 つの不変量は行列式とトレースである。任意の対角化可能な行列に対して、次が成り立つ。

$$\det(\mathbf{A}) = \det(\mathbf{V}\Lambda\mathbf{V}^{-1}) = \det(\mathbf{V})\det(\Lambda)\det(\mathbf{V}^{-1}) = \det(\Lambda) = \lambda_1\lambda_2\cdots\lambda_n. \tag{33.31}$$

すなわち、行列式は固有値の積に等しい。$\mathbf{V}$ がどのような体積の歪みをもたらしても $\mathbf{V}^{-1}$ がそれを元に戻すため、Λ による各軸のスケーリングだけが残る。トレースについては、その巡回性（33.2.1.4 節）を用いる。

$$\operatorname{tr}(\mathbf{X}\mathbf{Y}) = \operatorname{tr}(\mathbf{Y}\mathbf{X}),$$

これは積が正方行列になる任意の $\mathbf{X}$ と $\mathbf{Y}$ に対して成り立つ。$\mathbf{X} = \mathbf{V}$、$\mathbf{Y} = \Lambda\mathbf{V}^{-1}$ と置けば、次が得られる。

$$\mathrm{tr}(\mathbf{A}) = \mathrm{tr}(\mathbf{V}\Lambda\mathbf{V}^{-1}) = \mathrm{tr}(\Lambda\mathbf{V}^{-1}\mathbf{V}) = \mathrm{tr}(\Lambda) = \lambda_1 + \cdots + \lambda_n. \tag{33.32}$$

したがって、トレースは固有値の和に等しい。サーファーの行列では、$\mathrm{tr}\,\mathbf{P} = \frac{1}{2} + \frac{3}{4} = \frac{5}{4} = 1 + \frac{1}{4}$ および $\det\mathbf{P} = \frac{3}{8} - \frac{1}{8} = \frac{1}{4} = 1 \cdot \frac{1}{4}$ と、両方の恒等式を一目で確かめられる。任意の 2×2 行列では、これら 2 つの数から $\lambda^2 - (\mathrm{tr})\lambda + \det$ の根として 2 つの固有値が直接定まる。

実は、固有値を $\mathbb{C}$ 上で根としての重複度込みで数えれば、対角化可能かどうかにかかわらず**すべての**正方行列で両方の恒等式が成り立つ。**代数学の基本定理**（fundamental theorem of algebra）（ここでは既知とする）により、すべての n 次多項式は $\mathbb{C}$ 上で完全に因数分解できる。

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \prod_{i=1}^{n} (\lambda_i - \lambda).$$

$\lambda = 0$ を代入すれば、ただちに $\det\mathbf{A} = \prod_i \lambda_i$ が得られる。トレースについては、両辺の λ^{n-1} の係数を比較する。右辺の係数は $(-1)^{n-1}\sum_i \lambda_i$ である。一方、左辺で $\mathbf{A} - \lambda\mathbf{I}$ の展開項のうち次数 $n-1$ に達するのは対角成分同士の積だけであり、その係数は $(-1)^{n-1}\sum_i a_{ii}$ となる。ゆえに $\mathrm{tr}\,\mathbf{A} = \sum_i \lambda_i$ が成り立つ。実行列の実数でない固有値は必ず複素共役の対として現れるため、その和も積も実数のまま保たれる。

最後に、対角化可能な行列のランクは、重複度込みで数えた非ゼロ固有値の個数に等しい。可逆行列 $\mathbf{V}$ と $\mathbf{V}^{-1}$ を掛けても次元は保たれるため、$\mathrm{rank}\,\mathbf{A} = \mathrm{rank}\,\Lambda$ が成り立つからである。行列が対角化可能でない場合にどうなるかは、演習問題 2 で調べる。

33.3.1.4　代数的重複度と幾何学的重複度

サーファーの行列は 2 つの相異なる固有値を持ち、そこから 2 つの線形独立な固有ベクトルが得られた。これは偶然ではない。相異なる固有値からは、常に線形独立な固有ベクトルが生じる。

命題（相異なる固有値に属する固有ベクトルの線形独立性）。相異なる固有値に属する固有ベクトルは線形独立である。特に、任意の $n \times n$ 行列が n 個の相異なる固有値を持てば、その行列は対角化可能である。

証明。矛盾を導くため、固有ベクトル $\mathbf{v}_1, \dots, \mathbf{v}_m$（対応する固有値 $\lambda_1, \dots, \lambda_m$ は互いに異なる）が線形従属であると仮定しよう。非自明な線形従属関係のうち、非ゼロの係数が最も少ないもの $\sum_{i=1}^{m} c_i\mathbf{v}_i = \mathbf{0}$ を 1 つ選び、必要なら添字を付け替えて $c_m \neq 0$ とする。もし非ゼロの項が $c_m\mathbf{v}_m = \mathbf{0}$ だけなら $\mathbf{v}_m = \mathbf{0}$ となってしまうため、この関係式には少なくとも 2 つの非ゼロ係数が含まれる。ここでこの関係式に $\mathbf{A}$ を適用し、元の関係式の λ_m 倍を引く。

$$\mathbf{0} = \mathbf{A}\sum_i c_i\mathbf{v}_i - \lambda_m \sum_i c_i\mathbf{v}_i = \sum_{i=1}^{m-1} c_i(\lambda_i - \lambda_m)\,\mathbf{v}_i.$$

すると $\mathbf{v}_m$ の項が消え、より短い関係式が残る。しかもこれは依然として非自明である。他の任意の非ゼロ係数 c_i（$i < m$）は、固有値が互いに異なるため $c_i(\lambda_i - \lambda_m) \neq 0$ として生き残るからである。これは選んだ関係式の最小性に矛盾する。$\mathbf{A}$ が n 個の相異なる固有値を持つとき、その n 個の線形独立な固有ベクトルは $\mathbb{C}^n$（すべての固有値が実数なら $\mathbb{R}^n$）の基底をなすため、$\mathbf{V}$ は可逆となる。$\blacksquare$

固有値に重複がある場合は、2 種類の重複度を区別して追う必要がある。λ が特性多項式の根として現れる回数を**代数的重複度**（algebraic multiplicity）$a(\lambda)$ と呼ぶ。一方、その固有空間の次

元、すなわちその固有値がもたらす線形独立な固有ベクトルの個数を**幾何学的重複度**（geometric multiplicity）$g(\lambda)$ と呼ぶ。

命題。 正方行列のすべての固有値 λ に対して、$1 \le g(\lambda) \le a(\lambda)$ が成り立つ。

証明。 $\mathbf{A} - \lambda\mathbf{I}$ は特異であるから、その零空間は非自明であり、$g \ge 1$ が従う。上界を示すため、固有空間の基底 $\mathbf{v}_1, \dots, \mathbf{v}_g$ をとり、それを拡張して $\mathbb{C}^n$ の基底とし、これらの基底ベクトルを列として並べた行列を $\mathbf{B}$ とする。このとき、最初の g 列について $\mathbf{AB}$ の列は単に $\lambda\mathbf{v}_1, \dots, \lambda\mathbf{v}_g$ となるから、この新しい基底のもとで行列は次のブロック上三角行列の形をとる。

$$\mathbf{B}^{-1}\mathbf{A}\mathbf{B} = \begin{pmatrix} \lambda \mathbf{I}_g & \mathbf{X} \\ \mathbf{0} & \mathbf{Y} \end{pmatrix}.$$

Equation 33.30 より、この行列は特性多項式 p を $\mathbf{A}$ と共有する。いま固定した固有値を表すのに用いている記号と区別するため、多項式の変数を t と書こう。λ に対して $t\mathbf{I}$ を引いてもブロック上三角構造は保たれ、対角ブロックは $(\lambda - t)\mathbf{I}_g$ と $\mathbf{Y} - t\mathbf{I}$ になる。ブロック上三角行列の行列式は対角ブロックの行列式の積に等しいから（33.2.3.2 節）、$p(t) = (\lambda - t)^g \det(\mathbf{Y} - t\mathbf{I})$ を得る。したがって $(\lambda - t)^g$ は特性多項式を割り切り、$a \ge g$ が成り立つ。$\blacksquare$

1 つの固有値から得られる線形独立な固有ベクトルの個数が、根としての重複度を超えることは決してないが、それより少なくなることはあり得る。いま各固有空間から基底を 1 組ずつ選び、選んだベクトルをすべて集めよう。これらは全体としても線形独立である。実際、それらの線形結合が零ベクトルに等しいとして固有値ごとに項をまとめれば、各グループは零ベクトルか固有ベクトルのいずれかになり、相異なる固有値に関する命題からすべてのグループが零ベクトルでなければならないからである。ゆえに行列が持つ線形独立な固有ベクトルは高々 $\sum_{\lambda} g(\lambda)$ 個である。$\mathbb{C}$ 上では代数的重複度の和は n に等しく、かつすべての固有値に対して $g(\lambda) \le a(\lambda)$（各 λ ごと）が成り立つため、明快な判定条件が得られる。すなわち、**$\mathbf{A}$ の固有ベクトルが張る空間が $\mathbb{C}^n$ 全体に一致するのは、$g(\lambda) = a(\lambda)$ がすべての複素固有値 λ に対して成り立つとき、かつそのときに限る。**

実行列が**実**固有ベクトルからなる基底を持つためには、さらにその固有値がすべて実数でなければならない。90° の回転行列 $\mathbf{R}_{90} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ は、なぜこの追加条件が不可欠なのかを示している。その固有値 $\pm i$ はそれぞれ $g = a = 1$ を満たすため完全な複素固有基底を持つが、実固有ベクトルは 1 つも存在しない。

固有値が重複する場合には、対角化可能性そのものが成り立たなくなることがある。最も単純な反例は、33.2.1.2 節で登場した**せん断**（shear）である。

$$\mathbf{T} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{33.33}$$

この行列の特性多項式は $(1 - \lambda)^2$ であるから、$\lambda = 1$ の代数的重複度は 2 である。ところが $(\mathbf{T} - \mathbf{I})\mathbf{v} = \mathbf{0}$ は $\left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)\mathbf{v} = \mathbf{0}$ となり、これから $v_2 = 0$ が強制される。固有空間は $(1, 0)$ が張る直線だけであり、幾何学的重複度は 1 にとどまる。したがって固有基底は存在せず、このせん断は対角化可能でない。ある固有値に対して $g(\lambda) < a(\lambda)$ となる行列は**欠陥行列**（defective）と呼ばれる。Figure 33.18 を見れば、固有ベクトルが足りない理由が視覚的にわかる。せん断はすべての水平線をその高さに比例して横へ滑らせるため、原点を通る直線で向きが保たれるのは水平軸ただ 1 本だけなのである。（対照的に、重複固有値を持つ対角化可能な 2×2 行列は、その固有値を λ とすれば $\lambda\mathbf{I}$ にならざるを得ず、あらゆる方向を保つ。）

せん断を反復適用すると、欠陥性が行列のべき乗の振る舞いをどう変えるかがわかる。$\mathbf{T}^k = \left(\begin{smallmatrix} 1 & k \\ 0 & 1 \end{smallmatrix}\right)$ となるため、$\mathbf{T}^k\mathbf{e}_2 = (k, 1)$ は唯一の固有値が 1 であるにもかかわらず際限なく増大する。純粋

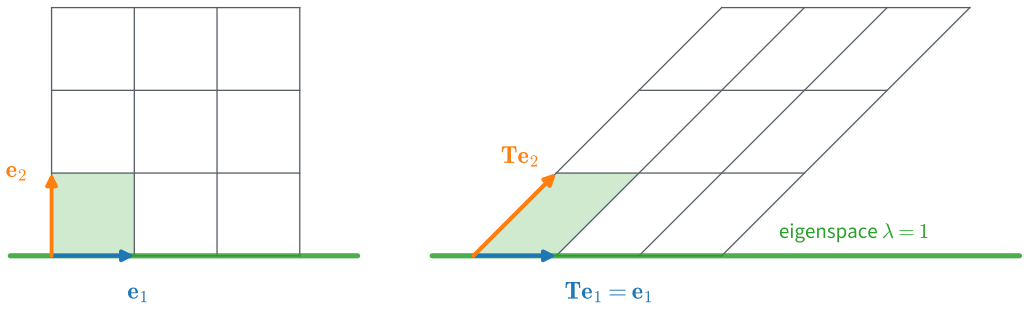


Figure 33.18: 平面に作用する欠陥を持つせん断。すべての水平線がその高さに比例した量だけ右へ滑ってグリッドを傾ける一方、x 軸（緑）は自分自身に写る。すなわち $\mathbf{T}\mathbf{e}_1 = \mathbf{e}_1$ が $\lambda = 1$ に対する唯一の固有方向である。もう一方の基底ベクトルは水平成分を得て $\mathbf{T}\mathbf{e}_2 = (1, 1)$ となるため、向きが保たれる第 2 の方向は存在せず、代数的重複度 2 に対して幾何学的重複度は 1 となる。

な固有展開からは、このような線形の因子 k は決して生じない。次項では、この因子がどこから現れるのかを説明する。

33.3.1.5 ジョルダン標準形

幾何学的重複度が代数的重複度に満たないとき、固有基底は存在しない。では、代わりにどのような構造が現れるのだろうか。本書の以降の議論において最も重要な帰結は 1 つ、すなわち欠陥固有値は λ^k に k の多項式を掛けるということである。複素数体上では、すべての正方行列は次の**ジョルダンブロック**（Jordan block）から構成されるブロック対角行列と相似になる。

$$\mathbf{J}_m(\lambda) = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix},$$

これは $m \times m$ のブロックであり、対角線上に単一の固有値 λ が並び、そのすぐ上の優対角線（superdiagonal）に 1 が並ぶ。この分解が**ジョルダン標準形**（Jordan normal form）であり (Horn and Johnson 2012)、本書では証明なしに用いる。Equation 33.30 により、その固有値は元の行列と一致する。

各ジョルダンブロックからは、その最初の基底ベクトルという 1 つの固有ベクトルだけが得られる。したがって λ の幾何学的重複度は λ を持つブロックの個数に等しく、代数的重複度はそれらのブロックのサイズの和に等しい。特に、行列が対角化可能であることと、すべてのジョルダンブロックのサイズが 1×1 であることは同値である。せん断 $\mathbf{T}$（Equation 33.33）は、単一のブロック $\mathbf{J}_2(1)$ そのものである。

反復適用のもとで、ジョルダンブロックは単なるスケーリング以上の働きをする。$\mathbf{J}_m(\lambda) = \lambda\mathbf{I} + \mathbf{N}$ と書こう。ここで $\mathbf{N}$ は対角線のすぐ上に 1 が並び、それ以外の成分がすべて 0 の行列である。幾何学的には、$\mathbf{N}$ は各基底ベクトルを 1 つ前の基底ベクトルへ送るシフトとして作用するため、$\mathbf{N}^m = \mathbf{0}$ となる（べき乗が零行列になる行列を**べき零**（nilpotent）と呼ぶ）。$\mathbf{I}$ と $\mathbf{N}$ は可換であるため、二項定理によってべき乗を展開できる。$m = 2$ のときは最初の 2 項より後のすべての項が $\mathbf{N}^2 = \mathbf{0}$ を含むため、次の項だけが残る。

$$\mathbf{J}_2(\lambda)^k = (\lambda \mathbf{I} + \mathbf{N})^k = \lambda^k \mathbf{I} + k\lambda^{k-1}\mathbf{N}.$$

長期的にはやはり固有値が増大と減衰を分けるものの、そこに k の多項式（次数は $m-1$、すなわち $m \times m$ ブロックのサイズより 1 小さい）が掛かるようになる。せん断のように $|\lambda| = 1$ のときは、反復列は線形にドリフトする。$|\lambda|$ が 1 よりわずかに小さいときには、さらに驚くべき現象が起きる。右上の成分 $k\lambda^{k-1}$ は最初増大し、$k \approx -1/\log|\lambda|$ 付近でピークに達してから、ようやく 0 に向かって減衰するのである。$\lambda = 0.99$ の場合、ピークは $k \approx 100$ 付近に現れ、この成分は $100 \cdot 0.99^{99} \approx 37$ に達する。この予測を数値的に確かめてみよう。

```
J = np.array([[0.99, 1.], [0., 0.99]])   # the Jordan block J_2(0.99)
for k in [1, 50, 100, 200, 800]:
    Jk = np.linalg.matrix_power(J, k)
    pred = k * 0.99 ** (k - 1)   # predicted corner entry, k * lambda^(k-1)
    print(f'k = {k:3d}   ||J^k|| = {np.linalg.norm(Jk, 2):8.3f}   '
          f'k * 0.99^(k-1) = {pred:8.3f}')
```

```
k =   1   ||J^k|| =    1.609   k * 0.99^(k-1) =    1.000
k =  50   ||J^k|| =   30.568   k * 0.99^(k-1) =   30.556
k = 100   ||J^k|| =   36.977   k * 0.99^(k-1) =   36.973
k = 200   ||J^k|| =   27.067   k * 0.99^(k-1) =   27.067
k = 800   ||J^k|| =    0.260   k * 0.99^(k-1) =    0.260
```

$\mathbf{J}$ の固有値はすべて 0.99 であるにもかかわらず、そのべき乗のノルムは約 37（$k = 100$）まで上昇し、27（$k = 200$）に留まったのち、ようやく指数因子 0.99^k が支配的になって減衰へ転じる。固有値だけを見ていても、この過渡的増大（transient growth）を予見することはできない。この現象は 33.4.2.3 節で改めて取り上げる。

ジョルダンブロックは、実行列が実固有基底を持てない 1 つ目の要因、すなわち固有ベクトルの個数そのものが足りない場合を表している。2 つ目の要因はまったく性質が異なる。実行列がある平面を**回転**させるために、その平面内に実固有ベクトルを 1 つも持たない場合である。

33.3.1.6 複素固有値

角度 θ の平面回転を考えよう（33.2.1.3 節）。

$$\mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

θ が π の倍数でない限り、$\mathbf{R}_\theta$ はゼロでないすべての実ベクトルの向きを変えてしまうため、実固有ベクトルは存在しない。その特性多項式 $\lambda^2 - 2\cos\theta\,\lambda + 1$ は、複素共役な根 $\lambda = \cos\theta \pm i\sin\theta = e^{\pm i\theta}$ を持つ。ここで絶対値 $|\lambda| = 1$ は $\mathbf{R}_\theta$ が長さを保つことを表し、偏角 $\pm\theta$ は回転角を記録している。

次に、成分を見ただけでは回転が隠れているとはわかりにくい、本章で通しで用いる 2×2 行列を見てみよう。

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}, \qquad \operatorname{tr}\mathbf{A} = 4, \quad \det\mathbf{A} = 5.$$

この行列の特性多項式は $\lambda^2 - 4\lambda + 5$ であり、その根は $\lambda = 2 \pm i = \sqrt{5}\,e^{\pm i\theta}$（ただし $\theta = \arctan\tfrac{1}{2} \approx 26.6^\circ$）となる。$2+i$ に対する固有ベクトルを求めるため、$\mathbf{A} - (2+i)\mathbf{I}$ の第 2 行を読み取ると $-w_1 + (1-i)w_2 = 0$ が得られる。$w_2 = 1$ と置けば次式を得る。

$$\mathbf{w} = \begin{pmatrix} 1 - i \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 \\ 1 \end{pmatrix}}_{\mathbf{u}} + i\,\underbrace{\begin{pmatrix} -1 \\ 0 \end{pmatrix}}_{\mathbf{v}}.$$

回転はどこに隠れているのだろうか。複素固有対を持つ任意の実行列 $\mathbf{A}$ をとり、$\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$（ただし $\lambda = a + ib$、$b \neq 0$）として、固有ベクトルを実部と虚部に分け $\mathbf{w} = \mathbf{u} + i\mathbf{v}$ と書く。$\mathbf{A}(\mathbf{u} + i\mathbf{v}) = (a + ib)(\mathbf{u} + i\mathbf{v})$ を展開して実部と虚部をそれぞれ比較すると、次の 2 式が得られる。

$$\mathbf{A}\mathbf{u} = a\mathbf{u} - b\mathbf{v}, \qquad \mathbf{A}\mathbf{v} = b\mathbf{u} + a\mathbf{v}.$$

どちらの像も $\operatorname{span}\{\mathbf{u}, \mathbf{v}\}$ の内部に留まるため、この部分空間は $\mathbf{A}$ のもとで不変である。しかもこれは正真正銘の 2 次元平面である。もし $\mathbf{u}$ と $\mathbf{v}$ が線形従属であれば、それらが張る空間は自分自身に写される 1 本の直線（$\mathbf{w} \neq \mathbf{0}$ より零空間ではない）となり、その直線上で $\mathbf{A}$ はある実スカラー μ 倍として作用することになる。すると $\mathbf{A}\mathbf{w} = \mu\mathbf{w}$ となって $\lambda = \mu$ が実数となり、$b \neq 0$ に矛盾する。そこで $\mathbf{v}$ と $\mathbf{u}$ を列に並べて $\mathbf{B} = [\mathbf{v}\ \ \mathbf{u}]$ と置くと、上の 2 つの等式は $\mathbf{A}\mathbf{B} = \mathbf{B}\left(\begin{smallmatrix} a & -b \\ b & a \end{smallmatrix}\right)$、すなわち同値な形として次式になる。

$$\mathbf{B}^{-1}\mathbf{A}\mathbf{B} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} = \underbrace{\sqrt{a^2 + b^2}}_{r} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \qquad \theta = \arg(a + ib). \tag{33.34}$$

不変平面の基底 $(\mathbf{v}, \mathbf{u})$ のもとでは、$\mathbf{A}$ は回転・スケーリングブロック $r\mathbf{R}_\theta$ として作用する。すなわち絶対値 r 倍にスケールし、偏角 θ だけ回転させる。この導出では $\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$ しか使っていないため、得られるブロックは $\mathbf{w}$ のスケールや位相の回転の選び方に依存しない。この対応は逆向きにも成り立つ。$(x, y) \in \mathbb{R}^2$ を複素数 $x + iy$ と同一視すれば、このブロックは (x, y) を $(ax - by,\ bx + ay)$ へ写すが、これは複素数 $a + ib$ の乗算にほかならない。

通しで用いる行列 $\mathbf{A}$ については、$\mathbf{B} = \left(\begin{smallmatrix} -1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ かつ $\mathbf{B}^{-1}\mathbf{A}\mathbf{B} = \left(\begin{smallmatrix} 2 & -1 \\ 1 & 2 \end{smallmatrix}\right)$ となり、$\sqrt{5}$ 倍のスケーリングと 26.6° の回転を表す。すると、1 ステップごとにベクトルの長さが一様に $\sqrt{5} \approx 2.24$ 倍になると思うかもしれない。`eig` が返す固有ベクトルからこのブロックを計算し、実際の反復列 $\mathbf{A}^k(1, 0)$ の長さを追ってみよう。

```
A = np.array([[1., 2.], [-1., 3.]])
lam, V = np.linalg.eig(A)
j = np.argmax(lam.imag)                  # the eigenvalue a + ib with b > 0
r, theta = np.abs(lam[j]), np.angle(lam[j])
u, v = V[:, j].real, V[:, j].imag
B = np.column_stack([v, u])              # basis (v, u) of the invariant plane
print(f'lambda = {lam[j]:.4f}   r = {r:.4f}   theta = {theta:.4f}')
print('B^-1 A B =\n', np.round(np.linalg.inv(B) @ A @ B, 4))
xs = [np.array([1., 0.])]
for _ in range(8):
    xs.append(A @ xs[-1])
norms = np.linalg.norm(xs, axis=1)
```

```
    print('norm ratios:', np.round(norms[1:] / norms[:-1], 2))
    x20 = np.linalg.matrix_power(A, 20) @ np.array([1., 0.])
    print(f'average rate ||A^20 x||^(1/20) = {np.linalg.norm(x20) ** (1 / 20):.3f}')
```

```
lambda = 2.0000+1.0000j   r = 2.2361   theta = 0.4636
B^-1 A B =
 [[ 2. -1.]
 [ 1.  2.]]
norm ratios: [1.41 2.92 3.45 2.76 2.27 1.88 1.5  1.54]
average rate ||A^20 x||^(1/20) = 2.252
```

eig がどの固有ベクトルを選んでも、得られるブロックは $\left(\begin{smallmatrix} 2 & -1 \\ 1 & 2 \end{smallmatrix}\right)$ となる。ところがベクトルの長さは、ステップごとに一定の倍率 2.24 で増大するわけでは**ない**。軌跡 $(1,0) \to (1,-1) \to (-1,-4) \to (-9,-11) \to (-31,-24)$ に沿って見ると、1 ステップあたりのノルムの比は約 1.4 から 3.5 の間で大きく揺れ動く。理由は、スケーリングが一様になるのが正規直交ではない座標系 $(\mathbf{v}, \mathbf{u})$ の内部に限られるからである。元の標準座標に戻すと、反復列は引き伸ばされた楕円に沿って外向きに渦を巻き、約 $2\pi/\theta \approx 13.6$ ステップで 1 周する。長期的な幾何平均 $\|\mathbf{A}^k\mathbf{x}\|^{1/k}$ だけが $\sqrt{5}$ に落ち着く（実際、2.25 に $k = 20$ までに到達する）。標準座標において剛体回転と一様なスケーリングになるのは、行列が**正規**（normal）であるときに限られる。この性質は 33.4.2.2 節で定義する。

実基底変換を除けば、すべての実正方行列はここまでに見いだした 3 つの基本動作、すなわち直線に沿った伸縮、ジョルダンブロック内のせん断、そして平面内の回転を伴うスケーリングを単独または組み合わせて構成される。この分解は**実ジョルダン標準形**（real Jordan form）として知られている (Horn and Johnson 2012)。

33.3.2 対称行列と二次形式

33.3.2.1 スペクトル定理

実ベクトルからなる正規直交固有基底（orthonormal eigenbasis）の存在が保証されるのは、どのような行列だろうか。対称性は、前節で出会った 2 つの障害をどちらも取り除く。複素固有値も、固有ベクトルの不足も生じなくなるのである。1.3 節の対称行列 $\mathbf{S}$ がその好例である。その単位固有ベクトル $(1,1)/\sqrt{2}$ と $(1,-1)/\sqrt{2}$ は互いに直交しており、次の分解を与える。

$$\mathbf{S} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \underbrace{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}}_{\mathbf{V}} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \underbrace{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}}_{\mathbf{V}^{\top}}.$$

固有ベクトル行列 $\mathbf{V}$ は直交行列（$\mathbf{V}^{\top}\mathbf{V} = \mathbf{I}$）であるため、その逆行列は単なる転置になる。スペクトル定理は、これが**すべての**実対称行列に対して成り立つことを保証する。これは本章の中心定理であり、正定値性、レイリー商（Rayleigh quotient）、そして 33.5 節の内容すべてがこの定理の上に築かれる。

定理（スペクトル定理）。 すべての実対称行列 $\mathbf{A}$ は、実固有値と実固有ベクトルからなる正規直交基底を持つ。同値な表現として、

$$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{\top}, \tag{33.35}$$

と書ける。ここで $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ は実行列であり、$\mathbf{V}$ は直交行列である。

これを 3 段階で証明する。第 1 にすべての固有値が実数であること、第 2 に相異なる固有値に属する固有ベクトルが直交すること、第 3 に空間全体を張る正規直交固有基底が常に存在することを示す。複素ベクトルに対してはエルミート内積（Hermitian inner product）$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{*}\mathbf{y}$ を用いる。ここで $\mathbf{x}^{*}$ は共役転置（conjugate transpose）であり、実ベクトル上では通常のドット積に帰着する。外部から借りる事実は、特性多項式が少なくとも 1 つの複素根を持つこと（代数学の基本定理）、すなわち $\mathbb{C}$ 上に少なくとも 1 組の固有対が存在することだけである。(i) と (ii) はそれぞれわずか 2 行で済む。本質的な議論は (iii) にある。

証明 (i)：固有値は実数である。 いったん複素成分を許して、$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$ かつ $\mathbf{v} \neq \mathbf{0}$ とする。$\mathbf{A}$ は実対称行列であるから $\mathbf{A}^{*} = \mathbf{A}^{\top} = \mathbf{A}$ が成り立ち、したがって次を得る。

$$\lambda\, \mathbf{v}^{*}\mathbf{v} = \mathbf{v}^{*}\mathbf{A}\mathbf{v} = (\mathbf{A}\mathbf{v})^{*}\mathbf{v} = \bar{\lambda}\, \mathbf{v}^{*}\mathbf{v}.$$

$\mathbf{v}^{*}\mathbf{v} = \|\mathbf{v}\|^{2} > 0$ であるから、両辺をこれで割れば $\lambda = \bar{\lambda}$ となり、固有値は実数でなければならない。するとその固有ベクトルは実係数の斉次連立一次方程式 $(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0}$ の解となるため、やはり実ベクトルとして選べる。$\blacksquare$

証明 (ii)：相異なる固有値の固有ベクトルは直交する。 $\mathbf{A}\mathbf{u} = \lambda \mathbf{u}$ および $\mathbf{A}\mathbf{v} = \mu \mathbf{v}$ とし、$\lambda \neq \mu$ とする。(i) により両固有値および両固有ベクトルは実数としてよい。$\mathbf{A} = \mathbf{A}^{\top}$ を用いて内積の中で $\mathbf{A}$ をもう一方へ移すと、次式が成り立つ。

$$\lambda \langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{A}\mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{A}\mathbf{v} \rangle = \mu \langle \mathbf{u}, \mathbf{v} \rangle.$$

よって $(\lambda - \mu)\langle \mathbf{u}, \mathbf{v} \rangle = 0$ となる。$\lambda \neq \mu$ であるから、$\langle \mathbf{u}, \mathbf{v} \rangle = 0$ でなければならない。$\blacksquare$

証明 (iii)：正規直交固有基底の存在。 n に関する帰納法で示す。$n = 1$ のとき主張は明らかである。$n > 1$ とすると、(i) より実固有対 $(\lambda_1, \mathbf{v}_1)$ で $\|\mathbf{v}_1\| = 1$ を満たすものが存在する。ここで $(n-1)$ 次元の直交補空間（orthogonal complement）$U = \mathbf{v}_1^{\perp}$ を考えよう。対称性により、$\mathbf{A}$ は U を再びそれ自身の中へ写す。実際、任意の $\mathbf{x} \in U$ に対して次が成り立つ。

$$\langle \mathbf{A}\mathbf{x}, \mathbf{v}_1 \rangle = \langle \mathbf{x}, \mathbf{A}\mathbf{v}_1 \rangle = \lambda_1 \langle \mathbf{x}, \mathbf{v}_1 \rangle = 0.$$

U の正規直交基底を 1 組選び、そのベクトルを列として並べた行列を $\mathbf{Q} \in \mathbb{R}^{n \times (n-1)}$ とする。$\mathbf{A}\mathbf{Q}$ の各列はすべて U に属するため、$\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{C}$ と書ける。ここで $\mathbf{C} = \mathbf{Q}^{\top}\mathbf{A}\mathbf{Q}$ は $\mathbf{A}$ を U に制限した変換を表す。$\mathbf{C}$ が対称行列であり（$\mathbf{C}^{\top} = \mathbf{Q}^{\top}\mathbf{A}^{\top}\mathbf{Q} = \mathbf{C}$）、サイズが $(n-1) \times (n-1)$ であることに注目しよう。帰納法の仮定より、$\mathbf{C} = \mathbf{V}' \Lambda' \mathbf{V}'^{\top}$ が、ある直交行列 $\mathbf{V}'$ に対して成り立つ。そこで $\mathbf{Q}\mathbf{V}'$ の各列を調べると、$(\mathbf{Q}\mathbf{V}')^{\top}\mathbf{Q}\mathbf{V}' = \mathbf{V}'^{\top}\mathbf{V}' = \mathbf{I}$ よりこれらは正規直交であり、U に属するから $\mathbf{v}_1$ と直交し、さらに $\mathbf{A}$ の固有ベクトルにもなっている。なぜなら $\mathbf{A}\mathbf{Q}\mathbf{V}' = \mathbf{Q}\mathbf{C}\mathbf{V}' = \mathbf{Q}\mathbf{V}'\Lambda'$ が成り立つからである。これらを $\mathbf{v}_1$ の後ろに並べれば、直交行列 $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{Q}\mathbf{V}' \end{bmatrix}$ が得られ、$\mathbf{A}\mathbf{V} = \mathbf{V}\Lambda$ を満たす。$\blacksquare$

幾何学的には、Equation 33.35 は任意の対称な線形変換を 3 つの明快な操作へと分解する。まず $\mathbf{V}^{\top}$ が正規直交固有ベクトルを座標軸に揃える。これは純粋な回転か、それに鏡映を組み合わせた変換である（上の $\mathbf{S}$ では $\det \mathbf{V} = -1$ であった）。次に Λ が各座標軸を対応する固有値倍に伸縮させる。最後に $\mathbf{V}$ が軸を元の向きへ戻す。これにより、Figure 33.16 の左パネルが $\mathbf{S}$ について図示した性質が、すべての対称行列に対して証明されたことになる。

長方形か正方かを問わず、あらゆる行列 $\mathbf{A}$ から対称行列 $\mathbf{A}^{\top}\mathbf{A}$ を作ることができる。その二次形式 $\mathbf{x}^{\top}\mathbf{A}^{\top}\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|^{2}$ は決して負にならない。次項ではこの性質に名前を付け、判定法を導き、活用していく。

33.3.2.2 正定値性

二次形式 $\mathbf{x}^\top \mathbf{A}\mathbf{x}$ が負の値をとることはあるだろうか。損失曲面の臨界点が最小点であるか、行列が正当な共分散行列であるか、そして最小二乗法の問題が一意解を持つかは、この 1 つの問いにかかっている。対称行列 $\mathbf{A}$ が次を満たすとき、**半正定値**（positive semidefinite、PSD）であるといい、$\mathbf{A} \succeq 0$ と書く。

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0 \quad \text{for all } \mathbf{x},$$

さらに、ゼロでないすべてのベクトルに対して不等号が狭義に成り立つとき、**正定値**（positive definite、PD）であるといい、$\mathbf{A} \succ 0$（すべての非ゼロ $\mathbf{x}$ に対して厳密に正）と書く。たとえば $\mathbf{x}^\top \mathbf{S}\mathbf{x} = 2x_1^2 + 2x_1x_2 + 2x_2^2 = x_1^2 + x_2^2 + (x_1+x_2)^2$ は $\mathbf{x} \neq \mathbf{0}$ のとき常に正であるから $\mathbf{S} \succ 0$ であるが、$\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ は $\mathbf{e}_2$ に対して -1 をとるため PD でも PSD でもない。手計算による平方完成は高次元になるとすぐに現実的でなくなるが、固有値を見れば任意の対称行列に対してこの問いに明快な決着がつく。

命題（固有値による定値性の判定）。 *対称行列 $\mathbf{A}$ に対して、次が成り立つ。*

$$\mathbf{A} \succeq 0 \iff \lambda_i \geq 0 \text{ for all } i, \qquad \mathbf{A} \succ 0 \iff \lambda_i > 0 \text{ for all } i.$$

証明。 スペクトル定理を適用して $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top$ と書き、長さを保つ直交基底変換によって固有ベクトル座標 $\mathbf{y} = \mathbf{V}^\top \mathbf{x}$ へ移ると、次式を得る。

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \mathbf{x}^\top \mathbf{V}\Lambda\mathbf{V}^\top \mathbf{x} = \mathbf{y}^\top \Lambda \mathbf{y} = \sum_{i=1}^{n} \lambda_i \, y_i^2 = \sum_{i=1}^{n} \lambda_i \, (\mathbf{v}_i^\top \mathbf{x})^2. \tag{33.36}$$

固有ベクトル座標では、二次形式は固有値を重みとする単純な重み付き二乗和になる。すべての $\lambda_i \geq 0$ であれば、この和が非負であることは明らかである。逆に $\mathbf{x} = \mathbf{v}_j$ を代入すると $\mathbf{y} = \mathbf{e}_j$ かつ $\mathbf{v}_j^\top \mathbf{A}\mathbf{v}_j = \lambda_j$ となるため、$\lambda_j \geq 0$ でなければならない。正定値の場合も、$\mathbf{y} \neq \mathbf{0}$ が $\mathbf{x} \neq \mathbf{0}$ のとき常に成り立つことから、等号を除いたまったく同じ議論により従う。$\blacksquare$

Equation 33.36 を $\mathbf{S}$ と $\mathbf{x} = \mathbf{e}_1$ で確かめてみよう。固有座標は $\mathbf{y} = \mathbf{V}^\top \mathbf{e}_1 = (1/\sqrt{2}, 1/\sqrt{2})$ であり、重み付き和は $3 \cdot \tfrac{1}{2} + 1 \cdot \tfrac{1}{2} = 2$ となって $\mathbf{e}_1^\top \mathbf{S}\mathbf{e}_1 = s_{11}$ と一致する。深層学習で正定値性が登場する場面の大半は、次の 3 つの帰結で説明できる。

- **PD ならば可逆である。** $\mathbf{A} \succ 0$ かつ $\mathbf{A}\mathbf{x} = \mathbf{0}$ とすると $\mathbf{x}^\top \mathbf{A}\mathbf{x} = 0$ となり、これから $\mathbf{x} = \mathbf{0}$ が導かれる。
- **グラム行列と共分散行列は PSD である。** 任意の行列 $\mathbf{X}$ に対して $\mathbf{x}^\top \mathbf{X}^\top \mathbf{X}\mathbf{x} = \|\mathbf{X}\mathbf{x}\|^2 \geq 0$ であるから、$\mathbf{X}^\top \mathbf{X} \succeq 0$ が成り立つ。これが狭義に PD となるのは、$\mathbf{X}$ が列フルランク（full column rank）を持つとき（すなわち $\mathbf{X}\mathbf{x} = \mathbf{0}$ から $\mathbf{x} = \mathbf{0}$ が従うとき）、かつそのときに限る。中心化したデータの標本共分散は、グラム行列をサンプル数で割ったものであるため、常に PSD である。
- **PD のヘッセ行列は狭義の極小値を保証する。** f が 2 回連続微分可能で $\nabla f(\mathbf{x}^*) = \mathbf{0}$ を満たし、そのヘッセ行列 $\nabla^2 f(\mathbf{x}^*)$（2 階偏微分の対称行列）が PD であるとする。微小なステップ δ に対し、二次テイラー展開（34.2.3.2 節）は $f(\mathbf{x}^* + \delta) = f(\mathbf{x}^*) + \tfrac{1}{2}\delta^\top \nabla^2 f(\mathbf{x}^*)\delta + o(\|\delta\|^2)$ と書ける。Equation 33.36 より二次項は少なくとも $\tfrac{1}{2}\lambda_{\min}\|\delta\|^2 > 0$ となるため、この関数は $\mathbf{x}^*$ で狭義の極小値をとる。なお、この条件は十分条件であって必要条件ではない。実際、$f(x) = x^4$ は 0 において狭義の最小値をとるが、そこでの二階導関数は 0 である。

正定値性には、**コレスキー分解**（Cholesky factorization）による高速なアルゴリズム的判定法もある。すなわち、$\mathbf{A} \succ 0$ が成り立つのは、$\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ を満たす対角成分が正の下三角行列 $\mathbf{L}$ が存在するとき、かつそのときに限る。一方の向きはただちに従う。そのような $\mathbf{L}$ は可逆であるから、$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \|\mathbf{L}^\top \mathbf{x}\|^2 > 0$ がゼロでないすべての $\mathbf{x}$ に対して成り立つ。コレスキー分解を試みることは正定値性を判定する標準的な数値計算手法であり、すべての固有値を計算するよりはるかに低コストで済む。さらに副産物として、同じコレスキー因子 $\mathbf{L}$ を用いれば、独立な標準ガウス分布からのサンプル $\mathbf{z}$ を、サンプル $\mathbf{L}\mathbf{z}$（共分散は $\mathbf{L}\mathbf{L}^\top = \mathbf{A}$）へと変換できる。

幾何学的には、固有値の符号が曲面 $z = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ の形状を決定する（Figure 33.19）。すべての固有値が正ならお椀型になり、固有値 0 の方向があればその方向に沿って平坦な谷樋型になり、符号が混在していれば鞍型になる。

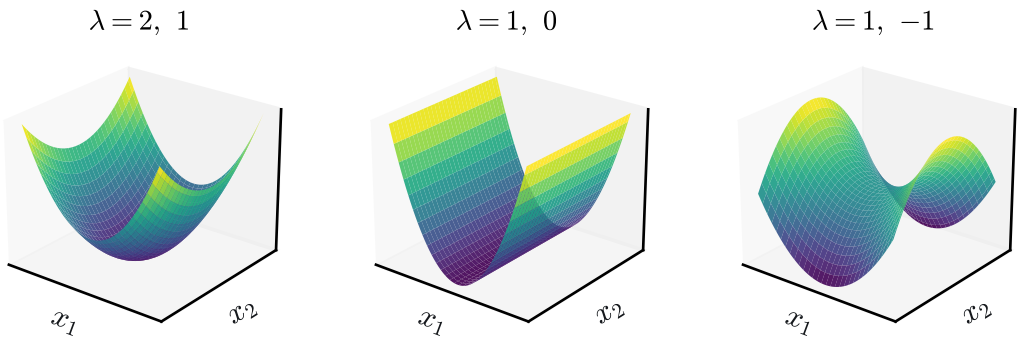


Figure 33.19: 3 つの対角行列に対する二次形式の曲面 $z = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ は、固有値の符号に応じた形状をとる。すべての $\lambda_i > 0$ ならお椀型（正定値）、1 つの $\lambda_i = 0$ なら谷樋型（半正定値）、符号が混在していれば鞍型（不定値）になる。

グラム行列のランク条件は、その固有値にそのまま現れる。以下では、列が線形独立であるため $\mathbf{X}$ に対して $\mathbf{X}^\top \mathbf{X}$ が PD になるはずの縦長行列と、第 2 列が第 1 列の 2 倍であるため固有値 0 を持つ PSD 行列になるはずの行列を比較する。

```
X_full = np.array([[1., 0.], [1., 1.], [0., 1.], [2., 1.]])  # rank 2
X_dep = np.array([[1., 2.], [1., 2.], [0., 0.], [2., 4.]])   # col 2 = 2*col 1
for X, name in [(X_full, 'full column rank'), (X_dep, 'dependent columns')]:
    eigvals = np.linalg.eigvalsh(X.T @ X).round(4)
    eigvals[np.abs(eigvals) < 1e-9] = 0.0  # clean up tiny round-off / -0.0
    print(f'{name}: eigenvalues of X^T X = {eigvals}')
```

```
full column rank: eigenvalues of X^T X = [1.1459 7.8541]
dependent columns: eigenvalues of X^T X = [ 0. 30.]
```

フルランクの場合は両方の固有値が正になっている。一方、列が線形従属な場合は片方の固有値が 0 に落ちる。二次形式は $(2, -1)$、すなわち $\mathbf{X}$ が零ベクトルへ写すまさにその方向に沿って 0 になる。

33.3.2.3 レイリー商

恒等式 Equation 33.36 の役割は符号の判定にとどまらない。この式によって、最大固有値と最小固有値が最適化問題の解として特徴づけられる。対称行列 $\mathbf{A}$ に対し、**レイリー商**を次のように定義する。

$$R(\mathbf{x}) = \frac{\mathbf{x}^{\top}\mathbf{A}\mathbf{x}}{\mathbf{x}^{\top}\mathbf{x}}, \qquad \mathbf{x} \neq \mathbf{0}.$$

これは単位長さの二乗あたりの二次形式の値を測る量であり、$\mathbf{x}$ が固有ベクトルのときには $R(\mathbf{x})$ はその固有値に一致する。本項全体を通して、固有値は絶対値ではなく符号付きの値によって $\lambda_1 \geq \cdots \geq \lambda_n$ の順に並べる。

命題（Rayleigh）。 固有値（A は対称行列）を $\lambda_1 \geq \cdots \geq \lambda_n$ とするとき、

$$\lambda_1 = \max_{\mathbf{x}\neq\mathbf{0}} R(\mathbf{x}), \qquad \lambda_n = \min_{\mathbf{x}\neq\mathbf{0}} R(\mathbf{x}),$$

が成り立ち、最大値と最小値はそれぞれ対応する固有ベクトルにおいて達成される。

証明。 $\mathbf{x}$ を定数倍しても $R(\mathbf{x})$ は変わらないため、単位ベクトルに限って考えればよい。$\mathbf{y} = \mathbf{V}^{\top}\mathbf{x}$ と置くとノルムが保たれ（$\|\mathbf{y}\| = \|\mathbf{x}\| = 1$）、Equation 33.36 は $R(\mathbf{x}) = \sum_i \lambda_i y_i^2$ となる。これは非負の重み y_i^2（総和は 1）による固有値の重み付き平均である。いかなる平均も最小値と最大値の間に収まるため、$\lambda_n \leq R(\mathbf{x}) \leq \lambda_1$ が成り立つ。$\mathbf{x} = \mathbf{v}_1$ または $\mathbf{x} = \mathbf{v}_n$ としてすべての重みを単一の座標に集中させれば、上下の限界がそれぞれ達成される。$\blacksquare$

通しで用いている対称行列 $\mathbf{S}$ に当てはめると、$R(\mathbf{x})$ は $[1, 3]$ の範囲を動き、45° の固有ベクトル方向で最大となり、135° の固有ベクトル方向で最小となる。その中間のあらゆる方向では、固有値 1 と 3 が固有座標の二乗に比例して混ぜ合わされる。

部分空間全体にわたって最大化すれば、中間のすべての固有値も復元できる。**クーラン・フィッシャーの定理**（Courant–Fischer theorem）によれば、$\lambda_j = \max_{\dim U = j} \min_{\mathbf{0}\neq\mathbf{x}\in U} R(\mathbf{x})$ は、すべての j 次元部分空間 U にわたって保証されるレイリー商の最大値を与える（演習問題 4d で証明の手順を追う）。

最大・最小固有値の 2 つの応用は、本書全体を通して繰り返し登場する。第 1 に、任意の行列 $\mathbf{A}$ に対し、$\mathbf{A}^{\top}\mathbf{A}$ のレイリー商の最大値は $\max_{\mathbf{x}\neq\mathbf{0}} \|\mathbf{A}\mathbf{x}\|^2/\|\mathbf{x}\|^2$ であり、これは $\mathbf{A}$ がベクトルを引き伸ばし得る最大倍率の二乗に等しい。第 2 に、凸二次関数 $f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^{\top}\mathbf{A}\mathbf{x}$（$\mathbf{A} \succ 0$）に対する勾配降下法を考えよう。その更新式は $\mathbf{x} \leftarrow (\mathbf{I} - \eta\mathbf{A})\mathbf{x}$ である。Equation 33.28 により、各固有座標にはステップごとに $1 - \eta\lambda_i$ が掛かるため、任意の初期点から反復が収束するための必要十分条件は、$|1 - \eta\lambda_i| < 1$ がすべての i で成り立つこと、すなわち $0 < \eta < 2/\lambda_{\max}$ となる。このように最大固有値はステップサイズの上限を定め（これは滑らかさの定数 L として 35.1 節に登場する）、ステップサイズが $\eta \leq 1/\lambda_{\max}$ の範囲にあるとき、最小固有値が最も遅い縮小率 $1 - \eta\lambda_{\min}$ を決定する。両者の比 $\lambda_{\max}/\lambda_{\min}$ は $\mathbf{A}$ の条件数（condition number）であり、33.5.3.2 節で任意の行列へと一般化する。

33.3.2.4 標本共分散とマルチェンコ・パスツール則

高次元データの中に構造を探すときには、標本共分散行列の大きな固有値を調べる。しかし、どの固有値が本当に大きいのかを判断するには、まずデータにまったく構造がないときに固有値がどう見えるかという基準が必要である。いま n 個の純粋なノイズのサンプルを d 次元空間から抽出し、各座標は独立で平均 0、分散 1（したがって真の共分散は $\mathbf{I}$）であるとして、それらを行として並べた行列を $\mathbf{X} \in \mathbb{R}^{n\times d}$ としよう。標本共分散 $\mathbf{C} = \frac{1}{n}\mathbf{X}^{\top}\mathbf{X}$ の固有値は、1 の近くに集まるだろうか。任意の単位ベクトル $\mathbf{w}$ に対して、レイリー商 $\mathbf{w}^{\top}\mathbf{C}\mathbf{w}$ はデータを $\mathbf{w}$ 上へ射影したときの標本分散になる。したがって $\mathbf{C}$ の最大固有値と最小固有値は、分散が最も大きい方向と最も小さい方向に沿った標本分散そのものであり、主成分分析（principal component analysis）（33.5.2.2 節）がデータの構造として解釈するまさにその量である。

驚くべきことに、固有値は 1 の近くに集まらない。$d, n \to \infty$ の極限を、**アスペクト比** $c = d/n \le 1$ を固定したまま考えよう。すると $\mathbf{C}$ の固有値のヒストグラムは、次の区間を台とする決定論的な密度関数へ収束する。

$$\left[(1-\sqrt{c})^2,\ (1+\sqrt{c})^2\right],$$

この区間は**マルチェンコ・パスツールのバルク**（Marchenko–Pastur bulk）として知られており (Marchenko and Pastur 1967)、ノイズが有限の 4 次モーメントを持つ限り、最大・最小固有値はちょうどその両端へ収束する。$c > 1$ のときは行列 $\mathbf{C}$ のランクが高々 $n < d$ となるため、このバルクに加えて少なくとも $d - n$ 個の固有値が恒等的に 0 になる。1 次元あたり 4 サンプル ($c = 1/4$) ある場合でさえ、純粋なノイズの固有値は $[0.25, 2.25]$ 全体に大きく広がる。ランダムな揺らぎが最も大きく出た方向の標本分散は真の分散の 2 倍を超え、最も小さい方向では 4 分の 1 にしかならない。$n = 4000$、$d = 1000$ の規模になれば、この漸近極限はすでに驚くほど正確である。実際に確かめてみよう。

```
n, d = 4000, 1000                       # aspect ratio c = d / n = 1/4
X = np.random.randn(n, d)               # pure noise: true covariance I
ev = np.linalg.eigvalsh(X.T @ X / n)
c = d / n
print(f'eigenvalues of C: min {ev.min():.3f}, max {ev.max():.3f}')
print(f'Marchenko-Pastur edges: {(1 - c ** 0.5) ** 2:.3f}, '
      f'{(1 + c ** 0.5) ** 2:.3f}')
```

```
eigenvalues of C: min 0.247, max 2.238
Marchenko-Pastur edges: 0.250, 2.250
```

何度実行しても、最小固有値と最大固有値は予測された両端 0.25 と 2.25 から 100 分の数以内の範囲に収まる。このバルクの内部に落ちる固有値はどれも純粋なノイズと区別できず、上端を明確に超える固有値だけが真の構造の証拠となる。固有値がバルクを抜け出すにはどれほど強い信号が必要かについては、演習問題 5 で調べる。

本節全体を通して、2 つのことを当然の前提としてきた。行列の固有値を実際に計算できること、そして固有値を見れば、多数回のステップ後だけでなく数ステップ後のべき乗の振る舞いもわかるということである。大規模な行列や、対称からほど遠い行列に対しては、どちらの前提も安全ではない。33.4 節ではこの両方を取り上げる。

33.3.3 まとめ

固有ベクトル座標のもとでは、対角化可能な行列はスカラーの並びとして作用し、$\mathbf{A}^k\mathbf{x} = \sum_i c_i \lambda_i^k \mathbf{v}_i$ によってマルコフ連鎖、深層線形ネットワーク、および二次関数上の勾配降下法における反復適用を明快に説明する。$\mathbb{C}$ 上で空間全体を張る固有基底が存在するための必要十分条件は、すべての固有値について幾何学的重複度が代数的重複度と一致することである。一致しない場合にはジョルダンブロックが k の多項式因子をもたらして過渡的増大を引き起こし得るほか、複素共役対は不変平面内での回転とスケーリングに対応する。実対称行列はこの両方の落とし穴を免れており、常に実固有値と正規直交固有基底を持つ。その二次形式は重み付き二乗和となるため、固有値の符号が正定値性を決定し、最大・最小固有値はレイリー商の最大値・最小値と一致する。最後に、純粋なノイズから計算した標本共分散の固有値はマルチェンコ・パスツール区間全体にわたって広がるため、バルクの上端を超える固有値だけが真の構造を示す信号となる。

33.3.4 演習問題

1. **トレース、行列式、定値性。** 恒等式 Equation 33.31 と Equation 33.32 は、和と積を通じて 2×2 行列の 2 つの固有値を定める。

   1. 特性多項式を作らずにトレースと行列式だけから $\left(\begin{smallmatrix} 4 & 1 \\ 1 & 4 \end{smallmatrix}\right)$ の固有値を求め、続いて固有ベクトルを求めよ。サーファーの行列 $\mathbf{P}$（トレースは $5/4$、行列式は $1/4$）についても同様に行え。
   2. 2×2 の対称行列が正定値であるための必要十分条件は、そのトレースと行列式がともに正であることを示せ。半正定値性を特徴づけるのは、トレースと行列式に対するどのような符号条件か。
   3. 行列 $\left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$、$\left(\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right)$、$\left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$ はそれぞれ、二次関数 $f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^{\top}\mathbf{A}\mathbf{x}$ の臨界点 $\mathbf{0}$ におけるヘッセ行列である。(b) を用いて、各 f について原点が狭義の最小点、狭義でない最小点、鞍点のいずれであるかを判定せよ。鞍点となるものについては、f が減少する方向を 1 つ挙げよ。
   4. コレスキー因子 $\mathbf{L}$（対象となる行列は $\left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$）の対角成分が $\sqrt{a}$ と $\sqrt{(ac - b^2)/a}$ になることを示せ。正の対角成分を持つ因子が存在する条件が (b) の判定条件と一致するのはなぜか。

2. **ランクと固有値 0。** 本節では、対角化可能な行列のランクが重複度込みの非ゼロ固有値の個数に等しいことを示した。

   1. $\mathbf{N} = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ のランクと固有値を求めよ。対角化可能な行列に対する議論のどの段階が $\mathbf{N}$ では破綻するか。
   2. g_0 を固有値 0（対象は $n \times n$ 行列 $\mathbf{A}$）の幾何学的重複度とし、ただし $g_0 = 0$（0 が固有値でない場合）とする。任意の正方行列に対して $\operatorname{rank}\mathbf{A} = n - g_0$ が成り立つことを示せ。さらにこれから、ランクは代数的重複度込みで数えた非ゼロ固有値の個数以上であり、等号が成り立つための必要十分条件は固有値 0 が欠陥固有値でないことであると結論せよ。
   3. 行列の**特異値**は $\sigma_i = \sqrt{\lambda_i(\mathbf{A}^{\top}\mathbf{A})}$、すなわち PSD 行列 $\mathbf{A}^{\top}\mathbf{A}$ の固有値の平方根で定義される。$\mathbf{N}$、および 33.3.1.5 節のジョルダンブロック $\mathbf{J}_m(0)$ の特異値を求めよ。非ゼロ特異値の個数を、ランクおよび非ゼロ固有値の個数と比較せよ。

3. **中心化行列。** $\mathbf{1} \in \mathbb{R}^n$ をすべての成分が 1 のベクトルとし、$\mathbf{H} = \mathbf{I} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^{\top}$ と置く。

   1. $n\mathbf{I} - \mathbf{1}\mathbf{1}^{\top}$ に対し、$\mathbf{x}^{\top}(n\mathbf{I} - \mathbf{1}\mathbf{1}^{\top})\mathbf{x}$ を差 $x_i - x_j$ の二乗和として表すことで、この行列が半正定値であることを示せ。続いて $n\mathbf{I} - \mathbf{1}\mathbf{1}^{\top}$ のすべての固有値と対応する固有ベクトルを求めよ。
   2. $\mathbf{H}$ が対称行列であり $\mathbf{H}^2 = \mathbf{H}$ を満たすことを示し、(a) からその固有値とランクを導け。
   3. $\mathbf{X} \in \mathbb{R}^{n \times d}$ の各行を n 個のサンプルとする。すべての行から標本平均を引くと $\mathbf{H}\mathbf{X}$ となるため、標本共分散は $\mathbf{C} = \tfrac{1}{n}\mathbf{X}^{\top}\mathbf{H}\mathbf{X}$ と書ける。$\mathbf{C} \succeq 0$ かつ $\operatorname{rank}\mathbf{C} \le n - 1$ を示せ。512 次元の埋め込みについて、256 個の例からなるミニバッチから計算した標本共分散は、少なくとも何個の固有値 0 を持つか。これは 33.3.2.4 節における $c = d/n > 1$ の場合にあたる。この性質により、$\mathbf{C}$ に対するどのような演算が不可能になるか。

   Gilbert Strang の MIT 18.065 問題集（第 13 回講義の問題 4）を改変。

4. [code] **計算された固有値の精度。**●`np.linalg.eigvals` の背後にあるアルゴリズムは後退安定である。すなわち、ある摂動行列を伴う $\mathbf{A} + \mathbf{E}$（ただし $\|\mathbf{E}\|_2$ は $\epsilon_{\text{mach}}\|\mathbf{A}\|_2$ と同程度の大きさであり、倍精度では $\epsilon_{\text{mach}} \approx 2.2 \times 10^{-16}$）の厳密な固有値を返す (Golub and Van Loan 1996)。微小な摂動によって、欠陥固有値は ε の分数べき乗だけ動き得る。

   1. $\mathbf{e}_1, \dots, \mathbf{e}_m$ を標準基底ベクトルとする。$\mathbf{J}_m(\lambda) + \varepsilon\,\mathbf{e}_m\mathbf{e}_1^{\top}$（ジョルダンブロックの左

下の隅に ε を置いた行列）の固有値が $\lambda + \varepsilon^{1/m}\omega$（ただし ω は 1 の m 乗根全体を動く）となることを示せ。

2. $m = 1, \dots, 4$ に対し、$\mathbf{Q}\mathbf{J}_m(2)\mathbf{Q}^{\top}$ を構成せよ（ただし直交行列 $\mathbf{Q}$ はガウスランダム行列の QR 分解からとる）。これを 20 回試行し、$\max_i |\hat{\lambda}_i - 2|$（$\hat{\lambda}_i$ は数値計算された固有値）の中央値を記録して $\epsilon_{\text{mach}}^{1/m}$ と比較せよ。対称行列 $\mathbf{Q}(2\mathbf{I})\mathbf{Q}^{\top}$ についても同じ実験を行え。
3. $\mathbf{A}$ と $\mathbf{E}$ を対称行列とする。33.3.2.3 節の Rayleigh の命題を用いて、$\lambda_{\max}(\mathbf{A}+\mathbf{E}) \le \lambda_{\max}(\mathbf{A}) + \|\mathbf{E}\|_2$ および $\lambda_{\min}(\mathbf{A}+\mathbf{E}) \ge \lambda_{\min}(\mathbf{A}) - \|\mathbf{E}\|_2$ を示せ。これによって (b) の対称行列の結果はどのように説明されるか。
4. クーラン・フィッシャーの定理 $\lambda_j = \max_{\dim U = j} \min_{\mathbf{0} \neq \mathbf{x} \in U} R(\mathbf{x})$ を、対称行列 $\mathbf{A}$（固有値は $\lambda_1 \ge \cdots \ge \lambda_n$）に対して証明せよ。上界については U と $\operatorname{span}(\mathbf{v}_j, \dots, \mathbf{v}_n)$（次元は $n - j + 1$ であるため、共通部分は少なくとも 1 本の直線を含む）の共通部分をとれ。下界については $U = \operatorname{span}(\mathbf{v}_1, \dots, \mathbf{v}_j)$ とせよ。さらにこれから、(c) をすべての固有値へ拡張するワイルの不等式 $|\lambda_j(\mathbf{A}+\mathbf{E}) - \lambda_j(\mathbf{A})| \le \|\mathbf{E}\|_2$（任意の j）を導け。

5. [code] **バルクを超えるスパイク。**●33.3.2.4 節では、マルチェンコ・パスツールのバルクを純粋なノイズの固有値が広がる範囲として扱った。いまデータに真の方向が 1 つ含まれているとしよう。すなわち n 個のサンプルは独立なガウス座標を持ち、第 1 座標を除くすべての座標の分散が 1 で、第 1 座標の分散だけが $1 + \beta$ であるとする。$n \to \infty$ とし $c = d/n$ を固定した極限において、Baik and Silverstein (2006) は $\mathbf{C} = \frac{1}{n}\mathbf{X}^{\top}\mathbf{X}$ の最大固有値が次へ収束することを示した。

$$\lambda_{\max} \to \begin{cases} (1+\sqrt{c})^2, & \beta \le \sqrt{c}, \\ (1+\beta)\left(1 + c/\beta\right), & \beta > \sqrt{c}. \end{cases}$$

1. $n = 2000$、$d = 500$ とし、$\mathbf{C}$ の最大固有値を $\beta \in \{0, 0.3, 0.5, 0.7, 1, 2\}$ のそれぞれについて 5 回の試行の平均として求め、理論予測と比較せよ。
2. $d = 500$、$\beta = 0.4$ に固定する。上の公式を用いて、最大固有値がバルクを抜け出すサンプル数 n の閾値を求めよ。次にサンプル数を $n \in \{500, 1000, 2000, 4000, 8000\}$ と増やしながら、最大固有値をバルクの上端、理論予測、および $\beta = 0$（同じ n）に対する最大固有値と比較せよ。$n = 4000$ では分離が見えにくいのはなぜか。
3. 実務者が、固有値がバルクの上端を超える主成分だけを残すとする。$c = 1/4$ においてある方向が残されるには、その方向の分散はどれほど大きくなければならないか。また、他の方向より分散が 10% 大きい方向を検出するには、1 次元あたり何サンプル必要か。

33.4 実践におけるスペクトル

33.3 節で扱った行列は手計算で分解できるほど小さく、固有値も期待どおりの働きをした。ランダムサーファーの誤差は、第 2 固有値が予測したとおりクリックごとに $1/4$ ずつ縮小したのである。しかし教科書の例を一歩出ると、この 2 つの好都合な前提は崩れ去る。ウェブのリンク行列は何十億もの行を持つため、特性多項式（characteristic polynomial）を書き下したり完全な固有値分解（eigendecomposition）を計算したりすることはない。また、たとえ固有値がわかっていたとしても、見た目ほど多くのことを保証してくれるわけではない。固有値がともに 0.99 であるジョルダンブロックは、減衰に転じる前にノルムが 37 まで増大したし（33.3.1.5 節）、リカレントネットワークにおける誤差逆伝播法（backpropagation）では、1 つの行列を 50 回掛けるのではなく、勾配に 50 個の**異なる**行列を次々と掛けることになる。

本節では、この 2 つの問題を順に取り上げる。第 1 に、分解するには大きすぎる行列の固有値について、何がわかるだろうか。実は驚くほど多くのことがわかる。行列の成分を見るだけで全固有値を複素平面上のいくつかの円盤内に絞り込めるほか、大規模になっても低コストな唯一の演算である行列・ベクトル積を繰り返せば、最大固有値とその固有ベクトルが求まる。PageRank はもともとこのようにして計算されていた。第 2 に、有限回のステップや、変化していく行列の積について、固有値は何を語り残しているだろうか。こちらの答えは厳しい現実を突きつける。固有値が支配するのは単一の行列のべき乗における長期的な振る舞いだけであり、それ以外を扱うには行列ノルム、そして最終的には 33.5 節の特異値（singular value）が必要になる。

以下で扱うすべての話題が深層学習において同じ重要度を持つわけではないため、どこに重点を置くべきかを知っておくとよい。本書全体を通じて繰り返し現れる考え方が 3 つある。べき乗法（power iteration、33.4.1.2 節）は層の最大の引き伸ばし率を推定する手法であり、スペクトル正規化（spectral normalization、33.5.4.3 節）の中核を担う。過渡的増大（transient growth）とヤコビ行列（Jacobian matrix）の積（33.4.2.3 節、33.4.2.5 節）は、8.5 節の勾配爆発と勾配消失（vanishing gradient）を裏づける数学を与える。そして 33.4.2.6 節のランダム行列は、4.4 節における初期化時の重み行列をモデル化する。Gershgorin の定理と PageRank は古典的な名結果だが深層学習に登場する頻度は低いため、急ぐ読者は流し読みしても構わない。擬スペクトル（pseudospectrum、33.4.2.4 節）も専門的な診断道具であり、初読時には同様に飛ばしてよい。以下の数値例はすべて NumPy を用いる。

```
import numpy as np
```

33.4.1 固有値の位置特定と計算

33.4.1.1 Gershgorin の円盤

固有値を一切計算することなく、行列の固有値が大まかにどこにあるかを知ることはできるだろうか。対角行列の場合、固有値は単に対角成分そのものである。非対角成分が小さければ、固有値も対角成分の近くにとどまるだろうと直観的には予想できる。次の定理はこの直観を厳密な形で定式化するものであり、その証明はわずか 3 行で完結する。

定理（Gershgorin (Gershgorin 1931)） **$A = (a_{ij})$ を任意の $n \times n$ 行列とし、各行 i に対して非対角成分の絶対値の和を $r_i = \sum_{j \neq i} |a_{ij}|$ とおく。$\mathcal{D}_i$ を、複素平面上で中心 a_{ii}、半径 r_i を持つ閉円盤とする。このとき、A のすべての固有値は和集合 $\bigcup_i \mathcal{D}_i$ に含まれる。**

証明： $A\mathbf{v} = \lambda \mathbf{v}$ かつ $\mathbf{v} \neq \mathbf{0}$ とし、添字 i として $\mathbf{v}$ の成分の絶対値が最大になるものを選ぶ。すると $|v_i| \geq |v_j|$ がすべての j について成り立ち、かつ $v_i \neq 0$ である。第 i 行を $A\mathbf{v} = \lambda \mathbf{v}$ から書き出すと、次を得る。

$$\sum_{j} a_{ij} v_j = \lambda v_i \quad \Longrightarrow \quad (\lambda - a_{ii})\, v_i = \sum_{j \neq i} a_{ij} v_j .$$

両辺を v_i で割って絶対値をとり、$|v_j / v_i| \leq 1$ がすべての j について成り立つことを用いると、次のようになる。

$$|\lambda - a_{ii}| = \left| \sum_{j \neq i} a_{ij} \, \frac{v_j}{v_i} \right| \leq \sum_{j \neq i} |a_{ij}| = r_i ,$$

これより $\lambda \in \mathcal{D}_i$ が従う。■

この証明が実際に示していることに注目しよう。各固有値は、対応する固有ベクトルの成分が最大となる行の円盤に収まるのである。どの行がそうなるかは事前にはわからないため、定理が保証するのは、すべての固有値が n 個の円盤の和集合のどこかに含まれるということだけである。具体例で確認するため、次の非対称行列を考えよう。

$$\mathbf{A} = \begin{pmatrix} 4 & 1 & 0 \\ -1 & 4 & 0.5 \\ 0.5 & 0 & 1 \end{pmatrix}. \tag{33.37}$$

3 つの円盤の中心は 4、4、1 であり、半径はそれぞれ 1、1.5、0.5 である。中心が 4 の 2 つの円盤は重なり合っている（実際には一方が他方の内部に含まれている）のに対し、中心が 1 の円盤は $|4-1| = 3 > 1.5 + 0.5$ であるため、それらから完全に離れている。実際の固有値は $3.99 \pm 0.96i$ と 1.03 であり、複素共役対は中心 4 の 2 つの円盤の内部に、実固有値は中心 1 の孤立した円盤の内部に収まっている。Figure 33.20 に、これらの円盤と固有値を並べて示す。

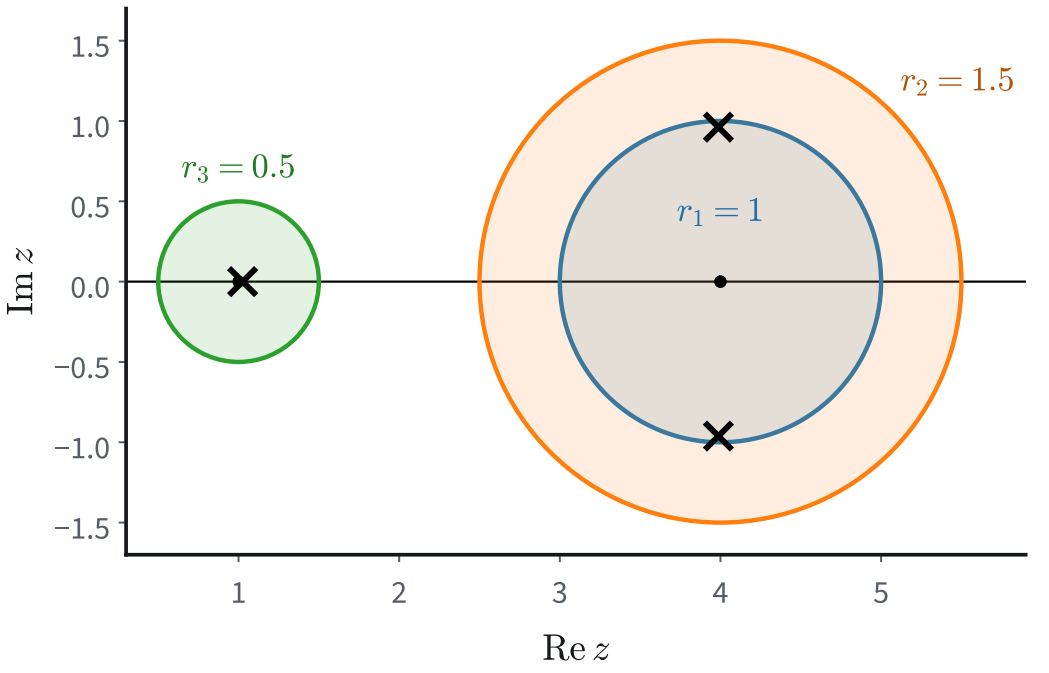


Figure 33.20: 本文中の 3×3 行列に対する Gershgorin の円盤（中心は対角成分、半径は各行の非対角成分の絶対値の和）とその固有値（十字印）。中心が 4 の重なり合う 2 つの円盤が複素共役対である 2 つの固有値を含み、中心が 1 の離れた円盤が 3 つ目の固有値を含む。

重なり合う 2 つの円盤に 2 個、孤立した円盤に 1 個の固有値が分かれて入ったのは偶然ではない。**m 個の円盤の和集合が連結であり、かつ残りのすべての円盤と交わらないならば、その領域には重複度込みでちょうど m 個の固有値が含まれる**。理由は連続性にある。非対角成分を 0 から実際の値まで連続的にスケールしていくと、固有値は対角成分から出発して連続的に移動するため、離れた円盤群のあいだにある空白を飛び越えることは決してできないからである (Horn and Johnson 2012)。中心、半径、固有値を数値計算で確かめてみよう。

```
A = np.array([[4.0, 1.0, 0.0],
              [-1.0, 4.0, 0.5],
              [0.5, 0.0, 1.0]])
centres = np.diag(A)
radii = np.abs(A).sum(axis=1) - np.abs(centres)
print('centres:', centres, '  radii:', radii)
for z in np.linalg.eigvals(A):
    print(f'eigenvalue {z:.3f}   distances to the centres: '
          f'{np.abs(z - centres).round(2)}')
```

```
centres: [4. 4. 1.]   radii: [1.  1.5 0.5]
eigenvalue 3.987+0.961j   distances to the centres: [0.96 0.96 3.14]
eigenvalue 3.987-0.961j   distances to the centres: [0.96 0.96 3.14]
eigenvalue 1.025+0.000j   distances to the centres: [2.97 2.97 0.03]
```

ここでは複素共役対が中心 4 から距離 0.96 の位置、すなわち半径 1 の内側の円盤の境界すぐ近くにあり、Gershgorin の評価がほぼタイトになりうることがわかる。

計算を一切必要としない直接の系が 1 つある。$\mathbf{A}$ の対角成分が正で**狭義対角優位**（strictly diagonally dominant）であるとき、すなわち $a_{ii} > r_i$ がすべての i について成り立つとき、すべての円盤は開右半平面の内部に完全に収まる。したがって 0 は固有値になりえず、$\mathbf{A}$ は自動的に正則（可逆）となる。さらに $\mathbf{A}$ が対称行列であれば、固有値はすべて実数かつ真に正となるため、正定値（positive definite）である。

33.4.1.2 べき乗法

Gershgorin の円盤が教えてくれるのは、固有値の大まかな位置にすぎない。では、10 億行の行列の正確な固有値を求めたい場合はどうすればよいだろうか。そのような行列を分解するのは問題外だが、疎行列であればベクトルとの積は依然として低コストに計算できる。初期ベクトル $\mathbf{v}_0$ から出発し、単に $\mathbf{A}$ を何度も繰り返し掛け続けるとどうなるだろうか。

答えを与えるのは、べき乗展開 $\mathbf{A}^k\mathbf{x} = \sum_i c_i \lambda_i^k \mathbf{v}_i$（33.3.1.3 節）である。$\mathbf{A}$ が対角化可能であり、固有ベクトル $\mathbf{v}_i$ が固有値の絶対値の大きい順に並んでいるとし、さらに最大固有値が**狭義優位**（strictly dominant）、すなわち $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$ であるとする。（$\mathbf{A}$ が実行列のとき、狭義優位性から λ_1 も実数になる。複素固有値であれば、その複素共役と同じ絶対値を共有してしまうからである。）$\mathbf{v}_0 = \sum_i c_i \mathbf{v}_i$ と展開して λ_1^k をくくり出すと、次式が得られる。

$$\mathbf{A}^k \mathbf{v}_0 = \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{v}_i = \lambda_1^k \left(c_1 \mathbf{v}_1 + \sum_{i \geq 2} c_i \left(\frac{\lambda_i}{\lambda_1} \right)^k \mathbf{v}_i \right). \tag{33.38}$$

比 $|\lambda_i / \lambda_1|$ はすべての $i \geq 2$ に対して 1 より真に小さいため、k が大きくなるにつれて後ろの項はすべて消えていく。$c_1 \neq 0$ である限り、括弧の中は $c_1 \mathbf{v}_1$ に収束する。したがって $\mathbf{A}^k \mathbf{v}_0$ の向きは主固有ベクトル $\mathbf{v}_1$ に近づき、その誤差は $|\lambda_2/\lambda_1|^k$ のオーダーで減衰する。一方、その長さはステップごとに $|\lambda_1|$ に漸近する倍率でスケールされる。成分のオーバーフローやアンダーフローを防ぐため、乗算のたびにベクトルを単位長に正規化し直す。この単純なループが**べき乗法**である (Golub and Van Loan 1996)。$\mathbf{A}$ が対称行列のときは、固有値そのものについてさらに速い収束が得られる。現在の反復ベクトルのレイリー商（Rayleigh quotient）は λ_1 に二乗のレート $|\lambda_2/\lambda_1|^{2k}$ で収束する。対称行列の固有ベクトルは互いに直交するため、大きさ δ の向きのずれが生じてもレイリー商は $O(\delta^2)$ しか変化しないからである（演習問題 2）。

具体的な 2×2 の例で、この両方の収束レートを確認できる。1.3 節では、$\mathbf{S} = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ を $\mathbf{e}_1$ から反復したベクトル列を計算し、$(2,1)$、$(5,4)$、$(14,13)$、一般に $\left((3^k+1)/2,\ (3^k-1)/2\right)$ となることを見た。単位固有ベクトル $\mathbf{v}_1 = (1,1)/\sqrt{2}$ と $\mathbf{v}_2 = (1,-1)/\sqrt{2}$ を用いれば $\mathbf{e}_1 = (\mathbf{v}_1 + \mathbf{v}_2)/\sqrt{2}$ と書けるため、次のようになる。

$$\mathbf{S}^k \mathbf{e}_1 = \frac{3^k \mathbf{v}_1 + \mathbf{v}_2}{\sqrt{2}}, \qquad \tan\angle(\mathbf{S}^k\mathbf{e}_1, \mathbf{v}_1) = 3^{-k}, \qquad 3 - R(\mathbf{S}^k\mathbf{e}_1) = \frac{2}{9^k + 1}.$$

$|\lambda_2/\lambda_1| = 1/3$ であるため、向きの誤差はステップごとに 3 分の 1 に減少する。実際、$\mathbf{v}_1$ とのなす角の正弦は 0.316、0.110、0.037（$k = 1, 2, 3$）となる。一方、レイリー商の誤差はステップごとに 9 分の 1 に減少し、0.2、0.024、0.0027 となる。Figure 33.21 は両方の曲線を対数スケールでプロットしたものである。

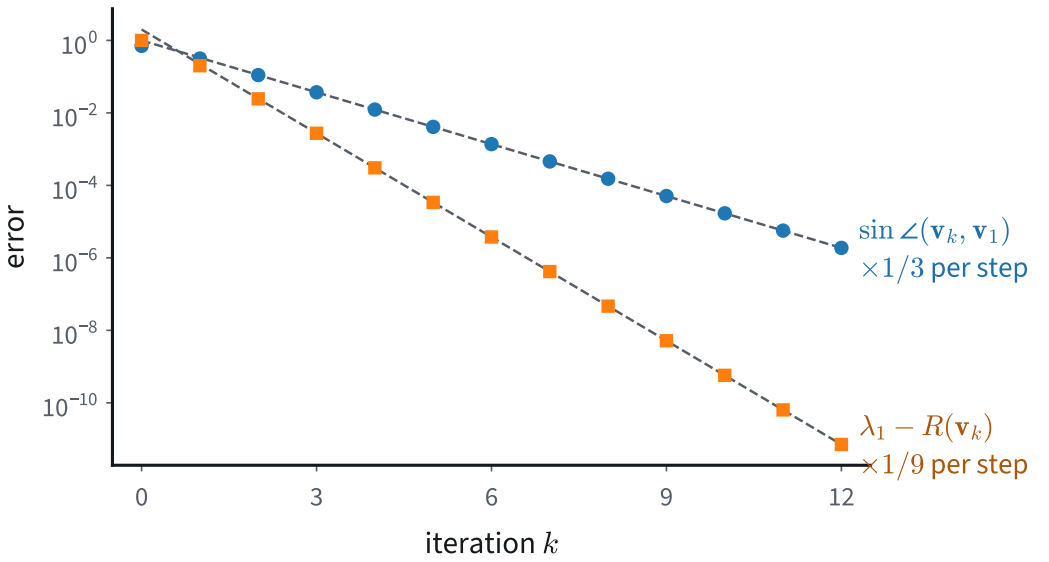


Figure 33.21: $\mathbf{S} = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ に対する $\mathbf{e}_1$ からのべき乗法。向きの誤差はステップごとに 3 分の 1 に減少するのに対し、固有値の推定誤差は直交する固有ベクトルに由来する二乗のレートにより、ステップごとに 9 分の 1 に改善する。破線は 3^{-k} と $2\cdot 9^{-k}$ を示す。

べき乗法が失敗するのはどのような場合だろうか。Equation 33.38 における 2 つの仮定が鍵を握る。第 1 に、初期ベクトルは $\mathbf{v}_1$ 方向に非ゼロの成分を持つ必要がある（$c_1 \neq 0$）。$\mathbf{v}_0$ をランダムに選べば、これは確率 1 で保証される。第 2 に、最大固有値が狭義優位でなければならない。最大の絶対値を複素共役対が共有している場合、反復ベクトルの主成分は 1 本の直線に落ち着くことなくいつまでも回転し続ける。すると連続するノルムの比は、33.3.1.6 節で扱った例題の行列とまったく同様に永遠に振動し続ける。ただしその場合でも、幾何平均 $\|\mathbf{A}^k\mathbf{v}_0\|^{1/k}$ は最大の絶対値に収束する。

すべての成分が真に正である行列では、この問題は完全に回避される。33.4.1.3 節で見るとおり、ペロン・フロベニウスの定理によって、実数かつ単根で狭義優位な固有値の存在が保証されるからである。これを確かめるため、べき乗法を 200 ステップ実行してみよう。5×5 のランダム行列の成分を $[0,1]$ から一様に抽出し、ランダムなベクトルから出発して、最終的なノルム比を真の最大固有値の絶対値と比較する。

```
A = np.random.rand(5, 5)                # positive entries
v = np.random.randn(5)
for _ in range(200):
    Av = A @ v
    ratio = np.linalg.norm(Av) / np.linalg.norm(v)
    v = Av / np.linalg.norm(Av)
mods = np.sort(np.abs(np.linalg.eigvals(A)))[::-1]
print(f'norm ratio after 200 steps = {ratio:.10f}')
print(f'largest |eigenvalue|       = {mods[0]:.10f}')
print(f'|lambda_2| / |lambda_1|    = {mods[1] / mods[0]:.3f}')
```

```
norm ratio after 200 steps = 2.5254088042
largest |eigenvalue|       = 2.5254088042
|lambda_2| / |lambda_1|    = 0.299
```

どの試行でも、ノルム比は表示されたすべての桁で最大固有値の絶対値と一致する。この速い収束がどれほど典型的なものかを見るため、ギャップ比 $|\lambda_2/\lambda_1|$ をこのような正行列 10,000 個にわたって計算してみよう。比較のため、正値性の保証がない標準ガウス分布の成分を持つ行列も 10,000 個生成し、主固有値がどのくらいの頻度で複素共役対をなすかを調べる。

```
mods = np.sort(np.abs(np.linalg.eigvals(np.random.rand(10000, 5, 5))), axis=1)
print('uniform entries, gap ratio |lambda_2| / |lambda_1|, 5% and 95% quantiles:',
      np.quantile(mods[:, -2] / mods[:, -1], [0.05, 0.95]).round(2))
lam = np.linalg.eigvals(np.random.randn(10000, 5, 5))
top = lam[np.arange(10000), np.abs(lam).argmax(axis=1)]   # a dominant eigenvalue
print('Gaussian entries, fraction with a complex dominant pair:',
      np.mean(top.imag != 0).round(2))
```

```
uniform entries, gap ratio |lambda_2| / |lambda_1|, 5% and 95% quantiles: [0.14 0.37]
Gaussian entries, fraction with a complex dominant pair: 0.43
```

一様な正の成分を持つ場合、10 回中 9 回の試行でギャップ比はおよそ 0.14 から 0.37 のあいだに収まるため、200 回の反復によって向きの誤差は浮動小数点精度（floating-point precision）をはるかに下回るまで小さくなる。対照的にガウス分布の成分を持つ場合は、10 回中約 4 回の試行で主固有値が複素共役対となり、そのような行列ではノルム比が決して収束しない。

33.4.1.3 PageRank

非負の成分を持つ行列では、主固有ベクトルこそが求めたい対象そのものである場合が多く、その計算にはべき乗法が用いられる。この固有ベクトルがいつ一意で扱いやすい性質を持つかを述べるため、有向辺 $j \to i$ を $a_{ij} > 0$ のときに張る有向グラフにおいて、すべての頂点から他のすべての頂点へ到達できるとき、その非負行列を**既約**（irreducible）と呼ぶことにする。**ペロン・フロベニウスの定理**（Perron–Frobenius theorem）(Horn and Johnson 2012) によれば、任意の既約な非負行列は全固有値の最大絶対値に等しい実数の正の固有値を持ち、この固有値は単根であり、対応する固有ベクトルはすべての成分が真に正となるように選べる。さらに行列のすべての成分が真に正であれば、この固有値は狭義優位でもあり、他のすべての固有値の絶対値はこれより真に小さくなる。

Google の初期の PageRank アルゴリズム (Brin and Page 1998) は、ウェブのランキングをまさにこのような固有ベクトル問題へと帰着させる。リンク $1 \to 2$、$2 \to 1$、$2 \to 3$ を持つ 3 ページからなる小規模なウェブを考えよう。ランダムサーファーは出力リンクの中から一様ランダムに 1 つを選んで辿る。ページ 3 には出力リンクがないため、サーファーがページ 3 から全ページの

いずれかへ等確率でジャンプするものとして、この行き止まりを補正する。補正後の遷移行列 $\mathbf{P}$ は列確率行列（column-stochastic matrix）となり、第 j 列がページ j からクリックして遷移する確率を表す。

$$\mathbf{P} = \begin{pmatrix} 0 & \tfrac{1}{2} & \tfrac{1}{3} \\ 1 & 0 & \tfrac{1}{3} \\ 0 & \tfrac{1}{2} & \tfrac{1}{3} \end{pmatrix}.$$

長期的に見てサーファーが各ページに滞在する時間の割合は、定常分布（stationary distribution）π、すなわち $\mathbf{P}\pi = \pi$ を満たすベクトルになる。これは手計算で解くことができる。第 1 行と第 3 行はいずれも $\pi_1 = \tfrac{1}{2}\pi_2 + \tfrac{1}{3}\pi_3 = \pi_3$ を与え、$\pi_1 = \pi_3$ を第 1 行に代入すると $\pi_2 = \tfrac{4}{3}\pi_3$ が得られる。成分の和が 1 になるよう正規化すると、$\pi = (0.3, 0.4, 0.3)$ となる。ページ 2 が最上位になるのは、ページ 1 がすべてのクリックをそこへ送るのに対し、ページ 2 はクリックをページ 1 とページ 3 に折半するからである。$\mathbf{P}$ の固有値は 1、$-\tfrac{2}{3}$、0 であるため、任意の初期分布からのべき乗法はステップあたり $2/3$ のレートで収束する。

実は、**すべての**列確率行列は最大絶対値を持つ固有値として 1 を持つ。第 1 に、すべての列の和が 1 であることから $\mathbf{1}^{\top}\mathbf{P} = \mathbf{1}^{\top}$ が成り立つ。これは 1 が $\mathbf{P}^{\top}$ の固有値であり、したがって $\mathbf{P}$ の固有値でもあることを意味する（$\det \mathbf{X}^{\top} = \det \mathbf{X}$ であるため、行列とその転置は同じ特性多項式を共有する）。第 2 に、Gershgorin の定理を $\mathbf{P}^{\top}$（その各行は $\mathbf{P}$ の各列である）に適用すると、すべての固有値は中心 $p_{ii} \geq 0$、半径 $1 - p_{ii}$ の円盤内に収まり、そのような円盤はすべて単位円盤の内部に含まれる。

しかし現実のウェブは 3 ページの例よりもはるかに複雑であり、リンクの巡回路や、サーファーが抜け出せない孤立したページ群を含んでいる。そのため、補正後の行列 $\mathbf{P}$ が既約でなくなったり、絶対値 1 の固有値を複数持ったりすることがある。たとえば 4 ページの単純な巡回路は固有値 ± 1 と $\pm i$ を持つため、減衰なしのべき乗法は決して収束しない（演習問題 4）。PageRank は、次の**減衰付き**（damped）行列を反復することで、これら両方の問題を一挙に解決する。

$$\mathbf{G} = \alpha \mathbf{P} + (1 - \alpha)\tfrac{1}{n}\mathbf{1}\mathbf{1}^{\top}, \qquad \alpha = 0.85,$$

この行列は、確率 0.85 でランダムなリンクを辿り、確率 0.15 で一様ランダムに選んだページへテレポートするサーファーを表す。$\mathbf{G}$ のすべての成分が真に正であるため、構成から直ちにペロン・フロベニウスの定理が適用できる。さらに都合のよいことに、Haveliwala and Kamvar (2003) は $|\lambda_2(\mathbf{G})| \leq \alpha$ を示した。つまりリンクグラフがどれほど入り組んでいても、べき乗法の誤差はステップごとに少なくとも 0.85 倍以下に縮小する。このたった 1 つの上界がウェブ規模の PageRank を実用的なものにした。グラフに何十億ページ含まれていようとも、50 回の反復で誤差は 3000 分の 1 未満にまで縮小するのである（$0.85^{50} \approx 3 \times 10^{-4}$）。

ウェブ規模では、$\mathbf{P}$ はリンクごとに 1 成分を持つ疎行列として保持され、密行列 $\mathbf{G}$ を陽に構築することは決してない。任意の確率ベクトル $\mathbf{p}$（$\mathbf{1}^{\top}\mathbf{p} = 1$）に対して、$\mathbf{G}\mathbf{p} = \alpha \mathbf{P}\mathbf{p} + (1-\alpha)\tfrac{1}{n}\mathbf{1}$ はわずか 1 回の疎行列・ベクトル積と定数シフトだけで計算できる。出力リンクを持たない行き止まりページの補正もランク 1 の項であり、まったく同様に処理される。

33.4.2 反復と安定性

33.4.2.1 スペクトル半径

べき乗法は 1 つの数、すなわち固有値の最大絶対値を浮き彫りにした。この数は**スペクトル半径**（spectral radius）と呼ばれ、$\rho(\mathbf{A}) = \max_i |\lambda_i|$ と表される。スペクトル半径は、単一の固定され

た行列のべき乗における長期的な増大率を支配する。ゲルファントの公式（Gelfand's formula）はこの事実を厳密に表しており、任意の正方行列 $\mathbf{A}$ と任意の行列ノルムに対して次が成り立つ。

$$\lim_{k \to \infty} \|\mathbf{A}^k\|^{1/k} = \rho(\mathbf{A}).$$

この主張のうち比較的容易な片側の不等式、すなわち $\rho(\mathbf{A}) \le \|\mathbf{A}^k\|^{1/k}$ がすべての $k \ge 1$ について成り立つことは、演習問題 1 で扱う。

ゲルファントの公式が教えて**くれない**のは、固定された有限のステップ k で何が起きるかであり、そのギャップが問題になる行列にはすでに 2 つ出会っている。せん断行列 $\mathbf{T} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ は $\rho(\mathbf{T}) = 1$ を満たすが、そのべき乗 $\mathbf{T}^k = \left(\begin{smallmatrix} 1 & k \\ 0 & 1 \end{smallmatrix}\right)$ は際限なく増大するため、$\rho \le 1$ であっても反復列が有界にとどまるとすら限らない。ジョルダンブロック $\mathbf{J}_2(0.99)$ は $\rho < 1$ でありいずれ 0 へ減衰するはずだが、そのべき乗は減少に転じる前にノルムが 37 まで上昇するため、$\rho < 1$ であっても大きな過渡的増大は排除できない。では、遠い極限だけでなく**すべての**ステップを固有値が制御するのは、どのような場合だろうか。

33.4.2.2 正規行列

せん断とジョルダンブロックという 2 つの反例はいずれも欠陥行列（defective matrix）であり、基底をなすのに十分な数の固有ベクトルを持たない。そこで一気に正反対の極端な場合へ進み、固有ベクトルからなる**正規直交**（orthonormal）基底を持つ行列を考えよう。そのような行列は互いに直交する方向に沿って独立に作用するため、すべてのステップにおいて固有値が振る舞いのすべてを語るはずである。

実行列は、自身の転置と可換であるとき、すなわち $\mathbf{A}^\top \mathbf{A} = \mathbf{A}\mathbf{A}^\top$ を満たすとき**正規**（normal）であるという（複素行列の場合は、転置を共役転置（conjugate transpose）$\mathbf{A}^*$ に置き換え、$\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^*$ を課す）。対称行列、直交行列、そして回転・スケーリングブロック $r\mathbf{R}_\theta$（33.3.1.6 節）はすべて正規行列である。対照的に、これまで例として用いてきた $\mathbf{A} = \left(\begin{smallmatrix} 1 & 2 \\ -1 & 3 \end{smallmatrix}\right)$ は正規ではない。実際、$\mathbf{A}\mathbf{A}^\top = \left(\begin{smallmatrix} 5 & 5 \\ 5 & 10 \end{smallmatrix}\right)$ であるのに対し、$\mathbf{A}^\top\mathbf{A} = \left(\begin{smallmatrix} 2 & -1 \\ -1 & 13 \end{smallmatrix}\right)$ となる。この非可換性こそが、反復点列が円ではなく楕円に沿って渦を巻いた理由である。

複素スペクトル定理（complex spectral theorem）(Horn and Johnson 2012) は、この行列のクラスを完全に特徴づける。すなわち、行列が正規であることと、$\mathbb{C}$ 上の正規直交固有基底を持つこと、言い換えれば $\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^*$ とユニタリ行列 $\mathbf{U}$（$\mathbf{U}^*\mathbf{U} = \mathbf{I}$）を用いて表せることは同値である。証明は 33.3.2.1 節の実スペクトル定理と同じ帰納法に従う。

1 ステップにおける最悪ケースの増幅率を測るため、$\|\mathbf{X}\|_2 = \max_{\|\mathbf{x}\|=1} \|\mathbf{X}\mathbf{x}\|$ と書いて $\mathbf{X}$ の**スペクトルノルム**（spectral norm）を表すことにする。これは $\mathbf{X}$ がベクトルを引き伸ばしうる最大の倍率である。ユニタリ行列による基底変換はユークリッド長を保つため、正規行列の任意のべき乗は次式を満たす。

$$\|\mathbf{A}^k\|_2 = \|\mathbf{U}\Lambda^k\mathbf{U}^*\|_2 = \max_i |\lambda_i|^k = \rho(\mathbf{A})^k. \tag{33.39}$$

正規行列では、漸近的にだけでなく、個々のすべてのステップ k において固有値が最悪ケースの増幅率を制御する。

$\mathbf{A}$ が対角化可能（$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$）ではあるが正規ではない場合はどうなるだろうか。このとき固有ベクトル行列 $\mathbf{V}$ はもはや直交行列ではなく、スペクトルノルムの劣乗法性（submultiplicativity）からは次の評価式（上界）しか得られない。

$$\|\mathbf{A}^k\|_2 \le \underbrace{\|\mathbf{V}\|_2\|\mathbf{V}^{-1}\|_2}_{\kappa_2(\mathbf{V})}\,\rho(\mathbf{A})^k. \tag{33.40}$$

前因子 $\kappa_2(\mathbf{V})$ は固有ベクトル行列の条件数（condition number、33.5.3.2 節）である。この値は固有ベクトルが直交するとき 1 に等しく、2 つの固有ベクトルがほぼ同じ向きを指すときいくらでも大きくなる。この前因子が許容する過渡的増大には、相殺という明快な幾何学的原因がある。固有ベクトルどうしがほぼ平行であると、通常の単位ベクトルであっても、互いにほぼ打ち消し合う巨大な固有成分の線形結合としてしか表せなくなる。そこに $\mathbf{A}$ を繰り返し作用させると、各成分が異なるレートで減衰するため微妙な打ち消し合いが崩れ、最終的に全体の指数減衰が支配的になる前に、反復ベクトルの長さが大きく膨れ上がるのである。

33.4.2.3 過渡的増大

この過渡的増大を示す最も単純な行列は、次の 2×2 上三角行列である。

$$\mathbf{M} = \begin{pmatrix} 0.9 & 4 \\ 0 & 0.8 \end{pmatrix}. \tag{33.41}$$

この行列の固有値は対角成分である 0.9 と 0.8 である。どちらも単位円の内側に十分収まっており、互いに異なるため $\mathbf{M}$ は対角化可能である。固有ベクトルを手計算で求め、単位ベクトル $\mathbf{e}_2 = (0,1)$ をそれらで展開してみよう。$\lambda = 0.9$ に対しては、$\mathbf{M} - 0.9\mathbf{I}$ の第 2 行が $-0.1 v_2 = 0$ となることから $\mathbf{v}_1 = (1,0)$ を得る。$\lambda = 0.8$ に対しては、$\mathbf{M} - 0.8\mathbf{I}$ の第 1 行が $0.1 v_1 + 4 v_2 = 0$ となることから $\mathbf{v}_2 = (-40, 1)$ を得る。これら 2 つの向きがいかに近いか注目してほしい。両者のなす角はわずか $\arctan(1/40) \approx 1.4^\circ$ しかない。このほぼ平行な基底で $\mathbf{e}_2$ を展開すると $\mathbf{e}_2 = 40\mathbf{v}_1 + \mathbf{v}_2$ となり、固有ベクトルによるべき乗展開（33.3.1.3 節）を適用すると次式が得られる。

$$\mathbf{M}^k \mathbf{e}_2 = 40\cdot 0.9^k\,\mathbf{v}_1 + 0.8^k\,\mathbf{v}_2 = \begin{pmatrix} 40\,(0.9^k - 0.8^k) \\ 0.8^k \end{pmatrix}. \tag{33.42}$$

$k = 0$ において、長さ 1 のベクトルが、長さおよそ 40 の 2 つの固有成分の差として表されている。両成分ともステップごとに減衰するが、0.8^k は 0.9^k よりも速く減少するため打ち消し合いが崩れ、和は最初かえって増大する。第 1 座標 $40(0.9^k - 0.8^k)$ が最大になるのは k に関する導関数が 0 になる点、すなわち $(9/8)^k = \ln 0.8 / \ln 0.9$ を満たす $k \approx 6.4$ であり、その高さは約 10.8 に達する。Figure 33.22 に、$k = 0$ と $k = 6$ における 2 つの成分を示す。比較として、単位固有ベクトルを用いた一般の上界 Equation 33.40 では $\kappa_2(\mathbf{V}) = 80$ となり、$\|\mathbf{M}^6\|_2 \le 80\cdot 0.9^6 \approx 42.5$ という予測を与える。この上界は確かに成り立っているものの、真のピーク値 10.8 と比べると約 4 倍控えめな評価になっている。

この計算を数値的に検証し、$\mathbf{M}$ をまったく同じ固有値を持つ正規な対角行列 $\Lambda = \mathrm{diag}(0.9, 0.8)$ と比較してみよう。

```
M = np.array([[0.9, 4.0], [0.0, 0.8]])
Lam = np.diag([0.9, 0.8])                  # same eigenvalues, normal
lam, V = np.linalg.eig(M)                  # unit eigenvectors as columns
print('eigenvalues:', lam)
print('coefficients of e_2 on the unit eigenvectors:',
      np.linalg.solve(V, [0., 1.]))
ks = np.arange(61)
```

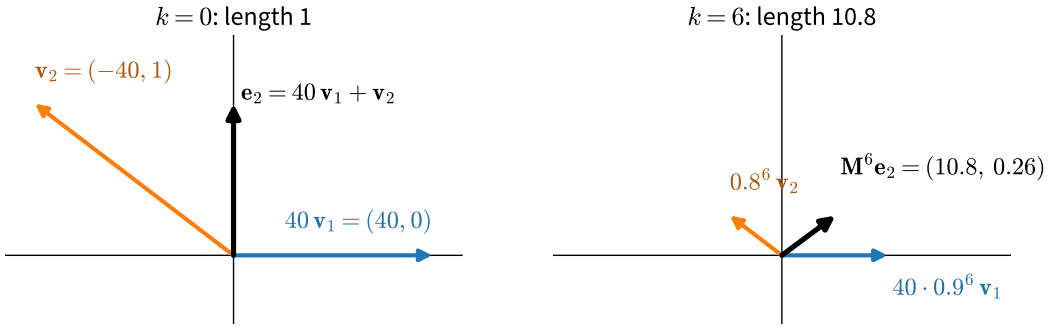


Figure 33.22: 減衰前の $\mathbf{M}^k\mathbf{e}_2$ の増大。左：$k = 0$ では、単位ベクトル $\mathbf{e}_2$ はほぼ打ち消し合う長さ約 40 の 2 つの固有成分の差として表される。右：6 ステップ後には 2 つの成分がそれぞれ 0.9^6 倍と 0.8^6 倍に減衰して相殺が崩れ、元の 10 倍を超える長さのベクトルが生じる。縦軸は約 30 倍に引き伸ばして描かれており、等倍で描けば $\mathbf{v}_1$ と $\mathbf{v}_2$ のなす角はわずか 1.4° しかない。

```
norm_M = [np.linalg.norm(np.linalg.matrix_power(M, k), 2) for k in ks]
norm_L = [np.linalg.norm(np.linalg.matrix_power(Lam, k), 2) for k in ks]
k_peak = int(np.argmax(norm_M))
print(f'max ||M^k|| = {norm_M[k_peak]:.3f} at k = {k_peak};  '
      f'max ||Lambda^k|| = {max(norm_L):.3f}')
print(f'kappa_2(V) = {np.linalg.cond(V):.1f}')
```

```
eigenvalues: [0.9 0.8]
coefficients of e_2 on the unit eigenvectors: [40.         40.01249805]
max ||M^k|| = 10.788 at k = 6;  max ||Lambda^k|| = 1.000
kappa_2(V) = 80.0
```

単位固有ベクトルで展開すると、$\mathbf{e}_2$ は絶対値が約 40 の 2 つの係数を持ち、これは $40\mathbf{v}_1$ と $\mathbf{v}_2$ の長さに一致する。スペクトルノルム $\|\mathbf{M}^k\|_2$ は最大 10.8（$k = 6$）まで上昇するのに対し、$\|\Lambda^k\|_2$ は初期値である 1 を決して超えない。両行列は同じ固有値を共有しているにもかかわらず、正規行列が最初のステップから縮小し続ける一方、非正規行列は減衰する前に 10 倍以上に増大する。Figure 33.23 の左パネルは両者の軌跡を対数スケールで描いたものであり、ピークを過ぎた後はゲルファントの公式が保証するとおり、どちらの曲線もやがて同じ傾き $\log 0.9$ で下降していく。

33.4.2.4 擬スペクトル

本項では新たな現象ではなく非正規性の診断手法を導入するため、初読時には飛ばしても構わない。過渡的増大と並ぶ非正規性のもう 1 つの兆候は、微小な摂動に対する固有値の極端な敏感さである。この敏感さを可視化するため、ε **擬スペクトル**は、$\mathbf{A} + \mathbf{E}$ の固有値となるすべての複素数（ただし $\mathbf{E}$ はスペクトルノルムが高々 ε のある摂動）を集めた集合として定義される。都合のよいことに、同じ集合を**レゾルベント** $(z\mathbf{I} - \mathbf{A})^{-1}$ から直接計算できる。

$$\begin{aligned}\Lambda_\varepsilon(\mathbf{A}) &= \{z \in \mathbb{C} : z \text{ is an eigenvalue of } \mathbf{A} + \mathbf{E} \text{ for some } \|\mathbf{E}\|_2 \le \varepsilon\} \\ &= \{z \in \mathbb{C} : \|(z\mathbf{I} - \mathbf{A})^{-1}\|_2 \ge 1/\varepsilon\},\end{aligned} \tag{33.43}$$

ここで、$\mathbf{A}$ 自身の固有値においてはレゾルベントノルムを無限大と約束する。これら 2 つの定義が同値であることの証明は、演習問題 3 とする。正規行列の場合、点 z におけるレゾルベントノ

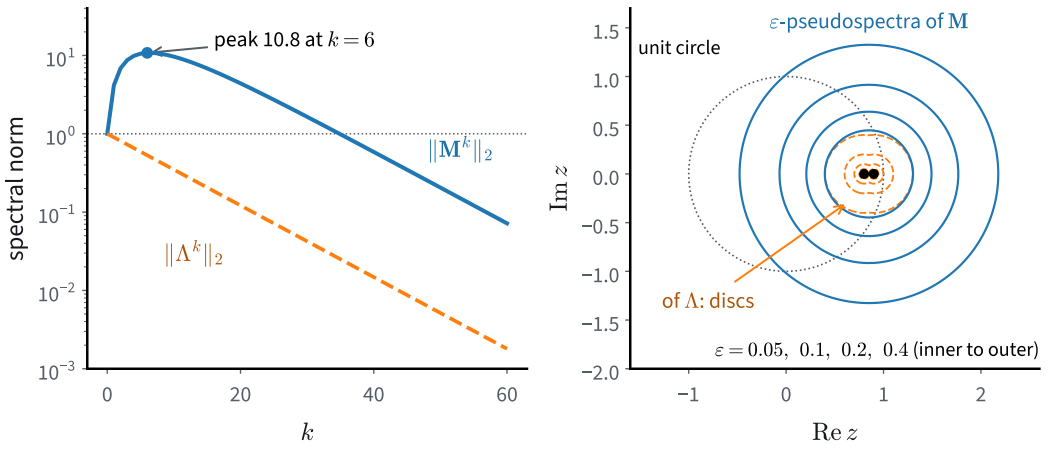


Figure 33.23: 左：同じ固有値 0.9 と 0.8 を持つ $\|\mathbf{M}^k\|_2$ と $\|\Lambda^k\|_2$。$\mathbf{M}$ は 10.8（$k = 6$）まで上昇した後、両曲線とも傾き $\log 0.9$ で減衰する。右：ε 擬スペクトル（$\varepsilon = 0.05, 0.1, 0.2, 0.4$）。正規行列 Λ の周りでは単に半径 ε の円盤になるのに対し、$\mathbf{M}$ の周りでは単位円を大きく越えて張り出す。

ルムは単に z から最も近い固有値までの距離の逆数になるため、$\Lambda_\varepsilon(\mathbf{A})$ は各固有値を中心とする半径 ε の円盤の和集合にすぎない。しかし $\mathbf{M}$ のような非正規行列では、単位円上の $z = 1$ で確かめられるように、レゾルベントノルムが桁違いに大きくなりうる。

```
I = np.eye(2)
print(f'||(I - M)^-1||      = {np.linalg.norm(np.linalg.inv(I - M), 2):.1f}')
print(f'||(I - Lambda)^-1|| = {np.linalg.norm(np.linalg.inv(I - Lam), 2):.1f}')
```

```
||(I - M)^-1||      = 200.3
||(I - Lambda)^-1|| = 10.0
```

Λ に対するレゾルベントノルムは 10 であり、これは距離 0.1（点 1 から固有値 0.9 までの距離）の逆数である。一方、$\mathbf{M}$ に対しては 200.3 まで跳ね上がる。つまり大きさ $1/200.3 \approx 0.005$ のごく小さな摂動だけで $\mathbf{M}$ の固有値は単位円上まで押し出されてしまうのに対し、Λ ではその 20 倍にあたる大きさ 0.1 の摂動が必要になる（Figure 33.23 右パネル）。浮動小数点の丸め誤差は行列成分への微小な摂動として働くため、非正規性の強い行列では数値計算された固有値が驚くほど不正確になることがある。33.6 節では、Kreiss の行列定理を通じてこの話題をさらに掘り下げる。

33.4.2.5 ヤコビ行列の積

ここまでは、ベクトルに**同じ**行列を繰り返し掛けてきた。しかしリカレントニューラルネットワーク（8.5 節）では、その前提さえ崩れる。リカレント層は隠れ状態を $\mathbf{h}_t = \phi(\mathbf{W}\mathbf{h}_{t-1} + \cdots)$ と更新し、各ステップで同じ重み行列 $\mathbf{W}$ を用いるが、通時的誤差逆伝播法（backpropagation through time、BPTT）は勾配にステップごとのヤコビ行列の系列 $\mathbf{D}_t = \partial \mathbf{h}_t / \partial \mathbf{h}_{t-1}$ を掛けていく。

$$\nabla_{\mathbf{h}_0} \mathcal{L} = \mathbf{D}_1^\top \mathbf{D}_2^\top \cdots \mathbf{D}_T^\top \nabla_{\mathbf{h}_T} \mathcal{L}.$$

非線形性 ϕ はタイムステップごとに異なる隠れ状態で評価されるため、ヤコビ行列 $\mathbf{D}_1, \ldots, \mathbf{D}_T$

は互いに異なる行列となり、単一の行列のべき乗に関する理論はもはや適用できない。

実際、個々の因子のスペクトル半径からは、それらの積について何ひとつ知ることはできない。次の 2 つの 2×2 行列を考えてみよう。

$$\mathbf{D}_1 = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{D}_2 = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}, \qquad \mathbf{D}_2\mathbf{D}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix}.$$

$\mathbf{D}_1$ と $\mathbf{D}_2$ はどちらも $\rho = 0$ のべき零（nilpotent）行列であり、それぞれのべき乗は 2 ステップ目にはゼロ行列になる。ところが両者の積は $\rho = 4$ を持つ。$\mathbf{D}_1$ が第 2 座標を 2 倍して第 1 座標にコピーし、$\mathbf{D}_2$ がそれをさらに 2 倍して元の座標へコピーし戻すからである。

したがって、反復される線形変換を調べる際には、3 つの異なる量を明確に区別しておくとよい。**スペクトル半径**は、1 つの固定された行列のべき乗における漸近的な増大率を与える。**スペクトルノルム**は、劣乗法性により $\|\mathbf{D}_T \cdots \mathbf{D}_1\|_2 \le \prod_t \|\mathbf{D}_t\|_2$（上の例では $4 \le 2 \cdot 2$）が成り立つため、変化する行列の任意の有限積を上から抑える。最後に、**最大リアプノフ指数**（top Lyapunov exponent）は次の式で定義される。

$$\mu = \lim_{T \to \infty} \frac{1}{T} \log \|\mathbf{D}_T \cdots \mathbf{D}_1 \mathbf{x}_0\|$$

この量は、変化する行列の長い積に沿った 1 ステップあたりの平均対数増大率を測るものである。各因子が独立同分布のランダム行列であるとき、緩やかな条件のもとでこの極限は確率 1 で存在する (Furstenberg and Kesten 1960)。

勾配が $\mu > 0$ であるヤコビ行列の系列を逆方向に伝播すると指数関数的に増大して爆発し、$\mu < 0$ であれば指数関数的に縮小して消失する (Bengio et al. 1994; Pascanu et al. 2013)。深層学習における標準的な対策が、固有値ではなくノルムやヤコビ行列そのものを対象としていることに注目しよう。直交初期化（orthogonal initialization）(Saxe et al. 2014) は重み行列を直交行列に選ぶことで、線形漸化式がすべてのステップであらゆるベクトルのユークリッド長を保つようにする。これは $\rho = 1$ よりもはるかに強い保証である。勾配クリッピングは、誤差逆伝播法の直後に勾配のノルムを直接頭打ちにする。LSTM や GRU のようなゲート付きアーキテクチャはヤコビ行列自体を変化させ、選択された方向が 1 に近い倍率で通過できる経路を作り出す（8.5.1.1 節）。

33.4.2.6 初期化時のランダム行列

初期化したばかりのニューラルネットワークのランダムな重み行列において、スペクトル半径とスペクトルノルムはどのくらいの大きさになるだろうか。独立な標準ガウス分布に従う成分を持つ $n \times n$ 行列 $\mathbf{A}$ を考えよう。これは実 **Ginibre アンサンブル**として知られるものである (Ginibre 1965)。$n \to \infty$ のとき、$1/\sqrt{n}$ でリスケールした固有値は複素平面上の単位円盤全体に一様に広がる。これが**円則**（circular law）であり、Tao and Vu (2010) は平均 0、分散 1 の独立かつ同一分布に従う成分に対して一般の場合にこれを証明した。円則はほとんどすべての固有値がどこに分布するかを教えてくれるが、それ単体では少数の外れ値が飛び出す可能性までは排除しない。しかし別の定理によって外れ値も単位円の外へは出ないことが示されており、確率 1 で $\rho(\mathbf{A})/\sqrt{n} \to 1$ が成り立つ (Geman 1986)。

一方、スペクトルノルムはまったく異なる振る舞いを示し、$\|\mathbf{A}\|_2/\sqrt{n} \to 2$ となる (Yin et al. 1988)。これはマルチェンコ・パスツール則（33.3.2.4 節）においてアスペクト比を $c = 1$ とした特別な場合にあたる。すなわち、$\frac{1}{n}\mathbf{A}^\top\mathbf{A}$ の最大固有値は上端 $(1+\sqrt{c})^2 = 4$ に収束し、$\|\mathbf{A}\|_2^2$ は単にその固有値の n 倍に等しい。$n = 1000$ 程度でも、1 回ランダムに生成しただけで両方の漸近極限のすぐ近くに収まる。

```
n = 1000
A = np.random.randn(n, n)
print(f'rho(A) / sqrt(n)       = {np.abs(np.linalg.eigvals(A)).max() / n ** 0.5:.3f}')
print(f'sigma_max(A) / sqrt(n) = {np.linalg.norm(A, 2) / n ** 0.5:.3f}')
```

```
rho(A) / sqrt(n)       = 1.030
sigma_max(A) / sqrt(n) = 1.990
```

n が大きいとき、スペクトル半径は $\sqrt{n}$ から数パーセント以内に収まる一方、最大の引き伸ばし率は $2\sqrt{n}$ の近くに位置する。つまりランダムなガウス行列は、固有値から予想される値の 2 倍も特定のベクトルを引き伸ばすのである。標準的な初期化手法がランダムな重みを $\sqrt{\text{fan-in}}$ で割るのは、これとはまったく別の理由による。成分の分散を $1/n$ にすると、$\mathbb{E}\|\mathbf{Ax}\|^2 = \|\mathbf{x}\|^2$ が任意の固定されたベクトル $\mathbf{x}$ に対して成り立ち、二乗長が平均的に保たれる。スペクトル半径が 1 の近くになるのはその副産物にすぎず、33.4.2.5 節で見たように、それによって異なる行列の積が制御されるわけではない。驚くべきことに、まさにこのスケーリングを持つ独立な行列を数多く掛け合わせると、個々の因子は平均的に二乗長を保っているにもかかわらず、典型的なベクトルは**縮小**して $\mu < 0$ となる。その理由は対数関数の凹性にある。単位ベクトル $\mathbf{x}$ に対して、$\log \|\mathbf{Ax}\|^2$ の期待値は $\|\mathbf{Ax}\|^2$ の期待値の対数（すなわち $\log 1 = 0$）より真に小さくなり、深い積に沿って足し合わされるのはこの対数のほうだからである（演習問題 5）。

非正規行列 $\mathbf{M}$ とランダムなガウス行列は、どちらも同じ結論を指し示している。有限ステップで行列がベクトルをどれだけ引き伸ばすかについて固有値が教えてくれることはほとんどなく、そもそも長方形行列には固有値が存在さえしない。任意の形状を持つあらゆる行列に対して引き伸ばしを直接測るため、次は 33.5 節の特異値へと進む。

33.4.3 まとめ

行列を分解しなくても、Gershgorin の円盤を用いれば成分から直接固有値の位置を絞り込むことができ、べき乗法を用いれば等比レート $|\lambda_2/\lambda_1|$ で主固有対を抽出できる。これが PageRank を支える仕組みである。スペクトル半径 $\rho(\mathbf{A})$ が支配するのは単一の固定された行列のべき乗における長期的なレートだけであり、それ以上のことは語らない。非正規行列は減衰に転じる前に劇的に増大することがあり、また変化する行列の積については個々の因子の固有値から何もわからない。変化する写像の積に対しては、スペクトルノルムが有限ステップを上から抑え、最大リアプノフ指数が長期的な平均レートを与える。初期化時の $n \times n$ ランダムガウス行列は、固有値と引き伸ばし率のあいだの乖離を具体的に示している。そのスペクトル半径は $\sqrt{n}$ に近い一方、スペクトルノルムは $2\sqrt{n}$ に近くなるのである。

33.4.4 演習問題

1. **スペクトル半径とノルム。**スペクトル半径は行列のすべてのべき乗を下から抑え、正規行列に対しては上からも抑える。
    1. $\rho(\mathbf{A}) \le \|\mathbf{A}^k\|_2^{1/k}$ が任意の正方行列 $\mathbf{A}$ とすべての $k \ge 1$ に対して成り立つことを示せ。
    2. 式変形 Equation 33.39 では 2 つの事実を証明なしに用いている。$\|\mathbf{UXU}^*\|_2 = \|\mathbf{X}\|_2$ が任意のユニタリ行列 $\mathbf{U}$ と任意の正方行列 $\mathbf{X}$ に対して成り立つこと、および対角行列のスペクトルノルムがその対角成分の最大絶対値に等しいことを示せ。
    3. Equation 33.39 の逆は成り立たないことを示せ。すなわち、正規ではない 3×3 行列であって、$\|\mathbf{A}^k\|_2 = \rho(\mathbf{A})^k$ をすべての $k \ge 1$ に対して満たすものを 1 つ見つけよ。
    4. 与えられた $t > 0$ に対し、対角化可能な 2×2 行列でスペクトル半径が $1/2$ かつ $\|\mathbf{A}\|_2 > t$ を満たすものを構成せよ。

2. [code] **べき乗法の収束レート。** $\mathbf{S}$ を Figure 33.21 の行列とする。
   1. Equation 33.38 より、反復ベクトルと $\mathbf{v}_1$ の向きのずれはレート $|\lambda_2/\lambda_1| = 1/3$ で減衰する。このずれを 100 分の 1 以下にするには何回の反復が必要か。
   2. $\mathbf{A}$ を対称行列とし、単位固有ベクトル $\mathbf{v}_1$（固有値 λ_1 に対応）をとる。また $\mathbf{x} = \mathbf{v}_1 + \delta\mathbf{u}$ とおく（ただし $\mathbf{u}$ は $\mathbf{v}_1$ に直交する単位ベクトル）。33.3.2.3 節のレイリー商が $R(\mathbf{x}) - \lambda_1 = O(\delta^2)$ を満たし、δ に関する一次の項を持たないことを示せ。したがって、べき乗法の反復ベクトルに対するレイリー商はどのレートで λ_1 に近づくか。また、この議論のどこで $\mathbf{A}$ が対称であることを用いているか。
   3. Figure 33.21 の実験を、ランダムな初期ベクトルから $\left(\begin{smallmatrix} 1 & 0.1 \\ 0.1 & 1 \end{smallmatrix}\right)$（$\mathbf{S}$ と同じ固有ベクトルを持ち、固有値が 1.1 と 0.9 である行列）に対して繰り返せ。$\mathbf{v}_1$ とのなす角の正弦とレイリー商の誤差を、30 ステップにわたって対数スケールでプロットせよ。両方の傾き、およびそれぞれの誤差が 100 分の 1 に縮小するまでに要するステップ数を予測し、なぜこの行列が $\mathbf{S}$ よりもはるかに遅く収束するのか説明せよ。

3. [code] **擬スペクトルと過渡的増大。** ●$\mathbf{M}$ を非正規行列 Equation 33.41 とし、$\Lambda = \mathrm{diag}(0.9, 0.8)$ を 33.4.2.3 節と同じく同一の固有値を持つ正規行列とする。
   1. Equation 33.43 の 2 つの集合が等しいことを証明せよ。まず $\|(z\mathbf{I} - \mathbf{A})^{-1}\|_2 = 1/\sigma_{\min}(z\mathbf{I} - \mathbf{A})$ を示せ（ただし $\sigma_{\min}(\mathbf{X}) = \min_{\|\mathbf{x}\|=1} \|\mathbf{X}\mathbf{x}\|$）。一方の包含関係については、単位ベクトル $\mathbf{u}$、$\mathbf{v}$ として $(z\mathbf{I} - \mathbf{A})\mathbf{v} = \sigma\mathbf{u}$（$\sigma = \sigma_{\min}(z\mathbf{I} - \mathbf{A})$）を満たすものをとり、$z$ が $\mathbf{A} + \sigma\mathbf{u}\mathbf{v}^*$ の固有値であることを示せ。逆の包含関係については、$(z\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{E}\mathbf{x}$ を固有ベクトル $\mathbf{x}$（$\mathbf{A} + \mathbf{E}$ の固有ベクトル）に適用せよ。33.4.2.4 節で出力したレゾルベントノルムを用いると、$z = 1$ を摂動後の $\mathbf{M}$ の固有値にするにはどの程度の大きさの摂動が必要か。また Λ の場合はどの程度か。
   2. $\|\mathbf{A}^k\|_2 \le K_0$ がすべての $k \ge 0$ で成り立つとする。$|z| > 1$ に対して級数 $\sum_{k\ge 0} \mathbf{A}^k / z^{k+1}$ が $(z\mathbf{I} - \mathbf{A})^{-1}$ に収束することを示し、次を結論せよ。

$$\sup_{k\ge 0} \|\mathbf{A}^k\|_2 \;\ge\; K(\mathbf{A}) = \sup_{|z|>1} \left(|z| - 1\right) \|(z\mathbf{I} - \mathbf{A})^{-1}\|_2 .$$

      量 $K(\mathbf{A})$ は Kreiss 定数と呼ばれる。この下界を擬スペクトルの言葉で言い換えよ。すなわち、$\Lambda_\varepsilon(\mathbf{A})$ が単位円の外側に距離 δ 離れた点を含むとき、$\mathbf{A}$ のあるべき乗は少なくともどのくらいの大きさにならなければならないか。
   3. $K(\mathbf{M})$ を複素点 $z (|z| > 1)$ のグリッド上で数値的に評価し、最大の $\|\mathbf{M}^k\|_2$（33.4.2.3 節で出力した値）と比較せよ。また $K(\Lambda)$ を手計算で求めよ。

4. [code] **PageRank における減衰。** ●$\mathbf{P}$ を列確率的な $n \times n$ 行列とし、$\mathbf{G} = \alpha\mathbf{P} + (1-\alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^\top$（$0 < \alpha < 1$）を 33.4.1.3 節の減衰付き行列とする。
   1. $\mathbf{P}\mathbf{x} = \lambda\mathbf{x}$ とする。$\mathbf{1}^\top\mathbf{x} = 0$ が $\lambda \ne 1$ のときに必ず成り立つこと、および $\mathbf{G}\mathbf{x} = \alpha\lambda\mathbf{x}$ が $\mathbf{1}^\top\mathbf{x} = 0$ のときに必ず成り立つことを示せ。
   2. 固有値 1（行列 $\mathbf{P}$ の固有値）は、リンクが互いに非連結なページ群に分かれている場合のように重複しうる。ここでは、任意の列確率行列 $\mathbf{P}$ が固有値 1 に対する非負の固有ベクトル（定常分布）を持つという事実を用いてよい。この固有空間に属するベクトル $\mathbf{x}$ のうち $\mathbf{1}^\top\mathbf{x} = 0$ を満たすものが、元の固有空間より次元が 1 小さい部分空間をなすことを示せ。これより、対角化可能な $\mathbf{P}$ に対して $\mathbf{G}$ の固有値は 1 と $\alpha\lambda_2(\mathbf{P}), \dots, \alpha\lambda_n(\mathbf{P})$ になり、したがって上界 $|\lambda_2(\mathbf{G})| \le \alpha$（Haveliwala and Kamvar (2003)）が得られることを結論せよ。

3. 4 つのページが巡回路 $1 \to 2 \to 3 \to 4 \to 1$ をなしてリンクしているとする。このリンク行列 $\mathbf{P}$ の固有値を求めよ。ページ 1 から出発する減衰なしのべき乗法が決して収束しないのはなぜか。
4. $\alpha = 0.85$ と (c) の巡回路に対し、$\mathbf{G}$ 上のべき乗法をページ 1 から実行し、定常分布までの ℓ_1 距離が 10^{-8} を下回るまでの反復回数を数えよ。(b) による予測 $0.85^{k} \le 10^{-8}$ と比較せよ。
5. 33.4.1.3 節の 3 ページの例について、$\mathbf{G}$ の固有値が固有値 1 を除いて α 倍された $\mathbf{P}$ の固有値になっていること、および定常分布が $(0.3, 0.4, 0.3)$ から約 $(0.303, 0.394, 0.303)$ へ移動することを数値的に確認せよ。減衰によって定常分布が一様分布へ近づくのはなぜか。

5. [code] **ランダム行列積のリアプノフ指数。●●** $\mathbf{D}_1, \mathbf{D}_2, \dots$ を独立な $n \times n$ 行列とし、各成分は独立に $\mathcal{N}(0, 1/n)$ に従うものとする（ファンイン初期化における深層ネットワークのヤコビ行列の線形モデル）。また $\mathbf{x}_0$ を単位ベクトルとする。33.4.2.6 節より、各因子のスペクトル半径は 1 に近い（n が大きいとき）。33.4.2.5 節では、この積の増大率を次の最大リアプノフ指数で要約した。

$$\mu = \lim_{T \to \infty} \frac{1}{T} \log \left\| \mathbf{D}_T \cdots \mathbf{D}_1 \mathbf{x}_0 \right\|,$$

この極限は Furstenberg と Kesten の定理により確率 1 で存在する。

1. $\mathbb{E}\|\mathbf{D}\mathbf{x}\|^2 = \|\mathbf{x}\|^2$ が任意の固定された $\mathbf{x}$ に対して成り立ち、したがってこのスケーリングが二乗ノルムを平均的に保つことを示せ。
2. $\mathbf{u}_0 = \mathbf{x}_0$ および $\mathbf{u}_t = \mathbf{x}_t / \|\mathbf{x}_t\|$ とおく（ただし $\mathbf{x}_t = \mathbf{D}_t \cdots \mathbf{D}_1 \mathbf{x}_0$）。$\mathbf{D}_1, \dots, \mathbf{D}_{t-1}$ が与えられた条件のもとで、$n\|\mathbf{D}_t \mathbf{u}_{t-1}\|^2$ はどの分布に従うか。その結果を用いて、$\log \|\mathbf{D}_T \cdots \mathbf{D}_1 \mathbf{x}_0\|$ を独立な項の和として表せ。さらに $\mathbb{E} \log \chi^2_n = \psi(n/2) + \log 2$（$\psi$ はディガンマ関数）を用い、Cohen and Newman (1984) による次の閉じた式を導出せよ。

$$\mu = \tfrac{1}{2}\big(\psi(n/2) + \log 2 - \log n\big).$$

3. (a) が成り立つにもかかわらず、$\mu < 0$ がすべての n に対して成り立つことを示せ。μ を $n = 1$ の場合に閉じた形で求めよ（$\psi(1/2) = -\gamma_{\mathrm{E}} - 2\log 2$ とオイラー定数 $\gamma_{\mathrm{E}} \approx 0.5772$ を用いよ）。さらに $\psi(x) = \log x - \frac{1}{2x} + O(x^{-2})$ を用いて、$\mu \approx -1/(2n)$ が大きな n に対して成り立つことを示せ。
4. 再正規化を伴う $T = 10^4$ ステップのシミュレーションを $n \in \{1, 2, 5, 50\}$ に対して実行し、計測された指数を (b) の理論値、ならびに単一因子のスペクトル半径の対数および最大特異値の対数と比較せよ。この初期化を用いた幅 50、深さ 100 の線形ネットワークでは、典型的な入力は何倍に縮小するか。また、この比較は 33.4.2.5 節の 3 つの量について何を示唆するか。

33.5 特異値分解と低ランク近似

33.3 節では、固有値分解（eigendecomposition）が正方行列を「単に伸縮するだけの方向」によって記述し、行列のべき乗を単なる数値のべき乗へと帰着させることを見た。しかし、深層学習に現れる大半の行列はそのような性質を持たない。重み行列はある幅の入力を別の幅の出力へ写し、データ行列はデータ例ごとに 1 行、特徴量ごとに 1 列を持つ。どちらも自身の出力へ繰り返

し適用されるわけではなく、正方行列である必要もない。さらに行列が**正方行列であっても**、固有値だけでは不十分なことがある。せん断行列 $\mathbf{T} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ は**欠陥**（defective）を持ち、平面を張るには 2 方向が必要なところ固有ベクトルの方向を 1 つしか持たない。また 33.4 節では、2 つの固有値がともに 1 より小さいにもかかわらず、6 乗するとベクトルを 10 倍に引き伸ばす行列に出会った。ここで必要なのは、**あらゆる**行列に対して存在し、伸縮を直接測ることができ、大きな行列を精度よく近似するのにどれほど少ない数値で足りるかを教えてくれる分解である。

最後の要件のありがたみは、具体的な写真を考えると実感しやすい。400×500 ピクセルの猫のグレースケール画像は、単なる 200,000 個の数値からなる行列である。他のあらゆる行列と同様、この画像も 1 本の列と 1 本の行の外積（outer product）で作られるランク 1 の成分の和として書ける。驚くべきことに、うまく選んだわずか 20 個の成分（18,020 個の数値、すなわち元データの 9%）だけで、相対誤差わずか 8% で写真を再現できる（Figure 33.28）。どの 20 個の成分がこれを達成するのか、また別の 20 個を選べばさらに良くできるのだろうか。深層学習の至るところでこれと同じ問いが現れる。大規模モデルのファインチューニング用アダプタにどれだけのパラメータが必要か、そしてデータセットのどの方向が本当に残す価値があるのかを、この問いが教えてくれるのである。

鍵となる考え方は、入力と出力の両方を単一の基底で済ませようとするのをやめることである。入力空間に 1 つの正規直交基底（orthonormal basis）を、出力空間に別の正規直交基底を用意すれば、**あらゆる**行列は独立な非負の伸縮率の列 $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ へと帰着する。このたった 1 つのリストから、行列のランク、最良の低ランク近似（low-rank approximation）、主成分分析（principal component analysis、PCA）、最小二乗法（least squares）の解、さらには連立一次方程式（linear system）が測定誤差に対してどれほど敏感かまで読み取れる。1.3 節では、行列の大きさを測る 2 つの方法として、フロベニウスノルム $\|\mathbf{A}\|_F$（全成分の二乗和の平方根）とスペクトルノルム $\|\mathbf{A}\|_2$（$\mathbf{A}$ が単位ベクトルを引き伸ばせる最大の倍率）を導入した。本節では、そのどちらも特異値（singular value）σ_i の単純な関数として表される。

以下の議論の柱となるのは 2 つの主要な結果である。1 つは任意の行列に対する特異値分解（singular value decomposition、SVD）の存在であり、もう 1 つは分解を途中で打ち切ることであらゆるランクにおける最良近似が得られると保証する Eckart–Young の定理である。そこから、擬似逆行列（pseudoinverse）と条件数（condition number）がわずか数行で導かれる。最後の小節ではこれらの道具を現代の深層学習と結びつけるが、初読時には飛ばしてもよい。以下の数値実験はすべて NumPy を用い、特に断らない限り、本節を通じて $\mathbf{T}$ はせん断行列、$\mathbf{A}$ は一般の行列を表す。

33.5.1 特異値分解

33.5.1.1 分解とその幾何学

1.3 節では、対称行列 $\left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ が単位円を楕円へ写し、その半軸が固有ベクトル $(1,1)$ と $(-1,1)$ の方向に沿って長さ 3 と 1 を持つことを見た。驚くべきことに、まもなく証明するように、**非対称**行列もまた単位円を楕円へと写す（行列が特異行列であれば線分に潰す）。違いは、この楕円の軸の上に着地する直交した入力方向が、もはやその軸自身の方向とは一致しない点にある。代わりに、入力空間用の正規直交フレームと出力空間用の正規直交フレームという 2 つの正規直交フレームが存在し、特異値分解はその両方に名前を与える。

定理（特異値分解）。 任意の実 $m \times n$ 行列 $\mathbf{A}$ は次のように書ける。

$$\mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V}^{\top}, \tag{33.44}$$

ここで、$\mathbf{U} \in \mathbb{R}^{m \times m}$ と $\mathbf{V} \in \mathbb{R}^{n \times n}$ は直交行列（$\mathbf{U}^{\top}\mathbf{U} = \mathbf{I}$、$\mathbf{V}^{\top}\mathbf{V} = \mathbf{I}$）であり、$\Sigma \in \mathbb{R}^{m \times n}$ は主対角線上の非負成分 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$ を除いてすべて 0 である。列ベクトル $\mathbf{v}_i$

（$\mathbf{V}$ の各列）を**右特異ベクトル**（right singular vector）、列ベクトル $\mathbf{u}_i$（$\mathbf{U}$ の各列）を**左特異ベクトル**（left singular vector）、対角成分 σ_i を**特異値**と呼ぶ。

幾何学的に言えば、形状にかかわらず**あらゆる**行列は、直交変換、対角方向の伸縮、そして 2 つ目の直交変換を順に施したものにほかならない。$\mathbf{AV} = \mathbf{U}\Sigma$ と書き直して第 i 列を読み取れば、1 方向ずつの作用が明示的にわかる。

$$\mathbf{A}\mathbf{v}_i = \sigma_i \, \mathbf{u}_i, \qquad i \le \min(m, n). \tag{33.45}$$

各入力方向 $\mathbf{v}_i$ は対応する出力方向 $\mathbf{u}_i$ へと写され、σ_i 倍に伸縮される。$n > m$ のときは、余分な入力方向 $\mathbf{v}_{m+1}, \dots, \mathbf{v}_n$ はゼロに潰される。Equation 33.44 を右から左へ読むと、$\mathbf{A}$ をベクトル $\mathbf{x}$ に作用させる操作は次の 3 段階で進む。

1. $\mathbf{V}^{\top}$ は入力を回転させ、各右特異ベクトル $\mathbf{v}_i$ を座標軸 $\mathbf{e}_i$ に揃える。
2. Σ は軸 i を σ_i 倍に伸縮し、$m \neq n$ のときは出力次元に合わせてゼロ座標を追加するか、余分な座標を切り捨てる。
3. $\mathbf{U}$ は伸縮された各軸を回転させ、軸 i が $\mathbf{u}_i$ の方向を向くようにする。

ここで「回転」とは直交変換の便宜的な略称であり、鏡映を含む場合もある。

Figure 33.24 に、せん断 $\mathbf{T}$ におけるこれら 3 段階を示す。特異値は $\sigma_1 = 1.618$ と $\sigma_2 = 0.618$ であり、第 1 右特異ベクトル $\mathbf{v}_1$ は 58.3° の角度で x 軸の上方を向く一方、その像の方向 $\mathbf{u}_1$ は 31.7° を向く。入力フレームと出力フレームが異なるため、$\mathbf{U} \neq \mathbf{V}$ となる。これらの正確な数値は 33.5.1.2 節で導出する。

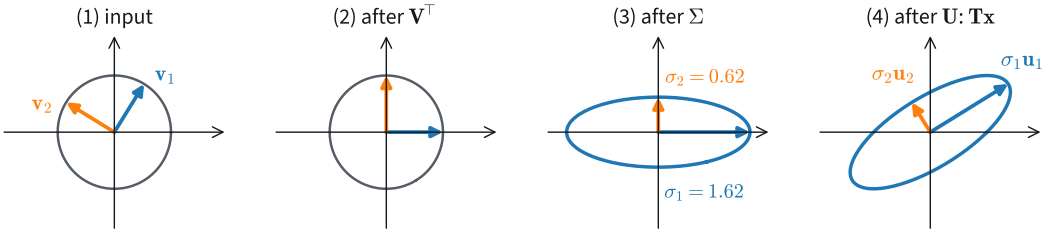


Figure 33.24: 単位円に作用するせん断 $\mathbf{T} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) = \mathbf{U}\Sigma\mathbf{V}^{\top}$（左から右へ読む）。$\mathbf{V}^{\top}$ が -58.3° 回転させ、Σ が $\varphi = 1.618$ 倍に x 方向へ、$1/\varphi = 0.618$ 倍に y 方向へ伸縮し、$\mathbf{U}$ が 31.7° 回転させる。入力フレーム $\{\mathbf{v}_i\}$ と出力フレーム $\{\mathbf{u}_i\}$ は異なる。

n 次元でも同じ 3 段階によって、単位球面はベクトル $\sigma_i\mathbf{u}_i$ を半軸とする楕円体へと変形される。入力方向のうち正の特異値を持つものが $r < n$ 個だけであれば、残りの $n - r$ 方向はゼロに潰れ、像は平坦な r 次元楕円体となって $\mathbf{u}_1, \dots, \mathbf{u}_r$ が張る空間の内部に収まる。

コンパクト形式と薄い形式。正の特異値の個数を r とする（33.5.1.5 節で r がまさに $\mathbf{A}$ のランクに一致することを確かめる）。Σ の最初の r 個を超える行と列はすべて 0 であるため、それらと対になる特異ベクトルは積に一切寄与しない。これらのゼロ成分を取り除くと、**コンパクト SVD**（compact SVD）$\mathbf{A} = \mathbf{U}_r\Sigma_r\mathbf{V}_r^{\top}$ が得られる。ここで $\mathbf{U}_r \in \mathbb{R}^{m \times r}$ と $\mathbf{V}_r \in \mathbb{R}^{n \times r}$ は最初の r 本の特異ベクトルを並べた行列であり、$\Sigma_r = \mathrm{diag}(\sigma_1, \dots, \sigma_r)$ である。完全形式とコンパクト形式の中間に位置するのが**薄い SVD**（thin SVD）である。これは数値計算ライブラリで縮小形式を指定した際に返される既定の形式であり、$\min(m, n)$ 本の列を $\mathbf{U}$ と $\mathbf{V}$ に残しつつ、$r < \min(m, n)$ のときにはゼロ特異値もそのまま記録する。ランク 2 の 3×2 行列では、完全形式の因子の形状は 3×3、3×2、2×2 であるのに対し、薄い形式の因子の形状は 3×2、2×2、2×2 となる（この行列は列フルランクであるため、ここではコンパクト SVD と一致する）。完全形式と薄い形

式は行列が正方でないときに異なり、薄い形式とコンパクト形式は行列がランク落ちしているときに異なる。Figure 33.25 に、3 つの形式がすべて異なるランク 3 の 6×4 行列を示す。

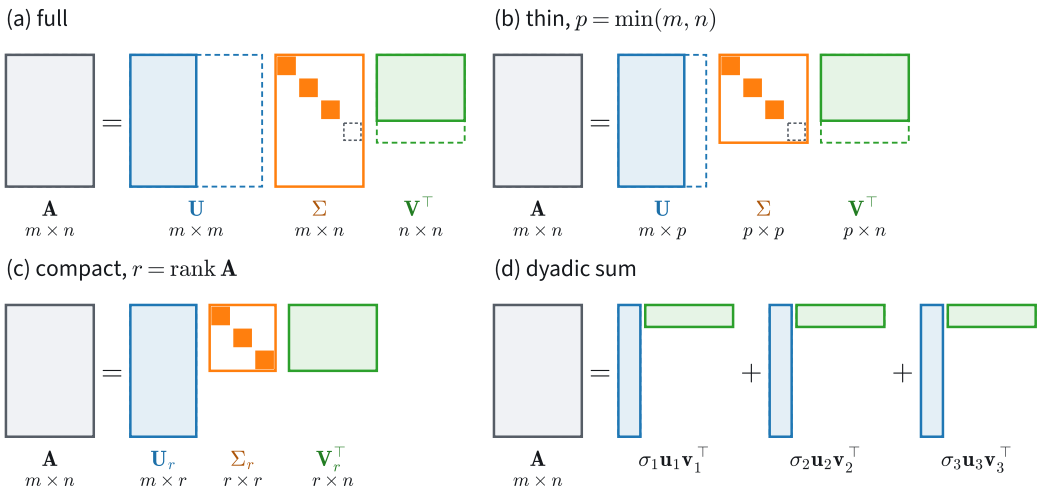


Figure 33.25: $m \times n$ 行列（ランク r）における SVD 因子の形状（$m = 6$、$n = 4$、$r = 3$ の縮尺で描画）。(a) 完全形式、(b) 薄い形式、(c) コンパクト形式、(d) r 個のランク 1 項からなるダイアド和。Σ の対角線上の塗りつぶされた正方形は正の特異値であり、破線の正方形はゼロ特異値である。$\mathbf{U}$ と $\mathbf{V}^\top$ の破線部分は Σ のゼロとしか掛からないため、より小さい各形式では積を変えずにそれらの一部を省ける。

ダイアド和。コンパクト SVD を列と行の積として展開すると、$\mathbf{A}$ はしばしば**ダイアド**と呼ばれるランク 1 行列の重み付き和（ダイアド和、dyadic sum）として表される。

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^\top . \tag{33.46}$$

各項 $\mathbf{u}_i \mathbf{v}_i^\top$ は出力方向と入力方向の外積に特異値の重みを掛けたものであり、重い項から軽い項へとあらかじめ並んでいる。せん断行列の場合は $\mathbf{T} = \varphi \, \mathbf{u}_1 \mathbf{v}_1^\top + \varphi^{-1} \, \mathbf{u}_2 \mathbf{v}_2^\top$ となり、重み 1.618 の成分と重み 0.618 の 2 つ目の成分に分解される。このうち重い成分だけを残すというのが、33.5.2.1 節で扱う低ランク近似の基本的な考え方である。

33.5.1.2 存在性

σ_i、$\mathbf{v}_i$、$\mathbf{u}_i$ はどこから現れ、なぜ出力方向まで自動的に正規直交になるのだろうか。33.3.2.1 節のスペクトル定理により、あらゆる対称行列は実固有値を持つ固有ベクトルの正規直交基底を備えている。これを**グラム行列**（Gram matrix）$\mathbf{A}^\top \mathbf{A}$、すなわちサイズ $n \times n$ の $\mathbf{A}$ の列同士のドット積からなる行列に適用できる。この行列は対称行列であり、さらに半正定値（positive semidefinite、33.3.2.2 節）でもある。なぜなら、$\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} = \|\mathbf{A}\mathbf{x}\|^2 \geq 0$ がすべてのベクトル $\mathbf{x}$ に対して成り立つからである。

Equation 33.44 の証明（存在性）。スペクトル定理より、$\mathbf{A}^\top \mathbf{A}$ は固有ベクトルの正規直交基底 $\mathbf{v}_1, \ldots, \mathbf{v}_n$ と、対応する固有値 $\lambda_1 \geq \cdots \geq \lambda_n$ を持つ。$\lambda_i = \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \|\mathbf{A}\mathbf{v}_i\|^2$ であるから、すべての固有値は非負である。そこで次のように定義する。

$$\sigma_i = \sqrt{\lambda_i}, \qquad r = \#\{i : \sigma_i > 0\}.$$

すなわち、特異値はグラム行列の固有値の平方根であり、r は正の特異値の個数を表す。各 $i \le r$ に対して $\mathbf{u}_i = \mathbf{A}\mathbf{v}_i / \sigma_i$ とおく。ここで最も重要な確認事項は、こうして写されたベクトルが自動的に正規直交になることであり、これは $\mathbf{v}_j$ が $\mathbf{A}^{\top}\mathbf{A}$ の固有ベクトルであることと、$\mathbf{v}_i$ が正規直交であることから直ちに従う。

$$\mathbf{u}_i^{\top}\mathbf{u}_j = \frac{1}{\sigma_i \sigma_j}\, \mathbf{v}_i^{\top}\mathbf{A}^{\top}\mathbf{A}\,\mathbf{v}_j = \frac{\lambda_j}{\sigma_i \sigma_j}\, \mathbf{v}_i^{\top}\mathbf{v}_j = \frac{\lambda_j}{\sigma_i \sigma_j}\, \delta_{ij} = \delta_{ij}.$$

したがって、入力フレームの正規直交性は $\mathbf{A}^{\top}\mathbf{A}$ を介してそのまま出力フレームへと受け継がれる。次に次元を確認しよう。r 本の正規直交ベクトル $\mathbf{u}_1, \dots, \mathbf{u}_r$ は $\mathbb{R}^m$ に属するため $r \le m$ であり、固有値は全部で n 個あるため $r \le n$ である。ゆえに $r \le \min(m, n)$ となり、$\lambda_i = 0$ がすべての $i > r$ で成り立つ。$\mathbf{u}_1, \dots, \mathbf{u}_r$ を延長して正規直交基底 $\mathbf{u}_1, \dots, \mathbf{u}_m$（$\mathbb{R}^m$ の基底）を作る（たとえば任意の基底延長に Gram–Schmidt 法を適用すればよい）。$r < i \le n$ に対しては $\|\mathbf{A}\mathbf{v}_i\|^2 = \lambda_i = 0$ であるから、$\mathbf{A}\mathbf{v}_i = \mathbf{0}$ となる。ここで $\mathbf{AV}$ と $\mathbf{U}\Sigma$ を 1 列ずつ比較すると、第 i 列（$\mathbf{U}\Sigma$ の列）は、$\sigma_i \mathbf{u}_i$（$i \le \min(m,n)$ のとき。$i > r$ ではゼロになる）または $\mathbf{0}$（$i > m$ のとき。この範囲には Σ の対角成分がない）である。いずれの場合も、第 i 列（$\mathbf{AV}$）は第 i 列（$\mathbf{U}\Sigma$）と一致する。よって $\mathbf{AV} = \mathbf{U}\Sigma$ が成り立ち、右から $\mathbf{V}^{\top} = \mathbf{V}^{-1}$ を掛ければ $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\top}$ が得られる。$\blacksquare$

この証明はそのまま計算手順にもなっている。対称行列 $\mathbf{A}^{\top}\mathbf{A}$ を対角化し、固有値の平方根をとって特異値を求め、固有ベクトルを $\mathbf{A}$ で写して左特異ベクトルを得ればよい。

各ステップの働きを具体的に確かめるため、せん断行列を手計算で追ってみよう。$\mathbf{T} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ に対し、グラム行列は $\mathbf{T}^{\top}\mathbf{T} = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ である。トレースは 3、行列式は 1 であるから、特性多項式は $\lambda^2 - 3\lambda + 1$ となり、固有値 $(3 \pm \sqrt{5})/2$ が得られる。$\varphi^2 = \varphi + 1$ を満たす黄金比 $\varphi = (1+\sqrt{5})/2$ を導入すると、これらの固有値は φ^2 と φ^{-2} へと綺麗に簡約される。したがって特異値は $\sigma_1 = \varphi \approx 1.618$ および $\sigma_2 = 1/\varphi \approx 0.618$ となり、その積は $1 = |\det \mathbf{T}|$ である。

続いて特異ベクトルを求めよう。第 1 右特異ベクトル (x, y) について、$(\mathbf{T}^{\top}\mathbf{T} - \varphi^2 \mathbf{I})(x, y) = \mathbf{0}$ の第 1 行は $(1 - \varphi^2)x + y = 0$ となり、$\varphi^2 - 1 = \varphi$ を用いると $y = \varphi x$、すなわち $\mathbf{v}_1 \propto (1, \varphi)$ が得られる。これに $\mathbf{T}$ を作用させると $\mathbf{T}(1, \varphi) = (1 + \varphi, \varphi) = \varphi\,(\varphi, 1)$ となるため、$\mathbf{u}_1 \propto (\varphi, 1)$ である。両ベクトルとも長さは $\sqrt{1 + \varphi^2} = \sqrt{\varphi + 2}$ であり、正規化すると $\mathbf{v}_1 \approx (0.526, 0.851)$（角度 $\arctan \varphi = 58.3^{\circ}$）および $\mathbf{u}_1 \approx (0.851, 0.526)$（角度 31.7°）となる。第 2 の組は直交性から $\mathbf{v}_2 = (-\varphi, 1)/\sqrt{\varphi + 2}$、$\mathbf{u}_2 = (-1, \varphi)/\sqrt{\varphi + 2}$ と求まり、実際 $\mathbf{T}\mathbf{v}_2 = (1 - \varphi, 1)/\sqrt{\varphi + 2} = \varphi^{-1}\mathbf{u}_2$ が $(1 - \varphi)\varphi = -1$ より成り立つ。$\mathbf{V}$ と $\mathbf{U}$ はどちらも回転行列 $\mathbf{R}_{\theta} = \left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$ であるため、せん断行列は次のように分解される。

$$\mathbf{T} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \mathbf{R}_{31.7^{\circ}} \begin{bmatrix} \varphi & 0 \\ 0 & 1/\varphi \end{bmatrix} \mathbf{R}_{58.3^{\circ}}^{\top}. \tag{33.47}$$

幾何学的には、このせん断は平面を -58.3° 回転させ、1.618 倍に x 軸方向へ、0.618 倍に y 軸方向へ伸縮し、最後に 31.7° だけ回転させて戻す。2 つの角度の和が 90° になるのは、$\mathbf{u}_1$ が単に $\mathbf{v}_1$ の座標を入れ替えたものだからであり、これは対称性 $\mathbf{T}^{\top} = \mathbf{J}\mathbf{T}\mathbf{J}$（入れ替え行列 $\mathbf{J} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ に関する関係）を反映している。

NumPy の `np.linalg.svd` はこの手計算と一致するだろうか。特異ベクトルは符号を除いてしか定まらないため（33.5.1.3 節）、以下のセルではせん断行列を `A` として格納し、まず $\mathbf{u}_1$ が第 1 象限に入り $\mathbf{U}$ が純粋な回転になるよう符号を揃えている。

```
A = np.array([[1., 1.], [0., 1.]])
U, s, Vt = np.linalg.svd(A)
if U[0, 0] < 0:                          # joint sign choice: u_1 in the first quadrant
    U[:, 0], Vt[0] = -U[:, 0], -Vt[0]
if np.linalg.det(U) < 0:                 # joint sign choice: U a rotation (then so is V)
    U[:, 1], Vt[1] = -U[:, 1], -Vt[1]
phi = (1 + np.sqrt(5)) / 2
print('singular values', s.round(4), '  phi, 1/phi', np.array([phi, 1 / phi]).round(4))
print('U =', U.round(3).tolist(), '  Vt =', Vt.round(3).tolist())
print('angles of u_1, v_1 (degrees):', np.degrees(np.arctan2(U[1, 0], U[0, 0])).round(1),
      np.degrees(np.arctan2(Vt[0, 1], Vt[0, 0])).round(1))
print('A v_1 = sigma_1 u_1:', np.allclose(A @ Vt[0], s[0] * U[:, 0]),
      '  sigma_1 sigma_2 =', s.prod().round(6))
```

```
singular values [1.618 0.618]   phi, 1/phi [1.618 0.618]
U = [[0.851, -0.526], [0.526, 0.851]]   Vt = [[0.526, 0.851], [-0.851, 0.526]]
angles of u_1, v_1 (degrees): 31.7 58.3
A v_1 = sigma_1 u_1: True   sigma_1 sigma_2 = 1.0
```

結果は一致する。ライブラリは φ と $1/\varphi$ に加え、Equation 33.47 の 2 つの回転を返す。

同じ構成法は長方形行列に対してもそのまま機能する。3×2 行列として、各行が $(3,1)$、$(1,3)$、$(0,2)$ であるものを考えよう。そのグラム行列は $\left(\begin{smallmatrix} 10 & 6 \\ 6 & 14 \end{smallmatrix}\right)$ であり、固有値は $12 \pm \sqrt{40}$ である。次のセルでは、証明の 3 ステップ（$\mathbf{A}^{\top}\mathbf{A}$ の固有ベクトル、平方根、および $\mathbf{u}_i = \mathbf{A}\mathbf{v}_i/\sigma_i$）をそのまま実行し、結果を `np.linalg.svd` と比較する。

```
A = np.array([[3., 1.], [1., 3.], [0., 2.]])
lam, V = np.linalg.eigh(A.T @ A)
lam, V = lam[::-1], V[:, ::-1]                    # descending order
sigma = np.sqrt(lam)
U = A @ V / sigma                                  # u_i = A v_i / sigma_i
U_lib, s_lib, Vt_lib = np.linalg.svd(A, full_matrices=False)
print('sigma from eigh:', sigma.round(5), '  from svd:', s_lib.round(5))
print('U orthonormal:', np.allclose(U.T @ U, np.eye(2)), '  same U, V up to sign:',
      np.allclose(abs(U), abs(U_lib)), np.allclose(abs(V), abs(Vt_lib.T)))
print('rebuild error:', np.linalg.norm((U * sigma) @ V.T - A).round(12))
```

```
sigma from eigh: [4.28072 2.38232]   from svd: [4.28072 2.38232]
U orthonormal: True   same U, V up to sign: True True
rebuild error: 0.0
```

どちらの方法でも $\sigma_1 = \sqrt{12+\sqrt{40}} \approx 4.281$ と $\sigma_2 = \sqrt{12-\sqrt{40}} \approx 2.382$ が得られる。証明で保証されたとおり、写されたベクトル $\mathbf{u}_i$ は正規直交となり、$\mathbf{U}$ と $\mathbf{V}$ はともに符号を除いてライブラリの出力と一致する。

33.5.1.3 一意性と最大伸縮率

行列の SVD は一意だろうか。特異**値**は常に一意である。$\mathbf{A}^{\top}\mathbf{A}$ の固有値の平方根であり、$\mathbf{A}$ によって完全に定まるからである。これに対し、特異**ベクトル**には次の 3 種類の自由度がある。

- 単根（重複しない）の $\sigma_i > 0$ に対しては、自由度は同時符号反転 $(\mathbf{u}_i, \mathbf{v}_i) \to (-\mathbf{u}_i, -\mathbf{v}_i)$ のみである。片方だけの符号を反転させると $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$ が崩れるため、両方のベクトルを同時に反転させなければならない。

- 重複特異値 $\sigma > 0$（添字集合を $\mathcal{J}$ とする）に対しては、列集合 $\mathbf{V}_{\mathcal{J}}$ に同じサイズの任意の直交行列 $\mathbf{Q}$ を掛けて $\mathbf{V}_{\mathcal{J}}\mathbf{Q}$ と回転させてよい。ただし、$\mathbf{U}_{\mathcal{J}}$ にも**同じ** $\mathbf{Q}$ を掛けて $\mathbf{U}_{\mathcal{J}} = \mathbf{A}\mathbf{V}_{\mathcal{J}}/\sigma$ を保つ必要がある。
- $\sigma = 0$ に対しては、零空間のベクトル $\mathbf{v}_{r+1}, \dots, \mathbf{v}_n$ と左零空間のベクトル $\mathbf{u}_{r+1}, \dots, \mathbf{u}_m$ はダイアド和 Equation 33.46 に現れないため、互いに完全に独立に回転させることができる。

2 つの数値計算ライブラリ（あるいは同じライブラリでも異なる 2 台の計算機）が特異ベクトルに対して逆の符号を返すことがあるのも、また本節のコードセルでベクトルを表示する前に符号の規約を固定しているのも、この自由度のためである。

$\mathbf{A}$ を分解するだけでなく、最大特異値には直接的な幾何学的意味がある。それは $\mathbf{A}$ が任意の単位ベクトルを引き伸ばせる最大の倍率である。

$$\sigma_1 = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\| = \|\mathbf{A}\|_2. \tag{33.48}$$

これは 33.3.2.3 節の Rayleigh の命題から直ちに従う。すなわち、単位ベクトル $\mathbf{x}$ 全体にわたって二次形式（quadratic form）$\mathbf{x}^{\top}\mathbf{S}\mathbf{x}$（対称行列 $\mathbf{S}$ に対するもの）を最大化すると、最大固有ベクトルにおいて最大固有値に等しい最大値をとる。$\mathbf{S} = \mathbf{A}^{\top}\mathbf{A}$ とおけば、$\|\mathbf{A}\mathbf{x}\|^2 = \mathbf{x}^{\top}\mathbf{A}^{\top}\mathbf{A}\mathbf{x}$ は $\mathbf{x} = \mathbf{v}_1$ で最大値 $\lambda_1 = \sigma_1^2$ をとることがわかる。したがって、σ_1 はまさにスペクトルノルム $\|\mathbf{A}\|_2$（1.3 節）に一致する。せん断行列では、いかなる単位ベクトルも $\varphi = 1.618$ 倍を超えて引き伸ばされることはなく、$\mathbf{v}_1$ がその最大伸縮率を達成する。残りの特異値も、先行する右特異ベクトルと直交する空間に制限して同じ最大化問題を解くことで順に得られる。この手続きを**デフレーション**と呼び、$\sigma_k = \max\{\|\mathbf{A}\mathbf{x}\| : \|\mathbf{x}\| = 1,\ \mathbf{x} \perp \mathbf{v}_1, \dots, \mathbf{v}_{k-1}\}$ と書ける。実際、演習問題 1a ではこの論理を逆転させ、スペクトル定理を使わずに最大伸縮率だけから SVD 全体を構成する。

33.5.1.4 特異値と固有値

特異値と固有値はどのような関係にあるのだろうか。Equation 33.44 を 2 つのグラム行列に代入し、$\mathbf{U}^{\top}\mathbf{U} = \mathbf{I}$ と $\mathbf{V}^{\top}\mathbf{V} = \mathbf{I}$ を用いると次式が得られる。

$$\mathbf{A}^{\top}\mathbf{A} = \mathbf{V}\Sigma^{\top}\Sigma\mathbf{V}^{\top}, \qquad \mathbf{A}\mathbf{A}^{\top} = \mathbf{U}\Sigma\Sigma^{\top}\mathbf{U}^{\top}. \tag{33.49}$$

これら両式はいずれも対称行列の固有値分解になっている。すなわち、右特異ベクトルは $\mathbf{A}^{\top}\mathbf{A}$ の固有ベクトル、左特異ベクトルは $\mathbf{A}\mathbf{A}^{\top}$ の固有ベクトルであり、二乗 σ_i^2 は 2 つのグラム行列が共有する非ゼロ固有値である。$\mathbf{A}$ が対称半正定値行列のときは、SVD と固有値分解は $\mathbf{U} = \mathbf{V}$ として一致する。

しかし、それ以外のあらゆる行列では 2 つの分解は異なり、特異値が固有値の絶対値に等しいと思い込むのはよくある誤りである。本章で繰り返し用いている行列 $\mathbf{A} = \left(\begin{smallmatrix} 1 & 2 \\ -1 & 3 \end{smallmatrix}\right)$ を考えよう。その固有値は $2 \pm i$ であり、絶対値はともに $\sqrt{5} \approx 2.236$ である（33.3 節）。一方、グラム行列 $\mathbf{A}^{\top}\mathbf{A} = \left(\begin{smallmatrix} 2 & -1 \\ -1 & 13 \end{smallmatrix}\right)$ はトレース 15、行列式 25 を持つため、$\sigma_{1,2}^2 = (15 \pm 5\sqrt{5})/2$ となり、したがって $\sigma_{1,2} = (5 \pm \sqrt{5})/2 \approx 3.618,\ 1.382$ となって 2.236 とは大きく異なる。それでもなお、あらゆる正方行列において 2 つのリストを結ぶ明快な恒等式が 1 つ成り立つ。

$$|\det \mathbf{A}| = \prod_{i} \sigma_i = \prod_{i} |\lambda_i|.$$

第 1 の等号は直交行列の行列式が ± 1 であることから従い、第 2 の等号は行列式が全固有値の積に等しいことから従う。2×2 行列が共通の絶対値 $|\lambda|$ を持つ共役複素固有値の対を備えるとき、

ここから $|\lambda| = \sqrt{\sigma_1 \sigma_2}$ が強制される。実際、この例でも $\sqrt{3.618 \cdot 1.382} = \sqrt{5}$ が成り立っている。またこの恒等式は、33.2 節で見た行列式の体積としての解釈も復元する。回転は面積を保つため、面積変化のすべてを対角伸縮 Σ が担い、上の例題行列は単位正方形を面積 $3.618 \cdot 1.382 = 5$ の平行四辺形へと写す。

実用上の SVD 計算。 本節の存在証明では $\mathbf{A}^\top \mathbf{A}$ から SVD を構成したが、数値計算ライブラリがその積を直接作ることは決してない。$\mathbf{A}^\top \mathbf{A}$ を作ると条件数が二乗され（33.5.3.2 節）、浮動小数点演算において小さな特異値が失われてしまうからである。その代わりにライブラリは、$\mathbf{A}$ に直接直交変換を施して二重対角形へと簡約し（Golub–Kahan 二重対角化）、その二重対角行列を対角化する (Golub and Van Loan 1996)。

33.5.1.5 4 つの基本部分空間

$\mathbf{A}$ はどの入力をゼロに潰し、どの出力に到達できるのだろうか。33.2 節では、これらの集合をそれぞれ零空間（null space）と列空間（column space）と名付け、線形独立な列の最大本数としてランクを定義した。$\mathbf{U}$ と $\mathbf{V}$ は可逆であるため、これらを掛けても次元は一切変わらない。したがって $\operatorname{rank} \mathbf{A} = \operatorname{rank} \Sigma$ となり、ランクは単なる非ゼロ対角成分の個数に帰着する。

$$\operatorname{rank} \mathbf{A} = \#\{i : \sigma_i > 0\} = r. \tag{33.50}$$

言い換えれば、ランクとは潰れずに生き残る方向の数を数えたものである。後述する丸め許容誤差と組み合わせれば、特異値を数える方法はランクを判定する上で数値的に最も信頼性が高い。ガウスの消去法より計算コストは高いものの、はるかに安定している。浮動小数点の丸め誤差（rounding error）のような摂動 $\mathbf{E}$ が加わっても、いかなる特異値も $\|\mathbf{E}\|_2$ を超えてずれることはないからである（33.5.2.1 節の Weyl の不等式）。したがって、大きな特異値と微小な特異値の間に明確なギャップがあれば、その結果を信頼できる。

さらに都合のよいことに、特異ベクトルを添字 r で分割すれば、$\mathbf{A}$ の 4 つの基本部分空間（fundamental subspace）すべての明示的な正規直交基底がダイアド和 Equation 33.46 から直接得られる。

- $\mathbf{A}$ の**列空間**（値域）は $\operatorname{span}\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ である。あらゆる出力 $\mathbf{A}\mathbf{x} = \sum_{i \le r} \sigma_i (\mathbf{v}_i^\top \mathbf{x}) \mathbf{u}_i$ はこれらのベクトルの線形結合であり、かつ各基底ベクトルが $\mathbf{u}_i = \mathbf{A}(\mathbf{v}_i / \sigma_i)$ によって実際に達成されるからである。
- $\mathbf{A}$ の**零空間**は $\operatorname{span}\{\mathbf{v}_{r+1}, \dots, \mathbf{v}_n\}$ である。同じ式より、$\mathbf{A}\mathbf{x} = \mathbf{0}$ が成り立つのは $\mathbf{v}_i^\top \mathbf{x} = 0$ がすべての $i \le r$ で成り立つとき、かつそのときに限る（$\mathbf{u}_i$ は線形独立であり、σ_i は正だからである）。これは $\mathbf{x}$ が $\mathbf{v}_1, \dots, \mathbf{v}_r$ に直交することを意味する。
- **行空間**（row space、$\mathbf{A}^\top$ の列空間）は $\operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ であり、**左零空間**（left null space、$\mathbf{A}^\top$ の零空間）は $\operatorname{span}\{\mathbf{u}_{r+1}, \dots, \mathbf{u}_m\}$ である。これは同じ 2 つの議論を $\mathbf{A}^\top = \mathbf{V} \Sigma^\top \mathbf{U}^\top$（単に $\mathbf{u}_i$ と $\mathbf{v}_i$ の役割を入れ替えたもの）に適用すれば導かれる。

これにより、あらゆる線形変換の見通しが驚くほど明快になる。入力空間は r 次元の行空間と $(n - r)$ 次元の零空間という直交する 2 つの部分空間に分解される。同様に、出力空間も r 次元の列空間と $(m - r)$ 次元の左零空間に分解される。行空間の上では、$\mathbf{A}$ は各 $\mathbf{v}_i$ を $\sigma_i \mathbf{u}_i$ へと写し、列空間への 1 対 1 の全単射を与える。一方、零空間の元はすべてゼロへと送られる。

最も単純な非自明の例でこれを見るため、33.2 節の特異行列 $\mathbf{B} = \left(\begin{smallmatrix} 2 & 1 \\ 4 & 2 \end{smallmatrix}\right)$ を考えよう。グラム行列 $\left(\begin{smallmatrix} 20 & 10 \\ 10 & 5 \end{smallmatrix}\right)$ の固有値は 25 と 0 であるため、$\sigma_1 = 5$、$\sigma_2 = 0$、$r = 1$ となる。4 つの部分空間は次の各ベクトルが張る直線である。

$$\mathbf{v}_1 = \tfrac{1}{\sqrt{5}}(2, 1), \quad \mathbf{v}_2 = \tfrac{1}{\sqrt{5}}(-1, 2), \quad \mathbf{u}_1 = \tfrac{1}{\sqrt{5}}(1, 2), \quad \mathbf{u}_2 = \tfrac{1}{\sqrt{5}}(-2, 1), \qquad \mathbf{B} = 5\, \mathbf{u}_1 \mathbf{v}_1^\top.$$

ここで列空間は直線 $y = 2x$（$\mathbf{u}_1$ の方向）であり、左零空間はそれに直交する $\mathbf{u}_2$ 方向の直線である。入力平面においては、零空間は $(1, -2)$ 方向の直線であり、行空間を張る行ベクトル $(2, 1)$ に直交している。Figure 33.26 に一般の幾何学的関係を示す。33.5.3.1 節で見るように、擬似逆行列はこの行空間から列空間への全単射をちょうど逆向きにたどる写像にほかならない。

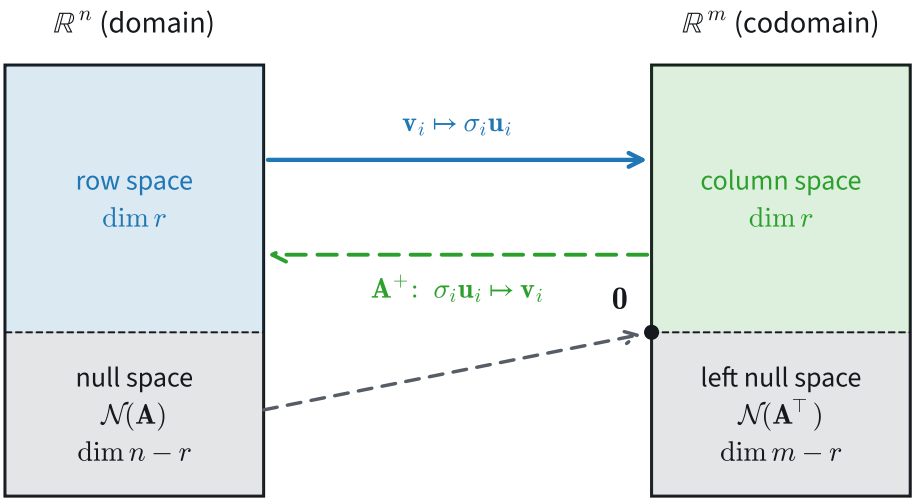


Figure 33.26: 4 つの基本部分空間。$\mathbf{A}$ は行空間 $\mathrm{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ を列空間 $\mathrm{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ の上へ $\mathbf{v}_i \mapsto \sigma_i \mathbf{u}_i$ によって 1 対 1 に写し、零空間をゼロへと写す。擬似逆行列 $\mathbf{A}^+$（破線）はこの写像を逆向きに $\sigma_i \mathbf{u}_i \mapsto \mathbf{v}_i$ とたどり、左零空間をゼロへと送る。

数値ランク。浮動小数点演算においては、数学的にランク落ちしている行列であっても、特異値が文字どおり厳密にゼロになるとは限らない。丸め誤差によって $\epsilon_{\text{mach}}\sigma_1$ 程度の微小な値が残るのが普通だからである。そのため実用上は、閾値を用いて数値ランク（numerical rank）を判定する。NumPy の `np.linalg.matrix_rank` は $\sigma_1 \max(m, n)\, \epsilon_{\text{mach}}$ を超える特異値の個数を数えており、この許容誤差は最大特異値と行列の次元の双方に応じて自然にスケールする。ランク 2 の 4×4 行列において、「ゼロ」であるはずの特異値が実際にどのような値になるか見てみよう。

```
A = np.random.standard_normal((4, 2)) @ np.random.standard_normal((2, 4))   # rank <= 2
s = np.linalg.svd(A, compute_uv=False)
tol = s[0] * max(A.shape) * np.finfo(float).eps
print('singular values:', np.array2string(s, formatter={'float_kind': '{:.1e}'.format}))
print(f'tolerance {tol:.1e};  exact zeros: {(s == 0).sum()};  '
      f'numerical rank: {np.linalg.matrix_rank(A)}')
```

```
singular values: [4.4e+00 2.2e+00 4.7e-17 0.0e+00]
tolerance 3.9e-15;  exact zeros: 1;  numerical rank: 2
```

2 つの特異値はオーダー 1 であるのに対し、残りの 2 つは純粋な丸めノイズであり、10^{-15} 以下と許容誤差を大きく下回っている。丸めの結果たまたま片方が厳密なゼロになることはあっても、両方がゼロになることはめったにない。したがって厳密なゼロを数えるとランクを誤判定してしまうが、閾値を用いれば真のランク 2 が正しく復元される。

33.5.2 低ランク近似

33.5.2.1 最良低ランク近似

ここで本節冒頭の猫の写真に戻ろう。どの 20 個のランク 1 成分を残すべきだろうか。ダイアド和 Equation 33.46 はすでに $\mathbf{A}$ の成分を重いものから軽いものへと並べているため、最初の k 項を残すのが自然な候補となる。こうして得られるのが**打ち切り SVD**（truncated SVD）である。

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^{\top}. \tag{33.51}$$

行列 $\mathbf{A}_k$ のランクは高々 k であり、これを保存するには k 個の特異値、k 本の長さ m の列、k 本の長さ n の列だけで足りるため、合計 $k(m+n+1)$ 個の数値で済む（元の mn 個の代わり）。400×500 の写真で $k = 20$ とすれば、200,000 個のうちわずか 18,020 個の数値である。すると直ちに 2 つの疑問が生じる。近似誤差はどれほど大きいか、そしてもっと巧妙なランク k 行列を作ればさらに良くできるのか、という問いである。

特異値で表される行列ノルム。近似誤差を測るには行列ノルムが必要である。1.3 節で導入したフロベニウスノルム $\|\mathbf{A}\|_F^2 = \sum_{ij} a_{ij}^2$ は、$\mathbf{A}$ の各列の長さの二乗和であり、それらは $\mathbf{A}^{\top}\mathbf{A}$ の対角成分に等しい。したがって $\|\mathbf{A}\|_F^2 = \operatorname{tr}(\mathbf{A}^{\top}\mathbf{A})$ となる。ここに SVD を代入し、トレースの巡回性（33.2 節）を使って $\mathbf{V}$ を反対側へ移すと次式が得られる。

$$\|\mathbf{A}\|_F^2 = \operatorname{tr}(\mathbf{V}\Sigma^{\top}\Sigma\mathbf{V}^{\top}) = \operatorname{tr}(\Sigma^{\top}\Sigma\,\mathbf{V}^{\top}\mathbf{V}) = \sum_{i} \sigma_i^2. \tag{33.52}$$

ゆえにフロベニウスノルムは特異値の二乗和の平方根であり、ここから左右どちらから直交行列を掛けてもフロベニウスノルムは決して変わらないことが直ちにわかる。せん断行列では $1+1+0+1 = 3 = \varphi^2 + \varphi^{-2}$ となる。一方、スペクトルノルムは単なる最大特異値 $\|\mathbf{A}\|_2 = \sigma_1$ である（Equation 33.48）。

両方のノルムが特異値で表されたところで、線形代数における中心的な近似定理を述べよう。

定理（Eckart–Young–Mirsky）。$\sigma_i = 0$ $(i > r)$ とおく。すべての $k \geq 0$ に対し、ランクが高々 k の行列で達成可能な最小のスペクトルノルム誤差は次式で与えられる。

$$\min_{\operatorname{rank}\mathbf{B} \leq k} \|\mathbf{A} - \mathbf{B}\|_2 = \sigma_{k+1}, \tag{33.53}$$

また、最小の二乗フロベニウス誤差は $\min_{\operatorname{rank}\mathbf{B}\leq k}\|\mathbf{A}-\mathbf{B}\|_F^2 = \sum_{i>k}\sigma_i^2$ である。いずれの最小値も、打ち切り SVD である $\mathbf{B} = \mathbf{A}_k$ において達成される。

要するに、どちらのノルムにおいても、いかなるランク k 行列も打ち切り SVD を上回ることはできない。最良のランク k 近似の誤差は、切り捨てた最初の特異値（スペクトルノルムの場合）、あるいは切り捨てたすべての特異値のエネルギーの総和（フロベニウスノルムの場合）に一致する。Eckart と Young がフロベニウスノルムの場合を証明し (Eckart and Young 1936)、後に Mirsky が左右両側からの直交行列の積に対して不変なすべてのノルムへと拡張した (Mirsky 1960)。まずスペクトルノルムの評価を証明しよう。フロベニウスノルムの評価は、同じ議論からそのまま導かれるもう 1 つの道具である Weyl の不等式を使えば明快に従う。

一般の証明を書く前に、$n = 3$、$k = 1$ の場合を思い描いてみよう。ランク 1 の任意の行列 $\mathbf{B}$ は入力のある平面全体をゼロに潰す一方、$\mathbf{A}$ の上位 2 本の右特異ベクトルは別の平面を張る。$\mathbb{R}^3$

において原点を通る 2 つの平面は、少なくとも 1 本の直線で交わらなければならない。その直線上では、$\mathbf{B}$ の出力はゼロであるのに対し、$\mathbf{A}$ はすべてのベクトルを少なくとも σ_2 倍に引き伸ばす。したがって $\mathbf{A} - \mathbf{B}$ は、ある単位ベクトルを少なくとも σ_2 倍に引き伸ばさざるを得ない（Figure 33.27）。以下の証明は、この次元の勘定を n 次元へそのまま持ち上げたものである。

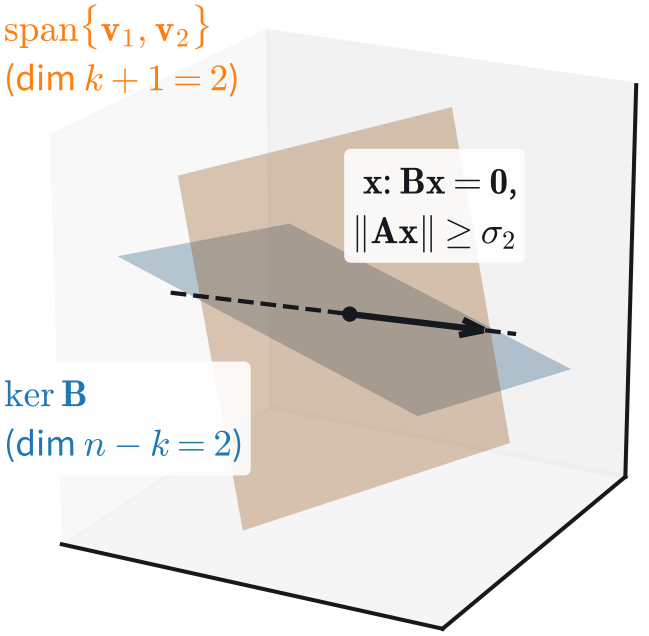


Figure 33.27: $n = 3$、$k = 1$ における Eckart–Young の背後にある次元の勘定。平面 $\ker \mathbf{B}$ と上位 2 本の右特異ベクトルが張る平面は直線で交わらなければならない。その直線上では $\mathbf{B}$ はゼロになる一方、$\mathbf{A}$ は少なくとも σ_2 倍に引き伸ばす。

証明（スペクトルノルム）。 第 1 に、打ち切り SVD は誤差 σ_{k+1} を達成する。残差 $\mathbf{A} - \mathbf{A}_k = \sum_{i>k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ はすでに最大特異値 σ_{k+1} を持つコンパクト SVD の形をしており、$\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}$ が Equation 33.48 より従う。同様に、Equation 33.52 から $\|\mathbf{A} - \mathbf{A}_k\|_F^2 = \sum_{i>k} \sigma_i^2$ も得られる。

第 2 に、ランクが高々 k のいかなる行列もこれを下回ることはできない。$k \geq r$ ならば誤差はゼロで証明すべきことはないため、$k < r$ と仮定し、$\operatorname{rank} \mathbf{B} \leq k$ とする。ランク・退化次数の定理（33.2 節）より、零空間 $\mathcal{X} = \ker \mathbf{B}$ の次元は少なくとも $n - k$ である。一方、張る空間 $\mathcal{V} = \operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_{k+1}\}$（上位 $k+1$ 本の右特異ベクトルによるもの）の次元は $k+1$ である。これら 2 つの次元の和は n を超えるため、2 つの部分空間は非ゼロベクトルを共有しなければならない。

$$\dim(\mathcal{X} \cap \mathcal{V}) = \dim \mathcal{X} + \dim \mathcal{V} - \dim(\mathcal{X} + \mathcal{V}) \geq (n-k) + (k+1) - n = 1,$$

ここで等号は 2 つの部分空間の次元公式であり、不等号は $\dim(\mathcal{X} + \mathcal{V}) \leq n$ を用いている（次元公式を確かめるには、$\mathcal{X} \cap \mathcal{V}$ の基底をそれぞれ $\mathcal{X}$ の基底と $\mathcal{V}$ の基底へ別々に延長すればよい。合わせたベクトル群は線形独立で $\mathcal{X} + \mathcal{V}$ を張り、その本数は $\dim \mathcal{X} + \dim \mathcal{V} - \dim(\mathcal{X} \cap \mathcal{V})$

となる)。そこで単位ベクトル $\mathbf{x} \in \mathcal{K} \cap \mathcal{V}$ を 1 つ選び、$\mathbf{x} = \sum_{i \le k+1} c_i \mathbf{v}_i$（$\sum_i c_i^2 = 1$）と展開する。すると $\mathbf{B}\mathbf{x} = \mathbf{0}$ であり、$\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$ と $\mathbf{u}_i$ の正規直交性を用いれば次式を得る。

$$\|\mathbf{A}\mathbf{x}\|^2 = \Big\| \sum_{i \le k+1} c_i \sigma_i \mathbf{u}_i \Big\|^2 = \sum_{i \le k+1} c_i^2 \sigma_i^2 \ge \sigma_{k+1}^2 \sum_{i \le k+1} c_i^2 = \sigma_{k+1}^2 .$$

ゆえに $\|\mathbf{A} - \mathbf{B}\|_2 \ge \|(\mathbf{A} - \mathbf{B})\mathbf{x}\| = \|\mathbf{A}\mathbf{x}\| \ge \sigma_{k+1}$ が成り立つ。$\blacksquare$

この幾何学的意味は一文で繰り返しておく価値がある。任意のランク k の競合行列 $\mathbf{B}$ は、上位 $k+1$ 本の右特異ベクトルが張る空間の中の少なくとも 1 方向をゼロに潰さなければならず、そして $\mathbf{A}$ はその空間内のあらゆる方向を少なくとも σ_{k+1} 倍に引き伸ばすのである。せん断行列 $\mathbf{T}$ で $k=1$ とすると、打ち切り $\mathbf{T}_1 = \varphi\, \mathbf{u}_1 \mathbf{v}_1^{\top} = \frac{1}{\sqrt{5}} \left(\begin{smallmatrix} \varphi & \varphi^2 \\ 1 & \varphi \end{smallmatrix} \right) \approx \left(\begin{smallmatrix} 0.724 & 1.171 \\ 0.447 & 0.724 \end{smallmatrix} \right)$ の誤差は両ノルムで $1/\varphi = 0.618$ となり（残差がランク 1 であるためスペクトルノルムとフロベニウスノルムが一致する）、いかなるランク 1 行列もこれより近づくことはできない。

フロベニウスノルムについて定理の証明を完結させるため、今証明したスペクトルノルムの評価から一般的な摂動不等式を先に導いておこう。

補題（特異値に対する Weyl の不等式）。同じ形状を持つ任意の 2 つの行列 $\mathbf{X}, \mathbf{Y}$ と、添字 $i, j \ge 1$ $(i + j - 1 \le \min(m,n))$ に対し、次式が成り立つ。

$$\sigma_{i+j-1}(\mathbf{X} + \mathbf{Y}) \le \sigma_i(\mathbf{X}) + \sigma_j(\mathbf{Y}). \tag{33.54}$$

証明。 $\mathbf{X}_{i-1}$ と $\mathbf{Y}_{j-1}$ を打ち切り SVD Equation 33.51 とする（$\mathbf{X}_0 = \mathbf{0}$ とおく）。その和 $\mathbf{X}_{i-1} + \mathbf{Y}_{j-1}$ のランクは高々 $i+j-2$ であるから、スペクトルノルムの下界を $\mathbf{X}+\mathbf{Y}$ に $k = i+j-2$ として適用すれば $\sigma_{i+j-1}(\mathbf{X}+\mathbf{Y}) \le \|\mathbf{X}+\mathbf{Y}-\mathbf{X}_{i-1}-\mathbf{Y}_{j-1}\|_2$ が得られる。三角不等式（triangle inequality）と Eckart–Young の達成可能性の議論より、右辺は高々 $\|\mathbf{X}-\mathbf{X}_{i-1}\|_2 + \|\mathbf{Y}-\mathbf{Y}_{j-1}\|_2 = \sigma_i(\mathbf{X}) + \sigma_j(\mathbf{Y})$ である。$\blacksquare$

言葉で言えば、$\mathbf{Y}$ を $\mathbf{X}$ に加えたとき、第 $(i+j-1)$ 特異値が第 i 特異値（$\mathbf{X}$ のもの）を上回る量は高々 $\sigma_j(\mathbf{Y})$ に抑えられる。$j = 1$ として不等式を両方向に適用すれば、行列に $\mathbf{Y}$ の摂動を加えてもいかなる特異値も $\|\mathbf{Y}\|_2$ を超えて動かないことがわかる (Horn and Johnson 2012)。詳細は演習問題 2 で扱う。

証明（フロベニウスノルム）。 Weyl の補題を $\mathbf{X} = \mathbf{A} - \mathbf{B}$、$\mathbf{Y} = \mathbf{B}$、$j = k+1$ として適用する。$\operatorname{rank} \mathbf{B} \le k$ より $\sigma_{k+1}(\mathbf{B}) = 0$ であるから、$\sigma_{k+i}(\mathbf{A}) \le \sigma_i(\mathbf{A} - \mathbf{B})$ がすべての $i \ge 1$ に対して成り立つ。両辺を二乗して i について和をとり、Equation 33.52 を適用すると次式を得る。

$$\|\mathbf{A} - \mathbf{B}\|_F^2 = \sum_{i \ge 1} \sigma_i(\mathbf{A} - \mathbf{B})^2 \ge \sum_{i \ge 1} \sigma_{k+i}(\mathbf{A})^2 = \sum_{j > k} \sigma_j(\mathbf{A})^2 ,$$

そしてこの下界を打ち切り $\mathbf{A}_k$ が達成する。$\blacksquare$

最小点の一意性。 2 つのノルムの間には 1 つ微妙な違いがある。スペクトルノルムでは、最良ランク k 近似は $k \ge 1$ かつ $\sigma_{k+1} > 0$ である限り一意になることは決してない。たとえば $\mathbf{A} = \operatorname{diag}(3,2,1)$、$k=1$ のとき、行列 $\mathbf{B} = \operatorname{diag}(3+t, 0, 0)$ $(|t| \le 1)$ はすべて $\|\mathbf{A} - \mathbf{B}\|_2 = \max(|t|, 2, 1) = 2 = \sigma_2$ を与える。対照的にフロベニウスノルムでは、$\mathbf{A}_k$ は $\sigma_k > \sigma_{k+1}$ または $\sigma_{k+1} = 0$（この場合は $\mathbf{A}_k = \mathbf{A}$）である限り**一意**な最小点（minimizer）となる (Golub and Van Loan 1996)。次に見る主成分分析がフロベニウスノルムで定式化される理由の 1 つが、この一意性にある。

それでは、400×500 の猫の写真でこの定理を検証してみよう。Eckart–Young は各打ち切りのスペクトル誤差を σ_{k+1} に定めるが、実際の画像の特異値がどれほど速く減衰するかや、素朴なランク k の競合手法がどれほど劣るかまでは教えてくれない。以下のセルでは `img/cat1.jpg` を読み込み、赤・緑・青に対する標準的な輝度重みを使ってグレースケールへ変換し、`np.einsum` を用いて打ち切り近似を構築する（この関数は入力を掛け合わせ、矢印の後に省かれたすべての添字について和をとる。第 i 行（`Vt` の行）は $\mathbf{v}_i^\top$ であるため、`'i,mi,in->mn'` でダイアド和が計算される）。

```
img = d2l.plt.imread('../img/cat1.jpg') @ np.array([0.299, 0.587, 0.114])  # 400 x 500
U, s, Vt = np.linalg.svd(img, full_matrices=False)
m, n = img.shape
fro = np.linalg.norm(img)
energy = np.cumsum(s**2) / (s**2).sum()                       # energy kept at each rank
slope = np.polyfit(np.log(np.arange(1, 101)), np.log(s[:100]), 1)[0]
print(f'{m} x {n} = {m * n} numbers;  slope of log sigma_i on log i: {slope:.2f}')
print(f'energy kept at rank 1: {energy[0]:.3f}, at rank 4: {energy[3]:.3f}')
for k in [1, 5, 20, 50]:
    A_k = np.einsum('i,mi,in->mn', s[:k], U[:, :k], Vt[:k])    # first k dyads
    print(f'k={k:2d}: ||A-A_k||_2={np.linalg.norm(img - A_k, 2):6.0f}  '
          f'sigma_{k + 1}={s[k]:6.0f}  rel. err {np.linalg.norm(img - A_k) / fro:.3f}  '
          f'stores {k * (m + n + 1) / (m * n):.1%}')
cols = np.argsort(-np.linalg.norm(img, axis=0))[:20]          # 20 largest columns
C = img[:, cols]
proj = C @ np.linalg.pinv(C) @ img
print(f'projection onto the 20 largest columns: rel. err {np.linalg.norm(img - proj) /
 ↪ fro:.3f}')
```

```
400 x 500 = 200000 numbers;  slope of log sigma_i on log i: -1.11
energy kept at rank 1: 0.886, at rank 4: 0.955
k= 1: ||A-A_k||_2=  9234  sigma_2=  9234  rel. err 0.337  stores 0.5%
k= 5: ||A-A_k||_2=  4048  sigma_6=  4048  rel. err 0.191  stores 2.3%
k=20: ||A-A_k||_2=   874  sigma_21=   874  rel. err 0.080  stores 9.0%
k=50: ||A-A_k||_2=   342  sigma_51=   342  rel. err 0.045  stores 22.5%
projection onto the 20 largest columns: rel. err 0.310
```

保証どおり、すべてのランクでスペクトル誤差は σ_{k+1} と一致し、ランク 20 ではわずか 9% の数値を保持するだけで相対誤差 8% に収まっている。Figure 33.28 を見ると、猫はランク 1 では明るいぼやけた塊にすぎないが、ランク 5 では 4 本の足を持つシルエットとして識別でき、ランク 20 になると仰向けに寝転ぶ猫だと紛れもなくわかる。左パネルを見れば、なぜこれほど少数の成分で足りるのかが明らかになる。最初の 100 個の添字にわたって、特異値はおよそべき乗則 $\sigma_i \propto i^{-1.1}$ に従って減衰しているのである。

エネルギー比。フロベニウス誤差の公式は、k を選ぶための実用的な指針を与える。二乗「エネルギー」の総和 $\sum_i \sigma_i^2$ のうち、ランク k の打ち切りによって保持される割合は次式で表される。

$$\frac{\sum_{i \le k} \sigma_i^2}{\sum_i \sigma_i^2} = 1 - \frac{\|\mathbf{A} - \mathbf{A}_k\|_F^2}{\|\mathbf{A}\|_F^2}, \tag{33.55}$$

すなわち、相対フロベニウス誤差は 1 からエネルギー比を引いた値の平方根になる。したがってエネルギーの 95% を保持することは相対誤差 $\sqrt{0.05} \approx 22\%$ に対応し、99% を保持すれば誤差は 10% まで下がる。ただし、この比率の解釈には注意が必要である。上の写真では、定数画像がランク 1 であり（すべての行が同一）、かつ写真の平均輝度が大きいため、最初のダイアドが主に

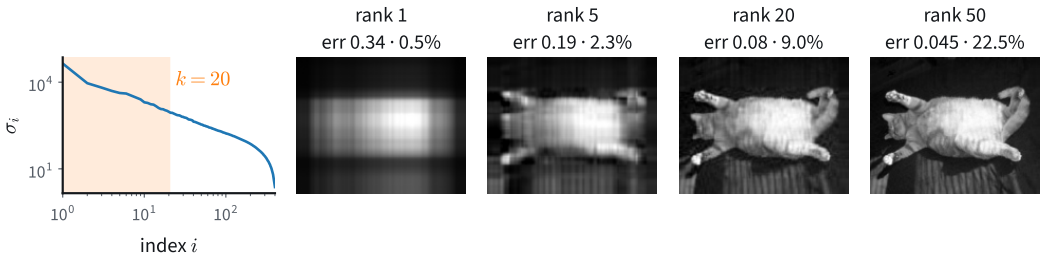


Figure 33.28: 400×500 の写真における Eckart–Young。左：特異値はおよそべき乗則に従って減衰する（両対数軸。網掛け帯はランク 20 のパネルで残した 20 個を示す）。右：各打ち切り $\mathbf{A}_k$ と、その相対フロベニウス誤差および元の数値数に対する保持割合 $k(m+n+1)/mn$。

その平均を捉えることになり、σ_1^2 だけでエネルギーの 89% を占めてしまう。そのままではランク 4 ですでにエネルギーの 95% を保持する（相対誤差 21%）が、ピクセル値の平均をあらかじめ差し引くと、95% に達するにはランク 8 が必要になる（4 ではなくなることは 33.5.4.3 節で確認する）。常に σ_1 が非ゼロの平均に支配されていないか確認するか、主成分分析と同様にまず平均を差し引くべきである。

素朴なランク 20 の手法と比べて Eckart–Young がどれほど優れているかを見るため、セルの末尾では写真からノルムが最大の実列を 20 本選び、すべての列をそれらが張る空間へ射影している（行列自身の行と列の部分集合で近似する CUR 分解の背後にある考え方である）。その相対誤差は 0.31 であり、打ち切り SVD の 0.08 と比べて 4 倍近く悪い。

ノイズ除去としての打ち切り。 SVD を打ち切る理由は圧縮だけではなく、ノイズ除去（denoising）も重要な用途である。観測された $m \times n$ 行列（$m \le n$）が低ランクの信号にランダムノイズが加わった $\mathbf{A} = \mathbf{B} + \mathbf{N}$ であり、$\mathbf{N}$ の各成分が平均 0、標準偏差 σ_{noise} で独立だとする。33.3.2.4 節のマルチェンコ・パスツール則を $\mathbf{N}\mathbf{N}^{\top}/n$ に適用すると、大きなノイズ行列の特異値は区間 $[(\sqrt{n}-\sqrt{m})\sigma_{\text{noise}}, (\sqrt{n}+\sqrt{m})\sigma_{\text{noise}}]$ の内部に集中する。特異値がこの上端を大きく超える信号成分はノイズの塊（バルク）からくっきりと浮かび上がる一方、弱い成分はその中に埋もれて消える。ノイズレベル（noise level）が既知の正方行列に対し、Gavish と Donoho は漸近的に最適なハード閾値が $\frac{4}{\sqrt{3}}\sqrt{n}\,\sigma_{\text{noise}} \approx 2.309\sqrt{n}\,\sigma_{\text{noise}}$ であり、ノイズの上端 $2\sqrt{n}\,\sigma_{\text{noise}}$ よりわずかに上に位置することを証明した (Gavish and Donoho 2014)。演習問題 8 で定数 $4/\sqrt{3}$ を導出する。

33.5.2.2 主成分分析

各データ例を行として並べたデータセット $\mathbf{X} \in \mathbb{R}^{n \times d}$（データ点数 n、特徴量数 d）が与えられたとしよう。データの変動の大半がごく少数の方向に集中していることは多い。ここで 2 つの自然な問いが生じる。データの最大の分散を捉える単位方向はどれか、そして二乗誤差を最小にする k 次元アフィン部分空間はどれか、という問いである。驚くべきことに、両者の答えは完全に一致し、中心化したデータの打ち切り SVD そのものになる。

中心化。 まず任意の k 次元部分空間とその直交射影行列 $\mathbf{P}$ を固定し、どのオフセット $\mathbf{c}$ が総二乗再構成誤差（reconstruction error）$\sum_i \|(\mathbf{I}-\mathbf{P})(\mathbf{x}_i - \mathbf{c})\|^2$ を最小化するかを考えよう。この目的関数は $\mathbf{c}$ に関する凸二次関数であり、その勾配 $-2(\mathbf{I}-\mathbf{P})\sum_i(\mathbf{x}_i-\mathbf{c})$ は標本平均 $\mathbf{c} = \bar{\mathbf{x}}$ において $\mathbf{P}$ によらずゼロになる。したがって最良のアフィン部分空間は常に平均を通るため、まずデータの**中心化**（centering）から始める。すなわち $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{\top}$ とおく。その経験共分散は対称半正定値行列 $\mathbf{C} = \frac{1}{n}\tilde{\mathbf{X}}^{\top}\tilde{\mathbf{X}}$ である（不偏分散の係数 $\frac{1}{n-1}$ を使ってもすべての固有値が同じ倍率でスケールされるだけであり、主方向もそれらが説明する分散の割合も変わらない）。

命題（Rayleigh による PCA）。 すべての単位ベクトル $\mathbf{w}$ にわたって、射影後の分散 $\frac{1}{n}\|\tilde{\mathbf{X}}\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{C}\mathbf{w}$ を最大化するのは $\mathbf{C}$ の最大固有ベクトル、すなわち最大右特異ベクトル $\mathbf{v}_1$（$\tilde{\mathbf{X}}$ のもの）であり、最大分散は σ_1^2/n である。

証明。 中心化した点を $\mathbf{w}$ の上へ射影するとスカラー座標 $\tilde{\mathbf{x}}^\top\mathbf{w}$ が得られ、その経験分散は $\frac{1}{n}\sum_i (\tilde{\mathbf{x}}_i^\top \mathbf{w})^2 = \frac{1}{n}\|\tilde{\mathbf{X}}\mathbf{w}\|^2 = \mathbf{w}^\top\mathbf{C}\mathbf{w}$ となる。33.3.2.3 節の Rayleigh の命題より、この二次形式は $\mathbf{C}$ の最大固有ベクトルで最大値 $\lambda_1(\mathbf{C})$ をとる。Equation 33.49 より $\mathbf{C} = \frac{1}{n}\mathbf{V}\Sigma^\top\Sigma\mathbf{V}^\top$ であるから、$\mathbf{C}$ の固有ベクトルは右特異ベクトル $\mathbf{v}_i$（$\tilde{\mathbf{X}}$ のもの）であり、固有値は σ_i^2/n である。■

$\mathbf{v}_1$ に直交する単位ベクトルの中では分散は $\mathbf{v}_2$ で最大化され、帰納的に続ければ上位 k 個の**主方向**（principal direction）$\mathbf{v}_1, \dots, \mathbf{v}_k$ が得られ、第 i 成分が分散 σ_i^2/n を説明する。ここで再構成誤差を確認しよう。中心化した各点を $\operatorname{span}\{\mathbf{v}_1,\dots,\mathbf{v}_k\}$ へ射影する操作は、$\tilde{\mathbf{X}}$ に右から $\mathbf{V}_k\mathbf{V}_k^\top$ を掛けることに対応する。$\mathbf{V}^\top\mathbf{V}_k$ は単位行列の最初の k 列を取り出すため、次式が成り立つ。

$$\tilde{\mathbf{X}}\mathbf{V}_k\mathbf{V}_k^\top = \mathbf{U}\Sigma\mathbf{V}^\top\mathbf{V}_k\mathbf{V}_k^\top = \mathbf{U}_k\Sigma_k\mathbf{V}_k^\top = \tilde{\mathbf{X}}_k.$$

したがって、上位 k 個の主方向への射影は、中心化データのランク k 打ち切り SVD をとることと完全に一致する。フロベニウスノルムにおける Eckart–Young の定理より、他のいかなる射影も、さらには他のいかなるランク k 行列も、より小さい二乗誤差で中心化データ点を再構成することはできない。切り捨てられた分散 $\sum_{i>k}\sigma_i^2/n$ は平均二乗再構成誤差に等しく、説明分散比（explained-variance ratio）はエネルギー比 Equation 33.55（$\tilde{\mathbf{X}}$ に対するもの）そのものである。

Figure 33.29 に、300 個の点からなる点群（標準偏差 3 と 0.8 を持ち、$\pi/5 = 36^\circ$ 回転した軸に沿う分布）と、その 2 本の主軸を示す。母集団においては、第 1 方向が分散の $9/9.64 \approx 93.4\%$ を占め、長軸に一致する。以下のセルでは、300 点の有限標本からこれらの値がどれほど正確に復元されるかを確認する。

```
theta = np.pi / 5
rot = np.array([[np.cos(theta), -np.sin(theta)], [np.sin(theta), np.cos(theta)]])
X = np.random.standard_normal((300, 2)) * np.array([3.0, 0.8]) @ rot.T   # the cloud
Xc = X - X.mean(0)
U, s, Vt = np.linalg.svd(Xc, full_matrices=False)
X1 = np.outer(U[:, 0] * s[0], Vt[0])                                    # rank-1 reconstruction
print('explained-variance ratio:', (s**2 / (s**2).sum()).round(3), ' (population 0.934,
 ↪  0.066)')
print('angle of v_1 to the long axis:',
      np.degrees(np.arccos(abs(Vt[0] @ rot[:, 0]))).round(1), 'degrees')
print('mean squared rank-1 error:', (((Xc - X1)**2).sum() / 300).round(4),
      '  sigma_2^2 / n:', (s[1]**2 / 300).round(4))
```

```
explained-variance ratio: [0.923 0.077]  (population 0.934, 0.066)
angle of v_1 to the long axis: 0.6 degrees
mean squared rank-1 error: 0.7111   sigma_2^2 / n: 0.7111
```

乱数抽出を繰り返すと、第 1 主方向は分散のおよそ 91% から 95% を説明し、真の長軸から数度以内に収まる。しかしデータの中心化を忘れると、平均のオフセットが点群の広がりを超えた途端に、第 1 特異ベクトルは点群の平均の方向へと振れてしまい、点群が**どのように**変動しているかではなく**どこに**位置しているかを表すようになってしまう。演習問題 3 でこの影響を定量化する。

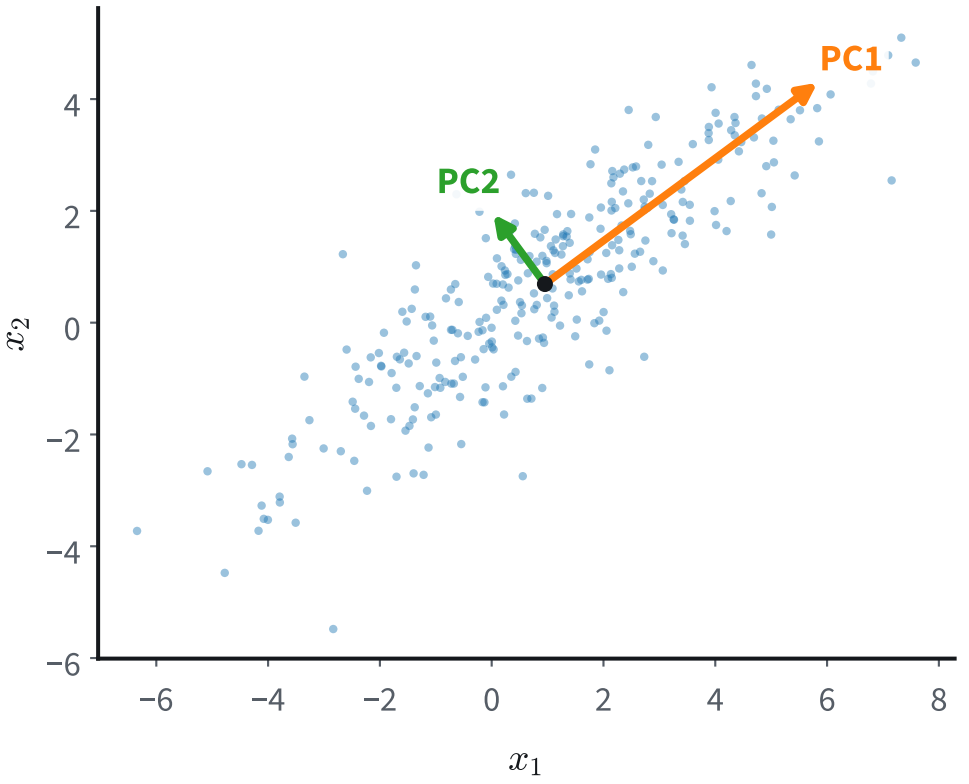


Figure 33.29: 300 個の点からなる点群（中心 $(1, 0.6)$、標準偏差 3 と 0.8、回転角 36° の軸に沿う分布）と、平均から描いた主軸（各軸の長さは標準偏差の 2 倍である $2\sigma_i/\sqrt{n}$）。第 1 主軸は最大分散の方向を指す。

33.5.3 SVD による連立一次方程式の解法

33.5.3.1 擬似逆行列と最小二乗法

未知数より方程式が多い連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ は一般に厳密解を持たず、逆に未知数より方程式が少ない連立一次方程式は無数の解を持つ。どちらの状況でも、33.1 節で導入した最小二乗法の解を求める（ここでは目的ベクトルを $\mathbf{b}$ と書き、$\mathbf{y}$ の代わりに用いる）。

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|^2,$$

さらに、複数の $\mathbf{x}$ が同じ最小残差を達成するときは、その中でノルムが最小のものを選ぶ。

まず最も単純な劣決定（underdetermined）の例として、2 つの未知数に対する 1 本の方程式 $x_1 + x_2 = 2$ を考えよう。解は直線全体をなし、その上のどこでも残差は 0 である。直線上のどの点を選ぶべきだろうか。ここでは $\mathbf{A} = \begin{pmatrix} 1 & 1 \end{pmatrix}$ であり、非ゼロ特異値は $\sigma_1 = \sqrt{2}$ の 1 つだけで、右特異ベクトルは $\mathbf{v}_1 = (1,1)/\sqrt{2}$、左特異ベクトルは $u_1 = 1$ となる。この方程式は $\mathbf{x}$ の $\mathbf{v}_1$ 方向の座標を $\sqrt{2}$ に固定する一方、零空間方向 $\mathbf{v}_2 = (1,-1)/\sqrt{2}$ に沿う座標は残差に一切影響しない。その自由な座標を 0 に設定すると、原点から直線へ下ろした垂線の足 $(1,1)$（ノルム $\sqrt{2}$）が得られ、他のあらゆる解 $(1+t, 1-t)$ は真に大きい二乗ノルム $2+2t^2$ を持つ（Figure 33.30）。

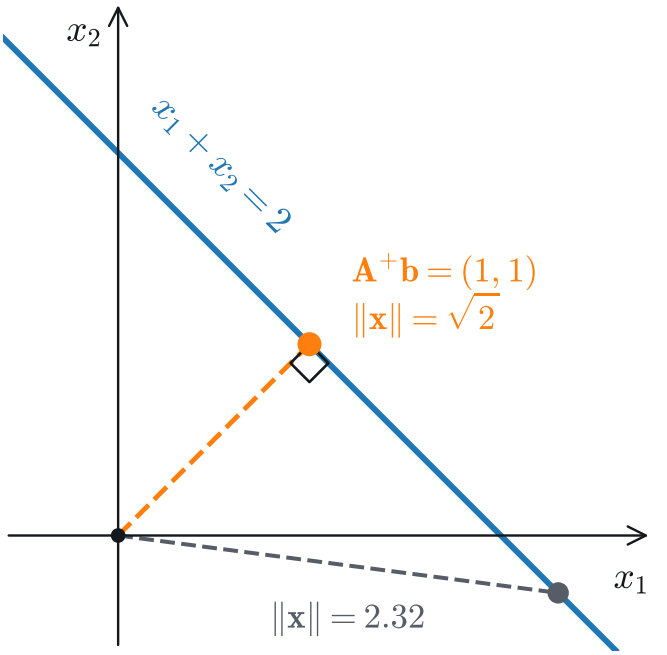


Figure 33.30: 方程式 1 本・未知数 2 個の最小二乗法。直線上のすべての点が $x_1 + x_2 = 2$ を満たし、擬似逆行列は原点に最も近い点、すなわち垂線の足を返す。

一般の規則は、単にすべての非ゼロ特異値の逆数をとり、直交因子を転置するだけである。$\mathbf{A}$ の**ムーア・ペンローズ擬似逆行列**（Moore–Penrose pseudoinverse）は次式で定義される。

$$\mathbf{A}^{+} = \mathbf{V}\Sigma^{+}\mathbf{U}^{\top}, \qquad \Sigma^{+}_{ii} = \begin{cases} 1/\sigma_i & \sigma_i > 0, \\ 0 & \sigma_i = 0, \end{cases} \tag{33.56}$$

ここで Σ^{+} の形状は $n \times m$ であり、Σ を転置した形状になっている。幾何学的には、$\mathbf{A}^{+}$ は元に戻せるすべての伸縮を逆転させ、左零空間をゼロへと送る。せん断行列に対しては $\mathbf{T}^{+} = \mathbf{T}^{-1} = \left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \end{smallmatrix}\right)$ となり、$\mathbf{A} = \begin{pmatrix} 1 & 1 \end{pmatrix}$ に対しては $\mathbf{A}^{+} = \mathbf{v}_1 \sigma_1^{-1} u_1 = \left(\tfrac{1}{2}, \tfrac{1}{2}\right)^{\top}$ となるため、$\mathbf{A}^{+} \cdot 2 = (1, 1)$ が得られる。

命題（最小ノルム最小二乗法）。$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ のすべての最小点の中で、ノルムが最小となる一意の解は $\hat{\mathbf{x}} = \mathbf{A}^{+}\mathbf{b}$ である。

証明。 $\mathbf{y} = \mathbf{V}^{\top}\mathbf{x}$、$\mathbf{c} = \mathbf{U}^{\top}\mathbf{b}$ とおいて特異基底へ変数変換する。$\mathbf{U}$ は直交行列でありユークリッドノルムを保つため、次式が成り立つ。

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 = \|\mathbf{U}^{\top}(\mathbf{A}\mathbf{x} - \mathbf{b})\|^2 = \|\Sigma\mathbf{y} - \mathbf{c}\|^2 = \sum_{i \le r} (\sigma_i y_i - c_i)^2 + \sum_{i > r} c_i^2 .$$

特異基底へ回転することで、連立方程式は互いに独立なスカラー方程式へと分離される。第 1 の和は、$y_i = c_i/\sigma_i$（$i \le r$）とおくことで項ごとに最小化される。第 2 の和は $\mathbf{b}$ の列空間外の成分を表す固定定数であり、$\mathbf{x}$ をどう選んでも減らすことはできない。一方、座標 y_i（$i > r$）は残差に一切現れず、$\|\mathbf{x}\| = \|\mathbf{y}\|$ であるから、それらを 0 に設定することで最小ノルムの一意な最小点が得られる。この選択は $\mathbf{y} = \Sigma^{+}\mathbf{c}$ であり、元の変数へ戻せば $\hat{\mathbf{x}} = \mathbf{V}\Sigma^{+}\mathbf{U}^{\top}\mathbf{b} = \mathbf{A}^{+}\mathbf{b}$ となる。$\blacksquare$

特異基底で見ると、残差の最小化とノルムの最小化は完全に互いに素な座標群（$i \le r$ と $i > r$）を支配しており、両方の目的が同時に達成される。$\mathbf{A}$ が正方かつ可逆のときは、すべての σ_i が正となり、$\mathbf{A}^{+} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{\top} = \mathbf{A}^{-1}$ となる。しかし同じ式は、最小二乗法の主な危険性もあらわにする。微小な σ_i で割ることにより、$\mathbf{b}$ に含まれる $\mathbf{u}_i$ 方向のノイズが $1/\sigma_i$ 倍に増幅されてしまうのである。

この不安定な方向を抑え込む標準的な処方箋が 2 つある。**リッジ回帰**（ridge regression）は $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \lambda\|\mathbf{x}\|^2$（$\lambda > 0$）を最小化し、$\mathbf{x}_\lambda = (\mathbf{A}^{\top}\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^{\top}\mathbf{b}$ を与える。一方、**打ち切り擬似逆行列**（truncated pseudoinverse）による解 $\mathbf{x}_k$ は、最初の k 個の特異値項だけを $\mathbf{A}^{+}\mathbf{b}$ から残す。SVD を代入し、$\mathbf{A}^{\top}\mathbf{A} + \lambda\mathbf{I} = \mathbf{V}(\Sigma^{\top}\Sigma + \lambda\mathbf{I})\mathbf{V}^{\top}$ に注意すると、両方の解は見通しのよいフィルタ形式をとる。

$$\mathbf{x}_\lambda = \sum_{i \le r} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \, \frac{\mathbf{u}_i^{\top}\mathbf{b}}{\sigma_i} \, \mathbf{v}_i, \qquad \mathbf{x}_k = \sum_{i \le k} \frac{\mathbf{u}_i^{\top}\mathbf{b}}{\sigma_i} \, \mathbf{v}_i . \tag{33.57}$$

どちらも正則化なしの擬似逆行列の項 $\mathbf{u}_i^{\top}\mathbf{b}/\sigma_i$ から出発する。リッジ回帰は第 i 成分に滑らかなフィルタ係数 $\sigma_i^2/(\sigma_i^2 + \lambda)$ を掛け、この係数は $\sigma_i^2 \gg \lambda$ のとき 1 付近にとどまり、$\sigma_i^2 \ll \lambda$ のとき 0 へと縮む。これに対し、打ち切りは 1 か 0 かの急峻な階段関数を適用する。たとえば $\lambda = 1$ のとき、特異値 3、1、0.3 に対するフィルタ係数はそれぞれ 0.9、0.5、0.083 となる。二乗損失で訓練される線形モデルにおいて、重み減衰（weight decay、2.7 節）はまさにリッジ回帰そのものであり、これらのフィルタ係数はどの方向が減衰されるかを正確に示している。

正規方程式。 古典的な教科書における最小二乗法の導出では、正規方程式（normal equations）$\mathbf{A}^{\top}\mathbf{A}\mathbf{x} = \mathbf{A}^{\top}\mathbf{b}$ を解く。任意の $\mathbf{A}$ に対してその解は最小二乗法の最小点となり（$\mathbf{A}^{\top}(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0}$ は残差が列空間に直交することを表すため）、$\mathbf{A}$ が列フルランクであれば解は一意に定まる。しかし数値計算上、$\mathbf{A}^{\top}\mathbf{A}$ を直接作るのは得策ではない。条件数が二乗されてしまうからである

(33.5.3.2 節)。そのため現代のライブラリは、代わりに SVD や QR 分解を用いて最小二乗問題を解く。

過剰決定（overdetermined）系と劣決定系を 1 つずつ使って擬似逆行列を試してみよう。1 つ目は直線 $b \approx x_1 + x_2 t$ を 4 つの観測値 $\mathbf{b} = (6, 5, 7, 10)$（$t = 1, 2, 3, 4$）に当てはめる問題であり、2 つ目は先ほどの 1 本の方程式 $x_1 + x_2 = 2$ である。

```
A = np.array([[1., 1.], [1., 2.], [1., 3.], [1., 4.]])     # 4 equations, 2 unknowns
b = np.array([6., 5., 7., 10.])
x_pinv = np.linalg.pinv(A) @ b
x_lstsq = np.linalg.lstsq(A, b, rcond=None)[0]
print('overdetermined: pinv', x_pinv.round(4), ' lstsq', x_lstsq.round(4),
      ' residual', np.linalg.norm(A @ x_pinv - b).round(4))
A1 = np.array([[1., 1.]])                                   # 1 equation: x1 + x2 = 2
x1 = np.linalg.pinv(A1) @ np.array([2.])
print('underdetermined: pinv gives', x1.round(4), ' norm', np.linalg.norm(x1).round(4))
```

```
overdetermined: pinv [3.5 1.4]  lstsq [3.5 1.4]  residual 2.0494
underdetermined: pinv gives [1. 1.]  norm 1.4142
```

擬似逆行列と `np.linalg.lstsq` は当てはめ直線 $3.5 + 1.4t$（残差 $\sqrt{4.2} \approx 2.049$）で一致し、1 本の方程式に対しては擬似逆行列は垂線の足 $(1, 1)$ を返す。

33.5.3.2 条件数

実際の計算では、連立一次方程式の各成分はノイズを含む測定値や有限精度の丸めから生じる。計算された解を信頼する前に、データのわずかな変化が答えをどれほど変え得るかを知っておく必要がある。単純な 2×2 の系を見るだけで、その増幅がいかに劇的になり得るかがわかる。$\mathbf{K}\mathbf{x} = \mathbf{b}$ において $\mathbf{K} = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1.0001 \end{smallmatrix}\right)$、$\mathbf{b} = (2, 2.0001)$ としよう。厳密解は $\mathbf{x} = (1, 1)$ である。ここで b_2 をほんの 10^{-4} だけ動かして 2.0002 にすると、解は一気に $(0, 2)$ まで飛び跳ねる。データにおける 10^5 分の 3 程度の摂動が、答えをまったく別物に変えてしまったのである。その理由は $\mathbf{K}$ の特異値（およそ 2.00005 と 5×10^{-5}）を見ればわかる。

行列 $\mathbf{A}$（ランク r）の**条件数**とは、最大の特異値と最小の非ゼロ特異値の比である。

$$\kappa(\mathbf{A}) = \frac{\sigma_1}{\sigma_r}. \tag{33.58}$$

幾何学的には、$\kappa(\mathbf{A})$ は $\mathbf{A}$ が異なる方向をどれほど不均一に伸縮するかを測っている。$\mathbf{K}$ では約 4×10^4 に達するのに対し、せん断行列ではわずか $\varphi^2 \approx 2.618$ である。$\mathbf{A}$ が対称正定値（positive definite）行列のときは特異値が固有値と一致するため $\kappa = \lambda_{\max}/\lambda_{\min}$ となり、33.3.2.3 節で見た二次関数のボウルに対する条件数が復元される。次の命題は、連立一次方程式を解く際の相対誤差に対する厳密な最悪増幅率が $\kappa(\mathbf{A})$ であることを示している。

命題（誤差増幅）。 **$\mathbf{A}$ を正方かつ可逆とし、$\mathbf{A}\mathbf{x} = \mathbf{b}$（$\mathbf{b} \neq \mathbf{0}$）とする。摂動 $\delta\mathbf{b}$ によって解が $\mathbf{x} + \delta\mathbf{x}$ に変化したとき、次式が成り立つ。**

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \kappa(\mathbf{A}) \, \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}, \tag{33.59}$$

さらに、任意の $\mathbf{A}$ に対して等号を達成する $\mathbf{b}$ と $\delta\mathbf{b}$ が存在する。

証明。 $\mathbf{A}\,\delta\mathbf{x} = \delta\mathbf{b}$ より $\delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b}$ である。$\mathbf{A}^{-1} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{\top}$ の最大特異値は $1/\sigma_n$ であるから、Equation 33.48 より $\|\delta\mathbf{x}\| \le \sigma_n^{-1}\|\delta\mathbf{b}\|$ が成り立つ。同様に、$\mathbf{b} = \mathbf{A}\mathbf{x}$ より $\|\mathbf{b}\| \le \sigma_1\|\mathbf{x}\|$ が得られる。これら 2 つの不等式を組み合わせると次式を得る。

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \le \frac{\sigma_1}{\sigma_n}\,\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} = \kappa(\mathbf{A})\,\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

両方の評価で同時に等号が成り立つのは、$\mathbf{b} = \mathbf{u}_1$（単位長さの右辺に対して $\mathbf{x} = \sigma_1^{-1}\mathbf{v}_1$ を最小の長さにするとき）かつ $\delta\mathbf{b} = \varepsilon\mathbf{u}_n$（$\delta\mathbf{x} = \varepsilon\sigma_n^{-1}\mathbf{v}_n$ を大きさ ε の摂動に対して最大の長さにするとき）である。$\blacksquare$

したがって、データの相対誤差は高々 κ 倍に増幅される。$\kappa = 10^{4}$ の場合、$\mathbf{b}$ の相対誤差 10^{-8} が $\mathbf{x}$ では 10^{-4} まで膨らみ得るため、有効精度が 4 桁失われる。先ほどの 2×2 の例では、$\mathbf{b}$ の相対変化量は $10^{-4}/\|\mathbf{b}\| \approx 3.5\times 10^{-5}$ であるため、上界は $\mathbf{x}$ の相対変化として 1.41 まで許容し、実際の変化量は 1.0 となっている。これは摂動 $(0, 10^{-4})$ が長さの $1/\sqrt{2}$ を最も増幅される方向 $\mathbf{u}_2 \approx (1,-1)/\sqrt{2}$ に持つ一方、$\mathbf{b}$ がほぼ真っ直ぐ $\mathbf{u}_1$ の方向を向いているためである。

理想的な極限にある直交行列では、すべての特異値が 1 に等しいため $\kappa = 1$ となり、回転と鏡映は誤差を決して増幅しない。直交因子 $\mathbf{U}$ と $\mathbf{V}$ を用いた計算が数値的に安全である理由も、直交初期化された重み行列があらゆる入力ベクトルのノルムを保つ理由もここにある。その対極として、グラム行列を作ると条件数は二乗される。

$$\kappa(\mathbf{A}^{\top}\mathbf{A}) = \kappa(\mathbf{A})^{2},$$

これは Equation 33.49 がそのまま $\mathbf{A}^{\top}\mathbf{A}$ の SVD になっており、その特異値が σ_i^2 となるからである（任意の形状・任意のランクの行列 $\mathbf{A}$ で成り立つ）。せん断行列では、κ は $\varphi^2 \approx 2.62$ から $\varphi^4 \approx 6.85$ へと跳ね上がる。以下のセルでは、2×2 の例を検証するとともに、先ほどの直線当てはめ行列について $\kappa(\mathbf{A})$ と $\kappa(\mathbf{A}^{\top}\mathbf{A})$ を比較する。

```
K = np.array([[1., 1.], [1., 1.0001]])
b = np.array([2., 2.0001])
x = np.linalg.solve(K, b)
x2 = np.linalg.solve(K, b + np.array([0., 1e-4]))
print('x =', x.round(4), '  after changing b_2 by 1e-4:', x2.round(4) + 0.)
kappa = np.linalg.cond(K)
print(f'kappa(K) = {kappa:.0f};  bound {kappa * 1e-4 / np.linalg.norm(b):.2f};  '
      f'realized {np.linalg.norm(x2 - x) / np.linalg.norm(x):.2f}')
A = np.array([[1., 1.], [1., 2.], [1., 3.], [1., 4.]])
print(f'kappa(A) = {np.linalg.cond(A):.3f};  kappa(A^T A) = {np.linalg.cond(A.T @ A):.2f};  '
      f'kappa(A)^2 = {np.linalg.cond(A)**2:.2f}')
```

```
x = [1. 1.]   after changing b_2 by 1e-4: [0. 2.]
kappa(K) = 40002;  bound 1.41;  realized 1.00
kappa(A) = 7.469;  kappa(A^T A) = 55.78;  kappa(A)^2 = 55.78
```

出力から手計算の結果が確認できる。解は $(1,1)$ から $(0,2)$ へ飛び、$\kappa(\mathbf{K}) \approx 40{,}002$ である。控えめなサイズの 4×2 直線当てはめ行列でさえ $\kappa = 7.47$ であり、正規方程式を作ると条件数はすでに $\kappa^2 = 55.8$ まで膨れ上がってしまう。

条件性とボウルの形状。 大きな条件数は、最適化においても細長く伸びた等位集合として直接現れる。列フルランクの最小二乗目的関数 $\tfrac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$ に対し、ヘッセ行列（Hessian）は $\mathbf{H} = \mathbf{A}^{\top}\mathbf{A}$

であり、その等位集合は半軸の長さが $1/\sigma_i$ に比例する $\mathbf{v}_i$ 方向の楕円体となる。したがって等高線の軸比は $\kappa(\mathbf{A})$ となる一方、ヘッセ行列自身の条件数は $\kappa(\mathbf{A})^2$ となる。このような細長い谷の上では、勾配降下法（gradient descent）は急峻な壁の間をジグザグに往復して谷底に沿う進行が遅くなり（Figure 33.31）、その収束率（convergence rate）は 35.1 節で導出するとおり $\kappa(\mathbf{H})$ によって決まる。

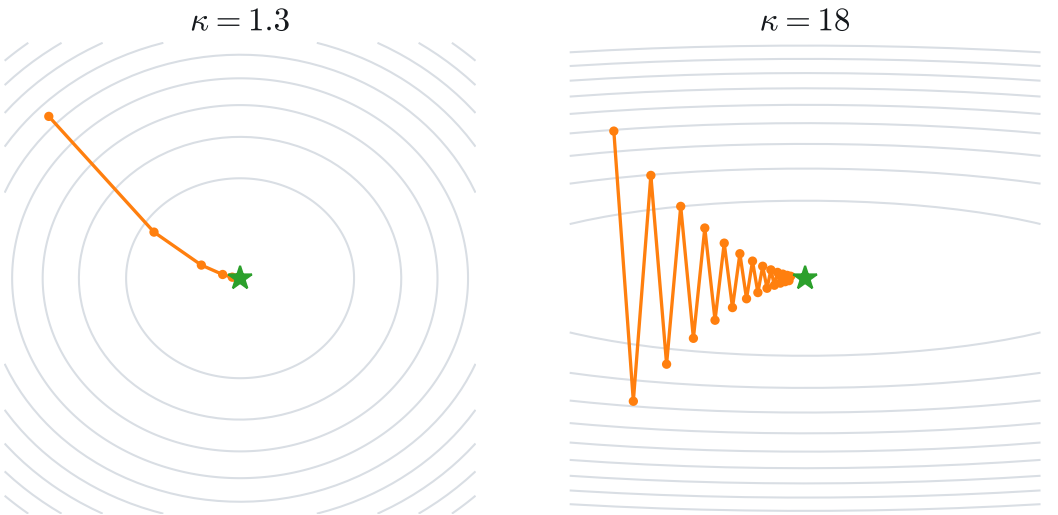


Figure 33.31: 良条件な二次関数のボウル（$\kappa(\mathbf{H}) = 1.3$、ほぼ円形の等高線、ほぼ直線的な経路）と悪条件なボウル（$\kappa(\mathbf{H}) = 18$、細長い谷を横切るジグザグの軌跡）における勾配降下法。パネルのタイトルはヘッセ行列の条件数 $\kappa(\mathbf{H})$ を示し、等高線はその平方根の倍率（第 2 パネルでは約 4.2 倍）だけ細長く伸びている。

33.5.4 深層学習における SVD

現代の大規模モデルの訓練で繰り返し現れる 3 つの問いは、重み行列の特異値へと直接帰着する。すなわち、ファインチューニング用アダプタにどれだけのパラメータが必要か、重み空間のあらゆる方向を対等に扱う更新をどう構成するか、そして単一の層が入力を最大どれだけ引き伸ばし得るか、という 3 つである。

33.5.4.1 低ランク更新

大規模な事前学習済みモデルの全パラメータをファインチューニング中に更新するのは、メモリと計算量の両面でコストが大きい。LoRA (Hu et al. 2021) はこれを回避するため、事前学習済みの各重み行列 $\mathbf{W} \in \mathbb{R}^{m \times n}$ を固定し、ランク k の補正項 $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$ のみを学習する。ここで $\mathbf{B} \in \mathbb{R}^{m \times k}$、$\mathbf{A} \in \mathbb{R}^{k \times n}$ である（ここでの $\mathbf{A}$ は原論文の記法に従い $k \times n$ の低次元射影を表しており、前節までの一般行列ではない）。このアダプタで訓練可能なパラメータ数は $k(m+n)$ 個となり、元の mn 個から大幅に減る。4096×4096 の層で $k = 8$ とすれば、パラメータ数は $16{,}777{,}216$ 個から $8 \cdot 8192 = 65{,}536$ 個、すなわちわずか 0.39% へと減少する。Figure 33.32 に示すように、固定された重みと低ランクバイパスは同じ入力に対して並列に作用し、それぞれの出力を足し合わせる。

ここで Eckart–Young の定理は何を教えてくれ、何を教えてくれ**ない**のだろうか。もしフルファインチューニングによって特異値 $\sigma_1 \geq \sigma_2 \geq \cdots$ を持つ更新が得られるとすれば、いかなるラン

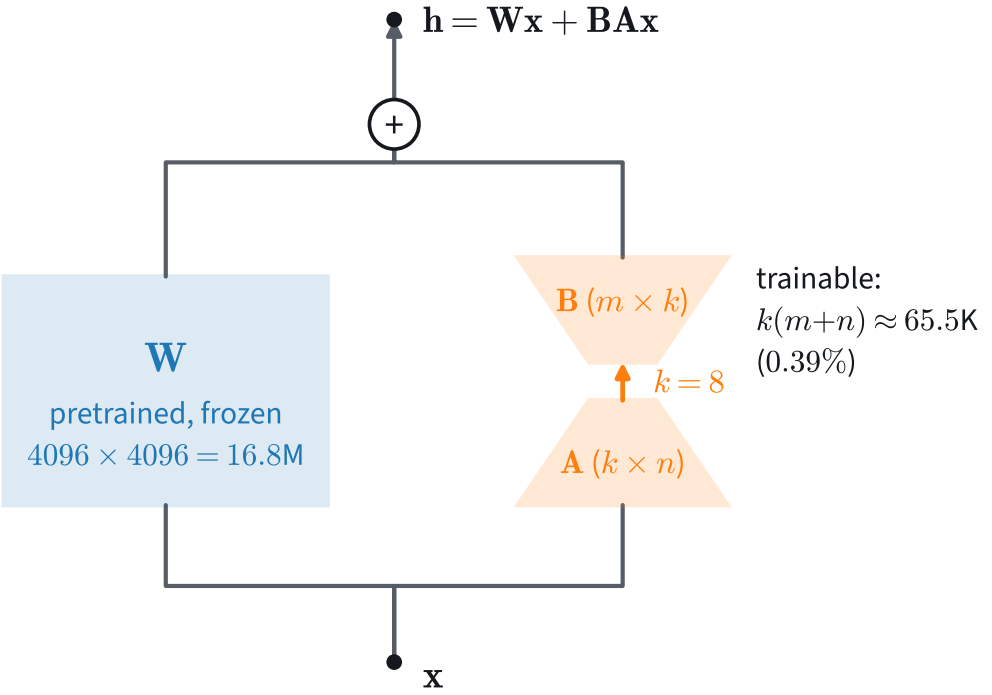


Figure 33.32: 1 層分の LoRA アーキテクチャ。事前学習済みの $\mathbf{W}$（全 $mn = 16.8\text{M}$ 個のパラメータ、$m = n = 4096$ の場合）は固定したままにし、並列バイパスがまず入力を $k = 8$ 次元へ $\mathbf{A}$ で圧縮してから $\mathbf{B}$ で元の次元へ戻すため、訓練されるのは $k(m+n) \approx 65.5\text{K}$ 個のパラメータ（約 0.39%）のみである。2 つの経路は $\mathbf{h} = \mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}$ と加算される。訓練後は $\mathbf{B}\mathbf{A}$ を $\mathbf{W}$ に統合できるため、デプロイ時の追加コストは一切かからない。

ク k アダプタもその特定の更新に対してスペクトルノルムで σ_{k+1} 未満、フロベニウスノルムで $(\sum_{i>k}\sigma_i^2)^{1/2}$ 未満には近づけないことが Eckart–Young から保証される。しかし、勾配降下法が実際にどのアダプタを学習するかまでは規定し**ない**。LoRA が成功するのは、たとえフルファインチューニングの更新に対する最良ランク k 近似でなくとも、**何らかの**低ランク更新が低い損失を達成するときである。多くの下流タスク（downstream task）においてそのような低ランク解が存在するという事実こそ、LoRA の背後にある経験的発見である。演習問題 2 では、ランク k アダプタが $\mathbf{W}$ の特異値を最大どれだけ動かし得るかを評価する。

33.5.4.2 直交化された更新

Muon オプティマイザ (Jordan et al. 2024) は、生のモメンタム行列 $\mathbf{M}$ の方向にそのまま重み行列を更新するのではなく、$\mathbf{M}$ の**極因子**（polar factor）、すなわち $\mathbf{M}$ の特異ベクトルを保ったまま非ゼロ特異値をすべて 1 に置き換えた行列の方向に更新する。オプティマイザ全体の動機と詳細は 9.6 節で扱うこととし、ここではそれが依拠する 2 つの線形代数の道具、すなわち極因子とは何か、そして行列積だけを使ってそれをどう計算するかを与える。

極分解。正方行列 $\mathbf{A}$ に対し、SVD の中に $\mathbf{V}^\top\mathbf{V}=\mathbf{I}$ を挿入すると、直交因子と対称な伸縮の積へと分解（極分解、polar decomposition）される。

$$\mathbf{A}=\mathbf{U}\Sigma\mathbf{V}^\top=(\mathbf{U}\mathbf{V}^\top)\,(\mathbf{V}\Sigma\mathbf{V}^\top)=\mathbf{Q}\,\mathbf{P}. \tag{33.60}$$

ここで $\mathbf{Q}=\mathbf{U}\mathbf{V}^\top$ は（直交行列の積として）直交行列であり、$\mathbf{P}=\mathbf{V}\Sigma\mathbf{V}^\top$ は固有値 σ_i を持つ対称半正定値行列である。このように、あらゆる正方行列は対称な伸縮に直交変換を続けたものへと分解される。これは複素数を極形式 $z=e^{i\theta}|z|$ と書くことの行列版にあたる。なお、$\mathbf{Q}$ は純粋な回転とは限らない。$\mathbf{A}$ が可逆のとき $\det\mathbf{Q}=\operatorname{sign}\det\mathbf{A}$ となるため、$\mathbf{Q}$ は $\det\mathbf{A}<0$ のとき常に鏡映を含む。せん断行列 $\mathbf{T}$ に対しては、Equation 33.47 より $\mathbf{Q}=\mathbf{R}_{31.7^\circ}\mathbf{R}_{58.3^\circ}^\top=\mathbf{R}_{-26.6^\circ}=\frac{1}{\sqrt5}\left(\begin{smallmatrix}2&1\\-1&2\end{smallmatrix}\right)$（$\tan 26.6^\circ=\frac12$ より）および $\mathbf{P}=\frac{1}{\sqrt5}\left(\begin{smallmatrix}2&1\\1&3\end{smallmatrix}\right)$ が得られる。積を計算すると、確かに $\mathbf{QP}=\frac15\left(\begin{smallmatrix}5&5\\0&5\end{smallmatrix}\right)=\mathbf{T}$ および $\mathbf{P}^2=\left(\begin{smallmatrix}1&1\\1&2\end{smallmatrix}\right)=\mathbf{T}^\top\mathbf{T}$ が成り立っている。

長方形行列 $\mathbf{M}$ に対しても、その薄い SVD を $\mathbf{M}=\mathbf{U}\Sigma\mathbf{V}^\top$ とすれば、極因子はやはり $\mathbf{U}\mathbf{V}^\top$ である。これは $\mathbf{M}$ と同じ特異ベクトルを持ち、すべての特異値が 1 に等しいため、その列（$m\ge n$ のとき）または行（$m\le n$ のとき）が正規直交になる。$\mathbf{M}$ がランク落ちしている場合（$r<\min(m,n)$）は極因子は一意ではないが、コンパクト SVD から得られる $\mathbf{U}_r\mathbf{V}_r^\top$ が自然な標準的選択となる。演習問題 6 では、正方行列において $\mathbf{U}\mathbf{V}^\top$ がフロベニウスノルムの意味で $\mathbf{A}$ に最も近い直交行列でもあることを示す。

行列の奇多項式。訓練ステップごとにすべての重み行列の完全 SVD を計算するのは、遅すぎて現実的ではない。幸いなことに、行列積（matrix multiplication）だけを使えば、特異ベクトルには一切触れずに特異値だけを変換できる。

命題（奇多項式は特異値に作用する）。$\mathbf{X}=\mathbf{U}\Sigma\mathbf{V}^\top$ を薄い SVD とし、$p(\sigma)=\sum_j c_j\sigma^{2j+1}$ を奇多項式とする。このとき次式が成り立つ。

$$p(\mathbf{X}):=\sum_j c_j\,\mathbf{X}(\mathbf{X}^\top\mathbf{X})^j=\mathbf{U}\,p(\Sigma)\,\mathbf{V}^\top. \tag{33.61}$$

証明。$\mathbf{X}^\top\mathbf{X}=\mathbf{V}\Sigma^2\mathbf{V}^\top$ であるから、べき乗をとると $(\mathbf{X}^\top\mathbf{X})^j=\mathbf{V}\Sigma^{2j}\mathbf{V}^\top$ となり、左から $\mathbf{X}=\mathbf{U}\Sigma\mathbf{V}^\top$ を掛けると $\mathbf{X}(\mathbf{X}^\top\mathbf{X})^j=\mathbf{U}\Sigma^{2j+1}\mathbf{V}^\top$ が得られる。これを j について足し合わせれば証明が完了する。■

したがって、任意の行列奇多項式は $\mathbf{U}$ と $\mathbf{V}$ をそのまま保ちながら、各特異値に対して独立にスカラー関数 p を適用する。極因子 $\mathbf{U}\mathbf{V}^\top$ に到達するには、繰り返し適用することですべての正の特異値を 1 へと向かわせる奇多項式 p を選べばよい。

Newton–Schulz。古典的な **Newton–Schulz 反復** $\mathbf{X} \leftarrow \frac{1}{2}(3\mathbf{X} - \mathbf{X}\mathbf{X}^\top\mathbf{X})$ は、すべての特異値に 3 次多項式 $p(\sigma) = \frac{1}{2}(3\sigma - \sigma^3)$ を適用する。その不動点（fixed point）は 0 と ± 1 である。区間 $(0, 1]$ 上では、p は 1 を行き過ぎることなくすべての値を上向きに押し上げる。なぜなら $p(\sigma) - \sigma = \frac{1}{2}\sigma(1 - \sigma^2) \geq 0$ であり、かつ p は最大値 $p(1) = 1$ をとるからである。実際、p は区間 $(0, \sqrt{3})$ 全体を $(0, 1]$ の中へ写すため、1 のこの**直接吸引領域**（immediate basin）に含まれるすべての初期値は 1 へと収束する一方、それより大きい初期値では不規則な挙動を示し得る（演習問題 5）。初期行列を $\mathbf{X}_0 = \mathbf{M}/\|\mathbf{M}\|_F$ とスケーリングすれば、すべての特異値が確実に $(0, 1]$ の内部から出発する。$\|\mathbf{M}\|_F \geq \sigma_1(\mathbf{M})$ だからである。またゼロ特異値はゼロのままとどまるため（$p(0) = 0$ より）、ランク落ちした $\mathbf{M}$ も綺麗に $\mathbf{U}_r\mathbf{V}_r^\top$ へと収束する。

これには何回の反復が必要だろうか。0 の近傍では $p(\sigma) \approx \frac{3}{2}\sigma$ であるから、小さな特異値 s は 1 ステップごとに $3/2$ 倍に成長し、オーダー 1 に達するまでにおよそ $\log_{3/2}(1/s)$ ステップを要する。1 の近傍では展開すると $p(1 - e) = 1 - \frac{3}{2}e^2 + \frac{1}{2}e^3$ となり、誤差が毎ステップおよそ二乗されるため、オーダー 1 から 10^{-6} まで縮むのに追加で約 4 ステップしかかからない。この 2 つの段階をつなぎ合わせると、およそ $\log_{3/2}(1/s_{\min}) + 4$ ステップと予測され、これは最小の正規化特異値 $s_{\min} = \sigma_{\min}(\mathbf{M})/\|\mathbf{M}\|_F$ によって決まる。モメンタムに見立てたランダムな 6×4 行列でこの大まかな見積もりを検証してみよう。

```
M = np.random.standard_normal((6, 4))               # a stand-in momentum matrix
X = M / np.linalg.norm(M)                          # Frobenius normalization: sigma in (0, 1]
s_min = np.linalg.svd(X, compute_uv=False)[-1]
for k in range(1, 41):
    X = 1.5 * X - 0.5 * X @ X.T @ X                 # p(sigma) = (3 sigma - sigma^3) / 2
    if np.abs(np.linalg.svd(X, compute_uv=False) - 1).max() < 1e-6:
        break
U, s, Vt = np.linalg.svd(M, full_matrices=False)
print(f'smallest scaled sigma {s_min:.3f};  predicted steps '
      f'{np.log(1 / s_min) / np.log(1.5) + 4:.1f}')
print(f'steps until max|sigma_i - 1| < 1e-6: {k};  distance to U V^T: '
      f'{np.linalg.norm(X - U @ Vt):.1e}')
```

```
smallest scaled sigma 0.146;  predicted steps 8.7
steps until max|sigma_i - 1| < 1e-6: 9;  distance to U V^T: 7.3e-11
```

実際のステップ数は予測と 1 反復以内の差で一致し、反復結果は完全 SVD から求めた厳密な極因子から 10^{-5} 以内に収まっている。$s_{\min}$ は乱数抽出ごとに変動するためステップ数も変動し、ほぼランク落ちに近い $\mathbf{M}$ ではより多くの反復を要する。実用上の Muon では、3 次多項式の代わりに調整された 5 次多項式を用い、1 への厳密な収束と引き換えに高速化を図っている（9.6.2 節）。

33.5.4.3 スペクトルの計測と制御

3 つ目の問いは線形層が入力を最大どれだけ引き伸ばし得るかというものであり、最大特異値がその答えを直接与える。より広く見れば、特異値スペクトル全体から行列が本当に低ランクに近いかどうかがわかる。そのため現代の訓練レシピ（training recipe）では、完全 SVD のコストを払うことなく特異値を抑え、調べ、近似する。

スペクトル正規化。@eq-mdl-sigma1-variational より、線形層 $\mathbf{x} \mapsto \mathbf{W}\mathbf{x}$ は $\|\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{x}'\| \leq$

$\sigma_1 \|\mathbf{x} - \mathbf{x}'\|$ を満たし、$\mathbf{x} - \mathbf{x}'$ が $\mathbf{v}_1$ と同じ方向を向くときに等号が成り立つ。写像が入力の差を最大どれだけ増幅し得るかを抑える最小の定数を**リプシッツ定数**（Lipschitz constant）と呼ぶため、ここでのリプシッツ定数は単なる $\sigma_1 = \|\mathbf{W}\|_2$ である。$\mathbf{W}$ を σ_1 で割って層の伸縮率の上限を 1 に抑える手法がスペクトル正規化（spectral normalization）であり、GAN の識別器（discriminator）の訓練を安定化させる (Miyato et al. 2018)。その際、完全 SVD を計算する代わりに、σ_1 を $\mathbf{W}^\top \mathbf{W}$ に対するべき乗法（power iteration、33.4.1.2 節）で推定する。具体的には $\mathbf{u} \leftarrow \mathbf{W}\mathbf{v}$ と $\mathbf{v} \leftarrow \mathbf{W}^\top \mathbf{u}$ を半ステップごとに正規化しながら交互に繰り返す。この 1 往復で $\mathbf{W}^\top \mathbf{W}$ を 1 回掛けたことになり、$\|\mathbf{W}\mathbf{v}\|$ が σ_1 の推定値を与える。1 往復ごとに $\mathbf{v}$ は $\mathbf{v}_1$ へ引き寄せられ、誤差はおよそ $(\sigma_2/\sigma_1)^2$ 倍に縮小する。Miyato et al. (2018) は、前ステップのベクトルを初期値とするウォームスタートを行えば、訓練ステップあたり 1 往復で十分であることを実験的に見出した（なお畳み込み層では、カーネルを 2D 行列に変形することで畳み込みのスペクトルノルムの厳密値ではないものの近似値が得られる）。

スペクトルの計測。先ほどの LoRA における 0.39% という数値は、ランク 8 で十分だと仮定していた。ある行列が本当に低ランクに近いかどうかは特異値がどれほど速く減衰するかに完全に依存するため、打ち切る前に計測すべきである。以下のセルでは、2 つの 400×500 行列、すなわち構造をまったく持たないランダムなガウス行列と、ピクセル値の平均を差し引いた猫の写真を比較する。

```
G = np.random.standard_normal(img.shape)                        # no structure
for name, W in [('Gaussian', G), ('photograph', img - img.mean())]:
    s = np.linalg.svd(W, compute_uv=False)
    r95 = int(np.searchsorted(np.cumsum(s**2) / (s**2).sum(), 0.95)) + 1
    print(f'{name:10s}: sigma from {s[0]:.3g} to {s[-1]:.2g};  95% energy at rank {r95:3d};  '
          f'adapter {r95 * (m + n) / (m * n):.1%} of W')
```

```
Gaussian  : sigma from 42.2 to 2.5;  95% energy at rank 269;  adapter 121.0% of W
photograph: sigma from 2.65e+04 to 2.4;  95% energy at rank   8;  adapter 3.6% of W
```

対比は鮮明である。ガウス行列の特異値はマルチェンコ・パスツール区間 $[\sqrt{n} - \sqrt{m}, \sqrt{n} + \sqrt{m}] \approx [2.4, 42.4]$（33.5.2.1 節）全体に広がって急激な減衰を見せず、エネルギーの 95% を捉えるには 400 のうちおよそランク 270 が必要となるため、低ランクアダプタの方が元の行列よりかえって**多くの**パラメータを使ってしまう。対照的に、中心化した写真はランク 8 でエネルギーの 95% に達し、パラメータ数 4% 未満のアダプタに相当する。どちらも訓練されたニューラルネットワークの重みではないため、この比較はあくまで両極端を示す例にすぎない。LoRA において問題となるのはタスク固有の更新行列であり、そのスペクトルは仮定するのではなく実際に計測しなければならない。

ランダム化 SVD。行列が大きすぎて直接分解できない場合、**ランダム化 SVD**（randomized SVD）(Halko et al. 2011) は高精度なランク k 近似を、ごく少数の行列・ベクトル積（$\mathbf{A}$ との積）だけで計算する。その中核となる要素が**値域探索**（range finder）である。すなわち、$\mathbf{A}$ にランダムなガウス行列 Ω（k より少し多い列を持つ）を掛け、$\mathbf{A}\Omega$ の列を正規直交化して $\mathbf{Q}$ とし、はるかに小さい行列 $\mathbf{Q}^\top \mathbf{A}$ の SVD を計算する。特異値が減衰していれば、$\mathbf{A}$ の列のランダムな線形結合は常に上位の特異方向に支配されるため、$\mathbf{Q}$ は高い確率でそれらを捉える。演習問題 7 で、減衰が遅いスペクトルに対するべき乗法による改良も含めてこのアルゴリズムを実装し、33.6 節でランダム化スケッチングというより広い文脈の中に位置づける。

最後に、SVD の 3 つの限界を心に留めておこう。第 1 に、完全 SVD には $O(mn \min(m, n))$ 回の演算コストがかかる（$4096^3 \approx 7 \times 10^{10}$ 回にものぼり、単一の 4096×4096 重み行列でも膨大である）。Newton–Schulz 反復やランダム化値域探索のような実用的アルゴリズムが行列積のみに頼るのはこのためである。第 2 に、低ランク性はデータが持つ性質であって無条件に保証されるものではない。このことは直前のガウス行列が示したとおりである。第 3 に、SVD が捉

えるのは**線形**な構造のみである。平面上の円周上に並ぶ点群は、円が本質的に 1 次元であるにもかかわらずランク 2 のデータ行列をなす。曲がった関数や非線形な構造を理解するには導関数（derivative）が必要であり、次章ではそのテーマへと進む。

33.5.5 まとめ

形状にかかわらず任意の実行列 $\mathbf{A}$ には、入力の正規直交基底 $\{\mathbf{v}_i\}$ と出力の正規直交基底 $\{\mathbf{u}_i\}$ が存在し、その間でこの行列は非負の伸縮率の単純な列 $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$（$\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$）として作用する。この分解は、グラム行列 $\mathbf{A}^\top\mathbf{A}$ にスペクトル定理を適用することで $\sigma_i = \sqrt{\lambda_i(\mathbf{A}^\top\mathbf{A})}$ として得られる。

順序付けられたこのたった 1 つの特異値のリストから、5 つの中心的な量が一度に導かれる。すなわち、ランク（正の σ_i の個数）、スペクトルノルムとフロベニウスノルム（$\|\mathbf{A}\|_2 = \sigma_1$ と $\|\mathbf{A}\|_F^2 = \sum_i \sigma_i^2$）、打ち切り SVD $\mathbf{A}_k$ が与える最良ランク k 近似（Eckart–Young の定理であり、中心化データに対しては主成分分析となる）、最小ノルム最小二乗解を返すムーア・ペンローズ擬似逆行列 $\mathbf{A}^+$、そして相対誤差の増幅率を抑える条件数 $\kappa = \sigma_1/\sigma_r$ である。深層学習においても、同じ特異値スペクトルがファインチューニング用の低ランクアダプタのサイズを決め、Muon が用いる極因子を定義し、スペクトル正規化が制御するリプシッツ定数を与える。

33.5.6 演習問題

1. ***ノルムと最大伸縮率***。本文ではスペクトル定理から Equation 33.48 を導いたが、最大伸縮率だけから逆に SVD を構成することもできる。
    1. $\sigma_1 = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|$ とおき、$\sigma_1 > 0$ と仮定して、スペクトル定理を使わずに SVD を構成せよ。(i) ある単位ベクトル $\mathbf{v}_1$ で最大値が達成されることを示し、$\mathbf{u}_1 = \mathbf{A}\mathbf{v}_1/\sigma_1$ とおけ。(ii) 単位ベクトル $\mathbf{w} \perp \mathbf{v}_1$ に対し、関数 $t \mapsto \|\mathbf{A}(\mathbf{v}_1 + t\mathbf{w})\|^2/(1+t^2)$ は $t=0$ で最大値をとるため、そこでの導関数は 0 になる。ここから $\mathbf{u}_1^\top \mathbf{A}\mathbf{w} = 0$、すなわち $\mathbf{A}$ が $\mathbf{v}_1^\perp$ を $\mathbf{u}_1^\perp$ の中へ写すことを導け。(iii) $\mathbf{A}$ の $\mathbf{v}_1^\perp$ への制限に対して帰納的に繰り返せ。単位球面のコンパクト性はどこで使われたか。
    2. $\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F \leq \sqrt{r}\,\|\mathbf{A}\|_2$（ただし $r = \operatorname{rank}\mathbf{A}$）を示せ。左右それぞれの不等号で等号を達成するのはどのような行列か。
    3. 比 $\|\mathbf{A}\|_F^2/\|\mathbf{A}\|_2^2$ は $\mathbf{A}$ の**安定ランク**と呼ばれ、(b) より 1 と $\operatorname{rank}\mathbf{A}$ の間にある。減衰スペクトル $\sigma_i = e^{-i/12}$（$i = 0, \ldots, 255$、幾何級数）に対して安定ランクを閉じた式で求め、このスペクトルのエネルギーの 95% を保持するにはランク 18 が必要であることを示せ。次に、400×500 のガウス行列（33.5.4.3 節）の安定ランクを $\|\mathbf{G}\|_F^2 \approx mn$ と $\sigma_1 \approx \sqrt{m} + \sqrt{n}$ から見積もり、エネルギーの 95% を保持するランク（約 270）と比較せよ。なぜ安定ランクは通常のランクよりはるかに小さくなり得るのか。
2. **Weyl 摂動限界**。更新 Δ、たとえばランク k の LoRA 更新 $\mathbf{B}\mathbf{A}$（原論文の記号）は、重み行列 $\mathbf{W}$ の特異値を変化させる。Weyl の不等式 Equation 33.54（33.5.2.1 節で証明）は、その変化幅の上限を与える。
    1. $j = 1$ を Equation 33.54 に代入し、不等式を両方向に適用することで、$|\sigma_i(\mathbf{W}+\Delta) - \sigma_i(\mathbf{W})| \leq \sigma_1(\Delta)$ がすべての i で成り立つことを導け。
    2. $\mathbf{W} = \operatorname{diag}(1, 2)$、$\Delta = 0.1\,\mathbf{1}\mathbf{1}^\top$ とする。$\mathbf{W}+\Delta$ の特異値を計算し、それぞれの変化量を $\sigma_1(\Delta)$ と比較せよ。
    3. 次に $\operatorname{rank}\Delta \leq k$ とする。33.5.2.1 節のフロベニウスノルムの証明では、Equation 33.54 を $j = k+1$ としてランクが高々 k の行列に適用した。同じ手順を Δ に適用して、交錯不等式 $\sigma_{i+k}(\mathbf{W}) \leq \sigma_i(\mathbf{W}+\Delta)$ および $\sigma_{i+k}(\mathbf{W}+\Delta) \leq \sigma_i(\mathbf{W})$ を導け。これらを用いて、$\mathbf{W}+\Delta$ と $\mathbf{W}$ のランクの差が高々 k であること、さらに高々 k

個の特異値（$\mathbf{W}+\Delta$ のもの）しか $\sigma_1(\mathbf{W})$ を超えないことを、$\sigma_1(\Delta)$ がどれほど大きくても成り立つ形で示せ。なぜ (a) の評価からはこれらの結論が得られないのか。

3. [code] **PCA における中心化と Fashion-MNIST**。$\mathbf{X}\in\mathbb{R}^{n\times d}$ の各行をデータ点とし、平均行ベクトルを $\bar{\mathbf{x}}$、共分散を $\mathbf{C}$ とする（33.5.2.2 節の主成分分析と同じ設定）。
   1. $\frac{1}{n}\mathbf{X}^\top\mathbf{X}=\mathbf{C}+\bar{\mathbf{x}}\bar{\mathbf{x}}^\top$ を示せ。
   2. 33.5.2.2 節と同様の点群（300 点、標準偏差 3 と 0.8、回転角 $\pi/5$ の軸に沿う分布）を生成し、すべての点を $c\,(1,-1)/\sqrt{2}$（$c\in\{0,1,3,10,30\}$）だけ平行移動させよ。各 c について、**中心化していない**データの最大右特異ベクトル、主軸 $(\cos\frac{\pi}{5},\sin\frac{\pi}{5})$ とのなす角、および比 $\sigma_1^2/\sum_i\sigma_i^2$ を計算せよ。
   3. (a) を用いて、最大特異ベクトルが主軸から平均の方向へと向きを変えるオフセットの大きさを $\mathbf{C}$ の固有値から予測せよ。
   4. 再び (a) を用いて比 $\sigma_1^2/\sum_i\sigma_i^2$ を大きな c に対して近似し、$c\to\infty$ における極限を求めよ。$c=10$ では、この比は中心化後の説明分散比に近い値となる。この一致はデータの性質を反映したものか、それとも選んだオフセットによるものか。
   5. `d2l.FashionMNIST` を用いて（33.1 節と同様に）Fashion-MNIST の T シャツとズボンのテスト画像を読み込み、長さ 784 のベクトルへ平坦化し、中心化した上で上位 2 本の右特異ベクトルを計算せよ。最初の 2 つの主成分が張る平面上に 2 クラスをプロットせよ。これら 2 方向は分散のどれだけの割合を占めるか。さらに $\mathbf{v}_1$ とクラス平均の差 $\bar{\mathbf{x}}_{\text{trouser}}-\bar{\mathbf{x}}_{\text{t-shirt}}$ のなす角を計算し、2 つの方向が近いものの厳密には一致しない理由を説明せよ。最後に、1 枚の T シャツ画像を最初の 5 個、20 個、50 個の主成分から再構成し、それぞれの相対誤差を報告せよ。

4. [code] **リッジなし極限と勾配降下法**。過剰パラメータ化された線形モデルは方程式より多くのパラメータを持つため、最小二乗法は無数の最小点を持つ。小問 (a) は任意の形状・ランクで成り立ち、(b) と (c) では $\mathbf{A}\in\mathbb{R}^{m\times n}$（$m<n$）を用いる。
   1. フィルタ形式 Equation 33.57 を用いて、$(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\top\to\mathbf{A}^+$（$\lambda\to0^+$）が任意の形状・ランクの行列 $\mathbf{A}$ に対して成り立つことを示せ。なぜ $(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$ は $m<n$ のとき存在しないのか。
   2. $\frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2$ に対するステップサイズ $0<\eta<2/\sigma_1^2$ の勾配降下法は、$\mathbf{x}_{t+1}=\mathbf{x}_t-\eta\,\mathbf{A}^\top(\mathbf{A}\mathbf{x}_t-\mathbf{b})$ と更新する。$\mathbf{x}_t-\mathbf{x}_0$ が常に $\mathbf{A}$ の行空間内にとどまること、および $\mathbf{x}_t$ が $\mathbf{A}^+\mathbf{b}+(\mathbf{I}-\mathbf{A}^+\mathbf{A})\mathbf{x}_0$ へ収束することを示せ。初期化 $\mathbf{x}_0=\mathbf{0}$ はどの解を選び出すか。
   3. 20×50 のガウス行列 $\mathbf{A}$ とガウスベクトル $\mathbf{b}$ に対して (a) と (b) を検証せよ。$\mathbf{x}_0=\mathbf{0}$ とランダムな $\mathbf{x}_0$ の両方から勾配降下法を実行し、それぞれの収束先とそのノルムを $\mathbf{A}^+\mathbf{b}$ と比較せよ。

5. [code] **Newton–Schulz の吸引領域**。Newton–Schulz 反復 $\mathbf{X}\leftarrow\frac{1}{2}(3\mathbf{X}-\mathbf{X}\mathbf{X}^\top\mathbf{X})$（33.5.4.2 節）は $\mathbf{X}=\mathbf{M}/\|\mathbf{M}\|_F$ から出発する。
   1. $\mathbf{X}=\mathbf{U}\Sigma\mathbf{V}^\top$ を薄い SVD とする。Equation 33.61 を使わずに直接計算することで、1 ステップ後の結果が $\mathbf{U}\,p(\Sigma)\,\mathbf{V}^\top$（各対角成分に $p(\sigma)=\frac{1}{2}(3\sigma-\sigma^3)$ を適用したもの）になることを示せ。
   2. $(0,\sqrt{3})$ が 1 の直接吸引領域であることを示せ。すなわち、p がそれを $(0,1]$ の中へ写し、反復列 $p(p(\cdots p(\sigma_0)))$ がすべての $\sigma_0\in(0,\sqrt{3})$ に対して 1 へ収束することを示せ。また、$\sqrt{3}$ を超える初期値の中にも 1 に到達するもの（2.15 を試せ）と到達しないもの（2 を試せ）があることを示せ。さらに $p(1-e)$ を e のべき乗で展開せよ。1 までの距離が小さくなった後は、どれほどの速さで縮小するか。
   3. 6×4 のガウス行列 $\mathbf{M}$ に対し、$\|\mathbf{X}-\mathbf{U}\mathbf{V}^\top\|_F<10^{-6}$ になるまでのステップ数を、初期値 $\mathbf{M}/c$ で $c=\|\mathbf{M}\|_F$ とした場合と $c=\sigma_1(\mathbf{M})$ とした場合のそれぞれで数えよ。

両方のステップ数を $\log_{3/2}(1/s_{\min})+4$（$s_{\min}$ はスケーリング後の最小特異値）と比較せよ。1 ステップ節約できるにもかかわらず、Muon が $\|\mathbf{M}\|_F$ で割るのはなぜか（$\sigma_1(\mathbf{M})$ ではなく）。

4. $\mathbf{M}/c$ を初期値として、$c=\sigma_1(\mathbf{M})/2$ の場合と $c=\sigma_1(\mathbf{M})/2.5$ の場合で反復を開始せよ。それぞれ何に収束するか。(b) の多項式 p を用いて両方の結果を説明せよ。

6. [code] **極因子と Procrustes 問題**。●極分解 Equation 33.60 は正方行列を $\mathbf{A}=\mathbf{QP}$（$\mathbf{Q}$ は直交行列、$\mathbf{P}$ は対称半正定値行列）と表す。

1. $\mathbf{P}=(\mathbf{A}^\top\mathbf{A})^{1/2}$ が $\mathbf{A}$ によって一意に定まること、さらに $\mathbf{Q}$ も $\mathbf{A}$ によって（$\mathbf{A}$ が可逆のとき）一意に定まることを示せ。また $\operatorname{diag}(1,0)$ の異なる 2 つの極分解を挙げよ。
2. フロベニウスノルムにおいて $\mathbf{Q}=\mathbf{UV}^\top$ が $\mathbf{A}$ に最も近い直交行列であることを示せ。その際、問題を $\operatorname{tr}(\mathbf{R}^\top\mathbf{A})$ の直交行列 $\mathbf{R}$ に関する最大化へと帰着させよ。
3. 回転だけ異なる 2 つの埋め込み空間は、対応点の組 $\mathbf{X},\mathbf{Y}\in\mathbb{R}^{d\times N}$ から $\|\mathbf{RX}-\mathbf{Y}\|_F$ を直交行列 $\mathbf{R}$ 上で最小化することで位置合わせできる。(b) を用いて、最小点が $\mathbf{YX}^\top$ の極因子であることを示せ。$d=50$、$N=1000$ とし、ランダムな直交行列 $\mathbf{R}$（ガウス行列の QR 分解の $\mathbf{Q}$ 因子）、ガウス行列 $\mathbf{X}$、および $\mathbf{Y}=\mathbf{RX}+\mathbf{E}$（$\mathbf{E}$ の成分は独立同分布な $\mathcal{N}(0,s^2)$、$s\in\{0,0.1,1,3\}$）でこれを検証せよ。$\|\hat{\mathbf{R}}-\mathbf{R}\|_F/\sqrt{d}$ を、この推定量と制約なし最小二乗推定量 $\mathbf{YX}^{+}$ について比較せよ。

7. [code] **ランダム化値域探索**。●33.5.4.3 節のランダム化 SVD は、$\mathbf{A}$ にガウス行列 $\Omega\in\mathbb{R}^{n\times(k+p)}$ を掛け、$\mathbf{Y}=\mathbf{A}\Omega$ を正規直交化して $\mathbf{Q}$ とし、$\mathbf{Q}^\top\mathbf{A}$ の SVD をランク k で打ち切る。

1. これを実装せよ。$\mathbf{A}\in\mathbb{R}^{1000\times 500}$ を $\mathbf{U}\operatorname{diag}(\sigma_1,\dots,\sigma_{500})\mathbf{V}^\top$（$\mathbf{U}$ と $\mathbf{V}$ はランダムな列正規直交行列）として構成し、速い減衰 $\sigma_i=e^{-i/5}$ の場合と遅い減衰 $\sigma_i=1/i$ の場合の 2 通りを用意せよ。$k=20$、$p\in\{0,5,10\}$ に対し、ランク k の近似結果のスペクトルノルム誤差を σ_{k+1}（Equation 33.53 の最小誤差）で割り、10 回の Ω の抽出にわたって平均せよ。なぜ片方のスペクトルではオーバーサンプリングだけで十分なのに、もう片方では不十分なのか。
2. $\mathbf{Y}$ を $(\mathbf{AA}^\top)^q\mathbf{A}\Omega$（$q\in\{1,2\}$）に置き換えよ。各スペクトルで誤差比はどう変化するか。q の効果を $(\mathbf{AA}^\top)^q\mathbf{A}$ の特異値を用いて説明せよ。また、各変種には $\mathbf{A}$ または $\mathbf{A}^\top$ との積が何回必要か（計算コストを答えよ）。

8. [code] **最適ハード閾値**。●●閾値 $\frac{4}{\sqrt{3}}\sqrt{n}\,\sigma_{\text{noise}}$（Gavish and Donoho (2014)）は、単一の信号成分の振る舞いから導かれる。$\mathbf{A}=\theta\,\mathbf{uv}^\top+\mathbf{N}$ を $n\times n$ 行列とし、$\mathbf{u},\mathbf{v}$ を単位ベクトル、$\theta>0$ とし、$\mathbf{N}$ の各成分は独立同分布なガウス分布 $\mathcal{N}(0,\sigma_{\text{noise}}^2)$ に従うとする。$x=\theta/(\sqrt{n}\,\sigma_{\text{noise}})$ とおき、$(\sigma_1,\hat{\mathbf{u}},\hat{\mathbf{v}})$ を $\mathbf{A}$ の最大特異三つ組とする（ただし対の符号は $\mathbf{u}^\top\hat{\mathbf{u}}\ge 0$ となるように選ぶ）。$n\to\infty$ のとき、Benaych-Georges and Nadakuditi (2012) によれば、σ_1 はバルク上端 $2\sqrt{n}\,\sigma_{\text{noise}}$ へ収束して特異ベクトルは $\mathbf{u}$、$\mathbf{v}$ と直交するようになり（$x\le 1$ のとき）、また $x>1$ のときは次式が成り立つ。

$$\frac{\sigma_1}{\sqrt{n}\,\sigma_{\text{noise}}}\to x+\frac{1}{x},\qquad (\mathbf{u}^\top\hat{\mathbf{u}})^2\to 1-\frac{1}{x^2},\qquad (\mathbf{v}^\top\hat{\mathbf{v}})^2\to 1-\frac{1}{x^2}.$$

$x=1$ におけるこの相転移は、演習問題 5（33.3 節）における $\beta=\sqrt{c}$ での固有値の相転移の特異値版にあたる。

1. 信号 $\theta\,\mathbf{uv}^\top$ に対する 2 つの推定量として、ランク 1 打ち切り $\sigma_1\hat{\mathbf{u}}\hat{\mathbf{v}}^\top$ とゼロ行列を考える。それぞれの二乗フロベニウス誤差を θ、σ_1、$\mathbf{u}^\top\hat{\mathbf{u}}$、$\mathbf{v}^\top\hat{\mathbf{v}}$ を用いて表せ。
2. 上の極限を用いて、打ち切りの方が漸近誤差が小さくなるのはちょうど $x>\sqrt{3}$ のときであり、$x=\sqrt{3}$ における特異値が $\frac{4}{\sqrt{3}}\sqrt{n}\,\sigma_{\text{noise}}$ であることを示せ。ノイズの上端

を超える特異値を生じるにもかかわらず切り捨てるべき信号強度はどの範囲か。

3. $n = 500$、$\sigma_{\text{noise}} = 1/\sqrt{n}$ とし、$x \in \{0.5, 1.2, 1.5, \sqrt{3}, 2, 3\}$ のそれぞれで 10 回ずつ乱数抽出して (a) と (b) を検証せよ。σ_1、余弦の二乗、および両方の誤差をそれぞれの極限値と比較せよ。

33.6 関連文献

本節では、本章で展開した線形代数を数値解析、ランダム行列理論、および深層学習の応用へと結びつけ、続いて本章の中核的な内容を網羅する厳選された教科書と講義の一覧を示す。主な関心が深層学習にある場合は、まずランダム射影（random projection）、反復ソルバ、ランダム行列と訓練可能な深さ、およびスケッチングから読み始め、その後に低ランクアテンションとスペクトルノルムオプティマイザを扱う最後の 2 項目へ進むとよい。本書の他の箇所は本節の内容に一切依存しない。

33.6.1 幾何学と線形変換

33.6.1.1 他の内積：カーネル法

33.1.4.2 節の最近傍平均規則では、テスト画像を 1 回のドット積によって 2 つのクラス平均と比較して分類するため、決定境界は必然的に平坦な超平面になる。しかし、T シャツとシャツを見分ける場合のように、クラスが湾曲していたり複数のクラスタに分かれていたりすると、この単純な仕組みは破綻する。カーネル法（kernel method）は、まったく同じ幾何学的規則を保ったまま内積だけを変更する。すなわち、$\mathbf{x} \cdot \mathbf{x}_i$ を、別の（場合によっては無限次元の）特徴空間における真の内積に対応する類似度関数 $k(\mathbf{x}, \mathbf{x}_i)$ に置き換えるのである。するとクラス平均はその隠れた空間での平均となり、各訓練例が重み $k(\mathbf{x}, \mathbf{x}_i)$ で投票するようになり、結果として得られる境界は元の入力空間内で自在に曲がることができる。どの訓練点が実際に非ゼロの重みを必要とするかを問うと、そのままサポートベクターマシンへと至る。Schölkopf and Smola (2002) はまさにこの分類器から筆を起こし、そこからカーネル法の理論全体を築き上げている。

33.6.1.2 ランダム射影

33.1 節では、高次元空間 $\mathbb{R}^d$ におけるランダムな方向どうしが互いにほぼ直交することを見た。驚くべきことに、この幾何学的な性質を利用すると、データを一切見ることなくデータセットの次元を縮小できる。Johnson–Lindenstrauss の補題 (Johnson and Lindenstrauss 1984) によれば、任意の n 個の点からなる集合に対し、適切にスケーリングしたランダムガウス行列を掛けて k 次元（オーダーは $\varepsilon^{-2} \log n$）へ写すと、$\binom{n}{2}$ 個すべてのペア間距離が高い確率で $1 \pm \varepsilon$ 倍の範囲内に保たれる。主成分分析（33.5.2.2 節）とは異なり、ランダム射影はデータにまったく依存せず、目標次元 k は外囲次元（ambient dimension）d にすら依存しない。したがって、この手法が最も威力を発揮するのは、d が $\varepsilon^{-2} \log n$ よりはるかに大きいときである。Dasgupta and Gupta (2003) は明示的な定数を伴う驚くほど短い証明を与えており、一方で Vershynin (2018) は確率ベクトル（random vector）の集中に関するより広い理論の中にこの補題を位置づけている。

33.6.1.3 大規模な連立一次方程式の解法

33.2.2.4 節では、LU 分解によって連立一次方程式を解き、大規模な疎行列（sparse matrix）の逆行列は一般に密になることを確認した。病的な最悪ケースの行列が存在するにもかかわらず、部分ピボット選択付きガウスの消去法（Gaussian elimination with partial pivoting）が浮動小数点演算でこれほど高い信頼性をもって機能するのはなぜだろうか。Trefethen and Bau (1997) は最悪ケースと実用上の安定性の間にあるこの乖離を明快に説明しており、Higham (2002) は丸め誤差（rounding error）に関する決定版の解析を与えている。

大規模な問題では、疎行列ソルバは決して $\mathbf{A}^{-1}$ を作らず、密な LU 因子の構築すら避けることが多い。疎行列の直接法（direct method）は、消去の過程で新たに生じる非ゼロ要素ができるだけ少なくなるよう、未知数の順序を注意深く並べ替える。反復法（iterative method）はさらに一歩進め、べき乗法（33.4.1.2 節）とまったく同様に、行列・ベクトル積 $\mathbf{Ax}$ のみを通じて行列にアクセスする。$\mathbf{A}$ が対称正定値であるとき、Hestenes and Stiefel (1952) の共役勾配法（conjugate gradient method）は、k ステップ後の誤差を高々 $2\big((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)\big)^{k}$ 倍（$\mathbf{A}$ が誘導するノルムのもとで）にまで縮小する。ここで、κ は 33.5.3.2 節で導入した条件数である。Golub and Van Loan (1996) は共役勾配法とその関連手法を詳細に扱っている。

33.6.1.4 縮約順序とテンソル分解

`einsum` 記法（33.2.1.4 節）は複数のテンソルを縮約した数学的結果を指定するが、ペアごとの縮約をどの順序で実行するかは定めない。この選択によって計算量は劇的に変わり得る。たとえば $n \times n$ 行列に対して、$(\mathbf{AB})\mathbf{x}$ の計算には約 n^3 回の乗算を要するが、$\mathbf{A}(\mathbf{Bx})$ ならばわずか $2n^2$ 回で済む。多くの因子が関わる場合、最も計算コストの低い順序を選ぶ問題は括弧の付け方の探索になり、`np.einsum(..., optimize=True)` や `opt_einsum` パッケージ (Smith and Gray 2018) がこれを自動化する。

低ランク近似（low-rank approximation）の考え方は行列を超えて高階テンソルへも広がるが、そこには厳しい落とし穴がある。3 元テンソルは、ランク 1 の外積 $\mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$ の和（CP 分解）、小さなコアテンソルに各モードに沿って因子行列を掛けた形（Tucker 分解）、あるいは 3 元コアの連鎖（Oseledets (2011) のテンソルトレイン）として表せる。しかし残念ながら、テンソル一般に対しては Eckart–Young の定理ほど明快な結果は成り立たない。CP ランクの決定は NP 困難であり、与えられた CP ランクでの最良近似が存在しないことさえある。CP 分解と Tucker 分解およびそれらのアルゴリズムについては、Kolda and Bader (2009) のサーベイが標準的な案内書である。

33.6.2 固有値とスペクトル

33.6.2.1 実用上の固有値計算

5 次以上の一般の多項式にはべき根による解の公式が存在しないため、実用的な固有値ソルバはすべて反復計算に頼らざるを得ない。さらに悪いことに、多項式の根はその係数に含まれる微小な丸め誤差に対して極端に敏感になり得るため、実用ライブラリが 33.3.1.2 節の特性多項式（characteristic polynomial）を陽に構成することもない。その代わり、`eig` はまず直交相似変換を適用し、$O(n^3)$ 回の演算で行列を Hessenberg 形（第 1 劣対角線より下がすべて 0 の形）へと簡約する。続いてシフト付き QR 法（shifted QR algorithm）を実行する。すなわち、$\mathbf{H} - \mu\mathbf{I} = \mathbf{QR}$ と分解し、$\mathbf{RQ} + \mu\mathbf{I}$ を作り、これを繰り返す。各ステップは相似変換であるため固有値は一切変化しない一方、劣対角成分は 0 に向かって縮小し、対角線上にスペクトルが現れる（実行列の複素共役ペアは 2×2 の対角ブロックに落ち着く）。対称行列では Hessenberg 形が三重対角になるため、`eigh` はこの構造を利用して大幅な高速化を実現している。Trefethen and Bau (1997) は部分空間上のべき乗法から QR 法を明快に導出しており、Golub and Van Loan (1996) は実用実装における工学的な詳細を網羅している。

33.6.2.2 擬スペクトルと Kreiss の行列定理

33.4.2.4 節では、非正規行列においては固有値が単位円盤の内部にあっても $\|\mathbf{A}^k\|_2$ の大きな過渡的増大（transient growth）は防げないことを見た。Kreiss の行列定理は、擬スペクトル（pseudospectrum）から得られる視覚的な直感を定量的な評価式へと変える。Kreiss 定数を $K(\mathbf{A}) = \sup_{|z|>1} (|z|-1)\|(z\mathbf{I}-\mathbf{A})^{-1}\|_2$ と定義すると、任意の $n \times n$ 行列に対して累乗ノルムの最悪ケースは次の不等式で上下から挟まれる。すなわち、$K(\mathbf{A}) \le \sup_k \|\mathbf{A}^k\|_2 \le e\,n\,K(\mathbf{A})$

である。下界は 33.4 節の演習問題で証明できる。一方、上界からは、擬スペクトルが単位円盤に密着している行列では非有界な過渡的増大が起こり得ないことが保証される。Trefethen and Embree (2005) はこの理論を網羅的に展開しており、線形化した固有値がすべて厳密に安定であるにもかかわらず乱流へ遷移する流体流れへの古典的な応用も詳しく扱っている。

33.6.2.3 混合時間とスペクトルギャップ

33.3 節の 2 ページからなるマルコフ連鎖では、第 2 固有値が $1/4$ であるため、定常分布までの距離がクリックごとに 4 分の 1 に縮小した。では、一般のマルコフ連鎖が混合するまでには何ステップ必要だろうか。可逆な連鎖に対し、$\lambda_\star$ を 1 以外のすべての固有値における絶対値の最大値とする。定常分布との全変動距離（total variation distance）が ε 以内に入るまでに必要なステップ数は、上下ともに $1/(1-\lambda_\star)$ の定数倍で抑えられる（定数は ε と最小の定常確率に依存する）。少なくとも確率 $1/2$ で現在の状態にとどまる怠惰な連鎖（lazy chain）では、すべての固有値が非負になるため、$\lambda_\star$ は単に第 2 固有値 λ_2 となる。さらに Cheeger の不等式は、スペクトルギャップ $1-\lambda_2$ を連鎖の**コンダクタンス**（部分集合の定常質量に対する、そこから流出する確率フローの最小比率）によって上下両側から評価する。言い換えれば、グラフのボトルネックから収束率を直接読み取ることができる。Levin and Peres (2017) は、33.4.1.3 節が Horn and Johnson (2012) から引用している収束の結果とあわせて、この理論をきわめて明快に解説している。

33.6.2.4 ランダム行列と訓練可能な深さ

33.4.2.5 節では、個々の層のヤコビ行列のスペクトル半径だけでは、ヤコビ行列の長い積の振る舞いを制御できないことを確認した。では、非常に深いネットワークを初期化からそのまま訓練できるかどうかは何によって決まるのだろうか。Saxe et al. (2014) は深層線形ネットワークの学習ダイナミクス（learning dynamics）を厳密に解き、直交する初期重みを用いれば訓練時間が深さに依存しなくなる一方、スケーリングしたガウス重みでは深さが増すにつれて学習が遅くなることを示した。この解析を非線形ネットワークへ拡張した Pennington et al. (2017) は、自由確率論（free probability）を用いて入出力ヤコビ行列の特異値分布を完全に求めている。その結果、すべての特異値が 1 の近傍に密に集まる**動的等長性**（dynamical isometry）と呼ばれるレジームは、直交重みと tanh 活性化関数の組み合わせでは達成できるものの ReLU では達成できず、このレジームで初期化したネットワークは層が深くてもはるかに高速に訓練できることが示された。

33.6.2.5 訓練済み重みのスペクトル

33.3.2.4 節のマルチェンコ・パスツール則（Marchenko–Pastur law）は、純粋な独立ノイズからなる行列の特異値スペクトルを記述するものであり、初期化したばかりのネットワークの重み行列はまさにこのような姿をしている。Martin and Mahoney (2021) は、訓練済みニューラルネットワークの重みのスペクトルをこのランダム行列のベースラインと比較し、訓練によって鋭い外れ値と重いべき乗則の裾の両方が生じることを観察した。同研究はこの裾指数（tail exponent）を暗黙的な自己正則化（implicit self-regularization）の兆候と解釈している。マルチェンコ・パスツール則からのこうした乖離は多くのアーキテクチャで一貫して見られるものの、テストデータで評価せずに裾指数を用いてモデルの品質を順位付けできるかについては、証明された定理ではなく経験的な相関にとどまっている。

33.6.3 打ち切り SVD を超える低ランク構造

33.6.3.1 行列補完と核ノルム

映画の評価値行列では全成分のうち数パーセントしか観測されていないことがある一方、Eckart–Young の定理（33.5.2.1 節）を用いるには最初から行列全体がわかっている必要がある。行列補完（matrix completion）はこの隔たりを埋めるため、観測されたすべての成分に一致する

という制約のもとで**核ノルム**（nuclear norm）$\|\mathbf{Z}\|_* = \sum_i \sigma_i$ を最小化する。これは、ベクトルのスパース性に対して ℓ_1 ノルムが凸な代理となるのと同様に、行列のランクに対する凸な代理として働く。Candès and Recht (2009) は、背後にある低ランク行列の特異ベクトルが少数のスパイクに集中せず座標全体に広がっている限り、この凸最適化問題によって n^2 個よりはるかに少ないランダムな観測成分から元の行列が高い確率で厳密に復元できることを証明した。Soft-Impute (Mazumder et al. 2010) は、欠損部分の補完ステップと特異値のソフト閾値処理（soft-thresholding）を交互に繰り返すことで、この問題のペナルティ付き版を解く。

33.6.3.2 CUR 分解とスケッチング

打ち切り SVD は近似誤差の面では最適であるものの、その特異ベクトルはすべての行と列を混ぜ合わせてしまうため解釈しにくく、さらに $m \times n$ 行列の完全な SVD を計算するには $O(mn \min(m,n))$ 回の演算コストがかかる。これに対し CUR 分解は、データ行列の実際の列 **C** と実際の行 **R** を取り出し、それらを小さなコア行列 **U** で結ぶことで近似を構成する。33.5.2.1 節では、ノルムが最大の列を単純に選ぶ素朴な戦略が打ち切り SVD に大きく劣ることを見た。Mahoney and Drineas (2009) はその改善策を示しており、上位の特異ベクトルへの射影の長さの二乗である**レバレッジスコア**に応じて列と行をサンプリングすれば、高い確率で Eckart–Young の誤差の小さな定数倍以内に収まる。

より広く見れば、33.5.4.3 節のランダム化値域探索（randomized range finder）(Halko et al. 2011) は**スケッチング**の 1 つの例である。すなわち、大きな行列に細長いランダム行列を掛け、はるかに小さなスケッチ上で問題を解く手法である。Woodruff (2014) は、これと同じ発想によって最小二乗法や回帰がどのように高速化され、元のデータセットよりはるかに少ない行数しか持たないスケッチから最適解の $1+\varepsilon$ 倍以内の解を返せるかを概説している。

33.6.3.3 アダプタの派生手法

LoRA（33.5.4.1 節）は、低ランク更新 **BA** を **B** $= \mathbf{0}$ およびランダムな **A** で初期化するため、最初の勾配ステップでは事前学習済みの重みとは無関係なランダムな k 次元部分空間を通してしか入力を捉えられない。いくつかの派生手法は、この初期化やパラメータ化を単純な方法で変更している。PiSSA (Meng et al. 2024) は **BA** を事前学習済みの重みのランク k 打ち切り SVD で初期化して残差を凍結するため、ファインチューニングの開始直後から元の行列の主方向を適応させることができる。DoRA (S.-Y. Liu et al. 2024) は重みの各列をスカラーの大きさと単位方向に分解し、大きさは直接訓練しつつ、低ランク更新を方向のみに適用する。

33.6.3.4 アテンションにおける低ランク構造

n トークンの系列に対する自己アテンションは $n \times n$ のスコア行列を作るため、アテンションの各章で扱う効率的なアーキテクチャのいくつかは、この行列またはそのキー・値キャッシュを低ランク因子分解（low-rank factorization）に置き換えている。線形アテンション（10.6 節）はアテンションスコアを特徴量のドット積 $\phi(\mathbf{q}) \cdot \phi(\mathbf{k})$ として表し、スコア行列のランクを特徴量の次元以下に抑える。マルチヘッド潜在アテンション（multi-head latent attention、11.3.3.2 節）(DeepSeek-AI 2024) は、ヘッドごとに別々のキーと値を保持する代わりにトークンあたり 1 つの低次元潜在ベクトルを保存することでキャッシュを圧縮する。Linformer (Wang et al. 2020) は、訓練済みアテンション行列の特異値が急速に減衰するという経験的な観測から出発し、キーと値を系列次元に沿って固定長へと射影する。

33.6.3.5 他のノルムのもとでの最急降下法

33.5.4.2 節では、勾配行列の極因子（polar factor）$\mathbf{UV}^\top$ を計算した。幾何学的には、この行列はステップサイズをユークリッドノルムではなくスペクトルノルムで測ったときの最急降下方向

である。Bernstein and Newhouse (2024) はこの知見を深層学習のオプティマイザに対する統一的な視点へと発展させた。移動平均を無効にすると、Adam は符号降下法（sign descent、すなわち ℓ_∞ ノルムのもとでの最急降下法）に帰着し、Shampoo はスペクトルノルムのもとでの最急降下法に帰着する。この視点に立てば、各層に適切なノルムを選ぶことこそが中核となる設計上の判断であり、更新則はそこから自動的に導かれる。この考え方に基づいて構築されたスペクトルノルムオプティマイザは 9.6 節で展開する。

33.6.4 書籍、講義、講義ノート

書籍

- 『Mathematics for Machine Learning』（Deisenroth、Faisal、Ong 著）：第 2 章（線形代数）、第 3 章（解析幾何）、第 4 章（行列分解）で本章の内容を機械学習の視点から網羅している無料の教科書である。
- 『Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares』（Boyd、Vandenberghe 著）：最小二乗法とデータを軸に構成されたスライド・コード付きの無料の書籍であり、33.1 節および 33.5 節とよく対応している。
- 『Introduction to Linear Algebra, 6th ed.』および『Linear Algebra and Learning from Data』（Gilbert Strang 著）：列による描像（column picture）と 4 つの基本部分空間を軸に据えた古典的な入門書と、データ解析および最適化に向けて SVD を中心に据えたその姉妹編である。
- 『Linear Algebra Done Right, 4th ed.』（Sheldon Axler 著）：行列式を使わずに固有値、作用素、およびスペクトル定理（33.3 節）を展開する無料の教科書である。
- 『Numerical Linear Algebra』（Trefethen、Bau 著）：条件性（conditioning）、安定性、QR 法、ならびに `solve`、`eig`、`svd` の背後にある浮動小数点アルゴリズムをわかりやすく解説した書籍である。

講義とチュートリアル

- 「MIT 18.06 Linear Algebra」および「MIT 18.065 Matrix Methods in Data Analysis, Signal Processing, and Machine Learning」（Gilbert Strang 担当）：講義ノートと演習問題を備えた完結した講義コースである。
- 3Blue1Brown による「Essence of Linear Algebra」および Steve Brunton による「Singular Value Decomposition」：線形変換、行列式、固有ベクトル、および SVD を視覚的に解説する講義シリーズである。
- Kolter、Do による「Linear Algebra Review and Reference」（Stanford CS229）および Ström、Åström、Akenine-Möller による『Immersive Linear Algebra』。

Chapter 34

微分積分と自動微分

本書全体を通して、ニューラルネットワークの訓練は同じリズムをたどる。損失を評価し、フレームワークにその勾配を求めさせ、すべての重みを谷の方向へ小さなステップだけ動かすという手順である。これまではその仕組みに関する 2 つの自然な問いに立ち止まることなく、このループに頼ってきた。第 1 に、学習率を 2 倍にした途端、訓練が突然発散したとする。勾配そのものは依然として正しいが、局所的な傾きで損失を確実に予測できる小さな近傍を、そのステップが飛び越えてしまったのである。この線形近似（linear approximation）はどこまで信頼できるのだろうか。第 2 に、勾配がいかに驚くほど低コストであるかに注目しよう。1 億個の重みを 1 つずつ摂動させてすべての傾きを推定するには 1 億回の順伝播（forward pass）を要するが、1 回の逆伝播の呼び出しは、およそ数回の順伝播に相当するコストですべての傾きを返す。なぜそのようなことが可能なのだろうか。

いずれの問いも微分積分に属する。微分積分とは、連続的な変化をどう測るか、そして無限に多くの微小な断片をどう足し合わせるかを Newton と Leibniz が問うた 17 世紀に生まれた分野である。その 2 つの道具である微分と積分は、互いに逆の関係にあることがわかる。深層学習において、導関数は現在の重みの近傍における損失曲面（loss surface）の最良の線形モデルを与え、損失の値そのものを除けば、オプティマイザが目にする唯一の情報である。一方、データや重みが確率的になると直ちに積分が登場する。確率や期待損失は曲線の下の面積であり、方策勾配（policy gradient、14.7 節）のような手法では、サンプリング元の分布そのものに関してその積分を微分しなければならない。

本書の序盤では、微分の基本規則を述べ（1.4 節）、自動微分を既製のツールとして扱った（1.5 節）。本編の各章ではそれ以上の知識を必要としないため、この付録の章は、その仕組みがなぜ機能し、その保証がどこから得られるのかを知りたい読者のための任意の読み物である。議論の根幹を担うのは 3 つの結果である。すなわち、曲率（curvature）の上界を 9.1 節で引用した安全なステップサイズの規則へと変換する降下補題（descent lemma）、損失側から逆向きに掛けることではるかに低コストとなり誤差逆伝播法（backpropagation、4.3 節）を導くヤコビ行列（Jacobian）の積としての連鎖律（chain rule）、そして汎用性と分散の間でトレードオフをなす、期待値の勾配に対する相補的な 2 つの推定量である。

本章は 3 つのパートで展開する。まず 1 変数（34.1 節）、続いて多変数（34.2 節）において関数の線形近似と二次近似（quadratic approximation）を構築し、安全なステップサイズに上界を与え、極小点と鞍点（saddle point）を見分けられるようにする。次に、計算グラフ（computation graph）上の経路にわたる和（34.3 節）から各層のヤコビ行列の積（34.4 節）、さらにはそれらを自動的に乗算するアルゴリズム（34.5 節）へと連鎖律をたどる。最後のパートでは積分（34.6 節）を導入し、サンプリングと微分を組み合わせて期待値の勾配を推定する（34.7 節）。

初等的な微分積分にすでに慣れている場合は、いくつかの節の冒頭にあるウォームアップのページを流し読みしてもよい。本章では、勾配を超えるベクトル解析（発散やストークスの定理など）および測度論（measure theory）は割愛する。微分方程式については第 38 章を参照されたい。

34.1 1 変数の微分積分

「基礎」パートのすべてのモデルは、勾配降下法（gradient descent）によって訓練した。各ステップにおいて、オプティマイザは現在の重みにおける損失とその傾きを評価し、重みを下り方向へ少しだけ動かして、これを繰り返す。他のどの点でも損失を調べないため、すべてのステップは局所的な予測に依拠している。すなわち、現在の点における傾きが、少し離れた位置の損失も記述していると仮定するのである。その予測はどれほど正確であり、どの程度大きなステップまで安全に支えられるのだろうか。

重みが 1 つだけの場合でも、その答えがどのような形になるかはすでに見て取れる。1 つの訓練例、1 つのスカラー重み w、およびロジスティック損失（logistic loss）$\ell(w) = \log(1 + e^{-w})$ を考えよう。$w = 0$ において、損失は 0.693147 であり、その傾きは $-\frac{1}{2}$ である。重みを $w = 0.1$ に動かすと、$w = 0$ を通りその傾きを持つ直線は、損失を $0.693147 - 0.05 = 0.643147$ と予測する。実際の損失は 0.644397 であるため、線形予測のずれはわずか 0.00125 にすぎない。注目すべきことに、この小さな差は単純な構造を持っている。表示された桁の範囲で、この差は $\frac{1}{2} \cdot \frac{1}{4} \cdot 0.1^2$ に等しく、ここで因子 $\frac{1}{4}$ は ℓ のグラフが $w = 0$ でどれだけ急激に曲がっているかを表す。傾きが予測を与え、曲がり具合がステップを大きくしたときにその予測がどれだけ速くずれていくかを示すのである。微分の公式を手にしたのち、ほどなくこれら両方の定数を導出しよう。

本節では、一階導関数（first derivative）から局所的な直線モデルを導き、さらに二階導関数（second derivative）を用いて曲がり具合も捉える放物線へと拡張する。議論の軸となるのは 3 つの結果である。点における傾きを区間上の変化へと結びつける平均値の定理（mean value theorem）、直線と放物線の両方の誤差に上界を与えるラグランジュの剰余項（Lagrange remainder）、そして曲率（curvature）の上界を勾配降下法の保証された安全なステップサイズへと変換する降下補題（descent lemma）である。さらに、ニュートン法や、接線が存在しないにもかかわらず誤差逆伝播法（backpropagation）が数値を返さなければならない ReLU の角も扱う。ここではこれらすべての概念を 1 変数で学び、多変数の最適化理論は第 35 章に委ねる。

先立つ 1.4 節では導関数（derivative）を導入し、微分の公式を列挙して、自動微分（automatic differentiation）を実用的なツールとして用いた。ここではそれらの公式を手短に復習し、証明、幾何学的直観、および誤差の上界に焦点を当てる。微分の公式にすでに慣れている場合は、34.1.3 節へ直接進んでもよい。あらゆる多変数の損失には 1 変数の構造が含まれているため、1 変数から始めるのは自然である。すなわち、$\ell(w_1, \dots, w_n)$ の重みのうち w_i 以外をすべて固定すれば 1 次元スライスが得られ、1 変数の微分積分とはそのようなスライスを調べる学問にほかならない。

本節を通して用いる具体例は、**ソフトプラス**関数 $s(x) = \log(1 + e^x)$ である。これは本章全体で使われる log-sum-exp 関数の 1 変数スライス $\log(e^0 + e^x)$ であり、冒頭のロジスティック損失は単に $\ell(w) = s(-w)$ と表せる。

34.1.1 導関数

34.1.1.1 差分商

滑らかな関数 f のグラフを点 x の近傍で十分に拡大すると、曲線は直線と見分けがつかなくなる（Figure 34.1）。その直線の傾きこそが導関数である。

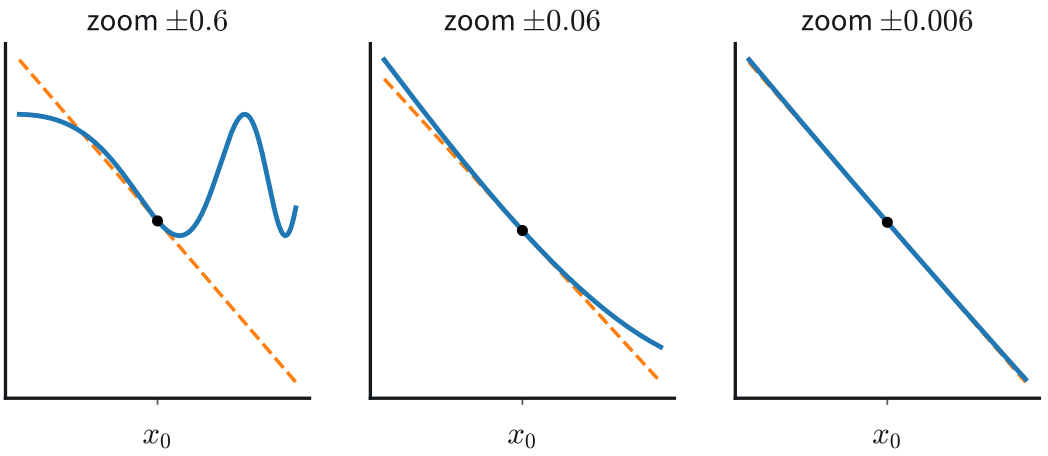


Figure 34.1: 曲線 $\sin(x^x)$ を $x_0 = 2$ の近傍で、半幅 0.6、0.06、0.006 の表示窓で示したもの。窓が狭まるにつれて、曲線（実線）はその接線（破線）と見分けがつかなくなる。

この傾きを代数的に捉えるため、1.4 節にならい、$(x, f(x))$ と $(x+\epsilon, f(x+\epsilon))$ を通る**割線**（secant line）から始める。その傾きは**差分商**（difference quotient）$\big(f(x+\epsilon) - f(x)\big)/\epsilon$ である。ϵ がゼロに向かって小さくなるにつれて、割線は接線に向かって回転する（Figure 34.2）。

割線の傾きが接線の傾きの信頼できる代用となるには、ϵ はどれほど小さくなければならないだろうか。残念ながら普遍的なしきい値は存在せず、関数に完全に依存する。多項式 $f(x) = x^2 + 1701(x-4)^3$（$x = 4$ における真の傾きは 8）を考え、ϵ が小さくなるにつれて割線の傾きがいかにゆっくりと落ち着くかを観察してみよう。

```
def f(x):
    return x**2 + 1701*(x-4)**3

# Slope of the secant through x=4, for shrinking epsilon
for epsilon in [0.1, 0.001, 0.0001, 0.00001]:
    print(f'epsilon = {epsilon:.5f} -> {(f(4+epsilon) - f(4)) / epsilon:.5f}')
```

```
epsilon = 0.10000 -> 25.11000
epsilon = 0.00100 -> 8.00270
epsilon = 0.00010 -> 8.00012
epsilon = 0.00001 -> 8.00001
```

$\epsilon = 0.1$ では、割線の傾きは 25.11 と真の極限の 3 倍以上にもなり、出力値が 8 の付近に落ち着くのは ϵ が 0.001 に達してからである。$f(4+\epsilon) - f(4) = 8\epsilon + \epsilon^2 + 1701\epsilon^3$ を展開すればその理由がわかる。差分商は $8 + \epsilon + 1701\epsilon^2$ に等しく、三次の項による寄与は 17（$\epsilon = 0.1$ のとき）に達する一方、1.7×10^{-7}（$\epsilon = 10^{-5}$ のとき）にすぎない。大きな係数は極限値そのものを変えることなく、極限への接近を遅らせる。自動微分はこの問題を完全に回避する。ϵ を選ぶことは一切せず、微分の公式を直接適用するからである（34.5 節）。

34.1.1.2 導関数と線形近似

関数 f の点 x における**導関数**は、割線の傾きの極限として次式で定義される。

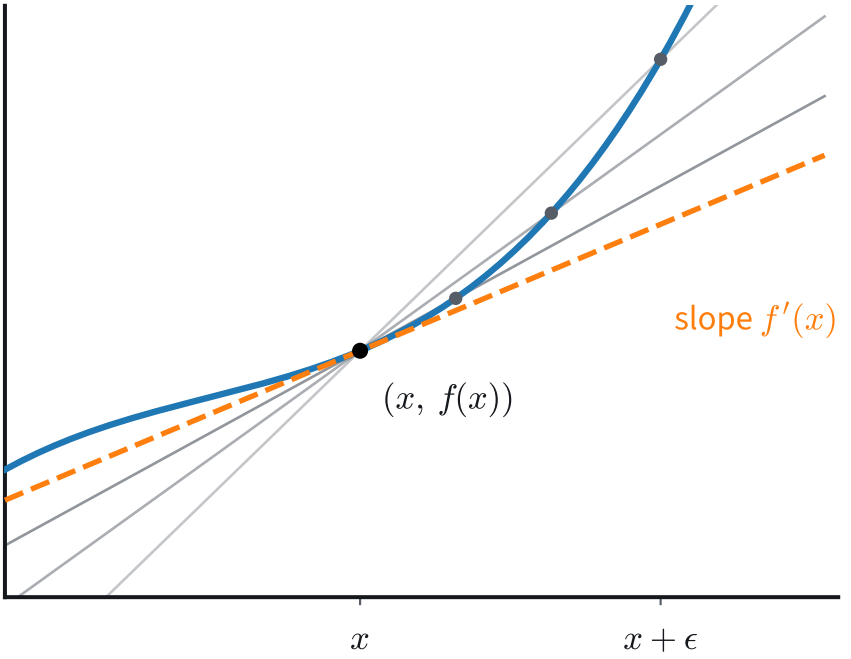


Figure 34.2: 差分商は $(x, f(x))$ と $(x+\epsilon, f(x+\epsilon))$ を通る割線の傾きである。$\epsilon \to 0$ につれて、この傾きは接線の傾き $f'(x)$ に近づく。

$$f'(x) = \frac{df}{dx}(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}, \tag{34.1}$$

ただし、これはこの極限が存在する場合に限られ、そのとき f は x で**微分可能**（differentiable）であるという。上の三次関数では $f'(4) = 8$ である。表記としては f' または $\frac{df}{dx}$ を用い、かさばる式を微分するときは $\frac{d}{dx}[...]$ と書く。また、$|x|$ の点 0 におけるように極限が存在しないこともあり、そのような角については 34.1.6 節で改めて扱う。

この定義を書き換えてみると示唆に富む。直線予測の誤差を $r(\epsilon) = f(x+\epsilon) - f(x) - \epsilon f'(x)$ と置くと、極限 Equation 34.1 は $r(\epsilon)/\epsilon \to 0$ を意味する。そこで、記法 $o(\epsilon)$ を、量 $r(\epsilon)$ のうち $r(\epsilon)/\epsilon \to 0$（$\epsilon \to 0$ のとき）を満たす任意のものを表すために用いると、次の**線形近似**（linear approximation）が得られる。

$$f(x+\epsilon) = f(x) + \epsilon f'(x) + o(\epsilon). \tag{34.2}$$

平易な言葉で言えば、微小なステップ ϵ によって出力は $\epsilon f'(x)$ だけ変化し、その誤差は ϵ より速くゼロに収束する。逆も成り立つ。すなわち、$f(x+\epsilon) = f(x) + a\epsilon + o(\epsilon)$ がある定数 a に対して成り立つならば、両辺を ϵ で割って $\epsilon \to 0$ とすることで、f が x で微分可能であり $f'(x) = a$ となることが証明される。したがって、展開 Equation 34.2 と極限 Equation 34.1 は同じ情報を持っており、和の微分公式や連鎖律の証明はこの同値性に依拠する。

$x+\epsilon$ の関数としてプロットすると、直線 $f(x)+\epsilon f'(x)$ は点 x におけるグラフの**接線**（tangent）、すなわち回転する割線の極限となる。これが $w = 0.1$ におけるロジスティック損失を予測した直

線であり、0.00125 の差がその $o(\epsilon)$ の剰余項であった。それ単体では、$o(\epsilon)$ はほとんど何も保証しない。$o(\epsilon)$ のオーダーの誤差は、ϵ よりほんのわずかに速く縮むだけかもしれないからである。しかし今回の例では差は ϵ^2 に比例して縮んでおり、34.1.5.2 節で見るように、二次の誤差減衰は 2 回微分可能なすべての関数に共通する法則である。

34.1.2 導関数の計算

実際には、極限を毎回計算して導関数を求める人はいない。その代わりに、1.4 節と同様、初等関数の導関数の短い表と、和・積・合成に関する公式を組み合わせる。ここでは各公式を線形近似から直接証明する。表の項目のうち、指数関数、対数関数、べき乗の微分公式については導出を行い、正弦・余弦の項目、および傾きが 1（点 0 において）となる指数関数の底が存在することは、微分積分の初等講義で既知のものとする。

34.1.2.1 代表的な導関数の表

以下の公式は右辺が定義されるすべての点で成り立つ。すなわち、べき乗の微分公式は実数 n のすべて（$x > 0$ のとき）および整数 n（x^n が定義されるすべての点）に対して成り立ち、対数関数は $x > 0$ で成り立つ。

- **定数の微分：** $\frac{d}{dx} c = 0$。
- **一次関数の微分：** $\frac{d}{dx}(ax) = a$。
- **べき乗の微分公式：** $\frac{d}{dx} x^n = n x^{n-1}$。
- **指数関数の微分：** $\frac{d}{dx} e^x = e^x$。
- **対数関数の微分：** $\frac{d}{dx} \log(x) = \frac{1}{x}$。
- **正弦の微分：** $\frac{d}{dx} \sin(x) = \cos(x)$。
- **余弦の微分：** $\frac{d}{dx} \cos(x) = -\sin(x)$。

指数関数の項目は、なぜ底 e が特別なのかも明らかにする 1 行の考察に基づいている。任意の $a > 0$ に対し、a^x の差分商は次のようにきれいに因数分解できる。

$$\frac{a^{x+\epsilon} - a^x}{\epsilon} = a^x \, \frac{a^\epsilon - 1}{\epsilon},$$

ここで 2 番目の因数は、単に点 0 における差分商である。したがって、a^x が点 0 で微分可能でありその傾きが c_a ならば、$\frac{d}{dx} a^x = c_a \, a^x$ がすべての x に対して成り立つ。すなわち、指数関数の傾きは常にその高さに比例し、底ごとにただ 1 つの比例定数を持つ。数値的には、$\epsilon = 10^{-6}$ において $(2^\epsilon - 1)/\epsilon = 0.69315$（底 2）および $(10^\epsilon - 1)/\epsilon = 2.30259$（底 10）が得られる。数 e はこの定数が 1 となる唯一の底として定義され、直ちに $\frac{d}{dx} e^x = e^x$ が従う。対数関数の導関数が得られれば、連鎖律によって他のすべての底に対する c_a も特定できる。

34.1.2.2 導関数を組み合わせる公式

損失関数は、上の表をもとに和、積、商、合成を用いて構築される。g と h は x で微分可能であるとする（連鎖律については、h が x で微分可能であり、g が $h(x)$ で微分可能であるとする）。このとき、次が成り立つ。

- **和の微分公式：** $\frac{d}{dx}\big(g(x) + h(x)\big) = g'(x) + h'(x)$。
- **積の微分公式：** $\frac{d}{dx}\big(g(x)\,h(x)\big) = g(x)h'(x) + g'(x)h(x)$。
- **商の微分公式：** $\frac{d}{dx}\frac{g(x)}{h(x)} = \frac{g'(x)h(x) - g(x)h'(x)}{h(x)^2}$（ただし $h(x) \neq 0$）。

- **連鎖律：** $\frac{d}{dx} g(h(x)) = g'(h(x))\, h'(x)$。

和と積の微分公式は型通りに示せるが、連鎖律だけは十分な注意を要する。素朴な議論では $h(x+\epsilon) - h(x)$ で割ることになるが、これは $\epsilon \neq 0$ であってもゼロになり得るからである。これらの証明はすべて 1 つの基本的な事実に依拠する。すなわち、**x で微分可能な関数は x で連続である**。なぜなら、$g(x+\epsilon) - g(x) = \epsilon \cdot \frac{g(x+\epsilon)-g(x)}{\epsilon}$ は $0 \cdot g'(x) = 0$ へと（$\epsilon \to 0$ のとき）収束するからである。

和の微分公式： g と h の線形近似を足し合わせると、次式が得られる。

$$g(x+\epsilon) + h(x+\epsilon) = g(x) + h(x) + \epsilon\big(g'(x) + h'(x)\big) + o(\epsilon),$$

これは、ϵ より速くゼロに収束する 2 つの量の和もまた ϵ より速くゼロに収束するためである。線形近似の逆命題により、ϵ の係数が導関数となる。すなわち、一次の変化は単に足し合わされる。■

積の微分公式： 分子において $g(x+\epsilon)h(x)$ を足して引くと、積の差分商は既知の 2 つの商へと分解される。

$$\frac{g(x+\epsilon)h(x+\epsilon) - g(x)h(x)}{\epsilon} = g(x+\epsilon)\,\frac{h(x+\epsilon) - h(x)}{\epsilon} + h(x)\,\frac{g(x+\epsilon) - g(x)}{\epsilon}.$$

$\epsilon \to 0$ のとき、連続性より $g(x+\epsilon) \to g(x)$ となるため、右辺は $g(x)h'(x) + h(x)g'(x)$ に収束する。■

連鎖律： $h(x+\epsilon) - h(x)$ で割る代わりに、内側の関数の変化を外側の関数の線形近似へ直接代入する。g は $h(x)$ で微分可能であるため、誤差比 $\rho(t) = \frac{g(h(x)+t) - g(h(x))}{t} - g'(h(x))$（$t \neq 0$ に対して定義し、$\rho(0) = 0$ と拡張する）は 0 へと（$t \to 0$ のとき）収束する。したがって、任意の t（$t = 0$ を含む）に対して次のように書ける。

$$g(h(x) + t) = g(h(x)) + \big(g'(h(x)) + \rho(t)\big)\, t.$$

ここで $t = h(x+\epsilon) - h(x) = \epsilon h'(x) + o(\epsilon)$ と置く。h の連続性により、$t \to 0$（$\epsilon \to 0$ のとき）となるため、$\rho(t) \to 0$ も成り立つ。この t を代入すると次式が得られる。

$$\begin{aligned} g(h(x+\epsilon)) - g(h(x)) &= \big(g'(h(x)) + \rho(t)\big)\big(\epsilon h'(x) + o(\epsilon)\big) \\ &= \epsilon\, g'(h(x))\, h'(x) + o(\epsilon), \end{aligned}$$

なぜなら、$g'(h(x))\, \epsilon h'(x)$ 以外のすべての項は、ϵ より速くゼロに収束する因数を含んでいるからである。線形近似の逆命題を適用すれば証明が完了する。■

合成関数に沿うと変化率は掛け合わされる。連鎖律をニューラルネットワークに層ごと適用すると局所的な傾きの積が生じ、その積を効率的に評価することこそが誤差逆伝播法の役割にほかならない。商の微分公式は、$g \cdot h^{-1}$ に積の微分公式と連鎖律を適用すれば直ちに従う。

34.1.2.3 逆関数

連鎖律は、別の関数の働きを打ち消す関数も微分できる。関数 g が $g(f(x)) = x$ をある点の近傍のすべての x に対して満たすとき、それをその点における f の**逆関数**（inverse）という。そのような g が存在し、$f(x)$ で微分可能であるとしよう。ここでは証明なしに用いる逆関数定理（inverse function theorem）によれば (Rudin 1976)、f' が x の近傍で連続かつ非ゼロであれば、この両方が成り立つ。$g(f(x)) = x$ の両辺を連鎖律で微分すると、次式が得られる。

$$g'(f(x))\,f'(x) = 1, \qquad \text{so} \qquad g'(y) = \frac{1}{f'(g(y))} \quad \text{at } y = f(x).$$

言い換えれば、逆関数の傾きは、対応する点で評価した f の傾きの逆数である。幾何学的には、g のグラフは f のグラフを対角線に関して折り返したものであり、直線を対角線に関して折り返すと縦の変化と横の変化の比が上下逆転する。

対数関数は e^x を打ち消すため、この公式から $\log'(y) = 1/e^{\log y} = 1/y$、すなわち先ほどの表の 5 番目の項目が得られる。なお、条件 $f'(x) \neq 0$ は不可欠である。$f(x) = x^3$ では $f'(0) = 0$ となるため、$y = 0$ でこの式は破綻する。実際、その逆関数 $x^{1/3}$ は点 0 で微分可能ではない。差分商 $\epsilon^{1/3}/\epsilon = \epsilon^{-2/3}$ が限りなく大きくなり、その点でグラフが垂直な接線を持つからである。

対数関数の微分が手に入ったことで、連鎖律によって表の残り 2 つの項目も解決する。$a^x = e^{x \log a}$ と書けば $\frac{d}{dx} a^x = (\log a)\, a^x$ が得られ、定数が $c_a = \log a$ であることがわかる。実際、$\log 2 = 0.69315$ と $\log 10 = 2.30259$ は先ほど数値的に測定した定数そのものである。同様に、$x^n = e^{n \log x}$（$x > 0$ のとき）と書けば、$\frac{d}{dx} x^n = e^{n \log x} \cdot \frac{n}{x} = n x^{n-1}$ がすべての実数 n に対して得られる。

34.1.2.4 ソフトプラス、シグモイド、ロジット

これで、本節の具体例である関数 $s(x) = \log(1 + e^x)$ にこれらの公式を適用できる。外側の関数を log、内側の関数を $1 + e^x$ として連鎖律を用いると、次式が得られる。

$$s'(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}} = \sigma(x),$$

したがって、ソフトプラスの導関数は**シグモイド** σ である。$\sigma(x) = (1 + e^{-x})^{-1}$ に対して、外側の関数を u^{-1}、内側の関数を $1 + e^{-x}$（傾きは $-e^{-x}$）として 2 回目の連鎖律を適用すると、次式が得られる。

$$\sigma'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = \sigma(x)(1 - \sigma(x)),$$

これは $1 - \sigma(x) = e^{-x}/(1 + e^{-x})$ だからである。積 $p(1-p)$ は $p = \frac{1}{2}$ で最大となるため、シグモイドは原点で最も急になり、そこでは $\sigma(0) = \frac{1}{2}$ かつ $\sigma'(0) = \frac{1}{4}$ である。σ の逆関数は**ロジット**関数 $\operatorname{logit}(p) = \log \frac{p}{1-p}$ であり、逆関数の微分公式から直ちに $\operatorname{logit}'(p) = 1/\sigma'(\operatorname{logit} p) = \frac{1}{p(1-p)}$ が得られる。$p = \frac{1}{2}$ ではこの値は $4 = 1/\frac{1}{4}$ となり、シグモイドの傾きの逆数と一致する。

ここで、冒頭のロジスティック損失の例における 2 つの定数を確かめることができる。$\ell(w) = s(-w)$ であるから、連鎖律より $\ell'(w) = -\sigma(-w)$ となり、$\ell'(0) = -\frac{1}{2}$ が従う。もう一度微分すると $\ell''(w) = \sigma(-w)\big(1 - \sigma(-w)\big)$ となり、原点での曲率 $\ell''(0) = \frac{1}{4}$ が得られる。$\ell(\epsilon)$ とそ

の線形近似 $\ell(0) - \frac{1}{2}\epsilon$ の差が、実際に ϵ^2 に比例して先頭係数 $\frac{1}{2}\ell''(0) = \frac{1}{8}$ を伴い縮んでいくかを数値的に確認してみよう。

```
# The logistic loss l(w) = log(1 + e^{-w}) near w = 0, where l'(0) = -1/2
def ell(w):
    return np.log(1 + np.exp(-w))

for eps in [0.1, 0.01, 0.001]:
    line = ell(0) - 0.5 * eps          # the linear approximation
    gap = ell(eps) - line
    print(f'eps = {eps:<5}: l = {ell(eps):.6f}, line = {line:.6f}, '
          f'gap = {gap:.2e}, gap / eps^2 = {gap / eps**2:.4f}')
```

```
eps = 0.1  : l = 0.644397, line = 0.643147, gap = 1.25e-03, gap / eps^2 = 0.1249
eps = 0.01 : l = 0.688160, line = 0.688147, gap = 1.25e-05, gap / eps^2 = 0.1250
eps = 0.001: l = 0.692647, line = 0.692647, gap = 1.25e-07, gap / eps^2 = 0.1250
```

確かに、ϵ を 10 分の 1 にするたびに差は 100 分の 1 に減少し、gap/ϵ^2 は $0.125 = \frac{1}{2}\ell''(0)$ へと落ち着く。線形近似の誤差は確かに二次であり、その先頭係数は曲率によって決まるのである。また、1 行目の出力が 0.1249 であり 0.1250 ではないことにも注目しよう。差は厳密には $\epsilon^2/8$ ではなく、このわずかな違いを生む高次の項については 34.1.5.2 節で特定する。

これまでの計算はすべて、1 つの点の無限小の近傍におけるものであった。傾きから区間全体にわたる関数の振る舞いについて何がわかるかという問いは、また別の種類の問題である。

34.1.3 平均値の定理

最適化では日常的に 2 つの直観に従って行動する。傾きが正ならば関数は上り坂にあり、傾きがゼロになる点は最小点の候補であるという直観である。勾配降下法はこの両方に依拠している。しかし、いずれの主張も個々の点における導関数から区間上の f の振る舞いを推論するものであり、導関数の定義自体に組み込まれているわけではない。まずはより単純な 2 番目の主張から証明しよう。

補題（Fermat） 関数 f が区間の内点 ξ で極大値または極小値をとり、かつ ξ で微分可能であるならば、$f'(\xi) = 0$ である。

証明：極大値をとる点では、$f(\xi + \epsilon) - f(\xi) \le 0$ が正負いずれの符号でも十分に小さいすべての ϵ に対して成り立つ。したがって差分商は、≤ 0（$\epsilon > 0$ のとき）かつ ≥ 0（$\epsilon < 0$ のとき）となる。f は ξ で微分可能であるため、両方の片側極限はともに $f'(\xi)$ に等しくなければならず、同時に ≤ 0 かつ ≥ 0 である数はゼロしかない。極小値の場合も、不等号の向きを逆にすれば同じ議論から従う。■

1 番目の直観を示すには、より深い道具である平均値の定理が必要になる。本節の残りで導く誤差の上界はすべて、適切に選んだ補助関数にこの定理を適用することで得られる。幾何学的には（Figure 34.3）、区間 $[a, b]$ 上のグラフの両端を結ぶ弦（chord）が、ある内点における接線と平行になることを主張する定理である。

$f(x) = x^2$ の区間 $[0, 2]$ 上といった単純な例では、弦の傾きは $(4 - 0)/2 = 2$ であり、接線の傾き $f'(\xi) = 2\xi$ は中点 $\xi = 1$ でこれと一致する。ただし、この定理にはすべての内点での微分可能性が真に必要であることに注意しよう。絶対値 $|x|$ は $[-1, 1]$ 上で $f(-1) = f(1)$ を満たすためその弦は水平であるが、傾き 0 を持つ内点は存在しない。傾きは左側で -1、右側で $+1$ であり、点 0 では定義されないからである。

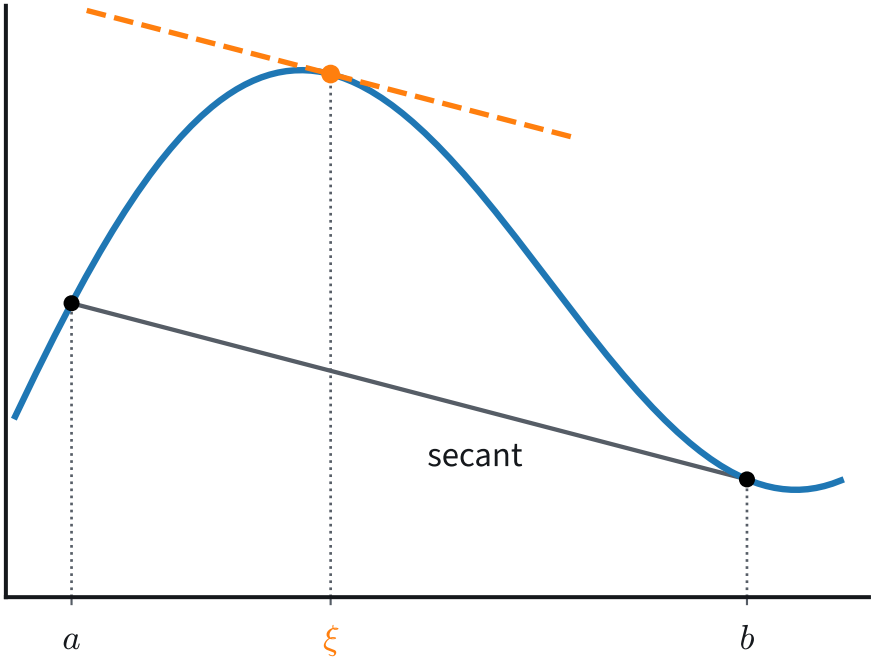


Figure 34.3: a から b までの区間において、両端を通る割線の弦の傾きは平均変化率に等しい。平均値の定理は、その弦と平行な同じ傾きの接線を持つ内点 ξ が存在することを保証する。

命題（平均値の定理） f **が** $[a,b]$ **上で連続かつ** (a,b) **上で微分可能であるならば、次式を満たす点** $\xi \in (a,b)$ **が存在する。**

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}. \tag{34.3}$$

証明：まず、ロルの定理（Rolle's theorem）として知られる両端の高さが等しい場合 $f(a) = f(b)$ を考える。ここで実解析の基礎的な結果である**最大値・最小値の定理**（extreme value theorem）、すなわち閉区間上の連続関数は最大値と最小値の両方をとるという事実を用いる (Rudin 1976)。もし両方の極値が端点で達成されるならば、f は定数であり内部のいたるところで $f' \equiv 0$ となる。そうでなければ、少なくとも 1 つの極値は内点 ξ で達成され、フェルマーの補題（Fermat's lemma）によりそこで $f'(\xi) = 0$ となる。

一般の場合には、弦を引いて両端の高さを揃える。傾きを補正した関数 $\tilde{f}(x) = f(x) - \frac{f(b)-f(a)}{b-a}(x-a)$ を定義しよう。$\tilde{f}(a) = \tilde{f}(b) = f(a)$ であるから、ロルの定理によりある内点 ξ で $\tilde{f}'(\xi) = 0$ となり、これを整理すれば直ちに Equation 34.3 が得られる。$\blacksquare$

言葉で言い換えれば、区間上の平均変化率は必ずその内部のある点における瞬間変化率として実現される。ここから導かれる 2 つの帰結は、本章全体で繰り返し登場する。

1. **ある区間上で** $f' > 0$ **ならば、その区間上で** f **は狭義単調増加である。**区間内の任意の $x_1 < x_2$ に対し、Equation 34.3 より $f(x_2) - f(x_1) = f'(\xi)(x_2 - x_1) > 0$ が両者の間にある何らかの ξ に対して成り立つ。同様に、$f' < 0$ ならば f は狭義単調減少であり、$f' = 0$ ならば f は定数である。

2. **$f' < 0$ が $(x_0 - \delta, x_0)$ 上で成り立ち、$f' > 0$ が $(x_0, x_0 + \delta)$ 上で成り立ち（ある $\delta > 0$ に対して）、さらに f が x_0 で連続であるならば、x_0 は狭義の極小点である。**それぞれの側に 1 つ目の帰結を適用すれば、f は左側から x_0 に向かって減少し、そこから右側へ向かって増加することがわかる。

通し例の関数に立ち返ると、いたるところで $s' = \sigma > 0$ であるから、ソフトプラスは $\mathbb{R}$ 全体で狭義単調増加である。

34.1.4 1 変数の勾配降下法

34.1.4.1 勾配降下法のステップ

線形近似は、関数 f が微小なステップ ϵ に対してどう応答するかを $f(x + \epsilon) \approx f(x) + \epsilon f'(x)$ と予測する。f を減少させるには、ϵ の符号を傾きと逆向きに選べばよい。$\epsilon = -\eta f'(x)$ と正の**ステップサイズ** $\eta > 0$ を用いて置くと、線形モデルは次の変化を予測する。

$$f(x - \eta f'(x)) - f(x) \approx -\eta\, f'(x)^2,$$

この値は $f'(x) \neq 0$ である限り真に負である。傾きが正か負かにかかわらず、それと逆向きに進めば、線形モデルは傾きの二乗に比例する量だけ減少する（Figure 34.4）。

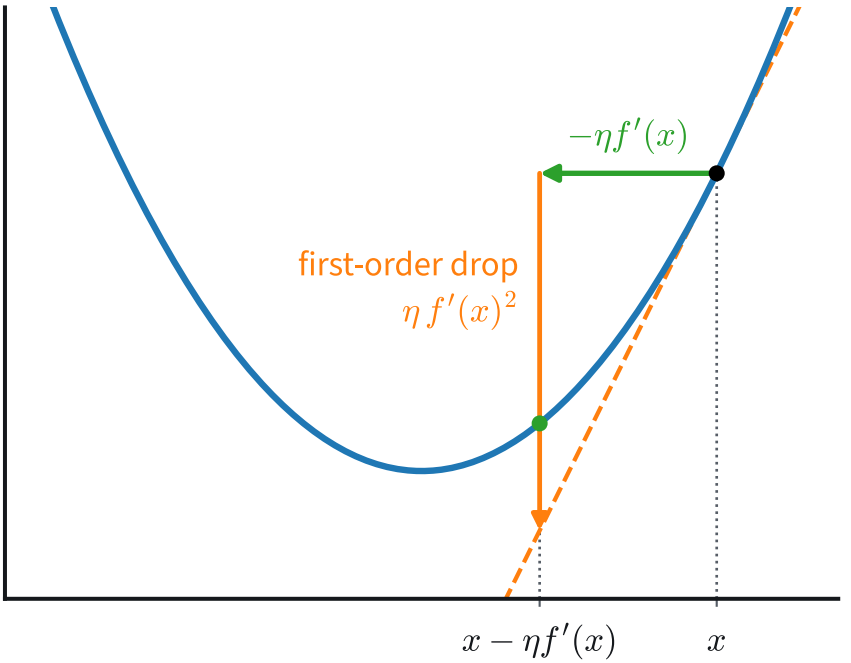


Figure 34.4: 1 次元の谷における勾配降下法の 1 ステップ。点 x における接線の傾きは $f'(x)$ であり、ステップ $-\eta f'(x)$ は傾きと逆向きに動いて曲線上のより低い点 $x - \eta f'(x)$ に到達し、一次近似での減少量は $\eta f'(x)^2$ に等しい。

この更新を反復すると、1 次元の**勾配降下法**が得られる。

$$x_{t+1} = x_t - \eta\, f'(x_t). \tag{34.4}$$

各ステップは局所的な傾きに逆らって進み、反復は**停留点**（stationary point）、すなわち $f'(x_t) = 0$ となる点で静止する。ニューラルネットワークでは、重みの更新式 $w_i \leftarrow w_i - \eta\, \partial \ell / \partial w_i$ の各座標がこの形をとり、すべての座標が同時に更新される。ロジスティック損失において $w = 0$ から出発すると、傾きは $\ell'(0) = -\frac{1}{2}$ であるから、大きさ η のステップは重みを $\epsilon = \eta/2$ だけ動かし、損失が $\eta/4$ 減少すると予測する。

ここで、線形モデルの明白な限界に注意しよう。予測される変化量 $-\eta f'(x)^2$ は、$\eta > 0$ がどれほど巨大であっても常に負になる。実際には、ステップが大きすぎると最小点を飛び越えてしまい、出発点よりも高い位置に着地することがある。接線は曲線がどう曲がっているかをまったく知らないため、この危険を警告できない。最も単純な谷である $f(x) = x^2$ を見るだけでも、線形予測がどこで破綻するかが明らかになる。

34.1.4.2 二次関数の場合

$f(x) = x^2$ の場合、傾きは $f'(x) = 2x$ であるから、勾配降下法の更新式は次のように簡約される。

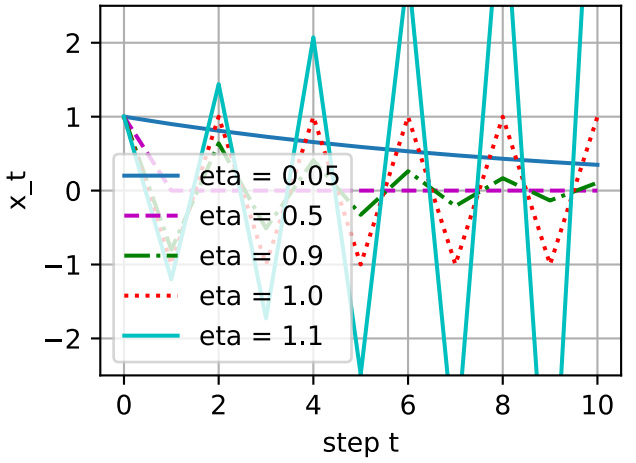
$$x_{t+1} = x_t - \eta\,(2x_t) = (1 - 2\eta)\, x_t.$$

各ステップは現在の反復点に同じ因子 $1 - 2\eta$ を掛けるため、閉じた式 $x_t = (1-2\eta)^t x_0$ が得られる。したがって、反復点列が最小点 $x = 0$ に収束するのは $|1 - 2\eta| < 1$、すなわち $0 < \eta < 1$ のとき、かつそのときに限る。この収束範囲のなかでも、小さい η では単調にじわじわと 0 へ近づき、$\eta = \frac{1}{2}$ では 1 ステップで最小点に到達し、$\frac{1}{2} < \eta < 1$ では最小点を飛び越えながら内側へ振動する。ちょうど境界の $\eta = 1$ では反復点は $+x_0$ と $-x_0$ の間を永久に往復し、$\eta > 1$ では発散する。

$x_0 = 1$ から各領域を代表する 5 通りのステップサイズについて 10 ステップ実行し、これらの軌跡を並べて見てみよう。

```
# Gradient descent on f(x) = x^2 from x0 = 1, for several step sizes
def gd(eta, steps=10, x0=1.0):
    xs = [x0]
    for _ in range(steps):
        xs.append(xs[-1] - eta * (2 * xs[-1]))  # x <- x - eta * f'(x)
    return xs

etas = [0.05, 0.5, 0.9, 1.0, 1.1]
trajectories = [gd(eta) for eta in etas]
for eta, xs in zip(etas, trajectories):
    print(f'eta = {eta:.2f} -> x_10 = {xs[-1]:+.5f}')
d2l.plot(list(range(11)), trajectories, 'step t', 'x_t', ylim=[-2.5, 2.5],
         legend=[f'eta = {eta}' for eta in etas],
         fmts=('-', 'm--', 'g-.', 'r:', 'c-'))
```

```
eta = 0.05 -> x_10 = +0.34868
eta = 0.50 -> x_10 = +0.00000
eta = 0.90 -> x_10 = +0.10737
eta = 1.00 -> x_10 = +1.00000
eta = 1.10 -> x_10 = +6.19174
```



出力された値は閉じた式のべき乗 $(1-2\eta)^{10}$ と一致している。慎重なステップ $\eta = 0.05$ はまだ 0.349 までしか下りておらず、$\eta = 0.9$ はジグザグに進んで 0.107 まで下がり、$\eta = 1$ はちょうど 1 に戻り、$\eta = 1.1$ はすでに 6.19 へと発散している。安定性のしきい値 $\eta = 1$ が $2/f''$ に等しいことにも注目しよう（$f'' \equiv 2$ が x^2 で成り立つためである）。34.1.5.4 節で見るように、これは偶然ではない。曲率の任意の上界 L から、すべての滑らかな関数に対して安全なステップサイズのしきい値 $2/L$ が導かれるのである。

34.1.5 曲率とテイラー級数

線形近似はどこまで信頼でき、どの程度大きなステップサイズなら安全なのだろうか。どちらの問いも、傾きそのものがどれだけ速く変化するかにかかっている。

34.1.5.1 二階導関数

f' それ自体が x の関数であるため、もう一度微分できる。その結果得られるのが**二階導関数** $f'' = \frac{d^2 f}{dx^2} = \frac{d}{dx}\left(\frac{df}{dx}\right)$ であり、傾きの変化率を測る。この操作を n 回繰り返すと、次の n 階導関数が得られる。

$$f^{(n)}(x) = \frac{d^n f}{dx^n} = \left(\frac{d}{dx}\right)^n f.$$

幾何学的には、$f'' > 0$ は傾きが増加していることを意味し、グラフはお椀のように上向きに曲がる。一方、$f'' < 0$ は傾きが減少していることを意味し、グラフはドームのように下向きに曲がる。停留点において、この符号は極小値と極大値を見分ける手がかりとなる。

命題（二階微分判定法） $f'(x_0) = 0$ **とし、** $f''(x_0)$ **が存在すると仮定する。** $f''(x_0) > 0$ **ならば** x_0 **は狭義の極小点であり、** $f''(x_0) < 0$ **ならば狭義の極大点である。**

証明： $f'(x_0) = 0$ であるから、$f''(x_0)$ の定義は $f''(x_0) = \lim_{\epsilon \to 0} f'(x_0 + \epsilon)/\epsilon$ に簡約される。この極限が正ならば、商 $f'(x_0 + \epsilon)/\epsilon$ は十分に小さいすべての $\epsilon \neq 0$ に対して正でなければならない。ゆえに $f'(x_0 + \epsilon)$ は ϵ と同じ符号を持ち、x_0 のすぐ左側では負、すぐ右側では正となる。平均値の定理の 2 番目の帰結により、x_0 は狭義の極小点である。$f''(x_0) < 0$ の場合も、すべての符号を反転させれば同様に示せる。■

この証明において本質的な役割を果たした箇所に注目しよう。f' の局所的な符号パターンを区間上の最小性へと結びつけるには、平均値の定理が不可欠なのである。また、この判定法が保証するのは x_0 の近傍のみであり、定義域全体ではないことにも留意されたい。多変数へ進むと、二階偏導関数のなす行列の固有値（eigenvalue）の符号がこれと同じ役割を果たすことになる。

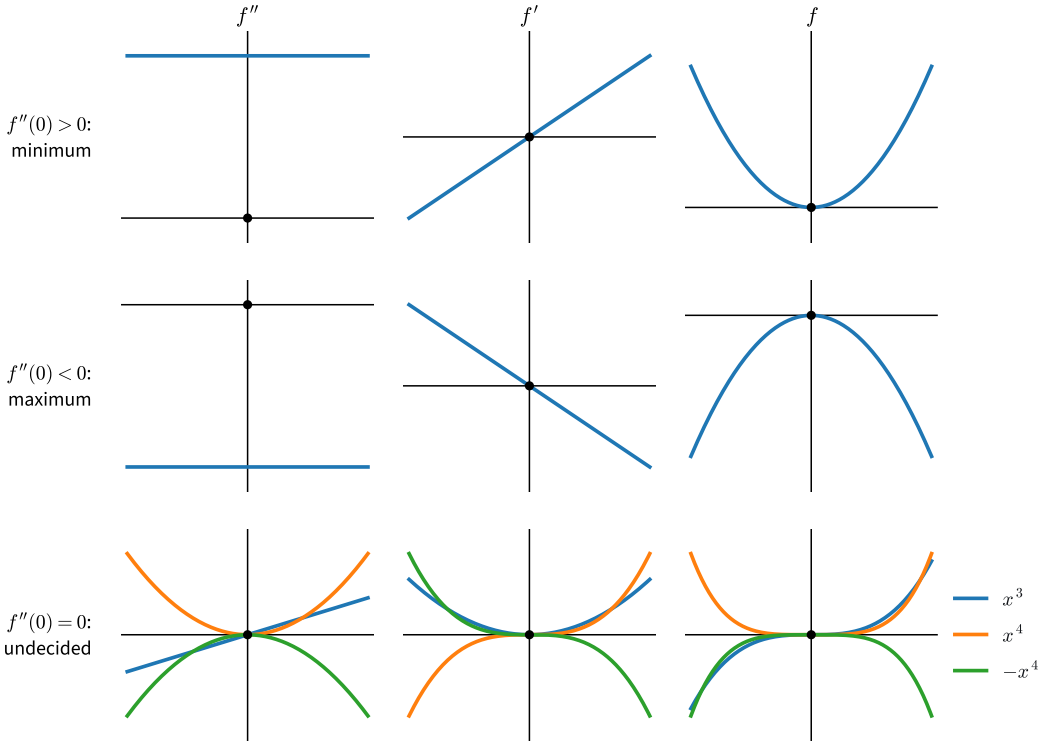


Figure 34.5: 二階微分判定法。各行は上から $f''(0) > 0$（お椀型、極小）、$f''(0) < 0$（ドーム型、極大）、および $f''(0) = 0$ を示し、最下行では x^3、x^4、$-x^4$ がいずれも $f'(0) = f''(0) = 0$ を満たしながら、点 0 でそれぞれ変曲点、極小、極大をとる。各列は左から f''、f'、f である。

停留点において $f''(x_0) = 0$ となるとき、この判定法は何も語らない（Figure 34.5 下段）。x^3、x^4、$-x^4$ はいずれも $f'(0) = f''(0) = 0$ を満たすが、点 0 は 1 番目の関数では変曲点（inflection point）、2 番目では極小点、3 番目では極大点である。このような平坦な停留点の判定には高階導関数が必要となる。

34.1.5.2 テイラー多項式

接線は、関数 f の基点 x_0 における値と傾きに一致する。曲がり具合まで捉えるには、放物線を当てはめればよい。一般の二次関数 $q(x) = ax^2 + bx + c$ の導関数は次の通りである。

$$q'(x) = 2ax + b, \qquad q''(x) = 2a,$$

したがって、その曲率は定数 $2a$ である。f の x_0 における曲率に一致させると $a = f''(x_0)/2$ と定まり、値と傾きに一致させると b と c が定まる。$f(x) = \sin(x)$ では $\sin' = \cos$ かつ $\sin'' = -\sin$ であるから、x_0 において最もよく当てはまる放物線は次式となる。

$$q(x) = \sin(x_0) + \cos(x_0)\,(x - x_0) - \tfrac{1}{2}\sin(x_0)\,(x - x_0)^2.$$

この放物線は正弦曲線の曲がりに追随するため、接線よりも目に見えて広い区間にわたって精度を保つ（Figure 34.6、放物線が非ゼロの曲率を持つよう $\sin x_0 \neq 0$ となる基点で描画している）。

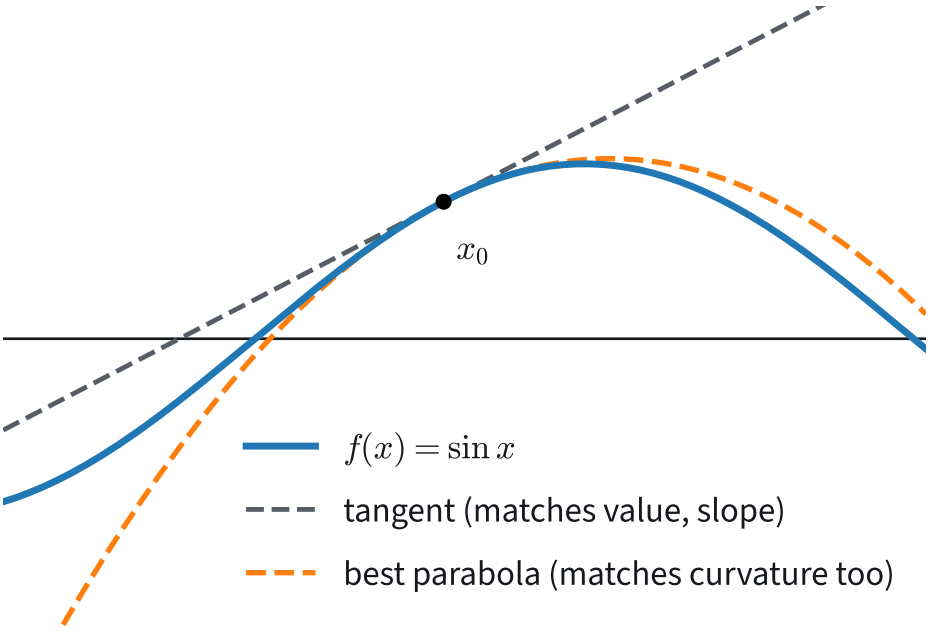


Figure 34.6: 正弦曲線と x_0 における 2 つの局所モデル。接線は値と傾きに一致し、最良の放物線はさらに曲率にも一致してより広い区間で正確さを保つ。

通し例の関数では、$x_0 = 0$ における値 $s(0) = \log 2$、$s'(0) = \frac{1}{2}$、$s''(0) = \frac{1}{4}$ から最良の放物線 $\log 2 + \frac{x}{2} + \frac{x^2}{8}$ が得られる。その二次の係数 $\frac{1}{8}$ は、ロジスティック損失において gap$/\epsilon^2$ が収束していった厳密な定数（$\ell(w) = s(-w)$ に対するもの）にほかならない。

同じ考え方は任意の次数 n へと拡張できる。次数 n の、関数 f の点 x_0 における**テイラー多項式**（Taylor polynomial）は次式で定義される。

$$P_n(x) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i,$$

これは高々 n 次の多項式のうち、値および最初の n 階までの導関数が f と点 x_0 において一致する唯一のものである。階乗が現れるのは、$(x - x_0)^i$ をちょうど i 回微分して x_0 で評価すると $i!$ が生じ、他のすべての項がゼロになるからである。この記法では、P_1 が接線、P_2 が最良の放物線にあたる。e^x の $x_0 = 0$ においてはすべての導関数が 1 に等しいため $P_n(x) = \sum_{i=0}^{n} x^i/i!$ が得られ、新しい項を加えるたびに多項式は原点からより遠くまで指数曲線に寄り添うようになる。P_n をどこまで信頼でき、その範囲を超えるとどれだけ速く誤差が増大するかを知るには、剰余項の明示的な公式が必要である。

34.1.5.3 ラグランジュの剰余項

平均値の定理を用いると、P_n の誤差を表す厳密な公式が、ただ 1 つの未知の内点を用いて得られる。これは本節の中心をなす結果であり、降下補題もニュートン法の二次収束率（convergence rate）も、その $n = 1$ の場合から直ちに導かれる。

命題（ラグランジュの剰余項） 関数 f が $(n+1)$ 回微分可能であるような開区間が x_0 と x（ただし $x \neq x_0$）を含むならば、次式が成り立つ。

$$f(x) = P_n(x) + R_n(x), \qquad R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\,(x - x_0)^{n+1}, \tag{34.5}$$

ここで ξ は x_0 と x の真の間にある何らかの点である。なお、$n = 0$ の場合は平均値の定理 $f(x) = f(x_0) + f'(\xi)(x - x_0)$ を復元する。

証明： 唯一の巧みな工夫は適切な補助関数を選ぶことであり、それ以降は素直な微分計算にすぎない。x を固定し、基点 t を変数とみなす。すなわち、t（x_0 と x の間）に対して次式を定義する。

$$G(t) = f(x) - \sum_{k=0}^{n} \frac{f^{(k)}(t)}{k!}\,(x - t)^k - M\,\frac{(x - t)^{n+1}}{(n+1)!},$$

これは点 x における（t を中心とした）テイラー多項式の誤差から、その誤差がとると予想される形の項を引いたものである。定数 M を $G(x_0) = 0$ となるように選ぶ。これで証明は 3 つのステップで完結する。

第 1 に、G は両端点でゼロになる。M の選び方から $G(x_0) = 0$ が成り立ち、また $G(x) = 0$ も成り立つ（$(x-t)$ の正のべき乗はすべて $t = x$ でゼロになり、$k = 0$ の項からは $f(x) - f(x) = 0$ が残るためである）。したがってロルの定理により、ある点 ξ が x_0 と x の真の間に存在して $G'(\xi) = 0$ を満たす。

第 2 に、$G(t)$ を t に関して微分する。積の微分公式により、$k \geq 1$ の各項からは次式が生じる。

$$-\frac{f^{(k+1)}(t)}{k!}(x - t)^k + \frac{f^{(k)}(t)}{(k-1)!}(x - t)^{k-1}$$

これが $G'(t)$ への寄与となり、一方 $k = 0$ の項からは $-f'(t)$ が生じる。隣り合う項どうしが対になって打ち消し合い（望遠鏡和）、最高階の導関数と M の項の導関数だけが残る。

$$G'(t) = -\frac{f^{(n+1)}(t)}{n!}\,(x - t)^n + M\,\frac{(x - t)^n}{n!}.$$

第 3 に、$t = \xi$ で評価する。ξ は x_0 と x の真の間にあるため $(x - \xi)^n \neq 0$ であり、ゆえに $G'(\xi) = 0$ から $M = f^{(n+1)}(\xi)$ が定まる。これを $G(x_0) = 0$ に戻せば Equation 34.5 が得られる。■

この剰余項はテイラー多項式の次の項とまったく同じ形をしているが、ただ $(n+1)$ 階導関数が未知の内点 ξ で評価される点だけが x_0 での評価と異なっている。したがって、$f^{(n+1)}$ が x_0 の近傍で有界である限り、誤差は $|x - x_0|^{n+1}$ のオーダーで変化する。すなわち、一致させる導関数が 1 階増えるごとに近似精度の次数が 1 つ上がるのである。

これを通し例の関数で試してみよう。ソフトプラスでは、$s''' = \sigma(1-\sigma)(1-2\sigma)$ が点 0 でゼロになり、$s''''(0) = -\frac{1}{8}$ となるため、次の四次多項式が得られる。

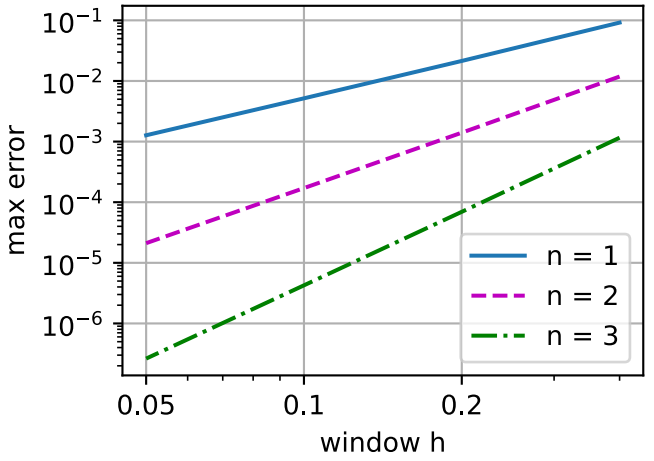
$$P_4(x) = \log 2 + \frac{x}{2} + \frac{x^2}{8} - \frac{x^4}{192}.$$

$x = 0.5$ で評価するとこの値は 0.974072 となり、真の値 $s(0.5) = 0.974077$ に対する誤差はわずか 5×10^{-6} である。実際、この誤差は x^5 による見積もりが示唆するものよりもさらに小さい。なぜなら $s(x) - \frac{x}{2} = \log(2\cosh\frac{x}{2})$ は偶関数であるため、より高次の奇数次の項はすべて消え、次の非ゼロの項は六次になるからである。この四次の項は、先ほどのスクリプトが 0.1249 と出力し 0.1250 にはならなかった理由（$\epsilon = 0.1$ のとき）も説明してくれる。そこで見えていたのは $\frac{1}{8} - \frac{\epsilon^2}{192} = 0.12495$ だったのである。

h^{n+1} のスケーリングはコードで直接確認できる。P_n の区間 $|x| \le h$ 上での最悪誤差が h^{n+1} のように振る舞うならば、h を半分にすると最大誤差は 2^{n+1} で割られる、すなわち接線では 4、放物線では 8、三次多項式では 16 で割られるはずである。この予測を e^x の $x_0 = 0$ において確かめてみよう。

```
# Max error of P_n for e^x on |x| <= h. Lagrange predicts error ~ h^(n+1),
# so halving h should divide it by 2^(n+1)
def taylor_error(n, h, m=1000):
    x = h * np.arange(-m, m + 1) / m
    P = sum(x**k / math.factorial(k) for k in range(n + 1))
    return np.max(np.abs(np.exp(x) - P))

hs = [0.4, 0.2, 0.1, 0.05]
errs = [[taylor_error(n, h) for h in hs] for n in (1, 2, 3)]
for n, e in zip((1, 2, 3), errs):
    ratios = ' '.join(f'{a / b:.1f}' for a, b in zip(e, e[1:]))
    print(f'n = {n}: ratios between successive windows {ratios}, '
          f'predicted {2**(n + 1)}')
d2l.plot(hs, errs, 'window h', 'max error', xscale='log', yscale='log',
         legend=['n = 1', 'n = 2', 'n = 3'])
```

```
n = 1: ratios between successive windows 4.3 4.1 4.1, predicted 4
n = 2: ratios between successive windows 8.4 8.2 8.1, predicted 8
n = 3: ratios between successive windows 16.7 16.3 16.2, predicted 16
```



実測された比は確かに 4、8、16 の近くに並び、両対数プロット上では 3 本の誤差曲線が傾き 2、3、4 の直線として現れ、$n+1$ と一致する。広い窓で比が理論値をわずかに上回るのは、剰余項に含まれる因子 e^{ξ} がそこではより大きくなるためである。

ラグランジュの剰余項は、無限テイラー級数がいつ元の関数自身を再現するかという問いにも答えてくれる。e^x ではすべての導関数が e^x に等しいため、Equation 34.5 を $x_0 = 0$ の周りで適用すると次式が得られる。

$$\left| e^x - P_n(x) \right| = \frac{e^{\xi}\,|x|^{n+1}}{(n+1)!} \le \frac{e^{|x|}\,|x|^{n+1}}{(n+1)!}.$$

任意の固定された x に対し、階乗はいずれべき乗よりも速く増大するため、$n \to \infty$ のとき右辺はゼロに収束する。ゆえに次式が成り立つ。

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots$$

これは $\mathbb{R}$ 全体で成立する。ただし注意が必要である。無限回微分可能であっても、関数がそのテイラー級数と一致するとは限らない。$f(x) = e^{-1/x^2}$（$f(0) = 0$ と拡張したもの）では、点 0 においてすべての導関数がゼロになるため、原点におけるテイラー級数は恒等的にゼロとなり、f とは $x = 0$ でしか一致しない（Figure 34.7）。幸いなことに、以降の議論で無限級数に依存する箇所は 1 つもない。扱うのは有限次のテイラー多項式 P_n のみであり、その誤差は Equation 34.5 によって直接抑えられる。

34.1.5.4 降下補題

曲率に上界があれば、x^2 のときと同様に、勾配降下法が任意の滑らかな関数を減少させることは保証されるのだろうか。$n = 1$ のラグランジュの剰余項は、この問いに肯定的な答えを与える。本節のすべての結果のなかで、最適化理論が最も頻繁に用いるのがこの補題である。

命題（降下補題）関数 f が開区間 I（x と $x - \eta f'(x)$ を含む）上で 2 回微分可能であり、$|f''| \le L$ が I 上で成り立つとする。このとき、次式が成り立つ。

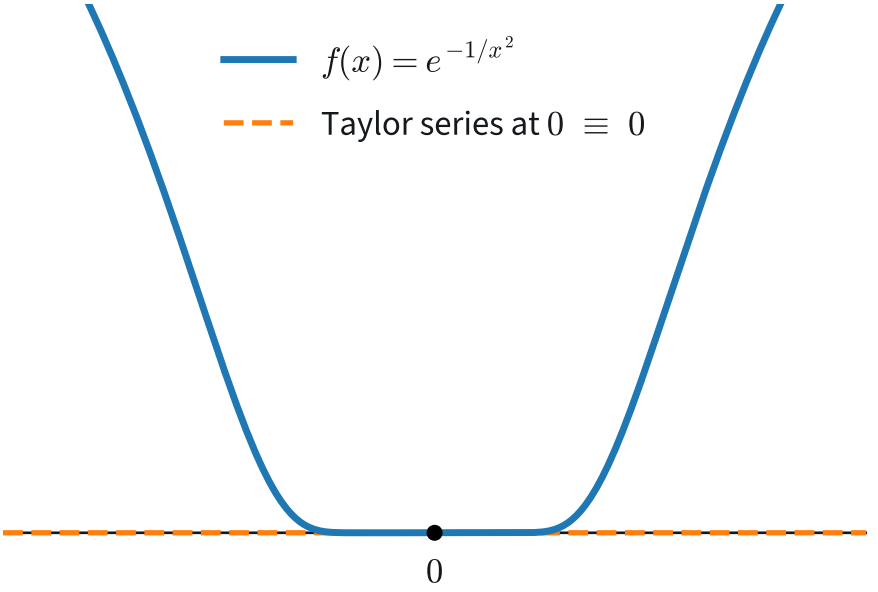


Figure 34.7: 滑らかな関数 $f(x) = e^{-1/x^2}$ は原点ですべての導関数がゼロに等しいため、そこでのテイラー級数は恒等的にゼロとなる。級数はいたるところで収束するが、f とは $x = 0$ でしか一致しない。すなわち、滑らかであることは解析的であることを意味しない。

$$f(x - \eta\, f'(x)) \;\le\; f(x) - \eta\Bigl(1 - \frac{L\eta}{2}\Bigr) f'(x)^2. \tag{34.6}$$

特に、f は $0 < \eta < 2/L$ かつ $f'(x) \neq 0$ である限り真に減少し、保証される減少量は $\eta = 1/L$ で最大となって $f'(x)^2/(2L)$ に等しくなる。

証明：任意のステップ u に対し、$n = 1$ のラグランジュの剰余項から $f(x+u) = f(x) + f'(x)u + \tfrac{1}{2}f''(\xi)u^2 \le f(x) + f'(x)u + \tfrac{L}{2}u^2$ が得られる。$u = -\eta f'(x)$ と置けば、右辺は $f(x) - \eta(1 - L\eta/2)f'(x)^2$ となる。因子 $1 - L\eta/2$ は $\eta < 2/L$ である限り正であり、二次関数 $\eta(1 - L\eta/2)$ は $\eta = 1/L$ で最大値をとる。$\blacksquare$

幾何学的には、この補題は未知の局所曲率 $f''(\xi)$ をその最悪の上界 L に置き換えている。得られる二次関数 $f(x) + f'(x)u + \tfrac{L}{2}u^2$ は、f に点 x で接し区間 I 全体でその上側に留まる上界の放物線として働く。ステップ $\eta = 1/L$ は、この天井の放物線の厳密な最小点（minimizer）へと一足飛びに移動するものである（Figure 34.8）。この上界は滑らかな最適化の主力となる道具である (Nesterov 2018)。実はこの結果は二階導関数さえ必要とせず、f' が L-**Lipschitz** である限り、すなわち任意の 2 点における傾きの差が高々その距離の L 倍（$|f'(u) - f'(v)| \le L|u - v|$）である限り成り立つ。多変数最適化（35.1 節）で用いられるのはこちらの形式であり、演習問題 1 でその 1 変数版を証明する。

よく知られた 2 つの関数でこの補題を確かめてみよう。x^2 では $L = 2$ であるから、$2/L = 1$ となって閉じた式から求めた厳密な安定性のしきい値が復元される。ロジスティック損失では、$\ell''(w) = \sigma(-w)\bigl(1 - \sigma(-w)\bigr) \le \tfrac{1}{4}$（等号成立は $w = 0$）であるから $L = \tfrac{1}{4}$ となり、任意のステップサイズ $\eta < 8$ に対して、どの初期重みからでも ℓ の減少が保証される。$w = 0$ から $\eta = 1$ で出発すると $w_1 = 0.5$ へ移動し、上界は損失が高々 $0.693147 - (1 - \tfrac{1}{8}) \cdot \tfrac{1}{4} = 0.4744$ になる

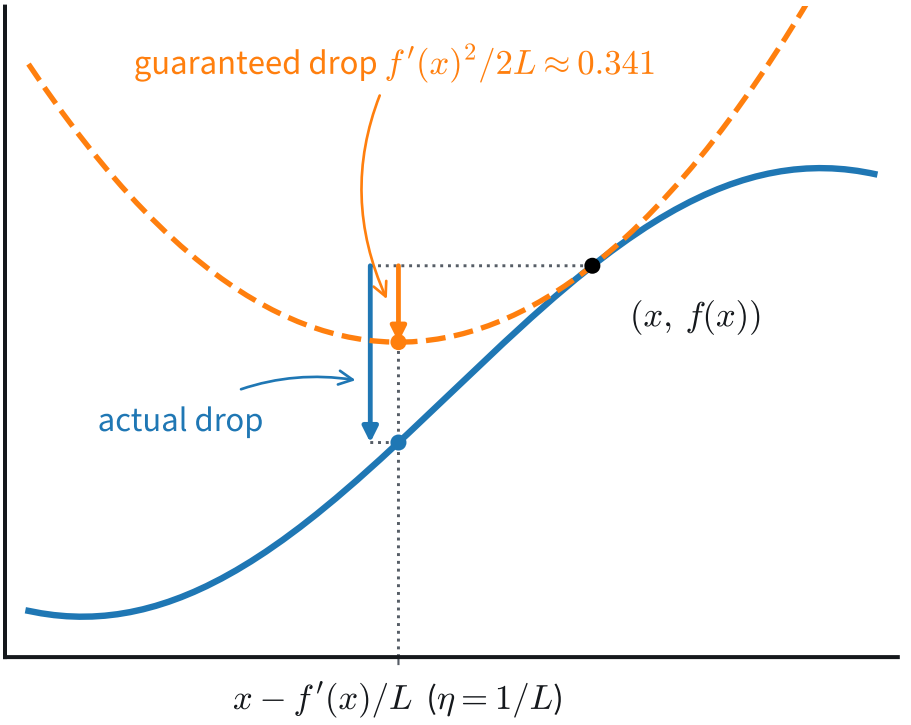


Figure 34.8: $|f''| \le L$ のとき、$f(x) + f'(x)u + \frac{L}{2}u^2$ はステップ u に関する二次の上界となり、基点で f と一致する。$\eta = 1/L$ とした勾配ステップはこの上界を最小化し、少なくとも $\frac{1}{2L}f'(x)^2$ の減少を保証する。ここでは $f(x) = \sin(2x)$（$L = 4$）であり、実際の減少量は最悪ケースの保証を上回っている。

ことを保証するが、実際の値は $\ell(0.5) = 0.4741$ である。この短いステップの間では曲率が最大値 L の近くに留まるため、上界はほぼタイトになっている。より大きなステップ $\eta = 4 = 1/L$ では、上界が約束するのは 0.1931 であるのに対し、実際の損失は $\ell(2) = 0.1269$ まで一気に下がる。原点から離れると曲率が平坦になるため、最悪ケースの天井が予測する以上に曲線が下っていくからである。

34.1.5.5 ニュートン法

降下補題は、大域的な最悪ケースの曲率 L を用いて二次モデルを構築していた。その代わりに現在の反復点における実際の局所曲率 $f''(x_t)$ を用いれば、任意の二次関数を一足飛びに最小化するステップが得られる。x_t における局所モデルは二次のテイラー多項式 P_2 である。

$$q(x) = f(x_t) + f'(x_t)\,(x - x_t) + \tfrac{1}{2} f''(x_t)\,(x - x_t)^2.$$

$f''(x_t) > 0$ である限り、この放物線は上向きに開き、唯一の最小点を持つ。$q'(x) = f'(x_t) + f''(x_t)(x - x_t) = 0$ と置いて x について解くと、**ニュートン法**の更新式が得られる。

$$x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)}. \tag{34.7}$$

これは適応的な局所ステップサイズ $1/f''(x_t)$ を持つ勾配ステップであり、上向きに開いた任意の放物線上では 1 ステップで厳密な最小点へ到達する。Figure 34.9 は、$f(x) = x^4/4 - x$（最小点は $x^\star = 1$）における 1 ステップを示している。$x_t = 2$ では $f(2) = 2$、$f'(2) = 7$、$f''(2) = 12$ であるから、放物線の最小点は $x_{t+1} = 2 - 7/12 = 1.4167$ に位置する。

$x^4/4 - x$ 上で $x_0 = 2$ からニュートン法がどれほど速く収束するかを見てみよう。あわせて 2 つ目の関数 $x \arctan x - \tfrac{1}{2}\log(1 + x^2)$ についても、$x_0 = 1.4$ と $x_0 = 1.3$ から出発したときに何が起きるかを試す。この 2 つ目の例については、数値を確認した直後に解析しよう。

```python
# Newton on f(x) = x^4/4 - x: solve f'(x) = x^3 - 1 = 0, root x* = 1
x = 2.0
for t in range(6):
    print(f't = {t}: x = {x:.12f}, error = {abs(x - 1):.1e}')
    x = x - (x**3 - 1) / (3 * x**2)  # x - f'(x) / f''(x)

# Newton on the convex f(x) = x arctan(x) - log(1 + x^2)/2,
# with f' = arctan(x) and f'' = 1/(1 + x^2)
for x0 in [1.4, 1.3]:
    x, xs = x0, []
    for t in range(6):
        x = x - math.atan(x) * (1 + x**2)
        xs.append(x)
    print(f'from x0 = {x0}:', ', '.join(f'{v:.2f}' for v in xs))
```

```
t = 0: x = 2.000000000000, error = 1.0e+00
t = 1: x = 1.416666666667, error = 4.2e-01
t = 2: x = 1.110534409842, error = 1.1e-01
t = 3: x = 1.010636768405, error = 1.1e-02
t = 4: x = 1.000111557304, error = 1.1e-04
t = 5: x = 1.000000012443, error = 1.2e-08
```

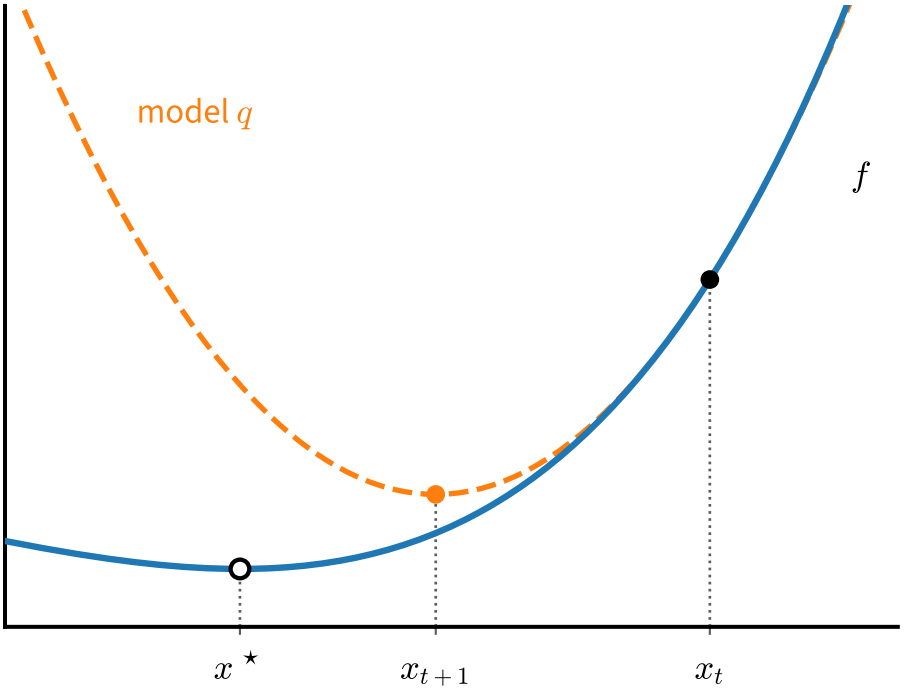


Figure 34.9: $f(x) = x^4/4 - x$ における $x_t = 2$ からの 1 回のニュートンステップ。放物線 q は f の x_t における値、傾き、曲率に一致し、その最小点が次の反復点 $x_{t+1} = x_t - f'(x_t)/f''(x_t) = 2 - 7/12$ となって最小点 $x^\star = 1$ に近づく。

```
from x0 = 1.4: -1.41, 1.45, -1.55, 1.85, -2.89, 8.71
from x0 = 1.3: -1.16, 0.86, -0.37, 0.03, -0.00, 0.00
```

$x^4/4 - x$ 上では、誤差は 1.0、0.42、0.11、1.1×10^{-2}、1.1×10^{-4}、1.2×10^{-8} と急減する。最小点に近づくと、各ステップは直前の誤差を二乗し、有効桁数を約 2 倍に増やしていくのである。対照的に、x^2 上の勾配降下法ではステップごとに定数因子 $|1 - 2\eta|$ 倍ずつしか誤差が縮まらず、反復ごとに一定の桁数しか稼げなかった。

なぜ誤差が二乗されるのかは、ラグランジュの剰余項によって説明できる。$x^\star$ を停留点とし、$e_t = x_t - x^\star$ をステップ t における誤差とする。Equation 34.5 を $n = 1$ として f' に基点 x_t の周りで適用し、$x^\star$ で評価すると、$0 = f'(x_t) - f''(x_t)\, e_t + \frac{1}{2} f'''(\xi_t)\, e_t^2$ が得られる。両辺を $f''(x_t) \neq 0$ で割り、ニュートン法の更新式 Equation 34.7 を代入すると、次式が残る。

$$e_{t+1} = \frac{f'''(\xi_t)}{2 f''(x_t)}\, e_t^2 \tag{34.8}$$

ここで ξ_t は x_t と $x^\star$ の間にある何らかの点である（ただし f は 3 回微分可能とする）。$x^4/4 - x$ では、前の係数は $f'''(1)/(2f''(1)) = 6/(2 \cdot 3) = 1$ に収束し、出力された誤差からも実際に $e_{t+1}/e_t^2 \approx 0.42$、$0.64$、$0.87$、$0.99$ が得られる。正の曲率を持つ最小点に十分近い任意の初期点からの二次収束（quadratic convergence）を与える一般の多変数定理は、35.2 節に登場する。

この「十分近い」という但し書きは、たとえ狭義凸関数であっても不可欠である。$f(x) = x \arctan x - \frac{1}{2} \log(1 + x^2)$ を考えよう。この関数は $f'(x) = \arctan x$ かつ $f''(x) = 1/(1 + x^2) > 0$ を満たすため狭義凸であり、点 0 に唯一の大域的最小値（global minimum）を持つ。ところが $x_0 = 1.4$ から出発すると、上の 2 つ目のループは $-1.41, 1.45, -1.55, 1.85, -2.89, 8.71$ と出力し、ニュートン法は激しく発散する。わずかに近い $x_0 = 1.3$ から出発すると滑らかに収束し（-1.16、0.86、-0.37、0.03、...）、収束と発散の境目は $x_0 \approx 1.39$ の付近にある。

何が問題だったのかは局所的な放物線が説明してくれる。原点から離れると曲率 $1/(1 + x^2)$ がごく小さくなるため、近似放物線 q はほぼ平坦になり、その最小点を $x^\star$ のはるか向こう側へ置いてしまうのである。標準的な対処法は、実際に f が減少することを保証するよう選んだ係数を掛けてステップを縮小する手法であり、これを**減衰**（damping）と呼ぶ。減衰付きニュートンステップ（Newton step）の構築と、高次元でそれを計算するコストについては 35.2.1 節で検討する。

34.1.6 微分不可能な点

あらゆる ReLU ネットワークは角だらけである。$\mathrm{ReLU}(x) = \max(0, x)$ は点 0 で導関数を持たない。それでも誤差逆伝播法はそこで数値を返す。いったいどの数値を選び、その数値はどのような意味で妥当な傾きなのだろうか。

34.1.6.1 片側導関数

差分商 Equation 34.1 を見れば、問題は直ちに明らかになる。$f(x) = |x|$ の $x = 0$ においては、次のようになる。

$$\frac{|0 + \epsilon| - |0|}{\epsilon} = \frac{|\epsilon|}{\epsilon} = \begin{cases} +1 & \epsilon > 0, \\ -1 & \epsilon < 0. \end{cases}$$

この商は 0 の両側でそれぞれ定数となるが、その 2 つの定数は一致しない。右側から近づくとグラフは傾き $+1$ で上り、左側から近づくと傾き -1 で下る。これらの極限が**片側導関数**（one-sided

derivatives）$f'_+(0) = +1$ および $f'_-(0) = -1$ であり、通常の導関数が存在するのはこの 2 つが一致するときに限られる。ReLU の点 0 でも同じ計算を行うと、平坦な左側では $f'_-(0) = 0$、右側では $f'_+(0) = 1$ となる。これらの角には接線の傾きが存在しないにもかかわらず、自動微分エンジンは何らかの数値を返さなければならない。実際に自動微分が何を返すか確認してみよう。

```
# What autograd returns at the corner, where no derivative exists
for name, fn in [('d|x|/dx', torch.abs), ("ReLU'", torch.relu)]:
    x = torch.tensor(0.0, requires_grad=True)
    fn(x).backward()
    print(f"{name} at 0: autograd returns {x.grad.item():.1f}")
```

```
d|x|/dx at 0: autograd returns 0.0
ReLU' at 0: autograd returns 0.0
```

$\mathrm{ReLU}'(0)$ に対しては、自動微分は左側の傾きである 0 を返す。$\frac{d}{dx}|x|$ の点 0 における値としては、規約によって定められた固定値を返す。

ここでは 0 を返す。これは規約 $\mathrm{sign}(0) = 0$ に従ったものである。

34.1.6.2 劣勾配と最適性

角にはただ 1 本の接線は存在しないが、その角でグラフに接し、かつ決してグラフの上側へ出ない直線の族全体が存在する。すべての弦がグラフ上またはその上側に位置する**凸**関数（convex function、35.3 節）においては、これらの支持直線（supporting line）が接線の役割を担う。実数 g が凸関数 f の点 x における**劣勾配**（subgradient）であるとは、次式が成り立つことをいう。

$$f(y) \geq f(x) + g\,(y - x) \quad \text{for all } y, \tag{34.9}$$

これは、$(x, f(x))$ を通り傾き g を持つ直線が、いたるところで f のグラフ上またはその下側にあることを意味する。点 x におけるすべての劣勾配からなる集合を**劣微分**（subdifferential）といい、$\partial f(x)$ と書く (Rockafellar 1970)。f が微分可能な任意の点では、$\partial f(x)$ はただ 1 つの数 $f'(x)$ のみを含む。一方、角においては両側の片側導関数を結ぶ閉区間へと広がり、Figure 34.10 に示す扇状の支持直線群に対応する。

$$\partial |x|(0) = [-1, 1], \qquad \partial\, \mathrm{ReLU}(0) = [0, 1].$$

先ほど自動微分が返した値はすべて、対応する区間のなかに収まっている。

通し例の関数を見れば、この区間がどこから現れるのかが直観的にわかる。鋭くしたソフトプラス $\mathrm{softplus}_\beta(x) = \frac{1}{\beta} s(\beta x)$ は $\mathrm{ReLU}(x)$ へと（$\beta \to \infty$ のとき）収束し、その導関数は $\sigma(\beta x)$ である（Figure 34.11）。$x = \pm 0.1$ における傾きは、0.40 と 0.60（$\beta = 4$ のとき）から、0.17 と 0.83（$\beta = 16$ のとき）へと急峻になる。劣微分 $[0, 1]$ は、$\beta \to \infty$ と $x \to 0$ を同時にとったときにこれらの滑らかな近似の傾きが近づいていく集合にほかならない。

劣勾配からは明快な最適性条件も得られる。$g = 0$ を Equation 34.9 に代入すると $f(y) \geq f(x)$ がすべての y に対して成り立つため、点 x が凸関数 f の大域的最小点となるための必要十分条件は次式で与えられる。

$$0 \in \partial f(x). \tag{34.10}$$

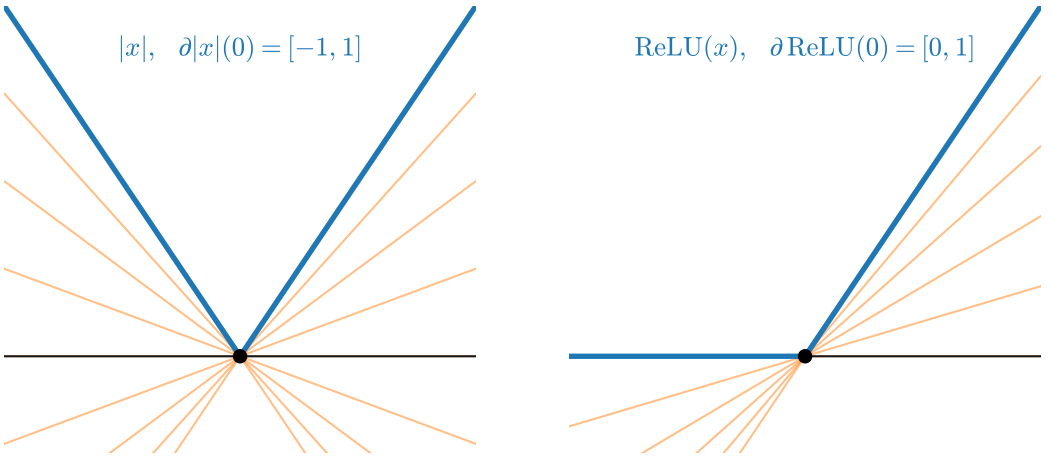


Figure 34.10: 2 つの角とその劣微分。$|x|$（左）と ReLU（右）の原点を通る扇状の支持直線群であり、その傾きはそれぞれ $\partial|x|(0)$ と $\partial\,\mathrm{ReLU}(0)$ にわたって動く。角以外の点では唯一の接線が存在し、劣微分は通常の導関数へと縮退する。

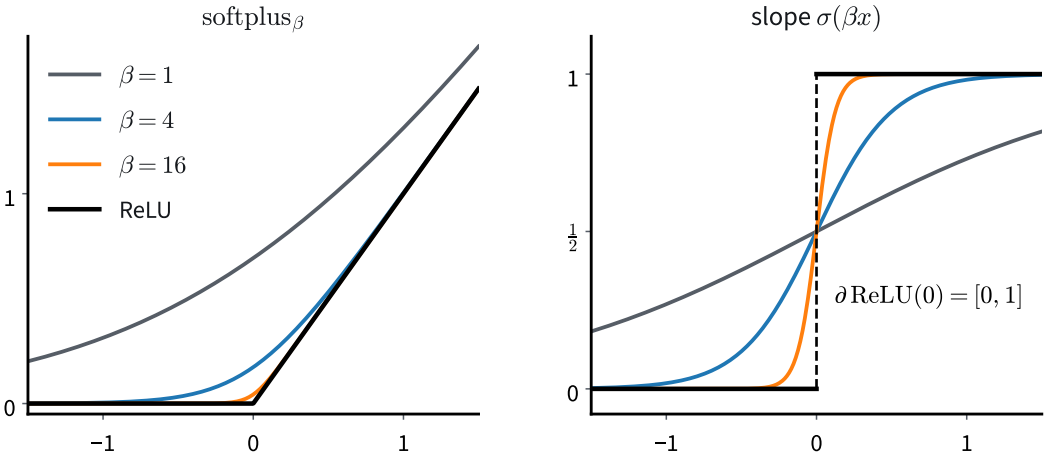


Figure 34.11: 角の平滑化。左：$\mathrm{softplus}_\beta(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$ は ReLU へと、β が大きくなるにつれて近づく。右：その傾き $\sigma(\beta x)$ は 0 から 1 への階段関数へと急峻になり、$x = 0$ の近傍では滑らかな近傍点での傾きが区間 $[0, 1]$、すなわち角における ReLU の劣微分を埋め尽くす。

この同値性の両方向は、その 1 行の式から直ちに従う。$|x|$ の場合、その点に導関数が存在しなくても、$0 \in [-1, 1]$ であることから原点が唯一の大域的最小値を与える点だと保証される。対照的に、凸性（convexity）がなければ $f'(x) = 0$ は必要条件にすぎない（フェルマーの補題）。実際、x^3 は $f'(0) = 0$ を満たすが最小値を一切持たない。

劣勾配は 35.3 節で多変数へと拡張するが、ここで 1 つ注意を述べておこう。真の勾配ステップとは異なり、劣勾配ステップは関数値を減少させる保証がない。$|x|$ の最小点そのものにおいて、実数 $g = \frac{1}{2}$ は妥当な劣勾配であるが、$x = -\eta/2$ へ進むと実際には $|x|$ が増加してしまう。そのため劣勾配法（subgradient method）は、毎ステップの前進を保証するのではなく、ステップサイズを徐々に縮小させることで収束を達成する (Nesterov 2018)。

34.1.6.3 ネットワーク内の折れ点

ニューラルネットワーク内の各折れ点（kink）は、$|x|$、ReLU、ヒンジ損失（hinge loss）といった単純な凸の基本要素に由来する。折れ点において、自動微分はその要素の劣微分から固定された元を 1 つ選び（たとえば $\mathrm{ReLU}'(0) = 0$）、連鎖律を用いて周囲の合成関数へと伝播させる。残念ながら、個々の部品の妥当な劣勾配を連鎖律でつなぎ合わせると、合成関数自体の劣勾配ではまったくない数値が生じることがある。最も単純な反例は、恒等関数を回りくどく書き表した次式である。

$$\varphi(x) = \mathrm{ReLU}(x) - \mathrm{ReLU}(-x) = x \quad \text{for all } x.$$

いたるところで $\varphi(x) = x$ であるから、点 0 における真の傾きは 1 しかない。ところが連鎖律を機械的に適用すると $\varphi'(x) = \mathrm{ReLU}'(x) + \mathrm{ReLU}'(-x)$ となり、これはほぼいたるところで 1 に等しいものの、$x = 0$ だけは例外であり、規約 $\mathrm{ReLU}'(0) = 0$ によって $0 + 0 = 0$ となってしまう。自動微分が実際にこのように振る舞うことを確かめてみよう。

```
# phi(x) = relu(x) - relu(-x) is the identity, so the true slope at 0 is 1
x = torch.tensor(0.0, requires_grad=True)
phi = torch.relu(x) - torch.relu(-x)
phi.backward()
print(f"autograd: phi'(0) = {x.grad.item():.1f}  (true slope: 1.0)")
```

```
autograd: phi'(0) = 0.0  (true slope: 1.0)
```

予測通り、自動微分は恒等関数に対して傾き 0 を返す。

このような誤差が引き起こされるには、活性化前の値が厳密に 0 に着地する必要がある。厳密な実数演算では、データと初期重みが連続分布から抽出されるとき、任意の固定されたステップでこれが起きる確率はゼロである。しかし有限精度の浮動小数点（floating-point）演算では、厳密なゼロが実際に生じる。Bertoin et al. (2021) は、小さな全結合ネットワーク上でこれがどの程度の頻度で起きるかを測定した。ここでは、出版版の実験におけるバグを修正して結論の一部を覆した著者らの 2023 年の正誤表（論文の改訂 arXiv 版）の数値を引用する。$\mathrm{ReLU}'(0) = 0$ と $\mathrm{ReLU}'(0) = 1$ による誤差逆伝播法は、少なくとも 1 つのミニバッチで異なる勾配を生成した割合が、16 ビット精度ではすべての初期化（initialization）、32 ビット精度では全初期化の半分弱、64 ビット精度では皆無であった。

それらの不一致が訓練に影響を与えるかどうかは別の問題であり、その答えは安心できるものである。実用上はまったく影響しない。32 ビット演算では、影響を受けるのは 100 万個のミニバッチのうちわずか数個にすぎず、そのミニバッチにおいてさえ 2 つの勾配の相対差は約 10^{-3} にとどまる。修正後の実験では、$\mathrm{ReLU}'(0)$ を $[0, 1]$ 内のどの値に選んでも実質的に同一の訓練損失（training loss）とテスト正解率が得られている。訓練が劣化するのは $\mathrm{ReLU}'(0) = 10^4$ のような

極端な選択をした場合だけであり、バッチ正規化（batch normalization）や Adam を用いればそれさえも抑え込まれる。したがって、この規約は実行間のわずかな数値的差異を生む実際の要因ではあるものの、無害なものである。演習問題 5 では、この現象を小規模な設定で調べる。

本節で扱ったのはすべて単一のスカラー重みであった。ニューラルネットワークでは何百万もの重みが連動して動く。どの同時方向が損失を下げるのか、そして高次元における曲率とは何を意味するのかが、続く 34.2 節の主題である。

34.1.7 まとめ

導関数 $f'(x)$ は、線形近似 $f(x+\epsilon) = f(x) + \epsilon f'(x) + o(\epsilon)$ の傾きである。この近似から証明される和・積・連鎖律によって、初等関数の短い表から組み立てられる任意の合成関数（ソフトプラスにおける $s' = \sigma$ など）を微分できる。単一の点における傾きから区間上の保証へと進むために用いるのが平均値の定理とその高次への拡張であるラグランジュの剰余項であり、次数 n のテイラー多項式の誤差を $|x - x_0|^{n+1}$ の定数倍で抑える（$f^{(n+1)}$ が有界である限り）。

次数 $n = 1$ で適用すると、ラグランジュの剰余項から降下補題が導かれる。すなわち、ステップ区間上で $|f''| \le L$ かつ $f'(x) \neq 0$ であれば、$0 < \eta < 2/L$ を満たすすべての勾配ステップは f を真に減少させる。大域的な上界 L を局所曲率 $f''(x_t)$ に置き換えればニュートンステップ $-f'/f''$ が得られ、$f'' > 0$ を満たす最小点の近傍では誤差を二乗するが、最小点から遠いとたとえ狭義凸関数上であっても発散し得る。最後に、凸関数の微分不可能な角においては劣微分 $\partial f(x)$ が導関数に取って代わり、$0 \in \partial f(x)$ が大域的最小点を特徴づける。自動微分は折れ点ごとに 1 つの元を選ぶが、浮動小数点演算においてそれらの選択の連鎖が真の劣勾配を与え損ねることがあっても、訓練への影響は無視できるほど小さい。

34.1.8 演習問題

1. **ロジスティック損失の滑らかさ。**降下補題 Equation 34.6 には曲率の上界 L が必要である。

   1. 平均値の定理 Equation 34.3 を用いて、ある区間上で $|f''| \le L$ ならばその区間上で f' が L-Lipschitz であることを示せ。逆に、定義 Equation 34.1 から、2 回微分可能な f が L-Lipschitz な f' を持つならば $|f''| \le L$ を満たすことを示せ。

   2. 1 つの重み w、入力 $a_i \in \mathbb{R}$、ラベル $y_i \in \{-1, 1\}$ を持つロジスティック回帰は、次の関数を最小化する。

   $$\ell(w) = \frac{1}{n} \sum_{i=1}^{n} \log\bigl(1 + e^{-y_i a_i w}\bigr). \tag{34.11}$$

   ℓ'' を σ を用いて表せ。最小の L（ℓ' が L-Lipschitz に $\mathbb{R}$ 全体でなるもの）、およびすべての初期点からの減少を Equation 34.6 が保証するステップサイズを求めよ。

   3. f が微分可能であることのみを仮定し、ステップの区間上で f' が L-Lipschitz である場合に降下補題を証明せよ。$u = -\eta f'(x)$ に対し $\psi(t) = f(x+tu) - f(x) - tf'(x)u - \frac{L}{2}t^2u^2$ と置く。$\psi(0) = 0$ および $\psi'(t) = u\,(f'(x+tu) - f'(x)) - Ltu^2 \le 0$ が $t \in [0,1]$ で成り立つことを示し、平均値の定理から $\psi(1) \le 0$ を導け。これは多変数関数の最適化で用いられる補題の形式である。

2. **エントロピーの極値。**表が確率 p で出るコインの二値エントロピーは次式で与えられる。

$$H(p) = -p \log p - (1-p)\log(1-p), \qquad 0 < p < 1. \tag{34.12}$$

1. $f(x) = x \log x$ の $x \geq 0$ における最小値を求めよ（ただし $f(0) = 0$ は極限値とする）。最小点が境界 $x = 0$ に位置しないのはなぜか。
2. $H'(p) = -\log \frac{p}{1-p}$、すなわち p の負のロジットとなることを示せ。H の停留点を求め、二階微分判定法によって分類せよ。
3. $H'(p)$ の $p \to 0$ における極限を求めよ。次にコインをそのロジット z でパラメータ化して $p = \sigma(z)$ と置き、$\frac{d}{dz} H(\sigma(z))$ を計算せよ。2 つのパラメータ化のうち、定義域全体で H の傾きが Lipschitz となり、Equation 34.6 によってすべての初期点に対して $-H$ 上の勾配降下法に共通のステップサイズが保証されるのはどちらか。

3. **退化した最小点におけるニュートン法。** $p > 1$ に対し、凸関数 $f_p(x) = |x|^p / p$ は $x = 0$ に唯一の最小点を持つ。

1. f_p が点 0 を含むいたるところで微分可能であって $f_p'(x) = \text{sign}(x)\,|x|^{p-1}$ を満たすことを示し、$f_p''(x)$ を $x \neq 0$ に対して計算せよ。どの p に対して $f_p''(0)$ は存在し、その値は何か。
2. ニュートン法 Equation 34.7 がすべての $x_t \neq 0$ を $x_{t+1} = \frac{p-2}{p-1}\, x_t$ に写すことを示せ。どの p に対してこの反復は収束するか。また、$p = 4$ ではどれほどの速さで収束するか。
3. 34.1.5.5 節における局所収束の命題の仮定のうち、$p = 4/3$ で成り立たないものはどれか。また、$p = 4$ で成り立たないものはどれか。
4. $p = 4/3$ において、減衰付き更新式 $x_{t+1} = x_t - \alpha f_p'(x_t) / f_p''(x_t)$ はニュートンステップを係数 $\alpha > 0$ だけ短縮する。どの α に対してすべての x_0 から収束し、その収束率はどのようになるか。

MIT 18.01SC、Session 33 演習問題「Cube Root of x」より改題。

4. [code] **折れ点にある最小点。** 1 つの重み x、重み減衰 $\lambda > 0$、およびラベルと入力の積が 1 に等しい 1 つの訓練例を持つサポートベクトルマシンは、次の関数を最小化する。

$$F(x) = \max(0, 1 - x) + \frac{\lambda}{2} x^2. \tag{34.13}$$

1. ヒンジ損失 $\max(0, 1-x)$ のすべての x における劣微分を Equation 34.9 から計算せよ。
2. 微分可能な凸関数 q を凸関数 h に加えるとすべての劣勾配に q' が加わること、すなわち $\partial(q + h)(x) = q'(x) + \partial h(x)$ は既知としてよい。最適性条件 Equation 34.10 を用いて、関数 F の最小点をすべての $\lambda > 0$ に対して求めよ。最小点が折れ点に位置するのはどの λ のときか。
3. $\lambda = 1/2$ とし、劣勾配法 $x_{t+1} = x_t - \eta_t g_t$ を $x_0 = 0$ から 10^4 ステップ実行せよ。ただし g_t は、ヒンジ損失を $F'(x_t)$ の計算において $\text{ReLU}(1 - x)$ と書いたときに自動微分が返す値とする。定数ステップ $\eta_t = 0.1$ および $\eta_t = 0.01$ を、減衰ステップ $\eta_t = 0.5/\sqrt{t+1}$ と比較せよ。最小点までの最終的な距離はスケジュールにどう依存するか。
4. 折れ点 $\{x = 1\}$ は単一の点であり、確率密度からの抽出がそこに当たる確率はゼロである。(b) と (c) の結果は、訓練が折れ点に出会うことはまれであるという 34.1.6.3 節の議論に対して何を示唆するか。

5. [code] **連鎖させた劣勾配。** 自動微分は $\text{ReLU}'(0) = 0$ を返し、34.1.6.3 節で見たように、あらゆる式にわたってこの値を連鎖律で伝播させる。

1. f_1 と f_2 を凸関数とし、$g_1 \in \partial f_1(x)$、$g_2 \in \partial f_2(x)$、$c_1, c_2 \geq 0$ とする。Equation 34.9 から $c_1 g_1 + c_2 g_2 \in \partial(c_1 f_1 + c_2 f_2)(x)$ を示せ。この議論が $\text{ReLU}(x) - \text{ReLU}(-x)$ には適用できないのはなぜか。

2. すべての実数 c に対して次式が成り立つことを示せ。

$$k_c(x) = \mathrm{ReLU}(x) + \mathrm{ReLU}(-x) + c\,(x - \mathrm{ReLU}(x) + \mathrm{ReLU}(-x)) = |x|. \tag{34.14}$$

自動微分が $k_c'(0)$ に対して返す値を予測し、$c = 0$、1、5 についてその予測を確かめよ。どの c に対して、返された値は $|x|$ の点 0 における劣勾配となるか。

3. 自動微分から得られる導関数を用いた勾配降下法を $\varphi(x) = \mathrm{ReLU}(x) - \mathrm{ReLU}(-x)$ 上で 10 ステップ、$x_0 = 0.5$ から、$\eta = 0.125$ の場合と $\eta = 0.1$ の場合についてそれぞれ実行せよ。各実行では何が起きるか。2 つのステップサイズで振る舞いが異なるのはなぜか。また、この違いは浮動小数点演算における 34.1.6.3 節の確率ゼロの議論に対して何を示唆するか。

6. **四次関数上の降下法。**●$f(x) = x^4/4$ の最小点は $f''(0) = 0$ であるため平坦である。勾配降下法 Equation 34.4 を f に適用すると $x_{t+1} = x_t - \eta x_t^3$ となる。

1. 問題 1a を用いて、最小の L（f' が L-Lipschitz に $[-R, R]$ 上でなるもの）を求めよ。どの初期点 x_0 が、与えられた η に対して降下補題 Equation 34.6 の適用対象となるか。固定された η ではすべての初期点を網羅できないのはなぜか。
2. 反復点列は 0 に $\eta x_0^2 < 2$ のとき収束し、$\eta x_0^2 > 2$ のとき発散することを示せ。$\eta x_0^2 = 2$ では何が起きるか。このしきい値を (a) で対象となった初期点と比較せよ。
3. $1/x_{t+1}^2 = 1/x_t^2 + 2\eta + O(x_t^2)$ を示し、$x_t^2 \approx 1/(x_0^{-2} + 2\eta t)$ を導け。この予測を $\eta = 0.1$、$x_0 = 1$ について $t = 10$、100、1000 で確かめよ。この収束率は、x^2 上の勾配降下法や、$x^4/4$ 上のニュートン法（問題 3b）と比べるとどうか。

7. [code] **中心差分。** ●前進差分 $D_+ f(x) = \frac{f(x+\epsilon) - f(x)}{\epsilon}$ と中心差分 $D_0 f(x) = \frac{f(x+\epsilon) - f(x-\epsilon)}{2\epsilon}$ はいずれも $f'(x)$ を近似する。

1. ラグランジュの剰余項 Equation 34.5 を用いて、$D_+ f(x) - f'(x) = \frac{1}{2} f''(\xi)\,\epsilon$ および $D_0 f(x) - f'(x) = \frac{1}{6} f'''(\xi')\,\epsilon^2$ がある ξ、ξ'（x の近傍）に対して成り立つことを示せ。ただし f は 3 回連続微分可能とする（f''' の連続性により 2 つの剰余項の点が 1 つにまとめられる）。$f(x \pm \epsilon)$ を必要な次数まで展開して差をとれ。
2. $f(x) = x^2 + 1701(x-4)^3$ の $x = 4$ において、2 つの誤差がそれぞれ $\epsilon + 1701\epsilon^2$ と $1701\epsilon^2$ であることを示せ。$\epsilon = 0.1$ ではそれぞれ 17.11 と 17.01 となり、ほぼ等しい。中心差分は誤差の次数を引き上げるが、それが効力を発揮するのは ϵ が小さくなってからである。中心差分の誤差が前進差分の誤差の 100 分の 1 になるのはどの ϵ のときか。

浮動小数点演算では、丸め誤差によって ϵ に下限が生じる。34.3 節の勾配チェックの演習問題では、これと (a) の打ち切り誤差との釣り合いをとる。

8. [code] **$2/L$ を超える勾配降下法。**●●$f(x) = x^2$ 上では、ステップサイズ $\eta > 2/L$ はすべて発散する。しかし他の関数では、このしきい値を超えた勾配降下法が発散する代わりに振動へと落ち着くことがあり (Chen and Bruna 2023)、これは安定性の縁の 1 次元版にあたる (Cohen et al. 2021)。$x^\star$ を 4 回連続微分可能な関数 f の極小点（$f''(x^\star) > 0$）とし、$T(x) = x - \eta f'(x)$ を Equation 34.4 の 1 ステップとする。

1. Equation 34.3 を用いて、$x^\star$ に十分近い点から出発した反復点列は $\eta < 2/f''(x^\star)$ のときそれに収束することを示せ。また、$\eta > 2/f''(x^\star)$ のとき、$x^\star$ のある近傍内の $x^\star$ 以外のすべての反復点について、次の反復点が $x^\star$ からより遠ざかることを示せ。

2. 周期 2 の軌道とは、$a \neq b$ かつ $T(a) = b$、$T(b) = a$ を満たす対である。任意の周期 2 の軌道に対し、ある点 ξ が a と b の真の間に存在して $f''(\xi) = 2/\eta$ を満たすことを示せ。

3. $\eta = (1 + \delta)\, 2/f''(x^\star)$（ただし $\delta > 0$ は微小）と置く。$T(T(x))$ を $x^\star$ の周りで 3 次まで展開し、周期 2 の軌道が $x^\star$ の近傍に存在して近傍の反復点を引き寄せるための条件が次式であることを示せ。

$$3 f'''(x^\star)^2 - f''(x^\star)\, f''''(x^\star) > 0. \tag{34.15}$$

   軌道の半幅 $|a - b|/2$ を δ の主要項まで導出せよ。

4. 二重井戸関数 $f(x) = \tfrac{1}{4}(x^2 - 1)^2$ 上で $x_0 = 0.5$ から勾配降下法を 5000 ステップ、各ステップサイズ $\eta \in [0.9, 1.5]$ について実行し、最後の 100 個の反復点を η に対してプロットせよ。$\eta = 1$ 近傍における軌道の半幅を (c) と比較せよ。さらに、安定な周期 2 の軌道が存在する η の範囲を、$1 < \eta < \sqrt{4.5} - 1$（この範囲では Chen and Bruna (2023) がほとんどすべての $0 < x_0 < 1$ から軌道への収束を証明している）と比較せよ。

5. (c) を用いて、$\eta = 2.02$ の勾配降下法が $\log\cosh x$ および $x^2/2 + x^4/4$（いずれも $x^\star = 0$ かつ $f''(x^\star) = 1$ を持つ）上でどのように振る舞うかを予測せよ。両方について $x_0 = 0.01$ から 500 ステップ実行し、予測を確かめよ。

34.2 多変数の微分積分

前節では、他のすべての重みを固定したまま、単一の重みの関数として損失を調べた。しかし実際の訓練はそのようには進まない。勾配降下法の各ステップはすべての重みを一度に少しずつ動かすため、同時移動に対して損失がどう応答するか、そしてどの同時移動が損失を最も速く減少させるかを知る必要がある。重みごとに 1 つずつ、100 万個の別々の傾きを持っているだけでは、そのどちらの問いにも答えられない。

ところが、1 つの穏やかな仮定のもとでは、それらの傾きが両方の問いに答えてくれる。損失が微分可能であるとき、**勾配**（gradient）と呼ばれる傾きのベクトルは、あらゆる微小な同時移動の影響を 1 次の範囲で一括して予測し、最も急激に増加する方向を指し示す。さらに踏み込んで極小点（local minimum）と鞍点（saddle point）を見分けるには、曲面がどのように曲がっているかを測る二階導関数の対称な表、すなわち**ヘッセ行列**（Hessian）が必要になる。本節では、これら 2 つの対象が多変数関数について何を教えてくれるかを展開する。大規模なネットワークに対してプログラムが実際にどうやって勾配を計算するかは別の問題であり、続く 34.3 節で取り上げる。

1.4 節ではすでに微分公式を述べ、勾配を実際に活用した。本節では、34.1 節が 1 変数について行ったのと同様に、その数学的基礎を与える。すなわち、線形近似がいつ成り立つか、そして二階導関数がどのように臨界点を分類するかを証明する。本節全体の柱となるのは、多変数の線形モデルと二階微分判定法（second-derivative test）という 2 つの結果である。接平面に関する短い項と末尾のラグランジュ乗数に関する内容は、初読時には飛ばしても差し支えない。本節を通して、各概念を共通の例題で検証していく。2 変数の log-sum-exp 関数 $f(x, y) = \log(e^x + e^y)$ を点 $(0, \log 2)$ で評価したものである。その勾配は、おなじみのソフトマックスになることがわかるだろう。

34.2.1 偏微分と勾配

34.2.1.1 偏微分

w_1 以外のすべての重みを固定すると、損失は 1 変数の関数になる。その通常の導関数が、1.4 節で導入した**偏微分**（partial derivative）$\partial \ell / \partial w_1$ である。幾何学的には、これは ℓ のグラフにおいて w_1 のみを動かして得られる 1 次元スライスの傾きを表す。1 変数の微分積分と同様に、この傾きは微小な変化 ϵ_1 を w_1 単独に加えたときの影響を次のように予測する。

$$\ell(w_1 + \epsilon_1, w_2, \dots, w_N) \approx \ell(w_1, w_2, \dots, w_N) + \epsilon_1 \frac{\partial \ell}{\partial w_1}(w_1, w_2, \dots, w_N). \tag{34.16}$$

例題である $f(x, y) = \log(e^x + e^y)$ の偏微分を計算してみよう。y を固定して x に関して微分すると $\partial f / \partial x = e^x / (e^x + e^y)$ が得られ、対称性から $\partial f / \partial y = e^y / (e^x + e^y)$ となる。点 $(0, \log 2)$ では $e^x = 1$ および $e^y = 2$ であるから、$f(0, \log 2) = \log 3 \approx 1.0986$、$\partial f / \partial x = \frac{1}{3}$、$\partial f / \partial y = \frac{2}{3}$ となる。言い換えれば、y だけをわずかに動かすと、f は x だけを動かしたときの 2 倍の速さで増加する。

34.2.1.2 微分可能性と勾配

偏微分が教えてくれるのは、各座標軸に沿った 1 方向ずつの傾きである。すべての座標が同時に動くとき、たとえば $(0, \log 2)$ から $\epsilon = (0.01, -0.03)$ だけステップを進めるときに何が起きるかは、偏微分からは自動的にはわからない。座標ごとの変化を単純に足し合わせて $\frac{1}{3} \cdot 0.01 + \frac{2}{3} \cdot (-0.03)$ になると推測したくなるかもしれない。しかしその推測は、実はその点の近傍における f の振る舞いについての強い主張であり、成り立つにはある仮定が必要である。

その仮定とは、$\mathbf{w}$ の近傍において、関数 ℓ がステップサイズより速く消える誤差を除いて線形に見えるというものである。形式的には、ℓ が点 $\mathbf{w}$ で**微分可能**（differentiable）であるとは、次の式を満たす線形写像 Λ が存在することをいう。

$$\ell(\mathbf{w} + \epsilon) = \ell(\mathbf{w}) + \Lambda(\epsilon) + o(\|\epsilon\|), \tag{34.17}$$

ここで $o(\|\epsilon\|)$ は、$\|\epsilon\|$ で割ったときに $\epsilon \to \mathbf{0}$ の極限でゼロに収束する任意の剰余項を表す。

そのような線形写像 Λ が存在するとき、偏微分によってそれは完全に決定される。$\epsilon = h\mathbf{e}_i$ を Equation 34.17 に代入し、両辺を h で割って $h \to 0$ とすると、$\Lambda(\mathbf{e}_i) = \partial \ell / \partial w_i$ が導かれる。Λ は線形であるため、任意の同時ステップ ϵ に対する作用は次のように定まる。

$$\Lambda(\epsilon) = \Lambda\Big(\sum_{i=1}^{N} \epsilon_i \mathbf{e}_i\Big) = \sum_{i=1}^{N} \epsilon_i \frac{\partial \ell}{\partial w_i} = \epsilon \cdot \nabla_{\mathbf{w}} \ell, \qquad \nabla_{\mathbf{w}} \ell = \left[\frac{\partial \ell}{\partial w_1}, \dots, \frac{\partial \ell}{\partial w_N} \right]^{\top}.$$

偏微分を並べたこの列ベクトル $\nabla_{\mathbf{w}} \ell$ を ℓ の**勾配**と呼ぶ。これを Equation 34.17 に代入すると、多変数の線形近似が得られる。

$$\ell(\mathbf{w} + \epsilon) = \ell(\mathbf{w}) + \epsilon \cdot \nabla_{\mathbf{w}} \ell(\mathbf{w}) + o(\|\epsilon\|). \tag{34.18}$$

これを 1 変数の近似式 $f(x + \epsilon) \approx f(x) + \epsilon f'(x)$ と比べてみよう。スカラーの傾きがベクトルに、通常の積がドット積に置き換わっただけで、式の形はまったく同じである。勾配とは、ステッ

プとのドット積が ℓ の 1 次変化を与える唯一のベクトルである。勾配の捉え方はいくつもあるが、最も心に留めておくべきなのはこの見方である。本書を通じて、勾配は列ベクトルとして扱う。

点 $(0, \log 2)$ における例題に戻ると、$\nabla f = [\frac{1}{3}, \frac{2}{3}]^\top$ であった。$\epsilon = (0.01, -0.03)$ を Equation 34.18 に代入すると、$f(0.01, \log 2 - 0.03) \approx \log 3 + \frac{1}{3} \cdot 0.01 - \frac{2}{3} \cdot 0.03 \approx 1.081946$ と予測される。なお、**あらゆる**点 (x, y) において、∇f の 2 つの成分は正であり、和が 1 になることに注目してほしい。すなわち log-sum-exp の勾配はソフトマックス $\mathbf{p} = \mathrm{softmax}(x, y)$ であり、その成分は $e^x/(e^x + e^y)$ と $e^y/(e^x + e^y)$ である。

ここで一度立ち止まり、なぜ微分可能性が単なる偏微分の存在以上のものを要求するのかを確認しておこう。$f(x, y) = 2xy/(x^2 + y^2)$（ただし $f(0, 0) = 0$）を考える。両方の座標軸上では f は恒等的にゼロであるため、原点における両偏微分はいずれも 0 となり、候補となり得る線形モデルは定数関数 0 しかない。ところが対角線 $x = y \neq 0$ に沿うと、原点にどれほど近づいても $f(x, x) = 1$ のままであるため、原点ではいかなる線形近似も成り立たない。この例では f は原点で連続ですらないが、演習問題 1 では、連続関数であってすべての偏微分が存在していてもなお微分可能でない場合があることを示す。

幸いなことに、Equation 34.17 を直接確かめなければならない場面はめったにない。使いやすい十分条件として、$\mathbf{w}$ のある近傍ですべての偏微分が存在し、かつ点 $\mathbf{w}$ で連続であれば微分可能となる (Rudin 1976)。log-sum-exp の偏微分はいたるところで連続であるため、log-sum-exp はいたるところで微分可能である。

微分可能性の定義が約束するのは、近似誤差が $o(\|\epsilon\|)$ であること、つまり $\|\epsilon\|$ より速く縮むことだけである。実際にはどれほど速く縮むのだろうか。次のセルを実行する前に、ステップ $\epsilon = (0.01, -0.03)$ を 10 分の 1 に縮めたとき、線形モデルの誤差が何分の 1 になるか予想してみてほしい。以下の実験では、通常の NumPy を使ってこれを確かめる。

```
import numpy as np
import torch
```

```
def lse(xy):  # f(x, y) = log(e^x + e^y), computed stably
    return np.logaddexp(xy[0], xy[1])

def grad_lse(xy):  # its gradient, the softmax of (x, y)
    return np.exp(xy - lse(xy))

xy0 = np.array([0.0, np.log(2)])
print('gradient:', grad_lse(xy0))
for scale in [1, 0.1, 0.01]:
    eps = scale * np.array([0.01, -0.03])
    linear = lse(xy0) + eps @ grad_lse(xy0)
    true = lse(xy0 + eps)
    print(f'|eps| {np.linalg.norm(eps):.1e}: linear {linear:.8f}, '
          f'true {true:.8f}, gap {true - linear:.3e}')
```

```
gradient: [0.33333333 0.66666667]
|eps| 3.2e-02: linear 1.08194562, true 1.08212418, gap 1.786e-04
|eps| 3.2e-03: linear 1.09694562, true 1.09694740, gap 1.779e-06
|eps| 3.2e-04: linear 1.09844562, true 1.09844564, gap 1.778e-08
```

$\|\epsilon\| \approx 0.03$ では、線形モデルと真の値とのずれは 1.79×10^{-4}（小数第 4 位の差）にすぎず、ステップを 10 分の 1 に縮めるたびにその誤差は 100 分の 1 に縮小する。つまり誤差はステップに

関して 2 次であり、微分可能性単独で保証されるものよりはるかに精度が高い。まもなく見るように、ヘッセ行列を使えばこの誤差を正確に予測できる（34.2.3.2 節）。

34.2.1.3 方向微分

ここまでは微小ステップに対する変化を予測するために勾配を使ってきたが、勾配は座標軸方向だけでなく、任意の方向に沿った厳密な瞬間的傾きも与える。任意の単位ベクトル $\mathbf{u}$ をとり、$\epsilon = h\mathbf{u}$ を Equation 34.18 に代入して h で割り、$h \to 0$ とする。すると剰余項 $o(\|h\mathbf{u}\|)$ は消え、次式が残る。

$$\lim_{h \to 0} \frac{\ell(\mathbf{w} + h\mathbf{u}) - \ell(\mathbf{w})}{h} = \mathbf{u} \cdot \nabla_{\mathbf{w}} \ell(\mathbf{w}). \tag{34.19}$$

左辺の極限は ℓ の $\mathbf{u}$ に沿った**方向微分**（directional derivative）であり、右辺はそれが勾配の $\mathbf{u}$ への射影にほかならないことを示している。したがって、たった 1 つのベクトルがあらゆる方向の傾きを同時に符号化しているのである。

点 $(0, \log 2)$ における例題で 3 つの方向を調べると、その振る舞いの全貌が見えてくる。対角方向 $\mathbf{u} = (1,1)/\sqrt{2}$ に沿った傾きは $(\tfrac{1}{3} + \tfrac{2}{3})/\sqrt{2} = 1/\sqrt{2} \approx 0.707$ である。実はこの傾きは**あらゆる**点で同じ値をとる。というのも $f(x+t, y+t) = \log(e^t(e^x + e^y)) = f(x,y) + t$ が成り立ち、対角方向に距離 $\sqrt{2}\,t$ だけ進むと f は常に t だけ増加するからである。勾配と正反対の方向 $\mathbf{u} = -(1,2)/\sqrt{5}$ に向かうと、傾きは $-(\tfrac{1}{3} + \tfrac{4}{3})/\sqrt{5} = -\sqrt{5}/3 \approx -0.745$ となる。最後に、$\mathbf{u} = (2,-1)/\sqrt{5}$ に沿った傾きは $(\tfrac{2}{3} - \tfrac{2}{3})/\sqrt{5} = 0$ となり、この方向には 1 次の範囲で f はまったく変化しない。

34.2.2 勾配の幾何学

34.2.2.1 最急降下

勾配降下法（2.1 節）は、損失が最も速く減少する方向へパラメータを少しだけ動かす手法である。Equation 34.19 によれば、求めたいのは単位ベクトル $\mathbf{v}$ のうち、射影 $\mathbf{v} \cdot \nabla \ell$ が最小（負として絶対値が最大）になるものである。1.4 節では $\mathbf{v}$ と勾配のなす角の余弦からこの方向を導いたが、コーシー・シュワルツの不等式（Cauchy–Schwarz inequality）を使えばさらに簡潔に証明でき、しかもその方向が一意であることまで確認できる。

命題（最急上昇と最急降下） $\nabla \ell \neq \mathbf{0}$ **となる任意の点において、すべての単位ベクトル** $\mathbf{v}$ **の中で、方向微分** $\mathbf{v} \cdot \nabla \ell$ **は** $\mathbf{v} = \nabla \ell / \|\nabla \ell\|$ **のとき最大となり、** $\mathbf{v} = -\nabla \ell / \|\nabla \ell\|$ **のとき最小となる。したがって、** $+\nabla \ell$ **は最急上昇（steepest ascent）の方向であり、** $-\nabla \ell$ **は最急降下（steepest descent）の方向である。**

証明： コーシー・シュワルツの不等式（33.1 節）によれば、$|\mathbf{v} \cdot \mathbf{g}| \le \|\mathbf{v}\|\|\mathbf{g}\|$ が成り立ち、等号は $\mathbf{v}$ と $\mathbf{g}$ が共線であるとき、かつそのときに限り成り立つ。単位ベクトル $\mathbf{v}$ と $\mathbf{g} = \nabla \ell$ に対して、この不等式は次のようになる。

$$-\|\nabla \ell\| \;\le\; \mathbf{v} \cdot \nabla \ell \;\le\; \|\nabla \ell\|.$$

上下それぞれの限界は、$\nabla \ell$ と共線な単位ベクトルによって一意に達成される。すなわち、上界は $\mathbf{v} = \nabla \ell / \|\nabla \ell\|$ によって、下界は $\mathbf{v} = -\nabla \ell / \|\nabla \ell\|$ によって達成される。$\blacksquare$

点 $(0, \log 2)$ における例題では、最急降下方向は $-(1,2)/\sqrt{5}$ であり、その傾きは $-\|\nabla f\| = -\sqrt{5}/3 \approx -0.745$ となって、先ほど計算した 3 つの傾きの中で最も小さい負の値になっている。$-\nabla_{\mathbf{w}}\ell$ の方向にステップを進めると、おなじみの勾配降下法の更新式が得られる。

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\, \nabla_{\mathbf{w}} \ell(\mathbf{w}).$$

この更新式では勾配を正規化していないことに注目してほしい。ステップの長さは $\eta\|\nabla\ell\|$ であり、最小点の近くで勾配が小さくなるにつれて自然に短くなる。

見落とされがちだが、上の命題における「単位」とは**ユークリッド**長が 1 という意味であり、これは自然法則ではなくモデル化における 1 つの選択にすぎない。代わりにステップの各座標の絶対値を高々 1 に制約すると、最急降下ステップは座標ごとに $-\operatorname{sign}(\nabla\ell)$ を適用したものになる。点 $(0, \log 2)$ ではそのステップは $(-1, -1)$ であり、傾きは $-(\frac{1}{3} + \frac{2}{3}) = -1$ となる。更新量の各座標をそのスケールの移動推定値で割る Adam のような適応的オプティマイザ（第 9 章）は、ユークリッドな最急降下よりもむしろこの符号ステップに近い振る舞いをする。演習問題 4 では、他のノルムのもとでの最急降下を調べる。

34.2.2.2 勾配と等位集合

ステップを一切持ち出さなくても、等高線図から直接勾配を読み取ることもできる。ℓ の**等位集合**（level set）とは、ℓ が一定の値 c をとるすべての点の集まりである。等位集合に沿って動いても ℓ は一定のままであるから、任意の接線方向に沿った方向微分は消えるはずである。

命題（勾配は等位集合に直交する） *ℓ は点 $\mathbf{x}$ で微分可能であるとする。このとき $\nabla\ell(\mathbf{x})$ は、点 $\mathbf{x}$ において等位集合 $\{\mathbf{x}' : \ell(\mathbf{x}') = c\}$（$\mathbf{x}$ を通るもの）に接するすべての方向と直交する。*

証明： ベクトル $\mathbf{v}$ が点 $\mathbf{x}$ において等位集合に**接する**とは、$\mathbf{v} = \gamma'(0)$ となる微分可能な曲線 $\gamma(t)$ が等位集合内に存在し、$\gamma(0) = \mathbf{x}$ を満たすことをいう。このような曲線に沿うと、$\ell(\gamma(t)) = c$ がすべての t で成り立つため、$t \mapsto \ell(\gamma(t))$ の $t = 0$ における微分係数はゼロでなければならない。1 次の範囲では $\gamma(t) = \mathbf{x} + t\mathbf{v} + o(t)$ であるから、Equation 34.18 よりこの微分係数は $\mathbf{v} \cdot \nabla\ell(\mathbf{x})$ に等しい。ゆえに $\mathbf{v} \cdot \nabla\ell(\mathbf{x}) = 0$ が成り立つ。$\blacksquare$

例題でこれを確かめてみよう。f の $(0, \log 2)$ を通る等位集合は曲線 $e^x + e^y = 3$ である。この曲線に沿って微分すると $e^x\,dx + e^y\,dy = 0$ となるので、$(0, \log 2)$ における接線方向は $(e^y, -e^x) = (2, -1)$ である。勾配とのドット積をとると $(2,-1)\cdot(\frac{1}{3}, \frac{2}{3}) = 0$ となり、先ほど見つけた傾きゼロの方向と確かに一致する。

したがって等高線図の上では、勾配はすべての等高線と直角に交わり、値が大きくなる方向を指す。さらに、隣り合う等高線の値の差が Δc で間隔が d であれば、等高線を横切る傾きはおおよそ $\Delta c/d$ になるため、等高線が最も密に詰まっている場所ほど勾配ベクトルは長くなる（Figure 34.12）。

34.2.2.3 接平面

線形近似 Equation 34.18 から誤差項を取り除き、予測される高さを z と書くと、f のグラフの点 $\mathbf{x}_0$ における**接平面**（tangent plane）が得られる。

$$z = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0). \tag{34.20}$$

点 $(0, \log 2)$ における例題では、この平面は $z = \log 3 + \frac{1}{3}x + \frac{2}{3}(y - \log 2)$ となる。Equation 34.20 を $\nabla f(\mathbf{x}_0)\cdot(\mathbf{x} - \mathbf{x}_0) - (z - f(\mathbf{x}_0)) = 0$ と書き直せば、ベクトル $[\nabla f(\mathbf{x}_0), -1]^{\top}$ が $(\mathbf{x}, z)$

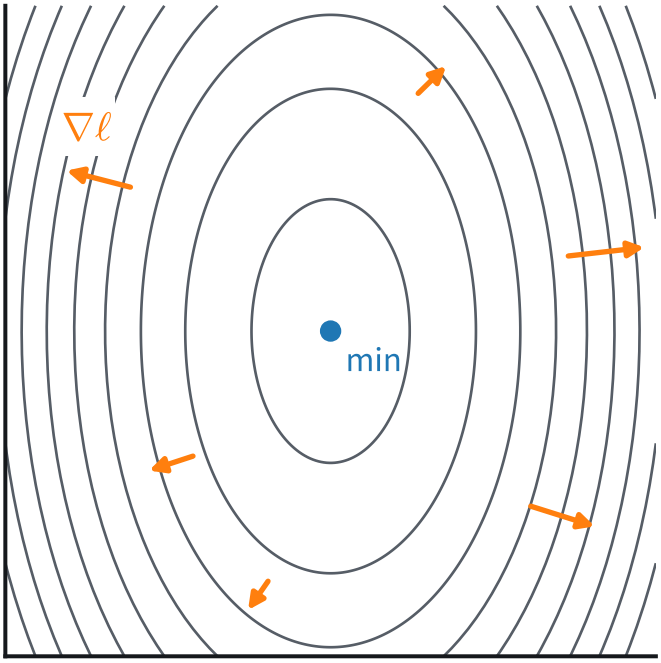


Figure 34.12: 損失 ℓ の等高線（灰色）と複数の点における $\nabla\ell$。各勾配は等高線に直交し、値が大きくなる方向を指し、等高線が最も密に詰まっている場所で最も長くなる。

空間における接平面の法線ベクトルであることが直ちにわかる。そこから高さ座標を落とすと $\nabla f(\mathbf{x}_0)$ が残り、これは底平面における等位曲線の法線ベクトルになっている（Figure 34.13）。

34.2.2.4 臨界点

深層学習の損失は閉じた形で最小化するにはあまりに複雑だが、それでも線形モデルはあらゆる最小点が満たすべき必要条件を与えてくれ、勾配降下法はまさにこの条件が成り立つ場所で停止する。以下では、f は $\mathbb{R}^n$ 全体で定義され、点 $\mathbf{x}_0$ で微分可能であるとする。

もし $\nabla f(\mathbf{x}_0) \neq \mathbf{0}$ ならば、$\epsilon = -\eta \nabla f(\mathbf{x}_0)$（$\eta > 0$）とおいて勾配と逆向きにステップを進めることができる。Equation 34.18 より、このときの f の変化量は $-\eta \|\nabla f(\mathbf{x}_0)\|^2 + o(\eta)$ であり、十分小さいすべての η に対して真に負となる。したがって $\mathbf{x}_0$ は極小点ではあり得ず、$+\eta$ を用いて同じ議論を繰り返せば極大点（local maximum）でもあり得ないことがわかる。ゆえに、**いかなる極小点・極大点においても $\nabla f(\mathbf{x}_0) = \mathbf{0}$ が成り立つ**。勾配がゼロになる点は**臨界点**（critical point）と呼ばれる。この一階条件（first-order condition）は必要条件であって十分条件ではない。大域的な極値を見つけたと言い切るためにあと何が必要かを、次の例で確認しよう。

$\psi(x, y) = x e^{-x^2 - y^2}$ を考えよう。これは $x > 0$ に正の山、$x < 0$ に対称な負のくぼみを持つ曲面である。その偏微分は次式で与えられる。

$$\frac{\partial \psi}{\partial x} = (1 - 2x^2)\, e^{-x^2 - y^2}, \qquad \frac{\partial \psi}{\partial y} = -2xy\, e^{-x^2 - y^2}.$$

第 2 の偏微分をゼロとおくと、$x = 0$ または $y = 0$ が導かれる。$x = 0$ とすると第 1 の偏微分が $e^{-y^2} \neq 0$ となってしまうため $y = 0$ でなければならず、すると $1 - 2x^2 = 0$ から $x = \pm 1/\sqrt{2}$

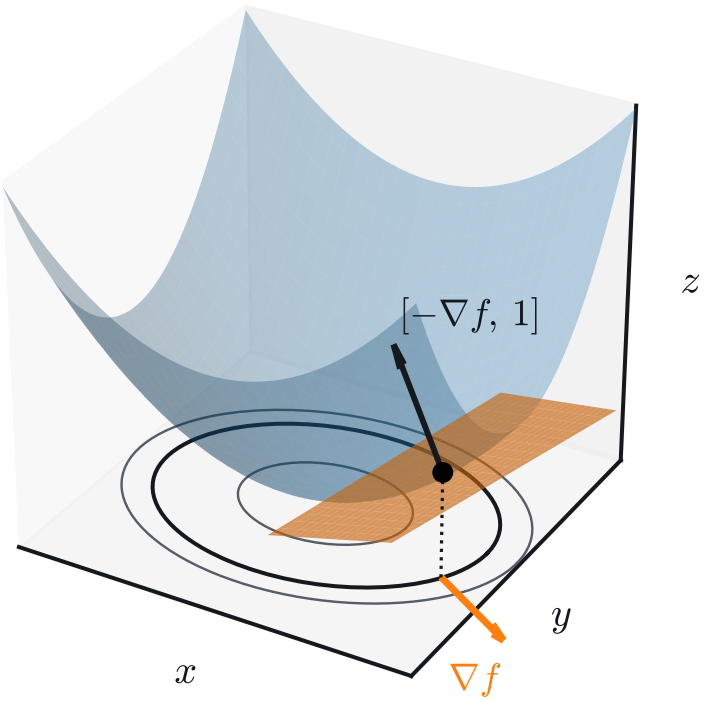


Figure 34.13: 曲面 $z = f(\mathbf{x})$（青）とある点における接平面（オレンジ）。ベクトル $[-\nabla f, 1]$（$[\nabla f, -1]$ の符号を反転したもの）は接平面に直交する（黒の矢印）。高さ座標を落とすと ∇f の定数倍が残り、底平面上（オレンジの矢印）ではその点を通る等位曲線（黒）に直交する。

が得られる。これら 2 つの臨界点における関数値は $\psi(\mp 1/\sqrt{2}, 0) = \mp e^{-1/2}/\sqrt{2} \approx \mp 0.4289$ である。

しかし、$(-1/\sqrt{2}, 0)$ が**大域的**最小点（global minimum）であると結論づけるには、そもそも大域的最小値が存在することも確かめなければならない。半径 $R = 2$ の円板の外側では、指数減衰（exponential decay）によって $|\psi| \le R e^{-R^2} < 0.04$ に抑えられるのに対し、円板の内部では $\psi(-1/\sqrt{2}, 0) \approx -0.43$ をとる。ψ は連続であるから閉円板上で最小値を達成するが、その最小値は -0.04 を下回るため境界上にはあり得ず、ψ の $\mathbb{R}^2$ 全体における大域的最小点、したがって臨界点でなければならない。2 つの候補を比較すれば $(-1/\sqrt{2}, 0)$ が大域的最小点であると特定でき、同じ議論から $(+1/\sqrt{2}, 0)$ が大域的最大点（global maximum）であるとわかる。存在確認を省くと論理が破綻する。たとえば x^3 は臨界点をただ 1 つ持つが、どこにも最小点を持たない。

一階の情報だけではできないことにも注意しておこう。最小点でも最大点でも勾配はゼロになるため、勾配だけでは両者を区別できない。それを見分けるために、二階導関数へと進む（34.2.3.3 節）。

34.2.3 ヘッセ行列

34.2.3.1 ヘッセ行列の対称性

線形モデルが捉えるのは 1 次の振る舞いだけである。臨界点が極小点か、極大点か、あるいはそのどちらでもないかは、曲面が接平面からどのように曲がって離れていくかに依存し、その曲率は二階導関数によって測られる。n 変数の関数には、次のように n^2 個の二階偏微分がある。

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}\right),$$

これらをまとめたものが**ヘッセ行列**である。

$$\mathbf{H}_f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}. \tag{34.21}$$

(i, j) 成分は、j 番目の傾きが座標 x_i に沿って動いたときにどれほど速く変化するかを測る。ヘッセ行列は $\nabla^2 f$ とも書かれ、対象となる関数と点が文脈から明らかなときは単に $\mathbf{H}$ と書く。

log-sum-exp について、$(x_1, x_2) = (x, y)$ と書いてソフトマックスの成分 $p_i = e^{x_i}/(e^{x_1} + e^{x_2})$ をもう 1 回微分してみよう。商の微分法則を適用すると $\partial p_i / \partial x_j = p_i \delta_{ij} - p_i p_j$ が得られる。ここで δ_{ij} は、1（$i = j$ のとき）または 0（それ以外のとき）をとる記号である。行列形式で書くと次のようになる。

$$\begin{aligned} \mathbf{H}_f &= \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^{\top}, \\ \mathbf{H}_f(0, \log 2) &= \begin{bmatrix} \frac{1}{3} - \frac{1}{9} & -\frac{2}{9} \\ -\frac{2}{9} & \frac{2}{3} - \frac{4}{9} \end{bmatrix} = \frac{2}{9}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \end{aligned} \tag{34.22}$$

この 2×2 行列は、固有値 0（$(1, 1)$ 方向）と固有値 $\frac{4}{9}$（$(1, -1)$ 方向）を持つ。ゼロ固有値はすでに見た事実と対応している。対角方向 $(1, 1)$ に沿うと $f(x + t, y + t) = f(x, y) + t$ は完全に

線形であり、曲率をまったく持たない。この半正定値（positive semidefinite）で特異なヘッセ行列には、のちほど二階微分判定法で再び出会うことになる。

ヘッセ行列の n^2 個の成分は互いに独立ではない。f が 2 回連続微分可能であれば微分の順序は結果に影響せず、ヘッセ行列は対称行列になる。これはクレロー・シュワルツの定理（Clairaut–Schwarz theorem）として知られる古典的な結果である (Rudin 1976)。この対称性はきわめて重要である。スペクトル定理（33.3 節）によって実固有値と互いに直交する固有ベクトルが保証され、それによって二階微分判定法が可能になるからである。

命題（ヘッセ行列の対称性：クレロー・シュワルツの定理） f **の一階偏微分、ならびに混合偏微分** $\dfrac{\partial^2 f}{\partial x_i \partial x_j}$ **と** $\dfrac{\partial^2 f}{\partial x_j \partial x_i}$ **が点** $\mathbf{x}$ **の近傍で存在し、さらにこれらの混合偏微分が点** $\mathbf{x}$ **で連続であるとする。このとき次式が成り立つ。**

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}) = \frac{\partial^2 f}{\partial x_j \partial x_i}(\mathbf{x}).$$

これがすべての i, j **について成り立つとき、ヘッセ行列は対称となる（**$\mathbf{H}_f = \mathbf{H}_f^{\top}$**）。**

証明：証明の着想は、両方の混合偏微分を**同一の**量、すなわち次の対称な二階差分の極限として表すことにある。

$$\Delta_h = \frac{f(\mathbf{x} + h\mathbf{e}_i + h\mathbf{e}_j) - f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} + h\mathbf{e}_j) + f(\mathbf{x})}{h^2}, \tag{34.23}$$

この式は、見た目から明らかなように i と j を入れ替えても変わらない。ここでの課題は、Δ_h を近くの点で評価した混合偏微分として表すことである。1 変数関数 $\varphi(t) = f(\mathbf{x} + t\mathbf{e}_i + h\mathbf{e}_j) - f(\mathbf{x} + t\mathbf{e}_i)$ を定めると、$h^2 \Delta_h = \varphi(h) - \varphi(0)$ となる。平均値の定理（34.1.3 節）を 2 回、まず φ に対して適用し、次に x_j に沿って φ' の内部にある一階偏微分の差に対して適用すると、ある数 τ と σ が 0 と h の間に存在して次式を満たす。

$$\begin{aligned} \varphi(h) - \varphi(0) &= h\,\varphi'(\tau) = h\Big[\frac{\partial f}{\partial x_i}(\mathbf{x} + \tau\mathbf{e}_i + h\mathbf{e}_j) - \frac{\partial f}{\partial x_i}(\mathbf{x} + \tau\mathbf{e}_i)\Big] \\ &= h^2\,\frac{\partial^2 f}{\partial x_j \partial x_i}(\mathbf{x} + \tau\mathbf{e}_i + \sigma\mathbf{e}_j). \end{aligned}$$

したがって、$\Delta_h = \dfrac{\partial^2 f}{\partial x_j \partial x_i}(\xi_h)$ が成り立つ。ここで点 ξ_h は、$\mathbf{x}$ と $\mathbf{x} + h\mathbf{e}_i + h\mathbf{e}_j$ を対角の頂点とする正方形の内部にあり、高々 $\sqrt{2}\,|h|$ の距離に $\mathbf{x}$ から収まる点である。i と j の役割を入れ替えても Δ_h は不変であり、同じ正方形内の別の点において $\Delta_h = \dfrac{\partial^2 f}{\partial x_i \partial x_j}(\xi_h')$ が得られる。$h \to 0$ とすると、点 $\mathbf{x}$ における混合偏微分の連続性により、2 つの式はそれぞれ $\mathbf{x}$ における 2 つの混合偏微分へと収束する。すべての h に対して両者は一致するため、その極限も一致しなければならない。$\blacksquare$

連続性の仮定は省くことができない。実際、混合偏微分がいたるところで存在するにもかかわらず原点で連続でなくなり、そこで両者の値が食い違う関数が存在する。

34.2.3.2 2 次テイラー近似

勾配が f に対する最良の線形近似を与えるのとちょうど同じように、ヘッセ行列は最良の二**次**近似を与える。係数がどこから現れるかを見るには、まず厳密な二次関数から係数を読み取ってみるのがわかりやすい。$f(x_1, x_2) = a + b_1 x_1 + b_2 x_2 + c_{11} x_1^2 + c_{12} x_1 x_2 + c_{22} x_2^2$ をとり、原点における関数値、勾配、およびヘッセ行列 Equation 34.21 を計算すると次のようになる。

$$f(0,0) = a, \qquad \nabla f(0,0) = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \qquad \mathbf{H}_f(0,0) = \begin{bmatrix} 2c_{11} & c_{12} \\ c_{12} & 2c_{22} \end{bmatrix},$$

これらの要素を組み合わせると、元の多項式 $f(\mathbf{x}) = f(\mathbf{0}) + \nabla f(\mathbf{0}) \cdot \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{H}_f(\mathbf{0})\, \mathbf{x}$ が厳密に復元される。係数 $\frac{1}{2}$ は、二乗を微分したときに生じる係数 2 を打ち消しているにすぎない。

一般の関数 f に対してはこの二次関数はもはや厳密には一致しないが、f を線分上に制限することでその近似精度がわかる。f が $\mathbf{x}_0$ と $\mathbf{x}$ を結ぶ線分の近傍で 2 回連続微分可能であるとする。変位を $\delta = \mathbf{x} - \mathbf{x}_0$ と書き、1 変数スライス $\varphi(t) = f(\mathbf{x}_0 + t\delta)$ を考える。線形モデル Equation 34.18 より、その一階導関数は $\varphi'(t) = \sum_i \frac{\partial f}{\partial x_i}(\mathbf{x}_0 + t\delta)\, \delta_i$ である。この和に現れる各偏微分もまた微分可能であるから、同じ操作をもう 1 回適用すれば二階導関数 $\varphi''(t) = \sum_{i,j} \delta_j \frac{\partial^2 f}{\partial x_j \partial x_i}(\mathbf{x}_0 + t\delta)\, \delta_i = \delta^\top \mathbf{H}_f(\mathbf{x}_0 + t\delta)\, \delta$ が得られる。

ここで、ラグランジュの剰余項を持つ 1 変数の 1 次テイラーの定理（34.1.5.2 節）を φ の区間 0 から 1 に適用しよう。するとある $\tau \in (0,1)$ が存在して $\varphi(1) = \varphi(0) + \varphi'(0) + \frac{1}{2}\varphi''(\tau)$ が成り立つ。これを f の言葉に戻すと次式が得られる。

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot \delta + \tfrac{1}{2} \delta^\top \mathbf{H}_f(\xi)\, \delta, \qquad \xi = \mathbf{x}_0 + \tau\delta.$$

この恒等式は厳密であるが、その代償としてヘッセ行列を線分上の未知の中間点 ξ で評価しており、基点 $\mathbf{x}_0$ での値にはなっていない。しかし二階偏微分は連続であるため、$\mathbf{H}_f(\xi)$ を $\mathbf{H}_f(\mathbf{x}_0)$ に置き換えても、二次の項の変化は高々 $\frac{1}{2}\|\mathbf{H}_f(\xi) - \mathbf{H}_f(\mathbf{x}_0)\|\, \|\delta\|^2 = o(\|\delta\|^2)$ に抑えられる。これにより **2 次テイラー近似**（second-order Taylor approximation）が得られる。

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) \\ &\quad + \frac{1}{2} (\mathbf{x} - \mathbf{x}_0)^\top \mathbf{H}_f(\mathbf{x}_0) (\mathbf{x} - \mathbf{x}_0) + o(\|\mathbf{x} - \mathbf{x}_0\|^2). \end{aligned} \tag{34.24}$$

この二次モデルと f とのずれはステップサイズの二乗より速く消える誤差だけであり、そのような性質を持つ二次関数はこれに限られる。f が 3 回連続微分可能であれば、誤差はステップに関して 3 次となる。

任意の単一方向 $\mathbf{v}$ に沿ってステップ $t\mathbf{v}$ をとると、二次の項 $\frac{t^2}{2}\mathbf{v}^\top \mathbf{H} \mathbf{v}$ が生じる。この値 $\mathbf{v}^\top \mathbf{H} \mathbf{v} = \frac{d^2}{dt^2} f(\mathbf{x}_0 + t\mathbf{v})\big|_{t=0}$ は**二階方向微分**（second directional derivative）と呼ばれ、f の $\mathbf{v}$ に沿ったスライスの曲率を測る。

これで、最初のコードセルで線形モデルが残した誤差を説明できる。ステップ $\epsilon = (0.01, -0.03)$ を点 $(0, \log 2)$ から進めたとき、ヘッセ行列 Equation 34.22 が予測する 2 次の補正項は $\frac{1}{2}\epsilon^\top \mathbf{H} \epsilon = \frac{1}{9}(\epsilon_1 - \epsilon_2)^2 = \frac{1}{9}(0.04)^2 \approx 1.78 \times 10^{-4}$ となり、実測された誤差 1.786×10^{-4} とよく一致する。$\mathbf{H}$ の固有値と予測される誤差の両方をコードで確かめてみよう。

```
p = grad_lse(xy0)                # the softmax at (0, log 2)
H = np.diag(p) - np.outer(p, p)  # Hessian of log-sum-exp
print('9 H =', (9 * H).round(6).tolist())
print('eigenvalues:', np.linalg.eigvalsh(H).round(6) + 0.0)  # no '-0.'
for scale in [1, 0.1]:
    eps = scale * np.array([0.01, -0.03])
    print(f'|eps| {np.linalg.norm(eps):.1e}: '
          f'predicted gap {0.5 * eps @ H @ eps:.4e}')
```

```
9 H = [[2.0, -2.0], [-2.0, 2.0]]
eigenvalues: [0.       0.444444]
|eps| 3.2e-02: predicted gap 1.7778e-04
|eps| 3.2e-03: predicted gap 1.7778e-06
```

行列そのものも、その固有値 0 と $\frac{4}{9}$ も Equation 34.22 と一致している。予測された誤差 1.7778×10^{-4} と 1.7778×10^{-6} は、実測値 1.786×10^{-4} と 1.779×10^{-6} に有効数字 2〜3 桁で一致する。最初の誤差に残っている約 10^{-6} のわずかなずれは 3 次の項に由来するものであり、ステップを 10 分の 1 に縮めるとその残差はおよそ 1,000 分の 1 に減少し、表示桁の下へと消えてしまう。

log-sum-exp の例では、1 つの方向が完全に平坦であった。今度は両方の方向に曲がる関数として、$\psi(x,y) = xe^{-x^2-y^2}$ を基点 $\mathbf{x}_0 = (-1,0)$ のまわりで見てみよう。先ほど求めた偏微分から $\psi(-1,0) = -e^{-1}$ および $\nabla\psi(-1,0) = (-e^{-1},0)$ が得られ、もう 1 回微分すると $\mathbf{H}_\psi(-1,0) = 2e^{-1}\mathbf{I}$ となる。これらを Equation 34.24 に代入すると、二次モデル $q(x,y) = e^{-1}(-1-(x+1)+(x+1)^2+y^2)$ が得られる。Figure 34.14 に示すように、曲面とその二次近似は $(-1,0)$ の近くではほとんど見分けがつかず、遠くへ離れるにつれて乖離していく。

この二次モデルの誤差が確かに 3 次であるなら、ステップサイズを 2 倍にすると誤差は約 8 倍になるはずである。次のセルでは、ψ と q を各座標方向にステップ d だけ $(-1,0)$ から動かした点で評価して確かめる。

```
def psi(x, y):
    return x * np.exp(-x**2 - y**2)

def quad(x, y):  # second-order Taylor model of psi at (-1, 0)
    return np.exp(-1) * (-1 - (x + 1) + (x + 1)**2 + y**2)

for d in [0.0, 0.05, 0.1, 0.3]:  # step d in each coordinate from (-1, 0)
    x, y = -1 + d, d
    print(f'step {d:.2f}: psi = {psi(x, y):.6f}, quadratic = {quad(x, y):.6f}, '
          f'gap = {abs(psi(x, y) - quad(x, y)):.6f}')
```

```
step 0.00: psi = -0.367879, quadratic = -0.367879, gap = 0.000000
step 0.05: psi = -0.384315, quadratic = -0.384434, gap = 0.000119
step 0.10: psi = -0.396388, quadratic = -0.397310, gap = 0.000921
step 0.30: psi = -0.391929, quadratic = -0.412025, gap = 0.020096
```

予想どおり、基点では誤差がゼロになり、ステップを 0.05 から 0.1 へ 2 倍にすると誤差は 8 倍弱（0.000119 から 0.000921）に増えて、3 次の剰余項の振る舞いと一致する。各ステップでこの局所的な二次関数を当てはめ、**その二次関数の**最小点へ一気に飛ぶ手法がニュートン法であり、1 変数の場合は 34.1.5.5 節で導入し、多変数のオプティマイザとしては 9.1.5.1 節で展開する。

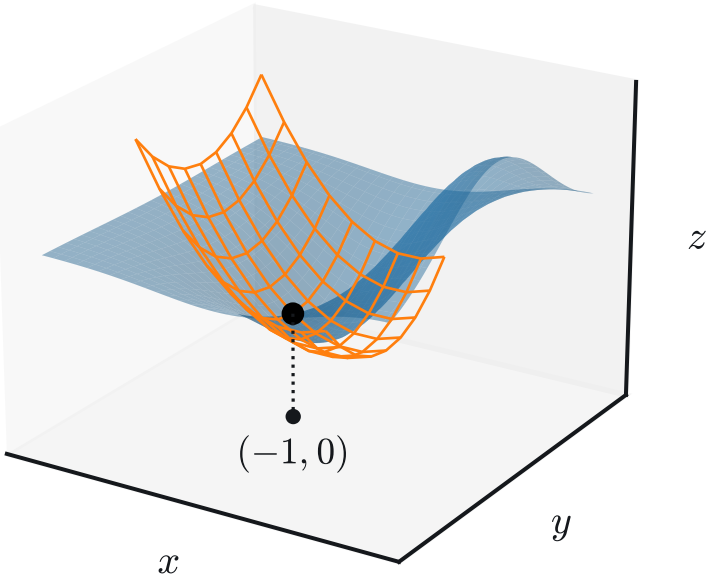


Figure 34.14: 曲面 $z = xe^{-x^2-y^2}$（青）と基点 $(-1, 0)$ における 2 次テイラー二次関数（オレンジ）。両者は基点で値・傾き・曲率が一致するため、基点の近くではほとんど見分けがつかず、外側へ離れると分かれていく。

34.2.3.3 二階微分判定法

臨界点 $\mathbf{x}_0$ では、Equation 34.24 の一次の項が消える。したがって $\mathbf{x}_0$ の近傍では、関数は $o(\|\mathbf{x}-\mathbf{x}_0\|^2)$ の誤差を除いて二次の項のように振る舞う。そのため、曲率がゼロでない限り、f が方向 $\mathbf{v}$ に離れたときに増加するか減少するかは $\mathbf{v}^{\top}\mathbf{H}\mathbf{v}$ の符号によって決まる。

$\mathbf{H}$ の固有値を使えば、これらの符号をすっきりと整理できる。$\mathbf{H}$ は対称行列であるため、スペクトル定理（33.3 節）により、固有ベクトル $\mathbf{q}_i$ からなる正規直交基底と実固有値 λ_i の存在が保証される。この基底で展開すると、二次形式（quadratic form）は二乗の単純な重み付き和になる。

$$\mathbf{v}^{\top}\mathbf{H}\mathbf{v} = \sum_{i=1}^{n} \lambda_i \, (\mathbf{q}_i^{\top}\mathbf{v})^2. \tag{34.25}$$

この式から局所的な幾何構造が一目で読み取れる。すべての $\lambda_i > 0$ が成り立てば、曲率はあらゆる方向で真に正となり、少なくとも $\lambda_{\min}\|\mathbf{v}\|^2$ 以上になる。一方、固有値に正と負の両方の符号が混在していれば、曲面はある固有ベクトルに沿っては上向きに曲がり、別の固有ベクトルに沿っては下向きに曲がる。

命題（二階条件） *f は臨界点 $\mathbf{x}_0$ の近傍で 2 回連続微分可能であるとし、$\mathbf{H} = \mathbf{H}_f(\mathbf{x}_0)$ と書く。*

1. *$\mathbf{x}_0$ が極小点ならば、$\mathbf{H} \succeq 0$（すべての固有値が ≥ 0）である。*
2. *$\mathbf{H} \succ 0$（すべての固有値が > 0）ならば、$\mathbf{x}_0$ は狭義の極小点である。*
3. *$\mathbf{H}$ が正負両方の符号の固有値を持つならば、ゼロ固有値の有無にかかわらず $\mathbf{x}_0$ は極小点でも極大点でもなく、鞍点である。*

極大点に関する対応する主張は、この命題を $-f$ に適用すれば得られる。$\mathbf{H}$ が特異かつ半定値であるときは 2 も 3 も適用できず、二階微分判定法では判定不能となる。

証明： (1) 背理法のため、$\mathbf{v}^{\top}\mathbf{H}\mathbf{v} < 0$ となる方向 $\mathbf{v}$ が存在すると仮定する。1 変数スライス $\varphi(t) = f(\mathbf{x}_0 + t\mathbf{v})$ は $\varphi'(0) = \nabla f(\mathbf{x}_0)\cdot\mathbf{v} = 0$ かつ $\varphi''(0) = \mathbf{v}^{\top}\mathbf{H}\mathbf{v} < 0$ を満たす。1 変数の二階微分判定法により、φ は $t=0$ で狭義の極大となるため、f は $f(\mathbf{x}_0)$ より真に小さい値を $\mathbf{x}_0$ のいくらでも近くでとることになり、$\mathbf{x}_0$ が極小点であるという仮定に矛盾する。

(2) すべての方向について同時に剰余項を一様に抑える必要があるため、このステップには注意を要する。$\nabla f(\mathbf{x}_0) = \mathbf{0}$ のもとで厳密なテイラーの恒等式を用いると、$f(\mathbf{x}_0+\delta) - f(\mathbf{x}_0) = \tfrac{1}{2}\delta^{\top}\mathbf{H}_f(\xi)\delta$ となる点 ξ が、$\mathbf{x}_0$ と $\mathbf{x}_0+\delta$ を結ぶ線分上に存在する。ヘッセ行列を $\mathbf{H}_f(\xi) = \mathbf{H} + \big(\mathbf{H}_f(\xi) - \mathbf{H}\big)$ と分解しよう。Equation 34.25 より、主要項は $\delta^{\top}\mathbf{H}\delta \geq \lambda_{\min}\|\delta\|^2$ を満たす。また、二階偏微分の連続性により、$\|\mathbf{H}_f(\xi)-\mathbf{H}\| \leq \tfrac{1}{2}\lambda_{\min}$ が十分小さいすべての δ に対して成り立つ。これら 2 つの評価を組み合わせると、$f(\mathbf{x}_0+\delta) - f(\mathbf{x}_0) \geq \tfrac{1}{2}(\lambda_{\min} - \tfrac{1}{2}\lambda_{\min})\|\delta\|^2 = \tfrac{1}{4}\lambda_{\min}\|\delta\|^2 > 0$ が十分小さいすべての $\delta \neq \mathbf{0}$ に対して成り立つ。

(3) $\lambda > 0$ である固有ベクトルに沿うと、スライス φ は $\varphi'(0)=0$ かつ $\varphi''(0) = \lambda > 0$ を満たすため、f は $f(\mathbf{x}_0)$ より真に大きい値を $\mathbf{x}_0$ のいくらでも近くでとる。同様に $\lambda < 0$ である固有ベクトルに沿うと、真に小さい値をとる。$\blacksquare$

3 つの主張のうち覚えておくべきは 2 番目であり、証明に実質的な手間がかかるのもこれだけである。また、判定不能となるケースは単なる理論上の注釈ではないことにも注意したい。ゼロ固有ベクトルに沿うと二次モデルは完全に平坦になり、高次の項が振る舞いを支配するからである。たとえば $x^2 + y^4$ と $x^2 - y^4$ は原点で同じヘッセ行列 $\operatorname{diag}(2,0)$ を共有するが、前者は原点で狭義の極小点を持つのに対し、後者は鞍点を持つ（演習問題 3 ではこの現象をさらに掘り下げる）。

Figure 34.15 は、これまでの例題の二次モデルを用いて 4 つのケースすべてを図示したものである。

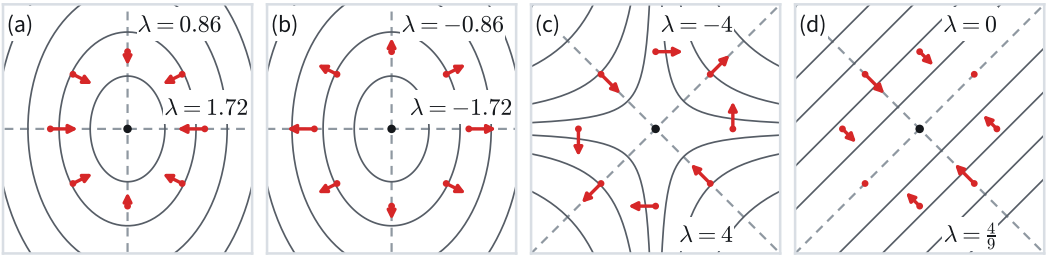


Figure 34.15: 二次モデル $\frac{1}{2}\mathbf{v}^\top \mathbf{H}\mathbf{v}$ の等高線（灰色）、固有値を添えた固有ベクトル方向（破線）、および降下方向の矢印 $-\mathbf{H}\mathbf{v}$（赤）。(a) $\mathbf{H} \succ 0$：極小点であり、すべての降下矢印が内側を向く。(b) $\mathbf{H} \prec 0$：極大点。(c) 不定値：鞍点であり、一方の固有ベクトルに沿って内側へ、他方に沿って外側へ流れる。(d) 特異かつ半定値：二次モデルはゼロ固有ベクトルに沿って平坦であり、2 次の情報では判定不能となる。

Figure 34.15 において、(a) は ψ の $(-1/\sqrt{2}, 0)$ におけるヘッセ行列 $e^{-1/2}\,\mathrm{diag}(2\sqrt{2}, \sqrt{2})$ を示し、(b) は $(+1/\sqrt{2}, 0)$ におけるその符号反転を示す。(c) は $x^4 + y^4 - 4xy$ の原点を示しており、そのヘッセ行列は $\left(\begin{smallmatrix} 0 & -4 \\ -4 & 0 \end{smallmatrix}\right)$ である。(d) は純粋に行列の例として log-sum-exp のヘッセ行列 Equation 34.22 を再利用したものである。log-sum-exp 自体は、その勾配であるソフトマックスが決してゼロにならないため臨界点を持たない。

二階微分判定法を数値的に実装し、平坦な方向を持つ点に対して実行すると何が起きるだろうか。有限差分（finite difference）から構成したヘッセ行列が厳密にゼロの固有値を返すことはまずないため、それ以下をゼロとみなす許容誤差を設定しなければならない。次のセルにおいて、中心混合差分 `fxy` は二階差分 Equation 34.23 をステップ $2h$、基点 $(x - h, y - h)$ で評価したものである。

```
def hessian(f, x, y, h=1e-4):  # 2x2 Hessian from second differences
    fxx = (f(x + h, y) - 2 * f(x, y) + f(x - h, y)) / h**2
    fyy = (f(x, y + h) - 2 * f(x, y) + f(x, y - h)) / h**2
    fxy = (f(x + h, y + h) - f(x + h, y - h)
           - f(x - h, y + h) + f(x - h, y - h)) / (4 * h**2)
    return np.array([[fxx, fxy], [fxy, fyy]])

def classify(lo, hi, tol=1e-6):  # from the smallest, largest eigenvalue
    if lo > tol:
        return 'minimum'
    if hi < -tol:
        return 'maximum'
    if lo < -tol < tol < hi:
        return 'saddle'
    return 'inconclusive'

quartic = lambda x, y: x**4 + y**4 - 4 * x * y
trough = lambda x, y: x**2 + y**4
for fn, x0 in [(psi, -1 / np.sqrt(2)), (psi, 1 / np.sqrt(2)),
               (quartic, 0.0), (trough, 0.0)]:
    lo, hi = np.linalg.eigvalsh(hessian(fn, x0, 0.0))
```

```
        print(f'({x0:+.4f}, 0): eigenvalues {lo:+.3f}, {hi:+.3f} '
              f'-> {classify(lo, hi)}')
```

```
(-0.7071, 0): eigenvalues +0.858, +1.716 -> minimum
(+0.7071, 0): eigenvalues -1.716, -0.858 -> maximum
(+0.0000, 0): eigenvalues -4.000, +4.000 -> saddle
(+0.0000, 0): eigenvalues +0.000, +2.000 -> inconclusive
```

点 $(-1/\sqrt{2}, 0)$ では固有値は 0.858 と 1.716（それぞれ $\sqrt{2}e^{-1/2}$ と $2\sqrt{2}e^{-1/2}$ に一致）となり、先ほど見つけた大域的最小点であることが裏づけられる。点 $(+1/\sqrt{2}, 0)$ では両方の符号が反転し、極大点であることが確認できる。$x^4 + y^4 - 4xy$ の原点では固有値は ± 4 となり、上昇方向と下降方向である $(1, -1)$ と $(1, 1)$ が座標軸から回転した鞍点を与える。示唆に富むのは 4 番目の点である。$x^2 + y^4$ の原点では、有限差分は平坦な方向に沿って $2h^2 = 2 \times 10^{-8}$ を返すが、これは許容誤差を下回るため、原点が実際には狭義の極小点であるにもかかわらず、プログラムは正しく「inconclusive」（判定不能）と報告する。2 次の情報からは 4 次の立ち上がりがそもそも見えず、許容誤差のおかげで数値判定が誤った結論を下さずに済んでいるのである。

34.2.3.4 高次元における鞍点

n 次元において、極小点であるためにはヘッセ行列の n 個すべての固有値が非負でなければならず、非退化な極小点であるためには n 個すべてが真に正でなければならない。もしランダムな臨界点における固有値の符号が独立なコイン投げのように振る舞うなら、n 個すべてが正になる確率は 2^{-n} であり、n が大きければそれだけでも無視できるほど小さくなる。

驚くべきことに、このコイン投げの経験則は実際には極小点が見つかる確率を**過大評価**している。対角成分が非対角成分の 2 倍の分散を持つランダム対称行列、すなわちガウス直交アンサンブル（Gaussian orthogonal ensemble、GOE）を考えよう。$n = 2$ ですら、両方の固有値が正になる厳密な確率は $(1 - 1/\sqrt{2})/2 \approx 0.146$ であり、0.25 より小さい（演習問題 4）。大きな n に対しては、この確率は指数関数よりはるかに速く $\exp\!\left(-\tfrac{\ln 3}{4} n^2\right)$ のオーダーで減衰する (Dean and Majumdar 2006)。ランダム対称行列の固有値は互いに反発し合い、ゼロの片側に密集することがほとんどないからである。

したがって高次元のランダムなランドスケープでは、典型的な（高い）関数値にある臨界点は圧倒的多数が鞍点である。関数値が低い領域に入って初めてヘッセ行列の固有値が正の側へシフトし、極小点が支配的になり始める (Bray and Dean 2007)。ニューラルネットワークも同じ振る舞いを示す。小規模なネットワークの損失曲面上でニュートン法により臨界点を探すと、損失が大きい臨界点は鞍点ばかりである (Dauphin et al. 2014)。実用上、オプティマイザが高い損失のまま停滞したとき、質の悪い極小点に陥ったという説明は最も可能性が低い。オプティマイザは鞍点の近くでもたついているか、あるいはさらに多いケースとして、勾配がまったく小さくない悪条件の領域を横切っている最中なのである。

34.2.4 制約付き最適化とラグランジュ乗数

ここまでは、変数はどの方向にも自由に動けるものとしてきた。しかし多くの機械学習の問題では、パラメータはある曲面上に制限される。単位長に正規化された重み、和が 1 になるよう制約された確率、あるいは曲線上で原点に最も近い点などがその例である。これらの制約付き最適化（constrained optimization）問題ではいずれも、$f(\mathbf{x})$ を制約（constraint）$g(\mathbf{x}) = c$ のもとで最小化したい。制約なしの条件 $\nabla f = \mathbf{0}$ はもはや成り立たない。最良の実行可能点（feasible point）において ∇f がゼロになる必要はなく、制約曲面上に留まる移動に対して 1 次の減少を生まないだけで十分だからである。注目すべきことに、$\nabla f = \mathbf{0}$ に取って代わる条件は、二階導関数を一切使わずに等位集合の幾何学から直接導かれる。

f と g は微分可能であるとする。実行可能点において許容される無限小移動は、方向 $\mathbf{v}$ のうち制約曲面 $\{g = c\}$ に接するものである。制約付きの局所最小点 $\mathbf{x}^\star$ では、いかなる許容移動も 1 次の範囲で f を減少させられないため、$\mathbf{v} \cdot \nabla f(\mathbf{x}^\star) \geq 0$ がすべての接線方向 $\mathbf{v}$ に対して成り立つ。$-\mathbf{v}$ もまた接線方向であるから、実は等号が成り立たなければならず、$\nabla f(\mathbf{x}^\star)$ はすべての接線方向と直交する。一方、等位集合の命題より $\nabla g(\mathbf{x}^\star)$ もすべての接線方向と直交するため、2 つの勾配は同じ直線上に並ばざるを得ない。

命題（ラグランジュ条件）$\mathbf{x}^\star$ を f の $\{\mathbf{x} : g(\mathbf{x}) = c\}$ 上における局所最小点とする。f は点 $\mathbf{x}^\star$ で微分可能であり、g は $\mathbf{x}^\star$ の近傍で連続微分可能であり、かつ $\nabla g(\mathbf{x}^\star) \neq \mathbf{0}$ であるとする。このとき、ラグランジュ乗数（Lagrange multiplier）と呼ばれるスカラー μ が存在して次式を満たす。

$$\nabla f(\mathbf{x}^\star) = \mu \, \nabla g(\mathbf{x}^\star). \tag{34.26}$$

証明： 証明は制約集合に関する 1 つの幾何学的事実に基づく。後半の 2 つの仮定のもとで、$\{g = c\}$ は $\mathbf{x}^\star$ の近傍で滑らかな曲面をなし、任意のベクトル $\mathbf{v}$ が $\nabla g(\mathbf{x}^\star) \cdot \mathbf{v} = 0$ を満たすならば、それは速度ベクトル $\gamma'(0)$ として、微分可能な曲線 γ で $\{g = c\}$ 内に含まれ $\gamma(0) = \mathbf{x}^\star$ を満たすものから得られる (Rudin 1976)。このような曲線に沿うと、1 変数関数 $t \mapsto f(\gamma(t))$ は $t = 0$ で極小となるため、そこでの微分係数はゼロでなければならない。等位集合の証明と同様、その微分係数は $\mathbf{v} \cdot \nabla f(\mathbf{x}^\star)$ に等しい。したがって、$\nabla f(\mathbf{x}^\star)$ は、ベクトル $\mathbf{v}$ のうち $\nabla g(\mathbf{x}^\star)$ に直交するもの全体がなす超平面と直交する。その超平面と直交するベクトルは $\nabla g(\mathbf{x}^\star)$ のスカラー倍に限られるため、$\nabla f(\mathbf{x}^\star) = \mu \nabla g(\mathbf{x}^\star)$ が成り立つ。$\blacksquare$

幾何学的には、制約付き最適点において f と g の勾配は平行であり、つまり f の $\mathbf{x}^\star$ を通る等位集合は制約曲面に接している。代数的には、n 個の式からなる Equation 34.26 と制約 $g(\mathbf{x}) = c$ を合わせることで、$n + 1$ 個の方程式が $n + 1$ 個の未知数 $\mathbf{x}$ と μ に対して得られる。

具体例として、双曲線 $xy = 1$ 上で原点に最も近い点を求めてみよう。すなわち、$f(x, y) = x^2 + y^2$ を制約 $g(x, y) = xy = 1$ のもとで最小化したい。勾配は $\nabla f = (2x, 2y)$ と $\nabla g = (y, x)$ であり、双曲線上のいたるところで $\nabla g \neq \mathbf{0}$ が成り立つ。ラグランジュ条件から $2x = \mu y$ と $2y = \mu x$ が得られる。第 2 式を第 1 式に代入すると $4x = \mu^2 x$、すなわち $x(4 - \mu^2) = 0$ となる。$x = 0$ は $xy = 1$ 上ではあり得ないため、$\mu = \pm 2$ でなければならない。$\mu = -2$ を選ぶと $y = -x$ となり、したがって $xy = -x^2 = 1$ となって実数解を持たない。残るのは $\mu = 2$ すなわち $x = y$ であり、2 つの候補 $(1, 1)$ と $(-1, -1)$ が得られ、いずれも $f = 2$ をとる。各分枝上で $f = x^2 + 1/x^2$ は $x \to 0$ または $|x| \to \infty$ のとき限りなく大きくなるため大域的最小値が必ず存在し、これら 2 点が確かに最も近い点である。点 $(1, 1)$ において円 $x^2 + y^2 = 2$ は双曲線にぴったりと接し、$\nabla f = (2, 2) = 2 \, \nabla g$ が成り立っている（Figure 34.16）。

双曲線は幾何学的イメージを描きやすいが、同じ道具を使えば本節の共通の例題へとまっすぐ戻ってくる。$\mathbf{z} \cdot \mathbf{p} - \sum_i p_i \log p_i$ を確率ベクトル $\mathbf{p}$ について制約 $\sum_i p_i = 1$ のもとで最大化することを考えよう。ラグランジュ条件を書き下すと $z_i - \log p_i - 1 = \mu$ が各座標 i について得られ、これから $p_i \propto e^{z_i}$ が従う。したがって最適な分布はソフトマックスであり、達成される最大値は log-sum-exp となる（35.7 節）。

正則性の仮定 $\nabla g(\mathbf{x}^\star) \neq \mathbf{0}$ は落とすことができない。これがないと、ラグランジュ乗数がそもそも存在しない場合がある（演習問題 6 で明快な反例を扱う）。のちほど 35.6 節では KKT 条件を通じてこの条件を不等式制約へと拡張し、35.7 節では双対問題（dual problem）を展開する。

本節のすべての結果は、勾配がすでにわかっていることを前提としてきた。黒板に書ける 2 変数の式であれば、手計算で勾配を求めるのは造作もない。しかし 100 万個の重みを持つニューラルネットワークでは、勾配を効率よく求めることこそが難しい問題となる。

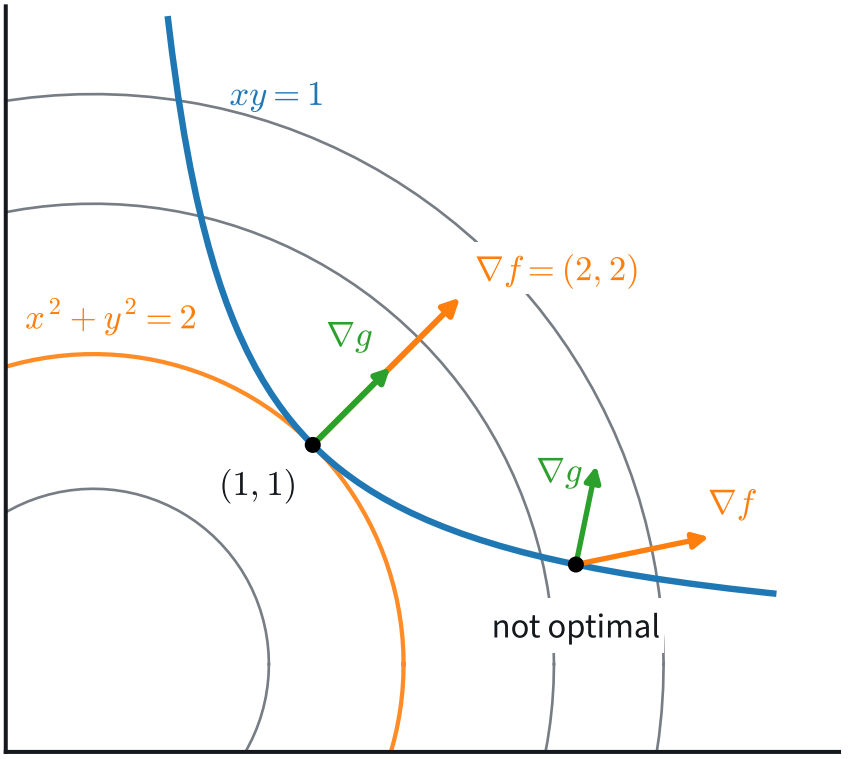


Figure 34.16: $xy = 1$ 上で原点に最も近い点。点 $(1, 1)$ では円 $x^2 + y^2 = 2$ が双曲線に接し、$\nabla f = (2, 2) = 2\nabla g$ となるため $\mu = 2$ である（鏡像解 $(-1, -1)$ は図の範囲外）。最適でない実行可能点では 2 つの勾配は異なる方向を指し、∇f は制約に沿った成分を残している。

34.2.5 まとめ

微分可能な関数は、1 次の範囲においてステップと勾配のドット積だけ変化する。単に偏微分が存在するだけでは、この性質は保証されない。幾何学的には、負の勾配は最急降下方向を指し、勾配はすべての等位集合に直交する。次数を 1 つ上げると、対称なヘッセ行列が局所的な二次近似を定める。臨界点においてはその固有値の符号が極小点・極大点・鞍点を判別し、ヘッセ行列が特異かつ半定値のときは判定不能となる。変数が制約曲面 $g(\mathbf{x}) = c$ 上に制限されるときには、等位集合の幾何学によって $\nabla f = \mathbf{0}$ がラグランジュ条件 $\nabla f = \mu \nabla g$ へと置き換えられる。

34.2.6 演習問題

1. **方向微分はあるが勾配を持たない関数。** $f(x,y) = x^2 y/(x^2+y^2)$（$(x,y) \neq (0,0)$ のとき）および $f(0,0) = 0$ とする。
   1. f が原点で連続であることを示せ。
   2. 原点における f の 2 つの偏微分と、任意の単位ベクトル $\mathbf{u} = [\cos\theta, \sin\theta]^\top$ に沿った原点での方向微分を計算せよ。
   3. いかなるベクトル $\mathbf{g}$ に対してもこれらの方向微分は $\mathbf{u}\cdot\mathbf{g}$ の形では表せず、したがって f は原点で微分可能でないことを示せ。34.2.1.2 節で述べた微分可能性の十分条件のうち、どの仮定が破れているか。また、同項の反例 $2xy/(x^2+y^2)$ では示せなかったどのような事実をこの関数は示しているか。

2. **引き伸ばされたボウルの勾配。** $f(x,y) = x^2 + \kappa y^2$（$\kappa \geq 1$）とする。その最小点は原点であり、$\kappa$ は f が y 方向に沿って x 方向よりどれだけ強く曲がっているかを測る。
   1. f の $(1,1)$ を通る等位曲線に接する方向を 1 つ求め、それが $\nabla f(1,1)$ と直交することを確かめよ。さらに、$-\nabla f(1,1)$ と、方向 $-[1,1]^\top$（$(1,1)$ から最小点へ向かう方向）のなす角を計算せよ。$(x,y) \neq (0,0)$ のうち、負の勾配が最小点をまっすぐ指すのはどの点か。
   2. すべての点 $(x,y) \neq (0,0)$ にわたるこの角の余弦の最小値と、それを達成する点を求めよ。（ヒント：$\rho = y/x$ とおき、ρ に関して最小化せよ。）$\kappa = 10$ のときの値を計算せよ。κ が大きくなるにつれて、このような損失に対する勾配降下法のステップに何が起きることをこの結果は示唆しているか。

3. **平坦な方向。**
   1. (x,y) の関数であって、原点に臨界点を持ち、そこでのヘッセ行列が $\mathrm{diag}(2,0)$ となるものを 3 つ挙げよ。ただし、1 つは狭義の極小点を持つもの、1 つは狭義でない極小点を持つもの、もう 1 つは極小点を持たないものとせよ。
   2. 関数 $f(x,y) = (x - y^2)(x - 2y^2)$ も原点でヘッセ行列 $\mathrm{diag}(2,0)$ を持つ。原点を通る任意の直線への制限が原点で狭義の極小点を持つことを示せ。平面上で $f < 0$ となる領域を図示し、それにもかかわらず原点は f の極小点ではないことを示せ。少数の方向に沿って損失を調べることで臨界点を判定する手法に対し、この事実は何を示唆するか。
   3. **猿の鞍** $f(x,y) = x^3 - 3xy^2$ は原点に臨界点を持つ。原点におけるヘッセ行列を計算し、f を極座標で表して、f が減少する方向をすべて求めよ。原点からの距離に対して $\|\nabla f\|$ は、鞍点 $x^2 - y^2$ と比べてどのようにスケールするか。これら 2 種類の鞍点の近くから出発した勾配降下法に対して、この違いは何を意味するか。

4. [code] **ランダムなヘッセ行列。**●極小点であるためにはヘッセ行列の n 個すべての固有値が非負でなければならず、非退化な極小点であるためにはすべて正でなければならない。本節のコイン投げの経験則では後者の確率は 2^{-n} となる。この経験則をランダム対称行列

$\mathbf{A} = (\mathbf{G} + \mathbf{G}^\top)/\sqrt{2}$ で検証せよ。ただし $n \times n$ 行列 $\mathbf{G}$ の各成分は独立な標準正規分布に従うとする。

1. $n = 2$ のとき、$s = (a_{11} + a_{22})/2$、$d = (a_{11} - a_{22})/2$、および a_{12} が独立な標準正規確率変数であること、そして $\mathbf{A} \succ 0$ となる必要十分条件が $s > \sqrt{d^2 + a_{12}^2}$ であることを示せ。3 次元の標準正規分布の回転対称性を用いて $P(\mathbf{A} \succ 0)$ を厳密に求め、2^{-2} と比較せよ。
2. モンテカルロ法により $P(\mathbf{A} \succ 0)$ を $n = 1, \dots, 6$ について推定せよ。ただしサンプル数は 10^6（$n \le 5$ のとき）およびバッチごとに生成した 10^7（$n = 6$ のとき）とせよ。$\log P$ を n に対して $\log 2^{-n}$ とともにプロットし、推定値に n の二次関数を当てはめよ。Dean and Majumdar (2006) は $\log P \approx -\frac{\ln 3}{4} n^2$ が大きな n で成り立つことを証明している。この係数と当てはめた二次の係数を比較せよ。コイン投げの経験則は極小点の確率を過大評価しているか過小評価しているか。また、固有値のどのような性質を無視しているか。
3. 大きな n に対して $\mathbf{A}$ の最小固有値は $-2\sqrt{n}$ の近くに位置する。$n = 5, 10, 20$ について、シフト β のうち行列 $\mathbf{A} + \beta \mathbf{I}$ のちょうど半分が正定値となるものを求め、$2\sqrt{n}$ と比較せよ。Bray and Dean (2007) のランダムなランドスケープでは、臨界点におけるヘッセ行列はそのようなシフトされた行列のように振る舞い、臨界点でのランダム関数の値が小さくなるほどシフトが大きくなる。この結果は、損失の小さい臨界点について何を示唆しているか。

5. [extended] **線形ネットワークの鞍点。●●** $\mathbf{A}$ を $n \times n$ 対称行列とし、固有値 $\lambda_1 > \lambda_2 > \cdots > \lambda_n > 0$ と固有ベクトルからなる正規直交基底 $\mathbf{q}_1, \dots, \mathbf{q}_n$ を持つとする。隠れユニットを 1 つ持ち、入力重み $\mathbf{v}$ と出力重み $\mathbf{u}$ を備えた線形ネットワークは、$\mathbf{x}$ を $\mathbf{u}\mathbf{v}^\top \mathbf{x}$ へと写す。白色化された入力（$\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{I}$）とターゲット $\mathbf{A}\mathbf{x}$ で訓練するとき、期待二乗誤差の半分は次式で与えられる。

$$\ell(\mathbf{u}, \mathbf{v}) = \tfrac{1}{2} \| \mathbf{A} - \mathbf{u}\mathbf{v}^\top \|_F^2 .$$

Baldi and Hornik (1989) は、このような線形ネットワークの極小点はすべて大域的最小点であることを示した。本問では臨界点の集合を完全に導出し、Dauphin et al. (2014) が非線形ネットワークについて計測した損失と指数の関係と比較する。なお、臨界点の**指数**とは、そのヘッセ行列の負の固有値の割合である。

1. $\nabla_{\mathbf{u}} \ell$ と $\nabla_{\mathbf{v}} \ell$ を計算せよ。臨界点が原点および $\mathbf{u} = \alpha \mathbf{q}_i$、$\mathbf{v} = (\lambda_i/\alpha)\mathbf{q}_i$（$\alpha \neq 0$）の形をした点であることを示し、それぞれの点における ℓ の値を計算せよ。
2. $2n \times 2n$ ヘッセ行列の固有値を、釣合いのとれた点 $\mathbf{u} = \mathbf{v} = \sqrt{\lambda_i}\,\mathbf{q}_i$ および原点において計算せよ。（固有ベクトル基底で展開すると、ヘッセ行列は 2×2 ブロックへと分解される。各ブロックは $\mathbf{q}_j$ に沿った $\mathbf{u}$ と $\mathbf{v}$ の座標対に対応する。）各臨界点で負の固有値は何個あるか。ゼロ固有値が生じる理由を ℓ の対称性から説明し、すべての極小点が確かに大域的最小点であると結論せよ。
3. $n = 4$、$(\lambda_1, \dots, \lambda_4) = (4, 3, 2, 1)$ とし、ランダムな正規直交固有ベクトル基底をとれ。スケールの異なる多数のランダムな初期点からニュートン法を実行し、到達したすべての臨界点について損失と指数を記録せよ。原点以外のすべての臨界点でヘッセ行列は特異になるため、各ニュートンステップは最小二乗法で計算せよ。Dauphin et al. (2014) の図 1 と同様に、損失を指数に対してプロットせよ。ニュートン法はどの種類の臨界点にどれくらいの頻度で到達するか。また、小さなランダム初期化から出発した勾配降下法はどこへ収束するか。
4. $\mathbf{u}\mathbf{v}^\top$ を隠れユニット 2 個の $\mathbf{U}\mathbf{V}^\top$ に置き換えよ（$\mathbf{U}$ と $\mathbf{V}$ は 4×2 行列となる）。ランク 1 の解析結果から臨界点での損失値を予測し、ニュートン法で確認せよ。臨界点に

おいてヘッセ行列はゼロ固有値を何個持つか。また、どの対称性がそれを説明するか。指数は損失の単調関数になっているか。

6. **ラグランジュ乗数とレイリー商。** $\mathbf{A}$ を $n \times n$ 対称行列とする。$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ を制約 $g(\mathbf{x}) = \|\mathbf{x}\|^2 = 1$ のもとで最大化せよ。
   1. ラグランジュ条件を書き下し、それが $\mathbf{A}\mathbf{x} = \mu\mathbf{x}$ と表せること、すなわち制約付き臨界点はすべて単位固有ベクトルであり、その乗数が固有値に一致することを示せ。
   2. すべての制約付き臨界点で $f(\mathbf{x}) = \mu$ となることを示せ。なぜ制約付き最大値が存在するのか。以上から最大値は $\lambda_{\max}(\mathbf{A})$、最小値は $\lambda_{\min}(\mathbf{A})$ であると結論せよ。
   3. $\mathbf{A} = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ に対して最大点・最小点とそれぞれの値を求め、ある最大点においてそこを通る f の等位集合が単位円に接していることを確かめよ。
   4. $g(\mathbf{x}) = \|\mathbf{x}\|^2$ は単位球面上のすべての点でラグランジュ条件の命題のどの仮定を満たしているか。次に、線形関数 $\mathbf{c}^\top \mathbf{x}$（$\mathbf{c} \neq \mathbf{0}$）を制約 $\|\mathbf{x}\|^2 = 0$ のもとで最小化せよ。最小点は何か。また、その点でいかなる乗数 μ も $\nabla f = \mu \nabla g$ を満たさないのはなぜか。

34.3 計算グラフ上の連鎖律

本書におけるすべての訓練ステップは、`backward()` または `grad` という 1 回の呼び出しから始まる。この呼び出しは、ネットワーク内のあらゆる重みに関する損失の傾きを返す。34.2 節では、それらの傾きが何を意味するかを見た。すなわち、それらをまとめたものが勾配（gradient）であり、パラメータを同時に微小変化させたときに損失がどう変化するかを予測するベクトルとなる。まだ説明していないのは、ネットワークを数回評価するのとほぼ同じ時間で、プログラムがどのようにして数百万個ものこうした傾きを計算しているのかという点である。重みを 1 つずつ微小に動かし、摂動のたびにネットワークを再実行する素朴な手法では、重みごとに別々のパスが必要となり、たちまち実行不可能になる。

連鎖律（chain rule）を使えばはるかに効率よく計算でき、その理由は数式を書き下す前にも容易に見て取れる。ニューラルネットワークは単純な関数を深く合成したものであるため、多数の重みに関する微分は計算の大部分を共有している。どの重みの影響も、後続の層を通過して損失へと到達しなければならないからである。入れ子になった 1 つの数式ではこの共有構造が隠れてしまうのに対し、中間値ごとに 1 つのノード、直接の依存関係ごとに 1 本の有向エッジを置く**計算グラフ**（computational graph）はその構造を明瞭に描き出す。グラフ上で見れば、連鎖律は経路にわたる和となり、共有された計算を活用することはその和をどうまとめるかを選ぶことに帰着する。出力から逆向きに経路をまとめれば、1 回のパスですべての入力に関する微分が得られる。このまとめ方こそが誤差逆伝播法（backpropagation）である。

すべてのステップを具体的で手計算でも確かめやすいものにするため、本節を通して整数値をとるスカラー変数からなる小さなグラフを扱う。ここでの議論全体は、34.2 節で学んだ 1 つの事実に基づいている。すなわち、関数が微分可能な任意の点の近傍において、関数の変化量は変数の変化ステップと勾配のドット積、つまり 34.2.1.2 節の線形モデルで近似されるということである。この考え方をベクトルからベクトルへ写す層へと拡張し、実際のネットワークにおける計算コストを測る議論は、続く 2 つの節で扱う。

34.3.1 経路にわたる和としての連鎖律

34.3.1.1 スカラーのグラフ

すべて手計算で追えるほど小さな具体例から始めよう。4 つのスカラー入力 w, x, y, z が中間変数の 2 つの層を経て、1 つのスカラー出力を生成する。

$$
\begin{aligned}
f(u,v) &= (u+v)^2, \\
u(a,b) &= (a+b)^2, \qquad v(a,b) = (a-b)^2, \\
a(w,x,y,z) &= (w+x+y+z)^2, \qquad b(w,x,y,z) = (w+x-y-z)^2.
\end{aligned}
\tag{34.27}
$$

これらの式は、Figure 34.17 に示す有向非巡回グラフ（directed acyclic graph）を定めており、各ノードが中間値を保持し、各矢印が直接の関数依存関係を表す。二乗演算自体に特別な意味はなく、中間値と導関数の値がすべて厳密な整数に保たれるという理由だけで選んでいる。本節を通して、対象とするグラフのスカラー出力を f で表す。

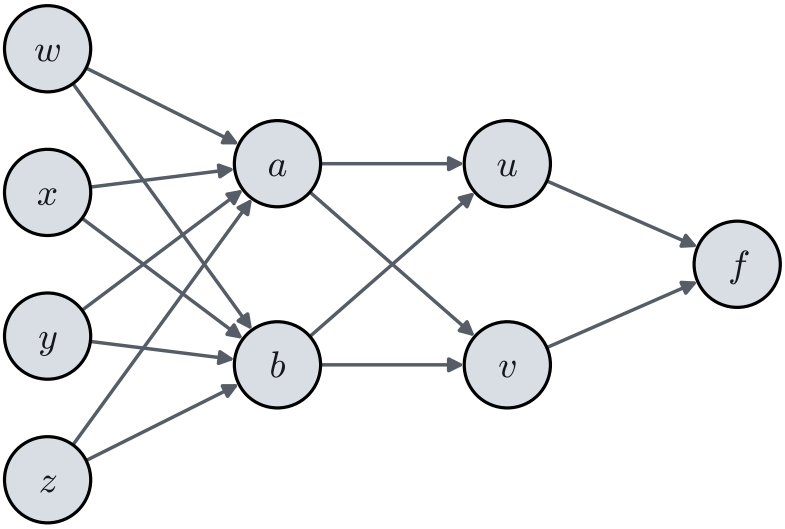


Figure 34.17: Equation 34.27 の依存関係グラフ。各ノードは変数を表し、各有向エッジは直接の関数依存関係を表す。

34.3.1.2 1 ステップを通る 2 つの経路

グラフ全体を扱う前に、最後の合成ステップ $f(u(a,b), v(a,b))$ だけを取り上げよう。f がどう応答するかを見るため、a を微小量 ϵ だけ動かす。2 つの中間ノードはそれぞれの偏微分に従ってこの摂動を受け、u は $u + \epsilon \frac{\partial u}{\partial a}$ へ、v は $v + \epsilon \frac{\partial v}{\partial a}$ へと移動する。これら両方の変化を、34.2.1.2 節で導入した f の一次線形近似に代入すると、次の式が得られる。

$$
f\big(u + \epsilon \tfrac{\partial u}{\partial a},\, v + \epsilon \tfrac{\partial v}{\partial a}\big) \approx f(u,v) + \epsilon \left[\frac{\partial f}{\partial u}\frac{\partial u}{\partial a} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial a} \right].
$$

ϵ の係数を読み取れば、a に関する合成関数の導関数、すなわち多変数の連鎖律が得られる。

$$
\frac{\mathrm{d}f}{\mathrm{d}a} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial a} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial a}.
$$

微分を表す 2 つの異なる記号に注目しよう。ここから先は、この 2 つを厳密に区別する。丸みを帯びた ∂ は、単一ノードを定義する関数について、他の引数を固定したまま直接の引数の 1 つに関してとった偏微分を表す。立体活字の d は**全**微分（total derivative）である $\mathrm{d}f/\mathrm{d}a$ を表し、a の下流にあるすべての変数が応答することを許す。（入力ノードについては、全微分は f を入力のみの関数とみなしたときの通常の偏微分と一致する。）幾何学的には、a が f に影響を及ぼす有向**経路**（path）は $a \to u \to f$ と $a \to v \to f$ の 2 つある。各経路はそのエッジに沿った微分の積として寄与し、全微分はすべての経路にわたってそれらの寄与を足し合わせる。

この導出は 1 変数の連鎖律の証明（34.1.2 節）と対応している。内側の関数の線形変化を、外側の関数の線形モデルに流し込んでいるからである。ただし、ここでは 2 つの変数が同時に動くため、f は (u, v) において単に偏微分を持つだけでなく、**微分可能**でなければならない。反例 $2xy/(x^2 + y^2)$（34.2.1.2 節）を振り返ろう。対角線 $x = y = t$ に沿って見ると、原点では両方の偏微分がゼロになるため、連鎖律は微分値 0 を $t = 0$ において予測するが、実際には関数はそこで 0 から 1 へと不連続に跳ぶ。

34.3.1.3 エッジ微分と全微分

同じ推論は、数学的帰納法によって任意の有向非巡回グラフへと拡張できる。あるノードの親の 1 つに関する**エッジ微分**（edge derivative）を、他のすべての親を固定したときの、その親に関するノードの関数の偏微分として定義しよう。このとき、出力 f の入力 x に関する全微分は次の式で与えられる。

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \sum_{\text{directed paths } x \to \cdots \to f} \; \prod_{\text{edges on the path}} (\text{edge derivative}). \tag{34.28}$$

言葉で言えば、x から f への各有向経路はそのエッジに沿った局所的な変化率を掛け合わせ、全微分はそれらの経路ごとの積を足し合わせたものになる。

スキップ接続（skip connection）を持つグラフを見ると、エッジ微分と全微分を決して混同してはならない理由がよくわかる。次の 5 ノードの系を考えよう。

$$\begin{aligned} p(x, y) &= x + y, \qquad q(x, y) = x - y, \\ r(p) &= p^2, \qquad s(q) = q^2, \\ f(r, p, s) &= r + p + s. \end{aligned} \tag{34.29}$$

ここでは、f は r と s に依存するだけでなく、スキップエッジを通じて p にも直接依存している（Figure 34.18）。その結果、入力 y は f に 3 つの異なる経路、すなわち $y \to p \to r \to f$、ショートカットである $y \to p \to f$、そして $y \to q \to s \to f$ に沿って到達する。局所的なエッジ微分は $\partial f/\partial r = \partial f/\partial p = \partial f/\partial s = 1$、$\partial r/\partial p = 2p$、$\partial s/\partial q = 2q$、$\partial p/\partial y = 1$、$\partial q/\partial y = -1$ である。3 つの経路の積を足し合わせると、次のようになる。

$$\begin{aligned} \frac{\mathrm{d}f}{\mathrm{d}y} &= \underbrace{1 \cdot 2p \cdot 1}_{y \to p \to r \to f} + \underbrace{1 \cdot 1}_{y \to p \to f} + \underbrace{1 \cdot 2q \cdot (-1)}_{y \to q \to s \to f} \\ &= 2p + 1 - 2q = 1 + 4y. \end{aligned}$$

すべてを f に代入すれば直接確認できる。すなわち $f = (x+y)^2 + (x+y) + (x-y)^2$ であり、その y に関する導関数は確かに $1 + 4y$ となる。$(x, y) = (1, 2)$ では $p = 3$、$q = -1$ であるから、3 つの経路の寄与は $6 + 1 + 2 = 9 = 1 + 4 \cdot 2$ となる。ノード p における明確な違いに注目

しよう。**エッジ**微分 $\partial f/\partial p$ は単なる 1（r を固定した場合）であるのに対し、**全**微分 $\mathrm{d}f/\mathrm{d}p$ はスキップエッジと r を経由する間接経路の両方を考慮するため、$2p + 1 = 7$ となる。この 2 つの量の混同は、多変数の連鎖律を適用するときに最もよく起こる誤りである。

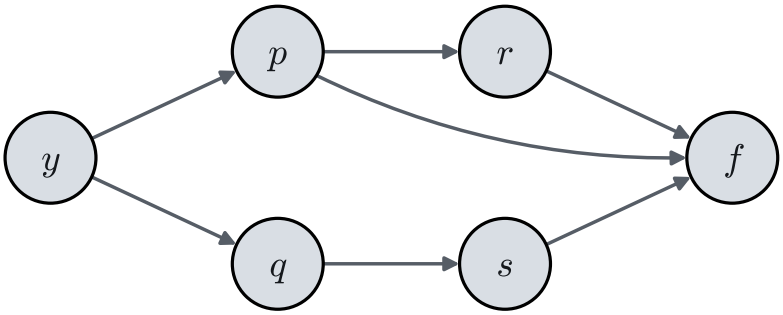


Figure 34.18: Equation 34.29 の依存関係グラフ。入力 y は f に 3 つの経路、すなわち $y \to p \to r \to f$、スキップエッジ $y \to p \to f$、および $y \to q \to s \to f$ によって到達する。

この経路和の視点は、深いアーキテクチャの内部を勾配がどのように伝わるかも説明してくれる。LSTM（12.1 節）のゲートや、7.4 節の残差ネットワーク（residual network）におけるスキップ接続のように経路を開閉する仕組みは、どの経路が初期の層へ勾配信号を運び戻すかを制御することで、学習を直接方向づける。

34.3.2 誤差逆伝播法

数学的な恒等式としては、経路にわたる和 Equation 34.28 は完全に一般的であり、あらゆる計算グラフのあらゆる導関数や微分値を与える。しかし、これを文字通りのアルゴリズムとして実行するのは絶望的である。深さとともに経路数が指数関数的に増大し得るからである。たとえば m 段の単純な連鎖であっても、各段に 2 個の並列ノードがあれば、始点から終点までにすでに 2^m 本の経路が含まれる。幸い、それらの経路は大きく重なり合っており、多数の経路が共有する部分積は 1 回だけ計算すればよい。こうした共有される部分積をどの順序でくくり出すかが、計算コストを決定する。

34.3.2.1 フォワードモード

4 入力の関数 Equation 34.27 を点 $(w, x, y, z) = (1, 1, -2, 1)$ で考え、和をくくり出す 2 つの方法を比較してみよう。**順伝播**（forward pass）では、トポロジカル順序に従って各ノードを評価し、$a = 1^2 = 1$、$b = 3^2 = 9$、$u = 10^2 = 100$、$v = (-8)^2 = 64$、そして $f = 164^2 = 26896$ を得る。これらのノード値から、局所的なエッジ微分がすべて次のように読み取れる。

$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial v} = 2(u+v) = 328,$$
$$\frac{\partial u}{\partial a} = \frac{\partial u}{\partial b} = 2(a+b) = 20,$$
$$\frac{\partial v}{\partial a} = 2(a-b) = -16, \qquad \frac{\partial v}{\partial b} = -2(a-b) = 16,$$
$$\frac{\partial a}{\partial w} = \frac{\partial a}{\partial x} = \frac{\partial a}{\partial y} = \frac{\partial a}{\partial z} = 2(w+x+y+z) = 2,$$
$$\frac{\partial b}{\partial w} = \frac{\partial b}{\partial x} = 2(w+x-y-z) = 6, \qquad \frac{\partial b}{\partial y} = \frac{\partial b}{\partial z} = -6.$$

各エッジ微分は、ノードとその直接の親だけから計算される局所的な数値である。

$\mathrm{d}f/\mathrm{d}w$ を求めるには、w から f への 4 つの経路がすべて、まず a または b を通り、次に u または v を通ることに注目すればよい。入力 w からグラフに沿って順方向に経路をくくり出すと、次のようになる。

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}w} &= \frac{\partial f}{\partial u}\left(\frac{\partial u}{\partial a}\frac{\partial a}{\partial w} + \frac{\partial u}{\partial b}\frac{\partial b}{\partial w}\right) + \frac{\partial f}{\partial v}\left(\frac{\partial v}{\partial a}\frac{\partial a}{\partial w} + \frac{\partial v}{\partial b}\frac{\partial b}{\partial w}\right) \\ &= 328\,(20\cdot 2 + 20\cdot 6) + 328\,(-16\cdot 2 + 16\cdot 6) \\ &= 328\cdot 224 = 73472.\end{aligned}$$

括弧内の 2 つの和は、それぞれ $\mathrm{d}u/\mathrm{d}w = 160$ と $\mathrm{d}v/\mathrm{d}w = 64$、すなわち w に関する中間ノードの全微分にほかならない。このように 1 つの入力に対する感度をグラフに沿って順方向に伝播させる手法を**フォワードモード**と呼ぶ。この計算を素の Python で確かめてみよう。

```
w, x, y, z = 1, 1, -2, 1
# Forward pass: node values
a, b = (w + x + y + z)**2, (w + x - y - z)**2
u, v = (a + b)**2, (a - b)**2
f = (u + v)**2
print('a, b, u, v, f =', a, b, u, v, f)

# Edge derivatives
df_du, df_dv = 2*(u + v), 2*(u + v)
du_da, du_db, dv_da, dv_db = 2*(a + b), 2*(a + b), 2*(a - b), -2*(a - b)
da_dw, db_dw = 2*(w + x + y + z), 2*(w + x - y - z)

# Forward mode: push the dependence on w through the graph
du_dw, dv_dw = du_da*da_dw + du_db*db_dw, dv_da*da_dw + dv_db*db_dw
df_dw = df_du*du_dw + df_dv*dv_dw
print('df/dw =', df_dw)
```

```
a, b, u, v, f = 1 9 100 64 26896
df/dw = 73472
```

フォワードモードは $\mathrm{d}f/\mathrm{d}w = 73472$ を返すが、その中間微分は x、y、z のいずれに対しても再利用できない。$\mathrm{d}u/\mathrm{d}w$ も $\mathrm{d}v/\mathrm{d}w$ も、w だけに結びついているからである。ニューラルネットワ

ークの訓練では単一のスカラー損失をすべてのパラメータに関して微分する必要があるため、入力を 1 つずつ計算していたのでは共有構造の大部分が活かされないままになる。

34.3.2.2 随伴変数と逆伝播

解決策は視点を逆転させ、入力ではなく出力を固定することである。フォワードモードでは 1 つの入力を固定し、下流のすべてのノードがそれにどう応答するかを問うた。今度は単一の出力 f を固定し、上流のすべてのノードに対してそれがどう応答するかを問う。**随伴変数**（adjoint）$\bar{n}$ をノード n に対して定め、出力の全微分 $\mathrm{d}f/\mathrm{d}n$（すなわち n に関する全微分）を、n から f へのすべての有向経路にわたるエッジ微分の積の和とする。そのような経路はすべて、n からその直接の子 c のいずれかを通って出る必要があり、c から f への残りの経路の和は $\bar{c}$ になる。経路和 Equation 34.28 をこの最初のエッジごとにまとめると、単純な漸化式が得られる。

$$\bar{f} = 1, \qquad \bar{n} = \sum_{\text{children } c \text{ of } n} \bar{c}\,\frac{\partial c}{\partial n}. \tag{34.30}$$

右辺に現れるのは、局所的なエッジ微分と下流ノードの随伴変数だけである。$\bar{f} = 1$ から出発して逆トポロジカル順序でノードをたどれば、必要になる時点ですべての子ノードの随伴変数がすでに求まっている。たとえばスキップエッジを持つグラフ Equation 34.29 では、この漸化式から $\bar{r} = 1$ および $\bar{p} = \bar{r}\cdot 2p + \bar{f}\cdot 1 = 7$ が得られ、先に求めた全微分と一致する。同じスイープを 4 入力のグラフに適用すると、次のようになる。

- $\bar{u} = \bar{v} = 328$;
- $\bar{a} = \bar{u}\cdot 20 + \bar{v}\cdot(-16) = 1312$、$\bar{b} = \bar{u}\cdot 20 + \bar{v}\cdot 16 = 11808$
- $\bar{w} = \bar{x} = \bar{a}\cdot 2 + \bar{b}\cdot 6 = 73472$、$\bar{y} = \bar{z} = \bar{a}\cdot 2 - \bar{b}\cdot 6 = -68224$

このように、1 回の**逆伝播**（backward pass）によって、4 つの入力すべてに関する偏微分が一度に得られる（Figure 34.19）。前のセルで計算したエッジ微分を再利用して、この逆方向スイープを Python で実行してみよう。

```
da_dy, db_dy = 2*(w + x + y + z), -2*(w + x - y - z)  # also for x, z
# Backward pass: adjoints from the output toward the inputs
u_bar, v_bar = df_du, df_dv
a_bar = u_bar*du_da + v_bar*dv_da
b_bar = u_bar*du_db + v_bar*dv_db
print(f'adjoints: u {u_bar}, v {v_bar}, a {a_bar}, b {b_bar}')
w_bar = a_bar*da_dw + b_bar*db_dw  # x enters a, b like w
y_bar = a_bar*da_dy + b_bar*db_dy  # z enters a, b like y
print('gradient:', {'w': w_bar, 'x': w_bar, 'y': y_bar, 'z': y_bar})
```

```
adjoints: u 328, v 328, a 1312, b 11808
gradient: {'w': 73472, 'x': 73472, 'y': -68224, 'z': -68224}
```

w と x に関する微分が一致し、同様に y と z に関する微分も一致するのは、f が $w+x$ と $y+z$ という組み合わせを通じてのみ入力に依存しているからである。

演算回数を簡単に数えてみると、その利点が具体的になる。フォワードモードでは入力 1 つあたり 6 回の乗算（$\mathrm{d}u/\mathrm{d}w$、$\mathrm{d}v/\mathrm{d}w$、$\mathrm{d}f/\mathrm{d}w$ にそれぞれ 2 回ずつ）が必要であり、4 つの入力すべてでは合計 24 回となる。一方、逆伝播はグラフの共有された上側部分を 1 回しかたどらないため、$\bar{a}$ と $\bar{b}$ の計算に 4 回、その後は入力 1 つあたり 2 回、合計 12 回の乗算で済む。トイグラフでの 2 倍という差は控えめに見えるかもしれないが、入力数が増えるにつれてその差は急速に広が

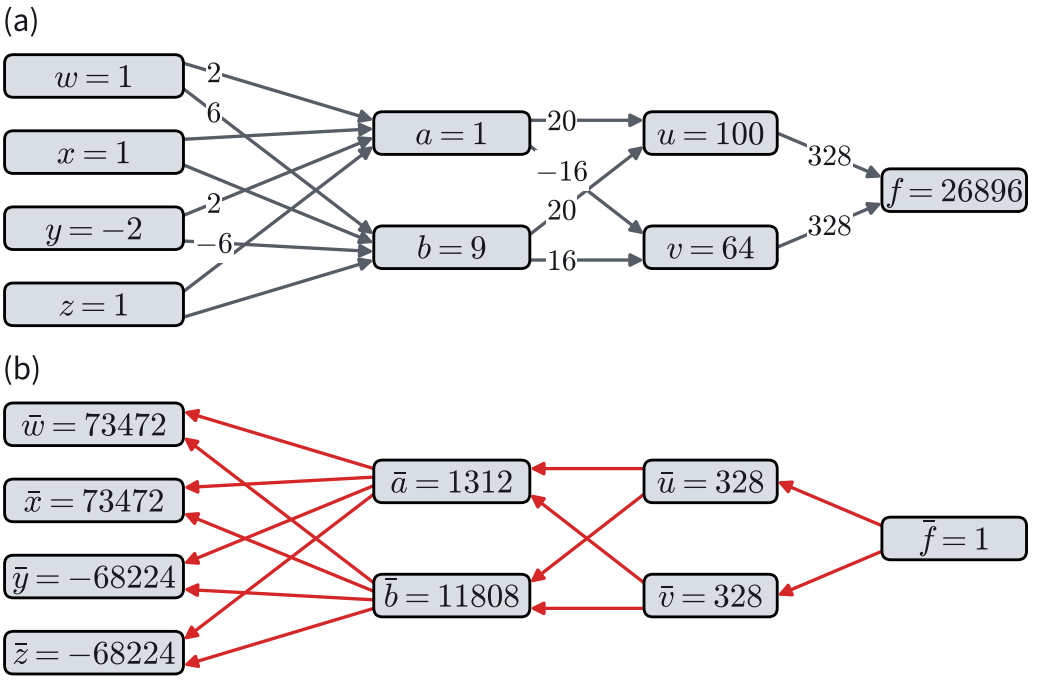


Figure 34.19: (a) 順伝播では、すべてのノード値と局所的なエッジ微分を記録する。(b) 逆伝播では、$\bar{f} = 1$ から始めて随伴変数を累積し、$\bar{a} = 328 \cdot 20 + 328 \cdot (-16) = 1312$ および $\bar{w} = 1312 \cdot 2 + 11808 \cdot 6 = 73472$ を得る。

る。入力が 1 つ増えるたびに、フォワードモードではグラフ全体のパスがまるごと 1 回余分にかかるのに対し、逆伝播では乗算がわずか 2 回増えるだけである。

34.3.2.3 フレームワークにおける誤差逆伝播法

出力から入力へと随伴変数を逆向きに伝播させて微分を計算する手法が**誤差逆伝播法**であり (Rumelhart et al. 1986)、4.3 節ではこれを多層パーセプトロン（multilayer perceptron）に適用している。現代のあらゆる自動微分（automatic differentiation）エンジンは、これと同じ 2 つのパスを実行する。すなわち、順伝播の間に中間値（またはエッジ微分を構成するのに必要な演算）を記録し、続いて逆伝播の間に随伴変数を累積するのである。同じ 4 つの入力に対して、手計算の結果をフレームワークの自動微分と照らし合わせてみよう。

```
wxyz = torch.tensor([1., 1., -2., 1.], requires_grad=True)
w, x, y, z = wxyz
a, b = (w + x + y + z)**2, (w + x - y - z)**2
u, v = (a + b)**2, (a - b)**2
f = (u + v)**2
f.backward()  # one backward pass
print(wxyz.grad)
```

```
tensor([ 73472.,  73472., -68224., -68224.])
```

フレームワークは $(73472, 73472, -68224, -68224)$ を返し、手計算で求めた随伴変数と一致する。この一致が近似ではなく厳密なのは、すべての中間値が 2^{24} 未満の整数であり、単精度（single precision）で丸め誤差（rounding error）なしに表現できるからである。

ここまでのグラフではすべてのノードがスカラーであったため、乗算の回数も簡単な手計算で数えることができた。実際のニューラルネットワークでは各層がベクトルをベクトルへ写すため、エッジ微分は行列となり、それらの行列を掛ける順序はスカラー乗算の順序よりもはるかに重要になる。そのステップには 34.4 節で進む。

34.3.3 まとめ

計算グラフにおいて、入力に関する出力の全微分は、両者を結ぶすべての有向経路にわたって局所的なエッジ微分の積を足し合わせたものに等しい。ノードの随伴変数、すなわちそのノードに関する出力の全微分は、そこから出るすべての経路を集約したものである。それらの経路を最初に出るエッジごとにまとめると漸化式が得られ、これを出力から逆向きに実行すれば、1 回のパスですべての随伴変数を計算できる。この逆方向の累積が誤差逆伝播法である。これに対し、1 つの入力を順方向に伝播させるとその入力に関する微分しか求まらないため、スカラー損失を多数のパラメータに関して微分する計算は常に逆向きに行われる。

34.3.4 演習問題

1. [code] **2 つの tanh ユニットに対する逆伝播**。1 つの入力 x、2 つの隠れユニット、1 つの出力を持つネットワークが、ターゲット t に対して次を計算するとする。

$$h_i = \tanh(w_i x + b_i) \quad (i = 1, 2), \qquad f = v_1 h_1 + v_2 h_2, \qquad \ell = \tfrac{1}{2}(f - t)^2.$$

   1. ℓ の依存関係グラフを描け。素の Python で、すべての中間値を保存する順伝播と、ℓ の 6 個すべてのパラメータ $w_1, b_1, w_2, b_2, v_1, v_2$ に関する偏微分を返す逆伝播を実装

せよ。$x = 1.5$、$t = 0.5$、$(w_1, b_1, w_2, b_2, v_1, v_2) = (0.5, 0, -1, 0.2, 1, -0.5)$ において、それらの値を自動微分と照合せよ。6 個の偏微分すべてが再利用する中間微分はどれか。

2. 他の値はそのままにして、$w_1 = 5$、$x = 2$ とせよ。素の Python による逆伝播で $\partial \ell / \partial w_1$ を計算し、その経路に沿ったエッジ微分の積から値の大きさを説明せよ。単精度の自動微分が返す値と比較し、その違いを説明せよ。このことは、活性化前の値が大きいユニットの訓練に対して何を示唆するか。
3. $w_1 = w_2$、$b_1 = b_2$、$v_1 = v_2$ とせよ。逆伝播から両方のユニットが同一の勾配を受け取り、その結果、勾配降下法が毎ステップで両者を同一のまま保つことを示せ。このことは、隠れユニットの初期化に対して何を示唆するか。

2. **残差スタックを通る経路。**●7.4 節の残差ネットワークの 1 次元モデルにおいて、スカラーブロックは $x_{l+1} = x_l + g_l(x_l)$（$l = 0, \dots, m-1$、ただし g_l は微分可能な関数）を計算する。

1. 連鎖律を用いて、$\partial x_m / \partial x_0$ を m 個の因子の積として表せ。
2. その積を展開し、各項を依存関係グラフにおける x_0 から x_m への有向経路に対応させよ。経路は全部で何本あり、そのうちちょうど k 個の関数 g_l を通るものは何本あるか。経路の数と、逆伝播が $\partial x_m / \partial x_0$ を計算するのに要する乗算の回数を比較せよ。
3. 訓練済みの $m = 54$ 個の残差ブロックからなるネットワークにおいて、Veit et al. (2016) は、勾配のほぼすべてが 5 から 17 個のブロックを通る経路に沿って流れることを見いだした。これらの経路は全経路のどれだけの割合を占めるか。次に、すべてのブロックで $g_l'(x_l) = c > 0$ であると仮定する。どのような c の値のときに、約 11 個のブロックを通る経路が $\partial x_m / \partial x_0$ に最も大きく寄与するか。また、そのとき 5 から 17 個のブロックを通る経路は微分全体のどれだけの割合を担うか。

3. [code] **単精度における勾配チェック。**●勾配チェックは、逆伝播を中心差分 $\big(f(w + h, x, y, z) - f(w - h, x, y, z)\big)/(2h)$ と比較する手法である。f を Equation 34.27 の関数とし、$(w, x, y, z) = (1, 1, -2, 1)$ における厳密な偏微分は $\partial f / \partial w = 73472$ であるとする。

1. NumPy で f を実装し、この点における中心差分を $h = 10^{-1}, 10^{-2}, \dots, 10^{-12}$ に対して `float64` と `float32` のそれぞれで計算せよ。両方の精度について相対誤差を h に対して両対数軸でプロットし、それぞれの最適な h と最小の相対誤差を読み取れ。
2. 中心差分の誤差は、オーダー h^2 の打ち切り誤差の項とオーダー $\varepsilon_{\text{mach}} |f| / h$ の丸め誤差の項の和となる。ここで計算機イプシロン $\varepsilon_{\text{mach}}$ は、約 2.2×10^{-16}（`float64`）および約 1.2×10^{-7}（`float32`）である。最適な h と最小の相対誤差は、`float64` から `float32` に変えたとき何倍変化するはずか。予測される倍率をプロットと比較せよ。
3. `float32` での勾配チェックにおいて、正しい逆伝播を誤って棄却しないためには相対誤差の許容誤差をどの程度に設定できるか。勾配チェックが倍精度で実行されるのはなぜか。
4. `float32` の推定値が最小領域の h に対して厳密にゼロになるのはなぜか。

34.4 行列微分

直前の 2 節では、損失をスカラー変数の関数として扱った。34.2 節では偏微分を並べた列として勾配を構成し、34.3 節ではノードごとに 1 つの数値を持つグラフ上の逆伝播によってその列を計算した。しかし、ニューラルネットワークはスカラーでは記述されない。各層はベクトルをベクトルへ写し、その重みは行列をなす。ベクトルや行列に関する損失の導関数とは何であり、どうすれば層全体に対して一度に計算できるのだろうか。

手計算できるほど小さなネットワークを見れば、何が必要かがわかる。2 つの入力、2 つの ReLU 隠れユニット、1 つの出力を持つ次のネットワークを考えよう。

$$g(\mathbf{x}) = \mathbf{w}_2^\top \varphi(\mathbf{W}_1 \mathbf{x} + \mathbf{b}), \qquad \mathbf{W}_1 = \begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ -4 \end{pmatrix}, \quad \mathbf{w}_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix},$$

ここで、φ は各成分に $\max(0, \cdot)$ を適用する写像である。$\mathbf{x} = (1, 1)$ において活性化前の値は $\mathbf{W}_1 \mathbf{x} + \mathbf{b} = (3, -2)$ となるため、2 番目の隠れユニットは非活性となり、隠れベクトルは $(3, 0)$、出力は $g = 6$ となる。訓練ステップでは、この出力に関して 3 つの問いが生じる。第 1 に、入力が微小な δ だけ動くと、出力は $2\delta_1 + 4\delta_2$ だけ変化する。この係数は 2 つの数値からなる行であり、列である勾配の転置になっている。第 2 に、重み行列が $\Delta \mathbf{W}_1$ だけ動くと、出力は $2\,\Delta W_{11} + 2\,\Delta W_{12}$ だけ変化し、第 2 行からはまったく影響を受けない。したがって、行列に関する導関数はそれ自体が行列 $\left(\begin{smallmatrix} 2 & 2 \\ 0 & 0 \end{smallmatrix}\right)$ となる。第 3 に、隠れ層 $\mathbf{x} \mapsto \varphi(\mathbf{W}_1 \mathbf{x} + \mathbf{b})$ はベクトルをベクトルへ写す。その導関数はスカラーの傾きでも勾配ベクトルでも表せないが、出力はまさにこの層を介してのみ $\mathbf{W}_1$ に依存している。

このような導関数は、手計算では驚くほど間違えやすい。正方の重み行列で転置を落としても、次元は一致したままでエラーも発生せず、ただ訓練が狂ってしまう。行列微分の公式集には何百もの項目が並んでいるが、公式を引くだけではこの種のうっかりミスはほとんど防げない。

本節では、これら 3 つの問いすべてに、単一の数学的対象と単一の原理で答える。ベクトル値写像の導関数は、その写像の最良線形近似として定義される行列、すなわち**ヤコビ行列**（Jacobian）である。写像の合成はヤコビ行列の積に対応するため、ネットワークの導関数は層ごとに 1 つずつの行列の積となる。さらに、層が必要とするあらゆる勾配（線形形式や二次形式、最小二乗法、行列積、ソフトマックス）は、1 つの手法から導かれる。関数の一次の変化を書き下し、その係数として勾配を読み取るのである。同じ手法から、4.3 節で用いた各層の逆伝播だけでなく、$\mathbf{p} - \mathbf{y}$ という 3.1.2.2 節の分類器の勾配も得られる。

特に覚えておくべき結果は 3 つある。ヤコビ行列の積 Equation 34.36、勾配は常にその変数と同じ形状を持つという規則、そして行列積の逆伝播則 Equation 34.39 である。初読時には連鎖律の証明を読み飛ばしても構わないが、全微分法（differential method、34.4.3.1 節）は読み飛ばせない。以降の議論はすべてこれに依拠するからである。

本節全体を通して、以下のインポートを使用する。

```
import numpy as np
import torch
```

34.4.1 ヤコビ行列

34.4.1.1 最良線形近似

$\mathbf{x} = (1, 1)$ の近傍では、冒頭のネットワークの隠れ層はアフィンであるため、その線形近似は厳密に一致し、調べるべき誤差が残らない。そこでまずは、本節と次節で共通の例として再利用する非線形な写像から始め、定義が整ったところで再びネットワークに戻ることにしよう。

$$\mathbf{f}(x, y) = \begin{pmatrix} x + \sin y \\ y + x^2 / 2 \end{pmatrix}, \qquad \mathbf{x}_0 = (0.5,\ 0.5).$$

$\mathbf{x}_0$ において、この写像の値は $\mathbf{f}(\mathbf{x}_0) = (0.9794,\ 0.6250)$ である。2 つの出力はそれぞれ 2 つの偏微分を持ち、$\partial f_1 / \partial x = 1$、$\partial f_1 / \partial y = \cos y = 0.8776$、$\partial f_2 / \partial x = x = 0.5$、$\partial f_2 / \partial y = 1$

となる。これら 4 つの数値を、出力ごとに 1 行、入力ごとに 1 列を持つ行列に並べる。

$$\mathbf{J}(\mathbf{x}_0) = \begin{pmatrix} 1 & \cos 0.5 \\ 0.5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.8776 \\ 0.5 & 1 \end{pmatrix}.$$

第 j 列は、入力 j が動いたときに両方の出力がどう応答するかを記録している。この行列は $\mathbf{f}$ の $\mathbf{x}_0$ における導関数である。すなわち、$\mathbf{f}(\mathbf{x}_0 + \delta) \approx \mathbf{f}(\mathbf{x}_0) + \mathbf{J}\delta$ が微小な δ に対して成り立ち、他のどの行列もこれほど精密には $\mathbf{f}$ を近似しない。

1 変数の場合、導関数は最良の直線近似 $f(x+\delta) \approx f(x) + f'(x)\,\delta$ の傾きを与える。ベクトル値への一般化も同じ局所近似に基づく。写像 $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ が点 $\mathbf{x}$ で**微分可能**（differentiable）であるとは、ある行列 $\mathbf{J} \in \mathbb{R}^{m \times n}$ が存在して次式を満たすことをいう。

$$\mathbf{f}(\mathbf{x} + \delta) = \mathbf{f}(\mathbf{x}) + \mathbf{J}\,\delta + o(\|\delta\|), \tag{34.31}$$

これは、線形モデル $\mathbf{f}(\mathbf{x}) + \mathbf{J}\delta$ の誤差を $\|\delta\|$ で割ったものが、$\delta \to \mathbf{0}$ のとき 0 に収束することを意味する。このような行列が存在するとき、それは一意に定まる。これを**ヤコビ行列** $\mathbf{J}_{\mathbf{f}}(\mathbf{x})$ と呼び、その各成分は単なる偏微分となる。

$$[\mathbf{J}_{\mathbf{f}}]_{ij} = \frac{\partial f_i}{\partial x_j}. \tag{34.32}$$

その理由を見るには、摂動 $\delta = t\,\mathbf{e}_j$（座標軸 j に沿った微小ステップ）を Equation 34.31 に代入し、第 i 成分に注目すればよい。t で割って $t \to 0$ とすると、$\partial f_i / \partial x_j$ が (i,j) 成分として残り、これは出発点とした行列 $\mathbf{J}$ の選び方によらない。したがって、偏微分は $\mathbf{J}$ の各成分を与えるが、それを定義するのは線形近似である。

偏微分の存在だけでは、微分可能性は保証されない。$f(x,y) = 2xy/(x^2+y^2)$（ただし $f(0,0) = 0$）という 34.2.1.2 節の関数を考えよう。原点では両方の偏微分が 0 であるにもかかわらず、f は値 1 を対角線 $x = y$ 上のいたるところでとる。ヤコビ行列の唯一の候補はゼロ行ベクトルであるが、対角線上の近似誤差は 1 のままであり、δ をどれほど小さくしても変わらないため、f は原点でヤコビ行列を持たない。このような振る舞いを排除するには、偏微分の連続性が十分条件となる。すなわち、すべての偏微分 $\partial f_i / \partial x_j$ が $\mathbf{x}$ の近傍で存在し、かつ $\mathbf{x}$ で連続であれば、$\mathbf{f}$ は $\mathbf{x}$ で微分可能である (Rudin 1976)。

Figure 34.20 は、この定義を幾何学的に示している。$\mathbf{x}_0$ の近傍では、写像はそのヤコビ行列のように振る舞う。半径 r で $\mathbf{x}_0$ を中心とする円は、楕円 $\mathbf{f}(\mathbf{x}_0) + \mathbf{J}\delta$（$\|\delta\| = r$）へとほぼ写され、その半軸の長さは r と $\mathbf{J}$ の特異値の積になる（33.5 節で示したように、任意の行列は単位円を特異値を半軸とする楕円へ写す）。ここでは特異値が 1.706 と 0.329 であるため、半径 0.35 の円は半軸 0.60 と 0.12 の楕円になる。真の像と楕円との最大のずれは 0.061（$r = 0.35$ のとき）および 0.015（$r = 0.175$ のとき）であり、半径を半分にするとずれは 4 分の 1 に縮まる。

定義は線形モデルの誤差を $\|\delta\|$ で割った比が 0 に収束することを保証するが、その速さまでは述べていない。摂動を 10 分の 1 ずつ縮めながら、その比を調べてみよう。

```
def f(v):                                  # the running map, R^2 -> R^2
    x, y = v
    return np.array([x + np.sin(y), y + x**2 / 2])

def J(v):                                  # its Jacobian, entry by entry
```

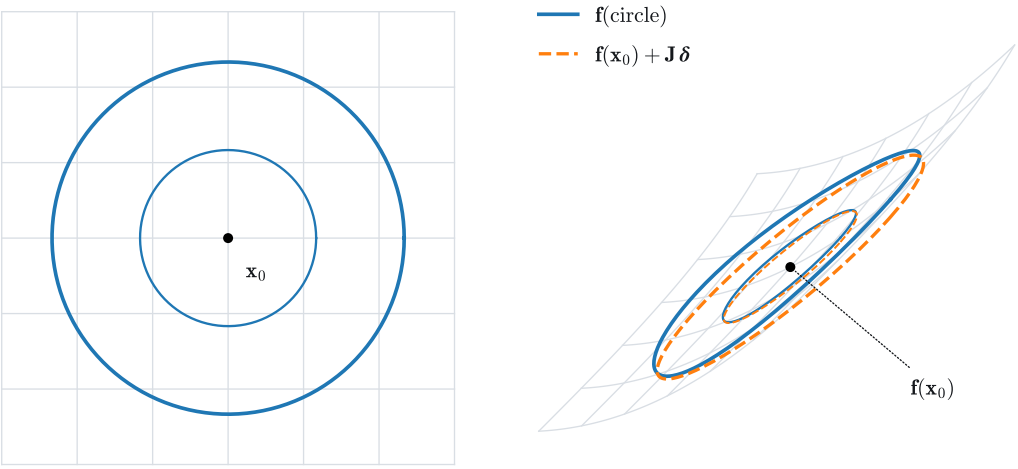


Figure 34.20: 微分可能な写像は局所的にそのヤコビ行列と一致する。半径 0.35 と 0.175 で $\mathbf{x}_0 = (0.5, 0.5)$ を中心とする 2 つの円（左）と、$\mathbf{f}(x, y) = (x + \sin y,\ y + x^2/2)$ によるそれらの像（右、青）および楕円 $\mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x}_0)\delta$（オレンジ、破線）の比較。各楕円の半軸は、半径に特異値 1.71、0.33（行列 $\mathbf{J}$ の特異値）を掛けた値である。半径を半分にすると、像と楕円の最大のずれは 0.061 から 0.015 へと縮まり、剰余項 $o(\|\delta\|)$ の振る舞いを示す。

```
    x, y = v
    return np.array([[1.0, np.cos(y)], [x, 1.0]])

x0 = np.array([0.5, 0.5])
for s in [1, 0.1, 0.01, 0.001]:
    delta = s * np.array([0.2, 0.1])
    err = np.linalg.norm(f(x0 + delta) - f(x0) - J(x0) @ delta)
    print(f'|delta| = {np.linalg.norm(delta):.1e}   '
          f'error / |delta| = {err / np.linalg.norm(delta):.1e}')
```

```
|delta| = 2.2e-01   error / |delta| = 9.0e-02
|delta| = 2.2e-02   error / |delta| = 9.0e-03
|delta| = 2.2e-03   error / |delta| = 9.0e-04
|delta| = 2.2e-04   error / |delta| = 9.0e-05
```

比は 0.09 から 0.009、0.0009、0.00009 へと減少する。0 に収束しており、定義の剰余項 $o(\|\delta\|)$ が確認できる。さらに、この比は 10 分の 1 に $\|\delta\|$ を縮めるごとに 10 分の 1 に減少していることにも注目したい。つまり誤差そのものは 100 分の 1 に減少し、$O(\|\delta\|^2)$ となっている。定義が保証するのは最初の性質だけであり、より速い二次の減衰が成り立つのは $\mathbf{f}$ が連続な二階導関数を持つからである。

34.4.1.2 勾配、ヘッセ行列、導関数の形状

34.2 節で扱った勾配とヘッセ行列（Hessian）は、どちらもヤコビ行列の特別な場合である。この観点に立つと、手計算のたびに生じる形状の疑問、すなわち導関数は行なのか列なのか、また行列に関する導関数はどう並べるべきなのかという問題に決着がつく。

スカラー値関数 $f : \mathbb{R}^n \to \mathbb{R}$（$m = 1$）に対して、ヤコビ行列は 1 行のベクトル $\partial f/\partial \mathbf{x} =$

$[\partial f/\partial x_1, \dots, \partial f/\partial x_n]$ となり、Equation 34.31 は一次展開 $f(\mathbf{x}+\delta) \approx f(\mathbf{x}) + (\partial f/\partial \mathbf{x})\,\delta$ となる。34.2 節では勾配を列ベクトルとして書いた。両者は単に互いの転置である。

$$\nabla_{\mathbf{x}} f = \Big(\frac{\partial f}{\partial \mathbf{x}}\Big)^{\top}, \tag{34.33}$$

本書では、それぞれの目的に応じて両方を使い分ける。ベクトル値写像のヤコビ行列は出力ごとに 1 行を持つ $m \times n$ 行列とし、後述する連鎖律が転置なしですっきりと合成できるようにする。一方、スカラー関数の勾配 $\nabla_{\mathbf{x}} f$ は $\mathbf{x}$ の形状に合わせた列ベクトルとし、オプティマイザがそのまま減算できるようにする。冒頭のネットワークでは、$\partial g/\partial \mathbf{x} = (2,\,4)$ が 1×2 のヤコビ行列であり、$\nabla_{\mathbf{x}} g = (2,4)^{\top}$ が列ベクトルとしての勾配である。

行列 $\mathbf{W} \in \mathbb{R}^{p \times q}$ のスカラー関数には、もう 1 つ規約が必要になる。その勾配 $\nabla_{\mathbf{W}} f$ を、$p \times q$ 行列で各成分が $\partial f/\partial W_{ij}$ であるものとして定義し、$\mathbf{W}$ の形状に一致させる。すると一次展開は次のように書ける。

$$f(\mathbf{W}+\mathbf{E}) = f(\mathbf{W}) + \langle \nabla_{\mathbf{W}} f,\, \mathbf{E} \rangle + o(\|\mathbf{E}\|), \qquad \langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{tr}(\mathbf{A}^{\top}\mathbf{B}) = \sum_{ij} A_{ij} B_{ij}, \tag{34.34}$$

ここで、内積 $\langle \cdot, \cdot \rangle$ は同じ形状の 2 つの行列を成分ごとに掛けて総和をとる演算であり、行列をベクトルに平坦化したときの通常のドット積と同じである。言葉で言えば、f の一次の変化はすべての成分にわたる「感度 × 微小変化」の和である。冒頭の例では、$\nabla_{\mathbf{W}_1} g$ は 2×2 行列 $\left(\begin{smallmatrix} 2 & 2 \\ 0 & 0 \end{smallmatrix}\right)$ であり、$\langle \nabla_{\mathbf{W}_1} g, \Delta \mathbf{W}_1 \rangle = 2\,\Delta W_{11} + 2\,\Delta W_{12}$ から先ほど求めた変化量が復元される。ここから、本節のあらゆる公式に対する 1 つの妥当性確認が得られる。すなわち、勾配は常にその変数と同じ形状を持つ。導出した勾配の形状が合わなければ、転置が抜けているのである。

もう一度微分すると、ヘッセ行列が現れる。勾配場 $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ はそれ自体がベクトル値写像であり、そのヤコビ行列は二階偏微分の行列、すなわちヘッセ行列 $\mathbf{H}_f$ であって、その成分は $[\mathbf{H}_f]_{ij} = \partial^2 f/\partial x_i \partial x_j$ で与えられる。厳密に言えば、$[\mathbf{J}_{\nabla f}]_{ij} = \partial^2 f/\partial x_j \partial x_i$ は 2 回の微分の順序が逆になるが、交差偏微分が連続である限り両者は一致し、$\mathbf{H}_f$ は対称になる（34.2 節のクレロー・シュワルツの定理）。

34.4.2 ヤコビ行列の連鎖律

ニューラルネットワークは層の合成である。スカラーのグラフ上では、連鎖律は入力から出力へ至るすべての経路にわたって導関数の積を足し合わせた（34.3.1 節）。各層がベクトル値写像になると、その経路上の和は行列積になる。

34.4.2.1 行列積としての合成

命題（連鎖律）$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^p$ が $\mathbf{x}$ で微分可能であり、$\mathbf{g} : \mathbb{R}^p \to \mathbb{R}^m$ が $\mathbf{f}(\mathbf{x})$ で微分可能であれば、合成写像 $\mathbf{g} \circ \mathbf{f}$ は $\mathbf{x}$ で微分可能であり、次式が成り立つ。

$$\mathbf{J}_{\mathbf{g}\circ\mathbf{f}}(\mathbf{x}) = \mathbf{J}_{\mathbf{g}}\big(\mathbf{f}(\mathbf{x})\big)\,\mathbf{J}_{\mathbf{f}}(\mathbf{x}). \tag{34.35}$$

証明 $\mathbf{y} = \mathbf{f}(\mathbf{x})$、$\mathbf{A} = \mathbf{J}_{\mathbf{f}}(\mathbf{x})$、$\mathbf{B} = \mathbf{J}_{\mathbf{g}}(\mathbf{y})$ とおく。基本的な考え方は、単に $\mathbf{f}$ の線形モデルを $\mathbf{g}$ の線形モデルに代入することである。Equation 34.31 より、$\mathbf{f}(\mathbf{x}+\delta) = \mathbf{y} + \mathbf{A}\delta + o(\|\delta\|)$ が成り立つ。この変位を $\eta = \mathbf{A}\delta + o(\|\delta\|)$ とおき、$\mathbf{g}$ に渡すと次式を得る。

$$\mathbf{g}\big(\mathbf{f}(\mathbf{x}+\delta)\big) = \mathbf{g}(\mathbf{y}) + \mathbf{B}\,\eta + o(\|\eta\|) = \mathbf{g}(\mathbf{y}) + \mathbf{BA}\,\delta + o(\|\delta\|).$$

2 つ目の等号は 2 つの事実に基づいている。第 1 に、$o(\|\delta\|)$ の剰余項に定数行列 $\mathbf{B}$ を掛けても $o(\|\delta\|)$ のままである。第 2 に、$\|\eta\| \le \|\mathbf{A}\|\,\|\delta\| + o(\|\delta\|)$ は $\|\delta\|$ の定数倍で上から抑えられるため、剰余項 $o(\|\eta\|)$（$\mathbf{g}$ の展開に現れる項）もまた $o(\|\delta\|)$ となる。純粋に代数的ではないステップはここだけである。したがって δ の係数は Equation 34.31 を満たす行列となり、一意性により合成写像のヤコビ行列 $\mathbf{J}_{\mathbf{g}\circ\mathbf{f}} = \mathbf{BA}$ に一致する。■

この積の 1 つの成分に注目すれば、スカラーの場合の経路上の和が復元される。すなわち $[\mathbf{BA}]_{ik} = \sum_j (\partial g_i/\partial y_j)(\partial f_j/\partial x_k)$ は、すべての中間変数 y_j にわたって、x_k から g_i への各経路に沿った導関数の積を足し合わせている。

34.4.2.2 ネットワークのヤコビ行列

冒頭のネットワークに戻ると、これは 3 つの写像の合成とみなせる。アフィン変換 $\mathbf{a} = \mathbf{W}_1\mathbf{x} + \mathbf{b}$、ReLU による活性化値 $\mathbf{h} = \varphi(\mathbf{a})$、そして線形出力 $g = \mathbf{w}_2^{\top}\mathbf{h}$ である。第 1 の写像はもともとアフィンであるため、そのヤコビ行列は $\mathbf{W}_1$ である。バイアスは傾きを変えずに値を平行移動させるにすぎない。ReLU は各成分に独立に作用し、傾きは 1（$a_i > 0$ のとき）および 0（$a_i < 0$ のとき）となるため、そのヤコビ行列は対角行列 $\operatorname{diag}(\mathbf{m})$（これらの傾きからなる**マスク** $\mathbf{m}$ を対角に並べた行列）となる。最後に、出力写像のヤコビ行列は $\mathbf{w}_2^{\top}$ である。$\mathbf{x} = (1, 1)$ においては $\mathbf{a} = (3, -2)$、$\mathbf{m} = (1, 0)$ であり、連鎖律により 3 つの因子の積をとると次のようになる。

$$\mathbf{J}_g(\mathbf{x}) = \mathbf{w}_2^{\top}\operatorname{diag}(\mathbf{m})\,\mathbf{W}_1 = \begin{pmatrix} 2 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 4 \end{pmatrix}.$$

マスクによって $\mathbf{w}_2^{\top}$ の第 2 成分が 0 になるため、$\mathbf{W}_1$ の第 2 行は出力に一切届かない。すなわち、非活性なユニットの重みは導関数にまったく寄与しない（Figure 34.21）。得られた行 $(2, 4)$ は、冒頭で見た変化量 $2\delta_1 + 4\delta_2$ と一致している。入力が変わればマスクも変わり、ヤコビ行列も変化する。たとえば $\mathbf{x} = (2, -3)$ では活性化前の値が $(-4, 5)$ となり、マスクは $(0, 1)$ に反転して $\mathbf{J}_g = (0,\ -1)\,\mathbf{W}_1 = (-3,\ 1)$ となる。ReLU は区分線形であるため、マスクが変わらないほど小さなステップをとれば、各軸に沿った有限差分によってこれらの成分が厳密に再現される（演習問題 1）。

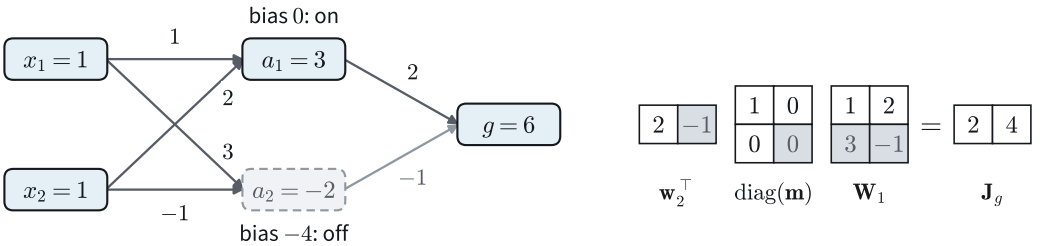


Figure 34.21: 2-2-1 ReLU ネットワークの $\mathbf{x} = (1, 1)$ におけるヤコビ行列は 3 つの行列の積である。第 2 ユニットの活性化前の値は -2 であるためマスク成分が 0 となり、$\mathbf{W}_1$ の第 2 行は出力に届かない。結果として $\mathbf{J}_g = \mathbf{w}_2^{\top}\operatorname{diag}(1, 0)\,\mathbf{W}_1 = (2,\ 4)$ となる。

あらゆる多層パーセプトロン（multilayer perceptron、MLP）は、同じ 2 種類の因子から構成される。全結合層 $\mathbf{x} \mapsto \mathbf{W}\mathbf{x}$ のヤコビ行列は $\mathbf{W}$ である。要素ごとの活性化関数 $\varphi(\mathbf{x}) =$

$(\varphi(x_1), \dots, \varphi(x_n))$ は異なる座標どうしを結合しないため、そのヤコビ行列は対角行列 $\mathbf{J}_\varphi(\mathbf{x}) = \operatorname{diag}(\varphi'(x_1), \dots, \varphi'(x_n))$ となる。ここでは φ がすべての x_i で微分可能であると仮定している。ReLU は 0 で導関数を持たないが、フレームワークでは慣例として固定値を割り当てる（34.1.6 節）。深さ K のネットワークは合成 $\mathbf{f} = \mathbf{f}_K \circ \cdots \circ \mathbf{f}_1$ であり、Equation 34.35 を繰り返し適用すると、その導関数は 1 つの積になる。

$$\mathbf{J} = \mathbf{J}_K \mathbf{J}_{K-1} \cdots \mathbf{J}_1, \tag{34.36}$$

ここで各因子は、それぞれの層への入力において評価される。全結合層と活性化関数が交互に並ぶ場合、これは $\mathbf{J} = \operatorname{diag}(\varphi'_K)\,\mathbf{W}_K \cdots \operatorname{diag}(\varphi'_1)\,\mathbf{W}_1$ となり、重み行列と対角行列が交互に現れる。ベクトルに対角行列を掛ける演算は各成分を定数倍するだけなので、各活性化因子の計算コストは $O(n)$ 回の演算で済み、$O(n^2)$ にはならない。

ここで、本節の後半でも再利用する 2-2-1 ネットワークをコードで定義し、積 $\mathbf{w}_2^\top \operatorname{diag}(\mathbf{m})\mathbf{W}_1$ を自動微分（automatic differentiation）が返す勾配と比較してみよう。

```
W1, b = torch.tensor([[1.0, 2.0], [3.0, -1.0]]), torch.tensor([0.0, -4.0])
w2, x = torch.tensor([2.0, -1.0]), torch.tensor([1.0, 1.0])

def g(x, W1, b, w2):                          # the 2-2-1 ReLU network
    return w2 @ torch.relu(W1 @ x + b)

m = (W1 @ x + b > 0).float()                  # the ReLU mask
print('by hand    :', (w2 @ torch.diag(m) @ W1).numpy())
print('by autograd:', torch.func.grad(g)(x, W1, b, w2).numpy())
```

```
by hand    : [2. 4.]
by autograd: [2. 4.]
```

予想どおり、どちらの行も $(2, 4)$ を出力する。

積 Equation 34.36 からは、深いネットワークを通過するうちに導関数がどれほど大きく、あるいは小さくなりうるかもわかる。すべての因子が同じ行列 $\mathbf{W}$ である場合（活性化関数を無視して、毎ステップ同一の重み行列を適用するリカレントネットワークのような場合）、積は $\mathbf{W}^K$ に簡約される。対角化可能な行列 $\mathbf{W} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ に対しては $\mathbf{W}^K = \mathbf{V}\Lambda^K\mathbf{V}^{-1}$ が成り立つため、大きな K に対してそのノルムは $|\lambda_{\max}|^K$ のように増大または縮小する。ここで $\lambda_{\max}$ は絶対値が最大の固有値であり、$0.9^{50} \approx 0.005$ であるのに対し、$1.1^{50} \approx 117$ となる。この指数関数的なスケーリングこそが、深層ネットワークやリカレントネットワークにおける勾配消失・勾配爆発問題である（4.4 節、8.5 節）。ただし、$|\lambda_{\max}|^K$ が表すのは長期的な変化率にすぎない。$\mathbf{V}$ の固有ベクトルが直交から程遠い場合、積は最終的に減衰する前に何ステップにもわたって増大しうるし（33.4.2.3 節）、異なる因子の積を支配するのは固有値ではなく特異値である（33.4.2.5 節）。

しかし、訓練という観点では Equation 34.36 に重要な限界がある。この式が与えるのは**入力**に関する導関数なのである。重み行列 $\mathbf{W}_k$ は変数ではなく固定された因子として現れているため、重みが変化したときに出力がどう変化するかはまだわからない。重み勾配を計算するには層ごとにあと 1 つの規則が必要であり、34.4.3.2 節でそれを導出して同じネットワークに適用する。

34.4.2.3 乗算の順序

行列積は結合法則を満たすため、積 Equation 34.36 はどのような順序で括弧を付けても厳密に同じ行列になる。しかし、括弧の付け方によって計算コストにはネットワークの幅に比例して拡大するほどの莫大な差が生じる。訓練で求めたいのはスカラー損失 ℓ の勾配であるため、連鎖の左

端は行ベクトル、すなわち $1 \times h$ のヤコビ行列 $\mathbf{u}^\top = \partial \ell / \partial \mathbf{h}_K$（最終層の出力 $\mathbf{h}_K$ に関する損失のヤコビ行列）から始まる。左から順に掛けていけば、中間結果は常に行ベクトルのまま保たれ、すべてのステップが行列・ベクトル積になる。反対に右から掛けていくと、最後にようやく行ベクトルが掛けられるまで、高コストな行列・行列積をずっと作り続けることになる。低コストな順序とは、すべてのステップを行列・ベクトル積に保つ順序である。

演算回数を簡単に数えれば、この差は具体的になる。$\mathbf{u}^\top \mathbf{J}_3 \mathbf{J}_2 \mathbf{J}_1$ の各因子の形状を $1 \times h$、$h \times h$、$h \times h$、$h \times n$ とし、$h = n = 1000$ とおく。なお、$p \times q$ 行列と $q \times r$ 行列の積には pqr 回の乗算を要する。左から掛ける場合、3 回の行列・ベクトル積を行うため、乗算回数は合計 $h^2 + h^2 + hn = 3 \times 10^6$ 回である。右から掛ける場合、$\mathbf{J}_2 \mathbf{J}_1$ と続く $\mathbf{J}_3(\mathbf{J}_2 \mathbf{J}_1)$ の計算にそれぞれ $h^2 n = 10^9$ 回の乗算を要し、最後の $\mathbf{u}^\top$ との積で $hn = 10^6$ 回が加わって合計 2.001×10^9 回となる。300 万回対 20 億回であり、約 670 倍、すなわち $2h/3$ 倍の差がある（Figure 34.22）。

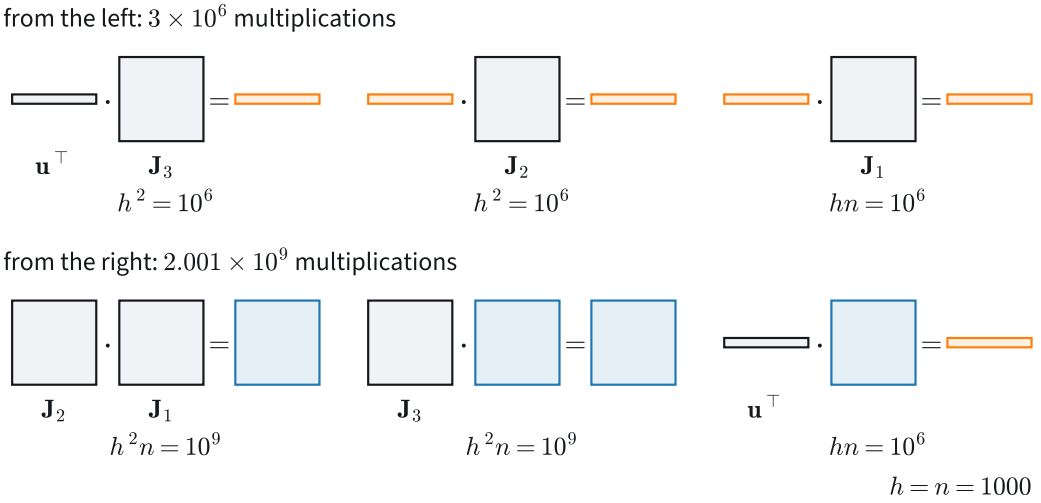


Figure 34.22: $\mathbf{u}^\top \mathbf{J}_3 \mathbf{J}_2 \mathbf{J}_1$（$h = n = 1000$）に対する 2 通りの計算順序。左から掛けると中間結果はすべて $1 \times h$ の行となり、各ステップが行列・ベクトル積となって乗算は 3×10^6 回で済む。右から掛けると最初の 2 ステップがそれぞれ 10^9 回の乗算を要する行列・行列積になる。コストを決めるのは深さではなく形状である。

これら 2 つの順序は、ネットワークをたどる 2 つの向きに対応している。左からの乗算は、層あたり 1 回の行列・ベクトル積のコストで、出力の感度を層に沿って逆方向へ伝播させる。一方、右からの乗算が低コストになるのは、連鎖の右端に列ベクトル $\mathbf{v}$ がある場合であり、単一の入力方向を順方向へ伝播させて $\mathbf{J}\mathbf{v}$ を求める。これら 2 つの順序を自動化するアルゴリズムが、それぞれリバースモードおよびフォワードモードの自動微分と呼ばれる（34.5 節）。

上の演算回数では、すべての層の幅が等しいと仮定した。各因子の形状が異なるとき、左端または右端から一方向に掛ける順序が最適とは限らない。連鎖状の積であれば、最小コストの括弧の付け方は動的計画法（dynamic programming）で求められるが（演習問題 2）、一般の計算グラフではこの問題は NP 完全である (Naumann 2008)。

34.4.3 ベクトルと行列に関する導関数

層を訓練するには、入力、重み、バイアスといったベクトルと行列の双方に関してスカラー損失の勾配を求める必要がある。本項では、必要となるすべての勾配を単一の手法で導出する。この

手法は、それぞれの結果を自動的に変数の形状で返す。

34.4.3.1 全微分法

変数の微小な摂動を $d\mathbf{x}$ と書き、df をその結果生じる f の変化のうち $d\mathbf{x}$ に関して線形な部分を表すものとする。これは単に、項 $\mathbf{J}\delta$（Equation 34.31 において δ を $d\mathbf{x}$ に書き換えたもの）にほかならない。Equation 34.31 と Equation 34.34 より、この線形部分は勾配との内積となる。

$$df = \langle \nabla f,\, d\mathbf{x} \rangle, \tag{34.37}$$

これはベクトル変数（$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^{\top}\mathbf{b}$）と行列変数のどちらに対しても成り立つ。Equation 34.37 は 2 通りに読むことができる。勾配が既知なら、微小変化に対して f がどう応答するかを教えてくれる。一方、f が与えられたなら、勾配を直接求める手順になる。すなわち、通常の微分規則を使って df を計算し、$\langle \mathbf{M},\, d\mathbf{x} \rangle$ の形に整理して、$\nabla f = \mathbf{M}$ を読み取るのである。この読み取りに曖昧さはない。なぜなら、$d\mathbf{x}$ の線形関数を $\langle \mathbf{M}, d\mathbf{x} \rangle$ と表す行列 $\mathbf{M}$ はただ 1 つしかないからである。実際、$d\mathbf{x}$ を、ただ 1 つの 1 を (i, j) 成分に持ち他がすべて 0 である行列にとれば、M_{ij} が取り出せる。

df の計算には、おなじみの 4 つの規則があれば十分である。第 1 に、微分は線形である。第 2 に、積の微分法則 $d(\mathbf{AB}) = d\mathbf{A}\,\mathbf{B} + \mathbf{A}\,d\mathbf{B}$ に従う。なぜなら、$(\mathbf{A} + d\mathbf{A})(\mathbf{B} + d\mathbf{B})$ を展開した式と $\mathbf{AB} + d\mathbf{A}\,\mathbf{B} + \mathbf{A}\,d\mathbf{B}$ との差は、二次の項 $d\mathbf{A}\,d\mathbf{B}$ だけだからである。第 3 に、転置と可換であり、$d(\mathbf{A}^{\top}) = (d\mathbf{A})^{\top}$ が成り立つ。第 4 に、成分ごとに適用されるスカラー関数は各成分でスカラーの連鎖律に従い、$d\varphi(\mathbf{X}) = \varphi'(\mathbf{X}) \odot d\mathbf{X}$ となる。ここで $\odot$ は要素ごとの乗算を表す。

df を $\langle \mathbf{M}, d\mathbf{x} \rangle$ の形に整理するには、もう 1 つだけ道具があればよい。**移動則**（move rule）は、$d\mathbf{x}$ の隣に掛かっている因子を内積の反対側へと移す規則である。

$$\langle \mathbf{A},\, \mathbf{BC} \rangle = \langle \mathbf{B}^{\top}\mathbf{A},\, \mathbf{C} \rangle = \langle \mathbf{AC}^{\top},\, \mathbf{B} \rangle,$$

これはトレースの巡回性 $\operatorname{tr}(\mathbf{XY}) = \operatorname{tr}(\mathbf{YX})$ から直接導かれる。実際、$\operatorname{tr}(\mathbf{A}^{\top}\mathbf{BC}) = \operatorname{tr}\big((\mathbf{B}^{\top}\mathbf{A})^{\top}\mathbf{C}\big) = \operatorname{tr}(\mathbf{CA}^{\top}\mathbf{B}) = \operatorname{tr}\big((\mathbf{AC}^{\top})^{\top}\mathbf{B}\big)$ が成り立つ。言葉で言えば、変数の左にある因子は転置をとって相手側の左へ移り、右にある因子は転置をとって相手側の右へ移る。要素ごとの積の因子はそのまま移り、$\langle \mathbf{A}, \mathbf{B} \odot \mathbf{C} \rangle = \langle \mathbf{A} \odot \mathbf{B}, \mathbf{C} \rangle$ となる。両辺とも $\sum_{ij} A_{ij}B_{ij}C_{ij}$ に等しいからである。

最も単純な 2 つの例は、それぞれ 1 行で済む。線形形式では $d(\mathbf{a}^{\top}\mathbf{x}) = \mathbf{a}^{\top}d\mathbf{x} = \langle \mathbf{a}, d\mathbf{x} \rangle$ より $\nabla_{\mathbf{x}}\, \mathbf{a}^{\top}\mathbf{x} = \mathbf{a}$ である。二次形式では、積の微分法則が 2 つの $\mathbf{x}$ の両方に作用する。

$$\begin{aligned} d(\mathbf{x}^{\top}\mathbf{Ax}) &= d\mathbf{x}^{\top}\mathbf{Ax} + \mathbf{x}^{\top}\mathbf{A}\,d\mathbf{x} = \mathbf{x}^{\top}\mathbf{A}^{\top}d\mathbf{x} + \mathbf{x}^{\top}\mathbf{A}\,d\mathbf{x} \\ &= \big\langle (\mathbf{A} + \mathbf{A}^{\top})\mathbf{x},\, d\mathbf{x} \big\rangle. \end{aligned}$$

2 つ目の等号では 1×1 行列の転置をとった。$d\mathbf{x}^{\top}\mathbf{Ax}$ はスカラーなので、自身の転置 $\mathbf{x}^{\top}\mathbf{A}^{\top}d\mathbf{x}$ に等しい。係数を読み取れば $\nabla_{\mathbf{x}}\, \mathbf{x}^{\top}\mathbf{Ax} = (\mathbf{A} + \mathbf{A}^{\top})\mathbf{x}$ が得られ、これは $2\mathbf{Ax}$ に簡約される（$\mathbf{A}$ が対称な場合）。

添字を書き下すことでも同じ結果に至る。成分ごとに計算すると、同じ導関数は $\partial(\sum_{ij} x_i A_{ij} x_j)/\partial x_k = \sum_j A_{kj}x_j + \sum_i x_i A_{ik}$ となり、第 k 成分として $(\mathbf{A} + \mathbf{A}^{\top})\mathbf{x}$ の対応する成分が復元される。添字による導出は長くなるが、転置に関して疑問の余地が残らない。4.3 節では添字を使って行列積の規則を導出している。

任意の行列恒等式に対する手早い妥当性確認は、すべての行列を 1×1 に置き換えることである。その場合、式はおなじみのスカラーの公式に帰着しなければならない。先ほどの二次形式の勾配は $\frac{d}{dx}(ax^2)=2ax=(a+a)x$ となるため、この確認を通過する。この 1×1 **チェック**は係数の誤りや項の抜けを確実に検出するが、転置や因子の順序には無力である。1×1 行列は可換であり、自身の転置に等しいからである。形状規則はたいていの転置ミスを検出してくれるが、残念ながら因子が正方行列のときには機能しない。これこそ本節の冒頭で警告した落とし穴である。正方の因子に対しては、数値チェックでしか誤りを見つけられない。$\mathbf{A}=\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)$、$\mathbf{x}=(1,2)$ としよう。このとき $\mathbf{x}^{\top}\mathbf{A}\mathbf{x}=x_1x_2$ であり、真の勾配 $(x_2,x_1)=(2,1)$ は $(\mathbf{A}+\mathbf{A}^{\top})\mathbf{x}$ に一致する。一方、素朴な推測 $2\mathbf{A}\mathbf{x}=(4,0)$（スカラーの公式 $2ax$ から連想される式）は、1×1 チェックを通過し、形状も正しいにもかかわらず、やはり誤りである。

全微分法は、行列変数も追加の手間なく扱える。最小二乗法の損失 $\ell=\|\mathbf{W}\mathbf{x}-\mathbf{y}\|^2$ において残差を $\mathbf{r}=\mathbf{W}\mathbf{x}-\mathbf{y}$ とおくと、$\mathbf{W}$ を変化させたとき $d\mathbf{r}=d\mathbf{W}\,\mathbf{x}$ となり、移動則によって $\mathbf{x}$ が反対側へ移る。すなわち $d\ell=2\mathbf{r}^{\top}d\mathbf{r}=\langle 2\mathbf{r},\,d\mathbf{W}\,\mathbf{x}\rangle=\langle 2\mathbf{r}\mathbf{x}^{\top},\,d\mathbf{W}\rangle$ であり、したがって次式を得る。

$$\nabla_{\mathbf{W}}\|\mathbf{W}\mathbf{x}-\mathbf{y}\|^2=2\,(\mathbf{W}\mathbf{x}-\mathbf{y})\,\mathbf{x}^{\top},\tag{34.38}$$

これは残差と入力のランク 1 の外積であり、自動的に $\mathbf{W}$ の形状になっている。代わりに $\mathbf{x}$ を変化させると、$d\mathbf{r}=\mathbf{W}\,d\mathbf{x}$ より $\nabla_{\mathbf{x}}\ell=2\mathbf{W}^{\top}\mathbf{r}$ となる。どちらの場合も、勾配は残差の 2 倍に、変数に掛かっている因子の転置を掛けたものになっている。

34.4.3.2 層の逆伝播則

ネットワークの内部では、層 $\mathbf{Y}=\mathbf{F}(\mathbf{X})$ はスカラー損失 ℓ に至る長い計算の 1 ステップである。その完全なヤコビ行列を作るのは現実的ではない。1000×1000 の重み行列を 1000 個の例からなるミニバッチに適用すると、10^6 個の出力が 10^6 個の重みから生成されるため、重みに関する出力のヤコビ行列は 10^{12} 個もの成分を持つことになる。幸い、誤差逆伝播法が必要とする量ははるかに少ない。層がすべきことは、自身の出力に関する損失の勾配を、入力と重みに関する勾配へと変換することだけである。34.3.2 節に従い、$\bar{\mathbf{Y}}=\nabla_{\mathbf{Y}}\ell$ と書いて $\mathbf{Y}$ の随伴変数（adjoint）を表す。これは $\mathbf{Y}$ と同じ形状の行列であり、$\mathbf{Y}$ の各成分に関する損失の全微分を保持する。

命題（層の逆伝播則） **$\mathbf{Y}=\mathbf{F}(\mathbf{X})$ が $\mathbf{X}$ で微分可能であり、ℓ が $\mathbf{Y}$ の微分可能なスカラー関数であるとする。このとき、$\bar{\mathbf{X}}$（合成関数 $\ell(\mathbf{F}(\mathbf{X}))$ の随伴変数）は、次式を満たす一意な行列である。**

$$\langle \bar{\mathbf{X}},\,d\mathbf{X}\rangle=\langle \bar{\mathbf{Y}},\,d\mathbf{Y}\rangle\quad \textit{for every } d\mathbf{X},$$

ここで、$d\mathbf{Y}$ は $\mathbf{F}$ において $d\mathbf{X}$ が引き起こす一次の変化である。

証明 連鎖律 Equation 34.35 により（行列を平坦化したベクトルとみなせば）、合成関数の線形部分は、ℓ の線形部分を $\mathbf{F}$ の線形部分に適用したものとなり、$d\ell=\langle \bar{\mathbf{Y}},d\mathbf{Y}\rangle$ が成り立つ。$d\mathbf{Y}$ は $d\mathbf{X}$ に関して線形であるため、$d\ell$ もそうであり、$\langle \mathbf{M},d\mathbf{X}\rangle$ の形で一意な行列 $\mathbf{M}$ を用いて表せる。Equation 34.37 より、$\mathbf{M}=\bar{\mathbf{X}}$ である。$\blacksquare$

実際には、この命題の適用は移動則の適用に帰着する。$d\mathbf{y}=\mathbf{J}\,d\mathbf{x}$ を満たすベクトル層に対しては、$\langle \bar{\mathbf{y}},\mathbf{J}\,d\mathbf{x}\rangle=\langle \mathbf{J}^{\top}\bar{\mathbf{y}},d\mathbf{x}\rangle$ となるため、$\bar{\mathbf{x}}=\mathbf{J}^{\top}\bar{\mathbf{y}}$ を得る。すなわち、層の逆伝播則はその線形化の転置にほかならない。線形変換の転置は随伴（adjoint）とも呼ばれ、随伴変数という名称はこれに由来する。$\bar{\mathbf{x}}^{\top}=\bar{\mathbf{y}}^{\top}\mathbf{J}$ と書き直せば、これは Equation 34.36 を左から右へ掛ける順序の 1 ステップそのものである。行ベクトルとヤコビ行列の積は**ベクトル・ヤコビ行列積**（vector–Jacobian product）と呼ばれる。この命題により、行列変数に対してもヤコビ行列を一度も作ることなくこれを計算できる。

深層学習で最も頻繁に現れる導関数は、行列積の導関数である。ミニバッチを処理するか単一のベクトルを処理するかによって重みはデータの右側または左側に置かれるため、両方の配置をまとめて導出しよう。1 行に 1 つの例を並べたミニバッチ $\mathbf{X} \in \mathbb{R}^{n\times d}$（本文全体の規約）に重み行列 $\mathbf{W} \in \mathbb{R}^{d\times q}$ を掛けると、$\mathbf{Y} = \mathbf{X}\mathbf{W}$ が得られる。積の微分法則より $d\mathbf{Y} = d\mathbf{X}\,\mathbf{W} + \mathbf{X}\,d\mathbf{W}$ となり、移動則によって内積は変数ごとの項に分解される。すなわち $\langle \bar{\mathbf{Y}}, d\mathbf{X}\,\mathbf{W}\rangle + \langle \bar{\mathbf{Y}}, \mathbf{X}\,d\mathbf{W}\rangle = \langle \bar{\mathbf{Y}}\mathbf{W}^\top, d\mathbf{X}\rangle + \langle \mathbf{X}^\top\bar{\mathbf{Y}}, d\mathbf{W}\rangle$ であり、両方の係数を読み取ると次式を得る。

$$\bar{\mathbf{X}} = \bar{\mathbf{Y}}\,\mathbf{W}^\top, \qquad \bar{\mathbf{W}} = \mathbf{X}^\top\bar{\mathbf{Y}}. \tag{34.39}$$

単一の例を列ベクトルとして $\mathbf{y} = \mathbf{W}\mathbf{x}$ と書いた場合も、まったく同じ計算から次式が得られる。

$$\bar{\mathbf{W}} = \bar{\mathbf{y}}\,\mathbf{x}^\top, \qquad \bar{\mathbf{x}} = \mathbf{W}^\top\bar{\mathbf{y}},$$

これは Equation 34.38 の形と一致しており、重み勾配は出力の随伴変数と入力の外積になる。2 つの配置は単に互いの転置である。$\mathbf{x}_i^\top$ と $\bar{\mathbf{y}}_i^\top$ をそれぞれ $\mathbf{X}$ と $\bar{\mathbf{Y}}$ の行とすれば、ミニバッチの公式 $\mathbf{X}^\top\bar{\mathbf{Y}} = \sum_i \mathbf{x}_i\bar{\mathbf{y}}_i^\top$ は例ごとにこの外積を足し合わせたものになっている。

したがって、線形層の逆伝播は順伝播の積と同じサイズの 2 つの行列積からなり、順伝播は ndq 回の乗算、逆伝播は $2ndq$ 回の乗算を要する。形状を確認すれば両方の転置も確かめられる。$\bar{\mathbf{Y}}\mathbf{W}^\top$ は $n\times d$ で $\mathbf{X}$ と一致し、$\mathbf{X}^\top\bar{\mathbf{Y}}$ は $d\times q$ で $\mathbf{W}$ と一致する。より複雑な行列演算（逆行列、行列式、各種の行列分解）の逆伝播則も、まったく同じ手法から導かれる (Mike B. Giles 2008)。

あと 2 種類の層が揃えば、多層パーセプトロンが完成する。要素ごとの活性化値 $\mathbf{Y} = \varphi(\mathbf{X})$ は $d\mathbf{Y} = \varphi'(\mathbf{X}) \odot d\mathbf{X}$ を満たすため、$\varphi'(\mathbf{X})$ を内積の反対側へ移すと $\bar{\mathbf{X}} = \varphi'(\mathbf{X}) \odot \bar{\mathbf{Y}}$ が得られる。これは先ほどの対角ヤコビ行列をそのままマスクとして適用したものである。バイアスの加算 $\mathbf{y} = \mathbf{z} + \mathbf{b}$ では $d\mathbf{y} = d\mathbf{b}$ より $\bar{\mathbf{b}} = \bar{\mathbf{y}}$ となる。ミニバッチに対しては、バイアスは $\mathbf{Z} = \mathbf{X}\mathbf{W} + \mathbf{1}\mathbf{b}^\top$ によって各行にブロードキャストされるが、これも単なる行列積であり、すべての例にわたって随伴変数の和をとる $\bar{\mathbf{b}} = \bar{\mathbf{Z}}^\top\mathbf{1}$ を与える。これら 3 つの局所規則が基本要素となり、4.3 節では多層パーセプトロン全体の逆伝播が組み立てられる。

これらの規則を使えば、冒頭の重み勾配行列も導出できる。Figure 34.21 のネットワークを、$\bar{g} = 1$ から 1 層ずつ逆向きに $\mathbf{x} = (1, 1)$ においてたどると次のようになる。

1. 出力 $g = \mathbf{w}_2^\top\mathbf{h}$ から、$\bar{\mathbf{h}} = \mathbf{w}_2 = (2, -1)$ および $\bar{\mathbf{w}}_2 = \mathbf{h} = (3, 0)$ が得られる。
2. ReLU から $\bar{\mathbf{a}} = \mathbf{m} \odot \bar{\mathbf{h}} = (2, 0)$ が得られ、バイアスから $\bar{\mathbf{b}} = \bar{\mathbf{a}} = (2, 0)$ が得られる。
3. アフィン写像 $\mathbf{a} = \mathbf{W}_1\mathbf{x} + \mathbf{b}$ から、$\bar{\mathbf{W}}_1 = \bar{\mathbf{a}}\,\mathbf{x}^\top = \left(\begin{smallmatrix} 2 & 2 \\ 0 & 0 \end{smallmatrix}\right)$ および $\bar{\mathbf{x}} = \mathbf{W}_1^\top\bar{\mathbf{a}} = (2, 4)$ が得られる。

$\bar{\mathbf{x}}$ は先ほど求めたヤコビ行列の行 $(2, 4)$ の転置になっており、$\bar{\mathbf{W}}_1$ は冒頭で重みを微小変化させて得た行列そのものである。コード上で `g` を $\mathbf{W}_1$、$\mathbf{b}$、$\mathbf{w}_2$ に関して微分すると、3 つの配列すべてが一致することが確認できる。

```
gW1, gb, gw2 = torch.func.grad(g, argnums=(1, 2, 3))(x, W1, b, w2)
h = torch.relu(W1 @ x + b)
show = lambda t: (t.numpy() + 0.0).tolist()     # + 0.0 turns -0.0 into 0.0
print('dg/dW1 = (w2*m) x^T:', show(torch.outer(w2 * m, x)),
      '  autograd:', show(gW1))
print('dg/db  = w2*m      :', show(w2 * m), '  autograd:', show(gb))
print('dg/dw2 = h         :', show(h), '  autograd:', show(gw2))
```

```
dg/dW1 = (w2*m) x^T: [[2.0, 2.0], [0.0, 0.0]]   autograd: [[2.0, 2.0], [0.0, 0.0]]
dg/db  = w2*m      : [2.0, 0.0]   autograd: [2.0, 0.0]
dg/dw2 = h         : [3.0, 0.0]   autograd: [3.0, 0.0]
```

どこにゼロが現れるかにも注目しておきたい。非活性な隠れユニットは 3 か所にゼロを残す。$\bar{\mathbf{W}}_1$ のゼロ行、$\bar{\mathbf{b}}$ のゼロ成分、そして $\bar{\mathbf{w}}_2$ のゼロ成分である（その出力 $h_2 = 0$ が $w_{2,2}$ に掛かるためである）。この入力に対する訓練ステップでは、そのユニットに属する 4 つのパラメータはすべて更新されないまま残る。

正方の因子に対する転置の誤りを検出する数値テストがまだ残っていた。失敗しえないテストは何も証明しないため、ここでは誤りが露呈するようにテストを設計する。以下の各因子はすべてランダムな正方行列（したがって非対称行列）とし、正しい規則の隣に同じ形状のもっともらしい誤答候補を並べて出力する。二次形式と行列積の両方をテストしよう。行列積については、スカラー $\langle \bar{\mathbf{Y}}, \mathbf{XW} \rangle$ をランダムな $\bar{\mathbf{Y}}$ に対して微分する。逆伝播則の命題により、$\mathbf{X}$ と $\mathbf{W}$ に関するその勾配はまさに $\bar{\mathbf{X}}$ と $\bar{\mathbf{W}}$ になる。

```
A, x = torch.tensor(np.random.randn(3, 3)), torch.tensor(np.random.randn(3))
X, W, Ybar = (torch.tensor(np.random.randn(3, 3)) for _ in range(3))  # square
ok = lambda u, v: torch.allclose(u, v, atol=1e-5)
grad = torch.func.grad
gx = grad(lambda x: x @ A @ x)(x)
gX, gW = grad(lambda X, W: (Ybar * (X @ W)).sum(), argnums=(0, 1))(X, W)
print('grad x^T A x: (A + A^T) x', ok(gx, (A + A.T) @ x),
      '| wrong guess 2 A x       ', ok(gx, 2 * A @ x))
print('Xbar        : Ybar W^T   ', ok(gX, Ybar @ W.T),
      '| wrong guess Ybar W      ', ok(gX, Ybar @ W))
print('Wbar        : X^T Ybar   ', ok(gW, X.T @ Ybar),
      '| wrong guess (X^T Ybar)^T', ok(gW, (X.T @ Ybar).T))
```

```
grad x^T A x: (A + A^T) x True | wrong guess 2 A x        False
Xbar        : Ybar W^T    True | wrong guess Ybar W       False
Wbar        : X^T Ybar    True | wrong guess (X^T Ybar)^T False
```

導出した規則は `True` を出力し、誤答候補は `False` を出力する。ここではテストの結果以上に、テストの設計が重要である。長方行列の因子、たとえば $\mathbf{X}$ を形状 3×4、$\mathbf{W}$ を形状 4×2 とした場合、積に対する 2 つの誤答候補は実行すらできない。転置の位置を誤ると数値を比較する前に形状エラーが発生するが、これは形状規則が役割を果たした結果である。同様に、対称な $\mathbf{A}$ を使うと、誤答候補 $2\mathbf{Ax}$ は真の勾配と一致してテストをすり抜けてしまう。厄介なのは正方かつ非対称な因子であり、数値チェックではまさにそれを生成しなければならない。

行列積の規則を使えば、積の両方の因子が未知である損失、たとえば行列因子分解（matrix factorization）の損失 $\|\mathbf{X} - \mathbf{UV}\|_F^2$（演習問題 3）も同じように容易に扱える。いずれの場合も、逆伝播則は順伝播でその変数に掛かっていた因子の転置を受け取った随伴変数に掛ける。また、各規則が順伝播のどの値を必要とするかにも注目したい。行列積では $\mathbf{X}$ と $\mathbf{W}$、活性化では $\varphi'(\mathbf{X})$ が必要であり、バイアスでは何も必要としない。逆伝播ではこれらの中間値を順伝播から保持しておく必要があり、そのメモリコストは 34.5 節で直接計測する。

34.4.3.3 ソフトマックスと交差エントロピー

分類器が最も頻繁に評価する導関数は、ソフトマックスと交差エントロピー損失を合成した関数の導関数である。ソフトマックスはスコアのベクトル $\mathbf{z}$、すなわち**ロジット**を確率ベクトルへ変換する。$S = \sum_k e^{z_k}$ とおけば $p_i = e^{z_i}/S$ である。2 つのロジットに対して 34.2 節で log-sum-exp のヘッセ行列を導いたのと同じ商の微分法則から、任意のクラス数に対するソフト

マックスのヤコビ行列が得られる。$\delta_{ij} = 1$（$i = j$ のとき）、それ以外では 0 と書けば、次式が得られる。

$$\frac{\partial p_i}{\partial z_j} = \frac{\delta_{ij} e^{z_i} S - e^{z_i} e^{z_j}}{S^2} = p_i(\delta_{ij} - p_j), \qquad \mathbf{J}_{\text{softmax}} = \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top. \tag{34.40}$$

言葉で言えば、ロジット i を増やすと p_i は変化率 $p_i(1-p_i)$ で増加し、他のすべての確率 p_j は変化率 $p_i p_j$ で減少する。簡単な検算として、確率の和は 1 であるため $\mathbf{J}\mathbf{1} = \mathbf{p} - \mathbf{p}(\mathbf{p}^\top \mathbf{1}) = \mathbf{0}$ となる。すべてのロジットに同じ定数を加えても $\mathbf{p}$ は変化しないため、これは当然成り立たなければならない。結果として、ソフトマックスのヤコビ行列は常に特異である。

また、$\mathbf{J}_{\text{softmax}}$ が対称であることにも注目したい。一般のヤコビ行列は対称とは限らないが、これが対称になるのは実際にはヘッセ行列だからである。すなわち、ソフトマックスは本章で繰り返し用いている例である log-sum-exp 関数 $\mathrm{LSE}(\mathbf{z}) = \log \sum_k e^{z_k}$ の勾配であり（$\partial\,\mathrm{LSE}/\partial z_i = e^{z_i}/S = p_i$）、勾配のヤコビ行列はその関数のヘッセ行列に一致する。2 ロジットの公式に Equation 34.40 が付け加えるのは、この行列の明快な統計的解釈である。これは確率 $\mathbf{p}$ に従って抽出されるワンホットベクトルの共分散行列になっており、したがって任意の $\mathbf{z}$ と任意のベクトル $\mathbf{v}$ に対して、二次形式 $\mathbf{v}^\top \mathbf{J} \mathbf{v} = \sum_i p_i v_i^2 - \left(\sum_i p_i v_i\right)^2$ は $\mathbf{v}$ の各成分の、分布 $\mathbf{p}$ の下での分散になる。分散は決して負にならないため、このヘッセ行列はいたるところで半正定値（positive semidefinite）となり、log-sum-exp が凸性を持つことが証明される。定数ベクトル $\mathbf{v}$ の分散は 0 であり、ここからも零ベクトルを与える方向 $\mathbf{1}$ が再び得られる。

ここで、ソフトマックスと交差エントロピー損失 $\ell = -\sum_i y_i \log p_i$（ターゲット分布 $\mathbf{y}$ に対する損失）を合成しよう。連鎖律により行ベクトル $\partial \ell / \partial \mathbf{p} = (-y_i/p_i)_i$ に Equation 34.40 を掛けると、式は劇的に簡約される。

$$\frac{\partial \ell}{\partial z_j} = -\sum_i \frac{y_i}{p_i} p_i(\delta_{ij} - p_j) = -y_j + p_j \sum_i y_i = p_j - y_j, \qquad \text{that is,} \qquad \nabla_{\mathbf{z}} \ell = \mathbf{p} - \mathbf{y}. \tag{34.41}$$

ロジットに関する勾配は、単に予測確率からターゲットを引いたものになり、これは $\hat{\mathbf{y}} - \mathbf{y}$ として 3.1.2.2 節で書いた公式そのものである。最後の変形では $\sum_i y_i = 1$ しか使っていないため、この結果はワンホットラベルだけでなく任意の確率ベクトル $\mathbf{y}$ に対して成り立つ。したがって、ラベル平滑化（label smoothing）や蒸留のターゲットもまったく同じ公式に従う。和が 1 にならないターゲットに対しては、勾配は代わりに $\left(\sum_i y_i\right)\mathbf{p} - \mathbf{y}$ となる。たとえば $\mathbf{z} = (1, -0.5, 2)$ では $\mathbf{p} = (0.254,\ 0.057,\ 0.690)$ であり、第 2 クラスをターゲットとするとき、勾配は $(0.254,\ -0.943,\ 0.690)$ となる。

実際のライブラリでは、数値安定性のためにソフトマックスと交差エントロピーを単一のプリミティブに融合している（3.3.5.2 節、35.8.2 節）。以下では、自動微分によって $\mathbf{z}$ におけるソフトマックスのヤコビ行列を計算し、その固有値を出力する。分散による議論から予測されるとおり、固有値は非負で 1 つが 0 になるはずである。続いて、融合された損失を、先ほどのワンホットターゲットと、成分の和が 1 にならないツーホットターゲット $\mathbf{y} = (1, 1, 0)$ の両方に対して微分する。

```
z = torch.tensor([1.0, -0.5, 2.0])
p = torch.softmax(z, 0)
J = torch.func.jacrev(lambda z: torch.softmax(z, 0))(z)   # the Jacobian, by autograd
```

```
print('eigenvalues of J:', torch.linalg.eigvalsh(J).numpy().round(3) + 0.0)
loss = lambda z, y: -(y * torch.log_softmax(z, 0)).sum()
for y in [torch.tensor([0.0, 1.0, 0.0]), torch.tensor([1.0, 1.0, 0.0])]:
    print(f'y = {y.numpy()}: d loss / d z = {torch.func.grad(loss)(z, y).numpy().round(4)}, '
          f'p - y = {(p - y).numpy().round(4)}')
```

```
eigenvalues of J: [0.    0.079 0.378]
y = [0. 1. 0.]: d loss / d z = [ 0.2537 -0.9434  0.6897], p - y = [ 0.2537 -0.9434  0.6897]
y = [1. 1. 0.]: d loss / d z = [-0.4926 -0.8868  1.3793], p - y = [-0.7463 -0.9434  0.6897]
```

固有値は 0、0.079、0.378 となり、方向 $\mathbf{1}$ に対応する 1 つのゼロと 2 つの正の値が得られた。ワンホットターゲットに対しては、融合された損失の勾配は $\mathbf{p}-\mathbf{y}=(0.2537,-0.9434,0.6897)$ となり、手計算と一致する。しかしツーホットターゲットに対しては $(-0.4926,-0.8868,1.3793)$ を返し、これは $2\mathbf{p}-\mathbf{y}$ であって、隣に出力された $\mathbf{p}-\mathbf{y}$ ではない。「予測からターゲットを引いた形」になるのは和が 1 になるターゲットの性質であり、損失単独の性質ではない。

本節では、単純な 3 層の連鎖に対して $\mathbf{J}_K\cdots\mathbf{J}_1$ を手計算で掛け合わせた。しかし実際には、プログラムによって分岐や合流を含む任意の計算グラフを扱い、各逆伝播則が必要とする順伝播の値を保持し、パラメータがどれほど多くてもネットワークの評価の小さな定数倍にとどまるコストですべての勾配を計算しなければならない。そのプログラムこそが自動微分であり、34.5 節の主題である。

34.4.4 まとめ

写像 $\mathbf{f}:\mathbb{R}^n\to\mathbb{R}^m$ のヤコビ行列とは、その最良線形近似を表す $m\times n$ 行列であり、各成分は $[\mathbf{J}_\mathbf{f}]_{ij}=\partial f_i/\partial x_j$ で与えられる。勾配（1 行のベクトルを列として書いたもの）もヘッセ行列（勾配のヤコビ行列）も、その特別な場合である。入力に関するネットワークの導関数は各層のヤコビ行列の積 $\mathbf{J}_K\cdots\mathbf{J}_1$ となり、乗算の順序がコストを左右する。幅 1000 の 3 層では、300 万回対 20 億回の乗算という差が生じる。全微分則 $df=\langle\nabla f,d\mathbf{x}\rangle$ を用いれば、行列積の逆伝播則 $\bar{\mathbf{X}}=\bar{\mathbf{Y}}\mathbf{W}^\top$、$\bar{\mathbf{W}}=\mathbf{X}^\top\bar{\mathbf{Y}}$ や、ソフトマックスと交差エントロピーの合成が $\mathbf{p}-\mathbf{y}$ に簡約される結果を含め、層が必要とするあらゆる勾配が自動的に変数の形状で導出される。手計算でのうっかりミスの大半は形状規則と 1×1 チェックで検出できるが、正方の因子における転置の誤りは両方をすり抜けるため、非対称行列を用いた数値テストが必要になる。

34.4.5 演習問題

1. **ReLU ネットワークのヤコビ行列。** $\mathbf{g}(\mathbf{x})=\mathbf{W}_2\,\varphi(\mathbf{W}_1\mathbf{x}+\mathbf{b})$ とし、$\mathbf{W}_1\in\mathbb{R}^{h\times n}$、$\mathbf{b}\in\mathbb{R}^h$、$\mathbf{W}_2\in\mathbb{R}^{m\times h}$、さらに要素ごとの ReLU を φ とする。活性化前の値 $\mathbf{W}_1\mathbf{x}+\mathbf{b}$ のどの成分も 0 でないと仮定する。Equation 34.35 より $\mathbf{J}_\mathbf{g}(\mathbf{x})=\mathbf{W}_2\mathbf{M}(\mathbf{x})\mathbf{W}_1$ であり、ここで対角行列 $\mathbf{M}(\mathbf{x})$ は活性な各隠れユニットに対して成分 1 を、非活性な各隠れユニットに対して成分 0 を持つ。
   1. $\mathbf{g}(\mathbf{x}+\delta)=\mathbf{g}(\mathbf{x})+\mathbf{J}_\mathbf{g}(\mathbf{x})\,\delta$ が十分小さな任意の δ に対して厳密に成り立ち、したがって Equation 34.31 の剰余項が恒等的に消えることを示せ。本節の 2-2-1 ネットワークの $\mathbf{x}=(1,1)$ において、最大の r として、すべての δ $(\|\delta\|_\infty<r)$ に対してこの等式が成り立つ値を求めよ。$n=m=2$ のとき、小さな円の $\mathbf{g}$ による像は Figure 34.20 の像とどう異なるか。
   2. k 個の隠れユニットが $\mathbf{x}$ において活性であるとする。$\operatorname{rank}\mathbf{J}_\mathbf{g}(\mathbf{x})\le k$ を示せ。$n=m=2$ かつ $k=1$ のとき、$\mathbf{x}$ の周りの小さな円の像はどうなるか。
   3. いずれかの活性化前の値が 0 になる入力の集合を記述せよ。$\mathbf{J}_\mathbf{g}$ は、$\mathbf{x}$ がこの集合を横切るときどう変化するか。

2. [code] **最小コストの連鎖順序。**●Equation 34.36 より、K 個の写像の連鎖のヤコビ行列は $\mathbf{J}_K \cdots \mathbf{J}_1$（ただし $\mathbf{J}_k \in \mathbb{R}^{d_k \times d_{k-1}}$）である。$p \times q$ 行列と $q \times r$ 行列の乗算には pqr 回の乗算を要する。
   1. 右から掛ける順方向順序のコストと、左から掛ける逆方向順序のコストを、$d_0, \dots, d_K$ の関数として表せ。広い層と狭い層が交互に並ぶ写像 $(d_0, \dots, d_4) = (1000, 10, 1000, 10, 1000)$ に対して両者を評価し、本節で求めた $(1\times h)(h\times h)(h\times n)$ の演算回数と比較せよ。
   2. 部分連鎖 $\mathbf{J}_j \cdots \mathbf{J}_i$ 上の動的計画法によって最小コストの括弧の付け方を求めるプログラムを書き、(a) の次元に対して実行せよ。どの順序が選ばれ、(a) の 2 つの順序のうち優れた方と比べてどれだけ低コストになるか。
   3. スカラー出力 $d_K = 1$ の場合について、数千個のランダムな次元ベクトル（K は 2 から 6 の範囲とし、$d_0, \dots, d_{K-1}$ の各成分を $\{1, \dots, 10\}$ から一様に抽出する）上で最適順序と逆方向順序を比較せよ。逆方向以外の順序の方が低コストになる次元ベクトルを特徴づけよ。
3. [code] **行列積の逆伝播則。**
   1. $\mathbf{Y} = \mathbf{X}\mathbf{W}$ に対して Equation 34.39 を全微分法で導出し、さらに添字を用いて再度導出して転置を確認せよ。
   2. ベクトル・ヤコビ行列積 `torch.func.vjp` を用い、$\mathbf{X} \in \mathbb{R}^{3\times 4}$、$\mathbf{W} \in \mathbb{R}^{4\times 2}$ およびランダムな随伴変数 $\bar{\mathbf{Y}}$ に対して両方の規則を検証せよ。順伝播の積と逆伝播の 2 つの積それぞれの乗算回数を数えよ。
   3. 行列因子分解の損失 $\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2$（$\mathbf{R} = \mathbf{X} - \mathbf{U}\mathbf{V}$）について、$\nabla_{\mathbf{U}} = -2\mathbf{R}\mathbf{V}^\top$ および $\nabla_{\mathbf{V}} = -2\mathbf{U}^\top\mathbf{R}$ を数値的に確認せよ。次に、一方の因子を固定して他方の勾配を 0 とおく交互最小二乗法の更新ステップを書き下し、各ステップが 1 回の線形方程式の求解になることを示せ。
4. **ソフトマックスのベクトル・ヤコビ行列積。**逆伝播ではソフトマックスのヤコビ行列 Equation 34.40 を陽に作ることは決してなく、確率の随伴変数 $\mathbf{u} = \bar{\mathbf{p}}$ をロジットの随伴変数 $\bar{\mathbf{z}} = \mathbf{J}_{\text{softmax}}^\top \mathbf{u}$ へ直接写す。
   1. $\mathbf{J}_{\text{softmax}}^\top \mathbf{u} = \mathbf{p} \odot \big(\mathbf{u} - (\mathbf{p}^\top \mathbf{u})\,\mathbf{1}\big)$ を示せ（$\odot$ は要素ごとの積）。n クラスの場合について、この式の演算回数と、$\mathbf{J}_{\text{softmax}}$ を作って $\mathbf{u}$ を掛ける場合の演算回数をそれぞれ数えよ。
   2. 自己アテンションは $n \times n$ のスコア行列 $\mathbf{S}$ の各行にソフトマックスを適用して重み $\mathbf{P}$ を生成する。$\mathbf{P}$ の（$\mathbf{S}$ に関する）ヤコビ行列は何個の成分を持ち、そのうち非ゼロになりうるのは何個か。float32 で $n = 4096$ トークンの場合について、完全なヤコビ行列、その非ゼロ成分、および (a) の式を行ごとに適用して返す $\mathbf{S}$ の随伴変数が要するメモリを比較せよ。
5. [code] **LayerNorm のヤコビ行列。**層正規化 (Ba et al. 2016) は、$\mathbf{x} \in \mathbb{R}^n$ を $\hat{\mathbf{x}} = (\mathbf{x} - \mu\mathbf{1})/\sigma$ へ写す。ここで $\mu = \frac{1}{n}\sum_i x_i$、$\sigma^2 = \frac{1}{n}\sum_i (x_i - \mu)^2 > 0$ である（学習されるスケールも ϵ もないものとする）。
   1. 全微分法により $\mathbf{J} = \frac{1}{\sigma}\big(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\hat{\mathbf{x}}\hat{\mathbf{x}}^\top\big)$ を示せ。**ヒント：**$d\mu = \frac{1}{n}\mathbf{1}^\top d\mathbf{x}$ および $d\sigma = \frac{1}{n}\hat{\mathbf{x}}^\top d\mathbf{x}$ である。
   2. $\mathbf{1}$ と $\hat{\mathbf{x}}$ が直交し、かつ $\|\hat{\mathbf{x}}\|^2 = n$ であることを示せ。これから、$\sigma\mathbf{J}$ が $\operatorname{span}\{\mathbf{1}, \hat{\mathbf{x}}\}$ の直交補空間への直交射影であり、$\operatorname{rank}\mathbf{J} = n - 2$（$n \geq 2$ のとき）となることを導け。入力のどのような 2 種類の変化を LayerNorm は感知できないか。また、それはなぜか。
   3. 自動微分で計算したヤコビ行列（ランダムな $\mathbf{x} \in \mathbb{R}^5$ に対するもの）と照合して (a) と 2 つの零ベクトルを確認し、$\mathbf{J}$ の固有値を出力せよ。
6. **対数行列式の勾配。**●正規化フロー (Rezende and Mohamed 2015) は、確率ベクトル $\mathbf{X}$ を可逆な層に順に通すモデルである。1 つの層 $\mathbf{Y} = \mathbf{g}(\mathbf{X})$ に対して、$\mathbf{Y}$ の点 $\mathbf{g}(\mathbf{x})$ における対数密度は、密度の変数変換の公式より $\log p_{\mathbf{X}}(\mathbf{x}) - \log\big|\det \mathbf{J}_{\mathbf{g}}(\mathbf{x})\big|$ となる。線形層 $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$

では補正項が $-\log|\det \mathbf{A}|$ となるため、最尤法による訓練では $\log|\det \mathbf{A}|$ を $\mathbf{A}$ に関して微分する。

1. $\det(\mathbf{I}+t\mathbf{M})$ を t の一次まで展開し、$\det(\mathbf{A}+d\mathbf{A}) = \det\mathbf{A}\,\det(\mathbf{I}+\mathbf{A}^{-1}d\mathbf{A})$ を用いて、ヤコビの公式 $d\det\mathbf{A} = \det\mathbf{A}\,\operatorname{tr}(\mathbf{A}^{-1}d\mathbf{A})$ を可逆行列 $\mathbf{A}$ に対して導け。さらに Equation 34.37 を用いて $\nabla_{\mathbf{A}}\log|\det\mathbf{A}| = \mathbf{A}^{-\top}$ を導け。なぜ $\det\mathbf{A}$ の符号は無関係なのか。
2. ランダムな 3×3 の可逆行列上で、`np.linalg.slogdet` の中心差分によってこの恒等式を確認せよ。
3. $\mathbf{A}$ が特異行列に近づくとき、勾配のノルムはどう振る舞うか。これは、重み行列がほぼ特異になったフロー層に対して何を意味するか。

7. **スペクトルノルムの勾配。**●スペクトル正規化 (Miyato et al. 2018) は、ネットワークの各重み行列 $\mathbf{W}$ をその最大特異値 $\sigma_1(\mathbf{W})$ で割る手法であり、訓練では σ_1 を微分することになる。
   1. $\mathbf{S}$ を単純固有値 λ と単位固有ベクトル $\mathbf{q}$ を持つ実対称行列とし、これを $\mathbf{S}+t\mathbf{E}$（$\mathbf{E}$ は対称行列）へ摂動させる。$\lambda(t)$ と $\mathbf{q}(t)$（$(\mathbf{S}+t\mathbf{E})\mathbf{q} = \lambda\mathbf{q}$ と $\mathbf{q}^\top\mathbf{q}=1$ で定まる関数）は $t=0$ の近傍で微分可能であるとしてよい（λ が単純固有値であるため陰関数定理から従う）。両式を $t=0$ で微分してヘルマン・ファインマンの定理 $\lambda'(0) = \mathbf{q}^\top\mathbf{E}\,\mathbf{q}$ を示し、$\nabla_{\mathbf{S}}\lambda$ を内積 $\operatorname{tr}(\mathbf{A}^\top\mathbf{B})$ の下で求めよ。
   2. (a) を $\sigma_1(\mathbf{W})^2 = \lambda_{\max}(\mathbf{W}^\top\mathbf{W})$ に適用し、$\nabla_{\mathbf{W}}\sigma_1 = \mathbf{u}_1\mathbf{v}_1^\top$ が（$\sigma_1>0$ が単純であるとき）成り立つことを示せ。ここで $\mathbf{u}_1$ と $\mathbf{v}_1$ は上位の左特異ベクトルと右特異ベクトルである。なぜ $\sigma_1>0$ が必要なのか。$\mathbf{W}=\mathbf{w}\in\mathbb{R}^{n\times 1}$ を $\mathbf{w}=\mathbf{0}$ の近傍で考察せよ。NumPy でランダムな 5×3 行列上の中心差分によってこの公式を確認せよ。
   3. $\sigma_1(\mathbf{I}+t\,\operatorname{diag}(1,-1))$ を正負両符号の小さな t に対して計算せよ。σ_1 は 2×2 単位行列において微分可能か。また、そこでは (a) のどの仮定が破れているか。

Alan Edelman および Steven G. Johnson による Matrix Calculus for Machine Learning and Beyond, problem set 2（MIT 18.S096、IAP 2023、問題 3）を改変。

34.5 自動微分

本書の「基礎」パートを通じて、あらゆる勾配はフレームワークの自動微分エンジンを 1 回呼び出すだけで得られていた。前節では、その呼び出しが数学的に何を計算しているかを示した。すなわち、ネットワークの導関数は各層に 1 つずつ対応するヤコビ行列の積であり、それらの行列をどの順序で掛けるかによってコストが決まる。しかし、実際のプログラムは整然と並んだ行列のリストではなく、加算、乗算、関数呼び出しの系列である。フレームワークは通常のコードからどのようにそれらの因子を取り出し、低コストな順序で掛け合わせているのだろうか。

低コストな順序によってどれだけ計算を節約できるかは、現代のネットワークを考えれば一目瞭然である。10^8 個のパラメータを持ち、単一のスカラー損失 ℓ を出力するモデルを考えよう。各訓練ステップでは、ℓ の 10^8 個すべてのパラメータに関する導関数が必要になる。有限差分（finite difference）による直接的な手法では、パラメータを 1 つずつ微小に動かし、ネットワークを再実行して、ℓ の変化量をその微小変化で割る。これには 1 ステップあたり 10^8 回の実行を要する。1 回の実行に 10 ms かかるとすると、1 つの勾配を計算するだけで 10^6 s、すなわち 12 日近くかかり、しかもステップサイズの選び方によって打ち切り誤差や丸め誤差が入り込む。これに対し、誤差逆伝播法（backpropagation）は 10^8 個すべての微分値を、浮動小数点の丸めを除いて厳密に、順伝播およそ 3 回分のコスト（約 30 ms）で返す。本節では、この 10^8 倍という差がどこから生まれるのかを解き明かす。

核心となるアイデアは驚くほど単純である。ヤコビ行列の積は、どちらの端からでも評価できる。入力側から始めると、1 回のパスで 1 つの入力方向が順方向に伝播するため、10^8 個の入力には

10^8 回のパスが必要になる。一方、損失側から始めると、1 回のパスで 1 つの出力方向が逆方向に伝播するが、スカラー損失の出力はただ 1 つしかない。

1.5 節では、深層学習フレームワークで両方のモードを呼び出す方法を見た。本節ではその内部の仕組みに踏み込む。数十行の Python コードでフォワードモードとリバースモードをゼロから構築し、それらが真の導関数を返すことを証明し、逆伝播を手計算でトレースし、各モードのコストを計測する。さらに、2 つのモードを合成してある方向に沿った二階の曲率を計算し、ソルバ自体を通して微分することなく方程式の解を微分する方法を示す。なかでも特に重要な要点は 2 つある。第 1 に、プログラムを**二重数**（dual numbers）上で評価すると自動的に微分値が得られること（本節で唯一の命題）、第 2 に、どのスイープもプログラムを 1 回評価する場合の小さな定数倍のコストしかかからず、フォワードモードは入力方向ごとに 1 回のスイープを、リバースモードは出力方向ごとに 1 回のスイープを必要とすることである。初めて読む読者は、34.5.3 節を 2 回目の通読時に回してもよい。

以下の議論では、3 つの小さな例を一貫して用いる。本章で扱っている勾配がソフトマックスになる log-sum-exp $\ell(\mathbf{x}) = \log \sum_j e^{x_j}$、前節で導入した平面写像 $\mathbf{f}(x, y) = (x + \sin y,\ y + x^2/2)$、そして手計算で完全にトレースできるほど小さな多項式 $y = (uv + u)^2$ である。

本節全体を通して、以下のインポートを使用する。

```
import numpy as np
import torch
```

34.5.1 フォワードモードと二重数

34.5.1.1 二重数

1 回の実行からプログラムの値と導関数の両方を、ステップサイズの調整も、あらかじめ f' の数式を書き下すこともなしに求めたい。その鍵となる仕掛けは、実数のごく小さな拡張である。0 ではないが二乗すると消える新しい記号 ε を添加する。

$$\varepsilon^2 = 0, \qquad \varepsilon \neq 0. \tag{34.42}$$

このたった 1 つの規則のもとで、**二重数**と呼ばれる $a + b\varepsilon$ の形の式を通常の代数法則に従って計算する。34.1 節のあらゆる微分公式は二次の項を切り捨てることで成り立っていたが、$\varepsilon^2 = 0$ とおくことで、その切り捨てが算術そのものに直接組み込まれる。

この規則を $3 + \varepsilon$ に適用するとどうなるか見てみよう。二乗および三乗すると、次のようになる。

$$(3 + \varepsilon)^2 = 9 + 6\varepsilon + \varepsilon^2 = 9 + 6\varepsilon, \qquad (3 + \varepsilon)^3 = (9 + 6\varepsilon)(3 + \varepsilon) = 27 + 27\varepsilon.$$

ε の係数に注目しよう。6 と 27 は、x^2 と x^3 の $x = 3$ における導関数の値である。微分公式は何ひとつ使っていないにもかかわらず、乗算から直接微分値が現れた。同じことは任意の積について成り立つ。2 つの二重数を掛け合わせ、ε^2 の項を落とすと次式を得る。

$$(a + b\varepsilon)(c + d\varepsilon) = ac + (ad + bc)\,\varepsilon. \tag{34.43}$$

b と d が、それぞれ値 a と c をとる 2 つの量の導関数であるとき、$ad + bc$ はそれらの積の導関数となる。つまり、二重数の乗算は積の微分法則そのものである。

懐疑的な読者は、0 でないにもかかわらず二乗すると 0 になる数などあり得ないと反論するかもしれない。もちろん実数には存在しないが、行列なら容易に構成できる。$\varepsilon = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ とおこう。すると $\varepsilon^2 = \mathbf{0}$ となり、二重数 $a + b\varepsilon$ は 2×2 行列 $\left(\begin{smallmatrix} a & b \\ 0 & a \end{smallmatrix}\right)$ になり、通常の行列積 $\left(\begin{smallmatrix} a & b \\ 0 & a \end{smallmatrix}\right)\left(\begin{smallmatrix} c & d \\ 0 & c \end{smallmatrix}\right) = \left(\begin{smallmatrix} ac & ad+bc \\ 0 & ac \end{smallmatrix}\right)$ は Equation 34.43 を再現する。複素数が平面の回転と拡大縮小と同じくらい具体的な対象であるのと同様に、二重数も 2×2 行列と同じくらい具体的な対象なのである。

実際のプログラムは sin、exp、log などの初等関数も呼び出す。関数 φ が a の周りのテイラー級数で与えられるとき、その級数に $a + b\varepsilon$ を代入すると $(b\varepsilon)^2$ 以降のすべての項が消え、一次近似だけが残る。

$$\varphi(a + b\varepsilon) = \varphi(a) + \varphi'(a)\, b\, \varepsilon.$$

したがって、$e^{a+b\varepsilon} = e^a + e^a b\,\varepsilon$ や $\sin(a + b\varepsilon) = \sin a + (\cos a)\, b\,\varepsilon$ は代数構造そのものから必然的に導かれる。解析的ではなく、単に a で微分可能であるにすぎないプリミティブについては、上の式を二重数に対する作用の定義として採用する。いずれの場合も、ε 成分は連鎖律に従う。すなわち、引数から伝わってきた導関数の値 b に局所導関数 $\varphi'(a)$ が掛けられる。

命題（二重数は導関数を計算する） **式 f が、定数、恒等写像、和、積、およびプリミティブ φ（sin、exp、log など）から構成されているとする。f を x で評価したときに各プリミティブが受け取る引数において、そのプリミティブが微分可能であるとする（log の場合は正の引数）。このとき、同じ式を二重数 $x + \varepsilon$ 上で評価すると次式が成り立つ。**

$$f(x + \varepsilon) = f(x) + f'(x)\,\varepsilon. \tag{34.44}$$

証明： 式の構造に関する構造的帰納法（structural induction）で示す。定数 c は $c + 0\varepsilon$、恒等写像は $x + 1\varepsilon$ と表されるため、それぞれの ε 成分はその導関数である 0 と 1 になる。次に、2 つの部分式が $g(x + \varepsilon) = g(x) + g'(x)\varepsilon$ および $h(x + \varepsilon) = h(x) + h'(x)\varepsilon$ を満たすと仮定する。これらの和の ε 成分は $g' + h' = (g + h)'$ である。また Equation 34.43 より、積の ε 成分は $g'h + gh' = (gh)'$ となる。最後に、プリミティブ φ を g に適用すると、$a = g(x)$ および $b = g'(x)$ とした上の規則により、ε 成分は $\varphi'(g(x))\, g'(x) = (\varphi \circ g)'(x)$ となる。部分式を組み合わせるどの操作でもこの性質が保たれるため、式全体も Equation 34.44 を満たす。■

したがって、プログラムを微分するには、入力を $x + \varepsilon$ として実行し、出力の ε 成分を読み取ればよい。ループや条件分岐を含むプログラムも同様に自動で処理される。任意の固定された入力に対して、プログラムは基本演算からなる 1 本の具体的な系列を実行するが、これはまさに上の命題の形の式になっており、二重数による実行はその実行経路を微分する（フレームワークでの実例は 1.5.3.1 節を参照）。ただし、得られるのはあくまで実行された経路の導関数であり、これが $f'(x)$ と一致するのは、近傍のすべての入力が同じ分岐を通る場合に限られる。たとえば、値 `x` を `x != 0` のときに返し、それ以外では `0.0` を返すプログラムは恒等写像を計算するが、$x = 0$ では定数側の分岐を実行するため、二重数による実行は傾き 0 を報告する。これは、$\mathrm{ReLU}(x) - \mathrm{ReLU}(-x)$ について 34.1.6.3 節で見た失敗と同じである。入力の ε 成分（ここでは 1）は**シード**と呼ばれる。f' の数式表現が作られることは一度もなく、ステップサイズを選ぶ必要もない。生じる誤差は浮動小数点の丸め誤差のみである。

34.5.1.2　フォワードモードのエンジン

演算子オーバーロードを使えば、この代数をそのままコードに落とし込める。以下の `Dual` クラスは、値 a とその微分パート b を組にして保持する。この微分パートを**タンジェント**と呼び、$\dot{v}$ と書く（元の量を v とする）。このクラスは Equation 34.43 を `*` に対して実装し、`sin`、`exp`、`log` に

プリミティブの微分規則を備えている。クラスカウンタ ops は実行された基本演算の回数を記録するもので、34.5.2.4 節でコストを計測する際に用いる。

```
class Dual:
    """A dual number a + b*eps with eps^2 = 0; b carries the derivative."""
    ops = 0                                          # primitive operations run
    def __init__(self, a, b=0.0):
        self.a, self.b = a, b                        # value, tangent
    def __add__(self, o):
        o = o if isinstance(o, Dual) else Dual(o)
        Dual.ops += 1
        return Dual(self.a + o.a, self.b + o.b)
    __radd__ = __add__
    def __mul__(self, o):
        o = o if isinstance(o, Dual) else Dual(o)
        Dual.ops += 1
        return Dual(self.a * o.a, self.a * o.b + self.b * o.a)   # product rule
    __rmul__ = __mul__
    def sin(self):
        Dual.ops += 1
        return Dual(np.sin(self.a), np.cos(self.a) * self.b)     # chain rule
    def exp(self):
        Dual.ops += 1
        return Dual(np.exp(self.a), np.exp(self.a) * self.b)
    def log(self):
        Dual.ops += 1
        return Dual(np.log(self.a), self.b / self.a)
    def __repr__(self):
        return f'{np.round(self.a, 6)} + {np.round(self.b, 6)} eps'
```

Dual を $f(x) = \sin(x^2) + e^x$ の $x = 1.3$ においてテストしてみよう。引数に $1.3 + \varepsilon$ を渡し、出力の ε 成分を手計算で求めた導関数 $f'(x) = 2x\cos(x^2) + e^x$ と比較する。

```
def f(x):
    return (x * x).sin() + x.exp()                   # sin(x^2) + e^x, in dual algebra

x0 = 1.3
out = f(Dual(x0, 1.0))                               # seed the derivative with 1
exact = 2 * x0 * np.cos(x0**2) + np.exp(x0)          # f'(x) by hand
print('dual result :', out)
print('value f(x)  :', np.sin(x0**2) + np.exp(x0))
print("derivative  : dual %.6f  vs  exact %.6f" % (out.b, exact))
```

```
dual result : 4.6622 + 3.360101 eps
value f(x)  : 4.662200318713363
derivative  : dual 3.360101  vs  exact 3.360101
```

1 回の実行で関数値 4.6622 と導関数の値 3.360101 の両方が返され、表示されたすべての桁で解析的な公式と一致している。

しかし、1 変数の例ではフォワードモードの最大の制約が隠れてしまう。3 次元入力 $\mathbf{x} = (1, -0.5, 2)$ における log-sum-exp を考えよう。その勾配はソフトマックス $\partial \ell / \partial x_j = e^{x_j} / \sum_k e^{x_k} = p_j$ である。この勾配全体を得るのに、フォワードモードは何回のパスを必要と

するだろうか。入力 j にシードとして 1 を、他のすべての入力に 0 を与えると、単一の偏微分 $\partial \ell / \partial x_j$ が出力の ε 成分に現れ、それ以外は何も得られない。

```
def lse_dual(xs):                                   # log-sum-exp, in dual algebra
    total = 0.0
    for x in xs:
        total = total + x.exp()
    return total.log()

x_lse = [1.0, -0.5, 2.0]
p = np.exp(x_lse) / np.exp(x_lse).sum()             # softmax, the known gradient
for j in range(3):                                  # one pass per input direction
    out = lse_dual([Dual(x, 1.0 if k == j else 0.0) for k, x in enumerate(x_lse)])
    print(f'seed e{j + 1}: value {out.a:.6f}, partial {out.b:.6f}, '
          f'softmax {p[j]:.6f}')
```

```
seed e1: value 2.371539, partial 0.253716, softmax 0.253716
seed e2: value 2.371539, partial 0.056612, softmax 0.056612
seed e3: value 2.371539, partial 0.689672, softmax 0.689672
```

3 回のパスが必要になる。各パスは同じ値 2.371539 を再計算しながら、ソフトマックスの要素を 0.253716、0.056612、0.689672 と 1 つずつしか取り出せない。より一般に、n 個の入力に関する勾配を求めるには n 回のパスを要する。冒頭で挙げた 10^8 パラメータのネットワークでは、結局プログラムを 10^8 回実行することに逆戻りしてしまう。フォワードモードは有限差分のステップサイズこそ不要にするが、実行回数は減らせないのである。

34.5.1.3 ベクトルタンジェントとヤコビ行列の列

次に、ベクトル値写像 $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ と、その $m \times n$ ヤコビ行列 $\mathbf{J}$ を考えよう。1 回の順伝播は何を計算するだろうか。任意の方向 $\mathbf{v}$ を入力のシードとし、入力 j を $x_j + v_j \varepsilon$ とおくとしよう。先ほどの命題を 1 次元のスライス $t \mapsto f_i(\mathbf{x} + t\mathbf{v})$ に適用すると、出力 i は $f_i(\mathbf{x}) + (\nabla f_i \cdot \mathbf{v})\,\varepsilon$ として現れることがわかる。したがって、ε 成分を m 個すべての出力にわたって集めると、ベクトル $\mathbf{J}\mathbf{v}$ が得られる。これは**ヤコビ行列・ベクトル積**（Jacobian–vector product、JVP）と呼ばれる。$\mathbf{v}$ が j 番目の標準基底ベクトル $\mathbf{e}_j$ であるとき、JVP は第 j 列を $\mathbf{J}$ から取り出す。すなわち、1 回の順伝播は 1 つの入力方向を写像全体にわたって伝播させる。

これを平面写像 $\mathbf{f}(x, y) = (x + \sin y,\; y + x^2/2)$ の点 $(0.5, 0.5)$ において確かめてみよう。第 i 行が出力 i の偏微分を並べたものになっているヤコビ行列は、次のようになる。

$$\mathbf{J} = \begin{pmatrix} 1 & \cos y \\ x & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.877583 \\ 0.5 & 1 \end{pmatrix}.$$

まず $\mathbf{e}_1$、次に $\mathbf{e}_2$ をシードとして 2 回の順伝播を実行すると、ヤコビ行列の 2 つの列が復元される。

```
def plane_dual(x, y):
    return [x + y.sin(), y + 0.5 * (x * x)]         # (x + sin y, y + x^2/2)

z0 = (0.5, 0.5)
cols = []
for seed in ((1.0, 0.0), (0.0, 1.0)):               # e1, then e2
```

```
        out = plane_dual(Dual(z0[0], seed[0]), Dual(z0[1], seed[1]))
        cols.append([out[0].b, out[1].b])              # tangent = J @ seed, a column
    J_fwd = np.array(cols).T                           # columns side by side
    J_exact = np.array([[1.0, np.cos(0.5)], [0.5, 1.0]])
    print('forward-mode J:\n', J_fwd.round(6))
    print('matches the analytic Jacobian:', np.allclose(J_fwd, J_exact))
```

```
forward-mode J:
 [[1.       0.877583]
 [0.5      1.      ]]
matches the analytic Jacobian: True
```

ここで、Dual の実装にはタンジェントをスカラーに限定する制約が何もないことに注目しよう。x にベクトルタンジェント $(1, 0)$ を、y にベクトルタンジェント $(0, 1)$ を与えれば、すべての演算が両方の入力方向を同時に伝播させ、出力 i のタンジェントは第 i 行を $\mathbf{J}$ から取り出したものになる。では、1 回のパスで行列全体を復元できるのだろうか。

```
    out = plane_dual(Dual(z0[0], np.array([1.0, 0.0])),     # identity seed:
                     Dual(z0[1], np.array([0.0, 1.0])))     # both directions at once
    J_one = np.array([out[0].b, out[1].b])                  # row i = tangent of f_i
    print('one-pass J:\n', J_one.round(6))
    print('matches the analytic Jacobian:', np.allclose(J_one, J_exact))
```

```
one-pass J:
 [[1.       0.877583]
 [0.5      1.      ]]
matches the analytic Jacobian: True
```

復元はできるが、計算量の節約は見かけ倒しである。タンジェントの各演算が長さ n の配列に対して作用するようになるため、その 1 回のパスはスカラーのパスに比べておよそ n 倍の算術演算を実行するからである。パス数を数えることと算術演算の回数を数えることは別問題であり、算術演算数は 34.5.2.4 節で直接計測する。10^8 個の入力を持つスカラー損失に対して、フォワードモードは 10^8 回のスカラーパスを実行するか、長さ 10^8 のタンジェントを伴う 1 回のパスを実行するかのいずれかを必要とする。どちらにしても、総計算量はプログラムの実行 n 回分に比例して増大する。

34.5.2 リバースモードとテープ

34.5.2.1 1 つの積に対する 2 つの計算順序

n 倍という計算量の増大をなくすには、導関数の連鎖を逆の順序で掛け合わせる必要がある。34.4 節で見たように、K ステップのプログラム $\mathbf{f} = \mathbf{f}_K \circ \cdots \circ \mathbf{f}_1$ のヤコビ行列は、各ステップのヤコビ行列の積で表される。

$$\mathbf{J} = \mathbf{J}_K \mathbf{J}_{K-1} \cdots \mathbf{J}_1, \tag{34.45}$$

そして、この積にどう括弧をつけるかによってコストが決まることも見た。たとえば幅 1000 の 3 つの層に左から行ベクトルを掛ける場合、左から順に結合すれば乗算は 3×10^6 回で済むのに対し、右から結合すると 2×10^9 回を要する（Figure 34.22）。自動微分エンジンは、これらのヤコビ行列を明示的に実体化することは決してない。その代わり、シードベクトルを 1 ステップず

つ順に伝播させる。フォワードモードは列方向のベクトル $\mathbf{v}$ を右から掛け、リバースモードは行ベクトル $\mathbf{u}^\top$ を左から掛ける。

$$\mathbf{J}\mathbf{v} = \mathbf{J}_K\Big(\cdots(\mathbf{J}_2(\mathbf{J}_1\mathbf{v}))\Big), \qquad \mathbf{u}^\top\mathbf{J} = \Big(\big((\mathbf{u}^\top\mathbf{J}_K)\mathbf{J}_{K-1}\big)\cdots\Big)\mathbf{J}_1. \tag{34.46}$$

最初からシードベクトルが存在するため、どのステップも行列・ベクトル積になり、行列・行列積になることは一度もない。左の式では、入力方向が順方向に伝わって JVP $\mathbf{J}\mathbf{v}$ を生成する。右の式では、出力方向が逆方向に伝わって**ベクトル・ヤコビ行列積**（vector–Jacobian product、VJP）$\mathbf{u}^\top\mathbf{J}$ を生成する。これは $\mathbf{J}$ の各行を $\mathbf{u}$ で重み付けした線形結合である。$\mathbf{u} = \mathbf{e}_i$ とおけば、第 i 行を $\mathbf{J}$ から取り出せる。

この対称性から、自動微分の基本的なコスト規則が導かれる。フォワードモードは入力方向ごとに 1 回のスイープを要し、リバースモードは出力方向ごとに 1 回のスイープを要するが、各スイープのコストはプログラムを 1 回評価する場合の小さな定数倍にすぎない。$+, -, \times, \div$ から構成されるプログラムについて、Baur と Strassen は、すべての偏微分をまとめて計算するのに必要な演算回数が、数える演算の種類に応じてプログラム自体の演算回数の高々一定倍（3〜5 倍）に収まることを証明した (Baur and Strassen 1983)。スカラー損失のヤコビ行列は $1 \times n$、すなわち 1 行だけであるため、1 をシードとする 1 回の逆方向スイープで勾配全体が一度に返される。Figure 34.23 は、この規則を行列の形状の観点から示したものである。

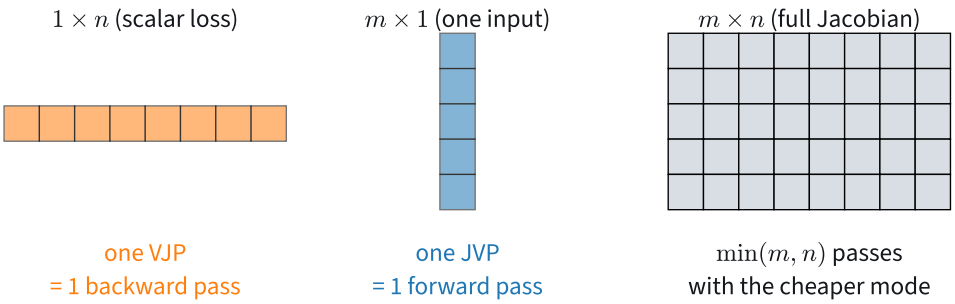


Figure 34.23: ヤコビ行列の形状によって低コストなモードが決まる。スカラー損失のヤコビ行列は $1 \times n$ であり、1 行、1 回の VJP、1 回の逆伝播で済む。1 入力の写像のヤコビ行列は $m \times 1$ であり、1 列、1 回の JVP、1 回の順伝播で済む。$m \times n$ ヤコビ行列全体を求めるコストは、低コストな方のモードを使えば $\min(m, n)$ 回のパスになる。

2 つのモードは、まったく同じ行列積を反対側の端から評価している。どちらが有利かは $\mathbf{J}$ の形状だけで決まり、シードの個数が短い方の辺に一致するモードの方が低コストになる。1 つのスカラー出力と数百万から数十億個のパラメータを持つニューラルネットワークの損失は、考えられる限り最も極端に縦横比が偏ったヤコビ行列を持っているのである。

34.5.2.2 テープ

フォワードモードとリバースモードが伝播させる量は、根本的に異なる。入力 x を出力 y に写すプログラムにおいて、タンジェント $\dot{v}$（中間変数 v に対するもの）は、v が入力の $\dot{x}$ 方向への移動に伴ってどれだけ速く変化するかを測る。これに対し、**随伴変数**（adjoint）$\bar{v}$（34.3 節で導入）は、最終出力 y の v に関する全微分であり、v から y へのすべての経路にわたって和をとったものである。タンジェントは v より**前**のステップのみに依存するのに対し、随伴変数は v より**後**の

ステップに依存する。たとえば $y = v^2$（ただし $v = 3x$）では、下側から伝わる $\dot{v} = 3\dot{x}$ と、上側から伝わる $\bar{v} = 2v$ が得られる。

この非対称性が、各モードの実行方式を決定づける（Figure 34.24）。タンジェントは値と同じ方向に進むため、フォワードモードはプログラム自体の実行と並行して 1 回のパスで完了し、`Dual` が過去の中間値を記憶しておく必要はまったくない。しかし随伴変数は、実行の流れに逆らって進む。あるノードの随伴変数を計算するには、そのノードを利用するすべてのノードの随伴変数が必要であり、それらはプログラムが出力に到達した後にしか存在しないからである。したがって、リバースモードは 2 段階で動作する。まずプログラムを順方向に実行しながらすべての基本演算を記録し、次にその記録を逆向きに再生する。この記録は、Wengert (1964) がフォワードモードのために導入した基本演算のリストにちなんで、**テープ**または Wengert リストと呼ばれる。このリストを逆順にたどることで、Linnainmaa (1970) が導入したリバースモードが得られる。

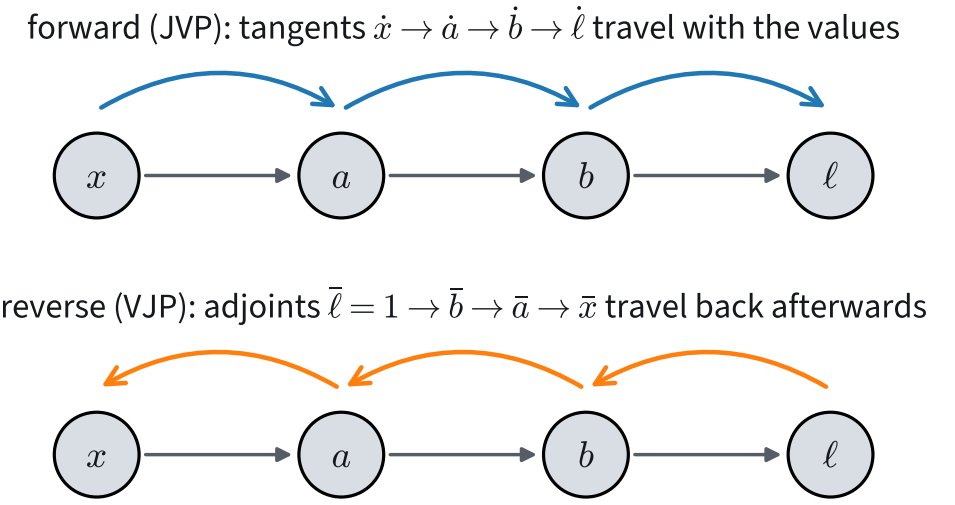


Figure 34.24: 連鎖 $x \to a \to b \to \ell$ におけるフォワードモードとリバースモード。どちらもまず左から右へ連鎖を評価する（灰色）。フォワードモード（上）は値とともにタンジェント $\dot{v}$ を運び、1 回のスイープで 1 つの入力方向を処理する。リバースモード（下）は先に連鎖を実行してから、随伴変数 $\bar{v}$（ℓ の v に関する全微分）をシード $\bar{\ell} = 1$ から逆向きに運び、1 回のスイープで 1 つの出力方向（スカラー損失の場合は勾配全体）を求める。

実用的に動くリバースモードのエンジンは、約 40 行の Python コードで構築できる。テープ上の各ノードは、順伝播の値、親ノード、および小さな逆伝播クロージャを保持する。そのノード自身の随伴変数 g が与えられると、クロージャは各親ノードの随伴変数にこのノードからの寄与を加算する。すなわち、$c = a + b$ に対しては両方の親に g を足し、$c = ab$ に対しては $b\,g$ を $\bar{a}$ に、$a\,g$ を $\bar{b}$ に足す。関数 `backprop` は、すべてのノードがその親より後に現れるようにグラフをトポロジカルソートし、出力の随伴変数にシードとして 1 を設定したうえで、クロージャを逆順に呼び出す。`Dual` と同様に、このクラスも順伝播と逆伝播の両ステップにわたって `ops` カウンタを管理する。

```
class Var:
    """A scalar node on the autodiff tape; .grad accumulates the adjoint."""
    ops = 0                                          # recorded ops + backward steps
```

```
    def __init__(self, value, parents=(), backward=lambda g: None):
        self.value = value
        self.grad = 0.0
        self._parents = parents                    # nodes this one depends on
        self._backward = backward                  # pushes adjoint to parents
    def __add__(self, o):
        o = o if isinstance(o, Var) else Var(o)
        Var.ops += 1
        out = Var(self.value + o.value, (self, o))
        def back(g):                               # d(a+b) = 1*g to each parent
            Var.ops += 1
            self.grad += g; o.grad += g
        out._backward = back
        return out
    __radd__ = __add__
    def __mul__(self, o):
        o = o if isinstance(o, Var) else Var(o)
        Var.ops += 1
        out = Var(self.value * o.value, (self, o))
        def back(g):                               # product rule, reversed
            Var.ops += 1
            self.grad += o.value * g; o.grad += self.value * g
        out._backward = back
        return out
    __rmul__ = __mul__

def backprop(node):
    """Topologically sort the tape and replay it backward from `node`."""
    order, seen = [], set()
    def visit(v):
        if v not in seen:
            seen.add(v)
            for p in v._parents:
                visit(p)
            order.append(v)
    visit(node)
    node.grad = 1.0                                # seed the output adjoint
    for v in reversed(order):
        v._backward(v.grad)
```

Var が実装しているのは + と * だけだが、3 つ目の例である $y=(uv+u)^2$ を扱うにはこれで十分である。コードを実行する前に、$u=2$、$v=-3$ における逆方向スイープ全体を手計算でトレースしてみよう。順伝播は 3 つの基本演算、すなわち $t=uv=-6$、次に $r=t+u=-4$、最後に $y=r\cdot r=16$ を記録する。逆方向スイープは y, r, t, u, v のような逆トポロジカル順序でノードを訪問し、各ノードがそれに依存するすべてのノードより後に処理されることを保証する（backprop はたまたま v を u より先に訪問するが、これも同様に正しい順序である）。各随伴変数は出力辺に沿って届く寄与の総和であり、それぞれの寄与は利用先ノードの随伴変数に局所導関数を掛けたものになる。

ノード	出力辺とその寄与	随伴変数
y	なし（シード）	$\bar{y}=1$
r	$y=r\cdot r$ の両方の引数：$\bar{y}\,r+\bar{y}\,r$	$\bar{r}=-8$

ノード	出力辺とその寄与	随伴変数
t	$r = t + u$: $\bar{r} \cdot 1$	$\bar{t} = -8$
u	$r = t + u$ および $t = uv$: $\bar{r} \cdot 1 + \bar{t} \cdot v$	$\bar{u} = -8 + 24 = 16$
v	$t = uv$: $\bar{t} \cdot u$	$\bar{v} = -16$

$y = (uv + u)^2$ を直接微分すれば、$\partial y/\partial u = 2(uv + u)(v + 1) = 2(-4)(-2) = 16$ および $\partial y/\partial v = 2(uv + u)\, u = 2(-4)(2) = -16$ となり、両方の答えが正しいことが確かめられる。グラフ内の 2 箇所、すなわち r が最後の積の両方の引数に入るところと、u が t と r の両方に入るところで、加算による累積がいかに重要かに注目しよう（Figure 34.25）。もし加算ではなく上書きしてしまうと $\bar{r} = -4$ になり、仮に $\bar{r}$ を正しく直したとしても、どちらのクロージャが最後に実行されたかによって $\bar{u}$ が -8 または 24 になってしまう。随伴変数はすべての出力辺にわたって和をとる（連鎖律における全経路の和に対応する）ため、すべての逆伝播クロージャは `+=` で累積しなければならない。

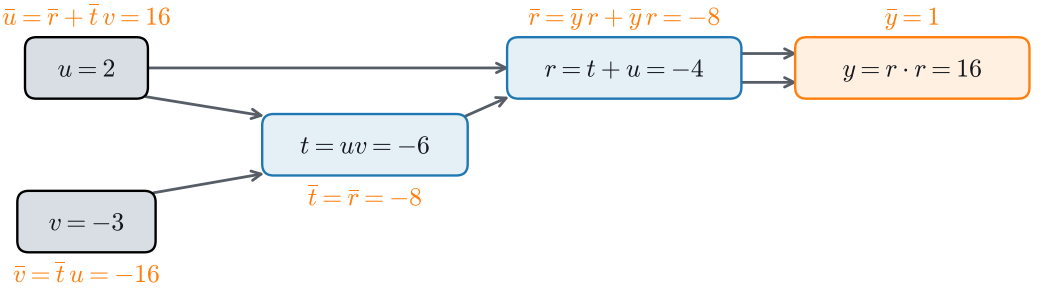


Figure 34.25: $y = (uv + u)^2$ の $u = 2$、$v = -3$ におけるテープ。各ボックスはそのノードの順伝播の値を示し、オレンジ色のラベルはその随伴変数、すなわちそのノードに関する y の導関数を示す。グラフは一直線の連鎖ではなくひし形になっている。u は $t = uv$ と $r = t + u$ の両方に入り、r は $y = r \cdot r$ の両方の引数に入るため、r と u の随伴変数はそれぞれ 2 つの寄与の和となる。

同じ式を `Var` で記録し、1 回の逆方向スイープを実行すると次の結果が得られる。

```
u, v = Var(2.0), Var(-3.0)
r = u * v + u                                   # t = uv, then r = t + u
y = r * r                                       # r feeds both arguments
backprop(y)
print('our tape  : dy/du = %.4f  dy/dv = %.4f' % (u.grad, v.grad))
```

```
our tape  : dy/du = 16.0000  dy/dv = -16.0000
```

これらは手計算のトレースで求めた $\bar{u}$ と $\bar{v}$ に一致する。実用レベルの深層学習フレームワークも、まったく同じアルゴリズムをテンソル規模で実行し、同じ 2 つの導関数の値を出力する。

```
ut = torch.tensor(2.0, requires_grad=True)
vt = torch.tensor(-3.0, requires_grad=True)
rt = ut * vt + ut
(rt * rt).backward()                            # framework autograd
```

```
print('torch     : dy/du = %.4f  dy/dv = %.4f' % (ut.grad, vt.grad))
```

```
torch     : dy/du = 16.0000  dy/dv = -16.0000
```

実用フレームワークは、ここで作った簡易的な `Var` エンジンと実務上の 3 つの点で異なるが、いずれも根底にあるアルゴリズムを変えるものではない。第 1 に、ノードはスカラーではなくテンソル全体を保持するため、各逆伝播クロージャは行列積に対する $\bar{\mathbf{X}} = \bar{\mathbf{Y}}\mathbf{W}^\top$ や $\bar{\mathbf{W}} = \mathbf{X}^\top\bar{\mathbf{Y}}$（34.4.3.2 節）のように、テンソル演算全体の VJP を実装する。第 2 に、プリミティブのライブラリがはるかに充実しており、それぞれに人手で最適化された VJP 規則が組み合わされている。第 3 に、フレームワークが計算グラフを構築する方式には 2 通りある。**即時テープ**（eager tape）は、`Var` と同じようにプログラムの実行に合わせて動的に演算を記録する。これに対し**トレーサ**は、抽象的なプレースホルダ入力に対してプログラムを 1 回実行して演算の系列を捕捉し、各ステップの線形化されたフォワードモード規則を構築したうえで、その系列を転置して逆順にする。どちらの方式でも、得られる逆方向スイープはまったく同じである。ニューラルネットワークに適用したとき、このリバースモードのスイープこそが深層学習で**誤差逆伝播法**と呼ばれるものである (Rumelhart et al. 1986)。

PyTorch は即時に記録する。`requires_grad=True` を持つテンソルに対するすべての演算がグラフにノードを追加し、`backward()` がそれらのノードを逆順に再生する。

各フレームワークはまた、二重数の命題における仮定、すなわち各プリミティブが受け取る引数において微分可能でなければならないという条件も引き継いでいる。実務上、この仮定が最も頻繁に破られるのがマスク処理である。パディングされたベクトルの有効な要素にわたって $\log x_i$ の和をとり、`where` によってパディング要素を 0 に置き換える場合を考えよう。`where` はすでに計算された 2 つの値のうち一方を選ぶ演算であるため、パディング要素（ここでは $x_2 = 0$）に対しても $\log$ が実行されてしまう。順伝播では問題は生じない。$-\infty$ は捨てられ、損失は $\log 2 \approx 0.6931$ となる。しかし逆方向スイープでは、`where` が捨てられた側の分岐にも随伴変数 0 を送り込み、$\log$ の VJP がそこに局所導関数 $1/0 = \infty$ を掛ける。浮動小数点演算では、$0 \cdot \infty$ は 0 ではなく NaN になる。

```
x = torch.tensor([2.0, 0.0], requires_grad=True)    # x[1] is padding
mask = torch.tensor([True, False])
loss = torch.where(mask, torch.log(x), 0.0).sum()   # sum of log x, real entries
loss.backward()
print('one where   : loss = %.4f, grad =' % loss.item(), x.grad.numpy())
x.grad = None
safe = torch.where(mask, x, 1.0)                    # padding never reaches log
loss = torch.where(mask, torch.log(safe), 0.0).sum()
loss.backward()
print('double where: loss = %.4f, grad =' % loss.item(), x.grad.numpy())
```

```
one where   : loss = 0.6931, grad = [0.5 nan]
double where: loss = 0.6931, grad = [0.5 0. ]
```

1 行目は勾配 $(0.5, \mathrm{NaN})$ を報告しているが、損失はパディング要素にまったく依存しておらず、そこでの真の導関数の値は 0 である。たった 1 つの NaN でも局所的な問題では済まず、x_2 の上流にあるすべてのパラメータへと伝播してしまう。2 行目は、標準的な対処法である**ダブル where** を示している。内側の `where` が、プリミティブに渡る前にマスクされた要素を無害な値（ここでは 1）に置き換えるため、無限大の局所導関数が作られることはなくなり、勾配は $(0.5, 0)$ となる。引数を定義域の端から離すようにクランプする手法も同じ役割を果たす。同じ落とし穴は $\sqrt{x}$ の 0 における評価や、それに伴うノルム $\sqrt{\sum_i x_i^2}$ の $\mathbf{x} = \mathbf{0}$ における計算にも潜んでいる。

34.5.2.3 パラメータ勾配

ここまでヤコビ行列の積 Equation 34.45 はネットワークの入力に関する導関数を追跡してきたが、訓練で必要になるのは重みに関する導関数である。幸いなことに、重みの勾配は層の VJP 規則（34.4.3.2 節）を通じて、まったく同じ逆方向スイープから自然に得られる。線形層 $\mathbf{h}_k = \mathbf{W}_k \mathbf{h}_{k-1}$（重み行列 $\mathbf{W}_k$ の形状は $d_k \times d_{k-1}$）に対して、この規則は次式で与えられる。

$$\nabla_{\mathbf{W}_k} \ell = \bar{\mathbf{h}}_k \, \mathbf{h}_{k-1}^{\top}, \tag{34.47}$$

これは、上側から届く随伴変数と下側から保存しておいた活性化値との外積である。直感的には、ある重みの重要度は、その出力が損失にどれだけ強く影響するかという度合いと、入力の活性化値がどれだけ大きかったかという度合いの双方に比例する。コスト解析において重要なのは、この規則がどれだけの算術演算を要するかである。

外積 $\bar{\mathbf{h}}_k \, \mathbf{h}_{k-1}^{\top}$ を作るには $d_k d_{k-1}$ 回の乗算を要し、これは順伝播の行列・ベクトル積 $\mathbf{W}_k \mathbf{h}_{k-1}$ と同じ回数である。随伴変数を前の層へ引き戻す計算 $\bar{\mathbf{h}}_{k-1}^{\top} = \bar{\mathbf{h}}_k^{\top} \mathbf{W}_k$ にも、さらに $d_k d_{k-1}$ 回の乗算がかかる。したがって乗算回数で測ると、逆方向スイープのコストは順伝播のおよそ 2 倍となり、順伝播と逆方向スイープを合わせたコストは順伝播の約 3 回分となって、冒頭で述べた 30 ms という数値と一致する。対照的にフォワードモードでは、個々の重みをそれぞれ独立な入力方向として扱わなければならず、たった 1 つの層だけでも $d_k d_{k-1}$ 回のパスが必要になってしまう。どのフレームワークもフォワードモードでニューラルネットワークを訓練しない理由はここにある。

34.5.2.4 2 つのモードのコスト

両方のエンジンを自前で構築したので、漸近的な議論に頼らず演算回数を直接計測できる。$f(\mathbf{x}) = (\sum_j x_j^2)^2$ を、$n = 200$ 個の入力 $x_j = 0.01\,j$（$j = 1, \dots, 200$）に対して考えよう。$+$ と $\times$ だけから構成される多項式を選んだのは、`Var` に `exp` や `log` のクロージャ（`Dual` のメソッドと対になるもの）を追加しなくてもそのまま記録できるようにするためである。この関数の勾配は $\nabla f = 4\big(\sum_j x_j^2\big)\mathbf{x}$ であり、通常の評価 1 回あたり $2n + 1 = 401$ 回の基本演算（n 回の二乗、n 回の加算、そして最後の 1 回の二乗）を実行する。フォワードモードが 200 回のパスでそれぞれ 401 回の演算を必要とすることは明らかである。以下の出力を読む前に少し立ち止まって、リバースモードが何回の演算を要するか予想してみてほしい。

```
n = 200
def f(xs):                          # f: R^n -> R, from + and * alone
    total = 0.0
    for x in xs:
        total = total + x * x
    return total * total            # (sum of squares)^2

xs0 = [0.01 * (j + 1) for j in range(n)]
Dual.ops = 0
grad_fwd = [f([Dual(x, 1.0 if k == j else 0.0) for k, x in enumerate(xs0)]).b
            for j in range(n)]   # forward mode: one pass per input direction
Var.ops = 0
xs = [Var(x) for x in xs0]          # reverse mode: one recorded pass...
backprop(f(xs))                     # ...and one backward sweep for all n partials
grad_rev = [x.grad for x in xs]
```

```
    gap = max(abs(a - b) for a, b in zip(grad_fwd, grad_rev))
    print(f'forward mode: {Dual.ops:,} operations ({n} passes of {2 * n + 1})')
    print(f'reverse mode: {Var.ops:,} operations (one recorded pass, one sweep)')
    print(f'ratio {Dual.ops / Var.ops:.1f}, largest gradient difference {gap:.1e}')
```

```
forward mode: 80,200 operations (200 passes of 401)
reverse mode: 802 operations (one recorded pass, one sweep)
ratio 100.0, largest gradient difference 0.0e+00
```

フォワードモードが $80{,}200 = n(2n+1)$ 回の演算を実行したのに対し、リバースモードはわずか $802 = 2(2n+1)$ 回（順伝播で記録された 401 回と、逆方向スイープ中の逆伝播クロージャ 401 回）しか実行しなかった。その比は $100 = n/2$ であり、2 つの勾配は表示されたすべての桁で一致している。これらのエンジンでは、パス数は n 倍、演算回数は $n/2$ 倍異なり、$n = 10^8$ においては小さな定数倍を除いて冒頭の倍率と一致する。

一見すると、ここで計測された 2 倍という倍率は、先ほど挙げた 3 倍という数字と矛盾するように見えるかもしれない。しかし矛盾はない。乗算の逆伝播クロージャは内部で 2 回の乗算を行っているにもかかわらず、`Var.ops` は各クロージャを 1 回の演算として数えているからである。スカラー乗算の回数を直接数えれば、逆方向スイープの 402 回に対して順伝播は 201 回であり、線形層の場合とまったく同様に、値と勾配を合わせて計算するコストは値のみの場合の 3 倍になる。随伴変数を累積する加算まで含めると逆方向スイープは 1204 回の演算となり、全体の倍率は 4 倍になるが、それでもなお Baur–Strassen の上界の範囲内に十分収まっている。

リバースモードは、この速度と引き換えにメモリを消費する。逆方向スイープが使い終えるまでテープは記録された 401 個すべての中間値を保持し続けなければならないのに対し、フォワードモードは元のプログラムと同様、現在の値とそのタンジェントだけを保持すればよい。

34.5.2.5 勾配チェックポイント

記録されるステップ数が T の計算では、逆方向スイープが到達するまで T 個の中間値をメモリ上に保持し続ける。深層ニューラルネットワークにおいて、それらの値とはバッチ内の各データ例に対するすべての層の活性化値であり、バッチサイズやネットワークの深さの上限を決める要因は、算術演算量ではなく活性化メモリであることが多い。

最も単純な対処法は、テープを s ステップずつの区間に分割することである。順伝播の間は s ステップごとの値、すなわち T/s 個の**チェックポイント**だけを保存し、その間にある中間値はすべて破棄する。逆方向スイープがある区間に到達したら、保存しておいたチェックポイントからその区間の s ステップを一時的なテープ上に再計算し、その短いテープを逆方向にスイープした後、ただちに解放する。どの時点においても、メモリが保持するのは T/s 個のチェックポイントと、処理中の長さ s の区間 1 つ分だけである。これら 2 つの項が釣り合うように s を選べば、ピークメモリが最小化される。

$$M(s) = \frac{T}{s} + s, \qquad \min_{s} M(s) = M(\sqrt{T}) = 2\sqrt{T}. \tag{34.48}$$

この最小値は、$M'(s) = 1 - T/s^2 = 0$ とおくか、あるいは相加・相乗平均の不等式 $T/s + s \geq 2\sqrt{T}$ から直接得られる。これによりピークメモリは計算長に対して線形ではなく、その平方根のオーダーでしか増大しなくなる。各ステップは 1 回だけ再計算されるため、追加の計算量は順伝播わずか 1 回分である。$T = 400$ の場合、区間長 $s = 1, 5, 10, 20, 100, 400$ に対するピークメモリは $M = 401, 85, 50, 40, 104, 401$ となり、$s = 20 = \sqrt{400}$ で最小値をとる。$T = 10^4$ で

は、長さ $s = 100$ の区間を用いることで、保存する値はわずか 200 個で済み、10,000 個すべてを保持する必要がなくなる (Chen et al. 2016)。Figure 34.26 にこのスケジュールを示す。

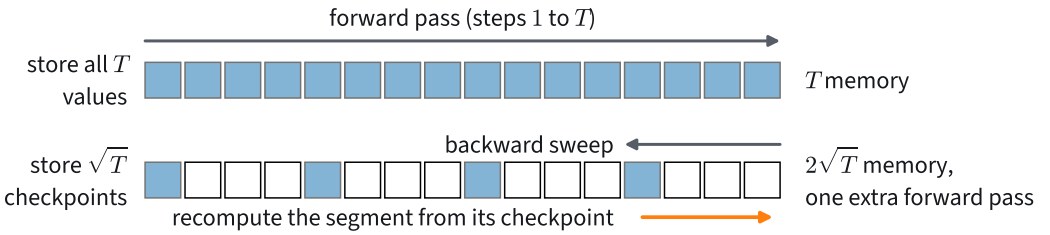


Figure 34.26: T ステップのテープにおける勾配チェックポイント。通常のリバースモード（上）は、逆方向スイープが使用し終えるまですべての T 個の値を保持する。チェックポイント保存（下）は s ステップごとの値だけを保持し、逆方向スイープがある区間に到達すると、その区間をチェックポイントから再計算してからスイープする。順伝播 1 回分の追加計算により、メモリ使用量は $T/s + s$、すなわち $2\sqrt{T}$（$s = \sqrt{T}$ のとき）となる。

重要な点として、再計算が決定論的である限り、チェックポイント保存は勾配を一切変えることなく計算時間とメモリ使用量だけを変化させる。再計算された中間値が元の値と完全に同一になるため、得られる導関数の値も通常のリバースモードとビット単位で一致するからである。深層学習フレームワークはこのトレードオフをネットワークのブロック単位で提供しており、チェックポイント対象ブロックの入力だけを保存し、誤差逆伝播法の実行中にそのブロックを再実行する。ただし実務では、この決定論性という前提条件に注意が要る。乱数生成器の状態を復元しなければ再実行されたブロックは新しいドロップアウトのマスクを引き直してしまい、バッチ正規化は移動統計量を 2 回更新してしまい、非決定論的な GPU カーネルは 1 回目の実行結果をビット単位で再現するとは限らない。同じアイデアを各区間の内部に再帰的に適用すれば、メモリ使用量を $O(\log T)$ まで削減できる（再計算のコストは $O(T \log T)$ となる）。

34.5.3 ヤコビ行列積の合成

自動微分は、ここまで簡易エンジンで構築してきた 2 つの線形作用素、すなわちフォワードモードの JVP $\mathbf{Jv}$ とリバースモードの VJP $\mathbf{u}^{\top}\mathbf{J}$ の両方を提供する。この 2 つを自在に合成できるプリミティブとして扱えるようになると、方向曲率を計算したり、数値ソルバを通して微分したり、必要に応じてヤコビ行列全体を組み立てたりできるようになる。

34.5.3.1 ヤコビ行列・ベクトル積とベクトル・ヤコビ行列積

1.5.4 節では、両方の積に対するフレームワークの API を紹介した。平面写像の点 $(0.5, 0.5)$ において、$\mathbf{v} = \mathbf{u} = (1, 2)$ としてこれらを評価してみよう。先ほど `Dual` で求めたヤコビ行列を用いると、$\mathbf{Jv} = (1 + 2 \cdot 0.8776,\ 0.5 + 2) = (2.7552,\ 2.5)$ および $\mathbf{u}^{\top}\mathbf{J} = (1 + 2 \cdot 0.5,\ 0.8776 + 2) = (2,\ 2.8776)$ になるはずであり、フレームワークの呼び出し結果もこれらの値と直接一致する。

```
def plane(z):                                       # (x + sin y, y + x^2/2)
    return torch.stack([z[0] + torch.sin(z[1]), z[1] + 0.5 * z[0]**2])

z0 = torch.tensor([0.5, 0.5])
v, u = torch.tensor([1.0, 2.0]), torch.tensor([1.0, 2.0])
_, Jv = torch.func.jvp(plane, (z0,), (v,))          # forward mode: J v
_, vjp_fn = torch.func.vjp(plane, z0)               # reverse mode: records once,
```

```
    uJ, = vjp_fn(u)                                  # then u^T J per seed
    J = torch.func.jacrev(plane)(z0)                 # one VJP per row
    print('J v   =', Jv.numpy().round(4))
    print('u^T J =', uJ.numpy().round(4))
    print('J     =', J.numpy().astype(float).round(4).tolist())
```

```
J v   = [2.7552 2.5   ]
u^T J = [2.     2.8776]
J     = [[1.0, 0.8776], [0.5, 1.0]]
```

JVP と VJP はどちらも 1 回のスイープ分のコストで済み、導関数に関する情報のベクトルだけを伝播させる（ただし VJP はテープ上に順伝播の中間値も保持する必要がある）。密な $m \times n$ ヤコビ行列を組み立てるには、低コストな方のモードで $\min(m, n)$ 回のスイープを実行するのに加え、数値 mn 個分の記憶領域が必要になる。行列全体を実体化するのが適切なのは、n が小さい場合（この 2×2 の例など）や、確率密度を $|\det \mathbf{J}|$ によって変換する正規化フロー（normalizing flow）のようにアルゴリズムが行列式を必要とする場合である。

34.5.3.2 ヘッセ行列・ベクトル積

$n = 10^6$ 個のパラメータを持つモデルでは、ヘッセ行列の要素数は 10^{12} 個に達し、float32 で保存すると 4 TB ものメモリを占有する。しかし、二次最適化手法が行列そのものを必要とすることはめったにない。ニュートン法がヘッセ行列を使うのは $\mathbf{H}\mathbf{d} = -\nabla \ell$ を更新ステップ $\mathbf{d}$ について解くためだけであり、ニュートン CG 法（または打ち切りニュートン法）はその連立一次方程式を共役勾配法（conjugate gradient method）で解く。共役勾配法は反復ソルバであり、$\mathbf{H}$ には行列・ベクトル積 $\mathbf{H}\mathbf{v}$ を通じてのみアクセスする (Nocedal and Wright 2006; Martens 2010)。では、$\mathbf{H}\mathbf{v}$ を、$\mathbf{H}$ を一度も作らずに計算できるだろうか。

リバースモードとフォワードモードを合成すれば計算できる。ヘッセ行列は勾配写像 $\nabla \ell$ のヤコビ行列であるため、積 $\mathbf{H}\mathbf{v}$ は単に勾配の JVP であり、方向 $\mathbf{v}$ に沿って勾配がどれだけ速く変化するかを測る量にほかならない。

$$\mathbf{H}\mathbf{v} = \mathbf{J}_{\nabla \ell}\,\mathbf{v} = \left.\frac{d}{dt}\right|_{t=0} \nabla \ell(\mathbf{x} + t\mathbf{v}). \tag{34.49}$$

逆方向スイープが $\nabla \ell$ を評価し、その勾配計算全体に対してタンジェント $\mathbf{v}$ をシードとするフォワードモードの順伝播を適用することで、勾配を $\mathbf{v}$ 方向に微分する。この合成は**フォワード・オーバー・リバース**と呼ばれる (Pearlmutter 1994)。次元 n によらず、そのコストは 1 回の勾配計算の小さな定数倍にとどまる。

log-sum-exp は閉じた形の解を持つ格好のテストケースとなる。その勾配はソフトマックス $\mathbf{p}$ であるため、ヘッセ行列はソフトマックスのヤコビ行列になる（34.4 節、同値な見方としてワンホットベクトル $\mathbf{e}_J$（ただし J は $\mathbf{p}$ からサンプリングされる）の共分散行列でもある）。したがって、次式が成り立つ。

$$\mathbf{H} = \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^{\top}, \qquad \mathbf{H}\mathbf{v} = \mathbf{p} \odot \mathbf{v} - \mathbf{p}\,(\mathbf{p}^{\top}\mathbf{v}), \tag{34.50}$$

ここで、$\odot$ は要素ごとの積を表す。右辺の評価に必要な乗算はわずか $3n$ 回であり、$n \times n$ 行列を作ることは一度もない。$\mathbf{x} = (1, -0.5, 2)$ および $\mathbf{v} = (1, 0.5, -1)$ においては、$\mathbf{p} = (0.2537, 0.0566, 0.6897)$、$\mathbf{p}^{\top}\mathbf{v} = -0.4077$ となり、次の結果が得られる。

$$\mathbf{Hv} = (0.3571,\ 0.0514,\ -0.4085).$$

これは二次形式 $\frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x}$ でテストするのとは異なり、本質的な検証になっていることに注目しよう。二次形式では $\mathbf{v}$ における勾配がたまたま $\mathbf{Hv}$ と一致するため、バグのある実装でもテストを通過してしまう恐れがある。一方 log-sum-exp では $\mathbf{H}$ が $\mathbf{x}$ に応じて変化し、$\mathbf{v}$ における勾配 $\text{softmax}(\mathbf{v}) = (0.574, 0.348, 0.078)$ は $\mathbf{Hv}$ とは似ても似つかない値になる。

両方のモードが利用できれば、フォワード・オーバー・リバースはわずか 1 行のコードで書ける。

```
x = torch.tensor([1.0, -0.5, 2.0])
v = torch.tensor([1.0, 0.5, -1.0])
lse = lambda x: torch.logsumexp(x, 0)
_, Hv = torch.func.jvp(torch.func.grad(lse), (x,), (v,))  # forward over reverse
p = torch.softmax(x, 0)
print('forward over reverse :', Hv.numpy().round(4))
print('p*v - p (p.v)        :', (p * v - p * (p @ v)).numpy().round(4))
```

```
forward over reverse : [ 0.3571  0.0514 -0.4085]
p*v - p (p.v)        : [ 0.3571  0.0514 -0.4085]
```

計算された積は Equation 34.50 と表示された全桁の精度で一致しており、その過程で 3×3 のヘッセ行列は一度も作られていない。演習問題 3 では、本節で実装した `Dual` クラスと `Var` クラスから同じヘッセ行列・ベクトル積を直接構築し、フォワード・オーバー・リバースとリバース・オーバー・リバースを比較する。

34.5.3.3 陰関数微分

興味の対象となるすべての量が陽な公式で与えられるとは限らない。漸化式の不動点、物理系の平衡状態、内側にある最適化問題の最小点など、多くの量は方程式の解として陰に定義され、反復ソルバによって計算される。20 回のニュートン反復をテープに記録し、そのすべてを通して逆伝播することも可能ではあるが、そうするとすべての中間反復点を保存しなければならず、しかもソルバがたまたまたどった特定の軌跡を微分することになる。定義方程式だけから直接導関数を求めることはできないだろうか。

考えられる最も単純な場合、すなわち未知数 1 つの方程式 1 本 $g(x, \theta) = x^2 - \theta = 0$（解は $x^\star = \sqrt{\theta}$）から始めよう。閉じた形の解を微分する代わりに、恒等式 $x^\star(\theta)^2 - \theta \equiv 0$ を θ に関して直接微分する。すると $2x^\star\, dx^\star/d\theta - 1 = 0$ が得られ、これを導関数について解くと次式になる。

$$\frac{dx^\star}{d\theta} = \frac{1}{2x^\star} = \frac{1}{2\sqrt{\theta}},$$

これは確かに $\sqrt{\theta}$ の導関数になっている（値は 0.25、ただし $\theta = 4$ のとき）。決定的に重要なのは、$x^\star$ の陽な公式を一切使わず、方程式 $g(x, \theta) = 0$ と解そのものの値だけを使ったという点である。

一般の結果は**陰関数定理**（implicit function theorem）(Rudin 1976) であり、ここでは証明なしで述べる。$\mathbf{g}(\mathbf{x}, \theta)$ を $\mathbb{R}^d \times \mathbb{R}^p$ から $\mathbb{R}^d$ への写像とし、$\mathbf{g}$ が解 $(\mathbf{x}^\star, \theta)$（$\mathbf{g} = \mathbf{0}$ の解）の近傍で連続微分可能であり、ヤコビ行列 $\partial \mathbf{g}/\partial \mathbf{x}$ がそこで正則であると仮定する。このとき、θ のある近傍において、解は微分可能な関数 $\mathbf{x}^\star(\theta)$ となる。恒等式 $\mathbf{g}(\mathbf{x}^\star(\theta), \theta) \equiv \mathbf{0}$ に連鎖律を適用すると $\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^\star}{\partial \theta} + \frac{\partial \mathbf{g}}{\partial \theta} = \mathbf{0}$ が得られ、したがって次式が成り立つ。

$$\frac{\partial \mathbf{x}^\star}{\partial \theta} = -\Big(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big)^{-1}\frac{\partial \mathbf{g}}{\partial \theta}. \tag{34.51}$$

右辺の 2 つのヤコビ行列は、どちらも解 $(\mathbf{x}^\star, \theta)$ においてのみ評価される。ソルバが何回の反復を要したかにかかわらず、導関数行列全体の計算コストはパラメータ 1 つあたり $\partial \mathbf{g}/\partial \mathbf{x}$ を係数行列とする連立一次方程式を 1 回解くだけで済む。さらに良いことに、VJP の形にすれば、すべてのパラメータを合わせても連立一次方程式をわずか 1 回解くだけで計算できる。すなわち $\mathbf{u}^\top \partial \mathbf{x}^\star/\partial\theta = -\mathbf{a}^\top \partial \mathbf{g}/\partial\theta$ となり、ここで随伴ベクトル $\mathbf{a}$ は $(\partial \mathbf{g}/\partial \mathbf{x})^\top \mathbf{a} = \mathbf{u}$ の解である。

これと同じ原理が、機械学習におけるいくつもの主要な手法を支えている。ODE ソルバを通して微分する随伴法（adjoint method）（38.3.2 節）、深層平衡モデル（deep equilibrium models）(Bai et al. 2019)（層全体を不動点 $\mathbf{z}^\star = \mathbf{f}(\mathbf{z}^\star, \theta)$ として定義し、Equation 34.51 によって訓練するモデル）、そして内側の最小化問題を通じた勾配ベースのハイパーパラメータ最適化などである。

Equation 34.51 を非線形な 2×2 の連立方程式 $g_1 = x_1^2 + \theta x_2 - 2 = 0$、$g_2 = x_1 x_2 + \theta - 1 = 0$（20 回のニュートンステップで解く）でテストし、陰関数微分による勾配をソルバ出力の中心差分と比較して確かめてみよう。

```
# Implicit gradient dx*/dtheta = -(dg/dx)^{-1} dg/dtheta on a 2x2 system,
# checked against a finite difference of the solver output
def g(x, th):                          # g(x, theta) = 0 defines x*(theta)
    return np.array([x[0]**2 + th * x[1] - 2, x[0] * x[1] + th - 1])

def dg_dx(x, th):                      # Jacobian in x, evaluated at (x, th)
    return np.array([[2 * x[0], th], [x[1], x[0]]])

def solve(th):                         # Newton iteration, from a fixed start
    x = np.array([1.0, 1.0])
    for _ in range(20):
        x = x - np.linalg.solve(dg_dx(x, th), g(x, th))
    return x

th, h = 0.3, 1e-6
xs = solve(th)
dg_dth = np.array([xs[1], 1.0])        # dg/dtheta at the solution
implicit = -np.linalg.solve(dg_dx(xs, th), dg_dth)
fd = (solve(th + h) - solve(th - h)) / (2 * h)
print('implicit-function gradient:', implicit.round(8))
print('finite-difference check   :', fd.round(8))
```

```
implicit-function gradient: [-0.11311451 -0.69322129]
finite-difference check   : [-0.11311451 -0.69322129]
```

2 つのベクトルは有限差分の精度の限界まで一致しており、しかも陰関数の公式には 20 回のニュートン反復が一切現れていない。導関数は $\mathbf{g}$ と解 $\mathbf{x}^\star$ のみに依存するため、同じ根に収束するソルバであればどれを使っても同じ勾配が得られる。

他のあらゆる定理と同様に、陰関数定理にも 2 つの明確な適用限界があり、どちらも実務で容易に観察できる。第 1 に、解において $\partial \mathbf{g}/\partial \mathbf{x}$ が正則でなければならない。スカラーの例 $x^2 = \theta$ では、$\theta = 0$ のとき $\partial g/\partial x = 2x^\star = 0$ となるため公式はゼロ除算を起こすが、これは原点において $\sqrt{\theta}$ の接線が垂直になることを反映している。第 2 に、この公式はソルバが実際に収束してい

ることを前提としている。収束していないときに何が起きるかを見るため、$x^2 = 4$ に対して初期値 $x = 1$ からニュートン法を実行し、$k = 1, 2, 3$ ステップで早期終了させて、陰関数の公式 $1/(2\tilde{x}_k)$（未収束の反復点 $\tilde{x}_k$ で評価したもの）と、k ステップ展開したソルバ自体の導関数（θ に関する中心差分で推定）とを比較してみよう。

```
def newton(theta, k, x=1.0):                        # k Newton steps on x^2 = theta
    for _ in range(k):
        x = x - (x * x - theta) / (2 * x)
    return x

theta, h = 4.0, 1e-6
for k in (1, 2, 3):
    x_k = newton(theta, k)
    unrolled = (newton(theta + h, k) - newton(theta - h, k)) / (2 * h)
    print(f'k={k}: x = {x_k:.5f}, formula {1 / (2 * x_k):.4f}, '
          f'unrolled {unrolled:.4f}, exact 0.25')
```

```
k=1: x = 2.50000, formula 0.2000, unrolled 0.5000, exact 0.25
k=2: x = 2.05000, formula 0.2439, unrolled 0.2900, exact 0.25
k=3: x = 2.00061, formula 0.2499, unrolled 0.2509, exact 0.25
```

1、2、3 ステップ後、ニュートン法の反復点はそれぞれ 2.5、2.05、2.00061 に達する。真の導関数の値 0.25 に対し、陰関数の公式は 0.2000、0.2439、0.2499 を返し、展開したソルバの微分値は 0.5000、0.2900、0.2509 となる。収束前はどちらも不正確である。陰関数の公式はソルバの残差誤差をそのまま引き継ぎ、打ち切られた反復経路を微分する展開微分はこの例ではさらに大きく外れている。対処法は明快で、まずソルバを十分に収束させればよい。

1 つだけ根本的な問題が未解決のまま残っている。自動微分は積 $\mathbf{J}_K \cdots \mathbf{J}_1$ を浮動小数点の丸めの範囲で正確に計算するが、その積の値自体を変えることはできない。多数の層を経るうちに、ヤコビ行列の長い積は依然として幾何級数的に縮小したり爆発したりし得る（34.4 節）。どれほど厳密に求めたところで、10^{-20} という勾配が訓練において 0 と同然であることに変わりはない。勾配消失や勾配爆発を抑えるには、注意深い初期化とアーキテクチャの設計が必要であり、これについては 4.4 節以降で取り上げる。

34.5.4 まとめ

プログラムの導関数はその基本ステップのヤコビ行列の積であり、自動微分はその積をシードベクトルに対して評価することで、すべてのステップを行列・行列積ではなく行列・ベクトル積として計算する。フォワードモードは二重数上でプログラムを実行して 1 パスあたり 1 つのヤコビ行列・ベクトル積（1 つの入力方向）を計算し、リバースモードは順伝播の間にテープを記録してそれを逆向きに再生することで 1 パスあたり 1 つのベクトル・ヤコビ行列積（1 つの出力方向）を計算する。どちらのモードが低コストになるかはヤコビ行列の形状によって決まる。スカラー損失の出力はただ 1 つであるため、1 回の逆方向スイープで全パラメータの勾配が、損失評価のおよそ 3 倍のコストで得られる。関数 $(\sum_j x_j^2)^2$（入力数 $n = 200$）では、リバースモードの演算回数が 802 回であったのに対し、フォワードモードは $80{,}200$ 回を要した。テープのメモリ使用量は計算長 T に比例して増大するが、勾配チェックポイントを用いれば順伝播 1 回分の追加計算と引き換えにピークメモリを $2\sqrt{T}$ に抑えられる。最後に、リバースモードの上にフォワードモードを合成すれば勾配評価 1 回分の小さな定数倍のコストでヘッセ行列・ベクトル積を計算でき、陰関数定理を用いれば収束した解において連立一次方程式を 1 回解くだけでソルバの出力を微分できる。

34.5.5 演習問題

1. [code] **二重数ステップと複素ステップ。** $\mathbf{f}(x,y)=(x+\sin y,\; y+x^2/2)$ の $\mathbf{x}_0=(0.5,0.5)$ におけるヤコビ行列の第 2 列 $\mathbf{J}\mathbf{e}_2=(\cos 0.5,\;1)$ を考える。中心差分 $(\mathbf{f}(\mathbf{x}_0+t\mathbf{e}_2)-\mathbf{f}(\mathbf{x}_0-t\mathbf{e}_2))/(2t)$ は、打ち切りによる t^2 オーダーの誤差と丸めによる u/t オーダーの誤差を併せ持つ（ここで $u\approx 10^{-16}$ は float64 の丸めの単位である）。

   1. 最適なステップサイズと最小誤差を予測し、その予測を $t=10^{-1},10^{-2},\dots,10^{-12}$ に対して確かめよ（およそ 2×10^{-11} の最小誤差が $t=10^{-5}$ 付近で得られるはずである）。さらに、`Dual` に $\mathbf{e}_2$ をシードとして与えた 1 回のパスと比較せよ。なぜ二重数の計算にはステップサイズが現れないのか。
   2. (a) と同様の実験を第 1 列 $\mathbf{J}\mathbf{e}_1$ について行い、そこでは任意の t に対して中心差分が丸め誤差を除いて厳密になる理由を説明せよ（ヒント：$\mathbf{f}$ は x に関する二次多項式である）。
   3. ε を虚数単位 i に置き換えると、複素ステップ推定値 $\operatorname{Im}\mathbf{f}(\mathbf{x}_0+it\mathbf{e}_2)/t$ が得られる (Squire and Trapp 1998)。これは本問の関数のように複素引数へ解析接続できる $\mathbf{f}$ に対して有効である。$\mathbf{f}(\mathbf{x}_0+it\mathbf{e}_2)$ を t の三次まで展開し、この推定値の誤差が $O(t^2)$ であることを示せ。なぜこの推定値ではほぼ等しい数値どうしの減算が生じないのか。
   4. NumPy の複素数演算を用いて $t=10^{-1},10^{-2},\dots,10^{-20}$ に対する複素ステップ推定値を計算し、その誤差を (a) の結果と比較せよ。$\varepsilon^2=0$ と $i^2=-1$ のどのような違いが、複素ステップにおける $O(t^2)$ の誤差を生むのか。

2. [code] **ハイパーパラメータ勾配。** 特徴量ごとに 1 つのペナルティ $\lambda_j>0$ を持つリッジ回帰は、次の最適化問題を解く。

$$\mathbf{w}^\star(\boldsymbol{\lambda})=\operatorname*{argmin}_{\mathbf{w}}\;\|\mathbf{X}\mathbf{w}-\mathbf{y}\|^2+\sum_{j=1}^{d}\lambda_j w_j^2$$

   ここで $(\mathbf{X},\mathbf{y})$ は訓練データであり、データ $(\mathbf{X}_{\mathrm{v}},\mathbf{y}_{\mathrm{v}})$ 上での検証損失は $V(\boldsymbol{\lambda})=\|\mathbf{X}_{\mathrm{v}}\mathbf{w}^\star(\boldsymbol{\lambda})-\mathbf{y}_{\mathrm{v}}\|^2$ である。

   1. $\mathbf{w}^\star$ が $\mathbf{g}(\mathbf{w},\boldsymbol{\lambda})=(\mathbf{X}^\top\mathbf{X}+\operatorname{diag}(\boldsymbol{\lambda}))\,\mathbf{w}-\mathbf{X}^\top\mathbf{y}=\mathbf{0}$ の解であることを示し、Equation 34.51 を用いて $\partial\mathbf{w}^\star/\partial\boldsymbol{\lambda}$ を導出せよ。
   2. $\nabla_{\boldsymbol{\lambda}}V=-\mathbf{a}\odot\mathbf{w}^\star$ となることを示せ。ただし $\mathbf{a}$ は行列 $\mathbf{X}^\top\mathbf{X}+\operatorname{diag}(\boldsymbol{\lambda})$ を係数とする 1 つの連立一次方程式の解である。その右辺を書き下せ。また、$\partial\mathbf{w}^\star/\partial\boldsymbol{\lambda}$ を経由するフォワード方式では d 個のペナルティに対して連立一次方程式を何回解く必要があり、この随伴方式では何回で済むか。
   3. V と随伴勾配を NumPy で実装し、訓練例 50 個、検証例 50 個、特徴量 $d=10$ 個のランダムなデータで評価せよ。求めた勾配を V の中心差分と比較して検証せよ。

   Alan Edelman および Steven G. Johnson、Matrix Calculus for Machine Learning and Beyond, problem set 2（MIT 18.S096、IAP 2023、問題 1）より改変。

3. [code] **ヘッセ行列・ベクトル積。●** 関数 $f(\mathbf{x})=\left(\sum_j x_j^2\right)^2$（34.5.2.4 節で入力 `xs0`、$n=200$ に対して評価したもの）は加算と乗算のみから構成されているため、`Dual` と `Var` の両方で評価できる。

   1. ∇f とヘッセ行列 $\mathbf{H}$ を f に対して導出し、$\mathbf{H}\mathbf{v}$ を $\mathbf{H}$ を一度も構築しない形で表せ。
   2. ランダムな方向 $\mathbf{v}$ を生成し、各入力 x_j を `Var(Dual(x_j, v_j))` として包んで `backprop` を実行せよ。得られた随伴変数の値パートが ∇f をなし、ε 成分が $\mathbf{H}\mathbf{v}$ をなすことを示し、その両方を (a) の結果および 34.5.3.2 節のヘッセ行列・ベクトル積を同じ f に

適用した結果と比較せよ。逆伝播クロージャのどの演算が二重数の算術で実行され、なぜそれによって Equation 34.49 の意味でのフォワード・オーバー・リバース計算になるのか。

3. 今度はテープ自体を入れ子にせよ。各入力を `Var(Var(x_j))` として包んで backprop を実行すると、随伴変数は `Var` オブジェクトになる。それらから $\sum_j (\nabla f)_j v_j$ を作り、それに対して backprop を実行することで、リバース・オーバー・リバースによって $\mathbf{H}\mathbf{v}$ を求めよ。たとえば `Var` オブジェクトの生成数を `Var.__init__` 内のカウンタで数え、両手法の生成数を通常の勾配計算 1 回分と比較せよ。逆伝播を記録しているのはどちらの手法であり、そのメモリコストはどうなるか。

4. [code] **テープ上のチェックポイント保存。**●$x_k = x_{k-1} - 0.01\, x_{k-1}^2$（$k = 1, \dots, T$、ただし $T = 400$、$x_0 = 1$）とおき、x_T を x_0 に関して微分せよ。`Var` には減算がないため、各ステップを `x + (-0.01) * x * x` として記録すること。

   1. 通常の backprop で連鎖全体をテープに記録し、逆方向スイープが始まる時点でテープが保持しているノード数を数えよ。
   2. backprop は出力の随伴変数にシードとして 1 を与えている。これを任意のシードを受け取れるように一般化せよ。区間 $x_i \mapsto x_j$ を新しい葉ノード x_i から記録し、一般化した backprop を x_j からシード $\partial x_T / \partial x_j$ で実行すると、葉ノードの随伴変数に $\partial x_T / \partial x_i$ が返されることを示せ。
   3. Figure 34.26 のスケジュールを s ステップの区間で実装せよ。すなわち、通常の浮動小数点数による順伝播で $x_0, x_s, x_{2s}, \dots$ を保存し、続く逆伝播では最後の区間から最初の区間へ 1 区間ずつ再記録して (b) を適用する。得られた結果が (a) の勾配と一致することを確かめよ。
   4. $s \in \{1, 5, 10, 20, 100, 400\}$ について、保存されるチェックポイント数、ステップ数で測った最大のテープ長、および最初の順伝播を含めたステップ $x \mapsto x - 0.01x^2$ の評価回数を報告せよ。メモリ使用量 $T/s + s$（Equation 34.48）、すなわち $401, 85, 50, 40, 104, 401$ と、その最小点 $s = \sqrt{T} = 20$ を確認せよ。さらに $T = 100$ および $T = 1600$ についても、条件 $s \le 800$ を満たす T の約数にわたって同様の実験を繰り返せ（1600 ステップを単一区間とすると backprop で Python のデフォルトの再帰上限を超える）。メモリを最小化する s は T に対してどのようにスケールし、最適なスケジュールは何回のステップ評価を必要とするか。

5. [extended] **平衡層。**●●深層平衡層 (Bai et al. 2019) は次の不動点を出力する。

$$\mathbf{z}^\star = \tanh(\mathbf{W}\mathbf{z}^\star + \mathbf{c}), \tag{34.52}$$

ここで $\mathbf{c}$ は層への入力に依存するベクトルである。損失を $\ell = \tfrac{1}{2}\|\mathbf{z}^\star - \mathbf{y}\|^2$ とし、$\mathbf{D} = \operatorname{diag}\big(1 - (z_1^\star)^2, \dots, 1 - (z_d^\star)^2\big)$ を不動点における tanh の導関数とする。

   1. 写像 $\mathbf{z} \mapsto \tanh(\mathbf{W}\mathbf{z} + \mathbf{c})$ が $\|\mathbf{W}\|_2 < 1$ のとき縮小写像となり、したがって不動点が存在して一意であること、さらにそのとき $\mathbf{I} - \mathbf{D}\mathbf{W}$ が正則になることを示せ。また、Equation 34.51 を用いて $\nabla_{\mathbf{W}} \ell$ を導出し、それが $(\mathbf{D}\mathbf{a})(\mathbf{z}^\star)^\top$（ただし随伴変数 $\mathbf{a}$ は $(\mathbf{I} - \mathbf{D}\mathbf{W})^\top \mathbf{a} = \mathbf{z}^\star - \mathbf{y}$ の解）に等しいことを示せ。
   2. $d = 20$ とし、標準正規分布に従う要素を持つ $\mathbf{G} \in \mathbb{R}^{d \times d}$ から $\mathbf{S} = \mathbf{G} + \mathbf{G}^\top$、$\mathbf{W} = \rho\, \mathbf{S} / \|\mathbf{S}\|_2$（$\rho \in \{0.5, 0.9, 0.99\}$）を作り、$\mathbf{c}$ と $\mathbf{y}$ の各要素を独立に $\mathcal{N}(0, 0.1^2)$ から生成せよ。NumPy を用い、不動点反復によって $\mathbf{z}^\star$ を求め、連立一次方程式を 1 回解くことで (a) の勾配を計算せよ。そのいくつかの要素を ℓ の中心差分と比較して検証せよ。
   3. 展開による微分は、K 回の反復を $\mathbf{z}_0 = \mathbf{0}$ から実行し、そのすべてを通して逆伝播する。1 反復あたり 1 回の VJP として、この逆方向スイープを手作業で実装せよ。K に

対して、(b) の勾配を基準とした展開勾配の誤差を、順伝播の誤差 $\|\mathbf{z}_K - \mathbf{z}^\star\|$ とともにプロットせよ。両誤差の減衰率を **DW** のスペクトル半径と比較せよ。K が大きくなるにつれて 2 つの誤差の比はどのように振る舞うか、またそれはなぜか。各勾配計算のメモリ使用量は K に対してどのようにスケールするか。

4. 連立一次方程式の求解を、反復 m ステップによる更新 $\mathbf{a} \leftarrow (\mathbf{z}^\star - \mathbf{y}) + (\mathbf{DW})^\top \mathbf{a}$（初期値 $\mathbf{a} = \mathbf{0}$）に置き換えよ。これは打ち切りノイマン級数であり、Geng et al. (2021) が**ファントム勾配**として用いているものである。この打ち切り随伴変数が、反復 m 回（初期値 $\mathbf{z}^\star$）を通した誤差逆伝播法と同じ勾配を与えることを示せ。打ち切り勾配と厳密な勾配のなす角の余弦（コサイン）を、m に対して各 ρ ごとにプロットせよ。$m = 1$ のときの勾配は連立一次方程式を解く必要がなく、Wu Fung et al. (2022) のヤコビ行列フリー誤差逆伝播法に一致する。これは依然として ℓ の降下方向になっているか。

6. [code] **1 回の順伝播による二階導関数**。ε を $\varepsilon^3 = 0$（ただし $\varepsilon^2 \neq 0$）を満たす元として添加し、$a + b\varepsilon + c\varepsilon^2$ を 3 つ組 (a, b, c) と書く。

   1. 積 $(a,b,c)(a',b',c') = (aa',\ ab' + ba',\ ac' + bb' + ca')$ を導出し、解析関数 φ に対して $\varphi(a + b\varepsilon + c\varepsilon^2) = \varphi(a) + \varphi'(a)\,b\,\varepsilon + \big(\varphi'(a)\,c + \tfrac{1}{2}\varphi''(a)\,b^2\big)\varepsilon^2$ が成り立つことを示せ。
   2. `Dual2` に `__add__`、`__mul__`、`sin`、`exp` を実装し、$f(x) = \sin(x^2) + e^x$ を $(1.3, 1, 0)$ 上で評価せよ。結果が $(f(1.3),\ f'(1.3),\ \tfrac{1}{2}f''(1.3))$ になることを示し、$f''(1.3) \approx -3.2806$ を手計算の公式 $f''(x) = 2\cos x^2 - 4x^2 \sin x^2 + e^x$ と比較して確かめよ。
   3. $\ell : \mathbb{R}^n \to \mathbb{R}$ の入力に方向 $\mathbf{v}$ をシードとして与えたとき、ε^2 成分は何を返すか（答えはスカラー値 $\tfrac{1}{2}\mathbf{v}^\top \mathbf{H}\mathbf{v}$ である）。偏極恒等式によってベクトル $\mathbf{Hv}$ を得るにはこのようなパスが何回必要か、1 回のフォワード・オーバー・リバースのパスと対比して答えよ。
   4. ε を 3×3 行列（ただし $\varepsilon^3 = \mathbf{0}$）として表し、(a) の積を行列積によって検証せよ。

34.6 積分法

これまでの章では、連続確率変数の確率がどのように計算されるかを問うことなく、それを用いてきた。正規分布をその確率密度によって記述し（2.1.3 節）、期待損失をその密度に関する平均として表し、どちらも所与のものとして扱ってきたのである。本節では、ある区間にわたって連続的に広がる量をどのように足し合わせるかを問う。

おなじみの釣鐘曲線である標準正規分布は、連続的な和になぜ注意が必要なのかをよく示している。標準正規確率変数 Z は密度 $p(z) = e^{-z^2/2}/\sqrt{2\pi}$ をもつ。$z = 0$ において曲線の高さは $p(0) \approx 0.40$ に達するが、ちょうどゼロをとる確率は $P(Z = 0) = 0$ であり、他のどの 1 点についても同様である。その一方で、区間は確かに確率をもつ。たとえば $P(-1 \le Z \le 1) \approx 0.68$ である。幾何学的には、この数値は p の下の -1 から 1 までの面積であり、$1/\sqrt{2\pi}$ は曲線の下の全体の面積を 1 にする定数である。では、$\sqrt{2\pi}$ はどこから現れ、0.68 はどのように計算するのだろうか。

こうした問いは釣鐘曲線にとどまらない。連続変数に関するあらゆる確率や期待損失は、この種の面積である。変分オートエンコーダ（variational autoencoder、VAE）や方策勾配法のようにランダムサンプリングを通じてモデルを訓練する場合には、このような面積をモデルパラメータに関して微分することも必要になる。

方針は明快である。領域を細いスライスに切り分け、それぞれの面積を足し合わせ、スライスの幅をゼロへと縮める。その極限が積分である。続いて微分積分学の基本定理（fundamental theorem of calculus）が、この無限の足し算を微分の逆演算へと置き換え、大半の積分を 1 行

で解決してくれる。ただし釣鐘曲線はこれだけでは解決せず、その正規化定数（normalizing constant）である $\sqrt{2\pi}$ を求めるには 2 次元を経由する回り道が必要になる。本節の軸となるのは、基本定理 Equation 34.55、多変数の変数変換（change of variables）の公式 Equation 34.63、そしてガウス積分（Gaussian integral）Equation 34.65 の 3 つの結果である。大学初年次の微分積分をすでに身につけている読者は、34.6.2.2 節まで流し読みしてよい。パラメータに関する積分の微分はこれらすべての道具の上に成り立っており、34.7 節で独立した節を設けて扱う。測度論とルベーグ積分（Lebesgue integral）は割愛し、関心のある読者には 34.8 節を案内する。

34.6.1 定積分

34.6.1.1 リーマン和

確率密度は単位長さあたりの確率を与えるため、幅 ϵ の短い区間における確率は、おおよそ密度と幅の積になる。より長い区間では、それを細いスライスに切り分け、各スライスで高さに幅を掛けて結果を足し合わせる。この考え方に密度特有の部分は何もなく、同じ手順が任意の関数に対して通用する。

まずは手計算で確かめられる例から始めよう。$f(x) = x^2$ を区間 $[0, 1]$ 上で考え、この区間を N 個の等しい幅 $\epsilon = 1/N$ のスライスに分割する。i 番目のスライス $[i/N, (i+1)/N]$ では、左端点における f の値 $(i/N)^2$ を高さとする長方形で面積を近似する。N 個の長方形をすべて足し合わせると、次式が得られる。

$$\sum_{i=0}^{N-1} \frac{1}{N}\left(\frac{i}{N}\right)^2 = \frac{1}{N^3}\sum_{i=0}^{N-1} i^2 = \frac{(N-1)N(2N-1)}{6N^3} = \frac{1}{3} - \frac{1}{2N} + \frac{1}{6N^2},$$

ここで途中の変形には古典的な恒等式 $\sum_{i=0}^{N-1} i^2 = (N-1)N(2N-1)/6$ を用いた。N が大きくなるにつれて長方形の和は $\frac{1}{3}$ に収束し、極限とのずれはおよそ $\frac{1}{2N}$、すなわちスライス幅 ϵ に関して 1 次の誤差となる。

補助関数 `left_sum` は、任意の関数 f に対してこの**左端点則**（left-endpoint rule）を実装したものである。すなわち、$[a, b]$ を幅 ϵ のスライスに分割し、各スライスの左端で f を評価する（Figure 34.27）。本節のすべてのグリッド和には `left_sum` を用いることとし、まずは x^2 で手早く確認してみよう。

```
def left_sum(f, a, b, eps):
    """Left-endpoint Riemann sum of f on [a, b] with slices of width eps."""
    x = a + eps * np.arange(round((b - a) / eps))   # left ends of the slices
    return float(np.sum(eps * f(x)))

for N in [10, 100, 1000]:
    print(f'N={N:<5} left sum={left_sum(lambda x: x**2, 0, 1, 1 / N):.6f}  '
          f'formula={1/3 - 1/(2*N) + 1/(6*N**2):.6f}')
```

```
N=10    left sum=0.285000  formula=0.285000
N=100   left sum=0.328350  formula=0.328350
N=1000  left sum=0.332834  formula=0.332834
```

数値計算による和は、表示されたすべての桁で閉形式の公式と一致している。

定義：この構成において、スライスの幅を等しくする必要も左端点を使う必要もない。より一般に、**分割**（partition）$P : a = x_0 < x_1 < \cdots < x_N = b$ は $[a, b]$ を部分区間へと切り分け、**代**

表点（tag）$\xi_i \in [x_{i-1}, x_i]$ は各スライスの内部から 1 つのサンプル点を選び、**分割の幅**（mesh）$\|P\| = \max_i (x_i - x_{i-1})$ は最も広いスライスの幅を測る。有界関数 f が $[a,b]$ 上で**リーマン積分可能**（Riemann integrable）であるとは、代表点の選び方によらず、分割の幅をゼロへと縮めたときにすべての代表点付きリーマン和が近づくただ 1 つの数値 I が存在することをいう。

$$\int_a^b f(x)\,dx = I = \lim_{\|P\| \to 0} \sum_{i=1}^{N} f(\xi_i)(x_i - x_{i-1}). \tag{34.53}$$

要するに、定積分とはスライスを限りなく細くしたときのスライス和の共通の極限である。先ほどの手計算により、$\int_0^1 x^2\,dx = \tfrac{1}{3}$ が確立されたことになる。$[a,b]$ 上のすべての連続関数はリーマン積分可能であり、有限個の跳躍不連続点をもつすべての有界関数も同様である (Rudin 1976)。有理数全体の指示関数のような一部の病的な有界関数はリーマン積分をもたないが、ルベーグ積分（34.8 節）はそのような関数へと積分を拡張しつつ、リーマン積分が存在するときは常にそれと一致する。

定義から直ちに 2 つの初等的な性質が導かれ、以下の証明の原動力となる。第 1 に、積分は**単調**（monotone）である。$f \le g$ が $[a,b]$ 上のいたるところで成り立つならば、f のどの代表点付きリーマン和も対応する g の代表点付きリーマン和以下となり、極限をとっても不等号が保たれるため $\int_a^b f\,dx \le \int_a^b g\,dx$ が得られる。第 2 に、積分は**部分区間に関して加法的**（additive over subintervals）である。任意の $a \le b \le c$ に対して $\int_a^c f\,dx = \int_a^b f\,dx + \int_b^c f\,dx$ が成り立つ。なぜなら、点 b を含む任意の分割は $[a,b]$ の分割と $[b,c]$ の分割へと綺麗に二分されるからである。

通し例：ガウス積分に到達するまで、本節を通して 1 つの具体例を扱う。

$$\int_0^2 \frac{x}{1+x^2}\,dx = \tfrac{1}{2}\log 5 \approx 0.80472.$$

この値がなぜ $\tfrac{1}{2}\log 5$ になるかは、評価公式（evaluation rule、Equation 34.56）を確立した時点で明らかになる。Figure 34.27 に、3 通りのスライス幅におけるこの曲線の下の左端点長方形を示す。

左端点和はどれほどの速さで $\tfrac{1}{2}\log 5$ に収束するだろうか。x^2 の $[0,1]$ 上での誤差は $\frac{1}{2N} = \frac{\epsilon}{2}$ であった。1 次収束が一般的であるならば、スライスを 10 倍細くすると誤差も 10 分の 1 に縮まるはずである。誤差を $0.2\,\epsilon$ と並べて出力し、確かめてみよう。

```
f = lambda x: x / (1 + x**2)
truth = math.log(5) / 2
for eps in [1e-1, 1e-2, 1e-3, 1e-4]:
    s = left_sum(f, 0, 2, eps)
    print(f'eps={eps:.0e}  sum={s:.6f}  truth-sum={truth - s:.2e}  '
          f'0.2*eps={0.2 * eps:.0e}')
print(f'truth = (1/2) log 5 = {truth:.6f}')
```

```
eps=1e-01  sum=0.783785  truth-sum=2.09e-02  0.2*eps=2e-02
eps=1e-02  sum=0.802710  truth-sum=2.01e-03  0.2*eps=2e-03
eps=1e-03  sum=0.804519  truth-sum=2.00e-04  0.2*eps=2e-04
eps=1e-04  sum=0.804699  truth-sum=2.00e-05  0.2*eps=2e-05
truth = (1/2) log 5 = 0.804719
```

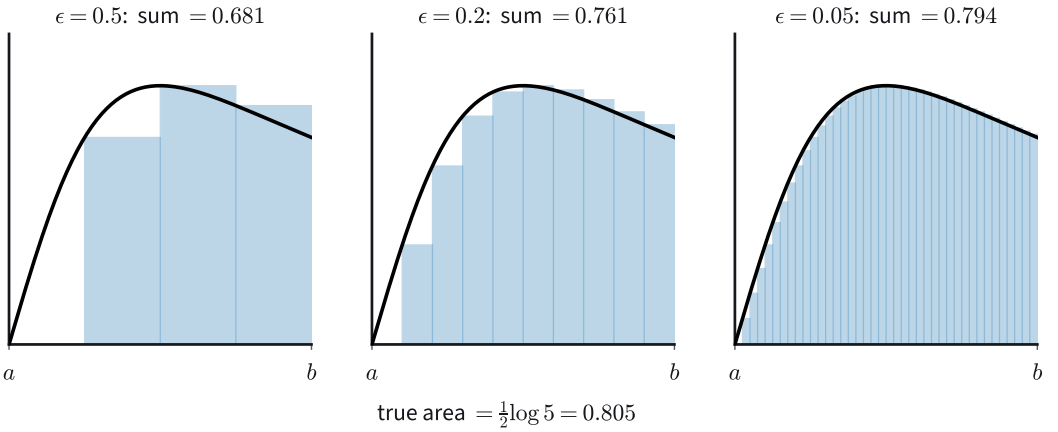


Figure 34.27: $f(x) = x/(1+x^2)$ の区間 $[0, 2]$ 上におけるスライス幅 $\epsilon = 0.5$、0.2、0.05 の左端点長方形。各スライスは左端点における f の高さをとり、長方形の和は面積 $\frac{1}{2}\log 5$ に近づく（$\epsilon \to 0$ のとき）。

ϵ が 10 分の 1 に縮まるたびに誤差も 10 分の 1 に縮まり、どの和も真の値を下回っている。定数 0.2 はちょうど $f(2) - f(0) = 0.4$ の半分になっている。これは偶然ではなく、基本定理の準備が整い次第、その背後にある公式を導出する。

34.6.1.2 符号付き面積

横軸より下にある長方形は負の高さ $f(\xi_i) < 0$ をもち、負の符号を伴ってリーマン和に加わる。したがって積分は**符号付き**（signed）の面積を測ることになり、$f < 0$ となる領域は $\int_0^1 (-1)\,dx = -1$ のように負の寄与をする。また規約により、右から左へ積分すると結果の符号も反転し、$\int_b^a f\,dx = -\int_a^b f\,dx$ となる。この規約によって、a、b、c の順序によらず加法性が成り立つことが保証される。Figure 34.28 に、1 本の曲線上でこれら両方の規約を示す。

34.6.2 微分積分学の基本定理

34.6.2.1 定理

スライスを足し合わせることで $\frac{1}{3}$ という x^2 の積分値が得られたのは、たまたま $\sum i^2$ が閉形式をもっていたからにすぎない。通し例では左端点和が $\sum_i \epsilon \cdot i\epsilon/(1 + i^2\epsilon^2)$ となり、これを簡約できる代数的な恒等式は存在しない。スライスを一切足し合わせずに積分を計算する方法が必要である。

突破口は、積分の上端を変数として扱うことから開ける。基点 a を固定し、**累積面積関数**（area-so-far function）を次式で定義しよう。

$$F(x) = \int_a^x f(y)\,dy, \tag{34.54}$$

この関数は、a から x まで累積された符号付き面積を記録する。このたった 1 つの関数に f のあらゆる定積分が符号化されている。加法性により、任意の区間 $[x_1, x_2]$ 上の面積は、単に 2 つの累積面積の差として表されるからである。

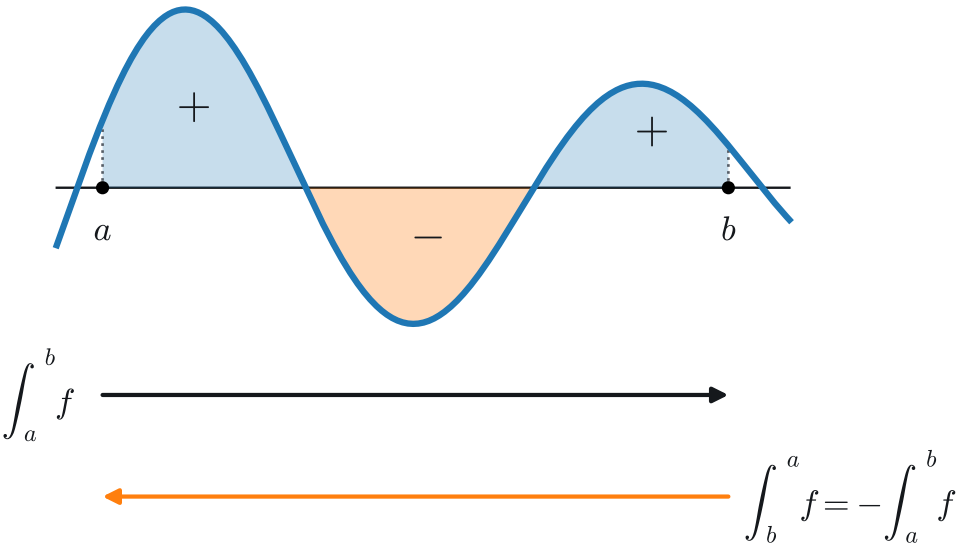


Figure 34.28: 符号付き面積。軸より上の膨らみは正、軸より下の膨らみは負として数えられるため、$\int_a^b f$ は網掛け領域を符号付きで合計する。積分区間の上下端を入れ替えると、同じ領域を右から左へたどることになり合計の符号が反転する。

$$\int_{x_1}^{x_2} f(x)\,dx = F(x_2) - F(x_1),$$

この関係は Figure 34.29 に示すとおりである。したがって F さえ見つかれば、問題全体が解決する。

F は、x をほんの少し右へ動かしたときにどう変化するだろうか。Figure 34.30 には、x までの累積面積と並んで、幅 ϵ の細い帯（$x+\epsilon$ まで動かすことで加わる部分）が示されている。この帯は高さ $f(x)$ の長方形にきわめて近いため、$F(x+\epsilon) - F(x) \approx \epsilon f(x)$ となる。両辺を ϵ で割ると、$F'(x) = f(x)$ であることが示唆される。

定理（微分積分学の基本定理）： **f を $[a,b]$ 上でリーマン積分可能な関数とし、$F(x) = \int_a^x f(y)\,dy$ とおき、さらに f が点 $x \in (a,b)$ で連続であるとする。このとき、F は x で微分可能であり、次式が成り立つ。**

$$\frac{dF}{dx}(x) = f(x). \tag{34.55}$$

言葉で言えば、面積が累積していく変化率は、その時点における曲線の高さに等しい。

証明： x を $\epsilon > 0$ だけわずかに動かし、新たに加わる面積に着目する。Equation 34.54 と加法性により、その変化量は 1 本の細い短冊となる。

$$F(x+\epsilon) - F(x) = \int_x^{x+\epsilon} f(y)\,dy.$$

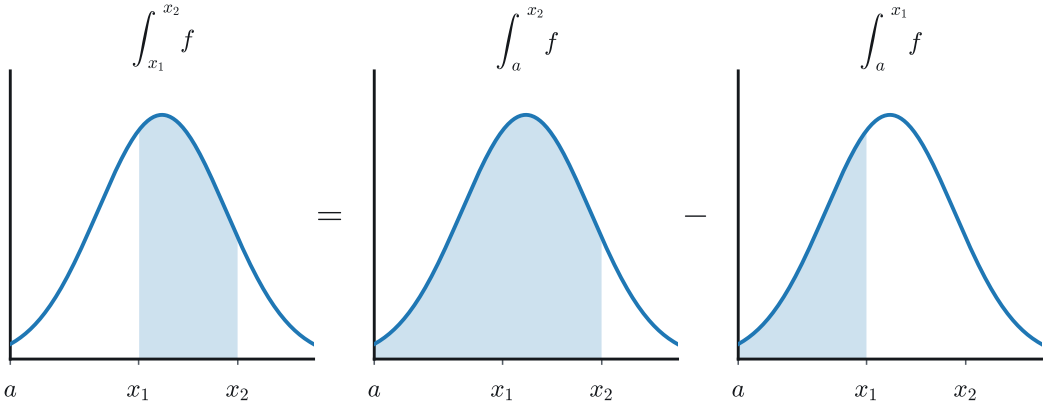


Figure 34.29: f の下の x_1 から x_2 までの面積は、図の左端にある基点 a から累積した 2 つの面積の差として表される（$\int_{x_1}^{x_2} f\,dx = \int_{a}^{x_2} f\,dx - \int_{a}^{x_1} f\,dx$）。

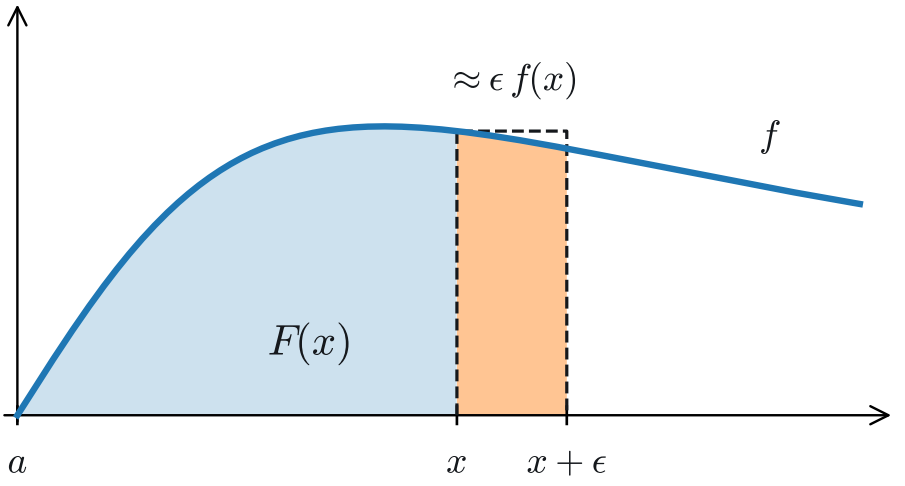


Figure 34.30: 累積面積 $F(x)$（$f(y) = y/(1+y^2)$ の下の $a = 0$ から $x = 1.2$ までの淡色部分）と、$x + \epsilon$ まで動かすことで加わる帯（濃色部分）。この帯は高さ $f(x)$ の破線の長方形に近いため、$F(x+\epsilon) - F(x) \approx \epsilon f(x)$ となる。

$m_\epsilon = \inf_{[x,x+\epsilon]} f$ と $M_\epsilon = \sup_{[x,x+\epsilon]} f$ を、この短冊上における f の最も厳しい定数の下界および上界とする。可積分関数は有界であるため、いずれも有限である。$m_\epsilon \le f \le M_\epsilon$ が $[x, x+\epsilon]$ 全体で成り立つため、積分の単調性により、短冊の面積は幅 ϵ の 2 つの長方形の間にはさまれる。

$$\epsilon\, m_\epsilon \;\le\; F(x+\epsilon) - F(x) \;\le\; \epsilon\, M_\epsilon.$$

ここで全体を ϵ で割る。f は x で連続であるため、任意の $\delta > 0$ に対し、区間 $[x, x+\epsilon]$ 全体はやがて $|f(y) - f(x)| < \delta$ となる領域の内部に収まる。したがって m_ϵ と M_ϵ はともに $f(x)$ へと収束し（$\epsilon \to 0$）、はさみうちの原理により $\lim_{\epsilon\to 0} \frac{F(x+\epsilon)-F(x)}{\epsilon} = f(x)$ が導かれる。$\epsilon < 0$ のときは短冊が x の左側に位置するが、差分商における 2 箇所の符号反転が打ち消し合うため同じ評価が成り立つ。この極限が $F'(x)$ である。■

この証明が、定義から読み取った単調性と加法性の 2 つの性質しか使っていないことに注目しよう。この定理により、面積を求める問題は**原始関数**（antiderivative）、すなわち関数 G であって $G' = f$ を満たすものを見つける問題へと帰着する。そのような関数は、34.1.2 節の導関数の表からそのまま読み取れることが多い。どの原始関数を用いても、次の**評価公式**によって定積分が得られる。**f が $[a,b]$ 上で連続であり、$G' = f$ が $[a,b]$ 全体で成り立つならば、次式が成り立つ。**

$$\int_a^b f(x)\,dx = G(b) - G(a). \tag{34.56}$$

なぜ F だけでなく、どの原始関数でもよいのだろうか。それは、F と G が同じ導関数 f を (a,b) 上で共有するため、その差 $G - F$ の導関数はゼロとなり、平均値の定理（34.1.3 節）によって定数でなければならないからである。その加法定数は $G(b) - G(a) = F(b) - F(a) = \int_a^b f\,dx$ において打ち消し合う。

「$[a,b]$ 全体で」という要件は不可欠である。$(-1/x)' = x^{-2}$ であるからといって盲目的に公式を適用すると、$\int_{-1}^{1} x^{-2}\,dx = [-1/x]_{-1}^{1} = -2$ となり、真に正の被積分関数に対して負の答えが出てしまう。ここで公式が適用できないのは、$G' = f$ が $x = 0$ で破綻しており、そこではどちらの関数も定義されていないからである。実際、この積分は発散する（34.6.2.2 節）。

逆向きに読めば、あらゆる微分公式が積分公式になる。$(y^3)' = 3y^2$ からはスライスを 1 つも切ることなく $\int_0^1 y^2\,dy = \frac{1}{3}$ が得られ、$(e^t)' = e^t$ からは $\int_0^x e^t\,dt = e^x - 1$ が得られる。通し例については、$G(x) = \frac{1}{2}\log(1+x^2)$ を試してみよう。連鎖律により $G'(x) = \frac{1}{2}\cdot\frac{2x}{1+x^2} = \frac{x}{1+x^2}$ となるため、次式が得られる。

$$\int_0^2 \frac{x}{1+x^2}\,dx = G(2) - G(0) = \tfrac{1}{2}\log 5 - \tfrac{1}{2}\log 1 = \tfrac{1}{2}\log 5,$$

これにより、先ほどのリーマン和が近づいていた極限が確かめられた。

左端点則の誤差：評価公式からは、左端点和の誤差に現れた定数 0.2 がどこから来たのかも説明できる。x_i から始まるスライス上で 1 次展開を行うと、$f(x) \approx f(x_i) + f'(x_i)(x - x_i)$ となる。$[x_i, x_i+\epsilon]$ 上で積分すると、このスライスの真の面積はその長方形 $\epsilon f(x_i)$ をおよそ $\frac{\epsilon^2}{2} f'(x_i)$ だけ上回ることがわかる。これら三角形の過剰分をすべてのスライスにわたって足し合わせると、$\frac{\epsilon}{2}$ と f' の左端点和の積になり、これは $\frac{\epsilon}{2}\int_a^b f'\,dx = \frac{\epsilon}{2}(f(b) - f(a))$ を近似する。したがって ϵ の 1 次の範囲で次式が成り立つ。

$$\text{left sum} - \int_a^b f(x)\,dx \approx \frac{\epsilon}{2}\big(f(a) - f(b)\big).$$

x^2 の $[0,1]$ 上の積分では、この式から $-\frac{\epsilon}{2} = -\frac{1}{2N}$ が得られ、手計算で求めた主要誤差項と一致する。通し例では $f(0) = 0$ と $f(2) = 0.4$ より $-0.2\,\epsilon$ となる。この評価は純粋な微分積分というより数値解析に属するが、本節の残り 3 つの実験に現れる残差を説明してくれるほか、36.5.1.7 節でランダムサンプリングを評価する際の基準線にもなる。

実は基本定理は $\int_0^2 f = G(2) - G(0)$ 以上のことを教えてくれる。累積面積関数 $F(x)$ は $G(x) - G(0)$ とすべての x において等しくなければならないのである。F を $[0,2]$ にわたる累積左端点和として計算し、すべてのグリッド点で $G(x) - G(0)$ と比較してこれを確かめてみよう。$[0,x]$ 上の左端点和はバイアス $\frac{\epsilon}{2}\big(f(0) - f(x)\big)$ を伴うため、両者は機械精度までは一致しない。このバイアスが最大になるのは f がピークをとる場所、すなわち $x = 1$（このとき $f(1) = \frac{1}{2}$）であるため、区間全体での最大誤差は $\frac{\epsilon}{4}$ になるはずである。

```
G = lambda x: np.log(1 + x**2) / 2            # antiderivative of f
for eps in [1e-2, 1e-3, 1e-4]:
    x = eps * np.arange(round(2 / eps))       # left ends of the slices of [0, 2]
    F = np.cumsum(eps * f(x))                 # F[k]: left sum on [0, x_k + eps]
    gap = np.abs(F - (G(x + eps) - G(0))).max()
    print(f'eps={eps:.0e}  max over x of |F(x) - (G(x) - G(0))| = {gap:.2e}')
```

```
eps=1e-02  max over x of |F(x) - (G(x) - G(0))| = 2.51e-03
eps=1e-03  max over x of |F(x) - (G(x) - G(0))| = 2.50e-04
eps=1e-04  max over x of |F(x) - (G(x) - G(0))| = 2.50e-05
```

どのスライス幅においても、最大誤差は予測どおり $\frac{\epsilon}{4}$ となっている。累積和は右端点 $x = 2$ だけでなく、区間全体にわたって原始関数を追跡している。

34.6.2.2 広義積分

正規分布や指数分布をはじめ多くの確率密度は非有界な定義域上にある一方、Equation 34.53 は有限区間 $[a,b]$ 上でのみ積分を定義している。そこで、積分を構成したときと同じく極限をとることで定義を拡張する。$[a,\infty)$ 上の**広義積分**（improper integral）は次式で定められる。

$$\int_a^{\infty} f(x)\,dx = \lim_{b\to\infty} \int_a^b f(x)\,dx, \tag{34.57}$$

これは b まで累積された面積の、大きな b における極限である。この極限が存在して有限であるとき、積分は**収束する**といい、そうでないときは**発散する**という。実数直線全体にわたって積分するには、$\int_{-\infty}^{\infty} f\,dx$ を任意の都合のよい点で 2 つの片側広義積分に分割し、その**両方**が独立に収束しなければならない。対称な極限だけでは不十分である。$\int_{-b}^{b} x\,dx = 0$ はすべての b に対して成り立つが、$\int_0^{\infty} x\,dx = \infty$ であるため、$\int_{-\infty}^{\infty} x\,dx$ は存在しない。

べき乗則 x^{-s}（指数 $s > 0$）は、裾が有限の面積を囲めるほど十分に速く減衰するかどうかを判定する標準的な基準である。$[1,b]$ 上で評価公式を適用し、$b \to \infty$ の極限をとると次式が得られる。

$$\int_1^b x^{-s}\,dx = \frac{b^{1-s}-1}{1-s} \xrightarrow{b\to\infty} \begin{cases} \dfrac{1}{s-1}, & s>1 \text{ (converges)},\\ \infty, & s\le 1 \text{ (diverges)}. \end{cases}$$

（$s=1$ では上の式は $0/0$ の形になるが、定積分は $\log b$ であり、これは $s\to 1$ としたときの公式の極限に一致し、$b\to\infty$ で発散する。）したがって $\int_1^\infty x^{-2}\,dx = 1$ である一方、$\int_1^\infty x^{-1}\,dx = \infty$ となり、面積が有限になるか無限になるかの境界線はちょうど $s=1$ にある。

べき乗則は典型的な**裾の重い**（heavy-tailed）分布である（36.1 節では、重い裾が平均にどう影響するかも調べる）。裾が $p(x)\propto x^{-s}$ に従う密度では、正規化定数が有限になるのは $s>1$ のときに限られ、また同じ判定基準を $x\cdot x^{-s}=x^{1-s}$ に適用して調べられる平均が有限になるのは $s>2$ のときに限られる。

端点で発散する被積分関数も、同じ極限の手法で扱える。すなわち $\int_0^1 f\,dx = \lim_{a\to 0^+}\int_a^1 f\,dx$ とする。べき乗則 x^{-s} に対しては、条件が逆転する。

$$\int_0^1 x^{-s}\,dx = \lim_{a\to 0^+}\frac{1-a^{1-s}}{1-s} = \frac{1}{1-s} \quad \text{for } s<1,$$

一方、$s\ge 1$ では積分は発散する。たとえば $s=\tfrac12$ では、$x^{-1/2}$ は 0 で無限大へと跳ね上がるが、その下の面積はわずか 2 である。無限遠の近くでは、べき乗則が有限の面積をもつには $1/x$ より速く減衰しなければならず、ゼロの近くでは $1/x$ より緩やかに増大しなければならない。

広義積分を数値的に近似するときには、2 種類の異なる誤差が入り込む。定義域を有限の b で打ち切ることによる誤差と、有限の幅 ϵ のスライスを足し合わせることによる誤差である。$\int_0^\infty e^{-x}\,dx = 1$ については、この 2 つを切り分けられる。以下では、厳密な部分積分値 $\int_0^b e^{-x}\,dx = 1-e^{-b}$（原始関数 $-e^{-x}$ から得られる）を、$\epsilon = 10^{-3}$ とした左端点和と並べて出力する。

```
for b in [1, 2, 5, 10, 20]:
    left = left_sum(lambda x: np.exp(-x), 0, b, 1e-3)
    print(f'b={b:<3}  left sum={left:.6f}  exact 1-e^-b={1 - math.exp(-b):.6f}')
```

```
b=1    left sum=0.632437  exact 1-e^-b=0.632121
b=2    left sum=0.865097  exact 1-e^-b=0.864665
b=5    left sum=0.993759  exact 1-e^-b=0.993262
b=10   left sum=1.000455  exact 1-e^-b=0.999955
b=20   left sum=1.000500  exact 1-e^-b=1.000000
```

厳密な部分積分値は急速に 1 へと収束する。b で打ち切ることによる**打ち切り誤差**（truncation error）は e^{-b} であり、$b=20$ の時点で小数第 6 位より小さくなっている。しかし左端点和は 1.0005 付近に落ち着き、1 にはならない。この残差が**離散化誤差**（discretization error）、すなわち先ほどの左端点則のバイアス $\frac{\epsilon}{2}\big(f(0)-f(b)\big)\to\frac{\epsilon}{2}=0.0005$ である。主要項のオーダーでは、2 つの誤差は単純に足し合わされる。b を大きくすれば打ち切り誤差が消え、ϵ を小さくすれば離散化誤差が消える。

34.6.2.3 部分積分

あらゆる微分法則は、評価公式を通して読むことで対応する積分法則を生み出す。積の微分法則は、部分積分（integration by parts）になる。

連続微分可能な関数 u と v に対し、$[a,b]$ 上で次式が成り立つ。

$$\int_a^b u(x)\,v'(x)\,dx = \Big[\,u(x)\,v(x)\,\Big]_a^b - \int_a^b u'(x)\,v(x)\,dx, \tag{34.58}$$

ここで $[\,h(x)\,]_a^b$ は $h(b)-h(a)$ の略記である。証明は 1 行で済む。積の微分法則により uv は $u'v+uv'$ の原始関数であるから、Equation 34.56 より $\int_a^b (u'v+uv')\,dx = [\,u(x)v(x)\,]_a^b$ となり、両辺から $\int_a^b u'v\,dx$ を引けば Equation 34.58 が得られる。

この公式がしているのは、境界項が生じる代わりに、一方の因子にかかっていた微分をもう一方の因子へと移すことである。新しい被積分関数 $u'v$ が元の uv' より単純になるときはいつでも、この変形が威力を発揮する。$u(x)=x$、$v'(x)=e^{-x}$（したがって $u'=1$、$v=-e^{-x}$）と選び、Equation 34.58 を $[0,b]$ 上で適用して $b\to\infty$ とすると、次式が得られる。

$$\int_0^\infty x\,e^{-x}\,dx = \Big[-x\,e^{-x}\Big]_0^\infty + \int_0^\infty e^{-x}\,dx = 0+1 = 1.$$

両端点において $xe^{-x}\to 0$ となるため境界項は消え、先ほど計算した広義積分だけが残る。この積分は指数分布の平均である（36.3 節）。

期待値は積の積分であるため、扱いが困難な因子から扱いやすい因子へと微分を移す手法は、確率論や機械学習で繰り返し登場する定石である。釣鐘曲線は、この技法を最も鮮やかに示してくれる。正規化されていない釣鐘型の重み $w(z)=e^{-z^2/2}$ を考えると、これは $w'(z)=-z\,w(z)$ を満たす。g を連続微分可能な関数で $g(z)w(z)\to 0$（$z\to\pm\infty$）を満たすものとし、以下の 2 つの積分が収束すると仮定する。$[-a,b]$ 上で部分積分を行い、続いて $a\to\infty$ と $b\to\infty$ の極限を別々にとると、次式が得られる。

$$\int_{-\infty}^{\infty} g'(z)\,w(z)\,dz = \Big[\,g(z)\,w(z)\,\Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} g(z)\,w'(z)\,dz = \int_{-\infty}^{\infty} z\,g(z)\,w(z)\,dz. \tag{34.59}$$

釣鐘型の重みのもとでは、z を掛けることは g を微分することと同じ効果をもつ。任意の多項式 g に対して上記の両方の仮定が成り立つ。$g(z)=z$ とおくと $\int z^2 w\,dz = \int w\,dz$ となり、釣鐘型の重みのもとでの z^2 の平均は 1 となる。これは正規化定数 $\int w\,dz$ がいくらになるかによらない（それが $\sqrt{2\pi}$ になることは Equation 34.66 で示す）。$g(z)=z^3$ とおくと $\int z^4 w\,dz = 3\int z^2 w\,dz = 3\int w\,dz$ となるため、z^4 の平均は 3 である。w を標準正規密度 $p=w/\sqrt{2\pi}$ へと正規化すれば、この式は $\mathbb{E}[g'(Z)]=\mathbb{E}[Z\,g(Z)]$ となり、標準正規分布に対するスタインの恒等式（Stein's identity）として知られる。未知の因子から既知の因子へと導関数を移す操作は、スコアマッチングの背後にある重要なステップでもある（38.6 節、(Hyvärinen 2005)）。$\mathbb{R}^d$ において対応する規則は 1 次元の導関数の代わりに発散を移すものであり、発散、ラプラシアン、およびその多変数の恒等式は、それらが最初に必要となる 38.5.2.1 節で導入する。

34.6.2.4 置換積分

連鎖律を評価公式を通して逆向きに読むと、置換積分（substitution）の公式が得られる。u が $[a,b]$ 上で連続微分可能であり、f が $u([a,b])$ を含む区間上で連続で原始関数 G をもつとする。連鎖律により $(G\circ u)'(x) = f(u(x))\,u'(x)$ であるから、次式が成り立つ。

$$\int_a^b f(u(x))\,u'(x)\,dx = \int_a^b (G\circ u)'(x)\,dx = G(u(b)) - G(u(a)) = \int_{u(a)}^{u(b)} f(y)\,dy.$$

これを右から左へ読んだものが、**変数変換の公式**（change-of-variables formula）である。

$$\int_{u(a)}^{u(b)} f(y)\,dy = \int_a^b f(u(x))\,\frac{du}{dx}\,dx. \tag{34.60}$$

言葉で言えば、y に関する積分は x に関する積分へと書き換えることができ、その際には被積分関数に局所伸縮率 du/dx を掛ければよい。なお、u は単調である必要さえない。

1 本の細いスライスを考えれば、この伸縮率の意味は一目瞭然である。$[a,b]$ を幅 ϵ のスライスに分割する。写像 u は、x にあるスライスを幅がおよそ $\epsilon\,u'(x)$ のスライス（位置 $y = u(x)$）へと移し、そこでの曲線の高さは $f(y) = f(u(x))$ である。したがってその面積は $f(u(x))\,u'(x)\,\epsilon$ となり、これこそが x の積分で足し合わせる量にほかならない（Figure 34.31）。

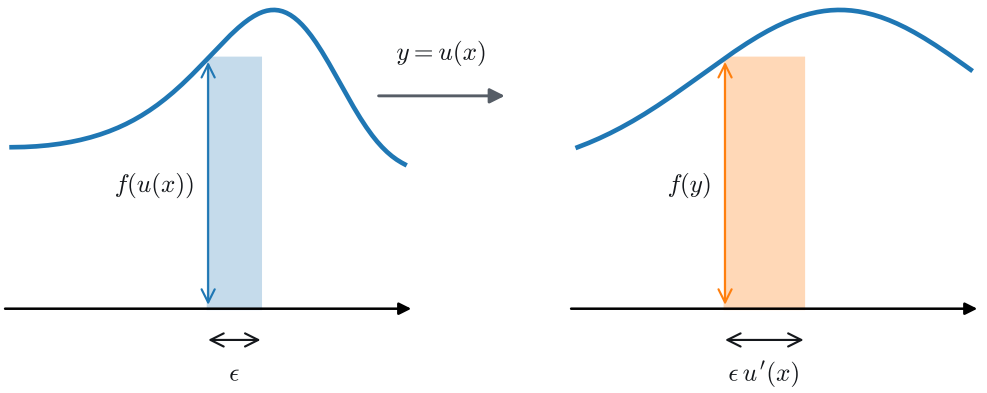


Figure 34.31: 置換 $y = u(x)$ のもとでの細い短冊。左は x 側（公式の右辺）で、幅 ϵ で位置 x にあり高さ $f(u(x))$ をもつ短冊である。右は y 側で、その像は幅 $\epsilon\,u'(x)$ で同じ高さをもつ。両者の面積を一致させるには因子 du/dx が必要になる。

実際の計算では、被積分関数の中に $f(u(x))\,u'(x)$ の形を見抜くことで置換積分を行う。$\int_0^1 x\,e^{-x^2}\,dx$ において $u(x) = x^2$ とおけば $u'(x) = 2x$ となり、被積分関数は $\tfrac{1}{2}e^{-u(x)}u'(x)$ となって次式が得られる。

$$\int_0^1 x\,e^{-x^2}\,dx = \frac{1}{2}\int_0^1 e^{-y}\,dy = \frac{1-e^{-1}}{2} \approx 0.316.$$

ここでは因子 x が、置換に必要な $u'(x)$ を補ってくれた。その x を取り除くと $\int e^{-x^2}\,dx$ が残るが、これは初等的な原始関数をもたず、釣鐘曲線の正規化定数 $1/\sqrt{2\pi}$ の背後にあるまさにその積分である。これを Equation 34.65 で計算するために、以下では 2 次元へ移って極座標に変換することで不足している因子を手に入れる。

34.6.3 重積分

ここからは、(Folland 1999) から標準的な 2 つの定理、すなわちフビニの定理と多変数の変数変換の公式を証明なしで引用する。

34.6.3.1 二重積分とフビニの定理

$\int e^{-x^2}\,dx$ への道筋は 2 次元を経由するため、まずは積分を区間から平面上の領域へと拡張する。2 次元の釣鐘曲面 $f(x,y) = e^{-x^2-y^2}$ を正方形 $[-2,2]^2$ 上で考えると、これは底面の上に体積を囲んでいる。面積を近似したときとまったく同じように、この体積を近似しよう。すなわち、底面を $\epsilon \times \epsilon$ の正方形で敷き詰め、各正方形の上に高さ f の角柱を立て、角柱の体積を足し合わせる（Figure 34.32）。関数 f が長方形 $R = [a,b] \times [c,d]$ 上で与えられたとき、$\epsilon \to 0$ としたときのこれらの直方体和の極限が**二重積分**（double integral）$\iint_R f(x,y)\,dx\,dy$ であり、曲面の下の符号付き体積を表す。

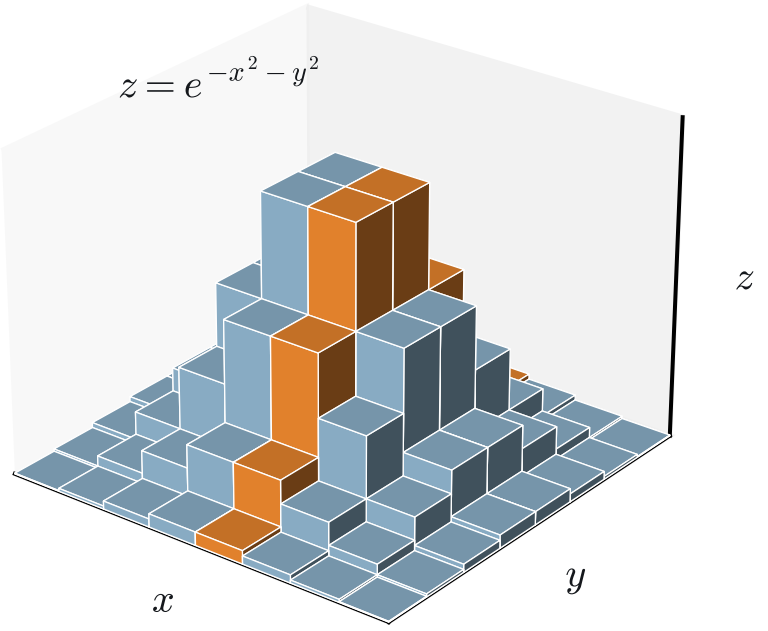


Figure 34.32: 釣鐘曲面 $z = e^{-x^2-y^2}$（正方形 $[-2,2]^2$ 上）。底面を $\epsilon \times \epsilon$ の正方形（$\epsilon = 0.5$）で敷き詰め、それぞれに高さ f の直方体を置くと曲面の下の体積が近似される。固定した x における強調表示された直方体の列は、フビニの定理における 1 つの内側の和に対応する。

1 次元の道具を使って、このような体積をどう計算できるだろうか。有限和であれば、どのような順序で合計しても結果は変わらない。グリッドの正方形を i（x 方向）と j（y 方向）で添字付けすれば、直方体を列ごとに足し合わせることができる（Figure 34.33）。

$$\sum_{i,j} \epsilon^2 f(a+\epsilon i, c+\epsilon j) = \sum_{i} \epsilon \Big(\sum_{j} \epsilon\, f(a+\epsilon i, c+\epsilon j) \Big).$$

内側の和は $\int_c^d f(a+\epsilon i, y)\,dy$、すなわち $x = a+\epsilon i$ における曲面の断面積（Figure 34.32 の強調された列）の左端点近似であり、外側の和はそのスライス断面積を x にわたって積分する。

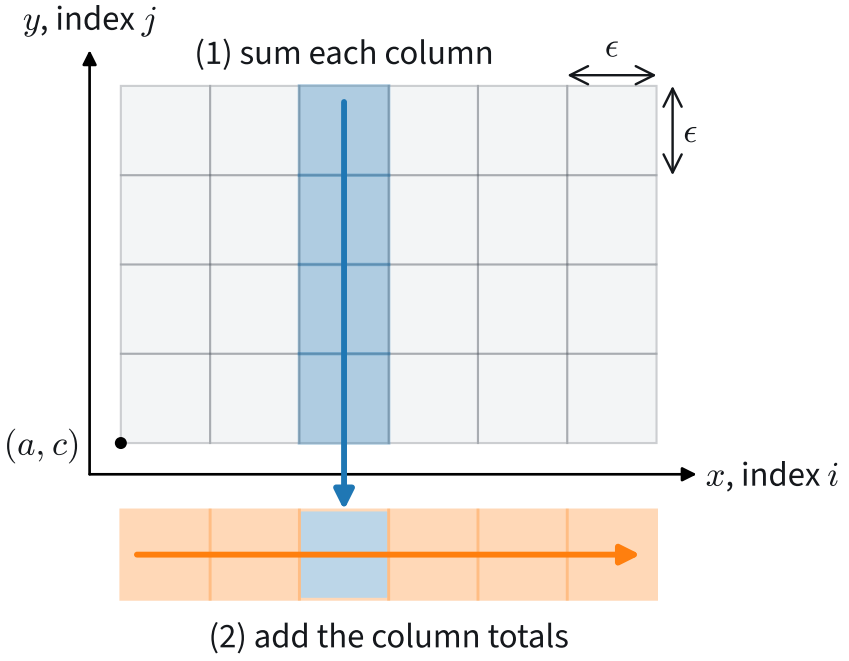


Figure 34.33: $\epsilon \times \epsilon$ のセルからなる角 (a, c) のグリッド（i が x 方向、j が y 方向の添字）の合計。(1) 固定した i において各列を j にわたって足し合わせる。(2) 列ごとの合計を左から右へ足し合わせる。有限和の順序を入れ替えても何も変わらないというこの離散的な事実が、累次積分の根底にある。

有限のグリッド和の順序を入れ替えるのは初等的であるが、$\epsilon \to 0$ の極限をとっても等号が保たれることを保証するには、1 つの定理と条件が必要になる。**フビニの定理**によれば、f が R 上で**絶対可積分**（absolutely integrable）、すなわち $\iint_R |f(x,y)|\,dx\,dy < \infty$ であるならば、二重積分はいずれの順序の累次積分（iterated integral）にも等しい。

$$\iint_R f(x,y)\,dx\,dy = \int_c^d \left(\int_a^b f(x,y)\,dx \right) dy = \int_a^b \left(\int_c^d f(x,y)\,dy \right) dx. \tag{34.61}$$

幾何学的に言えば、体積はどちらの方向にスライスしても計算できる。$f \geq 0$ のときは可積分性

の確認なしに同じ等式が成り立ち、3 辺すべてが $+\infty$ となる場合も含まれる（**トネリの定理**）。したがって確率密度や二乗損失などの非負の被積分関数では確認が不要であるが、符号が変化する被積分関数に対しては、まず $\iint |f| < \infty$ を確かめなければならない。同時密度から変数を周辺化して消去するときや、期待値と別の積分の順序を交換するときには、常にフビニの定理が働いている。

最もよく現れる特別な場合は、積の形をした被積分関数 $f(x,y) = g(x)h(y)$ である。$g(x)$ は y に関して定数であるため、内側の積分の外へそのままくくり出せて $\int_c^d g(x)h(y)\,dy = g(x)\int_c^d h(y)\,dy$ となり、次式が得られる。

$$\iint_R g(x)\,h(y)\,dx\,dy = \int_a^b g(x)\,dx \int_c^d h(y)\,dy. \tag{34.62}$$

積の被積分関数は因子ごとに積分できる。決定的に重要なのは、2 次元の釣鐘曲面が $e^{-x^2-y^2} = e^{-x^2}e^{-y^2}$ とこの形に因数分解されることであり、以下で行うガウス積分の計算はこの事実に基づいている。

絶対可積分性の仮定は単なる形式的な条件ではない。$f(x,y) = (x^2-y^2)/(x^2+y^2)^2$ を $[0,1]^2$ 上で考えよう。簡単に微分して確かめられるように $\partial_y\bigl(y/(x^2+y^2)\bigr) = f(x,y)$ であるから、評価公式より $\int_0^1 f(x,y)\,dy = \frac{1}{1+x^2}$（$x>0$）となり、さらに x で積分すると $\int_0^1 \frac{dx}{1+x^2} = \arctan 1 = \frac{\pi}{4}$ が得られる。ところが $f(x,y) = -f(y,x)$ であるため、逆の順序で積分すると $-\frac{\pi}{4}$ になってしまう。2 つの順序が食い違うのは $\iint |f| = \infty$ だからである。原点の近くで $|f|$ は距離の逆二乗のオーダーで増大し、その全体の体積は発散する。

領域 D が長方形でない場合、$\int_D f(\mathbf{x})\,d\mathbf{x}$ は**ゼロ拡張**（zero-extension）によって定義する。すなわち、f を 0 と D の外側で定めておき、D を含む任意の長方形上で積分する。

34.6.3.2 多変数の変数変換

1 次元の伸縮率 du/dx は高次元へ自然に一般化され、それがどのような形をとるべきかは線形変換を見るだけですでにわかる。線形変換はあらゆる面積や体積をその行列式の絶対値倍に拡大・縮小し（33.2 節）、行列式の符号は単に向きを追跡するにすぎない。たとえば写像 $\phi(u,v) = (au, bv)$（$a,b>0$）は行列式 ab をもち、単位円板（面積 π）を楕円 $x^2/a^2 + y^2/b^2 \le 1$ へと引き伸ばすため、その面積は πab となる。

滑らかな非線形写像は局所的には線形に見え、そのヤコビ行列が線形近似の役割を果たす。体積の拡大率が点ごとに変化するだけである。$\Omega \subseteq \mathbb{R}^n$ を開集合とし、$\phi : \Omega \to \mathbb{R}^n$ をその像の上への C^1 **級微分同相写像**（diffeomorphism）とする。すなわち、ϕ は連続微分可能かつ単射であり、そのヤコビ行列 $\mathbf{J}_\phi$（34.4 節で扱った偏微分の行列）は $\det \mathbf{J}_\phi(\mathbf{x}) \neq 0$ を Ω 上のいたるところで満たす。単射性は写像が空間を折り返して重ね合わせるのを防ぎ、非ゼロの行列式は領域を低次元へ押し潰すのを防ぐ。これらの条件のもとで、任意の絶対可積分関数 f に対して次式が成り立つ。

$$\int_{\phi(\Omega)} f(\mathbf{y})\,d\mathbf{y} = \int_{\Omega} f(\phi(\mathbf{x}))\,\bigl|\det \mathbf{J}_\phi(\mathbf{x})\bigr|\,d\mathbf{x}. \tag{34.63}$$

体積 $d\mathbf{x}$ をもつ点 $\mathbf{x}$ における微小立方体は、体積 $\bigl|\det \mathbf{J}_\phi(\mathbf{x})\bigr|\,d\mathbf{x}$ をもつ点 $\mathbf{y} = \phi(\mathbf{x})$ における微小平行多面体へと写され、ヤコビ行列式がその局所的な歪みに合わせて被積分関数を補正する。Figure 34.34 に 1 次元と 2 次元の図を並べて示す。

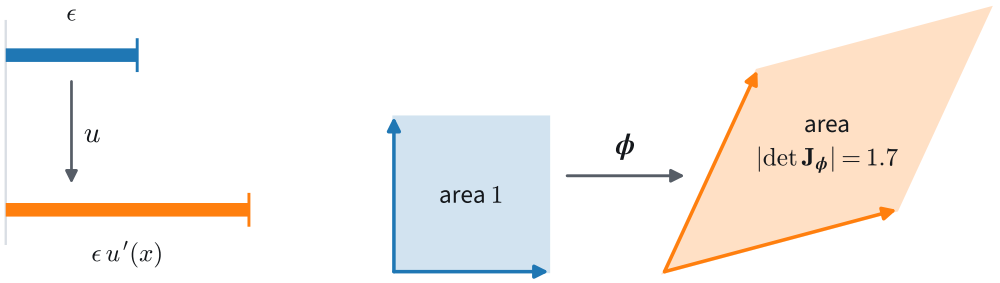


Figure 34.34: 左：1 次元の置換では、長さ ϵ の線分が $u'(x)$ 倍に伸縮される。右：2 次元では、線形変換 ϕ が単位正方形を平行四辺形へと写し、その面積が体積拡大率 $|\det \mathbf{J}_\phi|$ となる。

なぜ Equation 34.63 には $|\det \mathbf{J}_\phi|$ のように絶対値がつき、単なる $\det \mathbf{J}_\phi$ ではないのだろうか。左辺が順序付けられた端点の間ではなく、向きをもたない集合上で積分しているからである。1 次元では積分区間の上下端の順序が符号を担っている。もし u が減少関数であれば、Equation 34.60 の積分区間は逆向きになる一方で $u' < 0$ となるため、2 つの負号が打ち消し合い、像の区間上で $|u'|$ を掛けて積分するのと同値になる。$\mathbb{R}^n$ 内の領域には自然な順序が存在しないため、この符号なしの形式だけが一般化されるのである。

f が確率密度であるとき、Equation 34.63 は滑らかな可逆写像のもとで密度がどう変換されるかを表しており、正規化フロー（normalizing flow）が出力の密度を計算する仕組みそのものである（36.2 節）。

34.6.3.3 ガウス積分

釣鐘曲線の正規化定数 $1/\sqrt{2\pi}$ は、1 次元積分 $\int_{-\infty}^{\infty} e^{-x^2}\,dx$ に由来する。ここでは基本定理を使っても閉形式は得られない。e^{-x^2} は連続であるため原始関数は確かに存在するが、それは初等関数では表せない本質的に新しい関数だからである。驚くべきことに、それでも実数直線全体にわたる積分は、2 次元を経由する回り道をとることで閉形式で求められる。その議論は 3 つのステップからなり、本節で最も美しい計算である。（正規化した原始関数は**誤差関数**（error function）$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt$ であり、これから証明する結果により 1 へと収束する（$x \to \infty$）。標準ライブラリでは `math.erf` として提供されている。）

I. 二乗と因数分解： 正方形 $[-b, b]^2$ 上で積の公式 Equation 34.62 を適用すると、$\iint_{[-b,b]^2} e^{-x^2-y^2}\,dx\,dy = \left(\int_{-b}^{b} e^{-x^2}\,dx\right)^2$ となる。両辺で $b \to \infty$ とすると次式が得られる。

$$\left(\int_{-\infty}^{\infty} e^{-x^2}\,dx\right)^2 = \lim_{b\to\infty} \iint_{[-b,b]^2} e^{-x^2-y^2}\,dx\,dy = \iint_{\mathbb{R}^2} e^{-x^2-y^2}\,dx\,dy. \tag{34.64}$$

求めたい 1 次元積分の二乗は、単に 2 次元の釣鐘曲面の下の体積にほかならず、この曲面は原点の周りで回転対称である。

II. 極座標： 回転対称性に導かれて、極座標 $\phi(r, \theta) = (r\cos\theta, r\sin\theta)$ を開集合 $\Omega = (0, \infty) \times (0, 2\pi)$ 上で導入する。この集合上で写像は単射であり、そのヤコビ行列式は次のようになる。

$$\left|\det \mathbf{J}_{\phi}\right| = \left|\det \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}\right| = r(\cos^2\theta + \sin^2\theta) = r.$$

幾何学的には、動径方向の辺 dr と弧長 $r\,d\theta$ をもつ極座標セルの面積はおよそ $r\,dr\,d\theta$ であり（Figure 34.35）、原点から遠いセルほど幅が広がることを反映している。決定的に重要なのは、この因子 r が、先ほどの置換積分の例で現れた因子 x（$\int_0^1 x\,e^{-x^2}\,dx$）と同じ役割を果たし、不足していた因子を補ってくれることである。

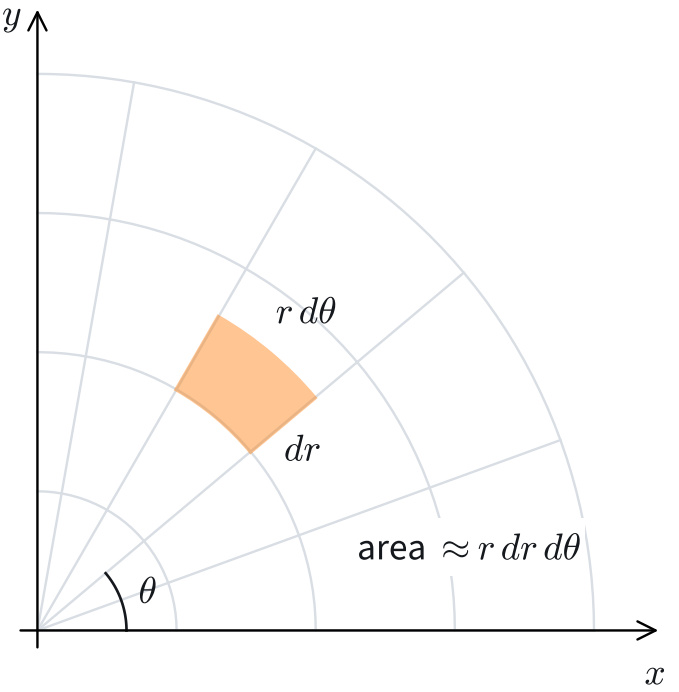


Figure 34.35: 1 つのセルを網掛けした極座標グリッド。動径方向の辺 dr と弧の辺 $r\,d\theta$ により、面積はおよそ $r\,dr\,d\theta$ となる。極座標に変換すると、被積分関数に $\left|\det \mathbf{J}_{\phi}\right| = r$ が掛かる。

III. 積分：@eq-mdl-change-var-nd を適用し、原始関数 $-\frac{1}{2}e^{-r^2}$（re^{-r^2} の原始関数）を用いると、次式が得られる。

$$\iint_{\mathbb{R}^2} e^{-x^2-y^2}\,dx\,dy = \int_0^{2\pi}\int_0^{\infty} r\,e^{-r^2}\,dr\,d\theta = 2\pi\cdot\frac{1}{2} = \pi.$$

Equation 34.64 の正の平方根をとれば、次式が導かれる。

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}. \tag{34.65}$$

こうして、e^{-x^2} の下の全体の面積は $\sqrt{\pi} \approx 1.7725$ と求まる。

3 つの技術的な要点を確認しておけば、この議論は完全に厳密になる。第 1 に、被積分関数が非負であるため、拡大する正方形上の積分と拡大する円板上の積分は同じ極限に収束する。どの円板もある正方形の内部に収まり、どの正方形もある円板の内部に収まるため、広がっていく領域の形状は結果に影響しない。第 2 に、この極座標写像は非負の半直線 $\theta = 0$ を除いているが、その面積はゼロであり積分に一切寄与しない。第 3 に、変数変換は被積分関数が有界かつ連続である有界な長方形 $(0, b) \times (0, 2\pi)$ 上で適用し、そのあとで $b \to \infty$ の極限をとる。

これで標準正規分布の正規化定数が、たった 1 回の置換から導かれる。$z = \sqrt{2}\,x$ と Equation 34.60 でおけば、次式が得られる。

$$\int_{-\infty}^{\infty} e^{-z^2/2}\,dz = \sqrt{2}\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{2\pi}. \tag{34.66}$$

釣鐘関数 $e^{-z^2/2}$ を $\sqrt{2\pi} \approx 2.5066$ で割ると全体の面積が 1 に規格化される。これが標準正規密度に現れる定数の由来である。

ステップ I を数値的に確かめてみよう。拡大する正方形 $[-b, b]^2$ は本当に平面を覆い尽くして π に収束するのだろうか。以下のコードでは、$[-b, b]^2$ 上で $\epsilon = 0.01$ とした 2 次元グリッド和を、正方形上の厳密な体積 $\left(\int_{-b}^{b} e^{-x^2}\,dx\right)^2 = \pi\,\mathrm{erf}(b)^2$ および極限値 π と並べて計算する。

```
eps = 0.01
for b in [1, 2, 3, 6]:
    g = -b + eps * np.arange(round(2 * b / eps))    # left ends of the slices
    x, y = np.meshgrid(g, g, indexing='ij')
    grid = np.sum(eps**2 * np.exp(-x**2 - y**2))
    print(f'b={b}  grid sum={grid:.6f}  pi*erf(b)^2={math.pi * math.erf(b)**2:.6f}'
          f'  pi={math.pi:.6f}')
```

```
b=1  grid sum=2.230949  pi*erf(b)^2=2.230985  pi=3.141593
b=2  grid sum=3.112266  pi*erf(b)^2=3.112270  pi=3.141593
b=3  grid sum=3.141454  pi*erf(b)^2=3.141454  pi=3.141593
b=6  grid sum=3.141593  pi*erf(b)^2=3.141593  pi=3.141593
```

グリッド和は $\pi\,\mathrm{erf}(b)^2$ を精度よく追跡しながら π へと上昇していき、$b = 3$ の時点で正方形の外に残された体積はわずか 10^{-4} 程度にすぎない。$\epsilon = 0.01$ で小数第 4 位以上まで一致するのには明快な理由がある。対称な区間上の偶関数に対しては、左端点則の 1 次バイアス $\frac{\epsilon}{2}\big(f(-b) - f(b)\big)$ が恒等的にゼロになるからである。

いまや標準正規確率 $P(-1 \le Z \le 1) \approx 0.68$ も計算できる。$z = \sqrt{2}\,t$ と置換すると、$p(z) = e^{-z^2/2}/\sqrt{2\pi}$ の下の面積は誤差関数へと変換される。

$$P(-1 \le Z \le 1) = \frac{1}{\sqrt{2\pi}}\int_{-1}^{1} e^{-z^2/2}\,dz = \frac{1}{\sqrt{\pi}}\int_{-1/\sqrt{2}}^{1/\sqrt{2}} e^{-t^2}\,dt = \mathrm{erf}\!\left(1/\sqrt{2}\right) \approx 0.683.$$

これで釣鐘曲線に関して冒頭で立てた 2 つの問いはともに解決し、$p(0) \approx 0.40$ であるのに $P(Z = 0) = 0$ となる見かけ上のパラドックスも解消する。密度が測っているのは単位長さあたりの確率であり、1 点は長さゼロの区間だからである。

冒頭で挙げた問いのうち 1 つがまだ残っている。ランダムサンプリングを通じてモデルを訓練するには、密度のパラメータが動いたときにこの種の面積がどう変化するかを知る必要があり、そのためには積分を微分しなければならない。

34.6.4 まとめ

定積分は細いスライスの和の極限であり、曲線の下の符号付き面積を測る。微分積分学の基本定理はこの極限を原始関数の計算へと置き換える。累積面積関数が $F' = f$ を満たすため、任意の原始関数 G によって $\int_a^b f\,dx = G(b) - G(a)$ が得られるからである。部分積分と置換積分は、それぞれ積の微分法則と連鎖律を逆向きに読んだものである。多次元では、フビニの定理が体積積分を 1 次元の累次積分へと帰着させ、変数変換の伸縮率は局所的な体積の歪みを表す $|\det \mathbf{J}_\phi|$ となる。これを極座標と組み合わせることで、$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$ と標準正規分布の正規化定数 $\sqrt{2\pi}$ が導かれる。

34.6.5 演習問題

1. **累積面積としての活性化関数：** H を単位ステップ関数、すなわち $H(t) = 1$（$t > 0$）、$H(t) = 0$（$t \le 0$）とする。また $\phi(t) = e^{-t^2/2}/\sqrt{2\pi}$ を標準ガウス密度、$\Phi(x) = \int_{-\infty}^{x} \phi(t)\,dt$ をその分布関数とする。
   1. $\int_{-\infty}^{x} H(t)\,dt = \mathrm{ReLU}(x)$ であることを示せ。基本定理 Equation 34.55 のどの仮定が $x = 0$ において破れているか。また、その点における積分の片側導関数はそれぞれどうなるか。
   2. $\Phi(x) = \Phi(0) + \int_0^x \phi(t)\,dt$ と表すことで $\Phi' = \phi$ を導け。続いて $t = \sqrt{2}\,s$ と置換し、$\Phi(x) = \tfrac{1}{2}\bigl(1 + \mathrm{erf}(x/\sqrt{2})\bigr)$ であることを示せ。
   3. 4.1 節の GELU 活性化関数は $x\Phi(x)$ である。その導関数を導出し、導関数が最大および最小となる点を求め、`math.erf` を用いてその点での値を計算せよ。GELU の傾きはどの区間に収まるか。また、その区間は ReLU の傾きである 0 および 1 と比べてどのようになっているか。
2. **重い裾：** 裾指数 $\alpha > 0$ をもつ**パレート密度**は、$p(x) = \alpha x^{-\alpha-1}$（$x \ge 1$ のとき）、それ以外で $p(x) = 0$ と定義される。また**コーシー密度**は、$p(x) = \frac{1}{\pi(1+x^2)}$（$\mathbb{R}$ 上）である。
   1. すべての $\alpha > 0$ に対してパレート密度が正規化されていることを示せ。α がどの範囲にあるとき $\mathbb{E}[X]$ および $\mathbb{E}[X^2]$ は有限になるか。有限となる範囲で $\mathbb{E}[X]$ を計算せよ。
   2. 原始関数 $\arctan x$（$\frac{1}{1+x^2}$ の原始関数）を用いて、コーシー密度が正規化されていることを示せ。
   3. コーシー密度において $\int_{-b}^{b} x\,p(x)\,dx = 0$ がすべての $b > 0$ に対して成り立つことを示せ。それにもかかわらず、$\int_{-\infty}^{\infty}$ の定義（Equation 34.57 に続く定義）のもとでコーシー密度が平均をもたないのはなぜか。また、$\int_0^b x\,p(x)\,dx$ は b とともにどれほどの速さで増大するか。
3. **部分積分によるガンマ関数： ガンマ関数**は $\Gamma(t) = \int_0^\infty u^{t-1}e^{-u}\,du$ で定義される。$t < 1$ のとき被積分関数は $u = 0$ で非有界となり、0 の近くでの積分は極限 $\int_0^1 = \lim_{a\to 0^+}\int_a^1$ によって定義される（34.6.2.2 節と同様）。
   1. どの t に対して積分は $u = 0$ で収束するか。その条件を $\int_0^1 x^{-s}\,dx$ の判定基準（34.6.2.2 節）と比較せよ。
   2. 部分積分 Equation 34.58 を一般の $t > 0$ において適用することで $\Gamma(t+1) = t\,\Gamma(t)$ を示し、$\Gamma(n+1) = n!$ がすべての整数 $n \ge 0$ に対して成り立つことを導け。この議

論のどこで $t > 0$ を用いているか。さらにこの結果を用いて、指数密度 e^{-u}（$u \geq 0$ 上、その平均は 34.6.2.3 節で求めた）の分散を計算せよ。

3. $u = x^2$ と置換し、ガウス積分 Equation 34.65 を用いて $\Gamma(\tfrac{1}{2})$ の値を求めよ。
4. 積の公式 Equation 34.62 を d 回適用すると $\int_{\mathbb{R}^d} e^{-\|\mathbf{x}\|^2}\, d\mathbf{x} = \pi^{d/2}$ が得られる。極座標形式では同じ積分が $S_{d-1}\int_0^\infty r^{d-1}e^{-r^2}\, dr$ となる（ここで S_{d-1} は $\mathbb{R}^d$ における単位球面の表面積である）。動径方向の積分が $\tfrac{1}{2}\Gamma(d/2)$ になることを示し、$S_{d-1} = 2\pi^{d/2}/\Gamma(d/2)$ を導け。続いて $S_{d-1}r^{d-1}$ を $r \in [0,1]$ 上で積分し、単位球体の体積が $V_d = \pi^{d/2}/\Gamma(d/2+1)$ となることを示せ。V_d を $d \in \{1,2,3,5,10,20\}$ に対して計算し、なぜ $V_d \to 0$ となるのかを説明せよ。超立方体 $[-1,1]^{10}$ 上で各軸あたり 3 点とる中点グリッドには 3^{10} 個の点があるが、そのうち単位球体の内部に入る点の割合はどれだけか。また、そのことから高次元におけるグリッドについて何が言えるか。

4. **積分順序の交換：**フビニの定理 Equation 34.61 と、長方形でない領域に対するゼロ拡張を組み合わせれば、積分の順序を自由に選ぶことができる。
   1. $\int_0^1 \big(\int_y^1 e^{-x^2}\, dx\big)\, dy$ の内側の積分は初等的な原始関数をもたない。三角形の積分領域を図示し、積分の順序を入れ替えて閉形式で積分値を求めよ。
   2. $X \geq 0$ が密度 p をもつとする。$x = \int_0^x dt$ を $\mathbb{E}[X] = \int_0^\infty x\, p(x)\, dx$ の内部に代入し、2 つの積分を交換することで $\mathbb{E}[X] = \int_0^\infty P(X > t)\, dt$ を示せ。$\mathbb{E}[X] = \infty$ のときでさえこの交換が正当化されるのはなぜか。密度 $\lambda e^{-\lambda x}$（$x \geq 0$）に対してこの公式を確かめよ。
5. **ガウス分布の正規化定数とモーメント：**本文では標準正規分布を正規化し、釣鐘型の重みに対するスタインの恒等式 Equation 34.59 を用いてそのモーメントを計算した。$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\!\big(-\frac{(x-\mu)^2}{2\sigma^2}\big)$（$\sigma > 0$）とする。
   1. $x = \mu + \sigma z$ と置換し、Equation 34.66 を用いて p が正規化されていることを示せ。どのような μ と σ^2 が密度 $\frac{1}{\sqrt{\pi}}e^{-x^2}$ を与えるか。
   2. X が密度 p をもち、g が連続微分可能で $g(x)\,p(x) \to 0$（$x \to \pm\infty$）を満たすとする。$p'(x) = -\frac{x-\mu}{\sigma^2}\,p(x)$ と部分積分を用いて次式を示せ。

$$\mathbb{E}\big[(X-\mu)\,g(X)\big] = \sigma^2\,\mathbb{E}\big[g'(X)\big]. \tag{34.67}$$

   これから $\mathbb{E}[(X-\mu)^2] = \sigma^2$ および $\mathbb{E}[(X-\mu)^4] = 3\sigma^4$ を導け。
   3. $\Sigma = \mathbf{L}\mathbf{L}^\top$ を $d \times d$ の共分散行列（ただし $\mathbf{L}$ は可逆）とする。変数変換 $\mathbf{x} = \mathbf{L}\mathbf{z}$ を Equation 34.63 に適用し、積の公式 Equation 34.62 と組み合わせて次式を示せ。

$$\int_{\mathbb{R}^d} \exp\!\big(-\tfrac{1}{2}\mathbf{x}^\top\Sigma^{-1}\mathbf{x}\big)\, d\mathbf{x} = (2\pi)^{d/2}\,(\det\Sigma)^{1/2}. \tag{34.68}$$

34.7 期待値とその勾配

これまでに最小化してきた損失のほとんどすべては、ミニバッチ内のデータ例、ランダムなドロップアウトマスク、あるいは入力に注入されたノイズに関する平均である。ランダム性の源がモデルパラメータに依存しない限り、そのような平均の微分は単純明快である。平均の導関数は導関数の平均に等しいため、ミニバッチ勾配降下法（minibatch gradient descent）は何ら特別な仕組みを必要とせず直接これを計算できる。難しさが生じるのは、パラメータが確率分布そのものを制御する場合である。変分オートエンコーダ（variational autoencoder、VAE）は、エンコーダが出力する平均と分散をもつ分布から潜在コード（latent code）を引き、方策勾配（policy gradient）エージェント（14.7 節）は、改善しようとしているまさにその方策（policy）から行

動をサンプリングする。いずれの場合も、目的関数はパラメータとともに分布が変化するサンプルに関する期待値となるが、いったん引かれたサンプルは単なる数値にすぎない。そこには、自身を生成したパラメータの記憶は一切残っていないのである。

この種の最も単純な問題を考えてみよう。$X \sim \mathcal{N}(\theta, 1)$ を平均 θ、単位分散をもつ正規確率変数とし、次の期待二乗損失を考える。

$$\ell(\theta) = \mathbb{E}\big[X^2\big] = \theta^2 + 1.$$

この閉形式は、$X = \theta + Z$（ただし Z は標準正規確率変数）とおけば直ちに導かれる。すなわち $\mathbb{E}[(\theta+Z)^2] = \theta^2 + 2\theta\,\mathbb{E}[Z] + \mathbb{E}[Z^2]$ であり、ここで対称性から $\mathbb{E}[Z]=0$、また $\mathbb{E}[Z^2]=1$ である（34.6.2.3 節）。$\theta = 1.5$ における厳密な導関数の値は $\ell'(1.5) = 3$ となる。しかし実際には、モデルは X のサンプルを引くことはできても、$\ell(\theta)$ を閉形式で評価することはできない。では、サンプルのみからどのようにして $\ell'(\theta)$ を推定できるだろうか。

微分積分は、根本的に異なる 2 つの答えを与えてくれる。どちらも期待値を積分として表し（34.6 節）、微分をその積分の内側へと通す。そこから先は道が分かれる。**スコア関数**（score-function）推定量はサンプルには手をつけず確率密度を微分するのに対し、**パスワイズ**（pathwise）推定量はノイズの密度には手をつけずサンプルを微分する。どちらの手法があらゆる設定で優位に立つわけでもなく、両者を比較すること（34.7.3 節）が本節の中心的な目標である。それに先立ち、まずは期待値がどのように積分として書かれるかを簡単に振り返り、微分を積分の内側へ通す操作がいつ正当化されるのかを確かめておこう。このステップこそ、巧妙な誤りが紛れ込みやすい箇所だからである。

34.7.1 積分としての期待値

何かを微分する前に、密度と期待値の記法を定めておこう。確率の章ではこれら両方の概念を詳しく展開するが（36.1 節）、ここでは前節の積分の言葉を用いて改めて述べる。領域 $\mathcal{X}$ 上の確率**密度**（probability density）とは、全体の積分が 1 に等しい非負関数のことである。

$$p(x) \geq 0, \qquad \int_{\mathcal{X}} p(x)\,dx = 1. \tag{34.69}$$

確率変数 X（密度 p をもつ）が区間 $[a,b]$ に入る確率は、その区間上における p の下の面積 $P(a \leq X \leq b) = \int_a^b p(x)\,dx$ で与えられる（36.1 節）。密度は確率そのものではなく単位長さあたりの確率を測るため、その値は 1 を超えうる。たとえば $[0, \tfrac{1}{2}]$ 上の一様密度は、その台（support）上のいたるところで 2 に等しい。本節の主な通し例は**標準正規**分布 $\mathcal{N}(0,1)$ であり、その密度 $p(z) = e^{-z^2/2}/\sqrt{2\pi}$ は 34.6 節のガウス積分によって正規化されている。

期待値（expectation）は、各結果にその確率密度で重み付けしたうえで、X あるいは関数 $g(X)$ の重み付き平均を計算するものである。

$$\mathbb{E}[X] = \int_{\mathcal{X}} x\,p(x)\,dx, \qquad \mathbb{E}[g(X)] = \int_{\mathcal{X}} g(x)\,p(x)\,dx. \tag{34.70}$$

期待値 $\mathbb{E}[g(X)]$ は、積分が絶対収束するとき、すなわち $\int_{\mathcal{X}} |g(x)|\,p(x)\,dx < \infty$ が成り立つときに確定した値をもつ。裾の重い分布はこの条件に反することがあり、たとえば演習問題 2 のコーシー密度は有限の平均をもたない。

深層学習では、期待値が閉形式の式をもつことはめったになく、しかも数千次元もの空間で定義されることが多いため、グリッド上の数値積分（numerical quadrature）は到底現実的ではない。そこで代わりに、独立なサンプル $x_1, \dots, x_n$ を p から引き、標本平均 $\frac{1}{n}\sum_{i=1}^{n} g(x_i)$ を作る。これは**モンテカルロ**推定値として知られる。1 サンプル $g(X)$ の標準偏差が σ であれば、モンテカルロ平均の二乗平均平方根誤差（root-mean-square error、RMSE）は次元にかかわらず $\sigma/\sqrt{n}$ となる（36.5.1.7 節）。本節で扱う 2 つの勾配推定量はいずれもこの形のモンテカルロ平均である。どちらも同じ真の勾配に対する不偏推定量であるため、違いは 1 サンプルあたりのばらつきにしか現れない。したがって、1 サンプルの標準偏差 σ が両者を比較する自然な尺度となる。

34.7.2 積分記号下の微分

34.7.2.1 順序交換

例題の損失 $\ell(\theta) = \mathbb{E}[X^2]$ は、パラメータ θ に応じて変化する積分である。一般に、パラメータが積分に作用する箇所は 2 つある。微分積分学の基本定理（34.6 節）が扱う積分区間の端点を通じて作用する場合と、被積分関数そのものを通じて作用する場合である。

積分区間の端点が固定されており、被積分関数のみが θ に依存するときは、微分を積分の内側へ通すことで微分できる。

$$\frac{d}{d\theta}\int f(x,\theta)\,dx = \int \frac{\partial f}{\partial \theta}(x,\theta)\,dx. \tag{34.71}$$

幾何学的には、全体の面積の変化率は、各点における高さの変化率を積分したものに等しい。以下では、この微分と積分の入れ替えを**順序交換**（swap）と呼ぶ。2 つの勾配推定量はいずれもこれに依拠している。導関数は差分商の極限であり、積分は和の極限であり、有限個の和と差分は常に可換であるから、一見するとこの恒等式には何の問題もないように思える。しかし、2 つの極限の交換は決して自動的に成り立つものではなく、正則性条件（regularity condition）を必要とする。x の有界区間上では、$\partial_\theta f$ が両変数に関して連続であれば十分である。非有界な領域上では、それに加えて x の可積分関数であって、$|\partial_\theta f(x,\theta')|$ をすべての θ'（θ の近傍）に対して上から抑えるものが存在することも必要となる。これは優収束定理（dominated convergence theorem）からの直接の帰結である（証明は省き (Folland 1999) による）。正規密度を用いた通し例 $f(x,\theta) = x^2 p_\theta(x)$ では、ガウス分布の裾がいかなる多項式の増大よりも速く減衰するため、このような優関数による上界が常に存在する。

34.7.2.2 ライプニッツ則

積分区間の端点も θ とともに動く場合、単純な順序交換だけでは不完全であり、誤った答えを与えてしまう。ライプニッツ則（Leibniz rule）は不足している境界項を補うものであり、内部の変化と移動する境界からの寄与を単純に足し合わせればよい（Figure 34.36）。

命題（ライプニッツ則）： **f および $\partial_\theta f$ が、$\{(x,\theta) : x \in [a(\theta), b(\theta)]\}$（ただし θ はある開区間を動く）を含む領域上で連続であり、かつ a と b が微分可能であるとする。このとき、次式が成り立つ。**

$$\frac{d}{d\theta}\int_{a(\theta)}^{b(\theta)} f(x,\theta)\,dx = \int_{a(\theta)}^{b(\theta)} \frac{\partial f}{\partial \theta}(x,\theta)\,dx + f\big(b(\theta),\theta\big)\,b'(\theta) - f\big(a(\theta),\theta\big)\,a'(\theta). \tag{34.72}$$

証明： $H(\alpha,\beta,\theta)=\int_\alpha^\beta f(x,\theta)\,dx$ と定義する。微分積分学の基本定理により $\partial_\beta H = f(\beta,\theta)$ および $\partial_\alpha H = -f(\alpha,\theta)$ となり、一方で有界区間 $[\alpha,\beta]$ 上での順序交換により $\partial_\theta H = \int_\alpha^\beta \partial_\theta f\,dx$ が得られる。これら 3 つの偏微分はすべて連続であるため、多変数の連鎖律（34.3 節）が合成写像 $\theta\mapsto H(a(\theta),b(\theta),\theta)$ に適用でき、3 つの項がそれぞれ $a'(\theta)$、$b'(\theta)$、1 で重み付けされて結合される。■

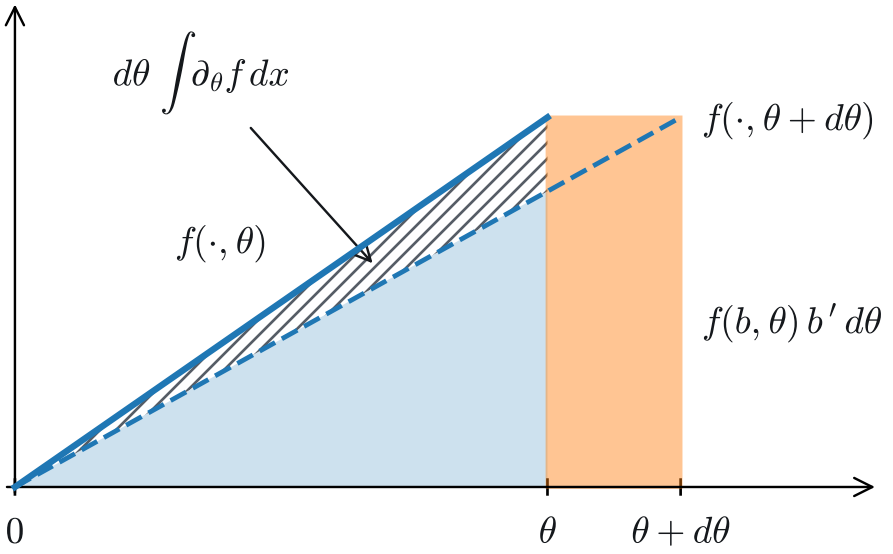


Figure 34.36: 一様分布の例である $f(x,\theta)=x/\theta$（区間 $[0,\theta]$ 上）に対するライプニッツ則。θ が $d\theta$ だけ増加すると、被積分関数の曲線は $f(\cdot,\theta+d\theta)$（破線。ここでは下がるため、斜線領域は失われる面積を表す）へと移動する一方、右側の境界は外側へ動いて高さ $f(b,\theta)$、幅 $b'\,d\theta$ の細い帯を掃き込む。

境界項を無視することは些細な誤差にとどまらず、導関数の符号を反転させてしまうことさえある。X が $[0,\theta]$ 上の一様分布に従い、その区間上で密度 $1/\theta$ をもつとしよう。その平均は $\mathbb{E}[X]=\int_0^\theta \frac{x}{\theta}\,dx=\frac{\theta}{2}$ であり、導関数は $+\frac12$ である。被積分関数だけを微分すると $\int_0^\theta -\frac{x}{\theta^2}\,dx=-\frac12$ となり、符号が逆になってしまう。ここに上端の境界からの寄与 $f(\theta,\theta)\,b'(\theta)=\frac{\theta}{\theta}\cdot 1=1$ を加えることで、$-\frac12+1=+\frac12$ と真の導関数が復元される。分布の台がパラメータに依存する場合には、単純な順序交換だけでは正当化されない（演習問題 1 では $\mathbb{E}[X^2]$ を計算する）。

34.7.2.3 Log-Integral-Exp

順序交換からは、log-sum-exp 関数の連続版も導かれる。連鎖律と Equation 34.71 を $\log\int e^{\theta x}p(x)\,dx$ に適用すると、次式が得られる。

$$\frac{d}{d\theta}\log\int e^{\theta x}p(x)\,dx=\frac{\int x\,e^{\theta x}p(x)\,dx}{\int e^{\theta x}p(x)\,dx}.$$

右辺は、x の平均を $e^{\theta x}p(x)$ に比例する指数的に**傾斜させた**（tilted）密度のもとでとったものである。標準正規密度の場合、$\theta z-\frac12 z^2=\frac12\theta^2-\frac12(z-\theta)^2$ と平方完成することで、傾斜密度

が $\mathcal{N}(\theta, 1)$ であり、かつ $\log \mathbb{E}[e^{\theta Z}] = \theta^2/2$ となることがわかる。実際、その導関数 θ は傾斜平均に一致している。log-sum-exp の勾配がソフトマックスベクトルであるのとちょうど同じように、log-integral-exp の勾配は期待値となる。この恒等式には、対数分配関数（log-partition function）の導関数 $A'(\eta)$ として 36.4 節で再び出会うことになる。

34.7.3 スコア関数勾配とパスワイズ勾配

期待損失に順序交換を適用すると、積分 $\int g(x)\, \partial_\theta p_\theta(x)\, dx$ が得られる。残念ながら、この積分をモンテカルロサンプリングで直接推定することはできない。$\partial_\theta p_\theta$ は確率密度ではないため（正と負の両方の値をとり、積分するとゼロになる）、そこからサンプルを引くことができないからである。しかし 2 つの単純な恒等式を使えば、この積分をサンプルから推定可能な期待値の形へと書き戻せる。

34.7.3.1 スコア関数推定量

p_θ を θ によってパラメータ化された密度とする。連鎖律を $\log p_\theta(x)$ に適用すると、次の**対数微分の恒等式**（log-derivative identity）が得られる。

$$\nabla_\theta\, p_\theta(x) = p_\theta(x)\, \nabla_\theta \log p_\theta(x). \tag{34.73}$$

言葉で言えば、密度の勾配は、密度そのものにその対数の勾配（**スコア**と呼ばれるベクトル $\nabla_\theta \log p_\theta$）を掛けたものに等しい。順序交換が正当化され、かつ p_θ の台が θ とともに動かないときには、スコアの期待値はゼロになる。すなわち $\mathbb{E}[\nabla_\theta \log p_\theta(X)] = \int \nabla_\theta p_\theta\, dx = \nabla_\theta \int p_\theta\, dx = \nabla_\theta 1 = \mathbf{0}$ である。$[0, \theta]$ 上の一様密度を見れば、なぜ台が固定されていることが不可欠なのかがわかる。そのスコアは $\partial_\theta \log(1/\theta) = -1/\theta$ であり、期待値は $-1/\theta \neq 0$ となる。というのも、$\int_0^\theta p_\theta(x)\, dx = 1$ を微分すると境界項 $p_\theta(\theta) = 1/\theta$ も現れるからである。なお、スコアの平均がゼロになる性質は、フィッシャー情報量（Fisher information、36.6 節）の出発点でもある。

Equation 34.73 に関数 $g(x)$（θ に依存しない）を掛けて積分すると、次の**スコア関数恒等式**（score-function identity）が得られる。

$$\nabla_\theta\, \mathbb{E}_{X \sim p_\theta}[g(X)] = \int g(x)\, \nabla_\theta p_\theta(x)\, dx = \mathbb{E}\big[g(X)\, \nabla_\theta \log p_\theta(X)\big]. \tag{34.74}$$

この恒等式により、期待損失の勾配は再び p_θ のもとでの期待値へと戻る。各サンプルの損失 $g(x)$ が、θ がそのサンプルの対数確率（log-probability）をどれだけ強く押し上げるかによって重み付けされるのである。これをモンテカルロ法で推定するには、サンプル $x_i \sim p_\theta$ を引き、$g(x_i) \nabla_\theta \log p_\theta(x_i)$ の平均をとればよい。強化学習（reinforcement learning、RL）において、これは方策勾配法（14.7 節）の REINFORCE 推定量である (Williams 1992)。（もし g も θ に直接依存する場合には、積の微分法則により単に $\mathbb{E}[\nabla_\theta g(X, \theta)]$ の項が加わる。）

スコアの平均がゼロであるため、定数の**ベースライン** b（x に依存しない）を任意に選んで $g(X)$ から引いても、推定量に統計的バイアスを生じさせることはない（14.8 節）。すなわち $\mathbb{E}[(g(X) - b) \nabla_\theta \log p_\theta(X)] = \mathbb{E}[g(X) \nabla_\theta \log p_\theta(X)]$ が成り立つ。b を適切に選べば、推定値の分散を大幅に減らすことができる。

34.7.3.2 パスワイズ推定量

2 つ目の恒等式は、サンプルに θ の記憶を取り戻させるものである。多くの場合、p_θ からのサンプルは、決定論的な変換 $x = T_\theta(\mathbf{z})$ を補助的なノイズ変数 $\mathbf{z} \sim q$（その分布は θ に依存しない）へ施すことで生成できる。たとえば正規確率変数 $\mathcal{N}(\mu, \sigma^2)$ は、単に $\mu + \sigma z$（ただし $z \sim \mathcal{N}(0,1)$）と表せる。期待値を $\mathbf{z}$ に関する積分として書き直せば、密度から θ が完全に取り除かれるため、順序交換によって次の**パスワイズ恒等式**（pathwise identity）が得られる。

$$\nabla_\theta \, \mathbb{E}_{X \sim p_\theta}[g(X)] = \nabla_\theta \int g(T_\theta(\mathbf{z}))\, q(\mathbf{z})\, d\mathbf{z} = \mathbb{E}_{\mathbf{z}\sim q}\Big[\nabla_\theta \, g(T_\theta(\mathbf{z}))\Big]. \tag{34.75}$$

この推定量は密度を微分する代わりに、サンプルそのものを通して微分する。自動微分が θ から T_θ を経て g に至る経路をたどるのである。これは**再パラメータ化トリック**（reparameterization trick）として知られ、変分オートエンコーダがガウス潜在変数（latent variable）を通して逆伝播する仕組みである (Kingma and Welling 2014; Rezende et al. 2014)。推定量自体の歴史はさらに古く、シミュレーションの分野ではパスワイズ微分や微小摂動微分（infinitesimal-perturbation derivative）の名で長く使われてきた (Mohamed et al. 2020)。正当に適用するには、T_θ が θ に関して微分可能であること、g が経路に沿って微分可能であること（Lipschitz 連続な g であれば、X が連続密度をもつ限り十分である）、そして順序交換のための可積分な優関数による上界が存在することが必要となる。なお、たとえ X の台が θ とともに動いたとしても、積分は $\mathbf{z}$（その密度 q は固定されたままである）に関してとられるため、境界項は一切現れないことに注目しよう。

g に対する微分可能性の要件は、無視してよい単なる技術的細目ではない。ステップ関数 $g(x) = \mathbf{1}(x > 0)$（すなわち 1 を正の x に対してとり、それ以外では 0 となる関数）を $X = \theta + z$ のもとで考えてみよう。真の期待値 $\mathbb{E}[g(X)] = P(z > -\theta)$ の導関数は $\varphi(\theta) = e^{-\theta^2/2}/\sqrt{2\pi}$ であり、その値は約 0.13（$\theta = 1.5$ のとき）である。ところがパスワイズ推定量のサンプル $\partial_\theta \mathbf{1}(\theta + z > 0)$ は 0 を値としてとり、それがすべての z（確率ゼロの事象である唯一の点 $z = -\theta$ を除く）で成り立つため、パスワイズ推定量は 0 を確率 1 で返してしまう。対照的に、スコア関数推定量のサンプル $\mathbf{1}(x > 0)\, z$ は期待値 $\int_{-\theta}^{\infty} z\,\varphi(z)\,dz = \big[-\varphi(z)\big]_{-\theta}^{\infty} = \varphi(\theta)$ をもち、厳密な導関数を復元する。スコア関数推定量は g やサンプリングのステップを通して微分することが一切ないため、不連続な損失にも、ベルヌーイ分布やカテゴリ選択のような離散分布（そこでは θ からサンプルへ至る微分可能な経路が存在しない）にもまったく同様に適用できる。

34.7.3.3 2 つの推定量の比較

これで同じ導関数に対する 2 つの不偏推定量（unbiased estimator）が出そろった。通し例である $\ell(\theta) = \mathbb{E}[X^2]$（ただし $X = \theta + z$）は十分に単純なため、両者の分散を手計算で求めることができる。パスワイズ推定量では、1 サンプルあたりの寄与は $\partial_\theta (\theta + z)^2 = 2X$ であり、その分散は $\mathrm{Var}(2X) = 4$ となる。スコア関数推定量では、正規位置族（normal location family）のスコアが $\partial_\theta \log p_\theta(X) = X - \theta = z$ であるから、1 サンプルあたりの寄与は $X^2 z$、あるいは $(X^2 - b)z$（ベースライン b を用いる場合）となる。$X^2 z$ の分散を求めるには、標準正規分布のモーメント $\mathbb{E}[z^2] = 1$、$\mathbb{E}[z^4] = 3$、$\mathbb{E}[z^6] = 15$ を用いる。これらはいずれもスタインの恒等式 $\mathbb{E}[g'(Z)] = \mathbb{E}[Z\, g(Z)]$ から導かれる（34.6.2.3 節）。対称性により、z の奇数乗の項は二次モーメント（second moment）においてすべて消える。

$$\mathbb{E}\big[(\theta + z)^4 z^2\big] = \theta^4\,\mathbb{E}[z^2] + 6\theta^2\,\mathbb{E}[z^4] + \mathbb{E}[z^6] = \theta^4 + 18\theta^2 + 15.$$

ここから平均の二乗 $(2\theta)^2$ を引くと、分散 $\theta^4 + 14\theta^2 + 15$ が残る。分散を最小化する定数ベースラインは $b^\star = \theta^2 + 3$ となることがわかり、これにより分散は $8\theta^2 + 6$ まで下がる（演習問題 2）。$\theta = 1.5$ における 3 つの推定量の 1 サンプル標準偏差は、パスワイズ推定量が 2、素のスコア

関数推定量が 7.18、最適ベースライン付きスコア関数推定量が 4.90 となる。ベースラインは目に見えて有効ではあるものの、この損失に関しては差を埋めきることはできない。$8\theta^2 + 6 > 4$ がすべての θ に対して成り立つからである。

これら 3 つの閉形式による予測を、1 回のモンテカルロシミュレーションで確かめてみよう。パスワイズ微分は単に $2x_i$ であるため、ここでは自動微分すら必要ない。ここまでの式変形が正しければ、3 つの標本標準偏差はそれぞれ 2、7.18、4.90 の近くに収まり、各標本平均は標準誤差（standard error）$\mathrm{sd}/\sqrt{n}$ の数倍以内で真の導関数の値 3 に一致するはずである。

```
theta, n = 1.5, 200_000
z = np.random.default_rng().standard_normal(n)     # noise, independent of theta
x = theta + z                                      # samples of N(theta, 1)
pathwise = 2 * x                                   # d/dtheta (theta + z_i)^2
score = x**2 * z                                   # g(x) times the score x - theta
baseline = (x**2 - (theta**2 + 3)) * z             # with the optimal baseline
exact_sd = {'pathwise': 2, 'score': math.sqrt(theta**4 + 14 * theta**2 + 15),
            'score+baseline': math.sqrt(8 * theta**2 + 6)}
print(f'true derivative 2*theta = {2 * theta}')
for name, s in zip(exact_sd, [pathwise, score, baseline]):
    print(f'{name:15s} mean={s.mean():.4f}  sd={s.std():.3f} '
          f'(exact {exact_sd[name]:.3f})  se={s.std() / math.sqrt(n):.4f}')
```

```
true derivative 2*theta = 3.0
pathwise        mean=3.0041  sd=2.000 (exact 2.000)  se=0.0045
score           mean=3.0068  sd=7.146 (exact 7.181)  se=0.0160
score+baseline  mean=2.9961  sd=4.858 (exact 4.899)  se=0.0109
```

典型的な実行結果では、3 つの平均はいずれも標準誤差の 2〜3 倍以内で 3 の近くに収まり、経験標準偏差も閉形式の予測と数パーセント以内で一致する。一定のサンプル数の予算のもとでは、パスワイズ推定値は素のスコア関数推定値より約 3.6 倍精度が高い。言い換えれば、素のスコア関数推定量が同じ精度に達するには約 13 倍のサンプル数を要する。

しかし、パスワイズ推定量のほうが常に分散が小さいと結論づけるのは誤りである。振動する損失 $g(x) = \sin(\omega x)$ を考えてみよう。このとき期待値の真の導関数は $\omega \cos(\omega\theta) e^{-\omega^2/2}$ となる。パスワイズ推定量のサンプル $\omega \cos(\omega X)$ は周波数 ω に比例して大きくなり、$\omega/\sqrt{2}$ に近い標準偏差をもつのに対し、スコア関数推定量のサンプル $\sin(\omega X)\, z$ の標準偏差は高々 $\sqrt{\mathbb{E}[z^2]} = 1$ である（$|\sin| \le 1$ だからである）。被積分関数が激しく振動する場合、関数値そのものは有界にとどまっていても、その導関数は巨大になりうる。すでに引いたサンプルを再利用し、$\omega = 1$ と $\omega = 3$ において 2 つの推定量を比較してみよう。

```
for omega in [1, 3]:
    true = omega * math.cos(omega * theta) * math.exp(-omega**2 / 2)
    path, score = omega * np.cos(omega * x), np.sin(omega * x) * z
    print(f'omega={omega}  true={true:+.4f}  pathwise {path.mean():+.4f} '
          f'(sd {path.std():.2f})  score {score.mean():+.4f} (sd {score.std():.2f})')
```

```
omega=1  true=+0.0429  pathwise +0.0414 (sd 0.66)  score +0.0432 (sd 0.54)
omega=3  true=-0.0070  pathwise -0.0067 (sd 2.12)  score -0.0080 (sd 0.71)
```

$\omega = 1$ の時点ですでにスコア関数推定量のほうがばらつきが小さく（約 0.5 対 0.7）、$\omega = 3$ では 3 倍の差をつけて優位に立つ（約 0.7 対 2.1）。どちらの標本平均も、標準誤差の数倍以内で厳密な導関数の値と一致している。

どちらの推定量も、あらゆる問題で優位に立つわけではない。パスワイズ推定量は g を微分するため、g' が大きいときに分散が増大する。一方、スコア関数推定量は g にスコアを掛けるため、g そのものが大きいときに分散が増大する。いずれの場合も、所定の精度を得るためにどれだけのサンプル数が必要かは、1 サンプルの標準偏差 σ によって決まる。この σ はサンプルを引く元の分布にも依存するため、σ を小さくするような提案分布（proposal distribution）を選ぶという発想から、直ちに重要度サンプリング（importance sampling）が導かれる。これについては 36.8 節で取り上げる。

34.7.4 まとめ

モデルのパラメータがサンプルの分布を支配するとき、期待損失の勾配をモンテカルロサンプリングで推定するには、まずその勾配を期待値の形に書き直さなければならない。そのための 2 つの方法はいずれも微分を積分の内側へと通すが、このステップには可積分な優関数による上界が必要であり、さらに積分区間の端点がパラメータとともに動く場合にはライプニッツ則の境界項も必要となる。スコア関数恒等式は対数密度を微分するものであり、任意の損失 g や離散確率変数に対して機能するが、サンプルの値が g の大きさに比例するため、適切なベースラインがなければノイズが大きくなりやすい。一方、パスワイズ恒等式は再パラメータ化されたサンプルを通して微分するものであり、微分可能な g を必要とするが、常にではないものの多くの場合ずっと低い分散を達成する。どちらの推定量も二乗平均平方根誤差 $\sigma/\sqrt{n}$ をもつため（36.5.1.7 節）、与えられた問題に対してどちらが望ましいかは、1 サンプルの標準偏差 σ によって決まる。

34.7.5 演習問題

1. **動く台：** X が区間 $[0,\theta]$（ただし $\theta > 0$）上の一様分布に従うとし、$\ell(\theta) = \mathbb{E}[X^2] = \int_0^\theta \frac{x^2}{\theta}\,dx = \frac{\theta^2}{3}$ かつ $\ell'(\theta) = \frac{2\theta}{3}$ とする。
   1. ℓ をライプニッツ則 Equation 34.72 を用いて微分せよ。その結果のどの部分が単純な順序交換 Equation 34.71 から得られ、どの部分が欠落するか。
   2. 台の上では $\log p_\theta(x) = -\log\theta$ である。スコア関数推定量 $X^2\,\partial_\theta \log p_\theta(X)$（Equation 34.74）の期待値を計算せよ。続いて g を $[0,\infty)$ 上の連続関数とする。$g(X)\,\partial_\theta \log p_\theta(X)$ の平均が、単純な順序交換 Equation 34.71 を $\mathbb{E}[g(X)] = \int_0^\theta g(x)\,p_\theta(x)\,dx$ に適用したものと等しいことを示せ。これが $\frac{d}{d\theta}\mathbb{E}[g(X)]$ の推定値としてもつ統計的バイアスは、$g(\theta)$ と θ を用いてどのように表されるか。
   3. $X = \theta U$（ただし U は $[0,1]$ 上の一様分布に従う）と表し、Equation 34.75 のパスワイズ推定量の期待値を計算せよ。X の台が θ とともに動くにもかかわらず、境界項が不要であるのはなぜか。
2. [code] **四次関数に対する推定量の分散：**●$X \sim \mathcal{N}(\theta,1)$ とし、$X = \theta + z$ と表し、$s(X) = \partial_\theta \log p_\theta(X) = z$ をスコアとする。
   1. $g(X) = X^4$ とおくと、$\frac{d}{d\theta}\mathbb{E}[X^4] = 4\theta^3 + 12\theta$ となる。1 サンプルのパスワイズ推定量を Equation 34.75 から、1 サンプルのスコア関数推定量を Equation 34.74 からそれぞれ導出せよ。演習問題 5 (b) の恒等式を用いて、両者の平均がいずれも $4\theta^3 + 12\theta$ となることを示せ。
   2. 定数ベースライン b に対し、$(g(X) - b)\,s(X)$ の分散が $b^\star = \mathbb{E}[g(X)s(X)^2]/\mathbb{E}[s(X)^2]$ によって最小化されることを示せ。$g(X) = X^2$ の場合に $b^\star = \theta^2 + 3$ となることを示し、さらに $\mathbb{E}[z^2] = 1$、$\mathbb{E}[z^4] = 3$、$\mathbb{E}[z^6] = 15$ を用いて、本文で挙げた 3 つの分散 4、$\theta^4 + 14\theta^2 + 15$、$8\theta^2 + 6$ を導出せよ。
   3. $b^\star$ を $g(X) = X^4$ に対して求めよ。10^6 個のサンプルを用い、$\theta \in \{0,1,2\}$ においてパスワイズ推定量、スコア関数推定量、ベースライン付きスコア関数推定量の標準偏差を推定せよ。どの θ においてベースラインによる改善効果が最も大きいか。また、パスワイズ推定量との差を埋めきることはあるか。g' と g の大きさから、これらの順

序関係を説明せよ。

4. $g(x) = \mathbf{1}(x > 0)$ を $\theta = 1.5$ において考える。本文では、パスワイズ推定量がほとんどすべてのサンプルで 0 となる一方、スコア関数推定量 $\mathbf{1}(x > 0)\, z$ は正しい平均 $\varphi(\theta) = 0.1295$ をもつことを示した。シミュレーションによって両者を確かめよ。

3. [extended] **コイン投げを通じた勾配：●●**x が離散的な選択であるとき、θ からサンプルへ至る微分可能な経路は存在しないため、パスワイズ推定量は存在しない。Gumbel-softmax (Jang et al. 2017; Maddison et al. 2017) のような連続緩和は微分可能な代理変数を構成し、ストレートスルー推定量 (Bengio, Léonard, et al. 2013) は順伝播でハードサンプルを用い、逆伝播でその代理変数を用いるが、いずれも元の目的関数に対して統計的バイアスをもつ。本問ではそのバイアスを測定する。Tucker et al. (2017) はこれを簡単な例題で示している。1 つのベルヌーイ変数 $b \in \{0, 1\}$ が $P(b = 1) = \sigma(\theta)$（ただし θ はロジット）を満たし、目的関数が $\ell(\theta) = \mathbb{E}[(b - t)^2]$（ターゲット $t = 0.45$）であるとする。U を $(0, 1)$ 上の一様確率変数とし、$\zeta = \log U - \log(1 - U)$ とおき、温度 $\tau > 0$ に対して $z = \sigma((\theta + \zeta)/\tau)$（Gumbel-softmax 緩和の 2 クラスの場合）とおく。

   1. $\ell(\theta)$ と $\ell'(\theta)$ を閉形式で求めよ。どの確率 $\sigma(\theta)$ が ℓ を最小化するか。また、有限の θ でそれは達成されるか。
   2. $b = \mathbf{1}(z > \tfrac{1}{2})$ が $P(b = 1) = \sigma(\theta)$ をすべての τ に対して満たすことを示せ。
   3. $\ell'(\theta)$ に対する 1 サンプル推定量を次の 3 通り実装せよ。すなわち、REINFORCE Equation 34.74（ベースラインは $\ell(\theta)$）、緩和推定量 $\frac{d}{d\theta}(z - t)^2$、そして損失にはハードサンプルを用い導関数には緩和サンプルを用いるストレートスルー推定量 $2(b - t)\frac{dz}{d\theta}$ である。$\ell'(\theta)$ と比較して各推定量の平均と標準偏差を、$\theta \in \{-2, 0, 2\}$ および $\tau \in \{1, 0.5, 0.1\}$ に対して表にまとめよ。各推定量のバイアスは θ と τ にどのように依存するか。
   4. θ に対する確率的勾配降下法を、初期値 $\theta = 0$、ステップサイズ 1 で 5000 ステップにわたり、REINFORCE、緩和推定量、ストレートスルー推定量のそれぞれ（各 τ について）で実行せよ。ただし REINFORCE のベースラインは過去の損失の指数移動平均に置き換えること。各実行に沿って $\ell(\theta)$ をプロットせよ。さらに緩和目的関数 $R_\tau(\theta) = \mathbb{E}[(z - t)^2]$ を U 上の数値積分によって計算し、その最小点を緩和推定量の実行が落ち着く θ の値と比較せよ。τ を小さくすると、緩和推定量のバイアスと分散はどのようなトレードオフを示すか。

34.8 関連文献

本章全体を通して、深層学習の実践では日常的に踏み越えられる見通しのよい仮定のもとで主要な結果を証明してきた。実際には、ReLU ネットワークには鋭い折れ点（kink）が至るところにあり、リバースモードのテープは容易に GPU メモリを超過し、1,000 次元の積分はグリッド上に配置できない。以下の短い解説では、こうした仮定が崩れたときに何が起こるかをまとめ、続いて本章の主要内容を扱う教科書、講義、チュートリアルを厳選して紹介する。なかでも日常的な訓練に直結するため最初に読んでおきたいトピックが 2 つある。1 つは、折れ点を経由する誤差逆伝播法（backpropagation）がどのような意味で依然として妥当であるかを説明する保存場（conservative field）であり、もう 1 つは、長い計算においてメモリを扱いやすい範囲に抑える再帰的チェックポイント保存（recursive checkpointing）である。本書の他の箇所で本節の内容を前提とするものはない。

34.8.1 折れ点における微分

34.8.1.1 保存場

34.1 節で例示したように、標準的な規約 $\mathrm{ReLU}'(0) = 0$ を採用すると奇妙な現象が生じる。恒等関数を $\mathrm{ReLU}(x) - \mathrm{ReLU}(-x) = x$ と表すと、誤差逆伝播法は傾き 0 を $x = 0$ において返すが、この値は x の真の導関数でもなければ有効な劣勾配（subgradient）ですらない。古典的な劣微分（subdifferential）ではこの挙動を説明できない。構成要素の関数が凸でなくなった途端、和の法則や連鎖律（chain rule）が集合の包含関係としてしか成り立たなくなるからである。この隔たりを埋めるため、Bolte and Pauwels (2021) は**保存場**を導入した。これは構成上そのまま連鎖律を満たし、測度ゼロの集合を除くいたるところで通常の勾配と一致する、導関数の集合値への一般化である。各折れ点における傾きの選択をどのように固定しても、誤差逆伝播法は常にそのような場の要素を返す。では、なぜ折れ点からなる例外集合は問題にならないほど小さいのだろうか。ラデマッハの定理（Rademacher's theorem）(Rademacher 1919) により、$\mathbb{R}^n$ 上のすべてのリプシッツ関数はほとんどいたるところで微分可能であるため、ReLU ネットワークの損失が持つ折れ点は測度ゼロの集合にとどまる。

同論文は、誤差逆伝播法によるこれらの値で駆動される確率的勾配降下法（stochastic gradient descent）が依然として収束するかという実務上の問いにも決着をつけている。標準的なニューラルネットワークの層から構成された損失に対し、反復点の有界性と、ステップサイズおよび勾配ノイズ（gradient noise）に関する通常の条件を仮定すれば、反復点は保存場がゼロを含む臨界点（critical point）の集合へと収束する。この保証は厳密演算のもとで成り立つものであるが、有限精度の浮動小数点演算では、反復点が実際に折れ点の上に着地することがある。34.1 節では、Bertoin et al. (2021) が計測し 2023 年の正誤表で修正した 3 つの数値精度にわたる発生頻度を紹介するとともに、$\mathrm{ReLU}'(0)$ の選び方が結果的に訓練へほとんど影響しないことを示している。

34.8.1.2 劣勾配法

滑らかな最適化は、勾配と逆向きの小さなステップが目的関数を下げることを保証する降下補題（descent lemma）に依拠している。滑らかでない関数ではこの保証が崩れる。$|x|$ の最小点において、有効な劣勾配 $\frac{1}{2}$ は実際には上り方向を指すことが 34.1 節に示されている。それでもなお驚くべきことに、劣勾配法（subgradient method）は収束する。その解析が目的関数値（objective value）そのものではなく、最小点（minimizer）までのユークリッド距離を追跡するからである。$\sum_{t} \eta_t = \infty$ かつ $\sum_{t} \eta_t^2 < \infty$ を満たす減衰ステップサイズを用いれば、それまでに得られた最良の目的関数値は最小値へと収束する。最適な収束率（convergence rate）を得るには別のスケジュールが必要となる。劣勾配のノルムが高々 G である凸関数に対し、最小点から距離 R の点を出発点として定数ステップサイズ $R/(G\sqrt{T})$ を用いると、T 個の反復点のうち最良のものは最小値から $RG/\sqrt{T}$ 以内に到達する。次元が T を超える場合、劣勾配のみに依拠するいかなる手法も、この収束率を定数倍を超えて改善することはできない (Nesterov 2018)。和の法則が等号で成り立つための厳密な条件を含め、劣微分法の体系的な展開については Rockafellar (1970) を参照されたい。

34.8.2 ベクトル写像と行列写像の微分

34.8.2.1 ベクトル値写像に対する平均値の定理

1 次元では、34.1 節の平均値の定理（mean value theorem）により、ある中間点 ξ が存在して $f(b) - f(a) = f'(\xi)(b - a)$ が成り立つことが保証される。ベクトル値曲線 $\gamma : \mathbb{R} \to \mathbb{R}^n$ に対しても同じ等式が成り立つと考えたくなるが、実際にはまったく成り立たない。単位円 $\gamma(t) = (\cos t, \sin t)$ を考えてみよう。$t = 2\pi$ で 1 周すると曲線は出発点に戻るため $\gamma(2\pi) - \gamma(0) = \mathbf{0}$ となるが、速度 $\gamma'(\xi)$ は単位長さを持ち、$\mathbf{0}$ となる ξ は一切存在しない。高次

元で生き残るのは、**平均値不等式**（mean value inequality）$\|\gamma(b)-\gamma(a)\| \le \sup_{\xi} \|\gamma'(\xi)\|\,|b-a|$ のみである（定理 5.19、Rudin (1976)）。幸い、34.2 節におけるテイラー展開の上界では、この定理のベクトル版は一切必要ない。スカラー値の損失を線分上に制限することで、通常の 1 変数の定理がそのまま適用できるからである。

34.8.2.2 行列値写像のヤコビ行列

34.4 節ではスカラー損失を微分するため、すべての勾配は都合よく対応する行列変数と同じ形状を持つ。しかし、行列値関数を行列入力に関して微分すると、完全な導関数は 4 つの添字を持つ配列になる。代数計算を線形で馴染みのある形に保つ標準的な方法は、列を縦に並べる作用素 vec で両方の行列を平坦化し、4 つの添字を持つテンソルを通常のヤコビ行列（Jacobian matrix）へと変換することである。この平坦化のもとでは、行列の全微分 $d\mathbf{F} = \mathbf{A}\,d\mathbf{X}\,\mathbf{B}$ は $d\,\mathrm{vec}\,\mathbf{F} = (\mathbf{B}^{\top} \otimes \mathbf{A})\,d\,\mathrm{vec}\,\mathbf{X}$ となり、連鎖律は単にクロネッカー積（Kronecker product）どうしの乗算になる。Mike B. Giles (2008) は、行列積、逆行列、行列式、および標準的な行列分解に対するフォワード・リバース両方向の微分規則を使いやすい一覧にまとめている。

34.8.3 自動微分

34.8.3.1 最適なヤコビ行列の累積

34.4 節で示したように、連鎖するヤコビ行列を掛ける順序を変えるだけで、計算コストはおよそ層の幅に相当する倍率だけ変化し得る。一直線に並んだ K 個の因子の積であれば、最も低コストな括弧の付け方を求める問題は古典的な連鎖行列積問題であり、動的計画法（dynamic programming）によって $O(K^3)$ 時間で解ける。しかし、分岐やスキップ接続を持つ一般の計算グラフ（computation graph）では、消去順序の探索空間がはるかに広がる。各中間ノードは、そこへ入る辺とそこから出る辺の上の偏微分を任意の順序で掛け合わせることで消去できるからである。乗算回数が最小となる消去順序を求める問題は、実は NP 完全であることが知られている (Naumann 2008)。そのため、深層学習フレームワークは消去順序の大域的な最適化を試みず、フォワードモード、リバースモード、およびそれらの合成を提供して、モードの選択をユーザーに委ねている。Griewank and Walther (2008) は、頂点・辺・面の消去を完全な一般性のもとで展開し、微分コンパイラで用いられるヒューリスティクスを概説している。

34.8.3.2 再帰的チェックポイント保存

34.5 節で扱った 1 段階のチェックポイント保存方式は、ピーク時の活性化メモリ（activation memory）を保存状態数 $T/s + s$ へと削減し、順伝播（forward pass）を 1 回追加するコストと引き換えに $2\sqrt{T}$（最適なブロックサイズ s のとき）まで抑える (Chen et al. 2016)。しかし、非常に長い系列や展開された反復ソルバでは、$O(\sqrt{T})$ のメモリでさえ大きすぎることがある。同じ考え方を再計算される各区間の内部に再帰的に適用すれば、メモリは一気に $O(\log T)$ まで下がり、必要な再計算は $O(T \log T)$ となる (Griewank 1992)。各ステップのコストが等しい場合、任意の固定されたメモリバジェットに対して Griewank の二項チェックポイント保存スケジュールが最適であることが証明されている (Griewank and Walther 2008)。実際には、チェックポイント保存のユーティリティは、指定されたブロックごとに 1 段階のチェックポイント保存を実装している。

PyTorch は `torch.utils.checkpoint` を提供している。

34.8.3.3 テイラーモード

k 階の方向微分（directional derivative）を求めるためにフォワードモードの自動微分（automatic differentiation）を k 回入れ子にするのは無駄が多い。入れ子を 1 段増やすごとに状態が倍増し、

1 つの値につき 2^k 個の成分を保持することになるからである。**テイラーモード**は、二重数（dual number）を打ち切りべき級数 $a_0 + a_1\varepsilon + \cdots + a_k\varepsilon^k$（ただし $\varepsilon^{k+1} = 0$）に置き換えることで、この指数的な肥大化を回避する。各基本演算は 1 回のパスで $k+1$ 個すべての係数を伝播する。具体的には、乗算は 2 つの係数系列を $O(k^2)$ 回の演算で畳み込み、exp、log、sin などの超越関数も同じ $O(k^2)$ のコストの漸化式から計算できる (Griewank and Walther 2008)。34.5 節にある、1 回の順伝播で二階導関数を計算する演習問題は $k = 2$ の場合にあたる。

34.8.3.4 フォワード勾配

リバースモードの微分では中間値のテープを保存する必要がある一方、フォワードモードではテープが不要になる代わりに、入力パラメータごとに 1 回のパスが必要となる。Baydin et al. (2022) は、ガウス分布に従うランダムな接ベクトル $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ に沿って単一の順伝播を実行し、$(\nabla \ell^\top \mathbf{v})\,\mathbf{v}$ を確率的勾配の推定値とすることで、この両方のコストを回避した。$\mathbb{E}[\mathbf{v}\mathbf{v}^\top] = \mathbf{I}$ であるためこの推定量は不偏であり、パラメータ数が n のとき平均二乗誤差（mean squared error）は $(n+1)\|\nabla \ell\|^2$ となる。メモリ上のテープは完全に不要になるが、そのトレードオフは厳しい。典型的な推定誤差は真の勾配のノルムのおよそ $\sqrt{n}$ 倍に達し、小規模なモデルでは許容できても、現代のニューラルネットワークのスケールでは現実的でない大きさになる。

34.8.4 積分

34.8.4.1 ルベーグ積分

リーマン積分は定義域を区間に分割するが、この方針は激しく不連続な関数に対して破綻する。34.6 節で示したように、有理数の指示関数では代表点付きの和が決して一定の値に落ち着かず、さらにリーマン可積分な関数列の各点極限でさえリーマン可積分にとどまるとは限らない。Lebesgue の着眼点は、定義域ではなく値域をスライスすることにある。すなわち、各高さの水準を f が上回る集合の大きさを測り、それらの測度を足し合わせるのである。有理数全体や極座標における半直線 $\theta = 0$ のような測度ゼロの集合は積分値に一切寄与しないため、有理数の指示関数はすっきりと 0 に積分される。34.6 節と 34.7 節のいずれにおいても、この理論の主力となる 3 つの定理、すなわちフビニ・トネリの定理、n 次元の変数変換（change of variables）、および無限領域上の積分記号下の微分を証明なしで用いた。これら 3 つの定理はすべて、極限と積分をいつ交換できるかを定める単調収束定理（monotone convergence theorem）と優収束定理（dominated convergence theorem）に基づいている。$f_k \to f$ が各点収束であり、かつ $|f_k| \le g$ が単一の可積分関数 g に対して成り立つならば、$\int f_k \to \int f$ となる (Folland 1999)。

34.8.4.2 準モンテカルロ法

36.5.1.7 節では、誤差が $N^{-2/d}$ で減衰する二次のグリッドと、次元によらず誤差が $N^{-1/2}$ で減衰するモンテカルロ積分を比較した。しかし、純粋な無作為標本では偶発的な密集や空白の偏りが避けられない。**低食い違い量**（low-discrepancy）点集合（ソボル列、ハルトン列、格子則など）は、単位超立方体をはるかに均一に埋め尽くすように構成されており、有界変動（Hardy と Krause の意味で）の被積分関数に対して $O(N^{-1}(\log N)^d)$ の誤差を達成する。ただし、純粋に最悪ケースの保証として見ると、この上界は厳しい現実を突きつける。定数因子を無視しても、この上界が $N^{-1/2}$ を下回るのは、数千点を超えてから（$d = 2$ のとき）、さらには 10^{15} 点を超えてから（$d = 5$ のとき）に限られる。それでも実際には、準モンテカルロ法（quasi-Monte Carlo）は高次元においてさえ通常のモンテカルロ法をしばしば上回る。現実の被積分関数は、多くの場合、一度に少数の有効な座標にしか強く依存しないからである。決定論的な点集合からはそのままではエラーバーが得られないため、**ランダム化準モンテカルロ法**（randomized quasi-Monte Carlo）では点集合にランダムなシフトやスクランブルを施し、独立な反復試行から不偏な分散推定値を得られるようにする。Owen (2013) の第 15〜17 章では、決定論的手法とランダム化手法の両方が解説されている。

34.8.4.3 スコア関数とパスワイズを超える勾配推定量

34.7 節では、スコア関数推定量とパスワイズ推定量を導出し、定数ベースラインを用いてスコア関数推定量の分散を低減した。より洗練された制御変量（control variate）は、モデル自体の構造を活用する。REBAR (Tucker et al. 2017) は、連続緩和（continuous relaxation）のパスワイズ勾配をスコア関数推定量から引き、その解析的な期待値を足し戻すことで、離散確率変数に対して推定量を不偏に保ったまま分散を下げる。Gumbel-softmax (Jang et al. 2017; Maddison et al. 2017) やストレートスルー推定器（straight-through estimator）(Bengio, Léonard, et al. 2013) のような連続緩和そのものは、元の離散的な目的関数に対する統計的バイアスが生じる代わりに低い分散を実現する。第 3 のアプローチである**測度値微分**（measure-valued derivative）は、確率密度の導関数を 2 つの密度のスケール付きの差として表し、両方からサンプリングすることで勾配を推定する。Mohamed et al. (2020) はこれら 3 つの系統を単一の枠組みに統合し、それぞれが適用できる条件を明示している。

34.8.5 書籍、講義、チュートリアル

書籍

- Deisenroth、Faisal、Ong による『Mathematics for Machine Learning』は、第 5 章（「Vector Calculus」）で機械学習に向けた勾配、ヤコビ行列、および連鎖律を扱っている（34.1 節、34.2 節）。
- Gilbert Strang による『Calculus』（MIT OpenCourseWare）は無料で公開されている教科書であり、積分、微分積分学の基本定理、および重積分に関する各章が 34.6 節の副読本となる。
- Petersen と Pedersen による『The Matrix Cookbook』は行列微分の恒等式をまとめた公式集であり、全微分法（34.4 節）で導出した結果を検算するのに役立つ。
- Griewank と Walther による『Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation』および Naumann による『The Art of Differentiating Computer Programs』は、フォワードモードとリバースモード、VJP と JVP、チェックポイント保存、および微分コンパイラに関する標準的な専門書である（34.5 節）。

講義とチュートリアル

- 3Blue1Brown による「Essence of Calculus」、および MIT OpenCourseWare の「Single Variable Calculus (18.01SC)」と「Multivariable Calculus (18.02SC)」。
- Edelman と Johnson による「Matrix Calculus for Machine Learning and Beyond (18.S096)」（MIT OCW）、Parr と Howard による「The Matrix Calculus You Need For Deep Learning」、および Laue、Mitterreiter、Giesen による matrixcalculus.org。
- Christopher Olah による「Calculus on Computational Graphs: Backpropagation」、Baydin らによる「Automatic Differentiation in Machine Learning: a Survey」、JAX ドキュメント内の「The Autodiff Cookbook」と「Autodidax」（34.5 節）、ならびにスコア関数（REINFORCE）推定量、パスワイズ再パラメータ化、制御変量を用いた分散低減（variance reduction）を扱う Mohamed らの「Monte Carlo Gradient Estimation in Machine Learning」（34.7 節）。

Chapter 35

最適化

本書に登場するすべてのニューラルネットワークは、同じ基本的なループで訓練される。すなわち、損失の勾配を計算し、その逆方向にステップを進め、これを繰り返す。第 9 章では、実際のネットワーク上でモメンタム、適応的ステップサイズ、学習率スケジュールがこのループにどう作用するかを実験的に確認したが、そこでは 3 つの疑問が未解決のまま残された。なぜ学習率には越えられない上限があるのか。その上限を十分に下回る安全なステップであっても、進行が極端に遅くなりうるのはなぜか。そして、なぜ勾配ノイズ（gradient noise）によって訓練が落ち着かなくなるのか。

最適化理論は、これらそれぞれの問いを目的関数の幾何構造と結びつける。最も急な方向の曲率が安全な最大ステップサイズを定め、最も急な曲率と最も緩やかな曲率の比が反復回数を定め、凸性（convexity）が停留点（stationary point）が大域的に最適かどうかを決定する。本章全体を通して、細長い鉢状の関数 $\frac{1}{2}(x_1^2 + 1000\,x_2^2)$ に繰り返し立ち返る。この関数はある方向がもう一方の方向より 1,000 倍急になっており、急な壁に沿って安全を保てるステップサイズでは、平坦な谷に沿った方向には反復点をわずかにしか動かせない。第 9 章が実践と計測に焦点を当てたのに対し、本章では自己完結した証明と背後にある幾何構造を与える。

本章は 3 部構成で展開する。まず厳密な勾配が得られる設定のもとで、35.1 節ではステップサイズの上限と反復回数を証明し、モメンタムによって曲率比がその平方根に置き換わる仕組みを示す。一方、35.3 節では停留点が大域的最小点となる条件を確立する。両者の間に位置する 35.2 節では、曲率を積極的にモデル化するニュートン法、準ニュートン法（quasi-Newton method）、信頼領域法（trust-region method）を扱う。次にノイズを含む勾配に目を転じ、35.4 節では固定ステップの SGD がステップサイズに比例するノイズフロアで停滞する理由を示し、35.5 節では座標ごとのスケーリングと AdamW をヘビーボールモメンタムと比較検証する。最後に、35.6 節と 35.7 節ではノルムバジェット（2.7 節）の背後にある KKT 条件、射影、近接ステップ、および双対性（duality）を展開し、35.8 節では浮動小数点精度、桁落ち、および条件性（conditioning）を解析する（5.5 節）。

本章では、勾配とヘッセ行列（34.2 節）、および固有値分解（33.3 節）に関する知識を前提とする。深層ネットワークの損失は大域的に滑らかでも凸でもないため、本章の定理は局所的なモデルとして機能するものであり、その適用限界は各節で明示する。なお、内点法（interior-point method）と作用素分割法（operator-splitting method）については 35.9 節で概観する。

記法

本章全体を通して、Table 35.1 にまとめた記法の慣例に従う。各節ではその節で頻繁に用いる記号を改めて示し、単一の節に固有の記法は初出箇所で導入する。

Table 35.1: 最適化の章で用いる記号。星印 $\star$ は最適点・最適値を、アスタリスク $*$ は凸共役を表す。行列は太字とする。

記号	意味
f、$\ell(\mathbf{w})$	目的関数、重みの関数としての訓練損失
$\mathbf{x}^\star$、$f^\star$	最小点と最小値
∇f、$\nabla^2 f$、$\mathbf{g}_k$	勾配、ヘッセ行列、ステップ k で用いる勾配（または確率的勾配）
L、μ	滑らかさの定数と強凸性の定数（$\mu\mathbf{I} \preceq \nabla^2 f \preceq L\mathbf{I}$）
$\kappa = L/\mu$、$\kappa(\mathbf{A})$	問題の条件数、行列の条件数 $\sigma_{\max}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$
λ_i、$\lambda_{\max}(\mathbf{A})$	ヘッセ行列または二次関数の行列 $\mathbf{A}$ の固有値
η、β	ステップサイズ（学習率）、モメンタム係数
k、t	反復の添字、適応的手法におけるオプティマイザのステップ
ϵ	目標精度、Adam において除算を有限に保つ定数
n、d、$\mathcal{B}$、b	データ例の数、変数の数、ミニバッチとそのサイズ
Σ	各データ例の勾配の共分散
$\mathbf{m}_t$、$\mathbf{v}_t$、β_1、β_2	Adam の一次・二次モーメント推定値、およびそれらの減衰率
$\mathbf{P}$	前処理行列（更新ステップは $-\eta\,\mathbf{P}^{-1}\mathbf{g}$）
g_i、h_j	不等式制約 $g_i(\mathbf{x}) \le 0$、等式制約 $h_j(\mathbf{x}) = 0$
λ、ν	不等式制約と等式制約の乗数（λ は減衰項 $\frac{\lambda}{2}\lVert\mathbf{w}\rVert^2$ の重みも表す）
$\mathcal{L}$、q	ラグランジアン、双対関数
$p^\star$、$d^\star$	主問題と双対問題の最適値
f^*	f の凸共役
Π_C、$\operatorname{prox}_{\psi}$	集合 C へのユークリッド射影、ψ の近接作用素
u、$\varepsilon_{\text{mach}}$	浮動小数点形式の丸めの単位と計算機イプシロン（$u = \varepsilon_{\text{mach}}/2$）

35.1 勾配降下法と曲率

実践において、学習率（learning rate）は試行錯誤で決められるのが普通である。適当な値を選び、損失の推移を観察し、訓練が発散すれば値を下げる。各ステップは損失の勾配と逆向きに進むため、勾配が方向を定め、学習率は進む距離だけを決める。しかし、この距離の選び方を誤ると、訓練は極端に遅くなるか、あるいは不安定になる。勾配に沿ったステップはどこまで信頼できるのか。そしてその限界によって、どれだけの反復回数が強いられるのだろうか。

わずか 2 変数の関数でも、この難しさはすでに露わになる。楕円状のボウル $f(x, y) = \frac{1}{2}(x^2 + 10\,y^2)$ を考えよう。この関数は y 方向に沿って x 方向より 10 倍急峻に曲がっている。初期点を $(2, 1)$ とすると、そこでは $f = 7$、$\nabla f = (2, 10)$ である。勾配と逆向きにサイズ η のステップを進むと、到達点は $(2 - 2\eta,\, 1 - 10\eta)$ となる。$\eta = 0.01$ では損失はほとんど動かず、6.01 に下がるにとどまる。$\eta = 0.1$ では 1.62 まで一気に減少する。ところが $\eta = 0.25$ では、逆に 12.4 へと**上昇する**。これらのステップはすべて下り坂を向いていたにもかかわらず、ステップ長を変えただけでこの違いが生じる。

原因は曲率（curvature）にある。急峻な y 軸に沿って安全を保てるほど短いステップは、平坦な x 軸に沿っては這うようにしか進まない。最良の固定学習率を選んでも、69 回の反復を費やしてようやく誤差が 10^6 分の 1 に縮まる（6 桁の精度に相当する）。曲率の比を 1000 に広げると（本章の通し例である $\tfrac{1}{2}(x_1^2 + 1000\,x_2^2)$）、勾配降下法（gradient descent）は $6{,}908$ ステップを要する。ここにモメンタムを加えれば、その回数は 315 まで減る。以下では、ヘッセ行列が $\mathrm{diag}(1, 10)$ でどの数値も検算しやすい緩やかなボウルを手計算で扱い、より硬い通し例については対応するステップ数を併記する。

解析は 3 段階で進める。第 1 に、滑らかさ（smoothness）を定式化して 34.2 節の局所線形近似を有限長のステップに対する保証へと引き上げ、降下補題（descent lemma）Equation 35.5 を導く。第 2 に、二次関数上で勾配降下法を厳密に解き、曲率の比が条件数（condition number）として現れて最適ステップ Equation 35.10 を定めることを見る。最後にモメンタムを導入し、その $\sqrt{\kappa}$ 収束則 Equation 35.15 を導出する。本節を通して扱う手法はいずれも曲率を尊重するにとどまり、曲率そのものをモデル化することはしない。ニュートン法のように曲率を直接モデル化する手法は 35.2 節で扱う。また、本節では厳密な勾配が得られるものと仮定し、ミニバッチ訓練におけるノイズを含んだ勾配は 35.4 節に譲る。以下のコードはすべて素の NumPy で書かれているため、各更新式を数式と直接照らし合わせて読むことができる。

35.1.1 降下方向と滑らかさ

すべての勾配ステップは、どの方向へ進むか、そしてどこまで進むかという 2 つの問いに答える必要がある。第 1 の問いには、一次のテイラー展開から直ちに答えが出る。第 2 の問いに答えるには、勾配そのものがどれほど速く変化しうるかを知る必要がある。この仮定こそが、局所的な線形近似を有限長のステップに対する保証へと変える。

35.1.1.1 最急降下方向

冒頭の例では、他に優れた方向がないか問うことなく $-\nabla f$ に沿って進んだ。なぜ負の勾配が自然な選択なのかを見るため、$f:\mathbb{R}^d\to\mathbb{R}$ を一次まで展開しよう。小さなステップサイズ $\eta>0$ で方向 $\mathbf{d}$ に進むと、関数値は次のように変化する。

$$f(\mathbf{x}+\eta\,\mathbf{d}) = f(\mathbf{x}) + \eta\,\nabla f(\mathbf{x})^{\top}\mathbf{d} + o(\eta).$$

一次近似の範囲では、変化量は単に η と方向微分（directional derivative）$\nabla f(\mathbf{x})^{\top}\mathbf{d}$ の積である。ベクトル $\mathbf{d}$ は、点 $\mathbf{x}$ において $\nabla f(\mathbf{x})^{\top}\mathbf{d}<0$ を満たすとき、その点における**降下方向**（descent direction）と呼ばれる。

命題（降下方向と最急降下方向）。 *f は点 $\mathbf{x}$ で微分可能であり、$\nabla f(\mathbf{x})\neq\mathbf{0}$ を満たすとする。*

1. *$\nabla f(\mathbf{x})^{\top}\mathbf{d}<0$ ならば、$f(\mathbf{x}+\eta\,\mathbf{d})<f(\mathbf{x})$ が十分小さなすべての $\eta>0$ に対して成り立つ。*
2. *すべての単位ベクトルの中で、方向微分を最小化する方向は正規化された負の勾配に一意に定まる。*

$$\begin{aligned} \min_{\|\mathbf{d}\|=1} \nabla f(\mathbf{x})^{\top}\mathbf{d} &= -\|\nabla f(\mathbf{x})\|, \\ \text{attained only at } \mathbf{d} &= -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}. \end{aligned} \tag{35.1}$$

証明。 第 1 の主張を示すため、$\varphi(\eta)=f(\mathbf{x}+\eta\,\mathbf{d})$ とおく。連鎖律により $\varphi'(0)=\nabla f(\mathbf{x})^{\top}\mathbf{d}<0$ であるから、差分商 $(\varphi(\eta)-\varphi(0))/\eta$ は十分小さなすべての $\eta>0$ に対して真に負となり、し

たがって $\varphi(\eta) < \varphi(0)$ が従う。第 2 の主張については、コーシー・シュワルツの不等式より $\nabla f(\mathbf{x})^{\top}\mathbf{d} \geq -\|\nabla f(\mathbf{x})\|\,\|\mathbf{d}\| = -\|\nabla f(\mathbf{x})\|$ がすべての単位ベクトル $\mathbf{d}$ に対して成り立ち、等号が成立するのは $\mathbf{d}$ が $\nabla f(\mathbf{x})$ の負のスカラー倍であるとき、かつそのときに限る。そのような単位ベクトルは $-\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$ に一意に定まり、これが下界を達成する。■

冒頭のボウルでは、$(2,1)$ における最急降下の単位方向ベクトルは $-(2,10)/\sqrt{104} = (-0.196,-0.981)$ であり、その傾きは $-\|\nabla f\| = -10.198$ となる。座標軸方向 $(-1,0)$ も下り坂を向いているが、傾きは -2 とずっと緩やかである。実際、降下方向は $-\nabla f$ と鋭角をなすベクトル全体の開半空間を占めており、負の勾配はその中で最も急な方向を選び出しているにすぎない。この結果はあくまで傾き（無限小ステップの極限）を記述するものであるため、次に 4 つの単位方向について、小さな有限ステップ $\eta = 0.01$ における実際の減少量をこれらの傾きがどれほど忠実に予測するか確かめよう。

```
def f(v):
    return 0.5 * (v[0]**2 + 10 * v[1]**2)

def grad(v):
    return np.array([v[0], 10 * v[1]])

x = np.array([2.0, 1.0])
g = grad(x)
steepest = -g / np.linalg.norm(g)
candidates = [('steepest -grad/|grad|', steepest),
              ('diagonal -(1,1)/sqrt(2)', -np.ones(2) / np.sqrt(2)),
              ('axis     (-1,0)', np.array([-1.0, 0.0])),
              ('uphill   +(1,1)/sqrt(2)', np.ones(2) / np.sqrt(2))]
eta = 0.01
for name, d in candidates:
    print(f'{name}:  slope = {g @ d:+8.4f},  '
          f'f(x + eta d) - f(x) = {f(x + eta * d) - f(x):+.6f}')
```

```
steepest -grad/|grad|:  slope = -10.1980,  f(x + eta d) - f(x) = -0.101498
diagonal -(1,1)/sqrt(2):  slope =  -8.4853,  f(x + eta d) - f(x) = -0.084578
axis     (-1,0):  slope =  -2.0000,  f(x + eta d) - f(x) = -0.019950
uphill   +(1,1)/sqrt(2):  slope =  +8.4853,  f(x + eta d) - f(x) = +0.085128
```

$\eta = 0.01$ では、どの方向でも実際の変化量が η と方向傾きの積に 1% 以内で一致しており、短いステップに対しては一次モデルが信頼できる指針となる。微分可能性が保証するのはそこまで、すなわち十分小さなステップで関数値が減少することだけである。では「十分小さい」とは具体的にどれほど小さいのか。次にこの問いへ進もう。

35.1.1.2 滑らかさと降下補題

勾配降下法は、固定された**ステップサイズ**（または学習率）$\eta > 0$ を用いて最急降下ステップを反復する。

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\,\nabla f(\mathbf{x}_k). \tag{35.2}$$

本節全体を通して、k は反復の添字を表し、$\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ と略記する。一次のテイラー展開は方向 $-\mathbf{g}_k$ の妥当性を示したが、それをどこまで遠くまで信頼してよいかについては何も語らない。反復点が移動して勾配が大きく変化してしまえば、もはや $\mathbf{g}_k$ は f を適切に近似しない。有限のステップサイズに保証を与えるには、勾配が変化しうる速さを上から抑える必要がある。

定義（L 滑らかさ）。微分可能な関数 f が **L 滑らか**であるとは、単位距離あたりの勾配の変化量が高々 L に抑えられることをいう。

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \;\le\; L\,\|\mathbf{x}-\mathbf{y}\| \qquad \text{for all } \mathbf{x},\mathbf{y}. \tag{35.3}$$

f が 2 回連続微分可能であるとき、L 滑らかであることは、ヘッセ行列のすべての固有値が $[-L, L]$ に収まることと同値である。固有値が有界であれば Equation 35.3 が成り立つことは、ヘッセ行列を $\mathbf{y}$ から $\mathbf{x}$ への線分に沿って積分すればわかり、逆は差分商の極限をとることで従う（演習問題 2a）。要するに、L はあらゆる方向の曲率を上から抑える定数である。冒頭のボウルは定数のヘッセ行列 $\operatorname{diag}(1, 10)$ を持つため、$L = 10$ である。対照的に、曲率が際限なく増大する関数は大域的な滑らかさの定数を持たない。たとえば $\tfrac{1}{4}x^4$ は $f''(x) = 3x^2$ となるため、有限な大域的定数 L は存在しない。そのような場合は、後ほどバックトラッキング直線探索（backtracking line search）を用いて対処する。

滑らかさは、局所的な接平面を大域的な上界、すなわち二次上界（quadratic ceiling）へと引き上げる。

$$f(\mathbf{y}) \;\le\; f(\mathbf{x}) + \nabla f(\mathbf{x})^{\top}(\mathbf{y}-\mathbf{x}) + \tfrac{L}{2}\,\|\mathbf{y}-\mathbf{x}\|^2. \tag{35.4}$$

幾何学的には、f のグラフは点 $\mathbf{x}$ における接平面から、距離の二乗の $\tfrac{L}{2}$ 倍を超えて上方に外れることは決してない。この不等式を証明する前に、勾配ステップ $\mathbf{y} = \mathbf{x} - \eta\,\mathbf{g}$ を代入して 2 つの項の競合を見てみよう。一次の線形減少量 $\eta\,\|\mathbf{g}\|^2$ は η に比例して大きくなるのに対し、二次の曲率ペナルティ $\tfrac{L}{2}\eta^2\|\mathbf{g}\|^2$ は η^2 に比例して増大する。小さな η では線形減少が勝るが、$\eta = 2/L$ で両項はちょうど打ち消し合う。

命題（降下補題）。 *f が L 滑らかであるならば、勾配ステップ $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\,\nabla f(\mathbf{x}_k)$ について、$0 < \eta \le 2/L$ ならば*

$$f(\mathbf{x}_{k+1}) \;\le\; f(\mathbf{x}_k) - \eta\left(1 - \tfrac{L\eta}{2}\right)\|\mathbf{g}_k\|^2. \tag{35.5}$$

特に、f は $0 < \eta < 2/L$ かつ $\mathbf{g}_k \neq \mathbf{0}$ である限り真に減少する。保証される減少量は $\eta = 1/L$ で最大となり、そのとき $f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) - \tfrac{1}{2L}\|\mathbf{g}_k\|^2$ が成り立つ。

証明。まず二次上界 Equation 35.4 を示す。f を $\mathbf{x}$ と $\mathbf{y}$ を結ぶ線分上に制限して $\psi(t) = f(\mathbf{x} + t(\mathbf{y}-\mathbf{x}))$（$t \in [0,1]$）とおくと、$\psi'(t) = \nabla f(\mathbf{x}+t(\mathbf{y}-\mathbf{x}))^{\top}(\mathbf{y}-\mathbf{x})$ となる。微分積分学の基本定理、コーシー・シュワルツの不等式、およびリプシッツ条件 Equation 35.3 を適用すると、次式が得られる。

$$\begin{aligned} f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^{\top}(\mathbf{y}-\mathbf{x}) &= \int_0^1 \big(\nabla f(\mathbf{x}+t(\mathbf{y}-\mathbf{x})) - \nabla f(\mathbf{x})\big)^{\top}(\mathbf{y}-\mathbf{x})\,dt \\ &\le \int_0^1 L\,t\,\|\mathbf{y}-\mathbf{x}\|^2\,dt = \tfrac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2. \end{aligned}$$

ここに勾配ステップ $\mathbf{y} = \mathbf{x}_k - \eta\,\mathbf{g}_k$ を代入すると、次式を得る。

$$f(\mathbf{x}_{k+1}) \;\le\; f(\mathbf{x}_k) - \eta\,\|\mathbf{g}_k\|^2 + \tfrac{L}{2}\,\eta^2\,\|\mathbf{g}_k\|^2.$$

係数 $\eta(1 - L\eta/2)$ が真に正となるのは $0 < \eta < 2/L$ のとき、かつそのときに限り、$\eta = 1/L$ で最大値をとる。■

なお、この証明は $\mathbf{x}_k$ と $\mathbf{x}_{k+1}$ への線分上の各点との間で勾配を比較しているだけなので、∇f がその線分上で L リプシッツ連続であれば十分である。冒頭のボウル（$\|\mathbf{g}\|^2 = 104$、$L = 10$）でこの補題を試してみよう。$\eta = 1/L = 0.1$ では、上界により $f \le 7 - 104/20 = 1.8$ が保証される。$\eta = 0.01$ では $f \le 7 - 104 \cdot 0.01 \cdot 0.95 = 6.012$ が保証される。$\eta = 0.25 > 2/L$ では因子 $1 - L\eta/2$ が負に転じ、上界が初期値を上回ってしまうため、もはや何の保証も与えない。これら 3 つのステップが実際に到達する関数値と比べると、これらの上界はどれほどタイトだろうか。

```
x, L = np.array([2.0, 1.0]), 10.0
g = grad(x)
for eta in (0.01, 0.1, 0.25):
    bound = f(x) - eta * (1 - L * eta / 2) * (g @ g)
    print(f'eta = {eta:4.2f}:  bound {bound:7.4f},  actual {f(x - eta * g):7.4f}')
```

```
eta = 0.01:  bound  6.0120,  actual  6.0102
eta = 0.10:  bound  1.8000,  actual  1.6200
eta = 0.25:  bound 13.5000,  actual 12.3750
```

安定な 2 つのステップでは、上界は驚くほどタイトである。6.0102 は 6.012 のすぐ下にあり、1.62 も 1.8 をきれいに下回っている。$\eta = 0.25$ では上界 13.5 は初期値 $f = 7$ よりはるかに大きく、実際このステップはオーバーシュートして 12.375 に達する。Figure 35.1 に両曲線を η の関数としてプロットした。探索方向に沿っては $\mathbf{g}^\top \mathrm{diag}(1, 10)\,\mathbf{g} = 1004$ であるから、真の目的関数は放物線 $7 - 104\,\eta + 502\,\eta^2$ となる一方、二次上界は $7 - 104\,\eta + 520\,\eta^2$ となる。2 つの放物線がこれほど密着しているのは、この勾配がほぼ硬い軸に沿っており、その方向の曲率が L に極めて近いためである。

要するに、ステップ $\eta = 1/L$ は二次上界を最小化し、f はその上界の下に収まることが保証される。34.1 節では、これと同じ幾何学的議論を 1 変数で展開している。

35.1.1.3 凸性を仮定しない停留点

驚くべきことに、凸性（convexity）を一切仮定しなくても、降下補題だけで長期的な反復に定量的な制約が課される。すなわち、勾配がいつまでも大きいままでいることはできない。

命題（勾配降下法は近似停留点を見つける）。 **f は L 滑らかであり、下から f_{low} で抑えられるとし、Equation 35.2 を $\eta = 1/L$ で実行する。このとき、すべての $K \ge 1$ に対して次式が成り立つ。**

$$\min_{0 \le k < K} \|\nabla f(\mathbf{x}_k)\|^2 \;\le\; \frac{2L\,(f(\mathbf{x}_0) - f_{\mathrm{low}})}{K}. \tag{35.6}$$

証明。 $\eta = 1/L$ のとき、降下補題を変形すると $\|\mathbf{g}_k\|^2 \le 2L\,(f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}))$ となる。$k = 0, \dots, K-1$ にわたって和をとると、右辺は望遠鏡和となって相殺し、$2L\,(f(\mathbf{x}_0) - f(\mathbf{x}_K)) \le 2L\,(f(\mathbf{x}_0) - f_{\mathrm{low}})$ となる。K 個の数の最小値はその平均以下であるから、両辺を K で割れば求める上界が得られる。■

したがって、最初の K 個の反復点のうち少なくとも 1 つは勾配ノルム（gradient norm）の二乗が $O(1/K)$ となり、これには f の下界と滑らかさの定数しか必要としない。ただし、この結果

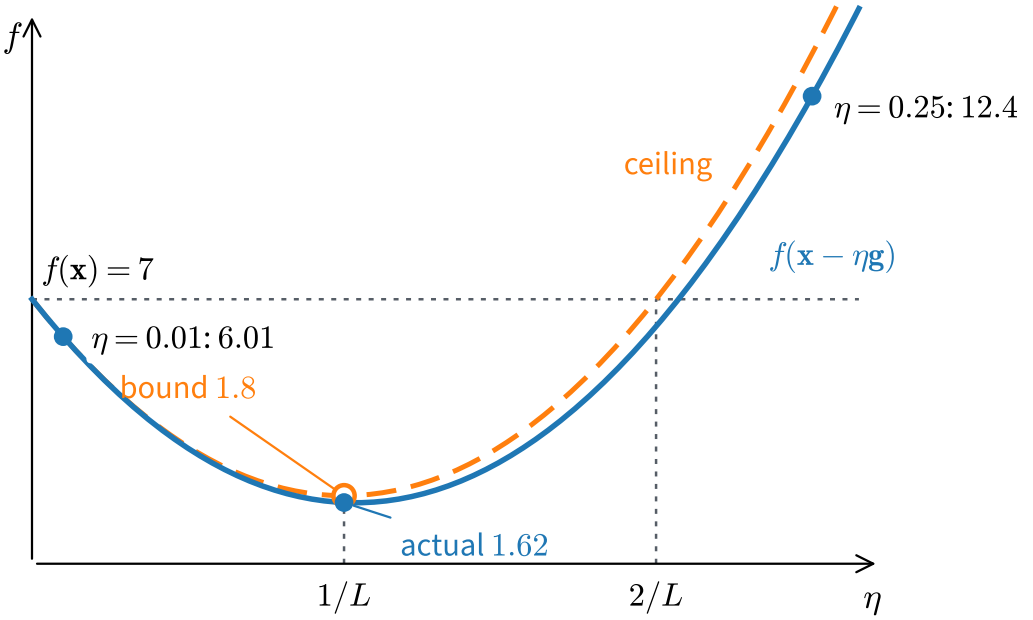


Figure 35.1: 冒頭の例における降下補題。$-\mathbf{g}$ に沿って $(2, 1)$ から進むとき、真の目的関数は下の放物線、二次上界 $f - \eta\|\mathbf{g}\|^2 + \frac{L}{2}\eta^2\|\mathbf{g}\|^2$ は上の放物線であり、$\eta = 0$ で互いに接する。上界を $\eta = 1/L = 0.1$ で最小化すると $f \le 1.8$ が保証され、実際のステップは 1.62 に達する。$2/L = 0.2$ を超えると上界は初期値を上回り何も保証しなくなる。$\eta = 0.25$ では目的関数は 12.4 まで上昇する。

は大域的な L の存在を仮定している点に留意しておこう。ReLU ネットワークは滑らかではないため、そこではこの上界は形式的な保証ではなく理想化されたモデルとして機能する。

しかし、勾配が小さいからといって最小点であるとは限らない。関数 $f(x, y) = \frac{1}{2}x^2 + \cos y$ を考えよう。この関数は下から -1 で抑えられ、かつ 1 滑らかである（ヘッセ行列 $\mathrm{diag}(1, -\cos y)$ の固有値が $[-1, 1]$ に収まる）。勾配降下法を $\eta = 1$ として $(1, 0)$ から開始すると、1 ステップで x は 0 に到達し、y は一切変化しない。というのも、$\partial f/\partial y = -\sin y$ が $y = 0$ でゼロになるからである。アルゴリズムは $(0, 0)$ で $f = 1$ のまま停止してしまう。ここは f が x 方向に上向き、y 方向に下向きに曲がる鞍点（saddle point）であり、真の最小値 -1（点 $(0, \pi)$ で達成）からは程遠い。鞍点でも勾配はゼロになるため定理自体は成り立っているが、到達したのがどの種類の停留点（stationary point）なのかまでは判別できない。この停滞自体は初期点の偶然によるものである。代わりに $(1, y_0)$（任意の小さな $y_0 \neq 0$）から開始すれば、各ステップ $y \leftarrow y + \eta \sin y$ によって y は 0 から離れていき、やがて最小点に落ち着く。一般に、負の曲率を持つ方向を備えた鞍点へ勾配降下法が収束するような初期点の集合は測度 0 であるから（9.1.2 節）、ランダム初期化（random initialization）を用いれば鞍点への収束は避けられる（ただし反復点がその近傍に長時間滞留することはある）。凸目的関数（convex objective）であれば、この問題はそもそも生じない。すべての停留点が自動的に大域的最小点（global minimum）となるからである（35.3 節）。

35.1.1.4 バックトラッキング直線探索

$\eta = 1/L$ を使うには、L の値を知っている必要がある。しかし実際にはそれが既知であることはまれであり、仮に大域的な L が存在したとしても、局所的な曲率は損失ランドスケープ（loss landscape）の場所によって変化するため、単一の大域定数では保守的になりすぎる。**直線探索**（line search）は、実行時にステップサイズを適応的に調整する手法である。候補ステップ η が与えられたとき、実際の減少量が一次モデルの予測する減少量の少なくとも一定の割合 $c \in (0, 1)$ に達した場合に限りそのステップを受理する。これが **Armijo 条件**（十分減少条件、sufficient decrease condition）である (Armijo 1966)。

$$f\big(\mathbf{x} - \eta \nabla f(\mathbf{x})\big) \;\le\; f(\mathbf{x}) - c\,\eta \,\|\nabla f(\mathbf{x})\|^2. \tag{35.7}$$

右辺は、テイラー線形予測の傾きを c 倍に緩めたものにすぎない。**バックトラッキング**は楽観的な試行ステップ η_0 から出発し、η を繰り返し半分にしながら Equation 35.7 が満たされるまで試す。降下補題により、$1 - L\eta/2 \ge c$、すなわち $\eta \le 2(1-c)/L$ となった時点でこの判定は必ず通過する。したがってループは常に有限回の半減で終了し、理論上のしきい値の 2 分の 1 以上にあたるステップ $\eta \ge \min\big(\eta_0, (1-c)/L\big)$ を、事前に L を一切知ることなく返す。

さらに都合がよいことに、この保証には反復点が実際に訪れる領域での L さえあれば足りる。四次関数 $f(x) = \frac{1}{4}x^4$（大域的な L を持たない）を考え、$x_0 = 3$、$c = \frac{1}{2}$ から開始しよう。バックトラッキングは f を決して増加させないため、すべての反復点は $[-3, 3]$ にとどまり、そこでは $f''(x) = 3x^2 \le 27$ が成り立つ。$\eta \le 1/27$ の試行ステップは x を $x(1 - \eta x^2)$ へ移すが、この点は $\frac{2}{3}x$ と x の間に位置するため線分全体が $[-3, 3]$ の内部に収まり、局所的には $L = 27$ で十分となる。この四次関数の曲率は原点から離れると急峻になり、ゼロ付近では消えるため、軌跡全体に適した単一の固定ステップサイズは存在しない。$x_0 = 3$ から両戦略を比較してみよう。一方は固定ステップ $\eta = 0.3$（原点付近であれば完全に安全な値）、もう一方は $\eta_0 = 1$ から始めるバックトラッキングである。

```
quartic = lambda v: 0.25 * (v ** 4).sum()
quartic_grad = lambda v: v ** 3
```

```
x = np.array([3.0])
for _ in range(6):                              # fixed step, too large at x0
    x = x - 0.3 * quartic_grad(x)
print(f'fixed eta = 0.3 from x0 = 3:  |x_6| = {abs(x[0]):.2e}  (diverged)')

x, etas = np.array([3.0]), []
for _ in range(25):                             # backtracking line search
    g = quartic_grad(x)
    eta = 1.0
    while quartic(x - eta * g) > quartic(x) - 0.5 * eta * (g * g).sum():
        eta *= 0.5                              # Armijo fails: halve the step
    x = x - eta * g
    etas.append(eta)
print('accepted steps:', np.array(etas[:8]), '...')
print(f'backtracking from x0 = 3:  f(x_25) = {quartic(x):.2e}  (monotone descent)')
```

```
fixed eta = 0.3 from x0 = 3:  |x_6| = 6.52e+103  (diverged)
accepted steps: [0.03125 0.0625  0.125   0.25    0.5     1.      1.      1.     ] ...
backtracking from x0 = 3:  f(x_25) = 1.09e-04  (monotone descent)
```

固定ステップは直ちに爆発し、わずか 6 回の反復で $|x_6| \approx 6.5 \times 10^{103}$ に達する。更新 $x \mapsto x(1-\eta x^2)$ が $|x|$ を縮小させるのは $\eta x^2 < 2$ のとき、かつそのときに限るため、$x_0 = 3$ からの固定ステップは $\eta < 2/9 \approx 0.22$ で収束し、それを超えると発散する。この安定性のしきい値が $2/9$ であり、$2/f''(x_0) = 2/27$ ではないことに注目しよう。$2/f''(x_0)$ という規則は二次関数のものである。原点へ向かうステップが受けるのは、f'' の 0 から x までの平均値（すなわち x^2）であり、最大値 $3x^2$ ではない。降下補題は線分上の最悪ケースの曲率に備えなければならないため、真の安全範囲の 3 分の 1 しか保証できないのである。

対照的に、バックトラッキングは最初のステップで 0.031 を受理する。その直前の試行値 0.0625 は Armijo 条件を満たさない。この四次関数において Armijo 条件が成り立つのは $\eta x^2 \le 0.456$（おおよそ $\eta \le 0.05$、$x_0 = 3$ のとき）のとき、かつそのときに限るからである。反復点が原点に近づいて曲率が平坦になるにつれ、直線探索がその場その場の局所曲率を探り当てるため、受理されるステップは自動的に大きくなっていく（$0.0625, 0.125, 0.25, 0.5, 1.0$ と増え、いずれも $0.3/x^2$ に近い）。もっとも、深層学習の訓練でバックトラッキングが使われることはほとんどない。ミニバッチ損失はノイズが大きすぎて 2 つの候補ステップを信頼性よく比較できないうえ、試行のたびに余分な順伝播のコストがかかるからである。代わりに深層学習の訓練では、通常スケジュールによってステップサイズを事前に指定する（9.7 節）。

35.1.1.5 緩和された滑らかさと勾配クリッピング

スケジュールでは、反復点が高曲率の領域に入ったときに反応できない。そのために訓練で用いられる安全策が勾配クリッピング（gradient clipping）であり、先ほどの四次関数はその仕組みをよく示している。曲率 $f''(x) = 3x^2$ は非有界だが、それが大きくなるのは勾配 $f'(x) = x^3$ も大きい場所に限られる。実際、$3x^2 \le 4 + |x|^3$ がすべての x に対して成り立つ。2 回微分可能な関数は、その曲率がこのように勾配によって抑えられるとき、(L_0, L_1) **滑らか**であるという (Zhang, He, et al. 2020)。

$$\|\nabla^2 f(\mathbf{x})\| \;\le\; L_0 + L_1 \|\nabla f(\mathbf{x})\| \qquad \text{for all } \mathbf{x}, \tag{35.8}$$

ここで左辺のノルムは、ヘッセ行列の固有値の絶対値の最大値を表す。$L_1 = 0$ とおけば L 滑らかさに戻り、四次関数は $L_0 = 4$ および $L_1 = 1$ でこの条件を満たす。この条件のもとで降下補

題が保証できるステップは、もはや定数 $1/L$ ではない。オーダー $1/(L_0 + L_1\|\mathbf{g}\|)$、すなわち勾配が小さいところでは一定のステップサイズ、勾配が大きいところでは一定の**距離** $\eta\,\|\mathbf{g}\|$（オーダー $1/L_1$）となる。$\mathbf{g}$ を $\min(1, c/\|\mathbf{g}\|)\,\mathbf{g}$（しきい値 c）に置き換える勾配クリッピングは、追加コストなしでこの形のステップを実現する。四次関数で $c = 1$ とすると、先ほど発散したステップ $\eta = 0.3$ でも、反復ごとに x を 0.3 ずつ動かして $|x| \le 1$ に到達させ、そこから収束する（演習問題 3）。

35.1.2 二次モデルと条件数

冒頭では、一方のボウルでは 69 ステップで済むのに、もう一方ではなぜ 6,908 ステップもかかるのかを説明すると約束した。どちらの目的関数も二次関数であり、二次関数上では勾配降下法を閉じた形で解くことができる。驚くべきことに、反復回数はたった 1 つの量、すなわち最大曲率と最小曲率の比に帰着する。

35.1.2.1 悪条件な谷

代数計算に入る前に、2 つの二次関数における勾配降下法を対比した Figure 35.2 を見てみよう。丸いボウル上では、反復点は最小点へ向かってほぼ一直線に進む。一方、細長い谷では壁が急峻で谷底が平坦になっている。壁が課す安定性の上限に近いステップをとると、谷を飛び越えて壁から壁へと跳ね返るばかりで、谷底に沿ってはほんの少しずつしか進まない。その結果、典型的なジグザグの軌跡が生じる。（図の谷は $\mathrm{diag}(1, 20)$ に対して $\eta = 0.097$ と、安定性の上限 0.1 のすぐ下に設定したものであり、キャプションに記した縮小率が生じる。）

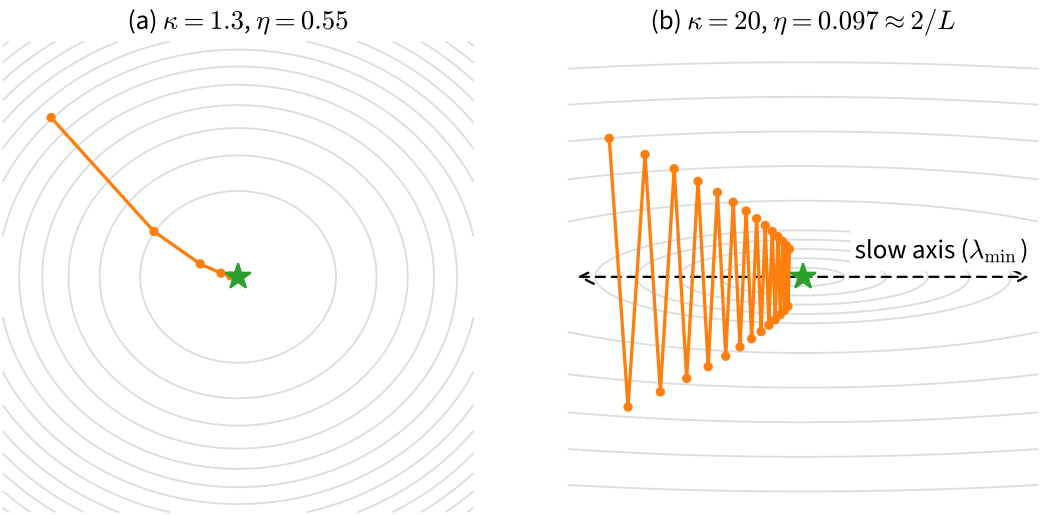


Figure 35.2: 2 つの二次関数 $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{A}\mathbf{x}$ における勾配降下法。左：良条件なボウルでは、反復点は最小点へほぼ一直線に向かう。右：悪条件な谷では、安定性の上限に近いステップをとると急峻なモードがオーバーシュートして符号を反転させ（因子 -0.94）、遅い軸はステップごとに 0.90 しか縮小しない。

二次関数を徹底的に解析する価値があるのは、あらゆる滑らかな関数が狭義極小点の近傍では二次関数のように振る舞うからである。最小点（minimizer）$\mathbf{x}^\star$ において正定値（positive definite）のヘッセ行列 $\mathbf{A}$ を持つとき、二次のテイラー展開は f を $f(\mathbf{x}^\star) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^\star)^\top\mathbf{A}\,(\mathbf{x} - \mathbf{x}^\star)$ で近似する。最小点を原点へ平行移動し、定数項を除けば、次のモデル問題が得られる。

$f(\mathbf{x}) = \tfrac{1}{2}\,\mathbf{x}^{\top}\mathbf{A}\,\mathbf{x}$,

$\mathbf{A}$ symmetric positive definite, with eigenvalues $0 < \lambda_{\min} = \lambda_1 \le \cdots \le \lambda_d = \lambda_{\max}$.

その勾配は $\mathbf{Ax}$、ヘッセ行列は $\mathbf{A}$ であるから、滑らかさの定数は単に $L = \lambda_{\max}$ となる。最大曲率と最小曲率の比を次式で定義する。

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}},$$

これがこの問題の**条件数**である。冒頭のボウルでは 10（$\mathrm{diag}(1, 10)$）、本章の通し例では 10^3 となる。$\mathbf{A}$ は対称正定値行列であるため、κ は特異値の比 $\kappa(\mathbf{A})$（33.5.3.2 節で導入）と一致する。

35.1.2.2 モードごとの縮小と安定性の上限

Figure 35.2 のジグザグ挙動は、最急降下方向が課すステップサイズの上限から直接生じている。その上限を正確に特定し、それが平坦な方向の進行をどれほど遅らせるかを見るため、二次関数上の勾配降下法を厳密に解こう。まずはいつもどおり 1 変数の場合から始める。

$f(u) = \frac{\lambda}{2}u^2$（$\lambda > 0$）とする。1 回の勾配ステップは u_k を $u_{k+1} = u_k - \eta\lambda u_k$ へ移すので、次式が成り立つ。

$$u_{k+1} = (1 - \eta\lambda)\,u_k, \qquad u_k = (1 - \eta\lambda)^k\,u_0.$$

反復のたびに、誤差には同じスカラー因子 $1 - \eta\lambda$ が掛けられるだけである。収束するのは $|1 - \eta\lambda| < 1$、すなわち $0 < \eta < 2/\lambda$ のとき、かつそのときに限る。最適点にあたる $\eta = 1/\lambda$ では、反復点は 1 ステップで 0 に着地する。$1/\lambda < \eta < 2/\lambda$ では因子が負になるため、u_k はステップごとに符号を反転させながらゼロへ向かって縮小する。たとえば $\lambda = 10$ の場合、$\eta = 0.1$ では因子が 0、$\eta = 0.19$ では -0.9 となり、$\eta = 0.21$ では -1.1 となって発散する。

d 次元の二次関数もまったく同じ構造を保つ。$\nabla f(\mathbf{x}) = \mathbf{Ax}$ であるから、Equation 35.2 の更新は次のようになる。

$$\mathbf{x}_{k+1} = (\mathbf{I} - \eta\mathbf{A})\,\mathbf{x}_k.$$

各ステップは同じ線形作用素を適用するため、k ステップ後には $\mathbf{x}_k = (\mathbf{I} - \eta\mathbf{A})^k\mathbf{x}_0$ となる。$\mathbf{A}$ は対称行列であるから、固有ベクトルからなる正規直交基底を持つ（33.3 節）。$\mathbf{x}_k$ をこの固有基底で展開した座標を $u_i^{(k)}$ としよう。行列 $\mathbf{I} - \eta\mathbf{A}$ は同じ固有ベクトルを共有し、対応する固有値は $1 - \eta\lambda_i$ となるため、各座標は完全に分離して独立に時間発展する。

$$u_i^{(k+1)} = (1 - \eta\lambda_i)\,u_i^{(k)}, \qquad \text{hence} \qquad u_i^{(k)} = (1 - \eta\lambda_i)^k\,u_i^{(0)}. \tag{35.9}$$

したがって、d 次元の二次関数上の勾配降下法とは、固有値すなわち**モード**ごとに 1 つずつ、計 d 個の独立な 1 変数漸化式を並べたものにほかならない。因子 $1 - \eta\lambda_i$ から、直ちに 2 つの帰結が導かれる。

- **収束**には、すべてのモードで $|1-\eta\lambda_i| < 1$ が必要であり、これは $0 < \eta < 2/\lambda_{\max} = 2/L$ に帰着する。このしきい値が**安定性の上限**（stability ceiling）である。二次関数上では、降下補題が十分条件として保証した境界 $2/L$ は、厳密な必要条件でもある。η をこれより大きくすると、最も硬いモードの因子が -1 を下回り、他のすべてのモードがきれいに減衰していても、その座標だけは振幅を増しながら振動する。
- **収束速度**は最も遅いモードによって決まる。誤差ノルムは $\rho(\eta)^k$ に従って減衰する。ここで $\rho(\eta) = \max_i |1-\eta\lambda_i|$ は $\mathbf{I} - \eta\mathbf{A}$ の**スペクトル半径**（spectral radius）、すなわち固有値の絶対値の最大値である。

$\mathbf{A} = \mathrm{diag}(1, 10)$ では、安定性の上限は $\eta = 0.2$ である。以下のスイープを実行する前に、60 ステップを $(1, 1)$ から進めた後に誤差が最小になるステップサイズを予想してみよう。慎重な 0.02 か、降下補題が推す $0.1 = 1/L$ か、それとも上限にもっと近い値だろうか。

```
A = np.diag([1.0, 10.0])                    # eigenvalues 1 and 10, so L = 10
x0 = np.array([1.0, 1.0])

def run_gd(A, x0, eta, steps):
    xs = [x0]
    for _ in range(steps):
        xs.append(xs[-1] - eta * (A @ xs[-1]))
    return np.array(xs)

print(' eta    |1-eta*1|  |1-eta*10|  spectral radius   |x_60|')
for eta in [0.02, 0.10, 2 / 11, 0.19, 0.20, 0.21]:
    f1, f10 = abs(1 - eta * 1.0), abs(1 - eta * 10.0)
    print(f'{eta:5.3f}    {f1:6.3f}     {f10:6.3f}        {max(f1, f10):6.3f}     '
          f'{np.linalg.norm(run_gd(A, x0, eta, 60)[-1]):9.1e}')
```

```
 eta    |1-eta*1|  |1-eta*10|  spectral radius   |x_60|
0.020     0.980      0.800         0.980       3.0e-01
0.100     0.900      0.000         0.900       1.8e-03
0.182     0.818      0.818         0.818       8.3e-06
0.190     0.810      0.900         0.900       1.8e-03
0.200     0.800      1.000         1.000       1.0e+00
0.210     0.790      1.100         1.100       3.0e+02
```

慎重な $\eta = 0.02$ は安全だが動きが鈍い（$\rho = 0.98$ で、誤差は 0.3 が 60 ステップ後にも残る）。$\eta = 0.1$ では硬いモードは 1 ステップで消えるものの、遅いモードが全体の収束率を $\rho = 0.9$ に律速する。興味深いことに、$\eta = 0.19$ も正反対の理由からまったく同じ収束率 $\rho = 0.9$ を示す。今度は硬いモードが因子 -0.9 で振動するためである。両者の間にある $\eta = 2/11 \approx 0.182$ は両端のモードをつり合わせ、他のどの選択よりも誤差を 2 桁小さく抑える（$\rho = 0.818$、誤差 8.3×10^{-6}）。上限ちょうどの $\eta = 0.2$ では、硬いモードの因子が -1 となって永久に ± 1 の間で符号を反転し続けるため、$\|\mathbf{x}_{60}\| = 1.0$ となる。それをわずかに超える $\eta = 0.21$ に踏み出すと、60 回の反復で誤差は 3×10^2 まで膨れ上がる。

35.1.2.3 最適なステップサイズ

安定範囲の中で、最も速く収束する固定ステップサイズ η はどれだろうか。先ほどのスイープがそのまま答えを示している。両端のモードをつり合わせればよいのである。任意の固定された η に対し、写像 $\lambda \mapsto |1-\eta\lambda|$ は頂点を $\lambda = 1/\eta$ に持つ V 字型を描く。区間 $[\lambda_{\min}, \lambda_{\max}]$ 上で V 字型関数が最大値をとるのはいずれかの端点であるから、$\rho(\eta) = \max\left(|1-\eta\lambda_{\min}|,\ |1-\eta\lambda_{\max}|\right)$ となる。η が $1/\lambda_{\max}$ を超えると、さらに大きくするにつれて $\lambda_{\min}$ にある遅いモードは速まる一

方、$\lambda_{\max}$ にある硬いモードは悪化する。

命題（二次関数上の最適固定ステップ）。$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^{\top}\mathbf{A}\mathbf{x}$ に対し、収束因子 $\rho(\eta) = \max_i |1-\eta\lambda_i|$ を最小化するステップサイズと最小値は次式で与えられる。

$$\eta^{\star} = \frac{2}{\lambda_{\min} + \lambda_{\max}}, \qquad \rho(\eta^{\star}) = \frac{\kappa - 1}{\kappa + 1}, \qquad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}. \tag{35.10}$$

証明。 任意の η に対して $\rho(\eta) \geq \max(1-\eta\lambda_{\min},\, \eta\lambda_{\max}-1)$ が成り立ち、2 つの数の最大値はそれらのいかなる凸結合（convex combination）以上でもある。重みとして $\lambda_{\max}/(\lambda_{\min}+\lambda_{\max})$ と $\lambda_{\min}/(\lambda_{\min}+\lambda_{\max})$ を選ぶと、η を含む項がきれいに消去される。

$$\rho(\eta) \;\geq\; \frac{\lambda_{\max}(1-\eta\lambda_{\min}) + \lambda_{\min}(\eta\lambda_{\max}-1)}{\lambda_{\min}+\lambda_{\max}} = \frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}} = \frac{\kappa-1}{\kappa+1}.$$

等号が成立するのは 2 つの項が一致する $1-\eta\lambda_{\min} = \eta\lambda_{\max}-1$ の場合に限り、これを解くと $\eta = \eta^{\star}$ となる。$\eta^{\star}$ において両端点の値はともに $(\kappa-1)/(\kappa+1)$ に等しくなり、V 字型の形状により内部のすべての固有値はその高さ以下に保たれるため、この下界が達成される。$\blacksquare$

Figure 35.3 は、この証明を $\mathrm{diag}(1, 10)$ 上で図示したものである。最適な V 字は、両端点が同じ高さに揃うものである。この場合 $\eta^{\star} = 2/11$、$\rho = 9/11 = 0.818$ となるため、6 桁の精度を得るには $\log 10^{6}/\log(11/9) \approx 69$ ステップを要する。本章の通し例（$\kappa = 1000$）では $\rho = 999/1001$ となり、同じ 6 桁の精度を得るのに 6,908 ステップが必要になる。

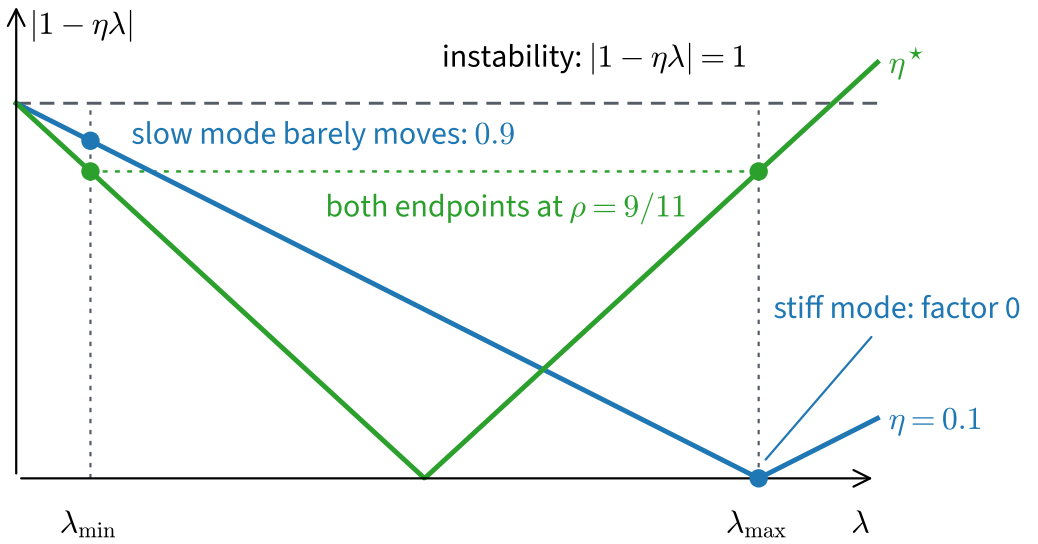


Figure 35.3: モードごとの縮小率 $|1-\eta\lambda|$ は $\lambda = 1/\eta$ を頂点とする V 字を描き、収束因子 $\rho(\eta)$ は両端の固有値 $\lambda_{\min} = 1$ と $\lambda_{\max} = 10$ における値のうち大きい方に等しい。$\eta = 0.1$ を選ぶと硬いモードは即座に消えるが遅いモードの縮小率は 0.9 にとどまる。最適ステップ $\eta^{\star} = 2/(\lambda_{\min}+\lambda_{\max})$ は両端点を $(\kappa-1)/(\kappa+1) = 9/11$ で等しくする。一方の端点を下げようとすれば、もう一方が上がってしまう。

$\kappa = 1$（完全に丸いボウル）のときは $\rho = 0$ となり、1 ステップで最小点に到達する。大きな κ に対しては $(\kappa-1)/(\kappa+1) \approx 1 - 2/\kappa$ であるから $\log(1/\rho) \approx 2/\kappa$ となり、誤差を ϵ 倍に縮めるには約 $\frac{\kappa}{2}\log\frac{1}{\epsilon}$ 回の反復が必要となる。すなわち、**勾配降下法の反復回数は条件数に対して線形に増大する**。ボトルネックは一文で言い表せる。安定性の要請からステップサイズが最も硬いモードに従う $\eta \lesssim 2/\lambda_{\max}$ に縛られるため、最も平坦なモードは反復ごとに $\eta\lambda_{\min} \approx 2/\kappa$ しか進めないからである。

$\eta^\star$ において、因子 ρ は漸近的な収束率よりもさらに強い意味を持つ。$\mathrm{diag}(1, 10)$ では、モードごとの因子は $+9/11$ と $-9/11$（先ほどのスイープの $\eta = 2/11$ の行）であり、遅いモードが単調に縮小する一方、硬いモードは符号を交互に反転させる。両座標とも同じ倍率で縮むため、$\|\mathbf{x}_k\|$ は平均的にではなく、毎ステップ厳密に $(\kappa-1)/(\kappa+1)$ 倍ずつ縮小する。

35.1.2.4 安定性の縁

安定性の上限は、ニューラルネットワークに対して単純な処方箋を示唆する。ヘッセ行列の最大固有値 $\lambda_{\max}(\nabla^2 f)$ である**鋭さ**（sharpness）を計算し、$\eta < 2/\lambda_{\max}$ と設定すればよいという考えである。しかし実際のニューラルネットワークの訓練はその通りには進まない。軌跡に沿ってヘッセ行列自体が変化していくからである。フルバッチ勾配降下法のもとでは、鋭さは通常おおよそ $2/\eta$ に達するまで上昇し続け、その後はそのしきい値のすぐ周りで推移する。その間、個々のステップでは損失が頻繁に跳ね上がるにもかかわらず、長期的には減少していく (Cohen et al. 2021)。9.1.4 節では小さな tanh ネットワーク上でこの実験を行い、詳しく調べている。ここでは問いを絞り、この振る舞いのどこまでを二次モデルで説明できるか考えよう。

答えは、おおよそ半分である。ヘッセ行列の固有基底において最大固有値モードの乗数は $1 - \eta\lambda_{\max}$ であり、-1 を下回るのはまさに $\lambda_{\max} > 2/\eta$ のときである。するとその座標は振幅を増しながら符号を反転させるため、全体として減少傾向にある損失が個々のステップでスパイクを起こす理由が説明できる。しかし固定された二次関数であれば、その増大は無限大へ発散してしまうはずである。鋭さを再び引き戻す仕組みは二次近似の外側にある。二次モデルが無視している三次のテイラー項が、その振動をより鋭さの低い平坦な領域への緩やかなドリフトへと変えるのである (Damian et al. 2023)。演習問題 8 では、モメンタムを加えるとこの安定性のしきい値が $(2 + 2\beta)/\eta$ に移動することを示す。

35.1.2.5 二次関数から凸関数へ

二次関数を超えて引き継がれるのは、κ に対する線形依存性である。微分可能な関数は、どの接平面に対してもその上または上方に位置するとき凸関数であり、さらに二次のマージンをもって接平面を上回るとき μ 強凸（strong convexity）であるという。すなわち $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2$ がすべての $\mathbf{x}, \mathbf{y}$ に対して成り立つことであり、35.3 節で両概念を詳しく展開する。f が L 滑らかかつ μ 強凸であれば、どの点においても曲率 μ と L の 2 つの二次関数の間に挟まれる。条件数は $\kappa = L/\mu$ が $\lambda_{\max}/\lambda_{\min}$ に取って代わり、$\eta = 1/L$ の勾配降下法は次式を満たす。

$$f(\mathbf{x}_k) - f^\star \leq \left(1 - \frac{1}{\kappa}\right)^k \left(f(\mathbf{x}_0) - f^\star\right), \tag{35.11}$$

これもやはり $O(\kappa \log\frac{1}{\epsilon})$ 回の反復を要する。f が強凸ではなく単に滑らかな凸関数であっても、勾配降下法は大域的最適解へ収束するが、その収束率はより遅い劣線形レート $O(1/k)$ になる。凸性も外してしまえば、残るのは停留性の上界 Equation 35.6 のみである。これらの証明は凸性に基づくものであり、35.3.3.3 節で展開し、二次関数上で達成されるよりシャープな収束率とも比較する。

35.1.3 モメンタムと加速

勾配降下法が $O(\kappa)$ 回の反復を要するのは、単一のステップサイズで最も硬いモードと最も平坦なモードの両方に同時に対応しなければならないからである。モメンタムは 1 ステップあたり 1 回の勾配評価というコストを保ったまま更新則を変更し、両方のモードが同じ速さで縮小できるようにする。

35.1.3.1 ヘビーボールモメンタム

谷の幾何構造を見れば、何を改めるべきかは直ちにわかる。急峻な壁に沿っては連続する勾配が逆向きを指すため（Figure 35.2 のジグザグ）、直近の勾配を平均すれば振動が打ち消し合う。平坦な谷底に沿っては連続する勾配は小さいものの一貫して同じ向きに揃っているため、それらを蓄積すれば速度が増す。1 つの単純な仕組みでこの両方が実現できる。反復点に記憶を伴う速度を持たせればよいのである。**ヘビーボールモメンタム** (Polyak 1964) は、通常の勾配ステップを次式に置き換える。

$$\mathbf{v}_{k+1} = \beta\,\mathbf{v}_k - \eta\,\nabla f(\mathbf{x}_k), \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1}, \tag{35.12}$$

ここで、$\beta \in [0,1)$ はモメンタム係数である。速度 $\mathbf{v}_k$ は過去の勾配の指数減衰和となっている。$\mathbf{v}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ であるから、$\mathbf{v}$ を消去すると等価な 2 ステップ形式が得られる。

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\,\nabla f(\mathbf{x}_k) + \beta\,(\mathbf{x}_k - \mathbf{x}_{k-1}).$$

各更新は勾配ステップを進めるとともに、直前の変位の β 倍を加える。$\beta = 0$ とおけば通常の勾配降下法に戻る。

二次関数上では、各モードは再び分離する。勾配 λu_k を持つ単一のモードでは、2 ステップ更新式は $u_{k+1} = (1 + \beta - \eta\lambda)\,u_k - \beta\,u_{k-1}$ となり、各反復点は直前の 2 点に依存する。この形の 2 階漸化式は、減衰調和振動子（摩擦 $1-\beta$ のあるボウルの中を転がる重い球）の離散版にあたり、β が減衰の強さを制御する。

Figure 35.4 は、$\mathrm{diag}(1, 10)$ 上で勾配降下法の最適ステップ $\eta = 2/11$ に統一したときに生じる 3 つのレジームを示している。モメンタムが小さすぎると、経路は通常の勾配降下法と同様にジグザグになる。ちょうど適切な量であれば、1 回のオーバーシュートの後に滑らかに落ち着く。モメンタムが大きすぎると、球は過剰な慣性を持って最小点の周りを渦巻き状に回る。次にこれら 3 つのレジームすべてに厳密な収束率（convergence rate）を与え、可能な限り最速の収束率を達成するにはより大きなステップサイズも必要になることを見いだす。

35.1.3.2 $\sqrt{\kappa}$ 則

ここで、適切な組 (η, β) を選べば、すべてのモードが同一の縮小率 $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ で縮小することを示そう。$\kappa = 10$ の場合、この最適な組は $\beta^\star \approx 0.27$、$\eta_{\mathrm{HB}} \approx 0.23$ であり、縮小率は勾配降下法の 0.82 から 0.52 へと下がり、6 桁の精度に 27 ステップで到達する（勾配降下法では 69 ステップ）。驚くべきことに、この結果全体は 1 つのモードに関するただ 1 つの代数的事実から導かれる。

命題（ヘビーボール法はすべてのモードを $\sqrt{\beta}$ で縮小させる）。 *$0 < \beta < 1$ とする。$f(u) = \tfrac{\lambda}{2}u^2$ 上で、ヘビーボール法の反復点は $u_{k+1} = (1+\beta-\eta\lambda)u_k - \beta u_{k-1}$ に従う。$(1-\sqrt{\beta})^2 \le \eta\lambda \le (1+\sqrt{\beta})^2$ である限り、特性多項式の 2 つの根はともに絶対値 $\sqrt{\beta}$ を持ち、$|u_k| \le C\,k\,\beta^{k/2}$ がすべての $k \ge 1$ とある定数 C（u_0、u_1、および $\eta\lambda$ に依存）に対して成り立つ。線形因子 k が必要になるのは、根が重根となる両端点のみである。*

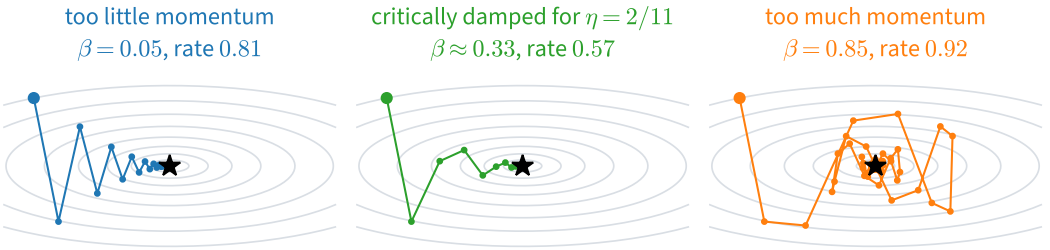


Figure 35.4: $\kappa = 10$ の谷 $\frac{1}{2}(x^2 + 10y^2)$ におけるヘビーボール法（3 つの実行すべてで勾配降下法のステップ $\eta = 2/11$ を使用）。モメンタムが小さすぎると勾配降下法のようにジグザグになり（硬いモードの係数 $1 + \beta - \eta\lambda_{\max}$ が負となり、負の実根を持つ）、このステップサイズに対する臨界減衰の β が最速となり、モメンタムが大きすぎると最小点の周りを渦巻き状に回る。最適な $\sqrt{\kappa}$ の縮小率である 0.52 に達するには、より大きなステップ η_{HB}（Equation 35.14）が必要であり、そのステップサイズでは $\beta = 0.05$ は発散する。

証明。試行解 $u_k = z^k$ を漸化式に代入し、z^{k-1} で割ると、次の特性方程式が得られる。

$$z^2 - (1 + \beta - \eta\lambda)\, z + \beta = 0. \tag{35.13}$$

2 つの根の積は β である。判別式 $(1+\beta-\eta\lambda)^2 - 4\beta$ が負のとき、すなわち $|1+\beta-\eta\lambda| < 2\sqrt{\beta}$（整理すると $(1-\sqrt{\beta})^2 < \eta\lambda < (1+\sqrt{\beta})^2$）のとき、2 つの根は複素共役対をなす。複素共役 $z, \bar{z}$ は $|z|^2 = z\bar{z} = \beta$ を満たすため、どちらの根も絶対値 $\sqrt{\beta}$ を λ の値にかかわらず持ち、一般解 $u_k = az^k + b\bar{z}^k$ は $|u_k| \le (|a| + |b|)\beta^{k/2}$ を満たす。区間の両端点では判別式がゼロとなり、実重根 $(1 + \beta - \eta\lambda)/2 = \pm\sqrt{\beta}$ が得られる。重根 z に対する一般解は $u_k = (a + bk)z^k$ となり、ここから余分な因子 k が生じる。■

系（最適調整）。両端のモードを複素根区間のちょうど両端に配置する。すなわち $\eta\lambda_{\min} = (1 - \sqrt{\beta})^2$ および $\eta\lambda_{\max} = (1+\sqrt{\beta})^2$ とおく。するとすべての中間固有値はこの区間の内部に入るため、あらゆるモードの根が絶対値 $\sqrt{\beta}$ を持つ。平方根をとって加減すると、$\sqrt{\eta\lambda_{\max}} - \sqrt{\eta\lambda_{\min}} = 2\sqrt{\beta}$ および $\sqrt{\eta\lambda_{\max}} + \sqrt{\eta\lambda_{\min}} = 2$ が得られる。第 2 式を η について解き、第 1 式を第 2 式で割れば $\sqrt{\beta} = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ となる。したがって、次のパラメータ設定をとる。

$$\eta_{\mathrm{HB}} = \left(\frac{2}{\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}}} \right)^2, \qquad \beta^\star = \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^2, \tag{35.14}$$

このとき、ヘビーボール法はすべてのモードを次の縮小率で縮小させる。

$$\rho_{\mathrm{HB}} = \sqrt{\beta^\star} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \;\approx\; 1 - \frac{2}{\sqrt{\kappa}}. \tag{35.15}$$

Equation 35.10 と比較してほしい。モメンタムは κ を $\sqrt{\kappa}$ に置き換え、反復計算量を $O(\kappa \log \frac{1}{\epsilon})$ から $O(\sqrt{\kappa} \log \frac{1}{\epsilon})$ へと劇的に削減する (Polyak 1964)。有限の k においては、両端のモードにおける重根が控えめな対数オーバーヘッドを加える。誤差は $k\rho^k$ のようにスケールするため、

$\log(1/\rho) \approx 2/\sqrt{\kappa}$ を用いると、精度 ϵ に達するステップ数は $k \approx \frac{\sqrt{\kappa}}{2}\big(\log\frac{1}{\epsilon} + \log k\big)$ の解となる。これは漸近則にオーダー $\sqrt{\kappa}\log\kappa$ の補正項を加えたものである。

Figure 35.5 は、この証明を $\kappa = 10$ について視覚化したものである。左図では、絶対値最大の根の絶対値が複素根区間全体にわたって $\sqrt{\beta}$ で水平に並んでいる。右図の複素平面では、すべてのモードの根の対が半径 $\sqrt{\beta^\star} = 0.52$ の円周上にきっちり乗っているのに対し、勾配降下法の実乗数は ± 0.82 まで広がっている。根が実軸から離れることを許すことで、モメンタムは平坦なモードに $1 - 2/\kappa$ の鈍足な進行を強いる代わりに、硬いモードと平坦なモードに単一の小さな絶対値を共有させるのである。

この命題は、Figure 35.4 の 3 つの軌跡（$\eta = 2/11$ により 2 つのモードが $\eta\lambda = 0.18$ と 1.82 に置かれている）の厳密な収束率も説明する。$\beta = 0.85$ では両モードとも区間の真に内部に入り、縮小率は $\sqrt{0.85} = 0.92$ となって複素根が渦巻き状のオーバーシュートを生む。区間が遅いモードを依然として捉える最小の β である $\beta = (1 - \sqrt{2/11})^2 \approx 0.33$ は、このステップサイズにおける最速の選択であり、縮小率 0.57 を達成する。$\beta = 0.05$ では区間 $[0.60, 1.50]$ が両モードとも外してしまうため、遅いモードは実根 0.81 を持ち、硬いモードはステップごとに符号を反転させる負の実根を持つ。このような符号交代は、連続時間（continuous-time）の物理的な振動子では決して起こりえない現象である。

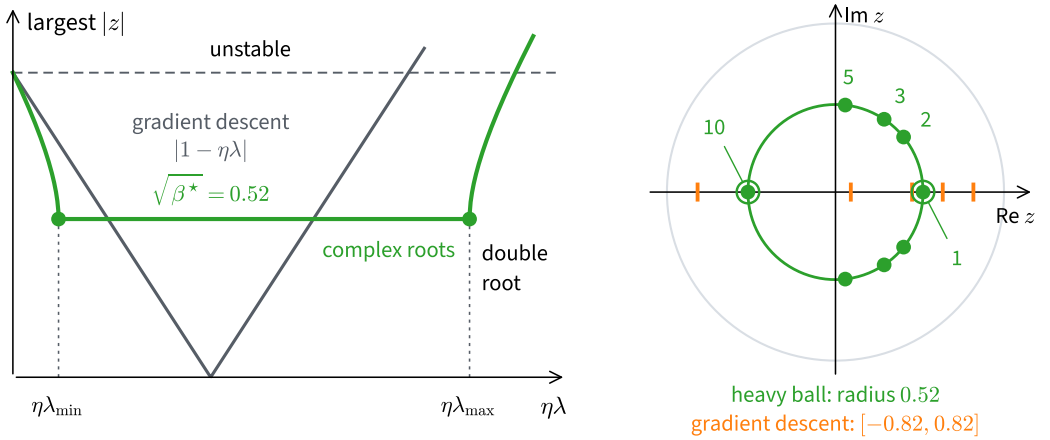


Figure 35.5: Equation 35.14 の調整を用いた $\kappa = 10$ の例におけるヘビーボール法。(a) 固定された β に対し、大きい方の根の絶対値は根が複素数となる領域全体で $\sqrt{\beta}$ の一定値をとり、最適調整は両端のモードをその区間の両端に配置する。(b) 複素平面において、すべてのモードの根の対（固有値 λ でラベル付け）は半径 $\sqrt{\beta^\star} = 0.52$ の円周上に乗る一方、勾配降下法の因子は実軸上の $[-0.82, 0.82]$ に広がる。両端のモードは重根となり、誤差に因子 k を生じさせる。

ただし、この $\sqrt{\kappa}$ 定理が厳密に成り立つのは二次関数に限られる点に注意が必要である。Lessard et al. (2016) は、1 次元の滑らかな強凸関数であって、Equation 35.14 の調整を用いたヘビーボール法が特定の初期値（初期化）から収束せず、反復点が安定なリミットサイクルに閉じ込められる例を構成した（演習問題 6）。したがって、深層学習でヘビーボールモメンタムが広く使われている背景 (Sutskever et al. 2013) は、大域的な収束定理ではなく、局所的な二次近似の描像と経験的な成功にある。

35.1.3.3 ネステロフの先読み

1 つの小さな修正によって大域的な保証が復活する。勾配を現在位置 $\mathbf{x}_k$ ではなく、モメンタムがまさに反復点を運ぼうとしている先読み点 $\mathbf{x}_k + \beta\,\mathbf{v}_k$ で評価するのである。**ネステロフの加速勾**

配法（Nesterov's accelerated gradient）(Nesterov 1983) は次式に従って更新する。

$$\mathbf{v}_{k+1} = \beta \mathbf{v}_k - \eta \nabla f(\mathbf{x}_k + \beta \mathbf{v}_k), \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1}. \tag{35.16}$$

ヘビーボール法との違いは、この評価点のずれだけである。直観的には、モメンタムが反復点を最小点の向こう側へ行き過ぎさせようとしているとき、先読み点における勾配はすでに上り坂の引き戻す向きを指しており、オーバーシュートが起きる前にブレーキをかける。

命題（二次関数上のネステロフ法）。 $\kappa = L/\mu$ **とする。** $\eta = 1/L$ **および** $\beta = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ **とおくとき、固有値が** $[\mu, L]$ **に含まれる二次関数のすべてのモードについて、特性根の絶対値は高々** $1 - 1/\sqrt{\kappa}$ **となり、遅いモード** $\lambda = \mu$ **で重根を持つ。したがって、誤差は** $C\,k\,(1-1/\sqrt{\kappa})^k$ **で上から抑えられる。**

証明。 固有値 λ の単一モードにおいて、$a = 1 - \eta\lambda \in [0, 1 - 1/\kappa]$ とおく。速度の更新式は $v_{k+1} = a\beta v_k - \eta\lambda u_k$ となり、$u_{k+1} = a u_k + a\beta v_k$ を得る。ここに $v_k = u_k - u_{k-1}$ を代入すると漸化式 $u_{k+1} = (1+\beta)a\,u_k - \beta a\,u_{k-1}$ が得られ、その特性多項式 $z^2 - (1+\beta)a\,z + \beta a$ の判別式は $a\,[(1+\beta)^2 a - 4\beta]$ となる。今回の β の選び方では $(1+\beta)^2 = 4\kappa/(\sqrt{\kappa}+1)^2$ が成り立つ。遅いモード $a = (\kappa-1)/\kappa$ においては角括弧の中身が $4(\kappa-1)/(\sqrt{\kappa}+1)^2 - 4\beta = 4\beta - 4\beta = 0$ となり、$(1+\beta)a/2 = (\sqrt{\kappa}-1)/\sqrt{\kappa} = 1 - 1/\sqrt{\kappa}$ に重根が生じる。それより硬いすべてのモードはより小さな a を持つため、角括弧の中身は真に負となり、複素根（または 0 の根、ただし $a = 0$ の場合）を生じさせ、その絶対値は $\sqrt{\beta a} < \sqrt{\beta(1-1/\kappa)} = 1 - 1/\sqrt{\kappa}$ となる。$\blacksquare$

二次関数上でのネステロフ法の縮小率は、0.684 ($\kappa = 10$)、0.900 ($\kappa = 100$)、0.968 ($\kappa = 1000$) となり、ヘビーボール法の 0.519、0.818、0.939 よりわずかに遅いものの、同じ $\sqrt{\kappa}$ のスケーリングを保つ。決定的に重要なのは、ヘビーボール法とは異なり、ネステロフ法が二次関数を超えてもその収束保証を維持する点である（ここでは証明なしに結果のみ述べる (Nesterov 2018)）。任意の μ 強凸かつ L 滑らかな関数上で、同じ定数 β を用いたネステロフ法の更新は $f(\mathbf{x}_k) - f^\star \le C\,(1 - 1/\sqrt{\kappa})^k$ を満たす（単位の違いに注意しよう。これは距離の二乗としてスケールする関数値ギャップ $f - f^\star$ を抑える式であるため、二次関数の命題における距離の上界と比べて指数が半分に対応している）。強凸性を持たない滑らかな凸関数上でも、スケジュール $\beta_k \to 1$ を用いれば $f(\mathbf{x}_k) - f^\star = O(1/k^2)$ を達成し、勾配降下法の $O(1/k)$ を改善する。さらに、一般にはこの速さを上回ることはできない。次元が反復回数に比べて大きい限り、過去の勾配のみから反復点を構成するいかなる手法も、最悪ケースにおいて $O(1/k^2)$ や $\sqrt{\kappa}$ 依存性を凌駕できないからである (Nemirovski and Yudin 1983)。演習問題 7 では、この限界を証明する最悪ケースの関数を実際に構成する。

ここで、本節の冒頭で挙げた反復回数を実際に検証してみよう。$\mathrm{diag}(1, \kappa)$ 上で $(1, 1)$ から出発するとき、6 桁の精度を得るのに要するステップ数は、勾配降下法ではおおよそ $\frac{\kappa}{2}\log 10^6 \approx 6.9\,\kappa$、ヘビーボール法では $6.9\sqrt{\kappa}$ を少し上回る程度になるはずである。

```python
def iterations(lam, eta, beta=0.0, lookahead=False, tol=1e-6):
    """Steps until |x_k| <= tol |x_0| on f(x) = sum_i lam_i x_i^2 / 2."""
    x, v = np.array([1.0, 1.0]), np.zeros(2)
    n0 = np.linalg.norm(x)
    for k in range(1, 10**6):
        g = lam * (x + beta * v) if lookahead else lam * x
        v = beta * v - eta * g
        x = x + v
        if np.linalg.norm(x) <= tol * n0:
            return k
```

```
    print('kappa      GD   heavy ball   Nesterov     GD/kappa   HB/sqrt(kappa)')
    for kappa in [10.0, 100.0, 1000.0]:
        lam, sk = np.array([1.0, kappa]), np.sqrt(kappa)
        gd = iterations(lam, 2 / (1 + kappa))
        hb = iterations(lam, (2 / (1 + sk))**2, ((sk - 1) / (sk + 1))**2)
        na = iterations(lam, 1 / kappa, (sk - 1) / (sk + 1), lookahead=True)
        print(f'{kappa:5.0f} {gd:7d} {hb:9d} {na:10d}      '
              f'{gd / kappa:7.1f}     {hb / sk:7.1f}')
```

```
kappa      GD   heavy ball   Nesterov     GD/kappa   HB/sqrt(kappa)
   10      69         27         43          6.9          8.5
  100     691         93        155          6.9          9.3
 1000    6908        315        508          6.9         10.0
```

表はすべての予測を裏付けている。勾配降下法のステップ数を κ で割った比は 6.9 に張り付いたままであり、$6{,}908$ 回の反復（$\kappa = 1000$ のとき）に達するのに対し、ヘビーボール法はわずか 315 回で完了する。なお、ヘビーボール法の比 $k/\sqrt{\kappa}$ が緩やかに上昇している（8.5、9.3、10.0）ことに注目しよう。これは両端のモードがそれぞれ重根を持つためであり、その 2×2 反復行列はただ 1 本の固有ベクトルしか持たない重複固有値のジョルダンブロックをなし（33.3.1.5 節）、その k 乗が因子 k をもたらすからである。精密化した予測式 $k \approx \frac{\sqrt{\kappa}}{2}(\log 10^6 + \log k)$ は実測とよく一致し（8.6、9.2、9.8）、$\kappa = 10^4$ ではヘビーボール法は $1{,}056$ ステップ、比にして 10.6 を要する（演習問題 5）。ネステロフ法の 508 回（$\kappa = 1000$ のとき）は両者のちょうど中間に収まっており、縮小率 0.968 対ヘビーボール法の 0.939 から予測されるとおりである。3 手法はいずれもステップごとに勾配を 1 回しか評価しないため、モメンタムによる高速化には追加の勾配計算コストが一切かからない。しかし同時に、$\sqrt{\kappa}$ は勾配だけで達成しうる限界でもある。一次手法の下界により、高次元において過去の勾配のみから構成される手法はこれを超えられない。κ への依存性を完全に取り除くには、曲率を実際にモデル化するステップが必要となる（35.2 節）。

35.1.4 まとめ

滑らかさは、勾配ステップをどこまで信頼できるかの上限を定める。L 滑らかな目的関数上では、$\eta = 2/L$ まで降下が保証される。二次関数上では、最良の固定ステップは反復ごとに誤差を $(\kappa - 1)/(\kappa + 1)$ 倍に縮小させるが、安定性が最も硬いモードに支配される一方で進行速度が最も平坦なモードに支配されるため、ステップ数は条件数 κ に対して線形に増大せざるをえない。モメンタムはすべてのモードを共通の縮小率 $\sqrt{\beta}$ に揃え、最適調整された因子 $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ によって、追加の勾配コストなしに反復回数を $O(\kappa)$ から $O(\sqrt{\kappa})$ へと削減する。ニューラルネットワーク上では、鋭さが選択したステップサイズの安定性の上限 $2/\eta$ まで上昇するが、この現象は二次モデルだけでは一部しか説明できない。残る問いは 3 つある。曲率を直接モデル化するとオプティマイザはどこまで達成できるのか（35.2 節）、停留点が大域的最小点であることがいつ保証されるのか（35.3 節）、そして勾配にノイズが加わると何が変わるのか（35.4 節）である。

35.1.5 演習問題

1. **前処理付き降下方向。** 35.1.1.1 節の命題では、方向の長さをユークリッドノルムで測っている。異なるノルムを選ぶと、異なる最急降下方向が得られる。
    1. $\mathbf{d} = -\mathbf{P}^{-1}\nabla f(\mathbf{x})$ が任意の正定値行列 $\mathbf{P}$ に対して降下方向となることを示せ。
    2. 長さを $\|\mathbf{d}\|_{\mathbf{P}} = \sqrt{\mathbf{d}^{\top}\mathbf{P}\,\mathbf{d}}$ で測るものとする。$\nabla f(\mathbf{x})^{\top}\mathbf{d}$ の $\|\mathbf{d}\|_{\mathbf{P}} = 1$ 上における最小点が $-\mathbf{P}^{-1}\nabla f(\mathbf{x})$ に比例することを示し、さらにどのような $\mathbf{P}$ を選べばニュートン方向 $-\big(\nabla^2 f(\mathbf{x})\big)^{-1}\nabla f(\mathbf{x})$（35.2 節の主題）が復元されるか特定せよ。変数変換

$\mathbf{u} = \mathbf{P}^{1/2}\mathbf{d}$ を用いればユークリッドノルムの場合に帰着する。

2. [code] **滑らかさの定数**。滑らかさの定数 L（Equation 35.3）は f の曲率を上から抑え、ロジスティック損失に対してはデータから直接計算できる。

   1. f を 2 回連続微分可能かつ L 滑らかとする。$\|\nabla^2 f(\mathbf{x})\,\mathbf{v}\| \le L\,\|\mathbf{v}\|$ がすべての $\mathbf{x}$ と $\mathbf{v}$ に対して成り立つことを示し、ヘッセ行列の固有値がすべて $[-L, L]$ に収まることを結論せよ。これにより、Equation 35.3 の直後で述べた同値性の逆方向が示される。

   2. 次の関数を考える。

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-y_i\,\mathbf{x}_i^{\top}\mathbf{w})\right) + \frac{\nu}{2}\|\mathbf{w}\|^2$$

   ここでラベルは $y_i = \pm 1$、$\nu = 0.1$ である。これは `loss` 関数として 35.2.1 節のデータセルで用いるものであり、$\mathbf{x}_i^{\top}$ は第 i 行として行列 $\mathbf{X} \in \mathbb{R}^{n \times d}$（$n = 200$、$d = 5$ で、各成分は独立な標準ガウス分布に従う）から取り出したベクトルである。$L = \lambda_{\max}(\mathbf{X}^{\top}\mathbf{X})/(4n) + \nu$ が正当な滑らかさの定数であり、ヘッセ行列が $\mathbf{w} = \mathbf{0}$ においてこの上界を達成することを示せ。

   3. この L を、標準ガウス分布に従う成分を持つ 200×5 行列 $\mathbf{X}$ と、ラベル $y_i \in \{\pm 1\}$（確率 $P(y_i = 1) = 1/(1 + \exp(-\mathbf{x}_i^{\top}\mathbf{w}_{\text{true}}))$、$\mathbf{w}_{\text{true}} = (1, -2, 1.5, 0, 2)$ に従って生成）に対して計算せよ。どの乱数サンプルでも値は 0.4 付近になる。標準ガウス成分に対して $\lambda_{\max}(\mathbf{X}^{\top}\mathbf{X})/n \approx (1 + \sqrt{d/n})^2$ が成り立つことを用いて、その理由を説明せよ。さらに、$\eta = 0.5$（基準ステップ）と $\eta = 1/L$ のそれぞれについて、$\mathbf{w} = \mathbf{0}$ から出発して $\|\nabla f(\mathbf{w})\| \le 10^{-10}$ に達するまでに要するステップ数を数えよ。どちらが速いか。また、$\eta = 0.5$ は $2/L$ 未満に収まっているか。

3. [code] **四次関数におけるクリッピング**。$f(x) = \tfrac{1}{4}x^4$ とし、クリップ付き更新 $x \leftarrow x - \eta\,\min(1, c/|g|)\,g$（ただし $g = x^3$）を考える。

   1. $3x^2 \le 4 + |x|^3$ がすべての x に対して成り立ち、$|x| = 2$ で等号が成立することを示し、したがって f が Equation 35.8 を $L_0 = 4$ および $L_1 = 1$ で満たすことを示せ。また、いかなる L_0 を選んでも $L_1 = 0$ では成り立たないことも示せ。
   2. クリップ付き更新を $x_0 = 3$ から $\eta = 0.3$、$c = 1$ で実行せよ（クリップなしの反復が 6 ステップで 6.5×10^{103} に達した設定である）。何ステップがクリップされ、その各ステップは x をどれだけ動かすか。
   3. クリップ付き更新は、$|x|$ を任意の初期点 $x_0 \neq 0$ から毎ステップ減少させることを、$\eta\,c^{2/3} < 2$ の仮定のもとで示せ。クリップされる場合 $|x| > c^{1/3}$ とクリップされない場合を分けて扱え。クリッピングなしの固定ステップでは、なぜこの性質を決して持ちえないのか。

4. **他のノルムにおける最急降下**。●問題 1b では、内積から誘導されるノルム $\|\cdot\|_{\mathbf{P}}$ で長さを測った。内積に由来しないノルムを用いると、まったく異なる形の最急降下方向が生じる。

   1. $\nabla f(\mathbf{x})^{\top}\mathbf{d}$ の $\|\mathbf{d}\|_{\infty} = 1$ 上および $\|\mathbf{d}\|_1 = 1$ 上における最小点と最小傾きをそれぞれ求め、最小点が一意となる条件を述べよ。
   2. パラメータを行列とし、その勾配 $\mathbf{G}$ の簡約特異値分解を $\mathbf{G} = \mathbf{U}\,\mathrm{diag}(\sigma_i)\,\mathbf{V}^{\top}$ とする。方向 $\mathbf{D}$ の方向傾きを $\mathrm{tr}(\mathbf{G}^{\top}\mathbf{D})$ とし、長さをスペクトルノルム $\|\mathbf{D}\|_2$ で測る。$\|\mathbf{D}\|_2 \le 1$ 上での最小傾きが $-\sum_i \sigma_i$ であり、それが $\mathbf{D} = -\mathbf{U}\mathbf{V}^{\top}$（Muon がモメンタム行列に適用する極因子、9.6 節）で達成されることを示せ。最初のステップと

して $\operatorname{tr}(\mathbf{G}^\top \mathbf{D}) = \sum_i \sigma_i \, \mathbf{u}_i^\top \mathbf{D} \, \mathbf{v}_i$ と書き直すと見通しがよい。これは Bernstein and Newhouse (2024) による最急降下の視点である。

5. [code] **ヘビーボール法：重根と因子 k。**●ヘビーボール法 Equation 35.12 を $f(u) = \tfrac{\lambda}{2} u^2$ に適用せよ。

   1. 1 回の反復を状態 (u_k, u_{k-1}) 上の線形変換として表し、その 2×2 行列の特性多項式が Equation 35.13 となることを示せ。
   2. Equation 35.14 の調整において、両端のモードは重複固有値 $\sqrt{\beta^\star}$ および $-\sqrt{\beta^\star}$ を持つ。このとき対応する行列が固有ベクトルを 1 本しか持たないこと、および k 乗したジョルダンブロック $\left(\begin{smallmatrix} \rho & 1 \\ 0 & \rho \end{smallmatrix}\right)$ の対角線の上の成分が $k\rho^{k-1}$ となることを示せ。
   3. $\|\mathbf{e}_k\| \approx k\rho^k$ および $\log(1/\rho) \approx 2/\sqrt{\kappa}$ のもとで、$k = \frac{\sqrt{\kappa}}{2}\big(\log \frac{1}{\epsilon} + \log k\big)$ を $\epsilon = 10^{-6}$ および $\kappa \in \{10, 100, 1000\}$ に対して数値的に解き、$k/\sqrt{\kappa}$ をモメンタムのコードセルの列 8.5、9.3、10.0 と比較せよ。
   4. モメンタムのコードセルのヘビーボール法を $\kappa = 10^4$ で実行し、そのステップ数を予測値と比較せよ。比 $k/\sqrt{\kappa}$ が、κ の増大に伴って決して定数に収束しない理由を説明せよ。

6. [code] **二次関数を超えたヘビーボール法。**●Lessard et al. (2016) は、勾配によって次の 1 次元関数を定義している。

$$\nabla f(x) = \begin{cases} 25x, & x < 1, \\ x + 24, & 1 \le x < 2, \\ 25x - 24, & x \ge 2. \end{cases}$$

   1. f が μ 強凸（$\mu = 1$）かつ L 滑らか（$L = 25$）であることを示せ。
   2. Equation 35.14 の調整を用いたヘビーボール法を 300 ステップ、$x_0 = 3.3$ かつ初速度ゼロから実行し、さらにネステロフ法 Equation 35.16 を $\eta = 1/L$ および $\beta = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ として同じ初期条件から実行せよ（ただし $\kappa = L/\mu$）。両手法について x_k を k に対してプロットせよ。
   3. ヘビーボール法の反復点は、区間 $x < 1$ を 2 回、区間 $x \ge 2$ を 1 回訪れる周期 3 のサイクルに漸近する。これらの区間にわたるヘビーボール法の更新を 3 つのサイクル点に関する連立一次方程式として表して解き、その解をシミュレーションと比較せよ。
   4. $[-6, 6]$ 上の初期点グリッドからヘビーボール法の実行を繰り返し、このサイクルに行き着く初期点の割合を報告せよ。f の各区間は曲率が $[\mu, L]$ に収まる二次関数であるにもかかわらず、Equation 35.15 の背後にあるモードごとの解析がなぜ f には適用できないのか。

7. [code] **一次手法に対する下界。**●この問題では、$O(1/k^2)$ の下界（Nemirovski and Yudin (1983) による結果のうち、35.1.3 節で滑らかな凸関数の場合として引用した部分）を、Nesterov (2018) の構成に従って導出する。ここで一次手法とは、反復点が $\mathbf{x}_k \in \mathbf{x}_0 + \operatorname{span}\{\nabla f(\mathbf{x}_0), \dots, \nabla f(\mathbf{x}_{k-1})\}$ を満たす任意のアルゴリズムを指すものとし、$\mathbf{x}_0 = \mathbf{0}$ ととる。$p \le d$ に対し、$\mathbf{A}_p \in \mathbb{R}^{d \times d}$ を先頭の $p \times p$ ブロックを除いてゼロとし、そのブロックは対角成分が 2、上下の副対角成分がともに -1 の三重対角行列であるとして、次式を定義する。

$$f_p(\mathbf{x}) = \frac{L}{4}\left(\tfrac{1}{2}\,\mathbf{x}^\top \mathbf{A}_p\, \mathbf{x} - x_1\right).$$

   1. f_p が凸関数かつ L 滑らかであることを示せ。

2. $\mathbf{x}$ の台（非ゼロ成分）が最初の j 個の座標に含まれるならば、$\nabla f_p(\mathbf{x})$ の台は最初の $j+1$ 個の座標に含まれることを示せ。ここから、任意の一次手法の第 k 反復点の台は最初の k 個の座標に含まれることを結論せよ。

3. f_p の最小点の各成分が $x_i^\star = 1 - i/(p+1)$（$i \le p$ のとき）およびそれ以降でゼロとなること、最小値が $-\frac{L}{8}\bigl(1 - \frac{1}{p+1}\bigr)$ であること、そして $\|\mathbf{x}^\star\|^2 \le (p+1)/3$ が成り立つことを示せ。

4. $2k+1 \le d$ とする。関数 f_{2k+1} と f_k は、台が最初の k 個の座標に含まれるすべてのベクトル上で一致する。この事実を用いて、f_{2k+1} に適用された任意の一次手法が、$\mathbf{x}^\star$ を f_{2k+1} の最小点とするとき、次式を満たすことを示せ。

$$f_{2k+1}(\mathbf{x}_k) - \min f_{2k+1} \;\ge\; \frac{3L\,\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{32\,(k+1)^2}.$$

   この構成は $d \ge 2k+1$ を必要とするため、この下界が支配するのはおおよそ最初の $d/2$ 回の反復である。固定された次元では長い反復の末に漸近的により速く収束しうるが、d が数百万にのぼるニューラルネットワークでは、この制約が実質的な足かせになることはない。

5. $k \in \{5, 10, 25, 50, 100, 200, 400\}$ のそれぞれについて、k ステップの勾配降下法（$\eta = 1/L$）およびネステロフ法の更新 Equation 35.16（$\eta = 1/L$、$\beta_j = (j-1)/(j+2)$、ステップ $j = 1, 2, \ldots$）を f_{2k+1}（$d = 2k+1$）上で実行し、それぞれの最適性ギャップを上記の下界で割れ。k が大きくなるにつれて有界にとどまるのはどちらの比か。

8. [code] **モメンタムを伴う安定性の縁。●●**Cohen et al. (2021) は、モメンタムを加えると鋭さが $2/\eta$ とは異なるしきい値に落ち着くことを報告している。

   1. 問題 5a の反復行列を用いて、ヘビーボール法 Equation 35.12 が $f(x) = \frac{\lambda}{2}x^2$（$0 \le \beta < 1$）上で収束する必要十分条件が $0 < \eta\lambda < 2 + 2\beta$ であることを示せ。ネステロフ法の更新 Equation 35.16 についても、対応する安定区間を導出せよ。
   2. 9.1.4 節の安定性の縁の実験を変更し、$\beta \in \{0.25, 0.5\}$ に対してヘビーボール法およびネステロフ法の更新で訓練せよ。ただし各実行では、予測されるしきい値が 5 から 12 の間に入るようなステップサイズを選ぶこと。鋭さの推移を $2/\eta$ と (a) で導出したしきい値の両方と比較して追跡せよ。
   3. 鋭さはどちらのしきい値に落ち着くか。またこの結果は、モメンタムを用いて訓練する際に、初期化時点で測定した鋭さから η を選ぶことに対してどのような示唆を与えるか。

35.2 曲率法：ニュートン法、準ニュートン法、信頼領域法

勾配降下法は、最小化する関数の曲率を決して測らず、ただそれに耐えるだけである。最も急な方向がステップサイズの上限を抑え、その結果として最も平坦な方向は這うようにしか進まず、反復回数は条件数 κ に比例して（モメンタムを加えても $\sqrt{\kappa}$ に比例して）増大する（35.1.3 節）。共通の例題である $\frac{1}{2}(x_1^2 + 1000\,x_2^2)$ では、勾配降下法で 6,908 ステップ、モメンタムを用いても 315 ステップを要した。しかし、この問題は一見して直ちに解ける。勾配の各座標をそれぞれの曲率（1 または 1000）で割れば、わずか 1 ステップで最小点に直接到達するのである。曲率を単にやり過ごすのではなく考慮に入れたとき、オプティマイザはどこまで到達できるだろうか。

古典的な答えは、勾配降下法より 1 世紀半も前に誕生したニュートン法（Newton's method）である。各反復点において目的関数に二次近似を当てはめ、その二次関数の最小点へと一気に跳ぶ。本節では、このステップがどれほど速く収束するか、どれだけのコストがかかるか、どこで破綻するか、そして二階導関数が得られないときに勾配だけから曲率情報をどこまで復元できるかを調べる。最小点の近傍ではニュートン法は驚異的に速いが、ニューラルネットワークに対してはヘッセ行列の構成が高コストすぎ、さらに非凸関数上では極小点と同じくらい容易に鞍点へと引き寄せられてしまう。そこで続いて 2 つの打開策に目を向ける。連続する勾配の差分から曲率を再構築する準ニュートン法（quasi-Newton method）と、二次モデルが実際の関数とよく一致する近傍に各ステップを制限する信頼領域法（trust-region method）である。

深層ネットワークの訓練では、こうした古典的手法は一次のオプティマイザに比べてはるかに使われる頻度が低いが、その理由は率直に述べておく価値がある。完全なヘッセ行列は大きすぎて構成できず、またミニバッチノイズが勾配の差分から作る曲率推定を損なってしまうからである。それでもなお、根底にある考え方は深層学習の随所に再び現れる。Adam の座標ごとのスケーリングはニュートン法の逆ヘッセ行列の対角近似として働き（35.5 節）、曲率行列に加えるダンピングは信頼領域と同じ安定化の役割を果たす。心に留めておくべき中心的な結果は、ニュートン法の二次収束（quadratic convergence）を表す Equation 35.19 であり、準ニュートン法の項は初読時には飛ばしてもよい。なお 35.1 節からは、バックトラッキング直線探索（backtracking line search）の Armijo 条件と、鞍点で勾配降下法を停滞させたテスト関数 $\frac{1}{2}x^2 + \cos y$ という 2 つの要素を引き継ぐ。

35.2.1 ニュートン法

勾配降下法は、現在の点において f の線形近似を構築する。線形関数には最小値が存在しないため、アルゴリズム自身でステップ長を決めることができず、外部から与える学習率に頼らざるを得ない。これに対し二次近似は、上向きに曲がっている限り唯一の最小点をもつため、局所的な幾何構造によって長さが自動的に決まるステップを与える。

35.2.1.1 ニュートンステップ

ニュートン法では、局所的な曲率によってステップの方向と長さの両方を決定する（9.1.5.1 節）。$\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ とおくと、関数 f は $\mathbf{x}_k$ の近傍において変位 $\mathbf{p}$ に関する 2 次のテイラー展開で近似できる。

$$m_k(\mathbf{p}) = f(\mathbf{x}_k) + \mathbf{g}_k^{\top}\mathbf{p} + \tfrac{1}{2}\,\mathbf{p}^{\top}\nabla^2 f(\mathbf{x}_k)\,\mathbf{p}. \tag{35.17}$$

ヘッセ行列 $\nabla^2 f(\mathbf{x}_k)$ が正定値であるとき、モデル m_k は唯一の最小点をもつ狭義凸の鉢状関数となり、その最小点は $\nabla m_k(\mathbf{p}) = \mathbf{g}_k + \nabla^2 f(\mathbf{x}_k)\,\mathbf{p} = \mathbf{0}$ とおくことで求まる。これを $\mathbf{p}$ について解いてその点へ移動すると、**ニュートン法**が得られる。

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \big(\nabla^2 f(\mathbf{x}_k)\big)^{-1}\nabla f(\mathbf{x}_k). \tag{35.18}$$

Figure 35.6 に、1 次元におけるこのようなステップを 1 回分示す。Equation 35.18 にはステップサイズが現れないことに注目してほしい。逆ヘッセ行列が各方向をそれぞれの曲率に応じて再スケーリングしているのである。より広く見れば、ニュートン方向は前処理付きステップ $\mathbf{d} = -\mathbf{P}^{-1}\nabla f$（ただし $\mathbf{P}$ は正定値行列であり、ここでは $\mathbf{P} = \nabla^2 f$）の族に属する。$\nabla f^{\top}\mathbf{d} = -\nabla f^{\top}\mathbf{P}^{-1}\nabla f < 0$ が成り立つため、このような方向はすべて降下方向である。実際、ステップ長をユークリッドノルムではなくヘッセ行列が誘導するノルムで測るとき、ニュートン方向はまさに最急降下方向にほかならない（演習問題 1）。

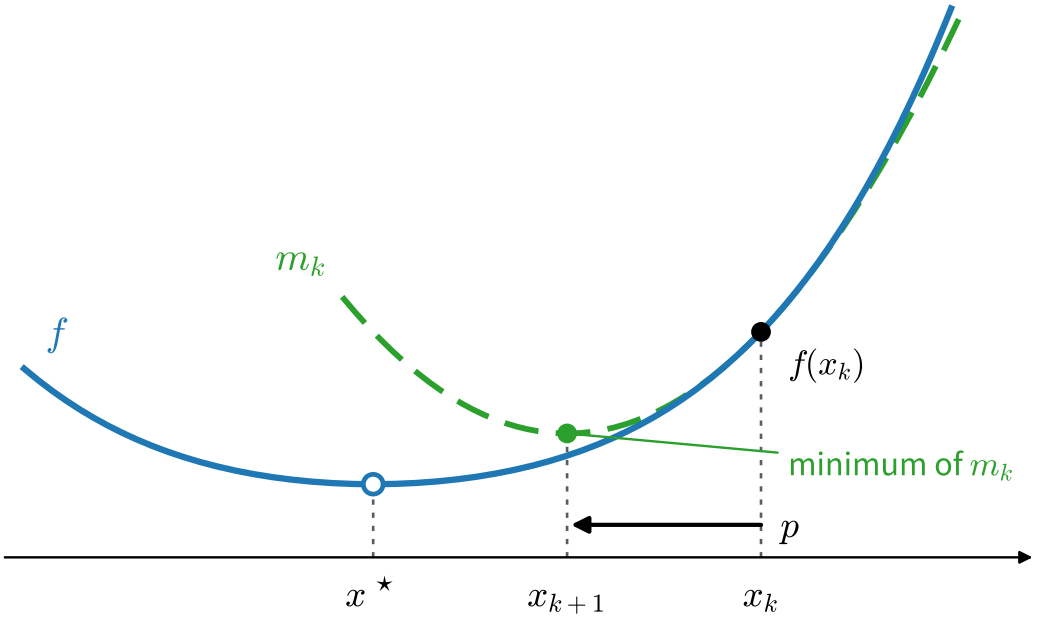


Figure 35.6: $f(x) = \frac{1}{4}x^4 + \frac{1}{2}x^2$ に対する $x_k = 1$ からの 1 回のニュートンステップ。モデル m_k（破線）は f の x_k における値・傾き・曲率を共有し、ステップ p はモデルの最小点 $x_{k+1} = \frac{1}{2}$ へと移動する。f の最小点は $x^\star = 0$ である。

正定値二次関数 $\tfrac{1}{2}\mathbf{x}^{\top}\mathbf{A}\mathbf{x}$ ではテイラーモデルが厳密に一致するため、1 回のニュートンステップで直ちに最小点 $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{A}^{-1}\mathbf{A}\mathbf{x}_0 = \mathbf{0}$ に到達し、条件数 κ の影響を受けない。その理由は単なる行列の相殺よりも深いところにある。ニュートン法は**アフィン不変**（affine-invariant）であり、その軌跡は座標系の選び方に依存しない。任意の可逆な線形変数変換 $\mathbf{x} = \mathbf{T}\mathbf{y}$ のもとで、$f(\mathbf{T}\mathbf{y})$ に対してニュートン法を実行すると、f に対するニュートン法の反復点をそのまま写した点列が厳密に得られる。対照的に、勾配降下法は非直交なあらゆる $\mathbf{T}$ のもとで経路が変化し、直交行列のスカラー倍に対してのみ実効ステップサイズが再スケーリングされるにとどまる（演習問題 1）。任意の正定値二次関数は何らかの線形変換によって完全な球面へと変換できるため、条件数がニュートン法に影響を与えることはあり得ない。ニュートン法にとって細長い谷とは、引き伸ばされた紙の上に描かれた丸い鉢にすぎないのである。

35.2.1.2 二次収束

一般の非二次な目的関数では、テイラーモデルは局所近似にすぎないため、もはや 1 ステップでは済まない。それでも失われないのが驚異的な収束速度である。反復点が最小点の近傍に入ると、ニュートン法は幾何収束よりもはるかに速く収束し、その証明には微分積分学の基本定理以上のものはほとんど必要ない。$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}^{\star}$ を反復 k における誤差とする。

命題（二次収束）。 **$\nabla f(\mathbf{x}^{\star}) = \mathbf{0}$ とし、$\mathbf{x}^{\star}$ の周りのある球上でヘッセ行列が $\nabla^2 f(\mathbf{x}) \succeq \mu \mathbf{I}$ および $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \le L_2 \|\mathbf{x} - \mathbf{y}\|$ を満たすと仮定する。$\mathbf{x}_k$ がこの球内にあるならば、次が成り立つ。**

$$\|\mathbf{e}_{k+1}\| \;\le\; \frac{L_2}{2\mu}\,\|\mathbf{e}_k\|^2 . \tag{35.19}$$

証明。 $\nabla^2 f_k = \nabla^2 f(\mathbf{x}_k)$ と略記する。$\nabla f(\mathbf{x}^{\star}) = \mathbf{0}$ であるため、ニュートン更新から $\mathbf{x}^{\star}$ を引くと $\mathbf{e}_{k+1} = \mathbf{e}_k - (\nabla^2 f_k)^{-1}\mathbf{g}_k = (\nabla^2 f_k)^{-1}\bigl[\nabla^2 f_k\,\mathbf{e}_k - (\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^{\star}))\bigr]$ が得られる。$\mathbf{x}^{\star}$ から $\mathbf{x}_k$ への線分に微分積分学の基本定理を適用すると、勾配の差は $\int_0^1 \nabla^2 f(\mathbf{x}^{\star} + t\mathbf{e}_k)\,\mathbf{e}_k\,dt$ と表されるため、角括弧内の項は $\int_0^1 \bigl[\nabla^2 f_k - \nabla^2 f(\mathbf{x}^{\star} + t\mathbf{e}_k)\bigr]\mathbf{e}_k\,dt$ となる。$\mathbf{x}_k$ と $\mathbf{x}^{\star} + t\mathbf{e}_k$ の差は $(1-t)\mathbf{e}_k$ であるから、ヘッセ行列のリプシッツ評価より、この積分のノルムは高々 $\int_0^1 L_2 (1-t)\|\mathbf{e}_k\|^2\,dt = \tfrac{L_2}{2}\|\mathbf{e}_k\|^2$ で抑えられる。これと $\|(\nabla^2 f_k)^{-1}\| \le 1/\mu$ を合わせれば証明が完了する。$\blacksquare$

Equation 35.19 が意味するところを読み取るため、両辺に $L_2/(2\mu)$ を掛けると $\tfrac{L_2}{2\mu}\|\mathbf{e}_{k+1}\| \le \bigl(\tfrac{L_2}{2\mu}\|\mathbf{e}_k\|\bigr)^2$ となる。スケールされた誤差は毎ステップ二乗される。初期値が 1 未満であれば誤差は単調に縮小し、それ以降のすべての反復点は球内にとどまる。いったん 0.1 を下回れば、10^{-2}、10^{-4}、10^{-8} へと急落し、各反復で正確な桁数がおよそ倍増していく。これを共通の例題における勾配降下法（精度を 1 桁上げるのに約 1,150 ステップを要する）と比較してほしい。ただし注意点も仮定の中にそのまま書き込まれている。この二次収束率はあくまで局所的なものであり、反復点がすでに $\mathbf{x}^{\star}$ の十分近くに入るまでは何も保証しない。

$\mathbf{x}^{\star}$ から遠い領域では、たとえヘッセ行列が正定値であっても、純粋なニュートンステップはオーバーシュートしうる。これに対する標準的な安全策が**ダンピング**である。更新を $\mathbf{x}_k + \alpha_k \mathbf{p}_k$（ただしニュートン方向は $\mathbf{p}_k = -(\nabla^2 f(\mathbf{x}_k))^{-1}\mathbf{g}_k$）とし、$\alpha_k \le 1$ を Armijo 条件 $f(\mathbf{x}_k + \alpha \mathbf{p}_k) \le f(\mathbf{x}_k) + c\,\alpha\,\mathbf{g}_k^{\top}\mathbf{p}_k$ が成り立つまでバックトラッキングで選ぶ（35.1 節の Armijo 則を一般の方向へ拡張した形）。正定値なヘッセ行列をもつ最小点の近傍では、完全なステップ $\alpha = 1$ が $c < \tfrac{1}{2}$ である限り自動的にこの判定を通過するため、ひとたび二次収束が始まればダンピングによる余分なオーバーヘッドは生じない。

これらの予測をコードで検証するため、ℓ_2 正則化ロジスティック回帰を $n = 200$ 個の人工データ例（$d = 5$ 次元）に対して用いる。

$$f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-y_i\, \mathbf{x}_i^{\top}\mathbf{w})\right) + \frac{\nu}{2}\|\mathbf{w}\|^2, \qquad \nu = 0.1,$$

ここで二値ラベル $y_i = \pm 1$ はロジスティックモデルからサンプリングする（重み減衰の係数には ν を用い、λ は固有値のために取っておく）。以下のセットアップセルでは、参照用の最適解 $\mathbf{w}^\star$ も 4,000 回の勾配ステップ（ステップサイズ $\eta = 0.5$）によって計算しておく。このステップサイズは $2/L$ を十分に下回っている。ここでは $L = \lambda_{\max}(\mathbf{X}^{\top}\mathbf{X})/(4n) + \nu$ が約 0.4 だからである（演習問題 2）。

```
rng = np.random.default_rng()
n, d, nu = 200, 5, 0.1
w_true = np.array([1.0, -2.0, 1.5, 0.0, 2.0])
X = rng.normal(size=(n, d))
y = np.where(rng.random(n) < 1 / (1 + np.exp(-X @ w_true)), 1.0, -1.0)

def loss(w, X, y, nu):
    return np.logaddexp(0, -y * (X @ w)).mean() + nu / 2 * (w @ w)

def grad_rows(w, X, y, nu):                  # row i: gradient of example i
    s = 1 / (1 + np.exp(y * (X @ w)))        # sigmoid(-y_i x_i^T w)
    return -(y * s)[:, None] * X + nu * w

w_star = np.zeros(d)
for _ in range(4000):                        # full-batch reference optimum
    w_star = w_star - 0.5 * grad_rows(w_star, X, y, nu).mean(axis=0)
```

これで両方の予測を直接検証できる。共通の二次関数 $\mathrm{diag}(1, 1000)$ では、1 回のニュートンステップが、最適な固定ステップサイズの勾配降下法に必要だった $\log 10^6 / \log(1001/999) \approx 6{,}908$ ステップ（35.1.3 節）に取って代わるはずである。また、非二次なロジスティック損失では、局所収束の段階に入れば勾配ノルムの指数が反復ごとにおよそ倍増するはずである。

```
A2 = np.diag([1.0, 1000.0])                  # the running example, kappa = 1000
x = np.array([1.0, 1.0])
x_new = x - np.linalg.solve(A2, A2 @ x)      # one Newton step
rate = (1000 - 1) / (1000 + 1)
print(f'Newton on the quadratic: |x_1| = {np.linalg.norm(x_new):.1e} after one step')
print(f'GD at the optimal step contracts by {rate:.4f}; it needs '
      f'{int(np.ceil(np.log(1e-6) / np.log(rate)))} steps for |x_k| <= 1e-6 |x_0|')
w = np.zeros(d)                              # Newton on the logistic loss
for it in range(6):
    s = 1 / (1 + np.exp(y * (X @ w)))
    g = grad_rows(w, X, y, nu).mean(axis=0)
    hess = (X * (s * (1 - s))[:, None]).T @ X / n + nu * np.eye(d)
    w = w - np.linalg.solve(hess, g)
    print(f'Newton iteration {it + 1}:  |grad| = {np.linalg.norm(g):.1e}')
print(f'max |w - w_star| = {np.abs(w - w_star).max():.1e}')
```

```
Newton on the quadratic: |x_1| = 0.0e+00 after one step
GD at the optimal step contracts by 0.9980; it needs 6908 steps for |x_k| <= 1e-6 |x_0|
Newton iteration 1:  |grad| = 3.5e-01
Newton iteration 2:  |grad| = 4.1e-02
Newton iteration 3:  |grad| = 1.7e-03
Newton iteration 4:  |grad| = 3.0e-06
Newton iteration 5:  |grad| = 9.9e-12
Newton iteration 6:  |grad| = 1.6e-17
max |w - w_star| = 4.4e-16
```

予測はどちらも的中する。二次関数では、1 回のニュートンステップで誤差がゼロになるのに対し、最適な固定ステップの勾配降下法は 1 ステップあたり 0.998 倍ずつしか縮小せず、$6{,}908$ 回の反復を要する。ロジスティック損失では、勾配ノルムは十分の一の数倍程度から始まり、小さくなると指数がおよそ倍増していき、5 回目の反復までに 10^{-8} を下回る。わずか 6 回の反復の後、ニュートン法の反復点は参照用の最適解と約 10^{-15} の誤差で一致し、参照解そのものの精度に到達する。勾配ステップで $4{,}000$ 回かかった計算を、連立一次方程式を 6 回解くだけで成し遂げたのである。

35.2.1.3 1 ステップあたりのコスト

残念ながら、反復回数だけを比較するのはニュートン法を過大評価することになる。ヘッセ行列は d^2 個の成分を含み、Equation 35.18 の連立一次方程式を解くには一般に 1 ステップあたり $O(d^3)$ 回の演算を要する。$d = 5$ であればこのコストは無視できるが、控えめに見積もって 10^8 個のパラメータをもつニューラルネットワークでさえヘッセ行列の成分数は 10^{16} 個に達し、最初の 1 ステップを踏み出す前に厳密なニュートン法は選択肢から外れる（9.1.5.1 節）。実用的な代替手法としては、以下で述べる準ニュートン法のように勾配からヘッセ行列を近似するか、安価な対角前処理行列や構造化前処理行列を用いる（35.5 節）。

35.2.1.4 鞍点

計算コストはニュートン法の唯一の欠点ではなく、最も示唆に富む欠点というわけでもない。ニュートンステップは $\nabla m_k = \mathbf{0}$ を解くため、二次モデルの停留点が極小点であれ、極大点であれ、鞍点であれ、そこへ一気に跳んでしまう。関数 $f(x, y) = \tfrac{1}{2}x^2 + \cos y$（35.1 節）に立ち返ろう。この関数は原点に鞍点をもち、最小値 -1 を $(0, \pi)$ でとる。初期点を $(0.5, 0.3)$ とすると、ここでの勾配は $(0.5, -\sin 0.3) = (0.5, -0.296)$ であり、ヘッセ行列 $\operatorname{diag}(1, -\cos 0.3) = \operatorname{diag}(1, -0.955)$ は不定値となる。すなわち、曲面は x 方向に上向き、y 方向に下向きに曲がっている。この点からニュートンステップが y を動かす向きは、サイズ $\eta = 1$ の通常の勾配ステップと比べてどちらになるだろうか。

```python
def f_s(v):
    return 0.5 * v[0]**2 + np.cos(v[1])

def grad_s(v):
    return np.array([v[0], -np.sin(v[1])])

def hess_s(v):
    return np.diag([1.0, -np.cos(v[1])])

x_n = x_g = np.array([0.5, 0.3])
ys_n, ys_g = [], []
for _ in range(6):
```

```
    x_n = x_n - np.linalg.solve(hess_s(x_n), grad_s(x_n))    # Newton
    x_g = x_g - 1.0 * grad_s(x_g)                            # GD, eta = 1
    ys_n.append(x_n[1])
    ys_g.append(x_g[1])
print('Newton y:', np.array(ys_n[:3]), f'  f = {f_s(x_n):.4f}')
print('GD     y:', np.array(ys_g).round(4), f'  f = {f_s(x_g):.4f}')
```

```
Newton y: [-9.33624961e-03  2.71275924e-07 -6.61744490e-21]   f = 1.0000
GD     y: [0.5955 1.1565 2.0718 2.9489 3.1404 3.1416]   f = -1.0000
```

ニュートン法の最初のステップは x を 0 に、y を $0.3 - \tan 0.3 = -0.0093$ に写し、2 回目のステップは y を 2.7×10^{-7} に写す。ここで起きているのはニュートン法の急速な局所収束（$y - \tan y \approx -y^3/3$ であるため、ここでは三次収束でさえある）であり、反復点を鞍点 $(0, 0)$（$f = 1$）へと一直線に追い込んでいる。これに対し勾配降下法は y を逆方向へ動かし、0.60、1.16、2.07、2.95、3.140、3.1416 を経て最小点 $(0, \pi)$（$f = -1$）に到達する。より精緻なモデルが、かえってより悪い点へと導いたのである。y 方向の曲率が負であるため二次モデルの停留点は鞍点となっているのだが、ニュートン法は鞍点と極小点の違いを区別できない。信頼領域（35.2.3 節）がこの欠陥を解消する。

35.2.2 準ニュートン法

ニュートン法は明示的な二階導関数を必要とする。代わりに、すでに計算している勾配から曲率を推定できないだろうか。あるステップに沿って勾配が急激に変化するとき、目的関数はその方向に大きな曲率をもっているはずである。こうした変化を記録するオプティマイザは、二階導関数を一度も評価することなく、ヘッセ行列のモデルを徐々に組み立てることができる。

35.2.2.1 割線法

1 次元では、この発想は直ちに形になる。直近 2 つの勾配を結ぶ弦の傾きによって f'' を推定すればよい。

$$h_k = \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}, \qquad x_{k+1} = x_k - \frac{f'(x_k)}{h_k}.$$

これが**割線法**（secant method）であり、ニュートンステップの $f''(x_k)$ を勾配の有限差分に置き換えたものにほかならない。関数 $f(x) = e^x - 2x$（最小点は $\log 2 \approx 0.6931$）を考え、$x_0 = 0$ および $x_1 = 1$ から開始しよう。最初の割線の傾きは $h_1 = (e-2) - (1-2) = e - 1 \approx 1.718$ であり、これから $x_2 = 1 - (e-2)/1.718 \approx 0.582$ が得られる。そこから誤差は 1.1×10^{-1}、1.6×10^{-2}、9.3×10^{-4}、7.7×10^{-6}、3.6×10^{-9} と縮小していく。各誤差は直前 2 つの誤差の積のおよそ半分になっており（この半分の係数は最小点における $f'''/(2f'')$ に由来する）、正確な桁数は 1 ステップあたり 1.6 に漸近する倍率で増加する。これはニュートン法の倍率 2 よりわずかに遅いものの、各反復で必要なのは 1 回の勾配評価のみであり、二階導関数は不要である。1 次元の二次関数 $\frac{\lambda}{2}x^2$ では、割線の傾きは厳密に λ と等しくなり、x_1 からのステップは最小点へ直接到達する。

d 次元では、対応する割線条件は 1 本のベクトル方程式にすぎず、$d \times d$ 行列を特定するには到底足りない。1 回のステップで明らかになるのは 1 つの方向に沿った曲率だけだからである。残りの方向を埋めるには、原理に基づいた規則が必要になる。

35.2.2.2 BFGS

準ニュートン法は、逆ヘッセ行列の近似 $\mathbf{M}_k$ を保持し、観測された勾配の変化を用いて各ステップの後にこれを更新する。$\mathbf{x}_k$ から $\mathbf{x}_{k+1}$ へ移動した後、次の割線ペアを記録する。

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \qquad \mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k).$$

二次関数 $\tfrac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x}$ では勾配の差分が $\mathbf{y}_k = \mathbf{A}\mathbf{s}_k$ を満たすため、真の逆ヘッセ行列は $\mathbf{y}_k$ を $\mathbf{s}_k$ に写し戻す。そこで、更新後の近似行列 $\mathbf{M}_{k+1}$ にも同じ**割線方程式**（secant equation）を満たすことを課す。

$$\mathbf{M}_{k+1}\,\mathbf{y}_k = \mathbf{s}_k. \tag{35.20}$$

Equation 35.20 を満たすすべての対称行列のうち、**BFGS**（Broyden、Fletcher、Goldfarb、Shanno にちなむ）は、適切に重み付けされたフロベニウスノルムにおいて直前の推定値 $\mathbf{M}_k$ に最も近い行列を選ぶ。

$$\mathbf{M}_{k+1} = (\mathbf{I} - \gamma_k \mathbf{s}_k \mathbf{y}_k^\top)\,\mathbf{M}_k\,(\mathbf{I} - \gamma_k \mathbf{y}_k \mathbf{s}_k^\top) + \gamma_k \mathbf{s}_k \mathbf{s}_k^\top, \qquad \gamma_k = \frac{1}{\mathbf{y}_k^\top \mathbf{s}_k}. \tag{35.21}$$

Equation 35.21 を導出しなくとも、その仕組みはそのまま読み取れる（見やすさのためペアの添字 k は省略する）。右側の因子 $\mathbf{I} - \gamma \mathbf{y}\mathbf{s}^\top$ は、$\mathbf{y}$ をゼロに写す（$\gamma\,\mathbf{s}^\top\mathbf{y} = 1$ であるため）一方、$\mathbf{s}$ に直交するすべてのベクトルをそのまま保つ。したがって第 1 項は $\mathbf{M}_k$ が以前 $\mathbf{y}$ 方向について予測していた内容を消去し、第 2 項が新しい観測を書き込むことで $\mathbf{M}_{k+1}\mathbf{y} = \gamma\,\mathbf{s}\,(\mathbf{s}^\top\mathbf{y}) = \mathbf{s}$ となり、割線方程式が満たされる。

それに劣らず重要な点として、この更新は正定値性を保つため、次の探索方向 $-\mathbf{M}_{k+1}\nabla f(\mathbf{x}_{k+1})$ が引き続き降下方向であることが保証される。任意の非ゼロベクトル $\mathbf{z}$ に対して $\mathbf{w} = (\mathbf{I} - \gamma \mathbf{y}\mathbf{s}^\top)\mathbf{z}$ とおくと、$\mathbf{z}^\top \mathbf{M}_{k+1}\mathbf{z} = \mathbf{w}^\top \mathbf{M}_k \mathbf{w} + \gamma(\mathbf{s}^\top\mathbf{z})^2$ となる。$\mathbf{M}_k \succ 0$ かつ $\gamma > 0$ のとき、両項はともに非負であり、同時に 0 になることはない。なぜなら、$\mathbf{w} = \mathbf{0}$ ならば $\mathbf{z} = \gamma(\mathbf{s}^\top\mathbf{z})\,\mathbf{y}$ となり、これには $\mathbf{s}^\top\mathbf{z} \neq 0$ が必要だからである。この条件 $\gamma > 0$ は**曲率条件**（curvature condition）$\mathbf{y}_k^\top \mathbf{s}_k > 0$ であり、ステップに沿って方向微分が増加したことを意味する。これは任意の狭義凸目的関数では自動的に成り立ち、一般の目的関数では直線探索において Wolfe 条件として知られる第 2 の判定条件によって課される (Nocedal and Wright 2006)。

割線ペアはどれほど速く真の曲率を再構築するだろうか。初期点 $(1, 1)$ から 2 次元の二次関数 $\mathrm{diag}(1, 10)$ に対して、$\mathbf{M}_0 = \mathbf{I}$ および厳密な直線探索 $\alpha = -\mathbf{g}^\top\mathbf{p} / (\mathbf{p}^\top \mathbf{A}\mathbf{p})$ を用いて BFGS を実行し、各ステップ後の $\mathbf{x}$ と $\mathbf{M}$ を確認してみよう。

```
def bfgs_update(M, s, y):
    gamma = 1.0 / (y @ s)
    V = np.eye(len(s)) - gamma * np.outer(s, y)
    return V @ M @ V.T + gamma * np.outer(s, s)

A = np.diag([1.0, 10.0])
x, M = np.array([1.0, 1.0]), np.eye(2)
for k in range(2):
    g = A @ x
    p = -M @ g
    alpha = -(g @ p) / (p @ A @ p)           # exact line search
```

```
    s = alpha * p
    M = bfgs_update(M, s, A @ s)              # y = A s on a quadratic
    x = x + s
    print(f'step {k + 1}: x = {x.round(4)},  M = {M.round(4).tolist()}')
print(f'A^-1 = {np.linalg.inv(A).tolist()},  '
      f'max |M_2 - A^-1| = {np.abs(M - np.linalg.inv(A)).max():.0e}')
```

```
step 1: x = [ 0.8991 -0.009 ],  M = [[1.009, -0.0001], [-0.0001, 0.1]]
step 2: x = [0. 0.],  M = [[1.0, -0.0], [-0.0, 0.1]]
A^-1 = [[1.0, 0.0], [0.0, 0.1]],  max |M_2 - A^-1| = 1e-16
```

わずか 1 ステップの後、$\mathbf{M}_1 \approx \mathrm{diag}(1.009, 0.100)$ は、直前に探索した急な方向に沿ってすでに $\mathbf{A}^{-1}$ に近づいている。2 ステップ目の後には、浮動小数点の丸め誤差の範囲内で $\mathbf{M}_2 = \mathbf{A}^{-1}$ となり、$\mathbf{x}_2 = \mathbf{0}$ に達する。この 2 次元での挙動はそのまま一般化できる。任意の d 次元の正定値二次関数において厳密な直線探索を用いると、各割線方程式はその後の更新を経ても満たされたまま保たれ、各ステップは互いに $\mathbf{A}$ 共役（$\mathbf{s}_i^{\top}\mathbf{A}\mathbf{s}_j = 0$、ただし $i \neq j$）となる。その結果、d ステップ後には $\mathbf{M}_d\mathbf{A} = \mathbf{I}$ となり、BFGS は厳密な最小点に到達する（演習問題 2）。

ただし、有限回での終了は厳密な直線探索を伴う二次関数に特有の性質であり、一般的な保証と受け取るべきではない。非二次な目的関数では直線探索は非厳密になり、真のヘッセ行列もステップごとに変化するうえ、確率的勾配が $\mathbf{y}_k$ を汚染する。2 つのミニバッチ勾配の差は、曲率ではなくサンプリングノイズに支配されることが多いからである。さらに、完全な BFGS は密な $d \times d$ 行列を保持する。**L-BFGS** は直近 m 個の割線ペアのみを保持し、得られる逆ヘッセ行列近似（定数倍した単位行列にランクが高々 $2m$ の補正を加えたもの）を $O(md)$ の計算時間とメモリで適用することで、このメモリコストを回避する (Liu and Nocedal 1989)。決定論的な中規模問題では非常に有効に機能するものの、新たなミニバッチノイズによって勾配の差分が信頼できなくなる状況では依然として扱いにくい。

35.2.3 信頼領域

ニュートン法が $(0.5, 0.3)$ で道を誤ったのは、局所的な二次モデルを空間全体にわたって信頼したからである。そのモデルは単一の点における導関数から構築されたものであり、近傍でしか f を正確に反映しない。信頼領域法はこの限界を文字通りに受け止め、負の曲率の方向を含む完全な曲率情報を保持したまま、モデルを信頼できる領域内にステップを制限する。

35.2.3.1 部分問題

点 $(0.5, 0.3)$ において、$\frac{1}{2}x^2 + \cos y$ の二次モデルは曲率 -0.955 を y 方向にもつ。下向きに際限なく曲がっているため、制約なしの最小点を求めることは意味をなさない。そこで**信頼領域法**は、近似が成り立つと考える範囲を表す半径 Δ_k の球の内部で二次モデルを最小化する。

$$\min_{\|\mathbf{p}\| \le \Delta_k} m_k(\mathbf{p}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^{\top}\mathbf{p} + \tfrac{1}{2}\mathbf{p}^{\top}\mathbf{B}_k\mathbf{p}. \tag{35.22}$$

ここで $\mathbf{B}_k$ は厳密なヘッセ行列、またはその対称近似である。任意の連続関数はコンパクト集合上で最小値を達成するため、$\mathbf{p}$ を閉球に制限することで、$\mathbf{B}_k$ が負の固有値をもつ場合でも最小点の存在が保証される。さらに都合のよいことに、この制約付き問題の厳密解は明快に特徴付けられる。

命題（部分問題の解）。 **$\mathbf{p}$ を $\|\mathbf{p}\| \le \Delta$ を満たすステップとする。これが Equation 35.22 の解となるのは、次を満たす $\tau \ge 0$ が存在するとき、かつそのときに限られる。**

$$(\mathbf{B}+\tau\mathbf{I})\,\mathbf{p}=-\mathbf{g},\qquad \mathbf{B}+\tau\mathbf{I}\succeq\mathbf{0},\qquad \tau\,(\Delta-\|\mathbf{p}\|)=0.$$

この結果は Nocedal and Wright (2006) の定理 4.1 であり、演習問題 3 でこれらの条件の十分性を証明する。幾何学的には、信頼領域は一様なシフト τ を $\mathbf{B}$ のすべての固有値に加え、シフト後の二次関数をちょうど凸にする。ここで 2 つの場合が生じる。$\mathbf{B}$ が正定値であり、かつ制約なしのニュートンステップがすでに球の内部に収まっているときは、$\tau=0$ となってそのままニュートンステップを採用する。そうでない場合、解は境界 $\|\mathbf{p}\|=\Delta$ 上にあり、τ はスカラー方程式 $\|(\mathbf{B}+\tau\mathbf{I})^{-1}\mathbf{g}\|=\Delta$ の区間 $\tau>-\lambda_{\min}(\mathbf{B})$ における根として定まる（この区間では τ が大きくなるにつれて左辺は狭義単調減少する）。

このスカラー方程式が根をもたないのは、退化した**ハードケース**、すなわち $\mathbf{g}$ が $\mathbf{B}$ の最小固有値に属する固有ベクトルと直交する場合に限られる（演習問題 3 の小問 d で扱う）。35.6 節で見るように、シフト τ は制約 $\tfrac12\|\mathbf{p}\|^2\le\tfrac12\Delta^2$ を課すラグランジュ乗数にほかならない。

反復の間で、半径 Δ_k はモデルが f の実際の減少量をどれほど正確に予測したかに応じて適応的に変化する。Equation 35.22 を解いた後、予測された減少量に対する実際の減少量の比を作る。

$$r_k=\frac{f(\mathbf{x}_k)-f(\mathbf{x}_k+\mathbf{p}_k)}{m_k(\mathbf{0})-m_k(\mathbf{p}_k)}. \tag{35.23}$$

この比が 1 に近いことは、二次モデルが目的関数とよく一致したことを示す。以下の実装では、$r_k>0.1$ である限りステップを採択する。それとは独立に、半径を $\Delta\leftarrow\Delta/4$ へと縮小する操作を $r_k<\tfrac14$ のときに行い、$r_k>\tfrac34$ かつステップが境界 $\|\mathbf{p}_k\|=\Delta_k$ に達したときには半径を 2 倍にする。

35.2.3.2 鞍点における信頼領域ステップ

点 $(0.5,0.3)$ において $\Delta=0.5$ とし、1 回の信頼領域ステップを手計算で求めてみよう。この点では $\mathbf{g}=(0.5,-0.296)$、$\mathbf{B}=\operatorname{diag}(1,-0.955)$ である。$\mathbf{B}+\tau\mathbf{I}$ を半正定値にするには $\tau\ge0.955$ が必要であるため、$\tau=0$ は不可能であり、ステップは球の境界上になければならない。最適性条件の連立方程式は $(1+\tau)p_1=-0.5$ および $(\tau-0.955)p_2=0.296$（制約は $p_1^2+p_2^2=0.25$）となり、その解は $\tau\approx1.60$、$\mathbf{p}\approx(-0.19,0.46)$ である。$(0.31,0.76)$ へ移動すると、f は 1.080 から 0.771 へと下がる。モデルが予測した減少量 0.316 に対して実際の減少量は 0.309 であるため、比は $r=0.98$ となり、ステップは採択されて半径は 2 倍になる。決定的に重要なのは、モデルが減少する負の曲率方向に沿って信頼領域ステップが鞍点から**遠ざかる**向きに進むのに対し、制約なしのニュートンステップ $(-0.5,-0.31)$ は鞍点へ向かって一直線に進んだという点である。

この部分問題の解が数回の反復でどこへ向かうかを見るため、まず `tr_step(g, B, Delta)` を実装する。この関数は、$\mathbf{B}\succ\mathbf{0}$ かつ $\|\mathbf{B}^{-1}\mathbf{g}\|\le\Delta$ のときには制約なしのニュートンステップを返し、そうでないときには τ に関して二分法を行い、$\|(\mathbf{B}+\tau\mathbf{I})^{-1}\mathbf{g}\|=\Delta$ となる点を求める。

```
def tr_step(g, B, Delta):
    """Minimize g p + p B p / 2 over |p| <= Delta; return (p, tau)."""
    eig_min = np.linalg.eigvalsh(B).min()
    if eig_min > 0 and np.linalg.norm(np.linalg.solve(B, -g)) <= Delta:
        return np.linalg.solve(B, -g), 0.0     # Newton step fits: tau = 0
    lo = hi = max(0.0, -eig_min)
    hi += 1e3
    for _ in range(100):                       # bisection for |p(tau)| = Delta
```

```
            tau = (lo + hi) / 2
            p = np.linalg.solve(B + tau * np.eye(len(g)), -g)
            lo, hi = (tau, hi) if np.linalg.norm(p) > Delta else (lo, tau)
        return p, tau
```

初期点 $\mathbf{x} = (0.5, 0.3)$、半径 $\Delta = 0.5$ から開始して f_s に対し信頼領域法を 6 反復実行し、各ステップでの反復点、目的関数値、ステップノルム $\|\mathbf{p}\|$、シフト τ、一致比 r、半径 Δ を出力する。

```
x, Delta = np.array([0.5, 0.3]), 0.5
print(' k        x            f      |p|    tau      r  Delta')
for k in range(6):
    g, B = grad_s(x), hess_s(x)
    p, tau = tr_step(g, B, Delta)
    r = (f_s(x) - f_s(x + p)) / -(g @ p + 0.5 * p @ B @ p)
    print(f'{k:2d} ({x[0]:6.3f},{x[1]:6.3f}) {f_s(x):7.4f} '
          f'{np.linalg.norm(p):6.3f} {tau:6.3f} {r:6.2f} {Delta:5.2f}')
    if r > 0.1:                                 # accept the step
        x = x + p
    if r < 0.25:
        Delta /= 4                              # poor prediction: shrink
    elif r > 0.75 and np.linalg.norm(p) > 0.99 * Delta:
        Delta *= 2                              # good, and on the boundary
print(f'final x = ({x[0]:.4f}, {x[1]:.4f}),  f = {f_s(x):.4f}')
```

```
 k        x            f      |p|    tau      r  Delta
 0 ( 0.500, 0.300)  1.0803  0.500  1.596   0.98  0.50
 1 ( 0.307, 0.761)  0.7711  1.000  1.419   0.87  1.00
 2 ( 0.180, 1.753) -0.1652  2.000  0.311   0.41  2.00
 3 ( 0.043, 3.749) -0.8205  0.696  0.000   0.88  2.00
 4 ( 0.000, 3.054) -0.9962  0.088  0.000   1.00  2.00
 5 ( 0.000, 3.142) -1.0000  0.000  0.000   1.00  2.00
final x = (0.0000, 3.1416),  f = -1.0000
```

1 行目は先ほどの手計算と一致している。半径は 0.5 から 1、さらに 2 へと倍増し、3 回目のステップ（比 0.41）は半径を変えずに採択される。3 番目の反復点 $(0.18, 1.75)$ に達するまでに、アルゴリズムはヘッセ行列が正定値となる領域に入っている。次の反復点以降は制約なしのニュートンステップが $\tau = 0$ のまま信頼領域の球内に収まるため、手法は継ぎ目なくニュートン法へと切り替わり、最小点 $(0, \pi)$（$f = -1$）へ二次収束する。6 回の反復で十分であり、Figure 35.7 に両者の軌跡の対比を示す。

信頼領域の部分問題を厳密に解くには、$\mathbf{B}_k + \tau \mathbf{I}$ を τ の各試行値に対して分解する必要がある。そのため大規模な実装では、1 回あたり勾配評価 1 回分とほぼ同じコストで計算できるヘッセ行列・ベクトル積を用いて近似的に解く（34.5 節）。こうした高速化を用いてもなお、大規模でノイズの多い深層学習の目的関数において、BFGS や信頼領域法が確率的勾配降下法に取って代わることはない。

関数 $\frac{1}{2}x^2 + \cos y$ の鞍点は、ここまでで 2 つのオプティマイザの足をすくった。勾配降下法は直線 $y = 0$ 上の初期点から開始したときにそこで停滞し、ニュートン法は一般的な初期点からそこへ一直線に飛び込んだ。信頼領域はそこから脱出できるものの、各局所モデルを真の目的関数と突き合わせて検証することが不可欠であった。しかし、こうした予防策が一切不要となる大きな関数クラスが存在する。鞍点も偽の盆地ももたず、すべての停留点が大域的最小点となる関数群である。そのような関数を見分けることが 35.3 節の主題である。

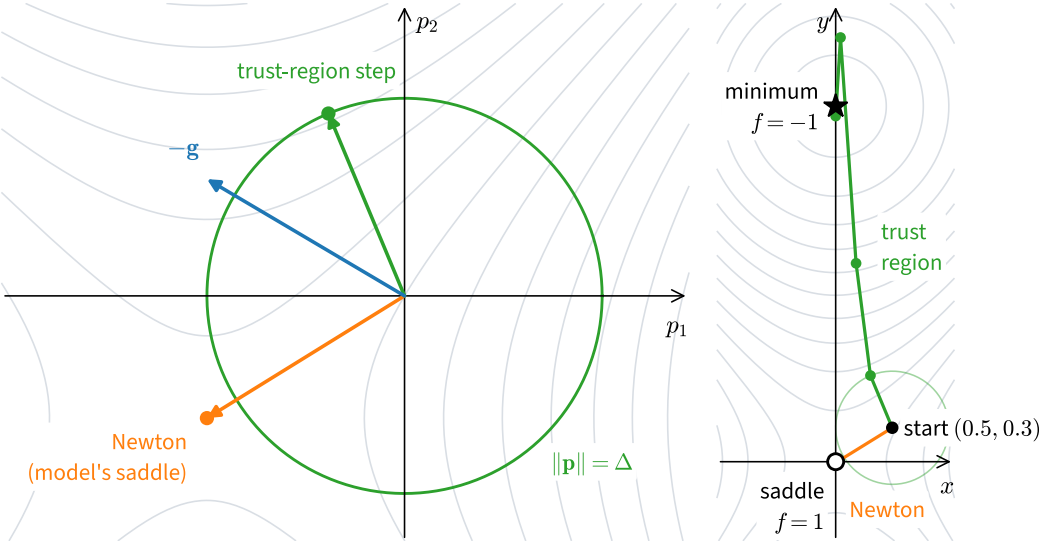


Figure 35.7: 点 $(0.5, 0.3)$ から開始した $\frac{1}{2}x^2 + \cos y$（ヘッセ行列は $\mathrm{diag}(1, -0.96)$）に対するニュートン法と信頼領域法。(a) 局所的な二次モデルは鞍面をなす。ニュートンステップはその停留点へと移動するのに対し、半径 $\Delta = 0.5$ の球上でモデルを最小化すると、負の曲率方向に沿って鞍点から遠ざかる。(b) ニュートン法は鞍点 $(0, 0)$（$f = 1$）へ二次収束する一方、信頼領域法（初期半径を表示）は最小点 $(0, \pi)$（$f = -1$）に 6 ステップで到達し、その間に半径を $0.5, 1, 2$ と拡大していく。

35.2.4 まとめ

ニュートン法は各反復点において f の局所的な二次モデルを最小化する。条件数にかかわらず任意の正定値二次関数を 1 ステップで解き、正定値かつリプシッツ連続なヘッセ行列をもつ最小点の近傍では反復ごとに誤差を二乗する。ただし、その速さには 2 つの障壁が伴う。各ステップに $O(d^2)$ のメモリと $O(d^3)$ の演算量を要するためニューラルネットワークでは到底扱えないこと、そして制約なしのステップが極小点と同じくらい強く鞍点にも引き寄せられることである。BFGS は過去のステップ差分と勾配差分から逆ヘッセ行列を再構築することで明示的な二階導関数を不要にし、d ステップで d 次元の二次関数上の逆ヘッセ行列を完全に復元する。信頼領域法は、各ステップを球内に閉じ込め、予測された減少量と f の実際の減少量を比較して球を拡大・縮小することで、任意の二次モデルを安全に利用できるようにする。

35.2.5 演習問題

1. **再パラメータ化と前処理。** $\mathbf{T}$ を可逆行列、$h(\mathbf{y}) = f(\mathbf{T}\mathbf{y})$ とし、$\mathbf{x} = \mathbf{T}\mathbf{y}$ を座標変換とする。
   1. ∇h と $\nabla^2 h$ を f の導関数を用いて表せ。さらに、ニュートン法の反復 Equation 35.18 を h に対して $\mathbf{y}_0 = \mathbf{T}^{-1}\mathbf{x}_0$ から開始すると、$\mathbf{y}_k = \mathbf{T}^{-1}\mathbf{x}_k$（ここで $\mathbf{x}_k$ は f に対するニュートン法の反復点である）を満たすことを示せ。
   2. ステップサイズ η の勾配降下法を h に対して実行し、すべての反復点を $\mathbf{x}_k = \mathbf{T}\mathbf{y}_k$ によって元の座標へ写す。写された反復点が $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\,\mathbf{P}^{-1}\nabla f(\mathbf{x}_k)$（ただし $\mathbf{P}$ はある正定値行列）に従うことを示し、$\mathbf{P}$ を求めよ。問題 1b により、この方向はノルム $\|\cdot\|_{\mathbf{P}}$ における最急降下方向である。$\|\mathbf{x}\|_{\mathbf{P}}$ は座標 $\mathbf{y}$ ではどのように表されるか。
   3. $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\,\mathbf{x}$（$\mathbf{A} = \mathrm{diag}(1, 1000)$、ニュートン法の項で扱った $\kappa = 1000$ の二次関数）に対し、対角行列 $\mathbf{T}$ として、h 上で $\eta = 1$ の勾配降下法が 1 ステップで最小点

に到達するようなものを求めよ。そのような $\mathbf{T}$ を一般の正定値行列 $\mathbf{A}$ に対して構成するには何が必要か。

2. [code] **二次関数上の BFGS。** ●$f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^{\top}\mathbf{A}\mathbf{x} - \mathbf{b}^{\top}\mathbf{x}$（$\mathbf{A}$ は正定値）とし、BFGS Equation 35.21 を厳密な直線探索のもとで $\mathbf{M}_0 = \mathbf{I}$ から実行する。
   1. $\mathbf{p}_k$ に沿った厳密な直線探索によって $\nabla f(\mathbf{x}_{k+1})^{\top}\mathbf{p}_k = 0$ となり、さらに $\mathbf{y}_k = \mathbf{A}\mathbf{s}_k$ が成り立つことを示せ。
   2. 遺伝性を示せ。すなわち、$\mathbf{M}_k\mathbf{y}_j = \mathbf{s}_j$ がすべての $j < k$ について成り立ち、かつステップ $\mathbf{s}_0, \dots, \mathbf{s}_{k-1}$、$\mathbf{s}_k$ が互いに $\mathbf{A}$ 共役であるならば、$\mathbf{M}_{k+1}\mathbf{y}_j = \mathbf{s}_j$ がすべての $j \le k$ について成り立つことを示せ。（ヒント：$\mathbf{s}_j^{\top}\mathbf{y}_k = \mathbf{s}_j^{\top}\mathbf{A}\mathbf{s}_k = 0$ が $j < k$ に対して成り立つ。）以上から、線形独立な d 回のステップの後に $\mathbf{M}_d\mathbf{A} = \mathbf{I}$ となることを結論せよ。
   3. 本文の BFGS ループを $d = 6$ 次元で、固有値が 1 から 1000 にわたるランダムに回転させた二次関数に対して実行せよ。各ステップの後に $\max_{j \le k} \|\mathbf{M}_{k+1}\mathbf{y}_j - \mathbf{s}_j\|$ を出力し、これが丸め誤差の水準にとどまり、かつ $\mathbf{M}_6 = \mathbf{A}^{-1}$ となることを確認せよ。同じ 6 回の勾配評価の後、最適な固定ステップサイズを用いた勾配降下法は最小点からどれだけ離れているか。
   4. 対称ランク 1 更新 $\mathbf{M}_{k+1} = \mathbf{M}_k + \frac{(\mathbf{s} - \mathbf{M}_k\mathbf{y})(\mathbf{s} - \mathbf{M}_k\mathbf{y})^{\top}}{(\mathbf{s} - \mathbf{M}_k\mathbf{y})^{\top}\mathbf{y}}$ もまた割線方程式を満たす。2×2 の具体例によって、$\mathbf{y}^{\top}\mathbf{s} > 0$ であっても正定値性が保たれるとは限らないことを示し、本文で述べた 2 つの確認事項のうちどちらが成り立たなくなるかを述べよ。
3. [code] **鞍点における信頼領域。** ●モデル $m(\mathbf{p}) = \mathbf{g}^{\top}\mathbf{p} + \tfrac{1}{2}\mathbf{p}^{\top}\mathbf{B}\mathbf{p}$ において、$\mathbf{B} = \mathrm{diag}(1, -1)$、$\mathbf{g} = (\tfrac{1}{2}, -\tfrac{3}{10})$ とする。
   1. m が $\mathbb{R}^2$ 上で下に非有界であること、およびニュートン点 $-\mathbf{B}^{-1}\mathbf{g} = (-\tfrac{1}{2}, -\tfrac{3}{10})$ が最小点ではなく m の鞍点であることを示せ。
   2. 部分問題の命題の条件が十分条件であることを証明せよ。すなわち、$(\mathbf{B} + \tau\mathbf{I})\mathbf{p} = -\mathbf{g}$ が $\tau \ge 0$、$\mathbf{B} + \tau\mathbf{I} \succeq \mathbf{0}$、$\tau(\Delta - \|\mathbf{p}\|) = 0$ とともに成り立つならば、任意の $\|\mathbf{q}\| \le \Delta$ が $m(\mathbf{q}) \ge m(\mathbf{p}) + \frac{\tau}{2}(\|\mathbf{p}\|^2 - \|\mathbf{q}\|^2) \ge m(\mathbf{p})$ を満たすことを証明せよ。（ヒント：$m(\mathbf{q}) - m(\mathbf{p})$ を $\mathbf{p}$ の周りで展開し、$\mathbf{g} = -(\mathbf{B} + \tau\mathbf{I})\mathbf{p}$ を用いる。）
   3. $\Delta = \tfrac{1}{2}$ に対し、$\tau \ge 1$ を $\|(\mathbf{B} + \tau\mathbf{I})^{-1}\mathbf{g}\| = \Delta$ に関する二分法によって求め（答えは $\tau \approx 1.65$、$\mathbf{p} \approx (-0.19, 0.46)$ となる）、このステップが負の曲率をもつ座標を増加させるのに対し、ニュートンステップはそれを減少させることを確認せよ。
   4. $\mathbf{g}$ を $(\tfrac{1}{2}, 0)$ に置き換えよ。$\|(\mathbf{B} + \tau\mathbf{I})^{-1}\mathbf{g}\| = \tfrac{1}{2}/(1 + \tau) \le \tfrac{1}{4}$ がすべての $\tau > 1$ について成り立ち、したがって境界の方程式は $\tau > 1$ の範囲に解をもたないことを示せ。球上の最小点を直接求め（それらは $\tau = 1$ および $p_2 = \pm\sqrt{\Delta^2 - p_1^2}$ を満たす）、勾配からは見えない負の曲率方向に対して球が何をするかを 1 文で説明せよ。

35.3 凸性

訓練が終了したとき、停止した点について通常わかっていることは 1 つしかない。そこではもはや損失が減少しないということである。安全なステップサイズを用いた勾配降下法は、勾配がほぼゼロになるそのような点に確実に到達し（35.1 節）、ニュートン法はさらに速くそこへ到達する。たとえその停留点が結果的に鞍点であったとしてもである（35.2 節）。しかし、どちらの保証からも、はるかに良い点が別の場所に隠れているかどうかはわからない。本節では、停留点が実際に利用可能な最良の点となるのはいつか、そして勾配降下法がそこへどれほど速く収束するかを問う。

単純な 1 次元の実験から、これが決して空論ではないことがわかる。500 個のランダムな初期点からまったく同じ勾配降下法のループを、まずボウル $f(x) = (x - \tfrac{1}{2})^2$ に対して、次に傾いた二重井戸関数 $\phi(x) = (x^2 - 1)^2 + x/2$ に対して実行するとしよう。ボウルでは、すべての試行が同じ最小点に到達する。一方、二重井戸関数では、ほぼ半数の試行が浅い方の谷に捕らわれ、そ

の損失は大域的最小値よりほぼ 1 だけ高い値にとどまる。立ち往生したこれらの試行でも勾配はゼロになっており（停留性の保証が約束するのはそれだけである）、勾配だけを見ても近くにより深い谷が存在する手がかりはまったく得られない。

これら 2 つの関数の違いは、ただ 1 つの幾何学的性質に帰着する。ボウル上の 2 点間に引いた弦はすべてグラフの上側を通るが、二重井戸関数には曲線を下側へ横切る弦が存在する。弦が常にグラフ上またはその上側にある関数は**凸**（convex）と呼ばれ、そのような関数に対しては冒頭の問いにこれ以上ないほど明快な答えが与えられる。すなわち、すべての極小点は大域的最小点である。その証明はわずか 3 行で済む（35.3.2.2 節）。さらにその背後には、いっそう有用な事実がある。凸関数の任意の接線は定義域全体にわたってグラフの下側にとどまるため、ただ 1 点における局所的な情報がいたるところで関数を制約するのである。この接線による下界を勾配降下法の反復点に沿って足し合わせれば、任意の初期点から成り立つ収束率が得られる。また、確率変数について平均をとれば、確率論および情報理論の章で中心的な役割を果たすイェンセンの不等式（Jensen's inequality）が得られる。

本節ではかなり広い範囲を扱うが、すべての結果が同じ重みをもつわけではない。特に常に念頭に置いておきたい考え方は 3 つある。接線による下界、極小点が大域的最小点に一致するという事実、そして勾配降下法の線形収束率（その証明は、凸性が成り立たなくても生き残るただ 1 つの不等式に依拠している）である。まず凸集合と凸関数に対する 3 つの同値な判定条件から始め、それらの主要な帰結を導き、凸損失を見分けるための簡潔な演算規則をまとめる。深層ネットワークの損失は凸ではないため、最後にこの理論のどの部分が深層ネットワークにも引き継がれるのかを考察して締めくくる。急いでいる読者は、35.3.2 節の弦条件と接線条件だけを読み、そのまま 35.3.3 節へ進んでよい。本節を通して 2 つの例を繰り返し用いる。収束率を測るための本章の通し例である二次関数 $\frac{1}{2}(x_1^2 + 1000\,x_2^2)$ と、あらゆる凸性の判定条件をことごとく破る二重井戸関数 ϕ である。

35.3.1 凸集合

凸関数は定義域の点を結ぶ線分に沿って定義されるため、まずはそれらの点が属する集合から始めよう。損失の定義域、制約付き問題の実行可能領域（feasible region）、そしてすべての最小点からなる集合は、いずれも集合である。幸い、必要な道具立てはごく少ない。1 つの定義、3 つの代表的な例の族、そして既存の凸集合から新たな凸集合を構成する 1 つの規則だけである。

35.3.1.1 定義と例

幾何学的には、任意の 2 点を結ぶまっすぐな線分が完全にその集合の内部にとどまるとき、その集合は凸である。形式的には、集合 $C \subseteq \mathbb{R}^d$（d 次元の点からなる集合）が**凸**であるとは、次の条件を満たすことをいう。

$$\theta \mathbf{x} + (1-\theta)\,\mathbf{y} \in C \qquad \text{for all } \mathbf{x}, \mathbf{y} \in C \text{ and all } \theta \in [0,1]. \tag{35.24}$$

θ が 1 から 0 へ動くにつれて、結合 $\theta\mathbf{x} + (1-\theta)\mathbf{y}$ は $\mathbf{x}$ から $\mathbf{y}$ までの線分を描く。Figure 35.8 では、凸集合と非凸な三日月形領域を対比し、さらに深層学習で最も頻繁に現れる 2 つの凸集合である確率単体（probability simplex）と半空間（half-space）を示している。

この定義によって何が除外されるかを見るために、円環領域 $\{\mathbf{x} : 1 \le \|\mathbf{x}\| \le 2\}$ を考えよう。この集合は $(\frac{3}{2}, 0)$ と $(-\frac{3}{2}, 0)$ の両方を含むが、その中点である原点は中央の穴の中にあり集合に含まれない。中点がたった 1 つ欠けているだけで、Equation 35.24 は破られる。

本章で出会う凸集合のほとんどは、3 つの族で網羅される。第 1 に、任意の超平面（hyperplane）$\{\mathbf{x} : \mathbf{a}^\top \mathbf{x} = b\}$ と任意の半空間 $\{\mathbf{x} : \mathbf{a}^\top \mathbf{x} \le b\}$ は、$\mathbf{a}^\top \mathbf{x}$ が線形であるため凸である。実際、b に等しい（あるいは高々 b である）2 つの数の重み付き平均は、再び b に等しい（あ

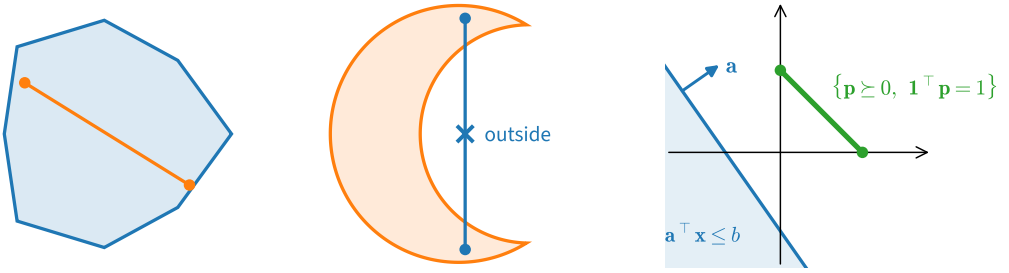


Figure 35.8: 凸集合と非凸集合の比較。左：任意の 2 点間の線分が内部にとどまる凸集合。中央：2 点間の弦が集合の外部を通る非凸な三日月形領域。右：確率単体 $\{\mathbf{p} \succeq 0,\ \mathbf{1}^\top \mathbf{p} = 1\}$ と半空間 $\{\mathbf{x} : \mathbf{a}^\top \mathbf{x} \le b\}$（いずれも凸集合）。

るいは高々 b である）。第 2 に、任意のノルム球 $\{\mathbf{x} : \|\mathbf{x}\| \le r\}$ は、三角不等式と斉次性 $\|\theta \mathbf{x} + (1-\theta)\mathbf{y}\| \le \theta\|\mathbf{x}\| + (1-\theta)\|\mathbf{y}\| \le r$ により凸である。このただ 1 つの確認だけで、重み減衰や勾配クリッピングに現れる ℓ_2 球も、スパース性を誘導する正則化に現れる ℓ_1 球もカバーされる。第 3 は確率単体 $\Delta = \{\mathbf{p} \in \mathbb{R}^d : \mathbf{p} \succeq 0,\ \mathbf{1}^\top \mathbf{p} = 1\}$ であり、ここで $\mathbf{p} \succeq 0$ はすべての成分が非負であることを意味する。アテンション重み、クラス確率、混合重みはいずれも Δ 上に存在し、その凸性は次に述べる規則からただちに従う。

35.3.1.2 既存の凸集合からの構成

有限個か無限個かを問わず、任意の凸集合の族の共通部分は常に凸集合である。2 点が共通部分に属するならば、その族のすべての集合に属するため、それらを結ぶ線分もすべての集合の内部にとどまり、したがって共通部分の内部にとどまる。直感的には、凸集合にはくぼみがなく、くぼみのない集合を別のくぼみのない集合で切り取ってもくぼみが生じることはない。対照的に、合併集合にはそのような保証はない。$[0,1] \cup [2,3]$ は 2 つの凸区間の合併であるが、1 から 2 までの線分は集合の外へ出てしまう。

この共通部分の規則 1 つで、中心的な 2 つの集合の凸性が一度に示される。確率単体は 1 つの超平面（$\mathbf{1}^\top \mathbf{p} = 1$）と d 個の半空間（$-p_i \le 0$）の共通部分であるため凸である。同様に、対称半正定値（positive semidefinite、PSD）行列の集合も、無限個の半空間の共通部分であるため凸である。実際、固定した各ベクトル $\mathbf{z}$ に対して、制約 $\mathbf{z}^\top \mathbf{A} \mathbf{z} \ge 0$ は $\mathbf{A}$ の成分に関する 1 つの線形不等式にほかならない。

内側から外側へ向かって凸集合を構成することもできる。集合 S の**凸包**（convex hull）とは、重み付き平均 $\sum_i \theta_i \mathbf{x}_i$（ただし $\mathbf{x}_i \in S$、$\theta_i \ge 0$、$\sum_i \theta_i = 1$）全体の集合である。凸包は凸集合であり、実際、S を含む最小の凸集合となる（演習問題 2）。この見方に立てば、確率単体とは標準基底ベクトル $\mathbf{e}_1, \ldots, \mathbf{e}_d$ の凸包にほかならない。すなわち、あらゆる確率分布は確定的な状態の重み付き平均である。

35.3.2 凸関数

凸集合にくぼみがないのと同様に、凸関数とは、グラフが上向きに湾曲して第 2 の谷へ落ち込むことのない関数である。以下では、それぞれ 0 階・1 階・2 階の微分を用いる 3 つの同値な凸性の判定条件を導き、続いて 2 つの発展的な概念を扱う。勾配降下法の収束の速さを支配する強凸性（strong convexity）と、角をもつ損失を扱うための劣勾配（subgradient）である。3 つの判定条

件のうち 1 つだけを覚えるなら、2 番目の条件にしてほしい。接線による下界こそ、本節のほぼすべての証明を動かす原動力だからである。

35.3.2.1 弦条件

第 1 の判定条件は微分を一切必要としないため、滑らかな目的関数だけでなく絶対値関数やヒンジ損失にも同じように容易に適用できる。関数 $f : C \to \mathbb{R}$（C は凸な定義域）が**凸**であるとは、グラフが任意の 2 点を結ぶ弦の上または下側にあることをいう。

$$f\left(\theta \mathbf{x} + (1-\theta)\,\mathbf{y}\right) \;\le\; \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \qquad \text{for all } \mathbf{x}, \mathbf{y} \in C,\ \theta \in [0,1]. \tag{35.25}$$

ここで左辺は、f を $\mathbf{x}$ と $\mathbf{y}$ を結ぶ線分上の点で評価した値であり、右辺はその真上にある、$(\mathbf{x}, f(\mathbf{x}))$ と $(\mathbf{y}, f(\mathbf{y}))$ を結ぶまっすぐな弦の高さを与える。具体例で確認してみよう。$f(x) = x^2$ において $x = 0$、$y = 2$、$\theta = \tfrac{1}{2}$ とすると、グラフの高さは $f(1) = 1$ であるのに対し、弦の高さは $\tfrac{1}{2}\cdot 0 + \tfrac{1}{2}\cdot 4 = 2$ であり、確かに $1 \le 2$ が成り立つ。任意の x、y、θ に対して二乗を展開すると、弦とグラフの高さの差は次のようになる。

$$\theta x^2 + (1-\theta) y^2 - \left(\theta x + (1-\theta) y\right)^2 = \theta(1-\theta)(x-y)^2 \;\ge\; 0.$$

したがって x^2 は凸であり、その差は 2 点間の距離の二乗に比例して大きくなる。このまったく同じ式に、まもなくイェンセンの不等式における分散として再び出会うことになる。

関連する 2 つの概念にもここで名前を付けておこう。不等式 Equation 35.25 が $\mathbf{x} \neq \mathbf{y}$ かつ $\theta \in (0,1)$ のとき常に狭義の不等号で成り立つならば、f を**狭義凸**（strictly convex）と呼ぶ。また、$-f$ が凸であるならば、f を**凹**（concave）と呼ぶ。線形関数 $f(x) = x$ は凸（かつ凹）であるが、すべての弦がグラフそのものに重なるため狭義凸ではないことに注意しよう。また、なぜ定義域 C が凸でなければならないかも確認しておきたい。その要請がなければ、中間の点 $\theta\mathbf{x} + (1-\theta)\mathbf{y}$ がそもそも f の定義される領域に属さない可能性があるからである。

凸関数と凸集合は、同じ幾何学的構造の表裏をなしている。関数 f の**エピグラフ**とは、そのグラフ上および上側の領域 $\{(\mathbf{x}, t) : t \ge f(\mathbf{x})\}$ のことである。エピグラフ内の 2 点を結ぶ弦がグラフの上側にとどまるのはまさに Equation 35.25 が成り立つときであるため、関数 f が凸であることと、そのエピグラフが凸集合であることは同値である。また凸性は、**劣レベル集合**（sublevel set）$\{\mathbf{x} \in C : f(\mathbf{x}) \le c\}$ を通じても関数から集合へと受け継がれる。実際、$f(\mathbf{x}) \le c$ かつ $f(\mathbf{y}) \le c$ であれば、弦不等式からただちに $f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \le \theta c + (1-\theta)c = c$ が得られる。任意の不等式制約 $g(\mathbf{x}) \le 0$（ただし g は凸関数）が自動的に凸な実行可能領域を切り出すのはこのためである。Figure 35.9(a) に放物線に対するこれら両方の集合を示す。

ただし注意してほしいのは、この逆は成り立たないという点である。$f(x) = \sqrt{|x|}$ の劣レベル集合はすべて凸区間 $\{|x| \le c^2\}$ となるが、f 自体は凸ではない。$x = 0$、$y = 1$、$\theta = \tfrac{1}{2}$ において、グラフは $\sqrt{1/2} \approx 0.71$ にあるのに対し、弦はその下側の 0.5 を通る（Figure 35.9(b)）。

35.3.2.2 極小点は大域的最小点

驚くべきことに、弦条件だけでも冒頭で立てた問いに答えるには十分である。Figure 35.10 は 2 つのランドスケープを対比している。凸目的関数における単一の吸引領域と、二重井戸関数における競合する 2 つの吸引領域である。

命題（極小点は大域的最小点）。 **f が凸集合 C 上で凸であるならば、f の C 上における極小点はすべて大域的最小点である。**

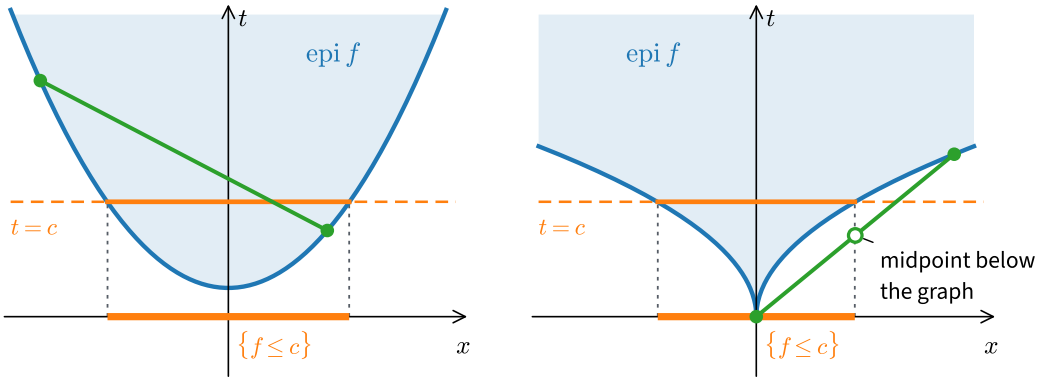


Figure 35.9: エピグラフと劣レベル集合。(a) 凸関数 $f(x) = \frac{1}{2}x^2 + \frac{1}{4}$ のエピグラフ（網掛け部分、上方へ限りなく続く）は凸集合であり、その任意の 2 点間の弦は内部にとどまる。エピグラフを高さ c で切り、x 軸へ射影すると劣レベル集合 $\{f \leq c\}$ が得られる。(b) $f(x) = \sqrt{|x|}$ の劣レベル集合はすべて区間となるが、エピグラフは凸ではない。弦の中点がグラフの下側に出てしまう。

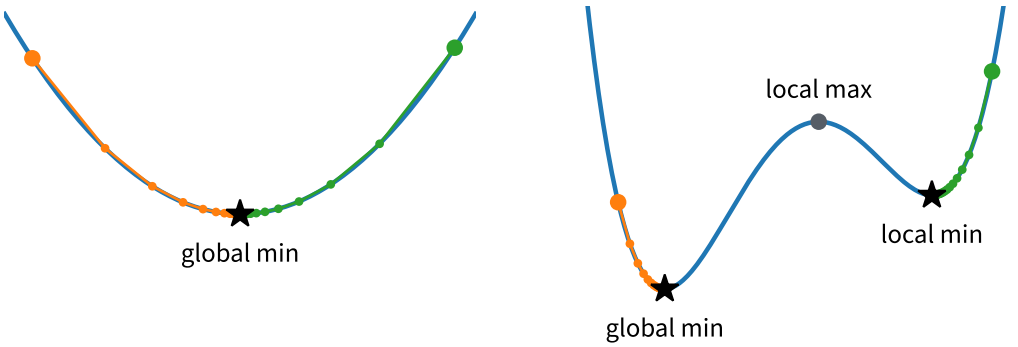


Figure 35.10: (a) 凸目的関数では、小さな固定ステップサイズの勾配降下法がどちらの初期点からも大域的最小点に到達する。(b) 二重井戸関数では、異なる初期点から同じ更新を行うと異なる極小点に到達する。2 つの吸引領域は極大点で接している。

証明。 $\mathbf{x}^\star$ を極小点とし、$\mathbf{y} \in C$ を任意の別の点とする。十分小さい $\theta > 0$ に対して、点 $\mathbf{z}_\theta = (1-\theta)\mathbf{x}^\star + \theta\mathbf{y}$（$\mathbf{y}$ へ向かう線分上の点）は $\mathbf{x}^\star$ が最小となる近傍内に収まる。局所最小性と弦不等式を組み合わせると次を得る。

$$f(\mathbf{x}^\star) \;\le\; f(\mathbf{z}_\theta) \;\le\; (1-\theta) f(\mathbf{x}^\star) + \theta f(\mathbf{y}).$$

両辺から $(1-\theta)f(\mathbf{x}^\star)$ を引き、$\theta > 0$ で割れば、$f(\mathbf{x}^\star) \le f(\mathbf{y})$ が得られる。■

幾何学的な描像も同様に明快である。もし真に小さな値を与える点 $\mathbf{y}$ が C 内のどこかに存在したならば、$\mathbf{x}^\star$ から $\mathbf{y}$ へ向かう弦はただちに下向きに傾き、グラフを $f(\mathbf{x}^\star)$ より下へ、しかも $\mathbf{x}^\star$ のいくらでも近くで押し下げることになり、局所最小性に矛盾する。

ここで冒頭の実験を実際に動かし、この命題が数値的にどう現れるかを見てみよう。同じ勾配降下法のループ（ステップサイズ 0.05、600 ステップ）をボウル $(x-\tfrac{1}{2})^2$ と二重井戸関数 ϕ の両方に適用し、同じ 500 個の点（$[-1.5, 1.5]$ 上の一様分布から抽出）から開始する。ステップサイズが小さいため、各反復点は中央の山を飛び越えることなく坂を転がり落ちる。したがって ϕ 上の試行は、極大点を境にして初期点が属する側の井戸に落ち着くはずである。その極大点を $\phi'(x) = 4x^3 - 4x + \tfrac{1}{2}$ の根として計算し、それより右側から始まる初期点の割合が、浅い右側の井戸に収束する試行の割合と一致するかを確かめる。

```
def gd(grad_fn, x, eta=0.05, steps=600):
    for _ in range(steps):
        x = x - eta * grad_fn(x)
    return x

phi = lambda x: (x**2 - 1)**2 + 0.5 * x        # tilted double well
dphi = lambda x: 4 * x * (x**2 - 1) + 0.5      # its derivative
rng = np.random.default_rng()
inits = rng.uniform(-1.5, 1.5, size=500)       # 500 random starting points
bowl = gd(lambda x: 2 * (x - 0.5), inits)      # convex: (x - 1/2)^2
well = gd(dphi, inits)
left, right = well[well < 0], well[well > 0]
print(f'bowl: all {bowl.size} runs end at x = {bowl.mean():.6f}'
      f' (spread {np.ptp(bowl):.0e})')
for name, w in (('left ', left), ('right', right)):
    print(f'well, {name}: {w.size} runs, x = {w.mean():.4f}, phi = {phi(w.mean()):.4f}')
x_max = np.sort(np.roots([4, 0, -4, 0.5]).real)[1]   # local max of phi
print(f'local max at {x_max:.4f}: predicted share right {(1.5 - x_max) / 3:.3f},'
      f' observed {right.size / inits.size:.3f}')
```

```
bowl: all 500 runs end at x = 0.500000 (spread 7e-16)
well, left : 263 runs, x = -1.0575, phi = -0.5148
well, right: 237 runs, x = 0.9304, phi = 0.4833
local max at 0.1271: predicted share right 0.458, observed 0.474
```

ボウルでは、500 回の試行すべてが数値丸めの範囲内で $x = 0.5$ に収束する。一方、二重井戸関数では試行が 2 つの陣営にくっきりと分かれる。一方は $x \approx -1.0575$（$\phi \approx -0.515$）、もう一方は $x \approx 0.9304$（$\phi \approx 0.483$）であり、後者はほぼ 1 だけ悪い。2 つの吸引領域を隔てる極大点は $x \approx 0.127$ にあるため、一様な初期点のうち $(1.5 - 0.127)/3 \approx 0.458$ の割合がその右側に落ちる。ランダムな抽出を繰り返しても、観測される割合は 46% の前後数パーセントポイント以内に収まり、500 個の初期点によるサンプリング誤差の範囲で予測と一致する。

両方の実験で動かしたのはまったく同じオプティマイザであり、変えたのは目的関数だけである。凸性は第 2 の吸引領域を消し去るが、それ自体が勾配降下法による最小点への到達を保証するわけではない。到達のためには、35.1.1 節で見たように、滑らかさと安全なステップサイズが依然として必要である。

35.3.2.3 一階条件

弦条件には微分を必要としないという利点があるが、最適化の証明は勾配を材料として組み立てられる。f が微分可能であるとき、凸性は同値な一階の形式をとる。すなわち、任意の点における接超平面が関数の**大域的な下からの評価式**（global under-estimator）となる。

$$f(\mathbf{y}) \;\geq\; f(\mathbf{x}) + \nabla f(\mathbf{x})^{\top}(\mathbf{y}-\mathbf{x}) \qquad \text{for all } \mathbf{x},\mathbf{y} \in C. \tag{35.26}$$

これを $f(x) = x^2$ の $x = 1$ において確かめてみよう。接線は $1 + 2(y-1) = 2y - 1$ であり、グラフと接線の縦方向の差は $y^2 - (2y-1) = (y-1)^2 \geq 0$ となってすべての y に対して成り立つ。Equation 35.26 がどれほど強い主張であるか、立ち止まって味わってほしい。ただ 1 つの点 $\mathbf{x}$ で計算したただ 1 つの勾配が、定義域内の他のすべての点において f を下から抑えるのである。本節の残りのほぼすべての証明が、この 1 本の不等式の上に築かれる。Figure 35.11 は最初の 2 つの判定条件を並べて示している。弦はグラフの上側にあり、接線はグラフの下側にある。

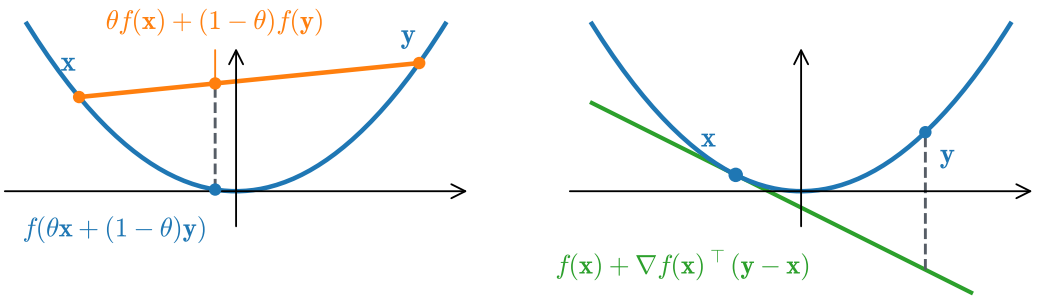


Figure 35.11: 2 つの同値な条件。(a) グラフの 2 点を結ぶ弦はグラフの上側にある：$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y})$。(b) 任意の点における接線はグラフの下側にある：$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^{\top}(\mathbf{y}-\mathbf{x})$。

35.3.2.4 二階条件

滑らかな目的関数に対して凸性を確かめる最も手早い方法は、多くの場合、行列による判定である。f が**開**凸定義域 C 上で 2 回微分可能であるとき、f が凸であることと、そのヘッセ行列がいたるところで半正定値であることは同値である。

$$\nabla^2 f(\mathbf{x}) \succeq 0 \qquad \text{for all } \mathbf{x} \in C.$$

1 次元では、これはおなじみの微分積分の規則 $f''(x) \geq 0$、すなわち曲線が上向きに曲がるという条件に帰着する。d 次元では、$\mathbf{x}$ を通るすべての 1 次元スライスが上向きに曲がることを意味する。実際、$\mathbf{v}^{\top}\nabla^2 f(\mathbf{x})\,\mathbf{v}$ は f の方向 $\mathbf{v}$ の直線に沿った二階導関数だからである（34.2 節）。

この行列による判定から、定番の 3 つの例がただちに片づく。第 1 に、二次関数 $\frac{1}{2}\mathbf{x}^{\top}\mathbf{A}\mathbf{x} + \mathbf{b}^{\top}\mathbf{x}$（$\mathbf{A}$ は対称行列）は定数のヘッセ行列 $\mathbf{A}$ をもつため、凸であることと $\mathbf{A} \succeq 0$ であることは同値で

ある。第 2 に、最小二乗損失 $\frac{1}{2}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|^2$（計画行列 $\mathbf{X}\in\mathbb{R}^{n\times d}$、$n$ 個の例、d 個の重み）のヘッセ行列は $\mathbf{X}^\top\mathbf{X}$ であり、$\mathbf{v}^\top\mathbf{X}^\top\mathbf{X}\mathbf{v}=\|\mathbf{X}\mathbf{v}\|^2\geq 0$ がすべての $\mathbf{v}$ について成り立つため、最小二乗法は常に凸である（33.3.2.2 節）。第 3 に、スカラー関数 e^x、$-\log x$、$x\log x$ は、それぞれの二階導関数 e^x、$1/x^2$、$1/x$ がいたるところで正であるため、いずれもその定義域上で凸である。

なぜ定義域 C が開集合であることにこだわったのだろうか。低次元の定義域上では、C の内部にとどまる方向だけが問題になるからである。たとえば x^2-y^2 は、2 次元全体のヘッセ行列が不定値であるにもかかわらず、直線 $y=0$ に制限すれば凸関数になる。

ここで、3 つの判定条件がまったく同じ関数クラスを特徴づけていることを確かめよう。証明は 4 つの短いパートからなるが、記憶に留める価値があるのは 2 番目のパートである。弦の**中間**点において接線による下界を適用すると勾配の項が相殺されるという手法であり、これとまったく同じ技巧をイェンセンの不等式の証明でも用いる。

命題（3 つの特徴づけの一致）。 **f を開凸定義域 C 上で微分可能な関数とする。このとき、弦条件 Equation 35.25 と一階条件 Equation 35.26 は同値である。さらに f が 2 回微分可能であれば、両者は $\nabla^2 f(\mathbf{x})\succeq 0$ が C 上で成り立つこととも同値である。**

証明。 **（弦条件 $\Rightarrow$ 一階条件。）** $\mathbf{x},\mathbf{y}\in C$ と $\theta\in(0,1]$ を固定する。中間点 $\mathbf{x}+\theta(\mathbf{y}-\mathbf{x})$ において弦不等式を適用し、両辺から $f(\mathbf{x})$ を引いて θ で割ると次を得る。

$$\frac{f(\mathbf{x}+\theta(\mathbf{y}-\mathbf{x}))-f(\mathbf{x})}{\theta}\;\leq\; f(\mathbf{y})-f(\mathbf{x}).$$

$\theta\downarrow 0$ とすると、左辺は方向微分 $\nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x})$ に収束し、Equation 35.26 が得られる。

（一階条件 $\Rightarrow$ 弦条件。） $\mathbf{z}=\theta\mathbf{x}+(1-\theta)\mathbf{y}$ とおき、点 $\mathbf{z}$ における接線の下界を各端点に向かって 1 回ずつ、計 2 回適用する。

$$f(\mathbf{x})\geq f(\mathbf{z})+\nabla f(\mathbf{z})^\top(\mathbf{x}-\mathbf{z}),\qquad f(\mathbf{y})\geq f(\mathbf{z})+\nabla f(\mathbf{z})^\top(\mathbf{y}-\mathbf{z}).$$

第 1 の不等式に θ を、第 2 の不等式に $1-\theta$ を掛けて足し合わせる。1 次の項はまとまって $\nabla f(\mathbf{z})^\top(\theta\mathbf{x}+(1-\theta)\mathbf{y}-\mathbf{z})=0$ となり、Equation 35.25 が残る。

（二階条件 $\Rightarrow$ 一階条件。） f を $\mathbf{x}$ と $\mathbf{y}$ を結ぶ線分に制限し、$g(t)=f(\mathbf{x}+t(\mathbf{y}-\mathbf{x}))$ と定義する。その二階導関数は $g''(t)=(\mathbf{y}-\mathbf{x})^\top\nabla^2 f(\mathbf{x}+t(\mathbf{y}-\mathbf{x}))\,(\mathbf{y}-\mathbf{x})\geq 0$ である。ラグランジュの剰余項をもつテイラーの定理（34.1 節）により、$g(1)=g(0)+g'(0)+\frac{1}{2}g''(\tau)$ が、ある $\tau\in(0,1)$ に対して成り立つ。非負の剰余項を落とせば Equation 35.26 が得られる。

（一階条件 $\Rightarrow$ 二階条件。） 背理法のため、$\mathbf{v}^\top\nabla^2 f(\mathbf{x})\,\mathbf{v}<0$ がある点 $\mathbf{x}\in C$ と方向 $\mathbf{v}$ に対して成り立つと仮定する。f を $\mathbf{v}$ に沿って展開すると次を得る。

$$f(\mathbf{x}+t\mathbf{v})=f(\mathbf{x})+t\,\nabla f(\mathbf{x})^\top\mathbf{v}+\frac{t^2}{2}\,\mathbf{v}^\top\nabla^2 f(\mathbf{x})\,\mathbf{v}+o(t^2).$$

十分小さい $t>0$ に対しては、負の 2 次の項が剰余項 $o(t^2)$ を支配するため、$f(\mathbf{x}+t\mathbf{v})$ は接線の値 $f(\mathbf{x})+t\nabla f(\mathbf{x})^\top\mathbf{v}$ を真に下回り、Equation 35.26 に矛盾する。■

35.3.2.5 3 つの判定条件の数値的検証

2 つの損失に対して、3 つの判定条件をすべて数値的に試してみよう。用いるのは、最小二乗目的関数 $f(\mathbf{w}) = \frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$（ランダムな例 $n = 20$ 個、重み $d = 2$ 個）と、冒頭の実験で用いた二重井戸関数 ϕ である。最小二乗法については、条件からの逸脱量の厳密な式がすでにわかっている。x^2 のときと同様に二乗を展開すれば、弦ギャップ（グラフ引く弦）は $-\frac{1}{2}\theta(1-\theta)\|\mathbf{X}(\mathbf{u}-\mathbf{v})\|^2$、接線ギャップ（接線引くグラフ）は $-\frac{1}{2}\|\mathbf{X}(\mathbf{v}-\mathbf{u})\|^2$ となり、いずれも非正である。以下のコードは、各損失について 1,000 組のランダムな弦と接線を $[-1.5, 1.5]^d$ 内でサンプリングし、それぞれの最大の違反量を報告するとともに、サンプル点全体にわたるヘッセ行列の最小固有値を調べる。

```
X, y = rng.normal(size=(20, 2)), rng.normal(size=20)  # least squares, d = 2
losses = {'least squares': (2, lambda w: 0.5 * ((X @ w - y) ** 2).sum(),
                            lambda w: X.T @ (X @ w - y), lambda w: X.T @ X),
          'double well': (1, lambda w: phi(w[0]), dphi,  # phi'' = 12 x^2 - 4
                          lambda w: np.diag(12 * w**2 - 4))}
print('               worst chord gap  worst tangent gap  smallest Hessian eig')
for name, (d, f, grad, hess) in losses.items():
    U, V = rng.uniform(-1.5, 1.5, size=(2, 1000, d))  # 1000 random pairs
    T = rng.uniform(size=1000)
    chord = [f(t * u + (1 - t) * v) - (t * f(u) + (1 - t) * f(v))
             for u, v, t in zip(U, V, T)]             # graph minus chord
    tangent = [f(u) + grad(u) @ (v - u) - f(v)        # tangent minus graph
               for u, v in zip(U, V)]
    eig = min(np.linalg.eigvalsh(hess(u))[0] for u in U)
    print(f'{name:15s}{max(chord):15.3f}{max(tangent):19.3f}{eig:22.3f}')
i = int(np.argmax(chord))                             # the double well's worst
print(f'worst chord of the double well: from {U[i, 0]:.2f} to {V[i, 0]:.2f}')
```

```
               worst chord gap  worst tangent gap  smallest Hessian eig
least squares           -0.000             -0.003                24.375
double well              0.972              2.997                -4.000
worst chord of the double well: from -0.96 to 1.10
```

最小二乗法では、式の上での予測どおり、サンプリングされた弦がグラフの下に沈むことは 1 度もなく、接線がグラフの上に出ることもなく、ヘッセ行列の固有値は正にとどまる。一方、二重井戸関数は 3 つの検査すべてに不合格となる。最悪の弦は 2 つの谷の点を結んでグラフよりほぼ 1 だけ下を通り、接線は曲線を貫き、二階導関数は中央の山の近くで -4 付近まで落ち込む（そこでは $\phi''(0) = -4$ である）。ただし、これらの検査の間にある非対称性に注意しよう。ランダムサンプリングは凸性を**反証する**ことはできるが（ϕ にとっては不適切な弦が 1 本あれば完全な反例となる）、凸性を**証明する**ことは決してできない。最小二乗法に対しては、違反を見つけられなかったというだけである。最小二乗法の凸性を証明しているのはヘッセ行列による判定であり、それは $\mathbf{X}^\top\mathbf{X}$ がすべての $\mathbf{w}$ において同一の PSD 行列だからである。

35.3.2.6 狭義凸性と強凸性

単なる凸性はいくらでも平坦な区間を許容するため、そのような平坦な場所では勾配降下法がいくらでも遅く這うように進むことがあり得る。速い収束を保証するには、いたるところでゼロから離れて下から抑えられた曲率が必要になる。Figure 35.12(a) に示す 3 つの単純な関数がこの階層構造をよく表している。直線 $f(x) = x$ は凸であるが、弦がグラフ上にぴったり重なるため狭義凸ではない。4 次関数 x^4 は狭義凸である（二階導関数 $12x^2$ がゼロになるのは 0 のみであり、傾きはいたるところで狭義単調に増加する）が、$f''(0) = 0$ のため最小点において完全に平坦に

なっている。これに対し、二次関数 x^2 はすべての点で一定の曲率 $f''(x) = 2$ をもつ。

強凸性は、この観察を定量的な下界へと定式化する。$\mu > 0$ に対し、関数 f が μ **強凸**であるとは、曲率 μ の二次関数を引いてもなお凸関数として残ること、すなわち $h(\mathbf{x}) = f(\mathbf{x}) - \frac{\mu}{2}\|\mathbf{x}\|^2$ が凸であることをいう。f が微分可能であるとき、h（その勾配は $\nabla f(\mathbf{x}) - \mu\mathbf{x}$）に一階条件を適用し、恒等式 $\|\mathbf{y}\|^2 - \|\mathbf{x}\|^2 - 2\mathbf{x}^\top(\mathbf{y} - \mathbf{x}) = \|\mathbf{y} - \mathbf{x}\|^2$ を用いると、同値な不等式が得られる。

$$f(\mathbf{y}) \;\geq\; f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\,\|\mathbf{y} - \mathbf{x}\|^2 \qquad \text{for all } \mathbf{x}, \mathbf{y}. \tag{35.27}$$

幾何学的には、f のグラフは点 $\mathbf{x}$ における接線の上側にあるだけでなく、そこで接する曲率 μ の放物線よりも上側になければならない。h に二階条件を適用すれば行列による判定条件が得られる。すなわち、2 回微分可能な f が μ 強凸であることと、いたるところで $\nabla^2 f(\mathbf{x}) \succeq \mu\mathbf{I}$（ただし $\mathbf{I}$ は単位行列）が成り立つことは同値である。したがって x^2 は 2 強凸である一方、x^4 は 0 のいかなる近傍においても強凸にならない。$f''(0) = 0 < \mu$ がすべての $\mu > 0$ に対して成り立ってしまうからである。実用上のうれしい副産物にも注目しよう。任意の凸損失にリッジペナルティ $\frac{\lambda}{2}\|\mathbf{w}\|^2$ を加えると、その和はただちに λ 強凸になる。$\frac{\lambda}{2}\|\mathbf{w}\|^2$ を引けば元の凸損失に戻るからである。重み減衰は単にパラメータを縮小するだけでなく、強凸性を保証するのである。

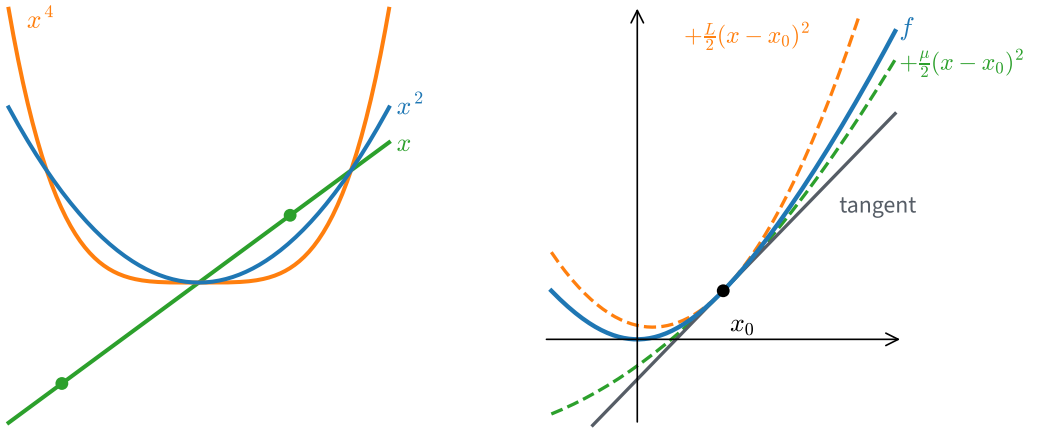


Figure 35.12: (a) 凸・狭義凸・強凸：x は弦がグラフ上に乗り、x^4 は原点で平坦になり、x^2 はいたるところで湾曲する。(b) μ 強凸かつ L 滑らかな関数（ここでは $f(x) = \frac{1}{2}x^2 + 3\log\cosh x$）は、$x_0 = 1$ で接する曲率 $\mu = 1$ と $L = 4$ の 2 つの放物線の間に挟まれる。両者の比が κ である。

強凸性は、滑らかさと自然な対をなしている。35.1.1 節で見たように、f が L **滑らか**であるとは、その勾配が L リプシッツ連続であること、すなわち $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ を満たすことであった。2 回微分可能な凸関数においては、これは $\nabla^2 f \preceq L\mathbf{I}$ を意味する。滑らかさはグラフの上側に二次上界 $f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2$ を置き、強凸性はグラフの下側に二次下界 Equation 35.27 を置く。両者を合わせると、f は曲率 μ と L をもつ 2 つの接放物線の間に挟まれる。Figure 35.12(b) はこれを $f(x) = \frac{1}{2}x^2 + 3\log\cosh x$ について描いたものであり、その二階導関数 $f''(x) = 1 + 3/\cosh^2 x$ は $\mu = 1$ から $L = 4$ の範囲を動く。

比 $\kappa = L/\mu$ は**条件数**（condition number）であり、35.1.2 節で二次関数に対して導入した固有値比を一般化したものである。通し例の二次関数 $\frac{1}{2}(x_1^2 + 1000\,x_2^2)$ では、極端な曲率は $\mu = 1$ と $L = 1000$ であり、$\kappa = 10^3$ となる。

35.3.2.7 劣勾配

ℓ_1 ノルム $\|\mathbf{x}\|_1$ やヒンジ損失 $\max(0, 1 - z)$ といった重要な損失には鋭い角があり、そこでは勾配が存在しないため Equation 35.26 をそのまま書くことはできない。しかし接線による下界をあきらめる代わりに、勾配が果たしていた**役割**（大域的な線形の下からの評価式を与えること）を残し、その役割自体を定義へと転じる。ベクトル $\mathbf{g}$ が f の点 $\mathbf{x}$ における**劣勾配**であるとは、次の条件を満たすことをいう。

$$f(\mathbf{y}) \;\geq\; f(\mathbf{x}) + \mathbf{g}^\top(\mathbf{y} - \mathbf{x}) \qquad \text{for all } \mathbf{y}. \tag{35.28}$$

点 $\mathbf{x}$ におけるすべての劣勾配からなる集合を**劣微分**（subdifferential）と呼び、$\partial f(\mathbf{x})$ と書く。凸関数 f が微分可能な点では、劣微分はただ 1 つのベクトルからなる集合 $\{\nabla f(\mathbf{x})\}$ に潰れる。しかし角においては、グラフを下から支える傾き全体の集合へと扇状に広がる（Figure 35.13）。$f(x) = |x|$ の $x = 0$ においては、すべての傾き $g \in [-1, 1]$ が $|y| \geq gy$ を任意の y に対して満たすため $\partial f(0) = [-1, 1]$ となる。同様に、ヒンジ損失 $\max(0, 1 - z)$ の折れ点 $z = 1$ では $\partial f(1) = [-1, 0]$ が得られる。

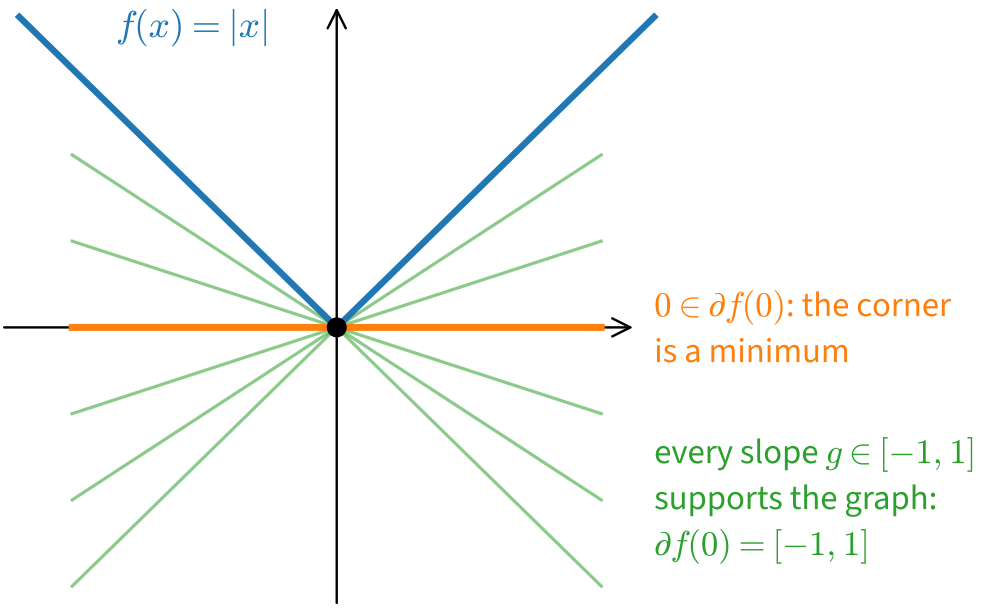


Figure 35.13: $f(x) = |x|$ の折れ点では勾配は存在しないが、劣微分は集合として定まる。すべての傾き $g \in (-1, 1)$ が V 字の下側で支持直線を定め、両端の傾き ± 1 は枝そのものに重なるため、$\partial f(0)$ は -1 から 1 までの閉区間全体となる。傾きゼロの要素（オレンジ）は最適性の証明 $0 \in \partial f(0)$ を与え、微分可能性なしにこの角が最小点であることが保証される。

ここからただちに、微分を用いない最適性の判定条件が得られる。すなわち、$\mathbf{x}^\star$ が f を最小化することと $\mathbf{0} \in \partial f(\mathbf{x}^\star)$ は同値である。証明は 1 行の同語反復にすぎない。$\mathbf{g} = \mathbf{0}$ を Equation 35.28 に代入すれば、$f(\mathbf{y}) \geq f(\mathbf{x}^\star)$ がすべての $\mathbf{y}$ について成り立つという式になり、これは文字どおり大域的最小点の定義そのものである。

凸関数は常に劣勾配をもつのだろうか。定義域のすべての内点において、答えはイエスである。この事実はイェンセンの不等式の証明で用いることになる。1 次元ではその理由は初等的である。

任意の 3 点 $a < b < c$（f が凸である開区間内の点）に対し、点 b における弦不等式を整理すると次を得る。

$$\frac{f(b)-f(a)}{b-a} \;\le\; \frac{f(c)-f(a)}{c-a} \;\le\; \frac{f(c)-f(b)}{c-b},$$

つまり、弦の傾きは左から右へ向かって非減少である。その結果、差分商 $q(x) = (f(x) - f(x_0))/(x - x_0)$ は任意の内点 x_0 において x に関して非減少となるため、その片側極限である左微分と右微分 $f'_-(x_0) \le f'_+(x_0)$ がともに有限な値として存在する。任意の傾き $g \in [f'_-(x_0), f'_+(x_0)]$ は劣勾配となる。実際、$y > x_0$ に対しては $q(y) \ge f'_+(x_0) \ge g$ であり、これを変形すれば $f(y) \ge f(x_0) + g(y - x_0)$ となる。$y < x_0$ の場合も対称である。

同じ保証は d 次元にも拡張される。凸関数はその定義域のすべての内点において少なくとも 1 つの劣勾配をもつため、常に支持超平面がグラフの下側に収まる（Rockafellar (1970)、Boyd and Vandenberghe (2004) 2.5 節）。本書ではこの結果を証明なしに用いる。ただし、探索方向としての劣勾配は真の勾配よりはるかに弱いことに注意しておこう。$-\mathbf{g}$ の方向にステップを進めても、f が減少するとは限らないのである（最小点 $x = 0$ において $|x|$ を考えると、正当な劣勾配 $g = \frac{1}{2}$ によるステップは $-\eta/2$ へ移動し、損失を増大させる）。そのため、劣勾配法には減衰するステップサイズが必要となり、収束も遅く、誤差は $1/\sqrt{k}$ のオーダーでしか減少しない。

35.3.3 凸性の帰結

本書の以降の章では凸性の 3 つの主要な帰結を活用するが、それら 3 つはすべて同じ幾何学的源泉から生まれる。すなわち、いたるところで関数を下から抑える支持直線または支持超平面である。この下界を確率変数について平均すれば、イェンセンの不等式が得られる。停留点において評価すれば、大域的最適性が証明され、35.3.2.2 節で弦条件のみから導いた結果がさらに研ぎ澄まされる。勾配降下法のステップに沿って足し合わせれば、局所的な前進が大域的な収束率へと変わる。イェンセンの不等式はそれ単独で完結しているため、最適化のみを目的として読んでいる場合はそのまま次の項へ進んでよい。

35.3.3.1 イェンセンの不等式

期待損失、平均対数尤度、KL ダイバージェンスはいずれも同じ基本的な問いに直面する。平均の関数値と、関数値の平均とはどう比較されるかという問いである。最も単純な 2 点の例を見れば、不等号がどちらを向くかがわかる。確率変数 Z が値 0 と 2 をそれぞれ確率 $\frac{1}{2}$ でとるとし、$f(x) = x^2$ とする。f を平均において評価すると $f(\mathbb{E}[Z]) = f(1) = 1$ となるのに対し、実現値にわたって f の平均をとると $\mathbb{E}[f(Z)] = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 4 = 2$ となる。この差 1 はまさに Z の分散にほかならない。実際、$f(x) = x^2$ に対しては差 $\mathbb{E}[Z^2] - (\mathbb{E}[Z])^2$ は常に分散となる。Figure 35.11(a) を振り返ってほしい。弦不等式は重み θ と $1-\theta$ をもつ 2 点分布に対してまさにこの比較をしており、先ほど計算した差 $\theta(1-\theta)(x-y)^2$（x^2 に対する弦とグラフの差）はその分布の分散そのものである。

この議論には 2 点に限られる理由はまったくない。Equation 35.25 を帰納的に適用すれば（演習問題 2）、次の有限形式が得られる。

$$f\left(\sum_i \theta_i \mathbf{x}_i\right) \;\le\; \sum_i \theta_i\, f(\mathbf{x}_i), \qquad \theta_i \ge 0,\; \sum_i \theta_i = 1. \tag{35.29}$$

和が 1 になる非負の重みは点 $\mathbf{x}_i$ 上の離散確率分布を定めるため、有限個の平均を任意の期待値に置き換えるのが自然な一般化となる。

命題（イェンセンの不等式）。 **f を凸集合 $C \subseteq \mathbb{R}^d$ 上の凸関数とし、$\mathbf{Z}$ を C に値をとる確率ベクトルで、$\mathbb{E}\|\mathbf{Z}\| < \infty$ かつ平均 $\mathbf{m} = \mathbb{E}[\mathbf{Z}]$ が C の内部に属するものとする。このとき、次が成り立つ。**

$$f(\mathbf{m}) \;\le\; \mathbb{E}\left[f(\mathbf{Z})\right]. \tag{35.30}$$

さらに f が狭義凸であるならば、等号が成り立つのは $\mathbf{Z} = \mathbf{m}$ が確率 1 で成り立つとき、かつそのときに限る。

証明。 $\mathbf{m}$ は C の内部にあるため、f は少なくとも 1 つの劣勾配 $\mathbf{g} \in \partial f(\mathbf{m})$ をもつ。点 $\mathbf{m}$ における支持超平面は、あらゆる実現値に対して一斉に f を下から抑える。すなわち Equation 35.28 により、次が成り立つ。

$$f(\mathbf{Z}) \;\ge\; f(\mathbf{m}) + \mathbf{g}^{\top}(\mathbf{Z} - \mathbf{m}).$$

ここで両辺の期待値をとる。$\mathbb{E}[\mathbf{Z} - \mathbf{m}] = \mathbf{0}$ であるため 1 次の項は消え、$\mathbb{E}[f(\mathbf{Z})] \ge f(\mathbf{m})$ が残る。f が狭義凸であるとき、この支持超平面はグラフと点 $\mathbf{m}$ **のみ**で接する。実際、もしある $\mathbf{x} \neq \mathbf{m}$ でも接したとすれば、狭義凸性により $f\big(\frac{\mathbf{x}+\mathbf{m}}{2}\big) < \frac{1}{2}f(\mathbf{x}) + \frac{1}{2}f(\mathbf{m})$ が強いられ、中点においてグラフが支持超平面より真に下側に来てしまい矛盾する。したがって、各点ごとの不等式は $\mathbf{Z} \neq \mathbf{m}$ である限り狭義で成り立つため、期待値で等号が成り立つには確率 1 で $\mathbf{Z} = \mathbf{m}$ でなければならない。$\blacksquare$

副産物として、同じ各点ごとの下界から $f(\mathbf{Z})$ が可積分な確率変数によって下から抑えられることがわかるため、$\mathbb{E}[f(\mathbf{Z})]$ は常に確定した値をもつ（$+\infty$ になり得るが、その場合も不等式は自明に成り立つ）。平たく言えば、平均を固定したまま確率変数のばらつきを広げると、凸関数の期待値は増加しこそすれ減少することはない。凹関数 f に対しては不等号の向きが逆転し、$\mathbb{E}[f(\mathbf{Z})] \le f(\mathbb{E}[\mathbf{Z}])$ となる。

機械学習においてイェンセンの不等式が最も頻繁に使われる場面は、KL ダイバージェンスが決して負にならないことの証明である。37.1 節では、この同じ結果にギブスの不等式（Gibbs' inequality）として出会う。

系（KL ダイバージェンスの非負性）。 **共通の有限アルファベット上の確率分布 p と q について、$q(x) > 0$ が $p(x) > 0$ となるすべての点で成り立つとする。このとき、**

$$D_{\mathrm{KL}}(p \,\|\, q) \;=\; \sum_{x} p(x) \log \frac{p(x)}{q(x)} \;\ge\; 0,$$

が成り立ち、等号成立は $p = q$ のとき、かつそのときに限る。

証明。 $Z \sim p$ を抽出し、狭義凸関数 $-\log$ と正の確率変数 $R = q(Z)/p(Z)$ に対してイェンセンの不等式を適用する。

$$D_{\mathrm{KL}}(p \,\|\, q) = \mathbb{E}_p\left[-\log R\right] \;\ge\; -\log \mathbb{E}_p[R] = -\log \sum_{x : p(x) > 0} p(x)\, \frac{q(x)}{p(x)} \;\ge\; -\log 1 = 0,$$

ここで最後の不等号には $\sum_{x:p(x)>0} q(x) \le 1$ を用いた。イェンセンの不等式の等号成立条件により、等号が成り立つのは、R が確率 1 で定数であり、かつ q が p の台の外に確率質量をもたないとき、すなわち $q = c\,p$（ただし $c = 1$）のとき、かつそのときに限る。$\blacksquare$

同じ発想を凹関数 log に適用すれば、古典的な**相加・相乗平均の不等式**（arithmetic–geometric mean inequality）が得られる。任意の正の数 $x_1, \dots, x_N$ に対し、凹性から $\log\big(\frac{1}{N}\sum_i x_i\big) \geq \frac{1}{N}\sum_i \log x_i$ が成り立ち、両辺の指数をとれば次式が得られる。

$$\frac{1}{N}\sum_{i=1}^{N} x_i \;\geq\; \left(\prod_{i=1}^{N} x_i\right)^{1/N},$$

ここで等号が成り立つのは、すべての x_i が等しいとき、かつそのときに限る（log が狭義凹関数だからである）。同様に、期待値の対数を下から評価するエビデンス下界（evidence lower bound、ELBO）（36.7 節）も、log に対するイェンセンの不等式の直接的な応用である。

イェンセンギャップと KL ダイバージェンスの等号成立付近の挙動を、ともに数値的に検証してみよう。第 1 に、$Z \sim \mathcal{N}(0,1)$ と $f(x) = e^x$ に対しては両辺を閉じた式で計算でき（$\mathbb{E}[e^Z] = \sqrt{e} \approx 1.649$ に対して $e^{\mathbb{E}[Z]} = 1$）、モンテカルロ推定がどれほど正確にこれに一致するかを確認できる。第 2 に、q を p へ $(1-\varepsilon)p + \varepsilon q$ によって混ぜ合わせていくとき、KL ダイバージェンスはどれほど速くゼロへ近づくだろうか。ε に関して線形だろうか、それともさらに速いだろうか。

```
rng = np.random.default_rng()
z = rng.normal(size=1_000_000)                # Z ~ N(0, 1)
print(f'E[exp(Z)] = {np.exp(z).mean():.2f} vs exp(E[Z]) = {np.exp(z.mean()):.2f}'
      f'  (population: {np.exp(0.5):.4f} vs 1)')
kl = lambda p, q: (p * np.log(p / q)).sum(axis=1)
p = rng.dirichlet(np.ones(10), size=1000)     # 1000 random pairs, 10 symbols
q = rng.dirichlet(np.ones(10), size=1000)
print('median KL(p || (1-eps) p + eps q), eps = 1e-2, 1e-3, 1e-4:',
      *[f'{np.median(kl(p, (1 - e) * p + e * q)):.1e}' for e in (1e-2, 1e-3, 1e-4)])
```

```
E[exp(Z)] = 1.65 vs exp(E[Z]) = 1.00  (population: 1.6487 vs 1)
median KL(p || (1-eps) p + eps q), eps = 1e-2, 1e-3, 1e-4: 1.6e-04 1.6e-06 1.6e-08
```

モンテカルロ平均はちょうど 1.65 対 1.00 の付近に落ち着く。サンプル数を増やせば推定ノイズは縮まるが、構造的なイェンセンギャップは一定のままである。KL ダイバージェンスの実験では、ダイバージェンスは ε よりはるかに速く消失し、ε を 10 分の 1 にするごとに中央値はおよそ 100 分の 1 に減少する。対数を展開すればその理由がわかる。すなわち、$D_{\mathrm{KL}}(p \,\|\, (1-\varepsilon)p + \varepsilon q) \approx \frac{\varepsilon^2}{2}\sum_x (q(x)-p(x))^2/p(x)$ が、$\varepsilon\, q(x)/p(x)$ がすべての x について十分小さくなった段階で成り立つからである。（p がいずれかの記号に極小の確率を割り当てるペアはこの 2 次の領域に入るのが最も遅れるため、ここでは最大値ではなく中央値を報告している。）等号には 2 次のオーダーで近づき、極限 $q \to p$ においてのみ達成される。

35.3.3.2 極小点と大域的最小点

凸関数のすべての極小点が大域的最小点であることはすでに証明した（35.3.2.2 節）。これと対になる 3 つの事実も、同じくらい手軽にそれぞれ 1 行で導かれる。第 1 に、すべての大域的最小点からなる集合は凸である。$\mathbf{x}$ と $\mathbf{y}$ がともに最小値 $f^\star$ を達成するならば、弦不等式によって両者の間のすべての点が $f^\star$ 以下に抑えられるからである。第 2 に、開集合上の微分可能な凸関数 f に対しては、$\nabla f(\mathbf{x}) = \mathbf{0}$ からただちに $f(\mathbf{x}) = f^\star$ が従う。勾配がゼロのとき、一階条件は $f(\mathbf{y}) \geq f(\mathbf{x})$ がすべての $\mathbf{y}$ に対して成り立つと読めるからである。第 3 に、勾配が小さければ準最適性ギャップ（suboptimality gap）も小さいことが保証される。一階条件を最小点 $\mathbf{x}^\star$ に向けて適用すれば、$f(\mathbf{x}) - f^\star \leq \nabla f(\mathbf{x})^\top(\mathbf{x} - \mathbf{x}^\star) \leq \|\nabla f(\mathbf{x})\|\,\|\mathbf{x} - \mathbf{x}^\star\|$ が得られる。したがって、

35.1.1 節で示した停留性の結果（$\eta = 1/L$ の勾配降下法が最小の勾配ノルムをゼロへ収束させること）は、凸目的関数に対しては、反復点が $\mathbf{x}^\star$ から有界な距離内にとどまる限り、**損失値**そのものに対する保証へと格上げされる。35.3.3.3 節では、反復点が実際に有界な距離にとどまることを証明し、正確な収束率を突き止める。

単なる凸性が保証**しない** 2 つの点には注意しておこう。最小点は一意であるとは限らず（凸関数は線分全体に沿って平坦になり得る）、そもそも存在するとも限らない（e^x は 0 に $x \to -\infty$ で限りなく近づくが、決してその値をとらない）。狭義凸性は複数の最小点を排除し、強凸性は存在と一意性の両方を保証する。

命題（一意性）。 狭義凸な f の最小点は高々 1 つである。連続な μ 強凸関数 f は、空でない内部をもつ空でない閉凸集合 C 上でただ 1 つの最小点をもつ。

証明。 $\mathbf{x} \neq \mathbf{y}$ がともに値 $f^\star$ を達成する大域的最小点であったと仮定する。$\theta = \frac{1}{2}$ における狭義の弦不等式により、両者の中点は $f(\frac{\mathbf{x}+\mathbf{y}}{2}) < f^\star$ を満たすことになり矛盾する。強凸性は狭義凸性を導くため、残るは存在の証明のみである。ここでは本書の制約なし損失をカバーする、微分可能な f と $C = \mathbb{R}^d$ の場合について証明する（一般の場合も、$f - \frac{\mu}{2}\|\cdot\|^2$ の C の内点における劣勾配を用いれば同じ手順で従う）。任意の初期点 $\mathbf{x}_0$ を固定する。Equation 35.27 において 1 次の項は少なくとも $-\|\nabla f(\mathbf{x}_0)\|\,\|\mathbf{y} - \mathbf{x}_0\|$ であるため、$f(\mathbf{y}) > f(\mathbf{x}_0)$ が $\|\mathbf{y} - \mathbf{x}_0\| > 2\|\nabla f(\mathbf{x}_0)\|/\mu$ である限り成り立つ。よって劣レベル集合 $\{\mathbf{y} : f(\mathbf{y}) \le f(\mathbf{x}_0)\}$ は有界であり、f の連続性から閉集合でもある。空でない有界閉集合上のすべての連続関数は最小値を達成し、この劣レベル集合上の最小値は自動的に $\mathbb{R}^d$ 全体での大域的最小値となる。■

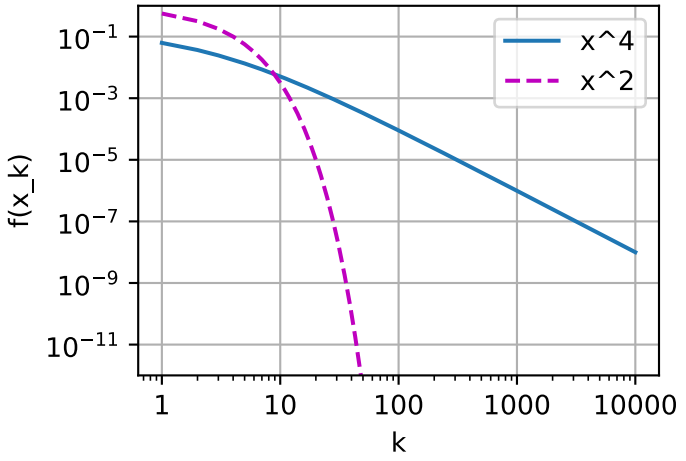
C と連続性に関する技術的な仮定は、定義域の境界で上向きに不連続に跳び上がる病的な関数を除外しているにすぎず、本書で扱う損失ではそのような事態は決して起こらない。対照的に、e^x に最小点が存在しない現象は実際の機械学習にも現れる。演習問題 3 では、これが線形分離可能なデータに対するロジスティック回帰へ直接つながることを確かめる。

35.3.3.3 収束率

凸性から勾配降下法が**どこで**止まるかはわかるが、そこへ**どれほど速く**到達するかはわからない。しかも速度に関しては、単なる凸性が約束してくれることは驚くほど少ない。$f(x) = x^2$ と $f(x) = x^4$ を比べてみよう。どちらも一意な最小点 0 をもつ狭義凸関数であるが、強凸なのは x^2 だけである。まったく同じ勾配降下法のループ（$\eta = \frac{1}{8}$、初期点 $x_0 = 1$）を両方の関数に対して 10^4 ステップ実行し、どれほど異なる振る舞いを見せるかを観察しよう。

```
eta, steps = 1 / 8, 10_000
x2, x4 = np.ones(steps + 1), np.ones(steps + 1)  # iterates from x_0 = 1
for j in range(steps):                           # same start, same step
    x2[j + 1] = x2[j] - eta * 2 * x2[j]          # f(x) = x^2
    x4[j + 1] = x4[j] - eta * 4 * x4[j]**3       # f(x) = x^4
k = np.arange(1, steps + 1)
print('x^2: first k with x_k < 1e-6:', k[x2[1:] < 1e-6][0])
print('x^4: x_k * sqrt(k) at k = 1e2, 1e3, 1e4:',
      (x4[1:] * np.sqrt(k))[[99, 999, 9999]].round(3))
d2l.plot(k, [x4[1:]**4, x2[1:]**2], 'k', 'f(x_k)', xscale='log',
         yscale='log', ylim=(1e-12, 1), legend=['x^4', 'x^2'])
```

```
x^2: first k with x_k < 1e-6: 49
x^4: x_k * sqrt(k) at k = 1e2, 1e3, 1e4: [0.973 0.996 1.   ]
```



対比は劇的である。x^2 では、各ステップが反復点に定数因子 $1-2\eta = 0.75$ を掛け、10^{-6} 未満へわずか 49 ステップで押し下げるため、損失はほぼ一瞬でプロットの下端を突き抜ける。対照的に x^4 では、各ステップが x_k に掛ける縮小率 $1-4\eta x_k^2$ は 1 へとじわじわ近づいていく（反復点が 0 の平坦な谷へ近づくにつれてそうなる）。出力された積 $x_k\sqrt{k}$ が 1 付近に落ち着くことから、x_k は $1/\sqrt{k}$ の速さでしか減衰せず、なお 0.01 の値に 10,000 ステップ後でもとどまっていることがわかる。そのため両対数プロット上では、損失ギャップ x_k^4 は傾き -2 の直線を描く。狭義凸性は複数の最小点を排除するが、速い収束は保証できない。幾何収束率を得るには $\mu > 0$ が必要である。

次の 2 つの命題は、この対比を完全に一般的な形で定式化する。すなわち、滑らかな強凸関数はすべて x^2 のように幾何収束するのに対し、$\mu > 0$ がなければ保証できるのは $O(1/k)$ の減衰にとどまる。どちらの証明も、35.1.1 節の**降下補題**（descent lemma）から出発する。L 滑らかな関数 f と勾配ステップ $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$ に対し、次が成り立つ。

$$f(\mathbf{x}_{k+1}) \;\le\; f(\mathbf{x}_k) - \frac{1}{2L}\,\|\nabla f(\mathbf{x}_k)\|^2.$$

各ステップは目的関数を少なくとも $\|\nabla f(\mathbf{x}_k)\|^2/(2L)$ だけ減少させる。一般の非凸関数では、ここから言えるのは勾配がゼロへ縮むことだけである。凸性は、勾配と準最適性ギャップを結ぶ欠けた環を補う。$f(\mathbf{x}_k) - f^\star$ がまだ大きい限り、前進を続けるのに十分な大きさの勾配が必ず存在するため、反復点が最適点から遠く離れた場所で停滞することはあり得ない。

命題（滑らかな凸関数に対する勾配降下法）。 *f を凸かつ L 滑らかな関数とし、$\mathbf{x}^\star$ をその最小点とする。$\eta = 1/L$ を用いた勾配降下法は、すべての $k \ge 1$ に対して次を満たす。*

$$f(\mathbf{x}_k) - f(\mathbf{x}^\star) \;\le\; \frac{L\,\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{2k}. \tag{35.31}$$

したがって損失ギャップは、滑らかさの定数と最小点までの初期距離の二乗を係数として、少なくとも $1/k$ の速さで減衰する。具体的な数値を入れると、$L = 1$ かつ $\|\mathbf{x}_0 - \mathbf{x}^\star\| = 10$ であれば上界は $50/k$ となり、ギャップが 10^{-2} 未満になることが 5,000 ステップ以内に保証される。

証明。 降下補題と、一階条件 Equation 35.26 を点 $\mathbf{x}_k$ において最適点へ向けて評価した式 $f(\mathbf{x}_k) - f(\mathbf{x}^\star) \le \nabla f(\mathbf{x}_k)^\top(\mathbf{x}_k - \mathbf{x}^\star)$ を組み合わせる。

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^\star) \;\le\; \nabla f(\mathbf{x}_k)^\top(\mathbf{x}_k - \mathbf{x}^\star) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2.$$

ここで巧妙な変形を行う。平方完成によって、右辺を $\mathbf{x}^\star$ までの 2 つの二乗距離の差に書き換えるのである。

$$\begin{aligned}\nabla f(\mathbf{x}_k)^\top(\mathbf{x}_k - \mathbf{x}^\star) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2 \\ = \frac{L}{2}\left(\|\mathbf{x}_k - \mathbf{x}^\star\|^2 - \left\|\mathbf{x}_k - \mathbf{x}^\star - \tfrac{1}{L}\nabla f(\mathbf{x}_k)\right\|^2\right).\end{aligned}$$

第 2 のノルムはまさに $\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2$ であることに注目しよう。$j = 0, \dots, k-1$ にわたって和をとると、右辺は望遠鏡和としてきれいに相殺される。

$$\begin{aligned}\sum_{j=0}^{k-1}\big(f(\mathbf{x}_{j+1}) - f(\mathbf{x}^\star)\big) &\le \frac{L}{2}\left(\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 - \|\mathbf{x}_k - \mathbf{x}^\star\|^2\right) \\ &\le \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2.\end{aligned}$$

降下補題により $f(\mathbf{x}_j)$ は決して増加しないため、最後の項 $f(\mathbf{x}_k) - f(\mathbf{x}^\star)$ がこの和の最小の項となり、したがって $k\,(f(\mathbf{x}_k) - f(\mathbf{x}^\star))$ は和全体で上から抑えられる。k で割れば Equation 35.31 が得られる。■

この証明には注目すべき特徴が 2 つある。第 1 に、次元 d が一度も現れない。10 億個の重みをもつモデルであっても、L と初期距離が同じであればまったく同じ最悪ケースの保証を享受する。第 2 に、和をとる前の 1 ステップごとの不等式は非負の左辺 $f(\mathbf{x}_{k+1}) - f^\star \ge 0$ をもつため、$\|\mathbf{x}_{k+1} - \mathbf{x}^\star\| \le \|\mathbf{x}_k - \mathbf{x}^\star\|$ が強制される。すなわち、1 ステップ進むごとにあらゆる最小点へ単調に近づいていくのである。

強凸性を加えると、この $1/k$ の減衰は幾何収束率へと一変する。

命題（強凸性のもとでの線形収束率）。 *f を μ 強凸かつ L 滑らかな関数とし、$\mathbf{x}^\star$ をその最小点とする。$\eta = 1/L$ を用いた勾配降下法は次を満たす。*

$$f(\mathbf{x}_k) - f(\mathbf{x}^\star) \;\le\; \left(1 - \frac{\mu}{L}\right)^{k}\left(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\right). \tag{35.32}$$

言葉で言えば、反復を 1 回行うごとに、残っている損失ギャップの少なくとも割合 $\mu/L = 1/\kappa$ が削り取られる。

証明。 証明は 2 つの短いステップに分かれる。まず、強凸性により勾配ノルムの二乗が準最適性ギャップによって下から抑えられる。Equation 35.27 の両辺を $\mathbf{y}$ について最小化する。左辺はその最小値 $f(\mathbf{x}^\star)$ を達成し、一方右辺は $\mathbf{y}$ に関する単純な二次関数であり、$\mathbf{y} = \mathbf{x} - \frac{1}{\mu}\nabla f(\mathbf{x})$ において最小化され、そのときの値は $f(\mathbf{x}) - \frac{1}{2\mu}\|\nabla f(\mathbf{x})\|^2$ となる。これを整理すると次式を得る。

$$\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \;\ge\; \mu\,(f(\mathbf{x}) - f(\mathbf{x}^\star)) \qquad \text{for all } \mathbf{x}. \tag{35.33}$$

次に、Equation 35.33 を降下補題へ直接代入する。

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^\star) \le f(\mathbf{x}_k) - f(\mathbf{x}^\star) - \frac{1}{2L}\|\nabla f(\mathbf{x}_k)\|^2$$
$$\le \left(1 - \frac{\mu}{L}\right)\left(f(\mathbf{x}_k) - f(\mathbf{x}^\star)\right).$$

この縮小を k 回適用すれば証明が完了する。■

この証明の後半をよく見てほしい。そこでは凸性を一度も使っておらず、用いたのは不等式 Equation 35.33 だけである。この驚くべき事実には 35.3.5 節で立ち返る。また、強凸でない場合と比べたときの劇的な高速化にも注目しよう。相対精度 ϵ に到達するのに必要なステップ数は、もはや $O(1/\epsilon)$ ではなくおよそ $\kappa \log(1/\epsilon)$ で済む。$\kappa = 10^3$ の通し例の二次関数では、6 桁の精度改善 $(1 - 10^{-3})^k \le 10^{-6}$ に要するステップ数は、この上界によれば約 13,800 ステップである。

Equation 35.32 は実際にはどれほど精密な評価なのだろうか。ヘッセ行列 $\mathbf{X}^\top\mathbf{X}$ が構成上、固有値 1 と 10 をもつ最小二乗問題（$\mu = 1$、$L = 10$、$\kappa = 10$）で測ってみよう。$\eta = 1/L$ で勾配降下法を実行し、連続する損失ギャップの実際の 1 ステップあたりの比 $(f(\mathbf{w}_{k+1}) - f^\star)/(f(\mathbf{w}_k) - f^\star)$ を追跡する。

```
rng = np.random.default_rng()
Q, _ = np.linalg.qr(rng.normal(size=(20, 2)))  # orthonormal columns
R, _ = np.linalg.qr(rng.normal(size=(2, 2)))   # random rotation
X = Q @ np.diag([1.0, np.sqrt(10.0)]) @ R      # X^T X has eigenvalues 1, 10
y = rng.normal(size=20)
f = lambda w: 0.5 * ((X @ w - y) ** 2).sum()
mu, L = np.linalg.eigvalsh(X.T @ X)[[0, -1]]
w_star = np.linalg.lstsq(X, y, rcond=None)[0]
w, gaps = np.zeros(2), [f(np.zeros(2)) - f(w_star)]
for _ in range(60):                            # gradient descent, eta = 1/L
    w = w - X.T @ (X @ w - y) / L
    gaps.append(f(w) - f(w_star))
r = np.array(gaps[1:]) / np.array(gaps[:-1])   # per-step contraction of f - f*
print(f'mu = {mu:.4f}, L = {L:.4f}: bound 1 - mu/L = {1 - mu / L:.4f},'
      f' its square {(1 - mu / L)**2:.4f}')
print('first three ratios:', r[:3].round(4), ' largest ratio:', r.max().round(4))
```

```
mu = 1.0000, L = 10.0000: bound 1 - mu/L = 0.9000, its square 0.8100
first three ratios: [0.6084 0.81   0.81  ]  largest ratio: 0.81
```

先ほどの上界が約束する 1 ステップあたりの縮小率は 0.9 であるが、2 ステップ目以降に実測された比は 0.81、すなわち $(1-\mu/L)^2$ となっている。（最初の 1 回目の比だけはランダムな初期方向に依存し、実行ごとに変動する。）更新を $\mathbf{X}^\top\mathbf{X}$ の固有ベクトル座標で見れば、何が起きたのかがわかる。$\eta = 1/L$ とすると、最初の 1 ステップが $\mathbf{w} - \mathbf{w}^\star$ の硬い固有成分に $1 - L/L = 0$ を掛けて完全に消し去るのである。2 次元では遅い成分だけが残る。その $\mathbf{w}^\star$ までの距離は 1 ステップごとに $1 - \mu/L$ 倍に縮むため、距離の二乗に比例する損失ギャップは $(1-\mu/L)^2$ 倍に縮む。より高い次元では、速い固有成分が減衰するにつれて、この比は $(1-\mu/L)^2$ へと近づいていく。

先ほどの短い証明では、縮小率の二乗分を大まかに見積もって手放していたのである。より精密な解析によれば、$\eta = 1/L$ の勾配降下法は実際には、$f - f^\star$ を 1 ステップあたり $(1-\mu/L)^2$ の割合で、**すべての** L 滑らかかつ μ 強凸関数に対して縮小させ (Taylor et al. 2018)、二次関数において等号が達成される。演習問題 5 では、これと密接に関連する上界 $\kappa(1-\mu/L)^{2k}$ を証明する。$\kappa = 10^3$ の通し例の二次関数に戻ると、損失ギャップで 6 桁の精度（最小点までの距離で 3 桁の

精度）を得るのに要するステップ数は、単純な上界では約 $13{,}800$ ステップ、真の $(1-\mu/L)^2$ のレートでは約 $6{,}900$ ステップ、さらに最適な固定ステップサイズ $\eta = 2/(\mu+L)$ ではその半分の $3{,}454$ ステップとなる。これは、同じ最適ステップサイズで距離の精度 6 桁を得るために要した $6{,}908$ ステップ（35.1 節）のちょうど半分である。最初の 2 倍の差は我々の証明の緩さから生じ、もう 1 つの 2 倍の差はステップサイズを 2 倍にしたことから生じている。

35.3.4 凸性の判定

大域的最適性や明快な収束率は、与えられた損失関数がそもそも凸であるかどうかを見分けられて初めて役に立つ。ヘッセ行列を手計算するのはすぐに煩雑になるうえ、折れ点ではそもそも使えないため、実務においてゼロから凸性を確認することはほとんどない。その代わりに、単純な凸の構成要素を組み合わせることで凸目的関数を見分ける。

35.3.4.1 凸関数の演算規則

4 つの基本的な演算が凸性を保存し、機械学習に現れるほぼすべての凸損失は、基本的な凸関数からこれら 4 つの規則だけを使って構成されている。

1. **非負重み付き和**。$f_1, \dots, f_m$ が凸であり $w_i \ge 0$ ならば、$\sum_i w_i f_i$ も凸である。各 f_i の弦不等式に w_i を掛けて足し合わせるだけでよい。特に、訓練例ごとの凸な損失を訓練セット全体で平均すれば、常に凸目的関数が得られる。

2. **アフィン写像との前合成**。f が凸ならば、$\mathbf{x} \mapsto f(\mathbf{A}\mathbf{x}+\mathbf{b})$ も凸である。アフィン写像は $\mathbf{x}$ と $\mathbf{y}$ を結ぶ線分を、まったく同じ重み θ を保ったままそれらの像を結ぶ線分へと写すため、弦不等式がそのまま引き継がれる。その結果、モデルの**予測**に関して凸である任意の損失は、その予測を生成する**線形モデルの重み**に関しても自動的に凸となる。

3. **各点最大値と上限**。すべての f_i が凸ならば、$\max_i f_i$ も凸である。最も手早い証明はエピグラフを用いるものである。各点最大値のエピグラフは個々のエピグラフの共通部分に一致し、凸集合の共通部分は凸である。したがって、アフィン関数のいかなる族の上側包絡線（upper envelope）も、その族がどれほど大きくても凸となる。

4. **単調合成**。g が凸であり、h が凸**かつ非減少**であるならば、合成関数 $h \circ g$ は凸である。2 行で証明できる。

$$h(g(\theta \mathbf{x} + (1-\theta)\mathbf{y})) \;\le\; h(\theta g(\mathbf{x}) + (1-\theta) g(\mathbf{y})) \;\le\; \theta\, h(g(\mathbf{x})) + (1-\theta)\, h(g(\mathbf{y})),$$

   ここで第 1 の不等号は g の凸性と h の単調性を合わせて用い、第 2 の不等号は h の凸性を用いている。h に対する単調性の条件を忘れないようにしよう。$h(t) = t^2$ は凸であるが非減少ではなく、これを凸関数 $g(x) = x^2 - 1$ と合成すると非凸な二重井戸関数 $(x^2-1)^2$ が生じる（ϕ はこれに $x/2$ を加えたものである）。

これら 4 つの規則により、標準的な凸損失のほぼすべてが一挙に保証される。ヒンジ損失 $\max(0, 1 - y\,\mathbf{w}^\top \mathbf{x})$ は $\mathbf{w}$ の 2 つのアフィン関数の最大値である（規則 3 と 2）。ℓ_1 ノルム $\|\mathbf{w}\|_1 = \sum_i \max(w_i, -w_i)$ はアフィン関数の最大値の和である（規則 1、3、2）。ロジスティック損失 $\log(1 + e^{-y\,\mathbf{w}^\top \mathbf{x}})$ は、スカラー関数 $t \mapsto \log(1+e^{t})$（二階導関数 $e^t/(1+e^t)^2$ が真に正であるため凸）にアフィン写像を前合成し（規則 2）、訓練例にわたって和をとったものである（規則 1）。これらのいずれにリッジペナルティ $\frac{\lambda}{2}\|\mathbf{w}\|^2$ を加えても、強凸な目的関数が得られる。ただし、規則 3 のもとでは凸性が自動的に強凸性へ格上げされるわけではないことに注意しよう。$\max(x^2, 1-x)$ は凸であるが、平坦なアフィン直線 $1-x$ と区間 $(-1.618, 0.618)$（$x^2 = 1-x$ の 2 つの根の間）全体にわたって一致するため、強凸ではない。

この一覧から目立って欠けているものに注目してほしい。アフィンでも凸でもない内側の写像との合成である。2 層ニューラルネットワーク $\mathbf{W}_2\,\mathrm{relu}(\mathbf{W}_1\mathbf{x})$ は、その重みに関してまさにそのような合成になっており、ここが凸性の崩れる境界線である（35.3.5 節）。

35.3.4.2 log-sum-exp とソフトマックス共分散

1 つの基本的な損失が、上の 4 つの規則のすぐ外側に位置している。あらゆるソフトマックス交差エントロピー損失の心臓部にある log-sum-exp 関数 $\mathrm{lse}(\mathbf{x}) = \log \sum_{i=1}^{d} e^{x_i}$ である。幾何学的には、$\mathrm{lse}(\mathbf{x})$ は $\max_i x_i$ の滑らかな上からの近似であり、その超過量は決して $\log d$ を超えない。これを無理に合成規則へ当てはめるよりも、ヘッセ行列を通じて直接その凸性を示す方がよい。実はそのヘッセ行列は、驚くほど明快な確率論的解釈をもっている。

命題（log-sum-exp の凸性）。 lse **の点 x におけるヘッセ行列は**

$$\nabla^2 \mathrm{lse}(\mathbf{x}) = \mathrm{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^\top, \qquad \mathbf{s} = \mathrm{softmax}(\mathbf{x}), \tag{35.34}$$

で与えられ、これは s から抽出されたクラスのワンホットエンコーディングの共分散行列に一致する。すべての共分散行列は PSD であるため、lse は凸である。

証明。 1 回微分すると $\partial\,\mathrm{lse}/\partial x_i = e^{x_i}/\sum_k e^{x_k} = s_i$ となるため、lse の勾配はまさにソフトマックスベクトル $\mathbf{s}$ である。$s_i = e^{x_i}/\sum_k e^{x_k}$ を商の微分公式でもう 1 回微分すると次式を得る。

$$\frac{\partial s_i}{\partial x_j} = \frac{\delta_{ij}\, e^{x_i}}{\sum_k e^{x_k}} - \frac{e^{x_i}\, e^{x_j}}{\left(\sum_k e^{x_k}\right)^2} = s_i\,(\delta_{ij} - s_j),$$

これは (i,j) 成分において $\mathrm{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^\top$ と一致する（34.4 節で導出したソフトマックスのヤコビ行列と同じものである）。ここで Y を確率 $P(Y=i) = s_i$ で抽出されるランダムなクラス添字とし、$\mathbf{e}_Y$ をそのワンホットベクトルとする。このとき $\mathbb{E}[\mathbf{e}_Y] = \mathbf{s}$ かつ $\mathbb{E}[\mathbf{e}_Y\mathbf{e}_Y^\top] = \mathrm{diag}(\mathbf{s})$ となるため、ヘッセ行列は文字どおり $\mathrm{Cov}(\mathbf{e}_Y) = \mathrm{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^\top$ に一致する。すべての共分散行列は PSD である。実際、任意の方向 $\mathbf{v}$ に対して明示的に書けば次のようになる。

$$\mathbf{v}^\top \nabla^2 \mathrm{lse}(\mathbf{x})\,\mathbf{v} = \sum_i s_i v_i^2 - \left(\sum_i s_i v_i\right)^2 = \mathbb{E}\left[v_Y^2\right] - \left(\mathbb{E}[v_Y]\right)^2 = \mathrm{Var}(v_Y) \;\geq\; 0. \;\blacksquare$$

言い換えれば、方向 $\mathbf{v}$ に沿った log-sum-exp の曲率は、ソフトマックス分布のもとでの座標 v_Y の分散にほかならない。

この公式から、実用上の 3 つの帰結がただちに従う。第 1 に、ロジットの関数としての交差エントロピー損失 $\ell(\mathbf{z}) = \mathrm{lse}(\mathbf{z}) - z_y$（正解クラス y）は log-sum-exp から 1 次項を引いたものであるため凸である。第 2 に、ソフトマックス回帰（3.1 節）はアフィンなロジット $\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$ を $\ell(\mathbf{z})$ に入力するため、規則 2 により損失は重みに関して凸となる。すなわち、固定された特徴量の上で線形分類器を訓練する問題は常に凸最適化問題である。第 3 に、分散の公式から、全成分が 1 のベクトル $\mathbf{1}$ の方向に**ゼロ**固有値が存在することがただちに予測される。$\mathrm{Var}(v_Y) = 0$ が、すべての成分 v_i が等しいときに成り立つからである。この平坦な方向は、数値的に安定なソフトマックスの計算を可能にするシフト不変性 $\mathrm{lse}(\mathbf{x} + c\mathbf{1}) = \mathrm{lse}(\mathbf{x}) + c$ を反映している。また、まったく同じヤコビ行列 $\mathrm{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^\top$ が、あらゆるアテンションヘッドの逆伝播も支配している。アテンション重みはクエリ・キースコアのソフトマックスだからである（10.2 節）。

log-sum-exp の曲率は最大でどれほど大きくなり得るだろうか。演習問題 6 で証明するように、ロジットがどのような値であってもヘッセ行列の固有値が $\frac{1}{2}$ を超えることは決してないため、$\frac{1}{2}$ が大域的な滑らかさの定数となる。その上限にはどのような場所で近づき、また典型的なロジットにおける損失の曲率はどの程度なのだろうか。以下では、ランダムな 6 クラスのロジットベクトルを 10,000 個抽出し、温度の乗数 c でスケール変換して、全サンプルにわたるヘッセ行列の最大固有値と中央値の両方を報告する。

```
z = rng.normal(size=(10_000, 6))              # 10,000 draws of six logits
print('  c   largest eigenvalue   median')
for c in (0.1, 1.0, 3.0, 10.0, 30.0):
    s = np.exp(c * z - (c * z).max(axis=1, keepdims=True))
    s /= s.sum(axis=1, keepdims=True)         # softmax of each row
    hess = s[:, :, None] * np.eye(6) - s[:, :, None] * s[:, None, :]
    top = np.linalg.eigvalsh(hess)[:, -1]     # largest eigenvalue per draw
    print(f'{c:5.1f}  {top.max():14.4f}  {np.median(top):12.4f}')
```

```
  c   largest eigenvalue   median
 0.1          0.2193        0.1843
 1.0          0.4546        0.3071
 3.0          0.4983        0.2995
10.0          0.5000        0.0165
30.0          0.5000        0.0000
```

予測どおり、最大固有値は決して $\frac{1}{2}$ を超えず、ソフトマックス交差エントロピーがあらゆる温度においてロジットに関し $\frac{1}{2}$ 滑らかであることが確認できる。ロジットが小さいとき（$c = 0.1$）、ソフトマックスは 6 クラスにわたってほぼ一様となり、曲率は $\frac{1}{6}$ をわずかに上回るあたりに集まる。中程度のスケールでは典型的な曲率がピークを迎え、中央値は 0.3 付近となる。しかし c をさらに大きくすると二極化が生じる。典型的なサンプルは強い確信をもつ（ほぼワンホットになる）ため、曲率の中央値はゼロに向かって潰れていく。すなわち、確信度の高い予測の周りでは損失はほぼ完全に平坦になる。その一方で、10,000 個のサンプルにわたる**最大値**は $\frac{1}{2}$ へと迫っていく。これは上位 2 つのロジットがほぼ同値になり、ソフトマックスが確率を 2 クラスに均等に分け合う稀なサンプルから生じている。

35.3.5 凸性を超えて

深層ニューラルネットワークの損失ランドスケープは凸ではない。この最後の項では、わずか 2 つのスカラー重みを用いてその理由を示し、凸性なしに勾配降下法の線形収束率を保つより弱い条件を特定し、さらに無限に多くの大域的最小点が存在するときにオプティマイザがどのように 1 つの解を選び出すのかを考察する。

35.3.5.1 非凸なネットワーク

$f(a, b) = (ab - 1)^2$ を考えよう。これは 2 つのスカラー重みを直列に重ねた線形モデルに二乗損失を課したものであり、考え得る最も単純な「深層」ネットワークである。Figure 35.14 に示すように、その大域的最小点は双曲線 $\{ab = 1\}$ をなす。35.3.3 節では任意の凸関数の最小点集合が凸でなければならないことを証明したが、双曲線はどう見ても凸ではない。2 つの最小点 $(1, 1)$ と $(-1, -1)$ の中点は原点 $(0, 0)$ であり、そこでは $f(0, 0) = 1$ となって最小値 0 よりはるかに高い値をとる。原点は最小点ではなく停留点であり、そのヘッセ行列 $\nabla^2 f(0, 0)$ は固有値 ± 2 をもつため鞍点となっている。

実際のニューラルネットワークも、これと密接に関連する構造的な理由から凸性を失う。任意の

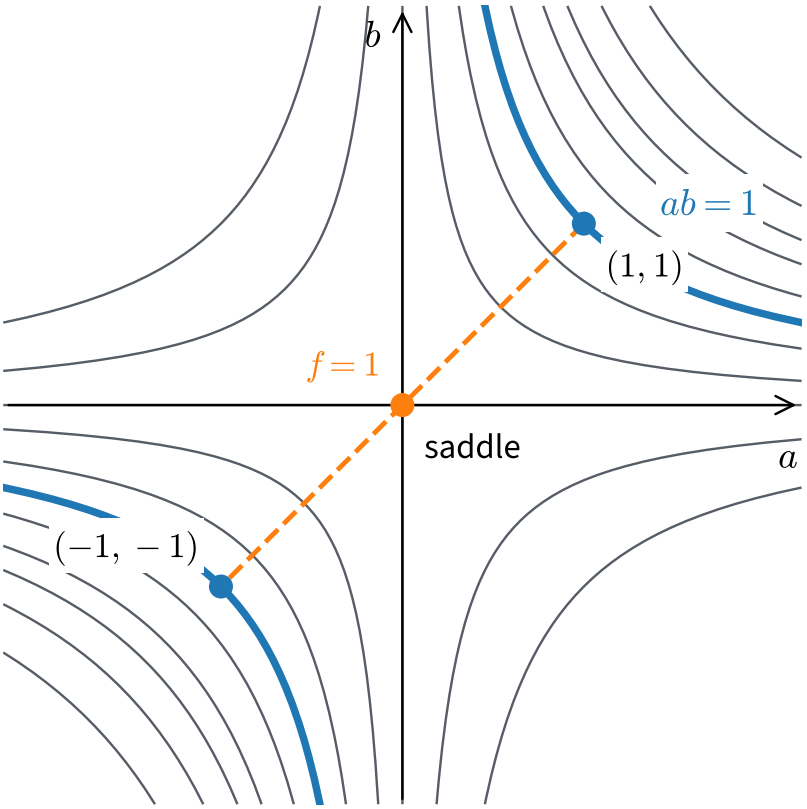


Figure 35.14: 2 重みモデル $(ab-1)^2$。その最小点は双曲線 $ab=1$ をなし、これは凸ではない。$(1,1)$ と $(-1,-1)$ の中点は原点であり、そこでは $f=1$ かつヘッセ行列の固有値は ± 2 である。

層の隠れユニットをその入力重みおよび出力重みとともに並べ替えても、ネットワークの入出力関数は変化しないため、あらゆる最小点にはパラメータ空間全体に散らばる膨大な数の対称なコピーが自動的に伴う。一般にこれらのコピーは互いに異なる点であり、そのうち 2 つを平均すると関数が変化して損失が増大する。ちょうど $(1,1)$ と $(-1,-1)$ の平均が原点の鞍点に落ちたのと同じである。異なる点を含みながらその中点を含まない最小点集合は、凸目的関数のものにはなり得ない。

2 個の ReLU ユニットしかもたない極小のネットワークでさえ、すでにこの破綻を示す。パラメータベクトル $(w_1, v_1, w_2, v_2) = (1, 1, -1, 1)$ は $v_1\,\mathrm{relu}(w_1 x) + v_2\,\mathrm{relu}(w_2 x)$ において $|x|$ を計算し、2 つのユニットを入れ替えてもまったく同じく $|x|$ を計算する。ところが、これら 2 つのパラメータベクトルを平均すると入力重みが打ち消し合ってゼロになり、ネットワークは恒等的にゼロを返す関数へと潰れてしまう。

35.3.5.2 Polyak–Łojasiewicz 条件

強凸性のもとでの線形収束率の証明がどのように進んだかを思い出してみよう。凸性を使ったのは最初のステップで Equation 35.33 を導くときだけであり、第 2 のステップでは凸性に一切触れていなかった。この中間の不等式には固有の名前が付いている。関数 f（下限 $f^\star$）が定数 $\mu > 0$ の **Polyak–Łojasiewicz（PL）不等式**を満たすとは、$\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \geq \mu\,(f(\mathbf{x}) - f^\star)$ がすべての $\mathbf{x}$ について成り立つことをいう。これは 1963 年に Polyak と Łojasiewicz が独立に導入した条件である (Polyak 1963; Łojasiewicz 1963)。直感的には、勾配が小さくなることが許されるのは損失がすでに最適値に近い場所に限られる、ということを述べている。

命題（PL 条件のもとでの線形収束率）。 *f を L 滑らかで下に有界（$f^\star = \inf f$）な関数とし、定数 μ の PL 不等式を満たすとする。このとき、$\eta = 1/L$ を用いた勾配降下法は $f(\mathbf{x}_k) - f^\star \leq (1 - \mu/L)^k\,(f(\mathbf{x}_0) - f^\star)$ を満たす。*

証明。 強凸性の証明の後半をそのまま繰り返せばよい。降下補題と PL 不等式を組み合わせると $f(\mathbf{x}_{k+1}) - f^\star \leq (1 - \mu/L)(f(\mathbf{x}_k) - f^\star)$ が得られ (Karimi et al. 2016)、これを k 回反復すれば証明が完了する。■

PL 条件がこれほど魅力的なのは、かなりの非凸性を許容する点にある。PL 関数は複数の最小点や、最小点がなす平坦な多様体、さらには負の曲率をもつ領域さえもち得る。しかし、譲れない要件が 2 つある。すべての停留点が大域的最小点でなければならないこと（$\nabla f(\mathbf{x}) = \mathbf{0}$ から $f(\mathbf{x}) = f^\star$ が強いられるため）、**かつ**最小点集合から離れるにつれて $f - f^\star$ が少なくとも 2 次のオーダーで増大しなければならないことである。4 次関数 x^4 は、なぜ第 2 の要件が重要かを示している。停留点は大域的最小点の 1 つしかもたないが、$\frac{1}{2}f'(x)^2 = 8x^6$ は μx^4 を $x \to 0$ においていかなる $\mu > 0$ に対しても上回り続けられないため、PL 条件を満たさない。まさにこの破綻こそ、35.3.3.3 節で勾配降下法が x^4 上を這うようにしか進めなかった理由である。要するに、強凸性から PL 条件は従う（先ほど証明したとおりである）が、その逆は成り立たない。

非凸な PL 関数の古典的な例は $f(x) = x^2 + 3\sin^2 x$ であり、その二階導関数 $2 + 6\cos 2x$ は -4 まで落ち込む一方で最大値 $L = 8$ に達する。区間 $[-5, 5]$ 上でその PL 定数を数値的に推定し、$x_0 = 4.5$ からステップサイズ $\eta = \frac{1}{16} = 1/(2L)$ を用いて勾配降下法を実行してみよう。$(1/(2L))$ を $1/L$ の代わりに選ぶのは、$\eta = 1/L$ では原点における局所的な縮小率 $1 - f''(0)\,\eta$ がゼロになり、漸近的な収束率が隠れてしまうためである。）$\eta = 1/(2L)$ においては、二次上界により 1 ステップあたり $\frac{3}{8L}f'(x_k)^2$ の減少が保証され、PL 不等式がこれを縮小率 $1 - \frac{3\mu}{4L}$ へと変換する。

```
f_pl = lambda x: x**2 + 3 * np.sin(x)**2       # nonconvex; f* = 0 at x = 0
df_pl = lambda x: 2 * x + 3 * np.sin(2 * x)
```

```
xs = np.linspace(-5, 5, 100001)
xs = xs[np.abs(xs) > 1e-3]                    # avoid 0/0 at the minimizer
mu = (0.5 * df_pl(xs)**2 / f_pl(xs)).min()
print(f'PL constant on [-5, 5]: mu = {mu:.4f}')
print(f"min f'' = {(2 + 6 * np.cos(2 * xs)).min():.2f}  (f is not convex)")
path = [4.5]
for _ in range(80):                           # gradient descent, eta = 1/16
    path.append(path[-1] - df_pl(path[-1]) / 16.0)
path = np.array(path)
ratios = f_pl(path[1:]) / f_pl(path[:-1])     # per-step contraction of f - f*
j = ratios.argmax()
print(f'guaranteed {1 - 3 * mu / 32:.4f}; largest ratio {ratios[j]:.4f}'
      f' from x = {path[j]:.2f}; last ratio {ratios[-1]:.4f}')
```

```
PL constant on [-5, 5]: mu = 0.1755
min f'' = -4.00  (f is not convex)
guaranteed 0.9835; largest ratio 0.9778 from x = 2.19; last ratio 0.2500
```

区間 $[-5, 5]$ 上での PL 定数は $\mu \approx 0.1755$ であり、f'' の最小値は -4 である。すなわち、この関数は非凸でありながら PL 条件を満たしている。この区間を調べるだけで十分である。というのも、その外側では比 $\frac{1}{2}f'(x)^2/f(x)$ はこの値を優に上回り続け、2 へと $|x| \to \infty$ で収束するからである。演習問題 4 でこの下界を証明する。

この μ を用いると、保証される縮小率は 1 ステップあたり約 0.98 となる。実測された最大の比は約 0.978 で、$x \approx 2.2$ の付近（$\frac{1}{2}f'(x)^2/f(x)$ が最小値 μ をとる場所）から生じており、そこでは理論保証はほぼタイトである。最小点の近傍では、この比は 0.25 に落ち着く。この極限値は局所的な二次近似から説明できる。0 の近くでは $f \approx \frac{1}{2}f''(0)\,x^2$（ただし $f''(0) = 8$）となるため、各ステップが x に $1 - 8/16 = \frac{1}{2}$ を掛け、x の二乗に比例するギャップには $\frac{1}{4}$ を掛けるからである。

PL 条件は空間の一部でのみ成り立つこともある。$(ab-1)^2$ では勾配が $\frac{1}{2}\|\nabla f\|^2 = 2(a^2+b^2)f$ を満たすため、PL 条件は定数 $2r^2$ で領域 $\{a^2+b^2 \geq r^2\}$ 上において成り立つ一方、原点の鞍点の近くではいかなる正の定数でも成り立たない。幅の広いネットワークの理論解析では、補間解の近傍においてこのような局所的な PL 不等式が示されている (C. Liu et al. 2022)。ただし、これが与えるのは訓練損失の収束率であり、汎化については何も語らない。

35.3.5.3 暗黙的バイアス

損失が凸であるか否かにかかわらず多数の大域的最小点をもつとき、そのどれに到達するかはオプティマイザが決める。すなわち、オプティマイザは**暗黙的バイアス**（implicit bias）をもつ。深層学習において、これは単なる珍しい現象ではない。訓練例よりも多くの重みをもつネットワークは、何通りもの方法でデータに適合でき、そのすべてが同じ訓練損失を達成しながら未見の点では異なる予測を出すため、どの解を選ぶかがモデルの汎化性能を左右するのである。最も単純な例は、訓練例が重みより少なく $\mathbf{X}$ が行フルランクである劣決定最小二乗問題である。すべての勾配 $\mathbf{X}^\top(\mathbf{X}\mathbf{w} - \mathbf{y})$ は $\mathbf{X}$ の行空間に属するため、$\mathbf{w}_0 = \mathbf{0}$ から出発した反復点は決して行空間の外へ出ない。したがって極限は行空間内の唯一の補間解、すなわち擬似逆行列が計算する**最小ノルム**補間解となる（演習問題 4、33.5 節）。分離可能なデータに対するロジスティック回帰では損失はそもそも最小点をもたないが、反復点の方向は最大マージン分離超平面、すなわちハードマージンサポートベクターマシンの解へと収束する (Soudry et al. 2018)（演習問題 8）。どの最小点が選ばれるかは、モデル・損失・初期点・更新則に依存する。たとえば固定された対角前処理行列 $\mathbf{D}$ を用いると、$\mathbf{w}^\top\mathbf{D}^{-1}\mathbf{w}$ を $\|\mathbf{w}\|^2$ の代わりに最小化する補間解へと到達する（演習問題 7）。その

ため、座標ごとにステップサイズを変える Adam（9.5 節）は、勾配降下法が選ぶ解と同じ解を選ぶとは限らない。

本節のすべての結果は、厳密な勾配が得られ、かつ重みが $\mathbb{R}^d$ 全体を自由に動けることを仮定している。実際の訓練では平均の意味でしか正しくないミニバッチ勾配を用い（35.4 節）、また球や単体や半空間に閉じ込められた重みベクトルを扱うには 35.6 節の射影や乗数が必要となる。

35.3.6 まとめ

凸関数はその弦の下側にある。微分可能な関数ではこれはすべての接線の上側にあることと同値であり、2 回微分可能な関数ではヘッセ行列が PSD であることと同値となる。また、角においては劣勾配が勾配の役割を担う。ただ 1 つの勾配がいたるところで関数を下から抑えるため、停留点は大域的最小点となり、降下補題は勾配降下法の大域的な収束率へと転じる。すなわち、$L\|\mathbf{x}_0 - \mathbf{x}^\star\|^2/(2k)$（滑らかな凸関数 f の場合）となり、強凸性のもとでは 1 ステップあたり $1-\mu/L$ の縮小率、したがっておよそ $\kappa \log(1/\epsilon)$ ステップとなる。イェンセンの不等式は期待値に対する弦不等式であり、$D_{\mathrm{KL}} \geq 0$ を導く。4 つの閉包規則によってヒンジ損失、ℓ_1 ノルム、ロジスティック損失の凸性が保証され、log-sum-exp（およびそれに伴うソフトマックス交差エントロピー）はそのヘッセ行列が共分散となるため凸である。ニューラルネットワークの損失は一般に非凸である。しかし PL 不等式が成り立つ領域では凸性なしでも線形収束率が維持され、多数の最小点が存在する場合にはオプティマイザのダイナミクスがそのうちの 1 つを選び出す。

35.3.7 演習問題

1. **ノルムと最大値の劣微分**。定義 Equation 35.28 に基づき、以下の各劣微分を求めよ。
   1. $\partial \|\mathbf{x}\|_2$ をすべての $\mathbf{x} \in \mathbb{R}^d$（$\mathbf{x} = \mathbf{0}$ を含む）において求めよ。
   2. $f(\mathbf{x}) = \max_i x_i$ とし、$\mathcal{A}(\mathbf{x}) = \{i : x_i = f(\mathbf{x})\}$ をその有効座標の集合とする。$\partial f(\mathbf{x})$ が $\{\mathbf{e}_i : i \in \mathcal{A}(\mathbf{x})\}$ の凸包であることを示せ。最大プーリングを通じた誤差逆伝播法は、受け取った勾配を最大値をとる単一の入力へと渡すため、この凸包の 1 つの頂点 $\mathbf{e}_i$ を用いている。2 つ以上の入力が最大値で並んだとしよう。$-\mathbf{e}_i$ に沿った微小なステップでは f が減少しないことを示せ。どのような $\mathbf{g} \in \partial f(\mathbf{x})$ に対して $-\mathbf{g}$ は降下方向となるか。また、そのうち f の $-\mathbf{g}/\|\mathbf{g}\|$ に沿った方向微分を最も負にするものはどれか。
   3. (a) と最適性条件 $\mathbf{0} \in \partial(\cdot)$ を用いて、$\lambda\|\mathbf{w}\|_2 + \frac{1}{2}\|\mathbf{w}-\mathbf{z}\|^2$ を $\mathbf{w} \in \mathbb{R}^d$ について最小化せよ（これにより $\lambda\|\cdot\|_2$ の近接作用素が得られる）。解のゼロパターンを、座標ごとのソフト閾値処理 $w_i = \mathrm{sign}(z_i)\max(|z_i| - \lambda, 0)$ のゼロパターンと比較せよ。$\mathbf{w}$ が 1 つの隠れユニットの出力重みをまとめたベクトルであるとき、このペナルティはどのような働きをするか。

2. **凸結合とイェンセンの不等式**。$\sum_{i=1}^N \theta_i \mathbf{x}_i$（ただし $\theta_i \geq 0$、$\sum_i \theta_i = 1$）により N 個の点の凸結合を表す。
   1. N に関する帰納法により、凸集合 C が C の任意の点の凸結合を含むことを示せ。これは 35.3.1 節における凸包の特徴づけを完結させる包含関係である。
   2. (a) を凸関数 f のエピグラフに適用し、イェンセンの不等式の有限形式 Equation 35.29 を導け。
   3. f を 2 回微分可能で、ある区間上で $\alpha \leq f'' \leq \beta$ を満たす関数とし、その区間は確率変数 Z（平均 m、分散 s^2）のとる値をすべて含むとする。イェンセンギャップ $\mathbb{E}[f(Z)] - f(m)$ が $\alpha s^2/2$ と $\beta s^2/2$ の間に収まることを示せ。本文の 2 点の例、すなわち $Z \in \{0, 2\}$、$f(x) = x^2$（このとき $\alpha = \beta = 2$）においてこの上下界を確認せよ。さらにこの結果を $f = -\log$ に適用し、$\log \mathbb{E}[Z] - \mathbb{E}[\log Z]$ を（Z が $[a, b] \subset (0, \infty)$ に値をとるとき）上下から評価せよ。

3. **最小点の存在。** $\ell(\mathbf{w}) = \sum_i \log\!\left(1 + e^{-y_i \mathbf{w}^\top \mathbf{x}_i}\right)$ をラベル $y_i \in \{\pm 1\}$ に対するロジスティック損失とし、$\mathbf{X}$ を $\mathbf{x}_i^\top$ を行にもつ行列とする。35.3.3 節の一意性の命題によれば、狭義凸関数の最小点は高々 1 つであり、空でない内部をもつ空でない閉凸集合上の連続な強凸関数はただ 1 つの最小点をもつ。

   1. ある $\mathbf{w}$ が $y_i \mathbf{w}^\top \mathbf{x}_i > 0$ をすべての i について満たす、すなわちデータが原点を通る超平面によって線形分離可能であるとする。このとき $\inf_{\mathbf{w}} \ell(\mathbf{w}) = 0$ であり、かつ ℓ が最小点をもたないことを示せ。
   2. ℓ は、$\mathbf{X}$ が列フルランクであるとき常に狭義凸となることを示せ。これが (a) と矛盾しないのはなぜか。
   3. $\ell(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2$ が、任意の $\lambda > 0$ と任意のデータセットに対してただ 1 つの最小点をもつことを示せ。このペナルティは一意性の命題のどの仮定を補っているか。

4. **凸性を超えた PL 条件。** $f(x) = x^2 + 3\sin^2 x$ を調べるコードセルでは、その PL 定数を $\mu \approx 0.1755$（区間 $[-5, 5]$ 上）と測定した。

   1. $|f'(x)|$ を下から、$f(x)$ を上から、$|x| \geq 5$ の範囲で評価せよ。以上から、PL 不等式 Equation 35.33 が $\mathbb{R}$ 全体にわたってその区間上で測定した定数のまま成り立つことを結論せよ。
   2. $\mathbf{X} \in \mathbb{R}^{n \times d}$（$n < d$）を行フルランクとし、$f(\mathbf{w}) = \frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$ とする。f は強凸ではなく無限に多くの最小点をもつが、$\mu = \lambda_{\min}(\mathbf{X}\mathbf{X}^\top)$（$\mathbf{X}\mathbf{X}^\top$ の最小固有値）として PL 不等式を満たすことを示せ。このとき $\eta = 1/L$ の勾配降下法はどのような線形収束率を達成し、L は $\mathbf{X}\mathbf{X}^\top$ とどう関係するか。

5. [code] **距離の縮小。●** 収束率 Equation 35.32 は 1 ステップあたりの因子 $1 - \mu/L$ を $f - f^\star$ に対して保証するが、収束率検証セルでは $(1-\mu/L)^2$ が測定された。f を $\mu\mathbf{I} \preceq \nabla^2 f \preceq L\mathbf{I}$ を満たす 2 回連続微分可能な関数とし、$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)$ とする。ここでは 35.1 節の 2 つの事実、すなわち二次上界 $f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2$ と、二次関数における最良の固定ステップサイズ $\eta^\star = 2/(\mu + L)$ を用いる。

   1. $\nabla f(\mathbf{x}_k) = \bar{\mathbf{H}}_k(\mathbf{x}_k - \mathbf{x}^\star)$（ただし $\bar{\mathbf{H}}_k = \int_0^1 \nabla^2 f(\mathbf{x}^\star + s(\mathbf{x}_k - \mathbf{x}^\star))\, ds$）と表すことで、次を示せ。

$$\|\mathbf{x}_{k+1} - \mathbf{x}^\star\| \leq \max\!\left(|1 - \eta\mu|,\, |1 - \eta L|\right) \|\mathbf{x}_k - \mathbf{x}^\star\|. \tag{35.35}$$

   2. $\eta = 1/L$、$\kappa = L/\mu$ とおく。(a) と、ともに点 $\mathbf{x}^\star$ においてとった下界 Equation 35.27 および二次上界を組み合わせることで、$f(\mathbf{x}_k) - f^\star \leq \kappa\,(1 - \mu/L)^{2k}\,(f(\mathbf{x}_0) - f^\star)$ を示せ。どの k 以降でこの上界は Equation 35.32 より強くなるか。

   3. $f - f^\star$ の 1 ステップあたりの縮小率を、収束率検証セルの $\kappa = 10$ の計画行列に対して $\eta^\star = 2/11$ のもとで予測せよ。（ヒント：両方の固有成分に絶対値 $9/11$ の数が掛けられる。）そのセルのステップサイズを変更して予測を検証し、$\eta = 1/L$ のときとは異なり最初の 1 回目の比がもはや特別ではなくなる理由を説明せよ。

6. **最大値の平滑化。●** ヒンジ型損失や最悪ケース目的関数の形であるアフィン関数の最大値は、2 つの要素が同値になる場所ごとに折れ点をもつ。温度 $\tau > 0$ に対し、$\mathrm{lse}_\tau(\mathbf{z}) = \tau \,\mathrm{lse}(\mathbf{z}/\tau)$（$\mathbf{z} \in \mathbb{R}^m$）とおく。この平滑化手法は Nesterov (2005) によるものである。

   1. $\max_j z_j \leq \mathrm{lse}(\mathbf{z}) \leq \max_j z_j + \log m$ を示し、そこから $\max_j z_j \leq \mathrm{lse}_\tau(\mathbf{z}) \leq \max_j z_j + \tau \log m$ を結論せよ。

2. Equation 35.34 を用い、$\nabla^2 \mathrm{lse}_\tau(\mathbf{z})$ の最大固有値が高々 $1/(2\tau)$ であることを、$\mathrm{Var}(v_Y)$ を $\mathbb{E}[(v_Y - c)^2]$（c は適切に選んだ定数）で上から抑えることによって示せ。この上界は達成されるか。

3. $F(\mathbf{w}) = \max_j (\mathbf{A}\mathbf{w} + \mathbf{r})_j$ および $F_\tau(\mathbf{w}) = \mathrm{lse}_\tau(\mathbf{A}\mathbf{w} + \mathbf{r})$ とし、F_τ は距離 R 以内（$\mathbf{w}_0$ から測って）に最小点をもつと仮定する。(b) を用いて、F_τ が L 滑らか（ただし $L = \|\mathbf{A}\|_2^2/(2\tau)$）であり、かつ k ステップの勾配降下法を F_τ に対して $\eta = 1/L$ で実行すると次が成り立つことを示せ。

$$F(\mathbf{w}_k) - \min F \le \tau \log m + \frac{\|\mathbf{A}\|_2^2 R^2}{4\tau k}. \tag{35.36}$$

この上界を最小化するように τ を選び、精度 ϵ を保証するステップ数を求めよ。勾配降下法を誤差 $O(LR^2/k^2)$ の加速法に置き換えると、必要なステップ数はどう変わるか。

7. [code] **暗黙的バイアスとオプティマイザ。**●演習問題 4（33.5 節）では、$\frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$（ただし $\mathbf{X} \in \mathbb{R}^{n \times d}$ は $n < d$ かつ行フルランク）に対する $\mathbf{w}_0 = \mathbf{0}$ からの勾配降下法が、$\mathbf{X}$ の行空間内にとどまり最小ノルム補間解 $\mathbf{X}^+\mathbf{y}$ へ収束することを机上で示した。ここでは初期点を $\mathbf{w}_0 = \mathbf{0}$ に固定したままアルゴリズムを変える。ランダムな 4×10 の系をとり、各手法の極限とそのノルムを `np.linalg.pinv(X) @ y` と比較せよ。

   1. $\eta = 1/\lambda_{\max}(\mathbf{X}^\top\mathbf{X})$ で勾配降下法を実行し、$\mathbf{w}_k$ の行空間に直交する成分のノルムを射影行列 $\mathbf{I} - \mathbf{X}^+\mathbf{X}$ を用いて計算し追跡せよ。
   2. ヘビーボールモメンタムは $\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\nabla f(\mathbf{w}_k) + \beta(\mathbf{w}_k - \mathbf{w}_{k-1})$ と更新する。その反復点が行空間にとどまることを示し、極限を予測せよ。(a) のステップサイズと $\beta = 0.9$ を用いて予測を検証せよ。
   3. 更新を対角前処理付きステップ $\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\,\mathbf{D}\mathbf{X}^\top(\mathbf{X}\mathbf{w}_k - \mathbf{y})$ に置き換えよ。ここで $\mathbf{D}$ は成分を $[0.1, 10]$ から一様に抽出した固定対角行列であり、$\eta = 1/\lambda_{\max}(\mathbf{D}^{1/2}\mathbf{X}^\top\mathbf{X}\mathbf{D}^{1/2})$ とする。変数変換 $\mathbf{u} = \mathbf{D}^{-1/2}\mathbf{w}$ を用いて極限を閉じた式で求め、数値的に検証せよ。この極限は依然として最小ノルム補間解か。この結果は、座標ごとのステップサイズが Adam（9.5 節）のように $\mathbf{D}$ として働く適応的手法の暗黙的バイアスについて何を示唆するか。
   4. 厳密な巡回座標降下法を 1000 スイープ実行せよ。ただし $\mathbf{A} = \mathbf{X}^\top\mathbf{X}$、$\mathbf{b} = \mathbf{X}^\top\mathbf{y}$ とし、更新 $w_j \leftarrow (b_j - \sum_{i \ne j} A_{ji} w_i)/A_{jj}$ を $j = 1, \dots, d$ の順に適用する。極限がデータを補間することを確認せよ。それは最小ノルム補間解か。また、行空間の議論のどの段階が成り立たなくなるか。座標を一様ランダムに選んで同じ実行を 3 回繰り返せ。極限は乱数の抽出結果に依存するか。

8. [extended] **勾配降下法の最大マージンバイアス。**●●原点を通る超平面によって分離可能なデータ上では、ロジスティック損失は最小点をもたない（演習問題 3 の (a)）。それでも Soudry et al. (2018) は、この損失に対する勾配降下法が一定の方向へ発散することを証明している。すなわち、反復点は $\mathbf{w}_k = \hat{\mathbf{w}}\log k + \rho_k$（ただしほとんどすべてのデータセットに対して ρ_k は有界）を満たす。ここで

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\| \quad \text{subject to} \quad y_i \mathbf{w}^\top \mathbf{x}_i \ge 1 \text{ for all } i \tag{35.37}$$

はハードマージン SVM の解であり、そのマージンは $\gamma = 1/\|\hat{\mathbf{w}}\|$ である。ベクトル $\hat{\mathbf{w}}$ は $\hat{\mathbf{w}} = \sum_i \alpha_i y_i \mathbf{x}_i$（$\alpha_i \ge 0$、かつ $\alpha_i = 0$ が $y_i \hat{\mathbf{w}}^\top \mathbf{x}_i = 1$ 以外のすべての点で成り立つ）によって特徴づけられ、等号を満たす点がサポートベクターである。

1. 発散の方向を発見的に導け。ロジスティック損失をその裾 e^{-q}（ただしマージン $q = y_i \mathbf{w}^\top \mathbf{x}_i$）で置き換え、$\mathbf{w}_k = g(k)\,\mathbf{u} + O(1)$（$\|\mathbf{u}\| = 1$、$g(k) \to \infty$）と仮定する。負の勾配が漸近的に、点 $y_i \mathbf{x}_i$（最小のマージン $y_i \mathbf{u}^\top \mathbf{x}_i$ をもつもの）の非負結合になることを示せ。以上から、$\mathbf{u}$ が $\hat{\mathbf{w}}$ に比例することを結論せよ。
2. 増大の速さを導け。勾配フロー $\dot{\mathbf{w}} = -\nabla \ell(\mathbf{w})$ においてサポートベクターの項のみを残し、$\dot{g} \approx c\, e^{-\gamma g}$ がある定数 $c > 0$ に対して成り立つことを示して、$\|\mathbf{w}(t)\| \approx \|\hat{\mathbf{w}}\| \log t$ を結論せよ。
3. サポートベクターとして $\mathbf{x}_1 = (0.5, 1.5)$ と $\mathbf{x}_2 = (1.5, 0.5)$（ラベル $+1$）、それらの符号反転（ラベル -1）、さらに各クラス 6 個ずつ計 12 個のランダムな点（$y_i \hat{\mathbf{w}}^\top \mathbf{x}_i > 1$ を $\hat{\mathbf{w}} = (\tfrac{1}{2}, \tfrac{1}{2})$ に対して満たすもの）をとる。$\mathbf{X}$ をサイズ 16×2 で各行が $\mathbf{x}_i^\top$ である行列とする。ロジスティック損失の和に対する勾配降下法を $\mathbf{w}_0 = \mathbf{0}$ から $\eta = 1/\|\mathbf{X}\|_2^2$ で 10^6 ステップ実行せよ（ベクトル化した NumPy なら数秒で終わる）。対数軸にとった k に対して、ノルム $\|\mathbf{w}_k\|$、訓練損失、$\mathbf{w}_k$ と $\hat{\mathbf{w}}$ のなす角、およびマージンギャップ $\gamma - \min_i y_i \mathbf{w}_k^\top \mathbf{x}_i / \|\mathbf{w}_k\|$ をプロットせよ。さらに、$\|\mathbf{w}_k\|$ の $\log k$ に対する傾きを (b) と比較せよ。
4. 訓練誤差は最初の数ステップ以内にゼロへ到達する。$k = 10^3$ と $k = 10^6$ の間で、マージンギャップと訓練損失はそれぞれ何倍変化するか。この比較は、訓練誤差がゼロになった瞬間に訓練を打ち切ることについて何を示唆するか。

9. **対数分配関数の凸性。** この演習問題では、log-sum-exp の命題を 36.4 節の土台となる対数分配関数へ一般化する。有限集合 $\mathcal{X}$、重み $h(x) > 0$、統計量 $\mathbf{T}(x) \in \mathbb{R}^p$、および自然パラメータ $\eta \in \mathbb{R}^p$（ここではステップサイズではない）に対し、$A(\eta) = \log \sum_{x \in \mathcal{X}} h(x) \exp(\eta^\top \mathbf{T}(x))$ および $p_\eta(x) = h(x) \exp(\eta^\top \mathbf{T}(x) - A(\eta))$ とおく。

   1. $\nabla A(\eta) = \mathbb{E}_\eta[\mathbf{T}]$ および $\nabla^2 A(\eta) = \mathrm{Cov}_\eta[\mathbf{T}]$ を示せ。ただし期待値と共分散は p_η のもとでとる。
   2. A が凸であること、および狭義凸となるための必要十分条件が、いかなる非ゼロの $\mathbf{v}$ に対しても $\mathbf{v}^\top \mathbf{T}(x)$ が $\mathcal{X}$ 上で定数にならないことであると結論せよ。
   3. Equation 35.34 を $\mathcal{X} = \{1, \dots, d\}$、$\mathbf{T}(x) = \mathbf{e}_x$、$h \equiv 1$ とおくことで復元し、A が平坦になる方向を答えよ。
   4. 観測値 $x_1, \dots, x_n$ と平均統計量 $\bar{\mathbf{T}} = \frac{1}{n}\sum_i \mathbf{T}(x_i)$ に対し、平均負の対数尤度 $A(\eta) - \eta^\top \bar{\mathbf{T}} - \frac{1}{n}\sum_i \log h(x_i)$ が η に関して凸であること、および最小点が $\mathbb{E}_\eta[\mathbf{T}] = \bar{\mathbf{T}}$（モーメントマッチング）を満たすことを示せ。(c) のソフトマックスの場合において、1 度も観測されないクラスがあると最小点が存在しないのはなぜか。

35.4 確率的勾配法

ニューラルネットワークの訓練では、損失の厳密な勾配を使うことはほとんどない。更新のたびにデータセット全体を走査する代わりに、各ステップでは数百個の例からなるランダムなミニバッチ上で勾配を計算する。ここまでのすべての節では厳密な勾配を仮定しており、そこでは曲率がステップ数を決定し、凸性によってあらゆる停留点が大域的最小点となった。ミニバッチ勾配は平均的には正しい方向を指すもののノイズを含んでおり、そのノイズによって最適化の中心的な問いは「実行に何ステップ必要か」から「最小点にどこまで近づけるか」へと変化する。本節では、勾配ノイズがもたらす代償と、それを克服する方法を考察する。

考え得る最も単純な損失でさえ、その代償は明らかになる。$f(x) = \frac{1}{2}x^2$（最小点は 0）を考え、すべての勾配に単位分散の独立なノイズを加えながら、大きさ $\eta = 0.1$ の定数ステップをとるとしよう。厳密な勾配であれば、各ステップで x は 10 分の 1 だけ縮小し、反復点を望むだけ 0 に近づけることができる。しかしノイズを含む勾配では、反復点は決して落ち着かない。その二乗平均は $0.1/1.9 \approx 0.053$ に収束するため、何ステップ進めようとも最小点から約 0.23 離れたあ

たりをさまよい続ける。損失をどこまで下げられるかは、ノイズの大きさ、ステップサイズ、そして許容できるステップ数という 3 つの量で決まる。

まず、ミニバッチ勾配のノイズを定量化する。次に二次目的関数上で、固定ステップサイズがどこで頭打ちになるか、またそこに到達するまでに何ステップかかるかを計算する。続いて、減衰するステップサイズがその頭打ちをどう解消するか、そして $1/k$ スケジュールがどのような定数を必要とするかを示したのち、固定された予算に合わせて調整した定数ステップが凸損失と非凸損失の両方で達成する誤差の上界を導く。最後に、有限個の訓練セット上で保存しておいたスナップショット勾配を用いてノイズを打ち消す方法を示す。本節全体を通じて念頭に置くべき要点は 3 つある。ノイズフロアはステップサイズに比例すること、$1/k$ スケジュールがその収束率を達成するには十分大きな定数が必要であること、そして K ステップの固定予算では $1/\sqrt{K}$ に比例するステップサイズが求められることである。分散低減（variance reduction）を扱う最後の小節は、初読時には飛ばしても構わない。

35.4.1 ミニバッチ勾配

35.4.1.1 全勾配のコスト

典型的な訓練目的関数は、n 個の例の損失を平均したものである。

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}),$$

ここで、f_i は例 i における損失であり、$\mathbf{x} \in \mathbb{R}^d$ はモデルパラメータを保持する。その勾配 $\nabla f = \frac{1}{n}\sum_i \nabla f_i$ も平均であるため、評価には n 個すべての例を参照する必要がある。$n = 10^9$ のとき、勾配降下法の 1 ステップにはデータセット全体の 1 パスが必要となり、しかも勾配降下法は通常数千ステップを要する。本章の通し例である $\frac{1}{2}(x_1^2 + 1000\,x_2^2)$（曲率が 1000 倍異なる）でさえ、最適な固定ステップで誤差を 100 万分の 1 に縮小するには約 7,000 回の反復が必要となる（35.1 節）。**確率的勾配降下法**（stochastic gradient descent、SGD）は、全体の平均をランダムな**ミニバッチ** $\mathcal{B}$（b 個のインデックスからなる）上の平均に置き換えることで、このコストを回避する。

$$\widehat{\mathbf{g}}_b(\mathbf{x}) = \frac{1}{b}\sum_{i \in \mathcal{B}} \nabla f_i(\mathbf{x}), \qquad \mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k\, \widehat{\mathbf{g}}_b(\mathbf{x}_k). \tag{35.38}$$

これにより、各ステップでは b 個の例だけを調べればよく、n 個すべてを調べる必要はなくなる。サイズ $b = 256$ のミニバッチは、10^9 個の例全体を 1 パス走査する場合と比べて約 400 万倍安価である。本節を通して k はオプティマイザのステップを表し、ステップサイズ η_k は k に応じて変化してよいものとする。解析にあたっては、毎ステップ新たなミニバッチを抽出し、その b 個のインデックスは $\{1, \dots, n\}$ 上で独立かつ一様であるとする（復元サンプリング）。実際の訓練ループでは通常、データセットをシャッフルして非復元サンプリングを行う。この場合、以下で導出する勾配の分散に $(n-b)/(n-1)$ が掛かるが、この係数は $b \ll n$ である限り 1 に近い（9.2 節）。

35.4.1.2 不偏性と $1/b$ 則

ミニバッチ勾配はどの程度正確だろうか。その質は 2 つの量で捉えられる。1 つは平均であり、平均的に正しい方向を指しているかを表す。もう 1 つは真の勾配からの平均二乗距離であり、これ

を**ノイズパワー**と呼ぶ。このノイズパワーは、個々の例の勾配がその平均の周りにどれだけ広く散らばっているかに依存し、共分散行列によって測られる。

命題（ミニバッチ勾配は不偏であり、ノイズパワーは $\operatorname{tr}\Sigma/b$ である）。$\mathbf{x}$ を固定し、次式とおく。

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} \big(\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\big)\big(\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\big)^{\top}$$

これは一様に抽出された 1 つの例の勾配の共分散である。b 個からなる $\mathcal{B}$ のインデックスが $\{1,\dots,n\}$ 上で独立かつ一様であるならば、次が成り立つ。

$$\mathbb{E}\big[\hat{\mathbf{g}}_b(\mathbf{x})\big] = \nabla f(\mathbf{x}), \qquad \mathbb{E}\big[\|\hat{\mathbf{g}}_b(\mathbf{x}) - \nabla f(\mathbf{x})\|^2\big] = \frac{\operatorname{tr}\Sigma}{b}. \tag{35.39}$$

証明。一様な単一のインデックス i に対して、$\mathbb{E}[\nabla f_i(\mathbf{x})] = \frac{1}{n}\sum_i \nabla f_i(\mathbf{x}) = \nabla f(\mathbf{x})$ である。したがって、ミニバッチ内の各項は平均 $\nabla f(\mathbf{x})$ を持ち、その平均も同様である。b 個のインデックスは独立であり、各項は共分散 Σ を持つため、その和の共分散は $b\Sigma$ となり、b で割ると共分散が $1/b^2$ 倍されて Σ/b となる。最後に、平均ゼロの任意の確率ベクトルの期待二乗ノルムは、その共分散行列のトレースに等しい。$\blacksquare$

最適化の過程では、$\mathbf{x}$ は現在の反復点 $\mathbf{x}_k$ であり、ミニバッチは過去のステップとは独立に抽出されるため、両方の等式は $\mathbf{x}_k$ を条件として成り立つ。なお、不偏性は一様サンプリングに依存する。サンプラが例 i を確率 p_i で抽出する場合、推定量を不偏に保つには各勾配に $1/(np_i)$ の重みを掛け直す必要がある。

ノイズパワーは $1/b$ のオーダーで減衰するため、その二乗平均平方根振幅は $1/\sqrt{b}$ の速さでしか減衰しない。つまり、勾配ノイズを 10 分の 1 に減らすには、1 ステップあたり 100 倍の例が必要になる。一方、真の勾配信号の二乗ノルムはバッチサイズによらず $\|\nabla f\|^2$ である。信号とノイズのパワーが等しくなるのはバッチサイズが $b_{\text{noise}} = \operatorname{tr}\Sigma/\|\nabla f\|^2$ のときであり、本編ではこれを**勾配ノイズスケール**（gradient-noise scale）と呼んでいる（9.8 節）。演習問題 1 では、曲率で重み付けした対応量を導出する。

具体的なデータ上でこのノイズがどれほど大きいかを確認するため、以下のステップサイズ実験でも再利用する小規模なロジスティック回帰問題を設定しよう。この問題は、標準正規分布に従う特徴量を持つ $n = 200$ 個の例 $\mathbf{a}_i \in \mathbb{R}^5$ と、ラベル $y_i \in \{-1, +1\}$（真の重み $(1, -2, 1.5, 0, 2)$ を持つロジスティックモデルから生成）からなり、正則化付き損失は次式で与えられる。

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \log\!\left(1 + e^{-y_i \mathbf{a}_i^{\top}\mathbf{x}}\right) + \frac{0.1}{2}\|\mathbf{x}\|^2.$$

二次のペナルティ項により、f は $\mu \ge 0.1$ の強凸関数となる。例 i の勾配は $\nabla f_i(\mathbf{x}) = -y_i s_i \mathbf{a}_i + 0.1\,\mathbf{x}$ であり、ここで $s_i = 1/(1 + e^{y_i \mathbf{a}_i^{\top}\mathbf{x}})$ はモデルが誤ったラベルに割り当てる確率である。以下のセルでは、データセットを生成して損失を定義するとともに、全 n 個の例ごとの勾配を $n \times d$ 行列の各行として返すベクトル化された補助関数を定義する。

```
n, d, reg = 200, 5, 0.1                          # examples, parameters, l2 weight
x_true = np.array([1.0, -2.0, 1.5, 0.0, 2.0])
rng = np.random.default_rng()
A = rng.normal(size=(n, d))                       # row i: the features a_i
```

```
y = np.where(rng.random(n) < 1 / (1 + np.exp(-A @ x_true)), 1.0, -1.0)

def loss(x, A, y, reg):
    """The regularized logistic loss f(x)."""
    return np.mean(np.log1p(np.exp(-y * (A @ x)))) + 0.5 * reg * x @ x

def example_grads(x, A, y, reg):
    """The n x d matrix whose row i is the gradient of f_i at x."""
    s = 1 / (1 + np.exp(y * (A @ x)))            # probability of the wrong label
    return -(y * s)[:, None] * A + reg * x
```

上の命題は厳密に成り立つため、シミュレーションは主に検算と、この問題における $\operatorname{tr}\Sigma$ の実際のスケールを示す役割を果たす。以下では、各サイズにつき $10{,}000$ 個のランダムなミニバッチを $\mathbf{x} = \mathbf{0}$ で評価し、経験誤差を $\operatorname{tr}\Sigma/b$ と比較する。

```
G = example_grads(np.zeros(d), A, y, reg)        # per-example gradients at x = 0
g = G.mean(axis=0)                                # the full gradient
tr_sigma = ((G - g)**2).sum(axis=1).mean()
print(f'tr Sigma = {tr_sigma:.3f}')
for b in [1, 4, 16, 64]:
    idx = rng.integers(0, n, size=(10000, b))    # minibatches, with replacement
    err = ((G[idx].mean(axis=1) - g)**2).sum(axis=1).mean()
    print(f'b = {b:2d}: measured {err:.4f}, tr Sigma / b = {tr_sigma / b:.4f}, '
          f'ratio {err * b / tr_sigma:.3f}')
```

```
tr Sigma = 1.131
b =  1: measured 1.1188, tr Sigma / b = 1.1311, ratio 0.989
b =  4: measured 0.2847, tr Sigma / b = 0.2828, ratio 1.007
b = 16: measured 0.0699, tr Sigma / b = 0.0707, ratio 0.989
b = 64: measured 0.0177, tr Sigma / b = 0.0177, ratio 1.003
```

1 例あたりのノイズパワー $\operatorname{tr}\Sigma$ は約 1.1 であり、4 つのバッチサイズすべてにわたって、実測誤差はサンプリング変動による数パーセント以内の精度で $\operatorname{tr}\Sigma/b$ と一致している。1 ステップのノイズがわかったところで、そのノイズが最適化の実行全体にわたってどのような影響を及ぼすかを問うことができる。

35.4.1.3 ノイズフロア

勾配ノイズは多数のステップにわたって平均化されて消えるのか、それとも蓄積するのか。9.2 節では、SGD が谷の内部で落ち着く点群を測定し、その半径を導出した。ここでは軌跡全体を閉じた形で解くことで、その点群に到達するまでに何ステップかかるかも明らかにする。最も単純な場合である二次関数の 1 座標 $f(x) = \tfrac{1}{2}\lambda x^2$（曲率 $\lambda > 0$、最小点 0）に、すでにその仕組みのすべてが含まれている。確率的勾配を、真の勾配 λx_k にランダムなノイズ ξ_k（x_k を与えたときの条件付き平均がゼロで分散が σ^2）を加えたものとしてモデル化しよう。ここで σ^2 は Σ/b の対角成分の役割を果たすため、σ^2 に比例するすべての量は $1/b$ としてスケールする。SGD の 1 ステップは次のようになる。

$$x_{k+1} = x_k - \eta\,(\lambda x_k + \xi_k) = \rho\, x_k - \eta\,\xi_k, \qquad \rho = 1 - \eta\lambda.$$

各ステップにおいて、決定論的な部分は厳密な勾配降下法と同様に反復点を係数 ρ で最小点へと引き寄せ、確率的な部分はランダムな揺さぶり $-\eta\xi_k$ を与える。反復点の大きさを追跡する

ため、その二次モーメント $e_k = \mathbb{E}[x_k^2]$ に着目する。更新式を二乗して期待値をとると、交差項 $-2\rho\eta\,\mathbb{E}[x_k \xi_k]$ は ξ_k が x_k を条件として平均ゼロを持つため消え、次式が残る。

$$e_{k+1} = \rho^2 e_k + \eta^2 \sigma^2 .$$

各ステップで二次モーメントは ρ^2 倍に縮小し、新たなノイズパワー $\eta^2\sigma^2$ が注入される。本節のほぼすべての結果は、このたった 1 つの漸化式から読み取ることができる。定数の駆動項を持つ線形漸化式は、不動点 V（すなわち $V = \rho^2 V + \eta^2\sigma^2$）を求めることで解ける。この不動点方程式を元の漸化式から引くと、剰余 $e_k - V$ が毎ステップ ρ^2 倍に縮小することがわかり、次式が得られる。

$$e_k = V + \rho^{2k}\,(e_0 - V) \tag{35.40}$$

ここで、**ノイズフロア**は次式で与えられる。

$$V = \frac{\eta^2\sigma^2}{1-\rho^2} = \frac{\eta\,\sigma^2}{\lambda\,(2-\eta\lambda)} \;\approx\; \frac{\eta\,\sigma^2}{2\lambda} \qquad (\eta\lambda \ll 1). \tag{35.41}$$

ノイズフロアはステップサイズおよびノイズ分散に対して線形に増大し、曲率に反比例して縮小する。いずれの式も、二次関数上の厳密な勾配降下法と同じ安定性の上限 $0 < \eta < 2/\lambda$ を必要とする。これを超えると、ノイズがなくても二次モーメントは発散する。

軌跡は 2 つの段階にきれいに分かれる。過渡項 $\rho^{2k}(e_0 - V)$ は決定論的な勾配降下法と同様に幾何級数的に減衰するのに対し、フロア V は純粋にノイズの影響によるものである。最小点から遠い領域では過渡項が支配的であり、SGD は勾配降下法とほぼ同様に振る舞う。最小点に近づくと、内側へ引き戻す力 $\eta\lambda x$ は弱まる一方でランダムな揺さぶりは同じ分散 $\eta^2\sigma^2$ を保ち続け、やがて両者は $e = V$ で釣り合う。それ以降、反復点は決して収束せず、その分布は最小点の周りの定常的な**ノイズ球**（noise ball）へと落ち着く。このノイズ球の二乗平均平方根半径はおよそ $\sqrt{\eta\sigma^2/(2\lambda)}$ であり、$\sqrt{\eta}$ に比例してスケールする。Figure 35.15 に、2 次元の二次関数ボウルにおける両段階の様子を示す。

実用上関心があるのは、二乗距離よりも期待損失である。$\mathbb{E}f(x_k) = \frac{1}{2}\lambda e_k$ であるから、損失の下限は次のようになる。

$$\tfrac{1}{2}\lambda V = \frac{\eta\,\sigma^2}{2\,(2-\eta\lambda)} \;\approx\; \frac{\eta\,\sigma^2}{4}.$$

驚くべきことに、ステップサイズが小さいとき曲率 λ は完全に相殺される。平坦な方向ほど反復点は遠くまでさまようが、その倍率と平坦さによって損失が割り引かれる倍率がちょうど一致するのである。

d 次元の二次関数上では、ヘッセ行列の固有ベクトルに沿った各座標がこのスカラー漸化式の d 個の独立なコピーへと分離し、それぞれが固有値 λ_i とノイズ分散 σ_i^2 を持つ。したがって、全体の期待損失の下限はその和 $\sum_i \eta\sigma_i^2/(2(2-\eta\lambda_i))$ となる。σ_i^2 は固有基底における Σ/b の対角成分であるため、その総和は $\operatorname{tr}\Sigma/b$ に等しい。すべての $\eta\lambda_i \ll 1$ が成り立つとき、全体の損失の下限はおよそ $\eta\operatorname{tr}\Sigma/(4b)$ へと簡約され、ステップサイズとバッチサイズにはその比 η/b を通じてのみ依存する。ゆえに、ステップサイズを半分にすることとバッチサイズを 2 倍にすることは同じ倍率だけ下限を引き下げる。これが 9.8 節の大バッチ学習率則の背後にある結びつきである。

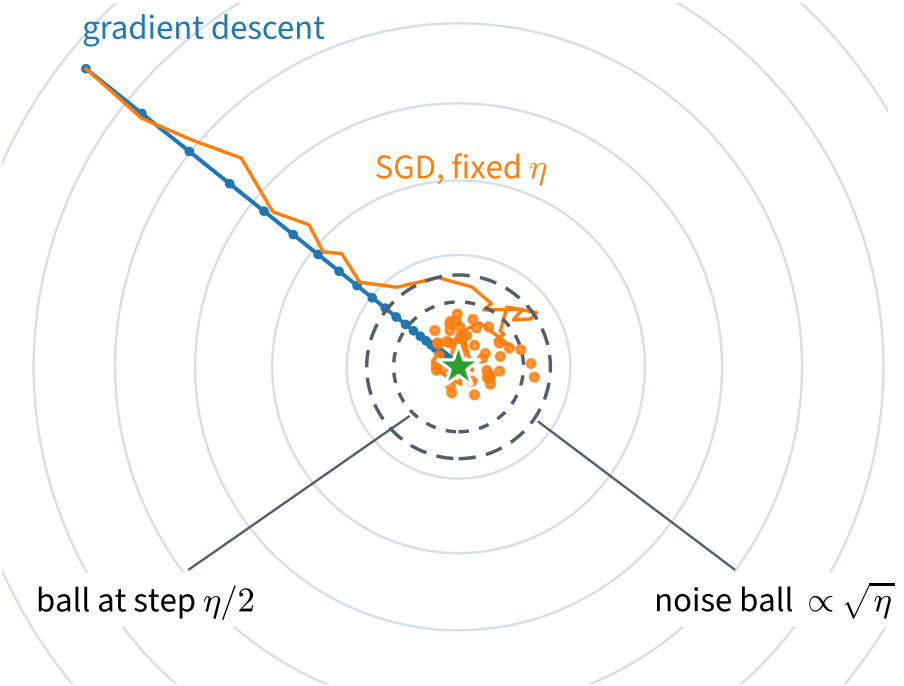


Figure 35.15: $\frac{1}{2}\|\mathbf{x}\|^2$ 上での $\eta = 0.15$、標準偏差 0.6 の勾配ノイズにおける勾配降下法と固定ステップ SGD。フルバッチの経路は最小点に収束するのに対し、SGD は勾配降下法のように降下したのちノイズ球の内部で変動し続ける。破線の円は、このステップサイズと半分のステップサイズにおけるノイズ球を表す。散らばりを囲むよう座標ごとの二乗平均平方根距離 $\sqrt{\eta\sigma^2/(2\lambda)}$ の 3 倍で描かれた両円の半径は、$\sqrt{2}$ 倍異なる。

実行はどれほど速くフロアに到達するだろうか。冒頭の例と同じく $\lambda = \sigma = 1$ としよう。そこでは $\eta = 0.1$ のとき $V = 0.053$ であった。ステップを $\eta = 0.3$ に引き上げると、フロアは $V = 0.176$（3 倍強の高さで、損失の下限は 0.088）に上昇する。$x_0 = 3$（$e_0 = 9$）から出発すると、過渡項 $e_k - V$ が V の 10 分の 1 未満に落ちるまでに 36 ステップを要する（$\eta = 0.1$、$\rho^2 = 0.81$ の場合）のに対し、わずか 9 ステップで済むのが $\eta = 0.3$（$\rho^2 = 0.49$）の場合である。ステップサイズを 3 分の 1 にするとフロアもおよそ 3 分の 1 に下がるが、そこに到達するには 4 倍のステップ数が必要になる。本節の残りで扱う手法はすべて、この速度と精度のトレードオフを乗り切るための異なるアプローチである。

ここで通し例の $\tfrac{1}{2}(x_1^2 + 1000\,x_2^2)$ に戻り、毎ステップ、分散 $\sigma^2 = 1$ で平均ゼロの独立なノイズを勾配 $(x_1, 1000\,x_2)$ の各座標に加えることでノイズを直接モデル化しよう。2 つの座標は $\lambda_1 = 1$ および $\lambda_2 = 1000$ のスカラー漸化式に従う。$\eta = 1/L = 10^{-3}$、すなわち安定性の上限 $2/1000$ の半分に設定する。硬い座標では $\rho_2 = 0$ となり、わずか 1 ステップで過渡項が完全に消え、フロア $V_2 = 10^{-3}/1000 = 10^{-6}$ 上に残る。遅い座標は $\rho_1 = 0.999$ とフロア $V_1 = 10^{-3}/1.999 = 5.0 \times 10^{-4}$ を持ち、その過渡項が 1,000 分の 1 に縮小するまでには $\log 1000 / (-2 \log 0.999) \approx 3{,}450$ ステップを要する。全体の損失の下限は 7.5×10^{-4} であり、その内訳は遅い座標からの 2.5×10^{-4} と硬い座標からの 5×10^{-4} である。皮肉な結果に注目してほしい。勾配降下法が 1 ステップで除去してしまう硬い座標が、損失の下限の 3 分の 2 を占めているのである。というのも、$\eta\lambda_2 = 1$ におけるその損失の下限は $\eta\sigma^2/2$ となり、小ステップ時の値 $\eta\sigma^2/4$ の 2 倍に達するからである。悪条件性は平坦な方向を減速させ、ノイズは両方向に沿って損失を押し上げる。

閉じた形で求めたこのフロアが厳密に成り立つのは、加法ノイズを伴う二次関数の場合のみである。次節では、「フロアが η に比例する」というスケーリングが非二次関数であるロジスティック損失でも成り立つかを検証し、ステップサイズの減衰がどのようにそのフロアを突き破るかを見ていく。

35.4.2 ステップサイズの減衰

35.4.2.1 Robbins–Monro 条件

ノイズフロアは η に対して線形にスケールするため、最小点に実際に到達するにはステップサイズをゼロに向けて縮小する必要がある。しかし、縮小が速すぎると反復点は目標の手前で失速してしまう。単位サイズの勾配のもとでは、移動できる総距離は $\sum_k \eta_k$ を超えられないからである。Robbins and Monro (1951) の古典的な条件は、これら両方の要件を同時に捉えている。

$$\sum_{k} \eta_k = \infty, \qquad \sum_{k} \eta_k^2 < \infty. \tag{35.42}$$

第 1 の条件は反復点が出発点からいくらでも遠くまで移動できることを保証し、第 2 の条件はノイズによって注入される累積分散 $\sigma^2 \sum_k \eta_k^2$ を有限に抑える。スケジュール $\eta_k = c/k$ は両方の条件を満たすが、$\eta_k = c/\sqrt{k}$ と定数ステップは第 1 の条件しか満たさない。最小点までの距離に対して分散が高々二次でしか増大しない不偏勾配と組み合わせることで、これら 2 つの条件は任意の強凸損失において SGD が最小点へ収束することを保証する (Robbins and Monro 1951; Bottou et al. 2018)。

スカラー二次関数上では、平均二乗収束に必要な条件は実際にはこれより弱い。漸化式を展開すると、$e_k \to 0$ となるには $\eta_k \to 0$ かつ $\sum_k \eta_k = \infty$（ただし $\eta_k < 2/\lambda$）であれば十分だとわかる。特に $\eta_k = c/\sqrt{k}$ とすると $e_k \approx c\sigma^2/(2\lambda\sqrt{k})$ となり、これは単に各時点のステップサイズに対応する瞬時的なフロアを追跡しているにすぎない。より強い二乗総和可能条件は、一般の

定理においてほとんどすべての個別の標本経路に沿った収束を保証するために必要となるものである。

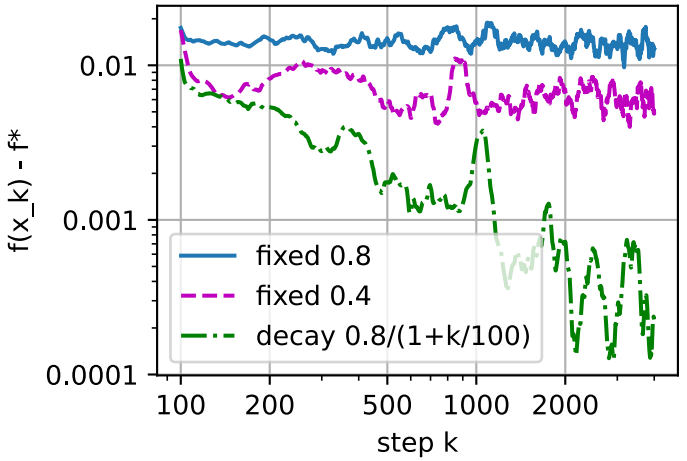
非二次目的関数については 2 つの疑問が残る。ノイズフロアは依然として η に対して線形にスケールするのか、そして減衰スケジュールはそのフロアを下回ることができるのかである。先のロジスティック回帰問題で確かめるため、まず基準となる最適値 $f^\star$ を $4{,}000$ 回の全勾配ステップで計算し、続いて毎ステップの準最適性ギャップ（suboptimality gap）$f(\mathbf{x}_k) - f^\star$ を記録するミニバッチ SGD ループを定義する。

```
x_star = np.zeros(d)
for _ in range(4000):                              # full-gradient reference optimum
    x_star -= 0.5 * example_grads(x_star, A, y, reg).mean(axis=0)
f_star = loss(x_star, A, y, reg)

def sgd(schedule, A, y, reg, f_star, rng, steps=4000, b=8):
    """Minibatch SGD from x = 0; returns the gap f(x_k) - f* after every step."""
    x, gaps = np.zeros(A.shape[1]), np.empty(steps)
    for k in range(steps):
        idx = rng.integers(0, len(y), size=b)
        x -= schedule(k) * example_grads(x, A[idx], y[idx], reg).mean(axis=0)
        gaps[k] = loss(x, A, y, reg) - f_star
    return gaps
```

二次モデルが信頼できる指針であるならば、固定ステップサイズを半分にすれば SGD が頭打ちになるギャップも半分になるはずである。ここでは $4{,}000$ ステップの訓練を $b = 8$ で実行し、定数ステップ $\eta = 0.8$ および $\eta = 0.4$ と、減衰スケジュール $\eta_k = 80/(k + 100) = 0.8/(1 + k/100)$（初期値は 0.8）を比較する。以下のセルでは、各実行の最後の $1{,}000$ ステップにわたる平均ギャップを表示してそれらの比を報告し、100 ステップ移動平均を両対数軸上にプロットする。

```
rules = {'fixed 0.8': lambda k: 0.8, 'fixed 0.4': lambda k: 0.4,
         'decay 0.8/(1+k/100)': lambda k: 0.8 / (1 + k / 100)}
gaps = {name: sgd(s, A, y, reg, f_star, rng) for name, s in rules.items()}
tail = [gap[-1000:].mean() for gap in gaps.values()]
print('mean gap, last 1000 steps: ' + ', '.join(f'{t:.1e}' for t in tail))
print(f'floor(0.8) / floor(0.4) = {tail[0] / tail[1]:.2f}, '
      f'floor(0.8) / decay = {tail[0] / tail[2]:.0f}')
window = np.ones(100) / 100                        # plot 100-step moving averages
d2l.plot(np.arange(100, 4001), [np.convolve(gap, window, 'valid') for gap in gaps.values()],
         'step k', 'f(x_k) - f*', xscale='log', yscale='log', legend=list(gaps))
```

```
mean gap, last 1000 steps: 1.4e-02, 5.9e-03, 4.1e-04
floor(0.8) / floor(0.4) = 2.35, floor(0.8) / decay = 34
```



固定ステップの両実行はいずれも頭打ちになる。$\eta = 0.8$ では、最後の 1,000 ステップの平均ギャップは $1\text{〜}2 \times 10^{-2}$ 付近に落ち着き、ステップサイズを半分にするとギャップもおよそ半分になる。繰り返し実行すると、その比は約 $1.7\text{〜}2.8$ の間で変動し、ロジスティック損失が二次関数でないにもかかわらず二次モデルが予測する 2 倍を中心として分布する。一方、減衰スケジュールはおよそ 20〜90 倍も $\eta = 0.8$ のフロアより低い値に到達し、両対数プロット上の曲線はミニバッチによる揺らぎを伴いながらもなお下降し続けている。序盤の大きなステップが最小点までの距離を素早く縮め、終盤の小さなステップがノイズ球を縮小させるのである。

35.4.2.2 $1/k$ スケジュールにおける定数

スケジュール $\eta_k = c/(k + k_0)$ は任意の $c > 0$ に対して Robbins–Monro 条件の両方を満たし、シフト k_0 によって初期ステップは安定性の上限より十分低く保たれる。では、どのような c を選んでも $O(1/k)$ の収束率が達成されるのだろうか。曲率 $\lambda = \mu$ のスカラー二次関数がその厳密な答えを与える。$t = k + k_0$ と書くと、二次モーメントの漸化式は次のようになる。

$$e_{k+1} = \left(1 - \frac{c\mu}{t}\right)^2 e_k + \frac{c^2 \sigma^2}{t^2}.$$

ここでは縮小が $1/t$ の速さで弱まる一方、注入されるノイズパワーは $1/t^2$ の速さで減少する。どちらの効果が支配的になるかは c に決定的に依存する。c が小さすぎると、反復点が最小点へ向かって縮小するよりも速くステップサイズが縮んでしまい、ノイズではなくスケジュール自体がボトルネックとなる。

命題（$1/k$ スケジュールの定数）。 **曲率 μ のスカラーモデルにおいてステップサイズを $\eta_k = c/(k + k_0)$ とするとき、$2c\mu > 1$ ならば $k\, e_k \to A$ となる。ただし、定数は次式で与えられる。**

$$A = \frac{c^2 \sigma^2}{2c\mu - 1}. \tag{35.43}$$

$2c\mu < 1$ のとき、e_k は $k^{-2c\mu}$ のオーダーで減衰し、これは $1/k$ より真に遅い。また境界 $2c\mu = 1$ では $\log k / k$ のオーダーで減衰する。任意の $c > 0$ に対して $e_k \to 0$ が成り立つ。

ここでは完全な証明は省略し、主要項の係数比較によって定数 A を導出するとともに、残る 2 つの領域の概略を直感的に説明するにとどめる。演習問題 2 では、しきい値より上では帰納法により、下では漸化式の展開によって完全な証明を与える。

定数の導出。漸近形 $e_k \approx A/t$ を仮定してみよう。漸化式の左辺は $A/(t+1) \approx A/t - A/t^2$ となる。右辺では $(1-c\mu/t)^2 \approx 1 - 2c\mu/t$ と展開することで、$A/t - 2c\mu A/t^2 + c^2\sigma^2/t^2$（$t^{-3}$ 以上の高次項を除く）が得られる。$1/t$ の項は自動的に一致し、$1/t^2$ の係数を等しいとおくと $-A = -2c\mu A + c^2\sigma^2$、すなわち $A(2c\mu - 1) = c^2\sigma^2$ が得られる。左辺の微妙な項に注目してほしい。t から $t+1$ へ進むことで生じる $-A/t^2$ は縮小項と同じオーダーであり、これがしきい値 $2c\mu > 1$ を生み出している。このシフトを無視すると、$A = c\sigma^2/(2\mu)$ がすべての $c > 0$ で成り立つという誤った予測になってしまう。

しきい値未満（直感的な説明）。$2c\mu < 1$ のとき、初期誤差と序盤のノイズの双方の減衰率は累積縮小によって決まる。$\log(1-u) \approx -u$ および $\sum_{j \le t} 1/j \approx \log t$ より、縮小率の積は $\prod_j (1 - c\mu/(j+k_0))^2 \approx (k_0/t)^{2c\mu}$ のように振る舞い、$1/t$ よりも遅く減衰する。境界 $2c\mu = 1$ では、ステップ j で注入されたノイズがステップ t においておよそ $c^2\sigma^2/(jt)$ の重みで生き残り、j について和をとることで余分な $\log k$ の因子が生じる。

この感度を具体的な数値で確認するため、$\mu = \sigma = 1$ としよう。$c = 1$ とすると $A = 1$ となり、$e_0 = 1$、$k_0 = 10$ から漸化式を反復すると $k\,e_k = 1.0001$（$k = 10^6$ 時点）が得られる。$c = 2$ に倍増させても定数の増加は緩やかで、$A = 4/3$ にとどまる。一方、しきい値を下回ると被害ははるかに大きい。$c = 0.25$ では両対数傾きが -0.50（$k = 10^5 \sim 10^6$ の区間）となり、$c = 0.1$ では傾きがわずか -0.20 になる。A を c について最小化すると最適定数 $c = 1/\mu$ が得られ、このとき $A = \sigma^2/\mu^2$、$e_k \approx \sigma^2/(\mu^2 k)$ となる。注目すべきことに、定数ステップ実行の反復点を平均化すれば、ステップサイズを一切減衰させずにこれとまったく同じ最適定数を達成できる（演習問題 5）。$c = 1/(2\mu)$ における鋭いしきい値ゆえに、$1/k$ スケジュールは実用上脆いものとなる。最小曲率 μ が既知であることはまれであり、c を過小に見積もると収束率そのものが犠牲になるからである。この脆さこそ、実用的な学習率スケジュールが $1/k$ よりも緩やかに減衰する主要な理由の 1 つである。

通し例の $\frac{1}{2}(x_1^2 + 1000\,x_2^2)$ では、シフト k_0 が第 2 の困難を生む。最初のステップにおける安定性から $c/k_0 < 2/L$、すなわち $k_0 > cL/2$ が要求される。したがって遅い座標に対する最適定数 $c = 1/\mu = 1$ を選ぶと、$k_0 > 500$ が強いられる。$k_0 = 1000$、$\mathbf{x}_0 = (1, 1)$ とすると、遅い座標の過渡項は初期誤差のうち $(k_0/(k+k_0))^2 \approx 0.008$ の割合を $k = 10^4$ ステップ後にも残しており、期待損失は 4.2×10^{-3} にとどまる。これは定数ステップの下限 7.5×10^{-4} よりも 5 倍以上悪い。硬い座標が要求する小さな初期ステップが遅い座標の前進を阻み、条件数が k_0 の内部に再び現れるのである。

35.4.2.3 厳密な漸化式のもとでのスケジュールの形状

現代の訓練で $1/k$ 減衰が使われることはまれであり、実務ではコサイン減衰やウォームアップ・安定・減衰が好まれる（ニューラルネットワーク上での比較は 9.7 節を参照）。深層ネットワーク上でこれらのスケジュールに順位をつける一般定理は存在しないものの、二次モデルを用いれば、サンプリングもランダムノイズも一切なしに**任意の**スケジュールの期待損失を厳密に計算できる。先に導出した二次モーメントの漸化式は η_k が変化してもステップごとに成り立つため、単に次式が得られる。

$$e_{k+1} = (1 - \eta_k \lambda)^2 e_k + \eta_k^2 \sigma^2, \qquad \mathbb{E} f(x_k) = \tfrac{1}{2}\lambda e_k.$$

この更新を反復すれば、K ステップの予算にわたる厳密な期待損失曲線を K 回の乗算で計算できる。ここでは同じ予算 K にわたって 4 つの形状を比較する（Figure 35.16）。すなわち、定数ステップ η_0、$1/k$ スケジュール $c/(k+k_0)$（$c = \eta_0 k_0$ として初期値を η_0 に揃える）、**コサイン減衰**（cosine decay）(Loshchilov and Hutter 2016)、そして**ウォームアップ・安定・減衰**

(warmup–stable–decay、WSD) (Hu et al. 2024) であり、WSD は η_0 を予算の最初の 80% にわたって一定に保ち、その後ゼロまで線形に減衰させる。

$$\eta_k^{\cos} = \tfrac{1}{2}\eta_0\Big(1+\cos\frac{\pi k}{K}\Big),$$
$$\eta_k^{\mathrm{WSD}} = \begin{cases} \eta_0, & k < 0.8K, \\ \eta_0\,\dfrac{K-k}{0.2K}, & k \geq 0.8K. \end{cases}$$

コサイン減衰と WSD はどちらも η_0 から始まりゼロで終わるが、コサイン減衰が実行全体を通じて連続的に減衰するのに対し、WSD は最後の 5 分の 1 まで減衰を一切遅らせる。冒頭の短いウォームアップ区間（35.5.4 節で議論する）は、二次モデルがステップ 0 からすでに η_0 で安定しているため省略する。WSD が大規模な事前学習で広く使われるのには運用上の理由がある。定数プラトー上のどのチェックポイントからでも任意の終端ステップに向けて減衰させられるため、総予算をあらかじめ確定しておく必要がないのである。この二次モデルが評価するのは最終的な期待損失であり、そうした柔軟性ではない。

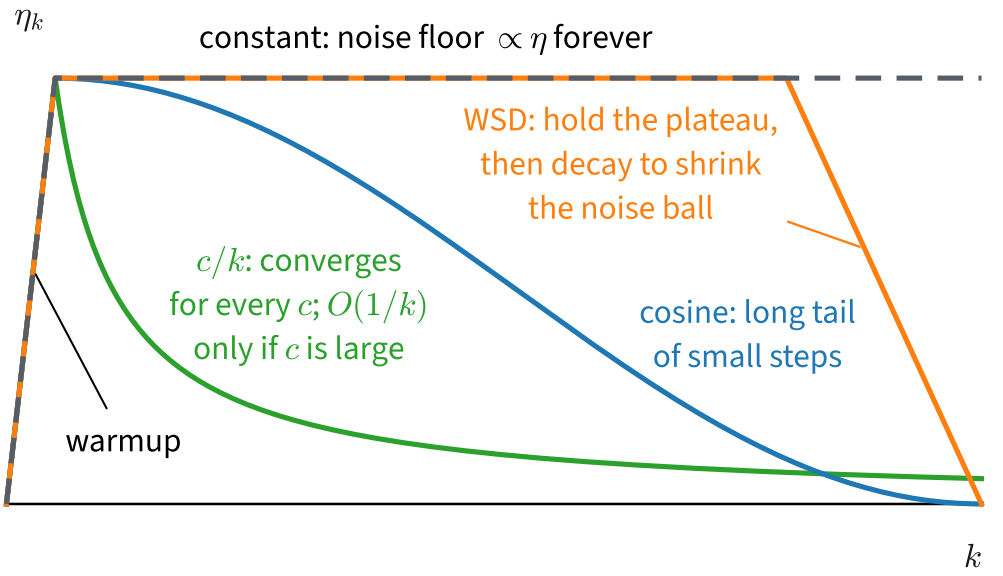
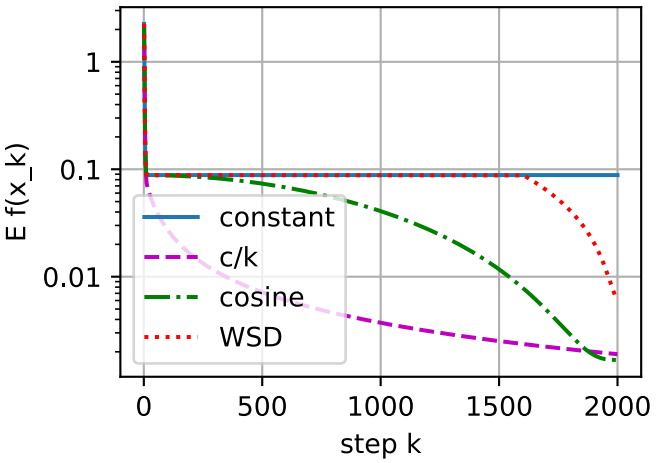


Figure 35.16: 1 つの予算 K にわたるスケジュールの形状。定数、$1/k$ 減衰（図中ラベルは c/k）、コサイン減衰、およびウォームアップ・安定・減衰（WSD）であり、いずれも以下の計算では省略する短い線形ウォームアップを冒頭に伴う。図はステップサイズの系列をプロットしたものであり、それぞれが二次モデル上で生み出す期待損失は以下で計算する。

これら 4 つのスケジュールのうち、最も低い最終損失に到達するのはどれだろうか。数値を計算する前に予想してみてほしい。以下のセルでは、$\lambda = \sigma = 1$、$K = 2000$、$\eta_0 = 0.3$、$x_0 = 3$、そして $k_0 = 50$（$1/k$ スケジュール用）として厳密な漸化式を反復し、予算の 80% 時点および終了時点での期待損失を表示したうえで $\mathbb{E}f(x_k)$ をプロットする。

```
lam, sigma, K, eta0, x0 = 1.0, 1.0, 2000, 0.3, 3.0

def expected_loss(schedule, lam, sigma, K, x0):
    """E f(x_k) = lam e_k / 2 for k = 1..K from the exact second-moment recursion."""
    e, Ef = x0**2, np.empty(K)
    for k in range(K):
        eta = schedule(k)
        e = (1 - eta * lam)**2 * e + eta**2 * sigma**2
        Ef[k] = 0.5 * lam * e
    return Ef

schedules = {
    'constant': lambda k: eta0,
    'c/k': lambda k: eta0 / (1 + k / 50),
    'cosine': lambda k: 0.5 * eta0 * (1 + np.cos(np.pi * k / K)),
    'WSD': lambda k: eta0 if k < 0.8 * K else eta0 * (K - k) / (0.2 * K)}
traces = {name: expected_loss(s, lam, sigma, K, x0) for name, s in schedules.items()}
print(f'floor {eta0 * sigma**2 / (2 * (2 - eta0 * lam)):.4f}   '
      + ''.join(f'{name:>10s}' for name in traces))
for label, k in [('E f at 80%', int(0.8 * K) - 1), ('E f at end', K - 1)]:
    print(f'{label:15s}' + ''.join(f'{Ef[k]:10.2e}' for Ef in traces.values()))
d2l.plot(np.arange(1, K + 1), list(traces.values()), 'step k', 'E f(x_k)',
         yscale='log', legend=list(traces))
```

```
floor 0.0882     constant       c/k    cosine       WSD
E f at 80%       8.82e-02  2.36e-03  7.87e-03  8.82e-02
E f at end       8.82e-02  1.90e-03  1.68e-03  6.13e-03
```



漸化式は決定論的であるため、数値は毎回完全に同一となる。定数ステップは損失の下限 0.0882 の 10 分の 1 以内にステップ 9 までに落ち着き、そこにとどまる。WSD は減衰段階が始まるまで定数ステップの実行**そのもの**であるため、予算の 80% 時点ではまったく同じフロア上に位置する。その後、最後の 400 ステップで損失を 14.4 分の 1 の 6.1×10^{-3} まで引き下げており、改善のすべてがこの区間で生じている。コサイン減衰と $1/k$ 減衰はそれぞれ 1.7×10^{-3} と 1.9×10^{-3} という近い値で終了し、WSD よりも約 3〜4 倍低い損失に達する。

$1/k$ スケジュールの終端値は先の命題とよく一致する。ここでは $c = 0.3 \cdot 50 = 15$ であるため

$A = 225/29 \approx 7.8$ となり、$\mathbb{E}f(x_K) \approx \frac{1}{2}A/(K + k_0) = 1.9 \times 10^{-3}$ が得られる。ただし、この順位はこれらの二次関数パラメータに対する厳密な結果であり、ニューラルネットワークにそのまま当てはまるとは限らないことは念頭に置いておきたい。二次関数そのものにおいては、定数プラトーを維持しても最終損失は改善しない。WSD の減衰段階を長くするほど最終損失は単調に減少し、実行全体にわたる線形減衰では 2.7×10^{-3} に達する。

35.4.3 固定予算における収束率

減衰スケジュールは、終わりを定めない実行のために設計されたものである。しかし実際には、計算予算 K があらかじめわかっていることが多く、問いは次のように書き換えられる。K ステップに対してどの定数ステップサイズを選ぶべきか、またそれによってどの程度の精度が保証されるのか。強凸性がなければ幾何級数的な縮小率も厳密なノイズフロアも存在しないため、厳密な軌跡ではなく上界を証明することになる。以下に示す 2 つの上界はいずれも同じ 2 項構造を共有している。すなわち、$1/(\eta K)$ として減少する過渡項と、η として増大するノイズ項の和である。両者を釣り合わせると $\eta \propto 1/\sqrt{K}$ が選ばれる（Figure 35.17）。

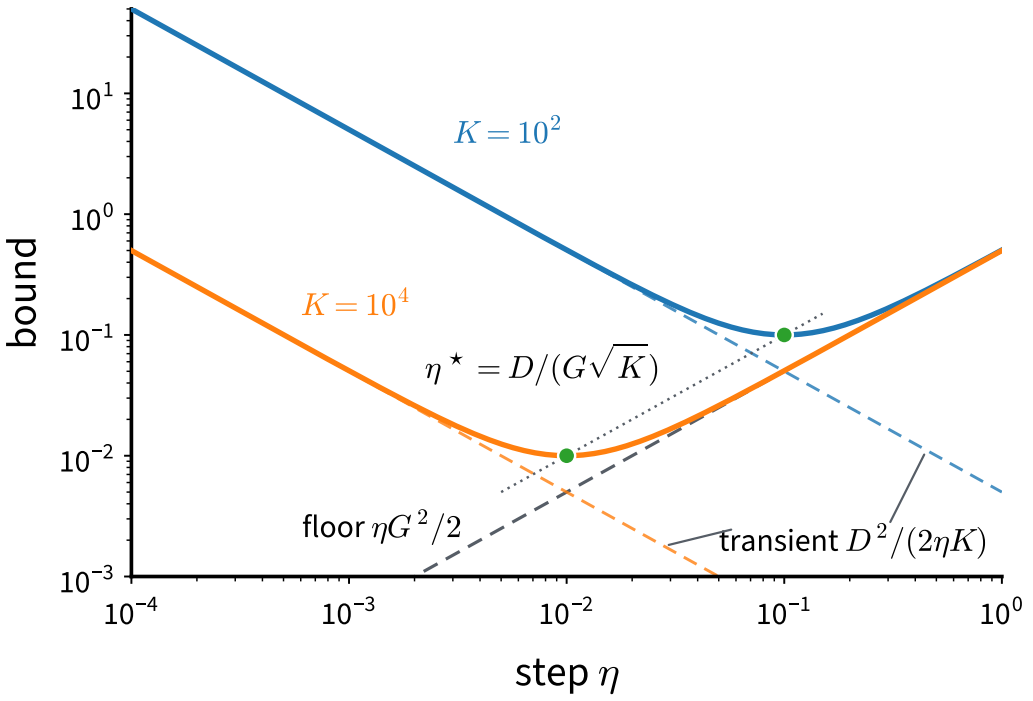


Figure 35.17: 2 項上界 $D^2/(2\eta K) + \eta G^2/2$（Equation 35.44）をステップサイズ η に対してプロットした図（$D = G = 1$、2 通りの予算）。過渡項は η とともに減少し、フロアはそれとともに上昇する。両者の和は最小値 $DG/\sqrt{K}$ に $\eta^\star = D/(G\sqrt{K})$ で達するため、予算を $100\times$ 大きくすると最適ステップと上界の双方が $10\times$ 低くなる。

35.4.3.1 凸損失における平均反復点

最初の上界に必要なのは凸性だけであり、35.3 節の劣勾配不等式を通じて適用する。すなわち、$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{s}^\top(\mathbf{y} - \mathbf{x})$ がすべての $\mathbf{y}$ およびすべての劣勾配 $\mathbf{s} \in \partial f(\mathbf{x})$ に対して成り立つ（$\nabla f(\mathbf{x})$ に帰着するのは f が微分可能な点である）。

命題（凸損失上の SGD、平均反復点）。 f を最小点 $\mathbf{x}^\star$ および最適値 $f^\star = f(\mathbf{x}^\star)$ を持つ凸関数とする。更新 $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\, \mathbf{g}_k$ を $\|\mathbf{x}_0 - \mathbf{x}^\star\| \le D$ を満たす初期点から実行し、各確率的勾配が $\mathbb{E}[\mathbf{g}_k \mid \mathbf{x}_k] \in \partial f(\mathbf{x}_k)$ および $\mathbb{E}[\|\mathbf{g}_k\|^2 \mid \mathbf{x}_k] \le G^2$ を満たすとする。このとき、平均反復点 $\bar{\mathbf{x}}_K = \frac{1}{K}\sum_{k=0}^{K-1} \mathbf{x}_k$ は次を満たす。

$$\mathbb{E}\big[f(\bar{\mathbf{x}}_K)\big] - f^\star \;\le\; \frac{D^2}{2\eta K} + \frac{\eta\, G^2}{2}, \tag{35.44}$$

さらに、$\eta = D/(G\sqrt{K})$ と選ぶと $\mathbb{E}[f(\bar{\mathbf{x}}_K)] - f^\star \le DG/\sqrt{K}$ が得られる。

証明。 最小点までの二乗距離を追跡する。1 回の更新を展開すると次式が得られる。

$$\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2 = \|\mathbf{x}_k - \mathbf{x}^\star\|^2 - 2\eta\, \mathbf{g}_k^\top(\mathbf{x}_k - \mathbf{x}^\star) + \eta^2 \|\mathbf{g}_k\|^2.$$

ここで $\mathbf{x}_k$ が与えられたときの条件付き期待値をとる。$\mathbf{s}_k = \mathbb{E}[\mathbf{g}_k \mid \mathbf{x}_k]$ と書くと、交差項は $-2\eta\, \mathbf{s}_k^\top(\mathbf{x}_k - \mathbf{x}^\star)$ となる。$\mathbf{y} = \mathbf{x}^\star$ における劣勾配不等式より、$\mathbf{s}_k^\top(\mathbf{x}_k - \mathbf{x}^\star) \ge f(\mathbf{x}_k) - f^\star$ が成り立つ。議論全体はこのステップにかかっている。損失ギャップが大きいときは常に、勾配が $\mathbf{x}^\star$ からの変位と同じ向きに揃い、反復点をより近くへと引き寄せるのである。最後の項を $\eta^2 G^2$ で上から抑え、全期待値をとって $r_k = \mathbb{E}\|\mathbf{x}_k - \mathbf{x}^\star\|^2$ とおくと、次式を得る。

$$2\eta\, \mathbb{E}\big[f(\mathbf{x}_k) - f^\star\big] \;\le\; r_k - r_{k+1} + \eta^2 G^2.$$

$k = 0, \dots, K-1$ にわたって和をとると距離が望遠鏡和となって打ち消し合い、$r_0 - r_K \le r_0 \le D^2$ であるから $2\eta \sum_{k<K} \mathbb{E}[f(\mathbf{x}_k) - f^\star] \le D^2 + K\eta^2 G^2$ となる。両辺を $2\eta K$ で割り、凸関数 f にイェンセンの不等式 $f(\bar{\mathbf{x}}_K) \le \frac{1}{K}\sum_{k<K} f(\mathbf{x}_k)$ を適用すると Equation 35.44 が得られる。2 つの項は $\eta = D/(G\sqrt{K})$ で一致し、そのとき各項は $DG/(2\sqrt{K})$ に等しい。$\blacksquare$

2 つの項の意味は明快である。$D^2/(2\eta K)$ は距離 D を K ステップ（ステップサイズ η）で移動するための過渡コストであり、$\eta G^2/2$ は 1 ステップあたりに注入されるノイズフロアである。特筆すべきことに、この証明は滑らかさを一切必要としないため、ヒンジ損失のように角を持つ非滑らかな損失に対しても上界が成り立つ。予算を 100 倍にすると、最適ステップサイズと保証される誤差の双方が 10 分の 1 に減少する。この $O(1/\sqrt{K})$ という収束率は最悪ケースにおいて改良不可能である。本命題が扱う問題クラス全体では、$\mathbb{E}\|\mathbf{g}\|^2 \le G^2$ を満たす不偏な確率的勾配のみを観測するいかなるアルゴリズムも、$DG/\sqrt{K}$ の定数倍より良い保証を与えることはできない (Nemirovski and Yudin 1983)。

強凸性を加えるとスケーリングは劇的に変化する。スカラー二次関数上では、予算 K に対する最適定数ステップは $e_K = \Theta(\log K / K)$ を達成する。強凸性によって、過渡項が ρ^{2K} のように幾何級数的に縮小し、$1/(\eta K)$ のような代数的減衰ではなくなるからである。

2 つの注意点がある。第 1 に、この命題が上界を与えるのは**平均**反復点 $\bar{\mathbf{x}}_K$ に対してである。最終反復点が従う上界はこれより弱く、既知の予算に対する最終反復点の最悪ケース最適スケジュールはゼロまで線形に減衰するものとなる (Defazio et al. 2023)。第 2 に、仮定 $\mathbb{E}\|\mathbf{g}_k\|^2 \le G^2$ は $\mathbb{R}^d$ 全体上の二次関数のように勾配が非有界な損失を除外する。ただし、有界な特徴量を持つ Lipschitz 損失（正則化なしロジスティック回帰など）や、反復点が有界集合内にとどまる場合には成り立つ。また最悪ケースの上界であるため、素性の良い問題では保守的になり得る。実際、球内に制限した先のロジスティック回帰課題では、真のギャップを約 1 桁過大評価する。

35.4.3.2 非凸損失におけるランダム反復点

ニューラルネットワークでは非凸な損失とノイズを含む勾配に同時に直面するが、ここまでの結果は一度に一方の困難だけを扱ってきた。関数 f の勾配が L-Lipschitz であるとき（L 滑らかな関数）、損失曲面はいたるところで曲率 L の接放物線の下側に位置する。

$$f(\mathbf{y}) \;\le\; f(\mathbf{x}) + \nabla f(\mathbf{x})^{\top}(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^{2} \quad \text{for all } \mathbf{x},\mathbf{y}.$$

厳密な勾配と $\eta = 1/L$ のもとでは、この二次上界から凸性の仮定なしに $\min_{k<K}\|\nabla f(\mathbf{x}_k)\|^2 \le 2L\Delta/K$ が導かれる。ここで $f^\star$ は f の下界であり、$\Delta = f(\mathbf{x}_0) - f^\star$ である（35.1 節）。一方、二次関数上でノイズを含む勾配を用いると、定数ステップはノイズフロアで頭打ちになる。Ghadimi and Lan (2013) の定理はこの両方の要素を組み合わせたものである。SGD の各ステップを $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\,\mathbf{g}_k$ とモデル化し、確率的勾配が次を満たすとする。

$$\mathbb{E}\left[\mathbf{g}_k \mid \mathbf{x}_k\right] = \nabla f(\mathbf{x}_k), \qquad \mathbb{E}\left[\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2 \mid \mathbf{x}_k\right] \;\le\; \sigma^2. \tag{35.45}$$

言葉で言えば、確率的勾配は不偏であり、そのノイズパワーは一様に σ^2 以下に抑えられる。Equation 35.39 より、上限が有限でありさえすれば、ミニバッチ勾配は $\sigma^2 = \sup_{\mathbf{x}} \operatorname{tr}\Sigma(\mathbf{x})/b$ として両方の仮定を満たす。以下で扱う有界な tanh モデルではこの上限は有限となるが、$\mathbb{R}^d$ 全体上の最小二乗法では最小点からの距離とともに勾配ノイズが増大するため無限大となる。

命題（SGD は近似停留点を見つける）。 **f を L 滑らかで下に $f^\star$ で有界な関数とし、$\Delta = f(\mathbf{x}_0) - f^\star$ とおく。Equation 35.45 を満たす確率的勾配と定数ステップサイズ $\eta \le 1/L$ を用いて SGD を実行するとき、次が成り立つ。**

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla f(\mathbf{x}_k)\|^2\right] \;\le\; \frac{2\Delta}{\eta K} + L\eta\sigma^2, \tag{35.46}$$

さらに、$\eta = \min\!\left(1/L,\ \sqrt{2\Delta/(L\sigma^2 K)}\right)$ と選ぶと、実行とは独立にインデックス R を $\{0,\dots,K-1\}$ から一様に抽出したとき、次が保証される。

$$\mathbb{E}\left[\|\nabla f(\mathbf{x}_R)\|^2\right] \;\le\; \frac{2L\Delta}{K} + 2\sigma\sqrt{\frac{2L\Delta}{K}}. \tag{35.47}$$

証明。 L 滑らかさによる上界を実際のステップ $\mathbf{y}-\mathbf{x} = -\eta\,\mathbf{g}_k$ に適用する。

$$f(\mathbf{x}_{k+1}) \;\le\; f(\mathbf{x}_k) - \eta\,\nabla f(\mathbf{x}_k)^{\top}\mathbf{g}_k + \tfrac{L\eta^2}{2}\,\|\mathbf{g}_k\|^2.$$

$\mathbf{x}_k$ を与えた条件付き期待値をとると、2 つのことが同時に達成される。不偏性により中央の項は $-\eta\|\nabla f(\mathbf{x}_k)\|^2$ となり、分散分解によって二次の項は信号とノイズの和 $\mathbb{E}[\|\mathbf{g}_k\|^2 \mid \mathbf{x}_k] = \|\nabla f(\mathbf{x}_k)\|^2 + \mathbb{E}[\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2 \mid \mathbf{x}_k] \le \|\nabla f(\mathbf{x}_k)\|^2 + \sigma^2$ に分解される。項をまとめると次式が得られる。

$$\mathbb{E}\left[f(\mathbf{x}_{k+1}) \mid \mathbf{x}_k\right] \;\le\; f(\mathbf{x}_k) - \eta\left(1 - \tfrac{L\eta}{2}\right)\|\nabla f(\mathbf{x}_k)\|^2 + \tfrac{L\eta^2\sigma^2}{2}.$$

これは標準的な降下補題（descent lemma）に、1 ステップあたり $\frac{1}{2}L\eta^2\sigma^2$ の分散ペナルティが加わったものである。$\eta \le 1/L$ のときは常に $1 - \frac{L\eta}{2} \ge \frac{1}{2}$ が成り立つ。全期待値をとって $k = 0, \dots, K-1$ にわたる和をとると、損失は望遠鏡和により $\mathbb{E}[f(\mathbf{x}_K)] - f(\mathbf{x}_0) \ge -\Delta$ となり、両辺を $\eta K/2$ で割れば Equation 35.46 が得られる。その左辺は、$\mathbb{E}[\|\nabla f(\mathbf{x}_R)\|^2]$（一様ランダムなインデックス R に対する期待値）にほかならない。最後に、釣り合いをとったステップサイズには 2 つの場合がある。$\sqrt{2\Delta/(L\sigma^2 K)} \le 1/L$ ならば Equation 35.46 の 2 項は等しくなり、それぞれ $\sigma\sqrt{2L\Delta/K}$ となる。そうでない場合は $\eta = 1/L$ かつ $\sigma^2 < 2L\Delta/K$ であるため、上界は $2L\Delta/K + \sigma^2 \le 2L\Delta/K + \sigma\sqrt{2L\Delta/K}$ となる。いずれの場合も Equation 35.47 で上から抑えられる。■

$\sigma = 0$ のときは第 2 項が消え、決定論的な収束率 $2L\Delta/K$ が復元される。ノイズが存在し、かつ予算が十分大きくて $K > 2L\Delta/\sigma^2$（したがって $\eta < 1/L$）となる領域では、ノイズ項が支配的となり $K^{-1/2}$ の速さでしか減衰しない。たとえば $L = \Delta = \sigma = 1$、$K = 10^4$ の場合、釣り合いのステップは $\eta \approx 0.014$ となり、上界は $2 \times 10^{-4} + 0.028$ となる。つまりノイズ項は決定論的な項の 140 倍も大きい。また、この上界が抑えているのは勾配ノルムの**二乗**である点にも注意が必要である。予算を 100 倍にすると $\mathbb{E}\|\nabla f\|^2$ は 10 分の 1 に減少するが、これは勾配ノルムそのものを 10 分の 1 に縮小するには 10^4 倍ものステップが必要になることを意味する。勾配推定量に統計的バイアスがある場合、証明の中で余分な内積項が生き残り、K を増やしても決して消えない定数のフロアが加わる（演習問題 4）。

なぜこの定理は最良の反復点ではなく**ランダム**な反復点 $\mathbf{x}_R$ を返すのだろうか。厳密な勾配があれば、毎ステップ $\|\nabla f(\mathbf{x}_k)\|$ を観測できるため、勾配が最小の反復点を単に選べばよい。しかし確率的勾配では、真の勾配ノルムはアルゴリズムからは見えない。そこでインデックス R を一様ランダムに選ぶという実装可能な規則を採用すると、Equation 35.46 の時間平均がまさにその期待二乗勾配ノルムに一致するのである。

凸の場合と比べると、2 つの点が変わっている。損失ギャップではなく二乗勾配ノルムを抑えていること、そして平均反復点ではなくランダム反復点で評価していることである。凸性がなければ停留点が大域的最小点であるとは限らないため、1 次法が保証できるのは勾配が小さいことだけである。凸の場合の上界と同様、この $K^{-1/2}$ という収束率は、Equation 35.45 を満たす確率的勾配のみにアクセスできるあらゆる手法にとって最悪ケースで最適である (Arjevani et al. 2023)。

35.4.3.3 単一の tanh ニューロンにおける収束率

この命題は上界であるため、実験によって収束率がタイトであることを証明することはできず、実際の実行が $K^{-1/2}$ という上限に対してどう振る舞うかを確かめられるのみである。以下では、ランダム反復点の指標と最良の反復点の双方を小規模な非凸問題上で測定する。具体的には、単一の tanh ニューロンを最小二乗法によって 5 次元のノイズを含む $n = 64$ 個の例に当てはめる。

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\tanh(\mathbf{a}_i^\top \mathbf{x}) - y_i\right)^2.$$

各成分は滑らかで有界であるが、引数が大きい領域で tanh が平坦なプラトーへと飽和するため、いずれも凸ではない。以下のセルではデータを生成し、パラメータベクトルのバッチ（X の各行に 1 つ）に対して厳密な全勾配を計算するベクトル化された補助関数を定義する。

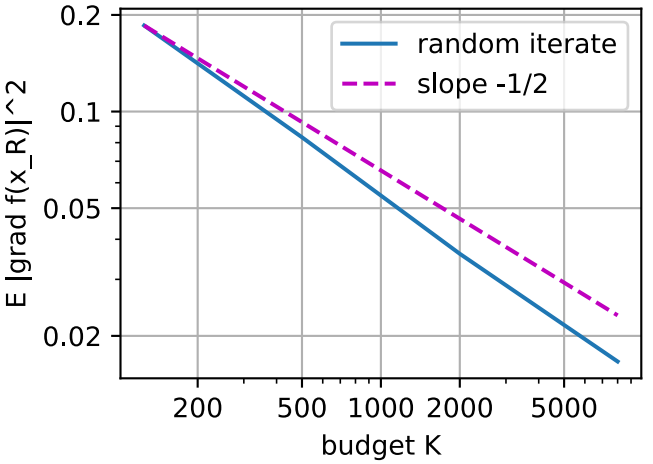
```
n_nc, d_nc = 64, 5                              # a small nonconvex problem
A_nc = rng.normal(size=(n_nc, d_nc))
y_nc = np.tanh(A_nc @ rng.normal(size=d_nc)) + 0.5 * rng.normal(size=n_nc)
```

```
def full_grad(X, A, y):
    """Exact gradient of mean_i (tanh(a_i^T x) - y_i)^2 at each row x of X."""
    t = np.tanh(X @ A.T)                          # (runs, n)
    return 2 * ((t - y) * (1 - t**2)) @ A / len(y)
```

予算 $K \in \{125, 500, 2000, 8000\}$ に対し、定理のスケーリング $\eta = 0.5/\sqrt{K}$ を用いて単一例の SGD を実行する。このとき 20 回の独立な実行を `X` の各行として並列に進める。各予算について、以下のセルは実行全体にわたる $\|\nabla f(\mathbf{x}_k)\|^2$ の平均（ランダム反復点の指標）とその最小値（最良の反復点）の双方を記録し、K に対する両対数傾きを表示したうえで、ランダム反復点の曲線を基準傾き $-1/2$ と並べてプロットする。

```
runs, Ks, avg, best = 20, np.array([125, 500, 2000, 8000]), [], []
for K in Ks:
    eta = 0.5 / np.sqrt(K)                        # the theorem's eta ~ 1/sqrt(K)
    X = np.zeros((runs, d_nc))                    # one row per independent run
    total, low = np.zeros(runs), np.full(runs, np.inf)
    for k in range(K):
        g2 = (full_grad(X, A_nc, y_nc)**2).sum(axis=1)   # |grad f(x_k)|^2
        total, low = total + g2, np.minimum(low, g2)
        i = rng.integers(0, n_nc, size=runs)             # one example per run
        t = np.tanh((X * A_nc[i]).sum(axis=1))
        X -= eta * (2 * (t - y_nc[i]) * (1 - t**2))[:, None] * A_nc[i]
    avg.append(total.mean() / K), best.append(low.mean())

def slope(v):
    return np.polyfit(np.log(Ks), np.log(v), 1)[0]

for name, v in [('random iterate', avg), ('best iterate', best)]:
    print(f'{name:15s}' + ' '.join(f'{u:.1e}' for u in v) + f'   slope {slope(v):.2f}')
d2l.plot(Ks, [avg, avg[0] * (Ks / Ks[0])**-0.5], 'budget K', 'E |grad f(x_R)|^2',
         xscale='log', yscale='log', legend=['random iterate', 'slope -1/2'])
```

```
random iterate 1.9e-01 8.3e-02 3.6e-02 1.7e-02   slope -0.58
best iterate   9.0e-03 2.1e-03 3.9e-04 4.9e-05   slope -1.25
```



予算を 64 倍に増やすと、ランダム反復点の指標は約 12 分の 1（実行ごとに概ね 9〜16 分の 1）に低下し、両対数傾きは -0.6 付近となる。予想どおり、定理の上界である $-1/2$ よりわずかに急である。最良の反復点はそれよりはるかに速く低下し、傾きは -1 前後かそれ以上に急になるが、前述のとおりノイズを含む勾配しか観測できないアルゴリズムにはそれを特定する術がない。

35.4.4 分散低減

35.4.4.1 制御変量

ステップサイズを減衰させれば η を縮めることでノイズフロアを解消できるが、前進が代数的な収束率に鈍化する代償を伴う。目的関数が n 個の例にわたる**有限**和である場合には、第 2 の選択肢がある。勾配ノイズそのものを除去するのである。最小点 $\mathbf{x}^\star$ では全勾配はゼロになるが、個々の例の勾配 $\nabla f_i(\mathbf{x}^\star)$ は一般にゼロにはならない。そして、これらの非ゼロの成分勾配こそが、定数ステップサイズが注入し続けるノイズの正体である。

対策が一切不要となる重要な領域も 1 つある。モデルが訓練セットを補間する（すべての例に損失ゼロで適合する）場合、最小点においてすべての $\nabla f_i(\mathbf{x}^\star)$ がゼロになり、強凸目的関数上の定数ステップ SGD はノイズフロアなしで線形収束する（演習問題 6）。大規模なニューラルネットワークは訓練セットを補間することが多く、このことが加法ノイズのフロアから予測されるよりも定数ステップ SGD がはるかに良好に機能する一因を説明している。

成分勾配が $\mathbf{x}^\star$ でゼロにならない場合には、その分散を直接打ち消さなければならない。これを行うのがモンテカルロ法の標準的な技法である。ノイズを含む推定量 $\nabla f_i(\mathbf{x})$ から、厳密な平均が既知である相関した確率変数を引き、その平均を足し戻すのである。このような補正項は**制御変量**（control variate）と呼ばれ、方策勾配のベースラインでも再び登場する（14.8 節）。有限和において自然な制御変量は、同じ例の勾配 $\nabla f_i(\tilde{\mathbf{x}})$（保存しておいたスナップショット $\tilde{\mathbf{x}}$ で評価したもの）であり、すべての i にわたるその平均はデータセット全体を 1 パス走査するごとに 1 回計算される全勾配 $\nabla f(\tilde{\mathbf{x}})$ である。これにより、**SVRG**（stochastic variance-reduced gradient、確率的分散低減勾配）推定量が得られる (Johnson and Zhang 2013)。

$$\tilde{\mathbf{g}}_i(\mathbf{x}) = \nabla f_i(\mathbf{x}) - \nabla f_i(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}). \tag{35.48}$$

言葉で言えば、SVRG はスナップショットにおける厳密な全勾配から出発し、$\tilde{\mathbf{x}}$ と $\mathbf{x}$ の間で単一の例の勾配がどれだけ変化したかによってそれを補正する。一様に抽出されたインデックス i に対して、この推定量は次のように不偏性を保つ。

$$\mathbb{E}_i\big[\tilde{\mathbf{g}}_i(\mathbf{x})\big] = \nabla f(\mathbf{x}) - \nabla f(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}) = \nabla f(\mathbf{x}),$$

一方で、反復点とスナップショットが収束するにつれて、その分散はゼロへと縮小する。

命題（SVRG の分散はスナップショットまでの距離で抑えられる）。各成分 f_i が L_i-Lipschitz 勾配を持ち、$L_{\max} = \max_i L_i$ とするとき、一様に抽出されたインデックス i に対して次が成り立つ。

$$\mathbb{E}_i\big\|\tilde{\mathbf{g}}_i(\mathbf{x}) - \nabla f(\mathbf{x})\big\|^2 \;\le\; \mathbb{E}_i\big\|\nabla f_i(\mathbf{x}) - \nabla f_i(\tilde{\mathbf{x}})\big\|^2 \;\le\; L_{\max}^2\,\|\mathbf{x} - \tilde{\mathbf{x}}\|^2. \tag{35.49}$$

証明。左辺は確率ベクトル $\nabla f_i(\mathbf{x}) - \nabla f_i(\tilde{\mathbf{x}})$ の分散であり、その平均は $\nabla f(\mathbf{x}) - \nabla f(\tilde{\mathbf{x}})$ である。任意の確率ベクトルの分散は高々その二次モーメント以下であるため、第 1 の不等式は直ちに従う。二次モーメントの各項に L_i-Lipschitz 上界を適用すれば、第 2 の不等式が得られる。$\blacksquare$

ノイズパワーは $\|\mathbf{x} - \tilde{\mathbf{x}}\|^2$ の定数倍で上から抑えられるため、$\mathbf{x}$ と $\tilde{\mathbf{x}}$ の双方が $\mathbf{x}^\star$ に近づくにつれて自動的にゼロとなり、定数ステップサイズであってもノイズフロアが取り除かれる。直感的には、近接する 2 点で同じ例を評価するとほぼ同一のノイズを共有する勾配が生じるため、差をとることで共有ノイズが相殺され、勾配の小さな真の変化だけが残るのである。

35.4.4.2 SVRG と SAGA

上界 Equation 35.49 を縮小させ続けるには、スナップショットが定期的に反復点に追いつかなければならないが、毎ステップ全勾配を再計算しては確率的最適化の意味が失われてしまう。SVRG はエポック単位で動作することでこれを解決する。各エポックの冒頭で現在の反復点をスナップショット $\tilde{\mathbf{x}}$ として保存し、その全勾配 $\nabla f(\tilde{\mathbf{x}})$ を計算する。続いて、新たなランダムインデックスを用いて m 回の内側更新 $\mathbf{x} \leftarrow \mathbf{x} - \eta\,\tilde{\mathbf{g}}_i(\mathbf{x})$ を実行し、最後の内側反復点を次のスナップショットに設定する。したがって 1 エポックのコストは、例の勾配評価 $n + 2m$ 回分となる。Johnson and Zhang (2013)（定理 1）は、すべての f_i が $L_{\max}$-Lipschitz 勾配を持つ凸関数であり、f が μ 強凸で、$\eta < 1/(4L_{\max})$ かつ 1 エポックあたりの縮小率が 1 未満になるほど m が十分大きければ、期待準最適性 $\mathbb{E}[f(\tilde{\mathbf{x}})] - f^\star$ がエポック数に対して幾何級数的に減衰することを証明している（定理ではランダムな内側反復点を次のスナップショットに選ぶが、以下の実装では実用上同様に振る舞う最終反復点を用いる）。

リッジ回帰上で SVRG を検証しよう。これは $f_i(\mathbf{x}) = \tfrac{1}{2}(\mathbf{a}_i^\top \mathbf{x} - y_i)^2 + \tfrac{0.1}{2}\|\mathbf{x}\|^2$、例数 $n = 300$、パラメータ数 $d = 12$ からなる強凸な有限和である。厳密な最小点は正規方程式を解くことで求まり、各成分の勾配は L_i-Lipschitz（$L_i = \|\mathbf{a}_i\|^2 + 0.1$）である。以下のセルではデータを生成して成分勾配と全勾配を定義し、$L_{\max}$ と理論上のステップサイズ上限 $1/(4L_{\max})$ を表示する。

```
n_r, d_r, reg_r = 300, 12, 0.1                  # ridge regression, a finite sum
A_r = rng.standard_normal((n_r, d_r))
y_r = A_r @ rng.standard_normal(d_r) + 0.5 * rng.standard_normal(n_r)
H_r = A_r.T @ A_r / n_r + reg_r * np.eye(d_r)     # Hessian of f
x_star_r = np.linalg.solve(H_r, A_r.T @ y_r / n_r)

def component_grad(x, i, A, y, reg):
    """Gradient of f_i(x) = (a_i^T x - y_i)^2 / 2 + reg |x|^2 / 2 (i: index or array)."""
    return A[i] * (A[i] @ x - y[i])[..., None] + reg * x

def full_grad_sum(x, A, y, reg):
    """Gradient of the finite sum f = mean_i f_i."""
```

```
    return A.T @ (A @ x - y) / len(y) + reg * x

def objective(x, A, y, reg):
    """The ridge objective f(x)."""
    return 0.5 * np.mean((A @ x - y)**2) + 0.5 * reg * x @ x

L_max = (A_r**2).sum(axis=1).max() + reg_r       # largest per-example smoothness
print(f'L_max = {L_max:.0f}, 1 / (4 L_max) = {1 / (4 * L_max):.4f}')
```

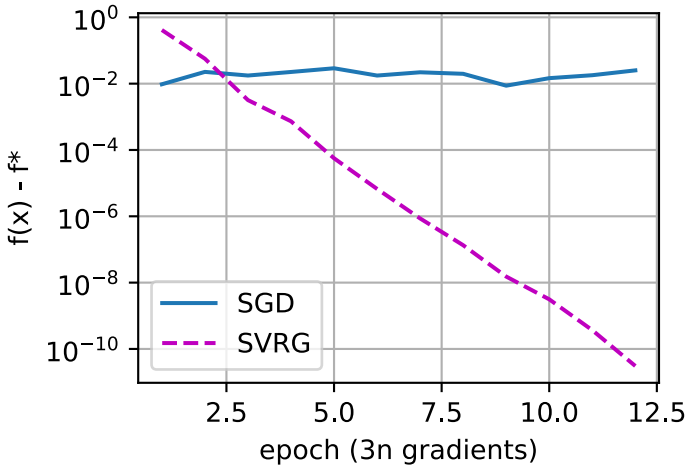
```
L_max = 29, 1 / (4 L_max) = 0.0088
```

データのランダムな生成に対して $L_{\max}$ は概ね 25〜40 の範囲に入り、定理のステップサイズ上限 $1/(4L_{\max})$ はおよそ 0.006〜0.01 の間となる。

次に、SVRG（$m = n$）と単一例の SGD を、同じ定数ステップサイズ $\eta = 0.02$（定理の上限の 2 倍超）のもとで 12 エポック（各エポックあたり例の勾配評価 $3n$ 回のコスト）にわたり比較する。あわせて、先に導出した二次関数のフロア公式も SGD 上で検証する。定常状態における損失ギャップは、ヘッセ行列の固有ベクトルにわたる $\eta\sigma_i^2/(2(2-\eta\lambda_i))$ の和に等しくなるはずである。ここで σ_i^2 は、$\mathbf{x}^\star$ における例の勾配の固有ベクトル i に沿った分散である。

```
eta, epochs = 0.02, 12
f_star_r = objective(x_star_r, A_r, y_r, reg_r)
snap, gaps_svrg = np.zeros(d_r), []
for _ in range(epochs):                          # n + 2n example gradients per epoch
    g_snap = full_grad_sum(snap, A_r, y_r, reg_r)
    x = snap.copy()
    for i in rng.integers(0, n_r, size=n_r):
        x -= eta * (component_grad(x, i, A_r, y_r, reg_r)
                    - component_grad(snap, i, A_r, y_r, reg_r) + g_snap)
    snap = x
    gaps_svrg.append(objective(snap, A_r, y_r, reg_r) - f_star_r)
x, gaps_sgd = np.zeros(d_r), []
for _ in range(epochs):
    for i in rng.integers(0, n_r, size=3 * n_r):  # the same 3n budget per epoch
        x -= eta * component_grad(x, i, A_r, y_r, reg_r)
    gaps_sgd.append(objective(x, A_r, y_r, reg_r) - f_star_r)
G_r = component_grad(x_star_r, np.arange(n_r), A_r, y_r, reg_r)  # noise at x*
ev, Q = np.linalg.eigh(H_r)                      # Hessian eigenvalues, eigenvectors
s2 = ((G_r @ Q)**2).mean(axis=0)                 # noise variance per eigenvector
floor = (eta * s2 / (2 * (2 - eta * ev))).sum()  # the loss floors add
print(f'SGD: predicted floor {floor:.1e}, mean gap in epochs 4-12 '
      f'{np.mean(gaps_sgd[3:]):.1e}')
print('SVRG: gap after epochs 1, 4, 12: '
      + ', '.join(f'{gaps_svrg[e - 1]:.0e}' for e in (1, 4, 12)))
d2l.plot(np.arange(1, epochs + 1), [gaps_sgd, gaps_svrg], 'epoch (3n gradients)',
         'f(x) - f*', yscale='log', legend=['SGD', 'SVRG'])
```

```
SGD: predicted floor 1.9e-02, mean gap in epochs 4-12 2.0e-02
SVRG: gap after epochs 1, 4, 12: 4e-01, 7e-04, 3e-11
```



SGD は序盤で急速に低下したのち、10^{-2} の数倍程度で頭打ちとなり、ちょうどフロア公式が予測する位置に落ち着く。繰り返し実行すると、実測されるプラトーは予測値の約 10% 下から 40% 上の間に収まる。下回るよりわずかに上回ることが多いのは、公式が $\mathbf{x}^\star$ でのみ勾配の分散を評価しているのに対し、最小点から離れると例の勾配がわずかに広く散らばるためである。SVRG は出だしこそ遅れるものの、最終的には大きく引き離す。最初の 1 エポック後（出発点に固定されたスナップショットのもとで n 回の内側ステップしか進まない）では、ギャップはまだ SGD を上回っている。しかし第 2 エポック以降は 1 エポックあたり約 8 分の 1（実行により 5〜13 分の 1）のペースで縮小し、10^{-13}〜10^{-8} に第 12 エポックまでに到達する。特筆すべきことに、SVRG は定理の上限の 2 倍を超えるステップサイズでこの線形収束を達成しており、理論上のステップ条件が十分条件であって必要条件ではないことを示している。この優位性を支える 3 つの仮定は念頭に置いておきたい。すなわち、有限和であること、データセットを複数パス走査できること、そして線形収束率のための強凸性である。各例を 1 度しか観測しないストリーミングデータでは、スナップショットをとるべき有限和が存在しない。

SAGA (Defazio et al. 2014) は、スナップショットを各例について直近に評価された勾配を保持するテーブル $\mathbf{t}_1, \ldots, \mathbf{t}_n$ に置き換えることで、定期的な全データ走査を完全に不要にする。各ステップでは推定量 $\nabla f_i(\mathbf{x}) - \mathbf{t}_i + \frac{1}{n}\sum_j \mathbf{t}_j$ を構成したのち、$\mathbf{t}_i$ を $\nabla f_i(\mathbf{x})$ で上書きする。SAGA は 1 ステップあたり新たな例の勾配を 1 つ計算するだけで済むが、代償として $n \times d$ 個の数値を保存する必要がある（ただし一般化線形モデルでは 1 例あたり 1 個のスカラーで足りる。演習問題 3 を参照）。

残念ながら深層ニューラルネットワークにおいては、SVRG も SAGA も一貫した改善をもたらしていない。損失ランドスケープが急速に変化したり、データ拡張（data augmentation）によってパスごとに各例が摂動を受けたりすると、制御変量がすぐに陳腐化してしまうからである。他方で、悪条件な谷の例以来つきまとってきた第 2 の制約も残されている。本節のすべての手法は全座標に単一の大域的なステップサイズを適用するため、通し例における最も硬い方向によって η が $2/1000$ 以下に制限されてしまうのである。次節では、これまでに観測した勾配から各座標が独自のステップサイズを適応的に調整できるようにすることで、このボトルネックを取り除く。

35.4.5 まとめ

サイズ b のミニバッチ勾配は ∇f の不偏推定量であり、そのノイズパワーは $\operatorname{tr}\Sigma/b$ である。二次関数上では、固定ステップサイズ η によって各座標の二次モーメントが縮小率 $(1-\eta\lambda)^2$ で幾何級数的に減衰してノイズフロア $\eta\sigma^2/(\lambda(2-\eta\lambda))$ へと収束し、座標ごとの損失の下限

$\eta\sigma^2/(2(2-\eta\lambda))$ が次元全体にわたって足し合わされる。フロアは η に対して線形に増大する一方、過渡項は η が大きいほど速く縮小するため、いかなる定数ステップも最小点には到達できず、本節の各手法はいずれもこのトレードオフに対処している。Robbins–Monro スケジュールはフロアを解消するが、$\eta_k = c/k$ が $O(1/k)$ を達成するのは $c > 1/(2\mu)$ の場合に限られる。K ステップの固定予算においては、定数ステップ $\eta \propto 1/\sqrt{K}$ を選ぶことで、凸損失上の平均反復点に対して $DG/\sqrt{K}$ が、滑らかな非凸損失上のランダム反復点における期待二乗勾配に対して $O(\sigma\sqrt{L\Delta/K})$ が保証される。最後に有限和においては、SVRG の制御変量が不偏性を保ちつつ分散を高々 $L_{\max}^2\|\mathbf{x}-\tilde{\mathbf{x}}\|^2$ に抑えるため、強凸問題上で固定ステップサイズのまま線形収束することが可能となる。

35.4.6 演習問題

1. [code] **ノイズスケールと臨界バッチサイズ。**●1 回のミニバッチステップは損失をどれだけ引き下げ、その改善は b にどう依存するだろうか。$\mathbf{g} = \nabla f(\mathbf{x})$、$\mathbf{H} = \nabla^2 f(\mathbf{x})$ と書き、ステップ $\mathbf{x} - \eta\,\hat{\mathbf{g}}_b$（Equation 35.38）の後の損失を $\mathbf{x}$ の周りでの 2 次テイラー展開によって近似せよ。以下の解析は McCandlish et al. (2018) に従う。
   1. 共分散 Σ/b（$\hat{\mathbf{g}}_b$ の共分散、Equation 35.39 の証明で導出）を用いて、ステップ後の期待損失が $\eta_{\text{opt}}(b) = \eta_{\max}/(1 + B_{\text{noise}}/b)$ で最小化され、そのときの期待改善量が $\Delta_{\max}/(1 + B_{\text{noise}}/b)$ となることを示せ。ただし、各量は次式で定義される。
   $$\eta_{\max} = \frac{\|\mathbf{g}\|^2}{\mathbf{g}^\top \mathbf{H}\mathbf{g}}, \qquad \Delta_{\max} = \frac{\|\mathbf{g}\|^4}{2\,\mathbf{g}^\top \mathbf{H}\mathbf{g}}, \qquad B_{\text{noise}} = \frac{\operatorname{tr}(\mathbf{H}\Sigma)}{\mathbf{g}^\top \mathbf{H}\mathbf{g}}.$$
   2. Goyal et al. (2017) の線形スケーリング則は、η に b と同じ倍率を掛ける。どのようなバッチサイズの領域でこの規則が $\eta_{\text{opt}}(b)$ から導かれるか。また、$b \gg B_{\text{noise}}$ となった領域では、バッチを大きくすることで 1 ステップあたり何が得られるか。
   3. $\mathbf{H}$ が単位行列の定数倍であるとき、B_{noise} は $B_{\text{simple}} = \operatorname{tr}\Sigma/\|\mathbf{g}\|^2$、すなわち本文の勾配ノイズスケール b_{noise} に帰着する。本節のロジスティック回帰問題について、B_{simple} と B_{noise} を $\mathbf{x} = \mathbf{0}$ ならびに 5、10、20 回の全勾配ステップ（$\eta = 0.5$）の後に計算せよ。計算には `example_grads` とヘッセ行列 $\mathbf{H}(\mathbf{x}) = \frac{1}{n}\sum_i s_i(1-s_i)\,\mathbf{a}_i\mathbf{a}_i^\top + 0.1\,\mathbf{I}$（ただし $s_i = 1/(1+e^{y_i \mathbf{a}_i^\top \mathbf{x}})$）を用いよ。反復点が最適点に近づくにつれてノイズスケールはどう変化するか。B_{noise} を (b) の 2 つの領域を分ける臨界バッチサイズとみなしたとき、訓練の実行を通じてバッチサイズをどのように変化させるべきか。
2. [code] $1/k$ **スケジュールにおける定数。**●●本文では定数 A（Equation 35.43）を主要項の比較によって求めた。本演習ではこれを厳密に証明し、通し例に適用する。
   1. $t = k + k_0$ とし、$u_{k+1} \le (1 - a/t)\,u_k + q/t^2$ がすべての $k \ge 0$ で成り立つとする（ただし $a > 1$、$q > 0$、$u_k \ge 0$、$k_0 \ge a$）。帰納法により、$u_k \le \nu/(k + k_0)$（ただし $\nu = \max\{q/(a-1),\ k_0 u_0\}$）を証明せよ。（ヒント：$1 - a/t \ge 0$ かつ $1/t - 1/t^2 \le 1/(t+1)$ である。）
   2. この補題を漸化式 $e_{k+1} = (1 - c\mu/t)^2 e_k + c^2\sigma^2/t^2$ に適用せよ。$(1 - c\mu/t)^2 \le 1 - (2c\mu - (c\mu)^2/k_0)/t$ が $t \ge k_0$ で成り立ち、したがって任意の $2c\mu > 1$ に対して $e_k = O(1/k)$ となる（k_0 が十分大きい場合）ことを示せ。また a と q を特定し、$q/(a-1) \to A$（$k_0 \to \infty$ のとき）を示せ。
   3. $2c\mu < 1$ のとき、$e_k = \Theta(k^{-2c\mu})$ となることを、積 $\prod_{j<k}(1 - c\mu/(j+k_0))^2$ および展開したノイズ和から示せ。
   4. $\mu = \sigma = 1$、$k_0 = 10$、$e_0 = 1$ として漸化式を $k = 10^6$ まで $c \in \{0.1, 0.25, 0.5, 1, 2\}$ の各値について反復せよ。$k = 10^5 \sim 10^6$ の区間における両対数傾きと、$k\,e_k$ の 10^6 時点での値を報告せよ。A の最小点の 2 倍の定数と半分の定数では、どちらの代償が大きいか。

5. 座標ごとに単位ノイズを伴う通し例 $\tfrac{1}{2}(x_1^2 + 1000\,x_2^2)$ では、最初のステップにおける安定性から $c/k_0 < 2/1000$ が強いられる。$c = 1/\mu = 1$、$k_0 = 1000$、$\mathbf{x}_0 = (1, 1)$、$K = 10^4$ とせよ。2 座標の漸化式によって期待損失を計算し（約 4.2×10^{-3} となる）、定数ステップ $\eta = 10^{-3}$（7.5×10^{-4}）および同じ η_0 から出発するコサイン減衰（1.0×10^{-4}）と比較せよ。$1/k$ 減衰はどちらの座標の前進を阻むか。また、なぜ条件数が k_0 に再び現れるのか。

3. [code] **SVRG の内側ループと SAGA**。本節のリッジ回帰データと関数 `component_grad`、`full_grad_sum`、`objective` を用い、例の勾配評価 $36n = 10{,}800$ 回（SVRG のセルで報告した 12 エポック分の予算）のもとで実験せよ。各実行は、1 エポックごとに確認してギャップを 10^{-10} 未満に下げるまでに要した評価回数で測ること。
   1. 内側ループを m ステップ実行し（1 エポックのコストは $n + 2m$ 回の評価となる）、$m \in \{10, 30, 75, 300, 1200\}$ を $\eta \in \{0.01, 0.02\}$ のそれぞれで変化させよ。各ステップサイズにおいて最良の m はどれか。スナップショット勾配のコストを償却することと制御変量が陳腐化することのトレードオフ、および最良の m が η に応じて変化する理由を説明せよ。
   2. SAGA を実装せよ。すなわち、すべての例の直近の勾配を保持するテーブルを用意し（$\mathbf{x}_0$ において n 回の評価で初期化する）、推定量として $\nabla f_i(\mathbf{x}) - \mathbf{t}_i + \frac{1}{n}\sum_j \mathbf{t}_j$ を用いよ（$\mathbf{t}_i$ は例 i に対するテーブルの項目）。両方のステップサイズで SAGA を最良の SVRG 実行と比較せよ。この問題でテーブルは何個の数値を保持するか。また、各例につきスカラー $\mathbf{a}_i^\top \mathbf{x} - y_i$ のみを保存する場合には何個で足りるか。

4. **バイアスのある勾配推定量**。●Equation 35.45 の分散上界を（ここでは条件付き平均の周りで測るものとして）保ちつつ、不偏性を $\mathbb{E}[\mathbf{g}_k \mid \mathbf{x}_k] = \nabla f(\mathbf{x}_k) + \delta_k$（ただし $\|\delta_k\| \le \delta$）に置き換える。
   1. Equation 35.46 の証明を $\eta \le 1/L$ のもとでやり直し、次式を示せ。

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\|\nabla f(\mathbf{x}_k)\|^2\right] \;\le\; \frac{2\Delta}{\eta K} + L\eta\sigma^2 + \delta^2. \tag{35.50}$$

   展開する前に、$L\eta \le 1$ を項 $\frac{L\eta^2}{2}\|\nabla f(\mathbf{x}_k) + \delta_k\|^2$ に適用せよ。
   2. 項 δ^2 が除去できないことを示せ。すなわち、1 次元の f と定数の統計的バイアスを与え、$\sigma = 0$ の SGD が $|f'|^2 = \delta^2$ となる点に収束する例を構成せよ。
   3. 非復元で一様に抽出された単一のミニバッチは、依然として全勾配の不偏推定量である。では、以下の推定量のうちこの意味で統計的バイアスを持つものはどれか。エポックごとに毎回新たなランダム順ですべての例を 1 度ずつ訪問するランダムな再シャッフル（9.2 節）の各ステップにおける例の勾配、固定された最大ノルムにクリッピングされた確率的勾配、非同期訓練のように τ ステップ前の反復点で評価された勾配。

5. [code] **反復点平均化**。●Polyak and Juditsky (1992) は、緩やかに減衰するステップまたは定数ステップの実行における反復点を平均化すると、最良の定数が達成されることを示した。定数ステップ $\eta \in (0, 2/\lambda)$ を持ち、定常分布から出発するスカラーモデルを考える。
   1. $\mathrm{Cov}(x_j, x_k) = V\rho^{|j-k|}$ を示せ。
   2. 平均 $\bar{x}_K = \frac{1}{K}\sum_{k<K} x_k$ が $\mathbb{E}[\bar{x}_K^2] = \frac{V}{K}\cdot\frac{1+\rho}{1-\rho} - O(K^{-2})$ を満たし、かつ $V(1+\rho)/(1-\rho) = \sigma^2/\lambda^2$ がそのようなすべての η に対して成り立つこと、すなわち主要項において平均反復点はステップサイズに依存しないことを示せ。
   3. $K = 2000$、実行回数 4,000 回として、$\eta = 0.3$ および $\eta = 0.1$（$\lambda = \sigma = 1$）でシミュレーションにより確認せよ。いずれの場合も $\mathbb{E}[\bar{x}_K^2] \approx 5 \times 10^{-4}$ となるはずである（最終反復点のフロアはそれぞれ 0.18 と 0.053 である）。
   4. 代わりに $x_0 = 3$ から出発すると、平均は過渡項の影響を記憶する。その $\mathbb{E}[\bar{x}_K^2]$ への寄与がおよそ $(x_0/(\eta\lambda K))^2$ であることを示し、どのようなときに実行の後半のみを

平均化するほうが良いか判定せよ。

5. 最良の $1/k$ スケジュール（定数は $A = \sigma^2/\mu^2$）と比較せよ。収束率 $\sigma^2/(\mu^2 K)$ に至るこれら 2 つの経路は、それぞれ何を必要とするか。

6. [code] **補間と線形収束。●●** 各 f_i が L_i-Lipschitz 勾配を持つ凸関数であり、f が μ 強凸で、モデルがすべての訓練例に厳密に適合する場合のようにすべての f_i が同一の点 $\mathbf{x}^\star$ で最小化されると仮定する。以下の解析は Vaswani et al. (2019) に従う。
   1. $\mathbb{E}_i \|\nabla f_i(\mathbf{x})\|^2 \le 2L_{\max}(f(\mathbf{x}) - f^\star)$ を示せ。（ヒント：凸かつ L 滑らかな関数 h が $\mathbf{x}^\star$ で最小化されるとき、$\|\nabla h(\mathbf{x})\|^2 \le 2L(h(\mathbf{x}) - h(\mathbf{x}^\star))$ が成り立つ。）
   2. 定数 $\eta \le 1/L_{\max}$ を用いる単一例の SGD が $\mathbb{E}\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2 \le (1 - \eta\mu)\,\mathbb{E}\|\mathbf{x}_k - \mathbf{x}^\star\|^2$ を満たすこと、すなわちフロアなしで線形収束することを導け。このとき $\operatorname{tr}\Sigma(\mathbf{x}^\star)$ はいくらになるか。
   3. 例数 $n = 50$、パラメータ数 $d = 100$、$\eta = 1/L_{\max}$、$\mathbf{x}_0 = \mathbf{0}$ の最小二乗法でこれを確認せよ。反復点は $\mathbf{A}$ の行空間内にとどまり、そこでは f が強凸であるため (b) が適用できる。整合するラベルの場合にも、標準偏差 0.5 のノイズで汚染されたラベルの場合にもギャップが線形に減少することを示し、なぜここではノイズが影響しないのかを説明せよ。
   4. $n = 200$、$d = 100$ として同様に実験せよ。今度はノイズ付きラベルによってフロアが復活する。それはなぜか。

35.5 適応的手法

本書で扱う Transformer は、大半の大規模ニューラルネットワークと同様に、通常の確率的勾配降下法（stochastic gradient descent、SGD）ではなく Adam またはその変種である AdamW（9.5.3 節）で訓練される。その主な理由はステップサイズ（step size）にある。35.4 節の各手法は、すべての座標に単一のステップサイズを適用し、それを時間方向にのみ調整する。しかし現代のネットワークには 10^7〜10^{11} 個ものパラメータがあり、その曲率（curvature）は何桁も異なるうえ、事前にはわからない。すべての座標が 1 つのステップサイズを共有すると、最も急な方向がステップの上限を抑えてしまい、平坦な方向はほとんど動かなくなる。本節では、オプティマイザが勾配のみから座標ごとに個別のステップサイズをどのように選べるのか、そしてその戦略がどこで破綻するのかを問う。

本章を通じた例題を見れば、この違いがいかに大きいかがわかる。細長い谷 $f(\mathbf{x}) = \tfrac{1}{2}(x_1^2 + 1000\,x_2^2)$ において $(1, 1)$ から出発する場合を考えよう。最適な単一のステップサイズ $\eta = 2/1001$ を用いる勾配降下法（gradient descent）では、f を 10^{-8} 未満にするのに 6,160 ステップを要する。x_2 を安定に保てるほど小さなステップは、x_1 にとっては 1,000 倍も小さすぎるからである。新しいステップに直前のステップの定数倍を加える 35.1 節のヘビーボール法（heavy-ball method）では、これが 296 ステップまで短縮される（これらの回数が同節で挙げた $6{,}908$ ステップおよび 315 ステップより少し少ないのは、同節が最小点までの距離で停止したのに対し、本節では f の値で停止するためである）。これに対し、観測した勾配のみから推定した個別のステップサイズを各座標に与えるオプティマイザは、わずか数十ステップしか必要としない。さらに驚くべきは、その回数が何に依存しないかという点である。谷をさらに 100 倍細くしても、反復回数はたった 6 ステップしか増えない。

このような目を引く数値は作り出すのも容易なら読み誤るのも容易であるため、本節の多くの部分は比較を公平にすることに充てられる。ヘビーボール法は理論上の最適設定で実行し、Adam には単純なステップサイズのスイープを与え、さらに谷を 45° 回転させた場合も調べる。このより公平な検証のもとでは、Adam の優位性は最初の数値が示唆するものとはかなり異なって見えてくる。

根底にある仕組みは単純であり、AdaGrad、RMSProp、Adam がするように、各座標のステップ

を過去の勾配の二乗平均平方根（root mean square、RMS）で割ることである。本節ではまず、この正規化によって Adam が各座標のスケールに対して不感になることを証明し、続いてその対称性が破れる境界、すなわち谷の回転、凸な損失の周期的な系列、重み減衰（weight decay）、訓練序盤のステップ、および層の行列構造へと議論を進める。特に重要な結果は、スケール不変性（scale invariance）の命題と AdamW の不動点（fixed point）解析の 2 つである。AdaGrad のリグレット上界と末尾の構造化前処理行列（structured preconditioner）については、初読時には飛ばしてもよい。本節を通して、t はオプティマイザのステップ番号、h_i は座標 i の曲率、λ は重み減衰係数を表す。

35.5.1 座標ごとのステップサイズ

35.5.1.1 対角二次関数

二次関数に対する勾配降下法は、条件数（condition number）$\kappa = h_{\max}/h_{\min}$ に比例するステップ数を要する。単一のステップサイズを $2/h_{\max}$ 未満に保たなければならないからである（35.1 節）。その解決策が見えてくる最も単純な設定は、次の 2 変数の対角二次関数である。

$$f(\mathbf{x}) = \tfrac{1}{2}(h_1 x_1^2 + h_2 x_2^2), \qquad \frac{\partial f}{\partial x_i} = h_i x_i.$$

各偏微分がそれぞれの座標のみに依存するため、2 つの座標でステップサイズを共有すべき必然性はない。各座標に個別のステップサイズ η_i を与えれば、x_i は $x_i - \eta_i h_i x_i = (1 - \eta_i h_i)\, x_i$ へと更新され、$\eta_i = 1/h_i$ と選べばすべての座標が 1 ステップでゼロに到達する。$h = (1, 1000)$ の場合、理想的なステップサイズは 1 と 10^{-3} である。これは対角なヘッセ行列に対するニュートン法（Newton's method）にほかならず、反復回数から条件数が消え去る。

この理想と実用的なオプティマイザの間には 2 つの障害が立ちはだかる。曲率 h_i が未知であり訓練の過程で変化すること、そして観測される勾配がノイズを含むミニバッチ推定値であることである。したがって、勾配の大きさを代理指標として用い、勾配のみから各座標のステップサイズを推定しなければならない。対角二次関数では、勾配の成分 $g_i = h_i x_i$ の大きさは $h_i |x_i|$ となる。各座標の大きさが同程度である限り、勾配が大きい座標は曲率も高いため、勾配の大きさで割ることは実質的に曲率で割ることに相当する。ただし座標間でスケールが大きく異なる場合でも、$g_i / |g_i|$ の大きさは依然として 1 であるため、正規化は 1 ステップあたりの移動距離を η に固定する働きをし、真のニュートンステップ（Newton step）である x_i をとるわけではない。

35.5.1.2 AdaGrad

曲率の違いは、各座標に固有のステップサイズを与える理由の 1 つである。これとは独立した 2 つ目の理由として、座標ごとに更新頻度が大きく異なることが挙げられる。オンライン最適化や非滑らかな凸最適化（convex optimization）における古典的な選択である $\eta/\sqrt{t}$ のような大域的な減衰ステップサイズは、すべての座標を同一に扱う。Bag-of-Words モデルにおける希少語の重みのように、座標 1 は毎ステップ ± 1 の勾配を受け取る一方、座標 2 は 10^4 ステップに 1 回しか勾配を受け取らないとしよう。$t = 10^4$ の時点では、大域スケジュールは両方のステップサイズを $\eta/100$ まで減衰させてしまうため、希少な座標に関して得られた唯一の証拠によっても、その座標は η の 100 分の 1 しか動かない。これに対し、各座標が自身の二乗勾配を個別に累積すれば、累積和はそれぞれ 10^4 と 1 になり、η をそれぞれの平方根で割ることで、頻繁に更新される座標には $\eta/100$ を与えつつ、希少な座標のステップサイズを η のまま保てる。

これが **AdaGrad** である (Duchi et al. 2011)。勾配の各成分を二乗したベクトルを $\mathbf{g}_t \odot \mathbf{g}_t$ と書き（以降は $\mathbf{g}_t^2$ と略記する）、ベクトルに対する平方根や除算も同様に要素ごとに適用するものとする。AdaGrad は二乗勾配の累積和 $\mathbf{s}_t$ を保持し、ステップをその平方根で割る。

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \mathbf{g}_t \odot \mathbf{g}_t, \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta}{\sqrt{\mathbf{s}_t} + \epsilon} \mathbf{g}_t, \tag{35.51}$$

初期値は $\mathbf{s}_0 = \mathbf{0}$ とし、小さな定数 ϵ でゼロ除算を防ぐ（本編の 9.5 節では ϵ を平方根の内側に置いているが、ここでの定性的な議論に違いはない）。これにより、各座標のステップサイズは全体のステップ数ではなく、その座標自身の勾配履歴に応じて減衰するようになる。

計量の観点。最急降下はステップ長をどう測るかに依存する。重み付きノルム $\|\mathbf{d}\|_{\mathbf{P}} = \sqrt{\mathbf{d}^\top \mathbf{P}\, \mathbf{d}}$（$\mathbf{P}$ は正定値行列）のもとでの最急降下方向は $-\mathbf{P}^{-1}\nabla f$ である（演習問題 1、35.1 節）。AdaGrad は対角計量 $\mathbf{P}_t = \operatorname{diag}(\sqrt{\mathbf{s}_t})$ における最急降下であり、座標 i に沿う移動には、その勾配が大きかったことの累積的な証拠 $\sqrt{\sum_{\tau \le t} g_{\tau,i}^2}$ に比例するペナルティが課される。ステップを進める前に勾配を再スケーリングする $\mathbf{P}_t$ のような行列を**前処理行列**（preconditioner）と呼ぶ。ニュートン法が真の曲率 $\operatorname{diag}(h_i)$ を用いるのに対し、AdaGrad は勾配の履歴でこれを代用する（演習問題 1）。

リグレットの観点。AdaGrad はもともと**オンライン凸最適化**（online convex optimization）のために設計された手法である。この枠組みでは、凸な損失 $f_1, f_2, \dots$ が 1 つずつ順番に到着し、学習者は $\mathbf{x}_t$ を f_t を観測する前に選ばなければならない。T ラウンドにわたる性能は**リグレット** $\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}} \sum_{t=1}^{T} f_t(\mathbf{x})$、すなわち事後的に選んだ最良の固定点に対する累積過剰損失によって測られる。この枠組みにおいて、アルゴリズムの平均リグレット（リグレットを T で割った値）がゼロに収束するとき、そのアルゴリズムは学習できているといえる。

ユークリッド直径 D_2 の凸集合上で勾配ノルムが $\|\mathbf{g}_t\| \le G$ と有界であるとき、適切に調整した大域ステップサイズをもつオンライン勾配降下法はオーダー $D_2 G \sqrt{T}$ のリグレットを達成する（演習問題 4 で証明する）。これに対し、AdaGrad は次のオーダーのリグレットを達成する (Duchi et al. 2011)。

$$D_\infty \sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} g_{t,i}^2},$$

ここで D_∞ は ℓ_∞ ノルムにおける集合の直径である。d 個の各座標は、全座標にわたる最悪ケースの勾配ノルムではなく、自身の勾配履歴に応じた分だけを負担すればよい。

ボックス $[-1, 1]^d$ 上では $D_2 = \sqrt{d}\, D_\infty$ が成り立つ。勾配の大きさが少数の座標群に集中するとき、AdaGrad の上界は最大で $\sqrt{d}$ 分の 1 まで小さくなり、すべての座標に均等に分散するときは 2 つの上界は一致する。座標 i に確率 p_i で非ゼロ成分が生じるワンホット勾配では、2 つの上界の期待値の比は高々 $\sum_i \sqrt{p_i}/\sqrt{d}$ である（演習問題 2）。p_i が一様な場合、個々の勾配がいかに疎であってもこの比は 1.00 となるのに対し、指数 1、$d = 10^4$ の Zipf 則に従う単語頻度では 0.63 まで下がる。したがって AdaGrad の優位性は、単なるスパース性（sparsity）そのものではなく、特定の座標への勾配の**集中**から生じる。

Equation 35.51 に現れる平方根は恣意的なものではなく、リグレットの証明において競合する 2 つの項をちょうど釣り合わせる厳密な指数である（演習問題 4c）。

しかしオンライン凸最適化を離れると、上限のない累積が AdaGrad の最大の弱点となる。分散 σ^2 の持続的な勾配ノイズ（gradient noise）のもとでは、累積和が $s_{t,i} \approx \sigma^2 t$ と増大するため、実効ステップサイズは $\eta/(\sigma\sqrt{t})$ のように縮小していく。つまり AdaGrad は、最適化の進捗ではなくステップ数に縛られた減衰スケジュールを内部に抱えている。この $1/\sqrt{t}$ という減衰率は凸リグレットに対しては最適だが、長時間の非凸な訓練では損失がまだ改善中であるかどうかにか

かわらずステップサイズを容赦なく減衰させ、オプティマイザが終盤のプラトーに達する頃にはステップサイズがほとんど残らなくなってしまう。

35.5.1.3 RMSProp

RMSProp (Tieleman and Hinton 2012) は、全期間の累積和を指数移動平均（exponential moving average）に置き換えることでこの減衰を防ぎ、スケール推定が直近の勾配に追随し続けるようにする。

$$\mathbf{v}_t = \beta_2\,\mathbf{v}_{t-1} + (1-\beta_2)\,\mathbf{g}_t^2, \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta}{\sqrt{\mathbf{v}_t} + \epsilon}\,\mathbf{g}_t. \tag{35.52}$$

この漸化式を展開すると、$\mathbf{v}_t$ は k ステップ前の二乗勾配に重み $(1-\beta_2)\beta_2^k$ を割り当てることがわかる。したがって古い勾配の重みは約 $\log 2/(1-\beta_2)$ ステップごとに半減し、RMSProp で慣例的に使われる $\beta_2 = 0.9$ では 7 ステップごと、$\beta_2 = 0.999$ では 693 ステップごとに半分になる。こうして構造的にステップサイズが消失することはなくなる。その代償として、単発の大きな勾配も同じ時間スケールで忘れ去られてしまい、この脆弱性は後述する反例で突くことになる。

35.5.1.4 Adam とバイアス補正

Adam (Kingma and Ba 2015) は、RMSProp の二次モーメント推定（second-moment estimate）に勾配そのものの指数移動平均を組み合わせ、さらに両方の移動平均に対してゼロ初期化に伴う偏りを補正する。9.5 節の記法に従えば、更新式は次のようになる。

$$\begin{aligned} \mathbf{m}_t &= \beta_1\,\mathbf{m}_{t-1} + (1-\beta_1)\,\mathbf{g}_t, & \hat{\mathbf{m}}_t &= \mathbf{m}_t/(1-\beta_1^t), \\ \mathbf{v}_t &= \beta_2\,\mathbf{v}_{t-1} + (1-\beta_2)\,\mathbf{g}_t^2, & \hat{\mathbf{v}}_t &= \mathbf{v}_t/(1-\beta_2^t), \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - \eta\,\frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}, \end{aligned} \tag{35.53}$$

ここで初期値は $\mathbf{m}_0 = \mathbf{v}_0 = \mathbf{0}$ であり、標準のデフォルト値は $\beta_1 = 0.9$、$\beta_2 = 0.999$、$\epsilon = 10^{-8}$ である。一次モーメント（first moment）$\mathbf{m}_t$ は 35.1.3 節のヘビーボールの速度を $1-\beta_1$ 倍したものであり、これによって累積和が重み付き平均へと変わり、通常の RMSProp がそのままステップに反映してしまうミニバッチノイズ（minibatch noise）を平滑化する。

$1-\beta_2^t$ で割る操作は、$\mathbf{v}_0 = \mathbf{0}$ から出発することを補償する。漸化式を展開すると $\mathbf{v}_t = (1-\beta_2)\sum_{\tau \le t} \beta_2^{t-\tau}\,\mathbf{g}_\tau^2$ となる。二乗勾配のスケールが定常であり、$\mathbb{E}[\mathbf{g}_\tau^2] = \overline{\mathbf{g}^2}$ がすべての τ について成り立つならば、等比級数の和をとることで次式が得られる。

$$\mathbb{E}[\mathbf{v}_t] = (1-\beta_2)\,\overline{\mathbf{g}^2}\sum_{\tau=1}^{t}\beta_2^{t-\tau} = (1-\beta_2^t)\,\overline{\mathbf{g}^2}. \tag{35.54}$$

したがって定常性のもとでは、$\hat{\mathbf{v}}_t$ が $\overline{\mathbf{g}^2}$ の不偏推定量であることがすべてのステップ t で成り立ち、β_1 についても同様の計算から $\hat{\mathbf{m}}_t$ が平均勾配の不偏推定量であることがわかる。

この補正がない場合、ゼロ初期化による不足分は大きく、しかも長く残る。デフォルトの $\beta_2 = 0.999$ では、$\mathbf{v}_{10}$ は定常時の目標値の約 1% しか捉えられず、$t = 1000$ に達してもまだ 63% にすぎない（9.5 節）。さらに注目すべきは、その誤差が時間とともにどう推移するかである。両方のバイアス補正（bias correction）を省くと、更新量 $\mathbf{m}_t/\sqrt{\mathbf{v}_t}$ のスケールは係数

$(1-\beta_1^t)/\sqrt{1-\beta_2^t}$ だけずれ、最初は 3.16（$t=1$）から始まり、一見意外なことにピークの 6.6（$t=12$）まで上昇する。一次モーメントの移動平均は約 10 ステップで立ち上がるのに対し、二次モーメント（second moment）の移動平均は約 1,000 ステップを要するためである。Figure 35.18 に、この不足分とそれによって生じるステップの膨張の両方を示す。

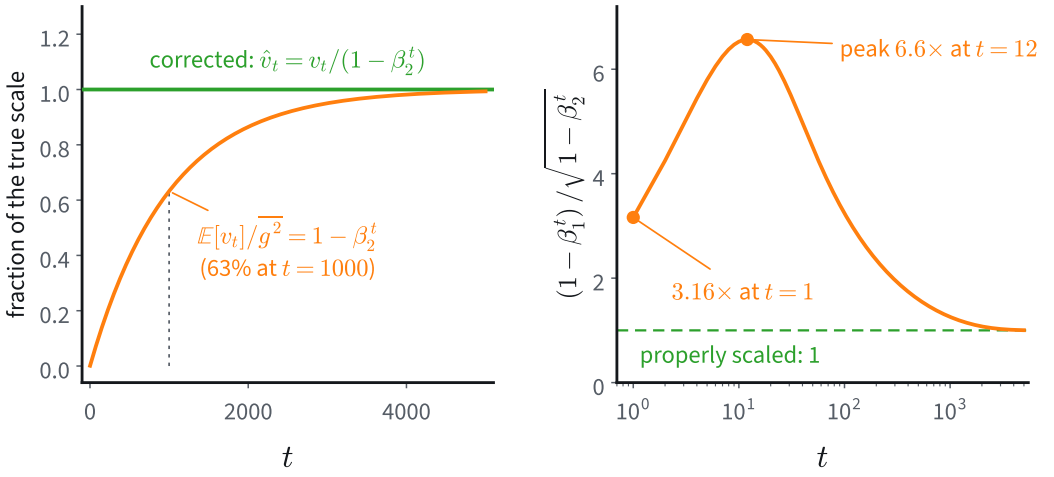


Figure 35.18: Adam におけるバイアス補正。左：ゼロから出発すると、移動平均 $\mathbf{v}_t$ は二乗勾配スケールのうち割合 $1-\beta_2^t$ しか捉えられない（63% に達するのに $t=1000$ を要する、$\beta_2=0.999$ の場合）が、$1-\beta_2^t$ で割ればすべてのステップで本来のスケールが復元される。右：バイアス補正を行わないと、更新量 $\mathbf{m}_t/\sqrt{\mathbf{v}_t}$ は $(1-\beta_1^t)/\sqrt{1-\beta_2^t}$ 倍に膨張し、3.16（$t=1$）から始まり、分子が 10 ステップで立ち上がる一方、分母は 1,000 ステップを要するため 6.6（$t=12$）でピークに達し、1 に戻るのは $1/(1-\beta_2)$ の時間スケールになってからである。したがって補正なしの Adam は、前処理行列が最も少ない勾配しか反映していないときに最大のステップを踏んでしまう。

$\hat{\mathbf{v}}_t$ が不偏であっても、ステップサイズ自体が不偏になるわけではない。Adam は $\sqrt{\hat{\mathbf{v}}_t}$ で割り、$\hat{\mathbf{v}}_t$ では割らないため、平方根の凹性により分母に下向きの統計的バイアスが生じる。すなわちイェンセンの不等式（Jensen's inequality）より、$\mathbb{E}\sqrt{\hat{v}_t} < \sqrt{\mathbb{E}[\hat{v}_t]}$ が（$\hat{v}_t$ が定数でない限り）成り立つ。最初のステップはその極端な例である。$t=1$ では $\hat{m}_1 = g_1$ かつ $\sqrt{\hat{v}_1} = |g_1|$ となるため、比は $\mathrm{sign}(g_1)$ となり、ステップの大きさは勾配がいかなる値であってもすべての座標で η となる（$\eta\,|g_1|/\sigma$ のように勾配のスケール σ で正規化された値にはならない）。これがどの程度影響するかは 35.5.4 節で検証する。

35.5.1.5 スケール不変性

冒頭の例題では $h_2 = 1000\,h_1$ であったが、Adam はまるでその 1,000 倍の比が存在しないかのように谷を横断する。その理由は更新式がもつ厳密な対称性にあり、これこそが本節の中心的な結果である。

命題（Adam は各座標のスケールを無視する）。 $\epsilon = 0$ **とし、** $\mathbf{C} = \mathrm{diag}(c_1, \ldots, c_d)$ **（すべての** $c_i > 0$**）とする。2 つの問題の勾配がすべての点において** $\nabla \tilde{f}(\mathbf{x}) = \mathbf{C}\,\nabla f(\mathbf{x})$ **（任意の** $\mathbf{x}$**）という関係にあるとする。このとき、同一の** β_1**、**β_2**、ステップスケジュール、および初期点をもつ Adam は、更新が定義され続ける限り、両方の問題上で同一の反復点列を生成する。**

言い換えれば、2 つ目の問題は 1 つ目の問題において座標 i の勾配を正の定数 c_i 倍に再スケーリ

ングしたものである。その問題に関する量にはチルダを付けて表す。

証明。 t に関する帰納法で示す。反復点が $\mathbf{x}_t$ まで一致していると仮定する。すると $\tilde{\mathbf{g}}_\tau = \mathbf{C}\mathbf{g}_\tau$ がすべての $\tau \le t$ に対して成り立つ。$\mathbf{m}$ と $\mathbf{v}$ の漸化式はそれぞれ $\mathbf{g}_\tau$ と $\mathbf{g}_\tau^2$ に関して線形であるため、$\tilde{m}_{t,i} = c_i m_{t,i}$ および $\tilde{v}_{t,i} = c_i^2 v_{t,i}$ が得られる。バイアス補正の分母は両方の問題で同じスカラーであるため、2 つ目の問題における補正後の比 $\hat{m}_{t,i}/\sqrt{\hat{v}_{t,i}}$ は $c_i \hat{m}_{t,i}/(c_i\sqrt{\hat{v}_{t,i}}) = \hat{m}_{t,i}/\sqrt{\hat{v}_{t,i}}$ となる。したがって更新量は一致し、$\tilde{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1}$ となる。■

対角二次関数では、$\nabla f(\mathbf{x}) = \operatorname{diag}(h)\,\mathbf{x}$ は丸いボウル $\frac{1}{2}\|\mathbf{x}\|^2$ の勾配に $\mathbf{C} = \operatorname{diag}(h)$ を掛けたものにほかならない。その結果、1000:1 の谷における Adam は、1 ステップごとに丸いボウル上の Adam とまったく同じ軌跡（trajectory）をたどり、しかも丸いボウル上では等しい値から出発した 2 つの座標は永遠に等しいままである。各座標は $\frac{1}{2}x^2$ 上の同じ 1 次元の反復に従う。

$$x_{t+1} = x_t\Big(1 - \frac{\eta}{\sqrt{\hat{v}_t}}\Big) \qquad (\beta_1 = 0),$$

ここで $\hat{v}_t$ は過去の二乗 x_τ^2（$\tau \le t$）の移動平均である。初期の反復点のほうが大きいため、2 ステップ目以降は $\sqrt{\hat{v}_t}$ が $|x_t|$ を上回り、各更新の移動量は η 未満となって縮小率 $1 - \eta/\sqrt{\hat{v}_t}$ で幾何級数的に縮小する。$\eta = 0.1$ のとき、この縮小率は 0.9（$t = 1$）から 0.67（$t = 48$）へと、$\hat{v}_t$ が縮小していく反復点に追いつくにつれて強まっていく。反復点そのものは h_2/h_1 にまったく依存しない。曲率比に依存するのは停止判定だけであり、それは $f(\mathbf{x}_t) = \frac{1}{2}(h_1 + h_2)\,x_t^2$ が共通の反復点を各曲率で重み付けするためである。したがって、谷をはるかに細くしても幾何級数的な縮小ステップが数回増えるだけで済み、κ 倍に膨らむことは決してない。

一方でステップ数が**実際に**依存するのは、η を単位として測った初期距離である。曲率にかかわらず、序盤のステップは座標あたりおよそ η の長さをもつからである。次項の実験では、これら両方の予測を直接検証する。

要するに、Adam はすべての座標をその座標自身の勾配の大きさを単位として測るため、座標ごとのスケーリングは Adam からは見えなくなる。ただし、この不変性には 3 つの留保条件がある。第 1 に、$\epsilon > 0$ の場合、ある座標の勾配が ϵ のオーダーまで縮小すると対称性が破れる。第 2 に、この命題は座標を混ぜ合わせる基底変換については何も述べておらず、これについては次項で回転した谷を用いて検証する。第 3 に、Adam は一般には符号降下法（sign descent）ではない。更新が**符号ステップ**（sign step）$\eta\,\operatorname{sign}(\mathbf{g}_t)$ に帰着するのは、$t = 1$ のとき、$\beta_1 = \beta_2 = 0$ のとき（演習問題 5d）、および $\beta_1 = 0$ で各座標の勾配の大きさが $\hat{\mathbf{v}}$ の記憶期間にわたって一定に保たれるときの 3 つの特別な場合に限られる。

35.5.2 座標スケーリングとその限界

35.5.2.1 軸に沿った谷と回転した谷

勾配降下法が要した 6,160 ステップのうち、座標スケーリング単独でどれだけ削減できるのか、また谷を回転させたときにもその利得は保たれるのだろうか。座標ごとのスケーリングとモメンタムを切り分けるため、4 つの手法を比較する。最適な単一ステップサイズを用いた勾配降下法、最適なステップサイズとモメンタムを用いたヘビーボール法、再スケーリングのみを行う $\beta_1 = 0$ の Adam、そしてデフォルトのモメンタム $\beta_1 = 0.9$ をもつ Adam である。問題を角度 θ だけ回転させるには、$\operatorname{diag}(h)$ を $\mathbf{A}_\theta = \mathbf{R}_\theta \operatorname{diag}(h)\,\mathbf{R}_\theta^\top$（$\mathbf{R}_\theta$ は 2×2 回転行列）に置き換え、初期点もそれに合わせて $\mathbf{x}_0 = \mathbf{R}_\theta(1, 1)^\top$ と回転させることで、初期点が両方の曲率軸に沿って同じ成分を保つようにする。$\theta = \pi/4$ では $\mathbf{A} = \left(\begin{smallmatrix} 500.5 & -499.5 \\ -499.5 & 500.5 \end{smallmatrix}\right)$ となる。以下のコードでは、谷の目的関数、目標許容誤差に達するまでのステップ数を数えるループ、および 3 種類のオプティマイザの更新式を定義する。

```
h = np.array([1.0, 1000.0])                    # curvatures h_i, kappa = 1000

def rotated(theta):
    """A = R diag(h) R^T and the start R (1, 1): the valley turned by theta."""
    c, s = np.cos(theta), np.sin(theta)
    R = np.array([[c, -s], [s, c]])
    return R @ np.diag(h) @ R.T, R @ np.ones(2)

def run(step, A, x0, K=40000, tol=1e-8):
    """First t with f(x_t) = x_t^T A x_t / 2 < tol, or None within K steps."""
    x, state = x0.copy(), {}
    for t in range(1, K + 1):
        x = step(x, A @ x, t, state)
        if 0.5 * x @ A @ x < tol:
            return t
    return None

def gd(eta):
    return lambda x, g, t, state: x - eta * g

def heavy_ball(eta, beta):
    def step(x, g, t, state):
        state['d'] = beta * state.get('d', 0.0) - eta * g    # velocity
        return x + state['d']
    return step

def adam(eta, beta1=0.9, beta2=0.999, eps=1e-8):
    def step(x, g, t, state):
        state['m'] = beta1 * state.get('m', 0.0) + (1 - beta1) * g
        state['v'] = beta2 * state.get('v', 0.0) + (1 - beta2) * g * g
        m_hat, v_hat = state['m'] / (1 - beta1**t), state['v'] / (1 - beta2**t)
        return x - eta * m_hat / (np.sqrt(v_hat) + eps)
    return step
```

ステップ数を見る前に、各手法にどのような情報が与えられているかに注目しよう。勾配降下法とヘビーボール法は、真の曲率 1 と 1000 から閉じた式で計算した理論上の最適設定で実行する。実際のニューラルネットワークで両端の極端な曲率を知る術はないため、これはかなり有利なハンディキャップである。一方、Adam には閉じた公式がないため、実務者が行うのと同じく 0.002 から 2 まで対数等間隔に並べた 7 個のステップサイズ（約 3 倍刻み）でスイープし、各行の最良値を読み取る。なお、このグリッドは意図的に $\eta = 1$ を避けている。Adam の最初のステップはすべての座標で長さ η をもつため（35.5.4 節）、$(1, 1)$ から $\eta = 1$ で出発すると 1 ステップで直接原点に着地してしまうからである。それは初期値の偶然による一致であり、オプティマイザの速度の公平な尺度にはならない。

```
mu, L = h.min(), h.max()
rho = (np.sqrt(L) - np.sqrt(mu)) / (np.sqrt(L) + np.sqrt(mu))
tuned = {'GD, eta = 2/(mu + L)': gd(2 / (mu + L)),
         'heavy ball, tuned': heavy_ball(4 / (np.sqrt(L) + np.sqrt(mu))**2,
                                         rho**2)}
valleys = {'aligned': rotated(0.0), 'rotated 45': rotated(np.pi / 4)}
print(f'{"steps to f < 1e-8":28s}{"aligned":>9s}{"rotated 45":>12s}')
```

```
    for name, step in tuned.items():
        hits = [run(step, *problem) for problem in valleys.values()]
        print(f'{name:28s}{hits[0] or "never":>9}{hits[1] or "never":>12}')
    etas = np.geomspace(0.002, 2.0, 7)              # step sizes for Adam
    print(f'\n{"Adam, eta =":28s}' + ''.join(f'{eta:>7.2g}' for eta in etas))
    for beta1 in (0.0, 0.9):
        for name, problem in valleys.items():
            hits = [run(adam(eta, beta1=beta1), *problem) for eta in etas]
            print(f'{f"beta1 = {beta1}, {name}":28s}'
                  + ''.join(f'{k or "never":>7}' for k in hits))
```

```
steps to f < 1e-8             aligned  rotated 45
GD, eta = 2/(mu + L)             6160        6160
heavy ball, tuned                 296         296

Adam, eta =                  0.002 0.0063   0.02  0.063    0.2   0.63      2
beta1 = 0.0, aligned          2122    760    249     78     22  never  never
beta1 = 0.0, rotated 45      never  never  never  never  never  never  never
beta1 = 0.9, aligned          2033    648    195    165    205    170    196
beta1 = 0.9, rotated 45      12862   9292   5898   3171   1581    799    559
```

まず軸に沿った谷を見てみよう。座標ごとのスケーリング単独（$\beta_1 = 0$）では、最良のステップサイズにおいてわずか 22 ステップしか要しない。勾配降下法の 6,160 ステップやヘビーボール法の 296 ステップと比べると、真の曲率を 2 つとも与えられたヘビーボール法より 13 倍少ないステップ数を、どちらの曲率も知らされていないオプティマイザが達成している。この行からは、Adam の調整が実際には何を合わせているのかも明らかになる。反復回数はほぼ $1/\eta$ に比例して変化する。序盤のステップ長が約 η であり、総ステップ数は主にステップ長を単位として測った最小点までの距離を反映するからである。ただし最大の 2 つのステップサイズでは、スケーリング単独では決して収束せず、反復点は最小点を飛び越えて各座標が $+\eta/2$ と $-\eta/2$ の間で跳ね返る周期軌道に陥る。つまり Adam は、曲率という未知量を、η を単位とする最小点までの距離という別の未知量と引き換えにしているのである。

Adam のデフォルトのモメンタム（$\beta_1 = 0.9$）を加えると、ステップサイズが 0.02 から 2 までの全範囲にわたり、反復回数は 165〜205 ステップに収まる。ここで際立つ特徴が 2 つある。1 つ目は頑健性（robustness）である。η を 100 倍変えても反復回数は 4 分の 1 程度しか動かず、モメンタムなしでは永久に振動していた最大の 2 つのステップサイズでも、モメンタムの移動平均が交互に反転する符号を減衰させるためきれいに収束する。ヘビーボール法にはそのような余裕（マージン）はない。その最適ステップサイズは安定限界 $(2 + 2\beta)/\lambda_{\max}$ の 0.1% 以内に位置しており（演習問題 8a、35.1 節）、296 ステップという結果は最大曲率をほぼ厳密に知っていることを前提としている。2 つ目の特徴は、その頑健性の代償である。モメンタムによって、Adam はスケーリング単独の場合より約 7 倍遅くなっている。その原因は $\sqrt{\beta}$ の縮小限界（35.1.3 節）にある。モメンタム β をもつ反復は、複素根をもつ任意のモードを 1 ステップあたり $\sqrt{\beta}$ の速さで縮小させ、それ以上速くはならない。座標ごとのスケーリングを施した後は、両座標とも実質的に条件数 1 の丸いボウル上を走るため、そこでの最適なモメンタムはゼロである。$\beta_1 = 0.9$ では 1 ステップあたりの縮小率は $\sqrt{0.9} \approx 0.95$ となり、許容誤差に達するまで約 230 ステップと予測され、実測値とよく一致する。悪条件（ill-conditioned）な谷に沿う進行を加速するモメンタムも、丸いボウルの上ではブレーキとして働くにもかかわらず、Adam のデフォルト設定はこれを無条件に適用する。

次に回転した谷に目を向けよう。勾配降下法とヘビーボール法は、更新式が回転不変であり問題の回転に伴ってそのまま回転するため、以前とまったく同じステップ数で収束する。しかし Adam の更新式はそうではない。スケーリング単独では、グリッド上のどのステップサイズでも

許容誤差に到達できない。モメンタム付きの Adam は収束するものの、グリッドの上端付近に限られる。最良のステップ数は $\eta = 2$ における 559 ステップでヘビーボール法の 296 ステップのほぼ 2 倍を要し、最小のステップサイズでは 12,862 ステップと通常の勾配降下法の 2 倍ものステップ数を要する。ステップサイズに対する頑健性も失われ、グリッド全体でのステップ数は 20 倍以上の開きを見せる。

したがって公平な結論は表裏二面をもつ。座標軸が主曲率方向に一致しているとき、座標ごとのスケーリングははるかに少ない事前知識でありながら、完璧に調整されたヘビーボール法に 1 桁の差をつけて勝つ。一方、谷が回転すると対角スケーリングの恩恵は完全に消え失せ、残された収束性はすべてモメンタムに由来するものとなり、その役割であれば適切に調整されたヘビーボール法のほうが 2 倍速くこなす。この両面をさらに詳しく見ていこう。

軸に沿った谷では、スケール不変性の命題により、硬い（曲率の大きい）座標が 1,000 分の 1 の実効ステップを受け取る一方で、両座標は同一の反復点をたどると予測される。以下では、反復点と実効ステップサイズ $\eta/\sqrt{\hat{\mathbf{v}}_t}$ をスケーリングのみの Adam（$\epsilon = 0$、$\eta = 0.1$）について出力する。さらに最後の 2 行では、曲率比を 5 桁にわたって変化させた場合と、初期点を原点から 100 倍遠くへ離した場合を比較し、反復回数を実際に支配している要因を検証する。

```
A, x = rotated(0.0)
step, state, eta, gap = adam(0.1, beta1=0.0, eps=0.0), {}, 0.1, 0.0
for t in range(1, 49):
    x = step(x, A @ x, t, state)
    gap = max(gap, abs(x[0] - x[1]))
    if t in (1, 10, 48):
        eff = eta / np.sqrt(state['v'] / (1 - 0.999**t))
        print(f't = {t:2d}: x = {x[0]:.3g}, {x[1]:.3g};  '
              f'eta/sqrt(v_hat) = {eff[0]:.3g}, {eff[1]:.3g}')
print(f'largest |x_1 - x_2| over the run: {gap:.1e}')
counts = [run(adam(0.1, beta1=0.0), np.diag([1.0, r]), np.ones(2))
          for r in (1.0, 1e3, 1e5)]
print('steps to f < 1e-8 at curvature ratios 1, 1e3, 1e5:', *counts)
print('steps from (100, 100) at ratio 1e3:',
      run(adam(0.1, beta1=0.0), A, 100 * np.ones(2)))
```

```
t =  1: x = 0.9, 0.9;  eta/sqrt(v_hat) = 0.1, 0.0001
t = 10: x = 0.262, 0.262;  eta/sqrt(v_hat) = 0.152, 0.000152
t = 48: x = 3.91e-06, 3.91e-06;  eta/sqrt(v_hat) = 0.329, 0.000329
largest |x_1 - x_2| over the run: 5.6e-17
steps to f < 1e-8 at curvature ratios 1, 1e3, 1e5: 40 48 54
steps from (100, 100) at ratio 1e3: 4145
```

最初の 1 ステップ目から、実効ステップサイズは正確に $1000{:}1$ の比をなしており、ヘッセ行列を一度も作ることなく勾配の大きさだけから真の曲率比が復元されている。2 つの座標は終始、浮動小数点精度（floating-point precision）の範囲で一致し、平坦な座標の実効ステップは 0.1 から 0.33 へと、縮小する勾配に $\hat{v}_t$ が適応するにつれて大きくなる。曲率比を 1 から 10^3、さらに 10^5 へと引き上げても、ステップ数は 40 から 48、54 へと増えるにすぎない。条件性が 10 万倍悪化しても、追加コストはわずか 14 ステップである。ところが初期点を 100 倍遠くに移すと、ステップ数は 4,145 まで跳ね上がる。反復回数は初期距離を η 単位で測っており、曲率の影響はほとんど受けないのである。

なぜ 45° の回転がこのすべてにとって致命的なのだろうか。回転後の行列は対角成分が 500 付近となり、急な方向 $(-1, 1)/\sqrt{2}$ が両方の座標軸に等しく射影される。そのため各座標の勾配は急なモードに支配され、両座標ともほぼ同一のスケール推定値を蓄積することになる（実行全体を

通じて両者の $\hat{v}$ は 1% 以内で一致する）。座標ごとの前処理行列は各座標に 1 つのスカラーしか割り当てられないため、ある座標の勾配に含まれる急な成分を縮小しようとすれば、平坦な成分も同じだけ縮小せざるを得ない。両座標のスケールが等しくなると、前処理行列は単位行列のスカラー倍に退化する。こうしてスケーリングのみの Adam（$\beta_1 = 0$）は、ステップサイズが急なモードに縛られた勾配降下法に先祖返りし、平坦なモードは這うようにしか進まなくなる。また $\beta_1 = 0.9$ の Adam はヘビーボール法に先祖返りする。モメンタムは最適値の 0.88 に近いものの、実効ステップ $\eta/\sqrt{\hat{v}_t}$ が曲率ではなく勾配履歴に支配されるため、グリッド上の最良点であっても調整済みヘビーボール法の 2 倍のステップ数を要する。

モメンタムがない場合、回転した谷での実行は最終的に振動に陥る。初めのうち急な成分が縮小するにつれて $\hat{\mathbf{v}}$ が減少し、実効ステップサイズが増大してやがて急なモードの安定限界 $2/h_{\max}$ を超える。軸に沿った谷での最適ステップサイズである $\eta = 0.2$ では、損失は一旦 10^{-3} 付近まで下がる（$t = 1{,}700$ 前後）ものの、その時点で急なモードが振動を始める。この定常的な振動によって勾配の大きさが一定に固定されると $\hat{\mathbf{v}}_t \to \mathbf{g}_t^2$ となり、更新は符号ステップ $\eta\,\mathrm{sign}(\mathbf{g}_t)$（谷を横切る長さ $\eta\sqrt{2}$）へと退化する。こうして反復点は振幅 $\eta/\sqrt{2}$、損失 $f = 250\,\eta^2 = 10$ の 2 周期軌道に永久に閉じ込められる（演習問題 5 ではステップサイズのグリッド全体でこの式を検証する）。モメンタムはこの交互に反転するステップを平均化して打ち消すため、$\beta_1 = 0.9$ の実行はやがて目標許容誤差に到達する。Figure 35.19 は、見やすく描画できる程度に緩やかな谷の上で、これら 3 つの挙動を示したものである。

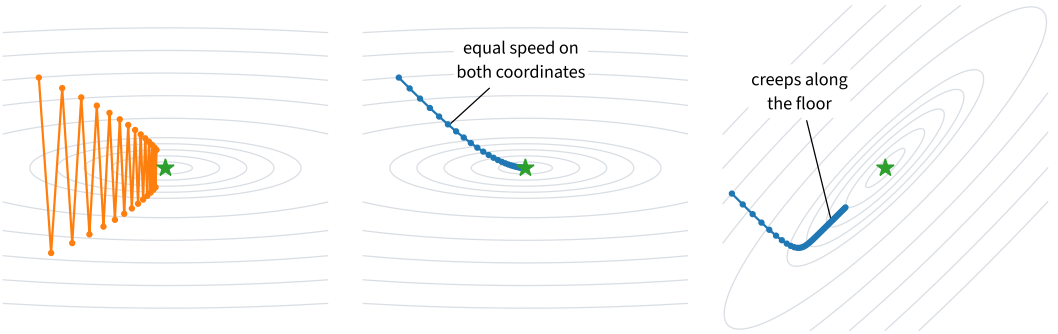


Figure 35.19: 同じ初期点から出発した 2 次元の谷（$\kappa = 20$。本文の実験では等高線が細すぎて描画できない $\kappa = 10^3$ を使用）における比較。(a) 安定限界付近の単一ステップサイズを用いた勾配降下法は、急な方向に振動しながら平坦な方向には這うように進む。(b) 座標ごとの正規化（$\beta_1 = 0$ の Adam）は両座標を同じ速さで動かし、滑らかな曲線を描いて最小点へ収束する。(c) 谷と初期点を 45° 回転させると座標軸と曲率方向がずれ、反復点は谷底に沿ってわずかにしか進めなくなる（40 ステップ時点で $f = 0.19$ であるのに対し、(b) では 4×10^{-7}）。

座標ごとのスケーリングが悪条件性を解消するのは、座標軸が曲率の固有ベクトルと一致するとき、かつそのときに限る。スケーリングは対角行列（diagonal matrix）によって計量を変える操作であり、いかなる対角行列も座標が混ざり合う方向の曲率を捉えることはできない。外積（outer product）$\mathbf{g}_\tau \mathbf{g}_\tau^\top$ から構成したフル行列であれば問題の回転に伴って回転するためこの欠陥を免れるが、大規模なニューラルネットワークではその保持と逆行列計算のコストが高すぎて実用的ではない。本節末尾で扱う構造化前処理行列は、座標を混合するコンパクトな因子を用いることで、重み行列におけるこの限界に対処する。

35.5.2.2 Adam に対する凸問題の反例

AdaGrad が過去の勾配を無期限に記憶するのに対し、RMSProp と Adam は過去の勾配を忘れていく。この忘却によって、単純な凸問題でさえ解けなくなることがあるのだろうか。Reddi et al.

(2018) は、凸な損失の 1 次元周期系列を用いて、実際に起こり得ることを示した。

命題（Adam は収束するとは限らない）。 **$C > 2$ とし、$x \in [-1, 1]$ 上での次の周期的な損失によるオンライン最適化を考える。**

$$f_t(x) = \begin{cases} C\,x, & t \bmod 3 = 1, \\ -x, & \text{otherwise}. \end{cases} \tag{35.55}$$

バイアス補正なしの Adam を $\beta_1 = 0$、$\beta_2 = 1/(1 + C^2)$、ステップサイズ $\eta_t = \eta/\sqrt{t}$（ただし $\eta < \sqrt{1 - \beta_2}$）とし、各反復点を $[-1, 1]$ 上へ射影（projection）しながら $x_1 = 1$ から実行する。このとき、すべてのステップで $x_t > 0$（任意の t）かつ $x_{3k+1} = 1$（すべての $k \geq 0$）となる。3 ステップの各周期がリグレットを少なくとも $2C - 4 > 0$ 増加させるため、平均リグレットがゼロに収束することはない。 (Reddi et al. 2018)

ここで $[-1, 1]$ への射影とは単なるクリッピングである（一般的な射影は 35.6 節で扱う）。3 ステップの各周期にわたる勾配の和は $C - 2 > 0$ であるため、事後的に最良の固定点は $x^\star = -1$ である。それにもかかわらず、Adam は毎周期の開始時点で区間内の最悪点である $x = +1$ に戻ってしまう。各周期で Adam は少なくとも $C - 2$ の損失を被る（最初のステップでは $x = 1$ にいるため C を被り、残る 2 ステップでもそれぞれ少なくとも -1 を被る）一方、$x^\star = -1$ での損失は $-(C - 2)$ であるため、1 周期あたり少なくとも $2C - 4$ のリグレット増分が生じる。証明は周期に関する帰納法で進み、演習問題 3 ではその要となるステップを確認する。そこでは $\eta < \sqrt{1 - \beta_2}$ という条件により、下向きのステップが $(0, 1]$ の内部にとどまることが保証される。また、この破綻は $\beta_1 = 0$ と置いたことによる特殊な現象ではない。同論文は $\beta_1 < \sqrt{\beta_2}$ を満たす任意の固定ペアに対し、Adam が収束に失敗するオンライン凸系列と確率的凸問題の両方を構成している。

$C = 4$、$\beta_2 = 1/17$ とし、v が定常的な周期サイクルに落ち着いた後の 1 周期を見れば、その仕組みを直接確認できる。勾配 $+4, -1, -1$ に伴い、二次モーメント推定値 v は 15.12、1.83、1.05 という値をとり、$\sqrt{v}$ は 3.89、1.35、1.02 となる。その結果生じる 3 回の変位 $-\eta g_t/\sqrt{v_t}$ は次のようになる。

$$-1.03\,\eta, \qquad +0.74\,\eta, \qquad +0.98\,\eta,$$

これにより、1 周期あたり $+0.69\,\eta$ の正味のドリフト（drift）が $+1$ に向かって生じる。正規化によってまれな大勾配 $+C$ による移動がおよそ η（大きさ 1 の勾配による移動とほとんど変わらない大きさ）に頭打ちにされるうえ、v は 2 ステップのうちに C^2 のスパイクを忘れてしまう。そのため、2 回の -1 の勾配が x を押し上げる量の合計が、1 回の $+C$ の勾配が引き下げる量を上回るのである。これに対し、ステップ $\eta g_t/C$ をもつ通常の SGD は $-\eta$、$+0.25\,\eta$、$+0.25\,\eta$ だけ動き、最適点に向かって正味 $-0.5\,\eta$ 進む。定数ステップ SGD のノイズフロア（35.4 節）とは異なり、Adam の逆方向のドリフトはステップサイズを縮小しても消えず、あらゆるステップサイズで $+1$ の方向を向く。

この欠陥を修正するため、Reddi et al. (2018) は **AMSGrad** を提案した。AMSGrad は分母の $\mathbf{v}_t$ を、補正なし移動平均の累積最大値 $\tilde{\mathbf{v}}_t = \max(\tilde{\mathbf{v}}_{t-1}, \mathbf{v}_t)$ に置き換える。これにより実効ステップ $\eta_t/\sqrt{\tilde{v}_t}$ が非増加となるため、大きな勾配が決して忘れられなくなる。先ほどの周期的な例では、$\tilde{v}$ が終始 15.12 に固定されるため、3 回の変位は $-1.03\,\eta$、$+0.26\,\eta$、$+0.26\,\eta$ へと変わり、最適点に向かう正味 $-0.51\,\eta$ の移動となる。したがって Adam は $+1$ へ収束する一方、AMSGrad と SGD は -1 へ収束するはずである。以下のセルでは、同じ $1/\sqrt{t}$ 減衰のもとで、Adam が最初から悪い端点に置かれることのないよう区間の中点 $x = 0$ から出発させて 3 手法すべてを検証する。

```
def online(update, C=4.0, T=15000, alpha=0.5):
    """Projected iterates on the periodic losses; beta_1 = 0, no bias correction."""
    beta2 = 1 / (1 + C * C)
    x, v, vmax, xs = 0.0, 0.0, 0.0, []
    for t in range(1, T + 1):
        g = C if t % 3 == 1 else -1.0            # rare +C, frequent -1
        v = beta2 * v + (1 - beta2) * g * g
        vmax = max(vmax, v)                      # running maximum (AMSGrad)
        x = min(1.0, max(-1.0, x - alpha / np.sqrt(t) * update(g, v, vmax, C)))
        xs.append(x)
    return xs

updates = {'adam': lambda g, v, vmax, C: g / (np.sqrt(v) + 1e-8),
           'amsgrad': lambda g, v, vmax, C: g / (np.sqrt(vmax) + 1e-8),
           'sgd': lambda g, v, vmax, C: g / C}
print('x_t at t =        30      300     3000    15000   (x* = -1)')
for name, update in updates.items():
    xs = online(update)
    print(f'{name:9s}' + ''.join(f'{xs[t - 1]:9.3f}' for t in (30, 300, 3000, 15000)))
```

```
x_t at t =        30      300     3000    15000   (x* = -1)
adam          0.745    1.000    1.000    1.000
amsgrad      -0.953   -0.985   -0.995   -0.998
sgd          -0.954   -0.986   -0.995   -0.998
```

ここで alpha は命題における η の役割を果たしており、$\eta = 0.5 < \sqrt{16/17}$ であるためステップサイズに関する仮定を満たしている。区間の中点から出発してもなお、Adam は $+1$ へ一直線に向かう。$t \approx 300$ 以降は毎周期 $+1$ に戻り、0.97 を下回ることすらなくなる。その一方で、AMSGrad と SGD は最適点 -1 へと着実に近づき、小数第 2 位まで一致する。この反例は Adam に一般的な凸問題での収束保証（convergence guarantee）がないことを証明するものであるが、典型的な深層学習問題での失敗を予測するものとしてではなく、あくまで理論的な存在結果として受け止めるべきである。

35.5.3 分離型重み減衰

35.5.3.1 結合型減衰と分離型減衰

現代のほぼすべての訓練レシピ（training recipe）には重み減衰が含まれており、そのため適応的オプティマイザには「減衰項を前処理行列に通すべきか」という設計上の選択が迫られる。SGD のもとでは、この区別は存在しない。損失に ℓ_2 ペナルティ $\frac{\lambda}{2}\|\mathbf{w}\|^2$ を加えると勾配に $\lambda\mathbf{w}$ が加わり、得られる更新 $\mathbf{w}_t - \eta\,(\mathbf{g}_t + \lambda \mathbf{w}_t) = (1 - \eta\lambda)\,\mathbf{w}_t - \eta\,\mathbf{g}_t$ は、損失に対する勾配ステップであると同時に重みの一様な縮小（shrinkage）でもある（9.5.3 節、ただし同節では重みを $\mathbf{x}$ と表記している）。しかし Adam の内部では、これら 2 つの仕組みは乖離する。ペナルティ付きの勾配を Equation 35.53 に入力して減衰項を取り出すと（見通しをよくするため $\beta_1 = 0$ と置けば更新量は $-\eta\,(\mathbf{g}_t + \lambda\mathbf{w}_t)/(\sqrt{\widehat{\mathbf{v}}_t} + \epsilon)$ となる）、座標 i の 1 ステップあたりの縮小量は次のようになる。

$$\text{coupled } (\ell_2 \text{ through Adam}): \quad \frac{\eta\,\lambda}{\sqrt{\widehat{v}_{t,i}} + \epsilon}\, w_{t,i} \qquad \text{versus} \qquad \text{decoupled:} \quad \eta\,\lambda\, w_{t,i}. \tag{35.56}$$

結合型（coupled）の Adam では、前処理行列が損失の勾配とペナルティの勾配の両方を割るため、各座標の縮小率はその勾配スケールに反比例する。すなわち、大きな勾配をもつ重みはほとんど減衰しない一方、勾配の小さい静かな重みは強く減衰させられる。**AdamW** (Loshchilov and Hutter 2019) はこの 2 つの操作を分離（decoupled）し、損失の勾配のみを前処理行列に通して、重み減衰はパラメータに直接作用させる。

$$\mathbf{w}_{t+1} = (1 - \eta\lambda)\,\mathbf{w}_t - \eta\,\frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}. \tag{35.57}$$

これにより、SGD とまったく同様にすべての座標が毎ステップ同じ係数 $1 - \eta\lambda$ で縮小し、モーメントの移動平均は損失の勾配のみを追跡するようになる。

`torch.optim.Adam(weight_decay=...)` は勾配に $\lambda\mathbf{w}$ を加えるため、結合型の更新則を実装している。`torch.optim.AdamW` は分離型の更新則 Equation 35.57 を実装している。

Equation 35.56 の違いが最も鮮明に現れる検証は、損失の勾配が平均ゼロの純粋なノイズであり、重み減衰だけが系統的なドリフトとなる重みを調べることである。ノイズの標準偏差が σ であれば $\sqrt{\hat{v}} \approx \sigma$ となるため、結合型 Adam では重みが約 $e^{-\eta\lambda T/\sigma}$（$T$ ステップ後）まで縮小すると予測されるのに対し、AdamW ではすべての重みが $(1 - \eta\lambda)^T$ へと σ によらず縮小するはずである。以下では、10^4 個の重み（ノイズスケール $\sigma = 10$）と別の 10^4 個の重み（$\sigma = 0.1$）をすべて $w = 1$ に初期化し、各手法を 4,000 ステップ（$\eta = 10^{-3}$、$\lambda = 0.1$）実行して、グループごとの平均値をこれらの予測値と比較する。

```
sigma, eta, lam, T, n = np.array([[10.0], [0.1]]), 1e-3, 0.1, 4000, 10_000
rng = np.random.default_rng()

def decay(decoupled, beta1=0.9, beta2=0.999, eps=1e-8):
    """Mean of w over n weights per noise scale after T Adam steps from w = 1."""
    w, m, v = np.ones((2, n)), np.zeros((2, n)), np.zeros((2, n))
    for t in range(1, T + 1):
        g = sigma * rng.standard_normal((2, n))     # loss gradient: pure noise
        if decoupled:
            w = (1 - eta * lam) * w                 # decay outside the preconditioner
        else:
            g = g + lam * w                         # l2 penalty inside the gradient
        m = beta1 * m + (1 - beta1) * g
        v = beta2 * v + (1 - beta2) * g * g
        w = w - eta * (m / (1 - beta1**t)) / (np.sqrt(v / (1 - beta2**t)) + eps)
    return w.mean(axis=1)

print('per-step decay rates: coupled', eta * lam / sigma.ravel(), ' AdamW', eta * lam)
print('mean w, coupled:', decay(False).round(3),
      '  predictions exp(-eta lam T / sigma):', np.exp(-eta * lam * T /
      ↪  sigma.ravel()).round(3))
print('mean w, AdamW:  ', decay(True).round(3),
      '  prediction (1 - eta lam)^T:', round((1 - eta * lam)**T, 3))
```

```
per-step decay rates: coupled [1.e-05 1.e-03]  AdamW 0.0001
mean w, coupled: [0.961 0.027]   predictions exp(-eta lam T / sigma): [0.961 0.018]
mean w, AdamW:   [0.67 0.67]   prediction (1 - eta lam)^T: 0.67
```

単一の λ の値から、2 つのグループには 1 ステップあたり 10^{-5} と 10^{-3} という 100 倍も異なる結合型減衰率が与えられる。ノイズの大きい重みは予測どおり初期の大きさの 0.96 を保ってい

る一方、静かな重みは約 0.03 まで低下しており、予測値の 0.018 をわずかに上回る。このわずかな差の原因は明快である。$w = 1$ 付近では、静かな重みに対するペナルティ勾配 $\lambda w = 0.1$ がノイズと同程度の大きさをもち、その重み自身の $\hat{v}$ を押し上げて初期の減衰を遅らせるためである。これとは対照的に、AdamW のもとでは両グループともぴったり $0.67 = (1 - \eta\lambda)^{4000}$ に到達する。正規化されたノイズステップの平均はゼロであり、乗法的な減衰が毎ステップ同一であるため、AdamW ではこの予測が期待値において厳密に成り立つ。

35.5.3.2 不動点

純粋なノイズの実験では勾配が決して落ち着かないため、停留点（stationary point）は存在しない。しかし厳密な勾配のもとではオプティマイザは静止でき、その不動点において λ は驚くほど異なる意味を帯びる。

命題（結合型減衰と分離型減衰の不動点）。 勾配 $\mathbf{g}(\mathbf{w}) = \nabla \ell(\mathbf{w})$（$\ell$ は損失）が厳密（ノイズなし）であり、$\epsilon > 0$ とする。モーメント推定値が定常状態に落ち着いた後に更新量がゼロになる点 $\mathbf{w}$ を、その更新の不動点と呼ぶ。(a) $\ell + \frac{\lambda}{2}\|\mathbf{w}\|^2$ 上の Adam が $\mathbf{w}$ に不動点をもつのは $\nabla \ell(\mathbf{w}) + \lambda \mathbf{w} = \mathbf{0}$ のとき、かつそのときに限り、これはペナルティ付き損失に対する SGD と同じ停留点である。(b) AdamW が $\mathbf{w}$ に不動点をもつのは、すべての座標 i について次式が成り立つとき、かつそのときに限る。

$$\lambda\, w_i (|g_i| + \epsilon) = -g_i, \qquad g_i = \partial_i \ell(\mathbf{w}). \tag{35.58}$$

したがって AdamW のすべての不動点はボックス $|w_i| < 1/\lambda$ の厳密な内部に存在し、$\epsilon \to 0$ の極限ではこの条件は二分法に帰着する。すなわち、$g_i = 0$ であるか、あるいは $|w_i| = 1/\lambda$ かつ w_i が g_i と逆符号になるかのいずれかである。

証明。 不動点ではモーメントの移動平均に入力されるベクトル $\mathbf{u}$ が定数となるため、$\hat{\mathbf{m}}_t \to \mathbf{u}$ かつ $\hat{\mathbf{v}}_t \to \mathbf{u}^2$ となる。(a) 結合型 Adam では $\mathbf{u} = \mathbf{g} + \lambda \mathbf{w}$ であり、座標 i の更新量は $-\eta\, u_i/(|u_i| + \epsilon)$ に収束するため、これがゼロになるのは $u_i = 0$ のとき、かつそのときに限る。(b) AdamW では $\mathbf{u} = \mathbf{g}$ であるため、座標 i の更新量は $-\eta\lambda w_i - \eta\, g_i/(|g_i| + \epsilon)$ に収束する。これをゼロと置いて両辺に $(|g_i| + \epsilon)/\eta$ を掛けると Equation 35.58 が得られる。絶対値をとれば $|w_i| = |g_i|/(\lambda(|g_i| + \epsilon)) < 1/\lambda$ となる。さらに $\epsilon \to 0$ とすると条件は $\lambda w_i |g_i| = -g_i$ となり、これは $g_i = 0$ または $\lambda w_i = -\mathrm{sign}(g_i)$ のときにちょうど成り立つ。■

この命題は不動点の性質を特徴づけるものであり、不動点の存在やそこへの反復点の収束を主張するものではない。また、この二分法は単なる極限における珍現象でもない。$1 - \lambda|w_i| = \epsilon/(|g_i| + \epsilon)$ が成り立つため、$\epsilon = 10^{-8}$ においては、すべての座標がボックス境界から割合 $\epsilon/|g_i|$ 以内に張り付いているか、さもなくば ϵ のオーダーの勾配をもつかの一方になる。

1 次元の例を見ると、この対比が具体的になる。制約なしの最小点が $\ell(w) = \frac{1}{2}(w - 3)^2$ に対して $w = 3$ である関数を考えよう。$\lambda = \frac{1}{4}$ のとき、結合型 Adam はペナルティ付き最小点 $3/(1 + \lambda) = 2.4$ で静止するのに対し、AdamW は 3.0 で静止する。ボックス $|w| \le 4$ の制約が有効にならないため、分離型重み減衰は最小点をそのまま残すのである。一方 $\lambda = 1$ のときは、結合型 Adam が 1.5 で静止するのに対し、AdamW はちょうどボックスの境界上にあたる 1.0 に張り付く。

このように、λ の 2 つの解釈はまったく異なる。ノイズが支配的な訓練領域では、AdamW が全座標に毎ステップ一様な縮小を適用する一方、結合型 Adam は勾配スケールによってそれを歪めてしまう。ところが厳密な勾配での不動点においては役割が逆転する。結合型 Adam は重みにガウス事前分布（Gaussian prior）を置いた最大事後確率推定（maximum a posteriori、MAP）の解を保つのに対し（36.6 節）、AdamW は λ を半径 $1/\lambda$ の ℓ_∞ 制約ボックスへと変えるのである

(Xie and Li 2024)。演習問題 6 では、このボックスを制約付き最適化（constrained optimization）と直接結びつける。

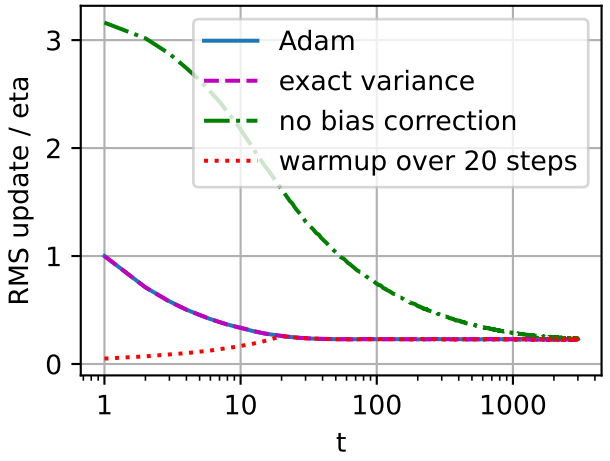
35.5.4 ウォームアップ

ステップ $t = 1$ では、バイアス補正後の Adam のモーメントは $\hat{\mathbf{m}}_1 = \mathbf{g}_1$ および $\hat{\mathbf{v}}_1 = \mathbf{g}_1^2$ となるため、更新量は単に $\eta \operatorname{sign}(\mathbf{g}_1)$、すなわちすべての座標でフルサイズ η のステップとなる。たった 1 つのミニバッチから得られた証拠だけで、勾配が 10^{-6} の座標も勾配が 10^{3} の座標とまったく同じ距離だけ動くのである。**ウォームアップ**は、訓練の序盤ステップにわたってステップサイズをゼロ付近から徐々に引き上げることで (Goyal et al. 2017)、こうした初期の更新を小さく抑える。この初期の不安定性は前処理行列 $\hat{\mathbf{v}}_t$ が $t = 1$ において 1 つのサンプルにしか基づいていないせいだと考えたくなるかもしれない。しかし単純な実験を行ってみると、別の要因が働いていることがわかる。

大きさ η のステップは本当に大きすぎるのだろうか。向きが一貫した勾配をもつ座標にとっては定常状態のステップも η であるため、大きすぎるわけではない。しかし勾配が純粋なノイズである座標にとっては、それは無駄な動きとなる。この無駄な動きがどれほど大きいかを測るため、各ステップで 2×10^4 個の独立な勾配座標を $\mathcal{N}(0, 1)$ からサンプリングし、デフォルトの Adam を実行して、二乗平均平方根の更新量を η 単位で追跡する。ここでは 4 本の曲線を比較する。標準の Adam、$\hat{\mathbf{v}}_t$ を厳密な分散 $\sigma^2 = 1$ に置き換えた Adam（前処理行列の推定誤差を完全に除いたもの）、両方のバイアス補正を取り除いた Adam、そして最初の 20 ステップにわたる線形ウォームアップを掛けた Adam である。

```
n, T, beta1, beta2, eps, T_w = 20_000, 3000, 0.9, 0.999, 1e-8, 20
m, v = np.zeros(n), np.zeros(n)
rms = {'Adam': [], 'exact variance': [], 'no bias correction': [],
       f'warmup over {T_w} steps': []}
for t in range(1, T + 1):
    g = rng.standard_normal(n)                      # pure noise, sigma = 1
    m = beta1 * m + (1 - beta1) * g
    v = beta2 * v + (1 - beta2) * g * g
    m_hat, v_hat = m / (1 - beta1**t), v / (1 - beta2**t)
    step = m_hat / (np.sqrt(v_hat) + eps)          # Adam's update / eta
    for r, u in zip(rms.values(), (step, m_hat, m / (np.sqrt(v) + eps),
                                   min(1.0, t / T_w) * step)):
        r.append(np.sqrt(np.mean(u**2)))
d2l.plot(np.arange(1, T + 1), list(rms.values()), 't', 'RMS update / eta',
         legend=list(rms), xscale='log')
ts = (1, 2, 3, 10, 20, 30, 100, 1000, 3000)
print(f'{"t:":24s}' + ''.join(f'{t:>6d}' for t in ts))
for name, r in rms.items():
    print(f'{name + ":":24s}' + ''.join(f'{r[t - 1]:6.2f}' for t in ts))
```

```
t:                          1     2     3    10    20    30   100  1000  3000
Adam:                    1.00  0.71  0.58  0.33  0.26  0.24  0.23  0.23  0.23
exact variance:          1.00  0.71  0.58  0.33  0.26  0.24  0.23  0.23  0.23
no bias correction:      3.16  3.02  2.87  2.17  1.61  1.32  0.74  0.29  0.24
warmup over 20 steps:    0.05  0.07  0.09  0.17  0.26  0.24  0.23  0.23  0.23
```



標準の Adam は $1.00\,\eta$ の RMS 更新量から始まり、$0.33\,\eta$（$t = 10$）まで下がり、約 50 ステップ以内に $\sqrt{(1-\beta_1)/(1+\beta_1)}\,\eta = 0.23\,\eta$ へと落ち着く。これは独立な単位ノイズの指数移動平均がもつ定常状態の大きさである。したがって純粋なノイズ座標においては、Adam の序盤ステップは定常時の約 4 倍の大きさになり、この過渡状態は約 10 ステップ続く。

原因はノイズを含んだ前処理行列にあるのだろうか。2 本目の曲線はそれを否定する。$\hat{\mathbf{v}}_t$ を厳密な分散に置き換えても、RMS 更新量はすべてのステップで 0.01 以内（2×10^4 個の座標によるサンプリング誤差の範囲）で変わらない。4 倍の膨張は**モメンタム**の移動平均に由来するものであり、独立なノイズを平均化して打ち消すまでに約 $1/(1-\beta_1) = 10$ ステップを要するのである。$t = 1$ において前処理行列が実際に行っているのは異なるスケールの座標を均一化することであり、すべての座標が $\sigma = 1$ を共有している設定ではその作用が隠れているにすぎない。

はるかに影響が大きいのはバイアス補正である。これがないと、最初の更新量は $3.16\,\eta$ まで跳ね上がり（$(1-\beta_1)/\sqrt{1-\beta_2}$ という Figure 35.18 の倍率と一致する）、過渡状態も 10 ステップから数百ステップへと引き伸ばされる。補正なしの更新量は $0.75\,\eta$ 前後（$t = 100$）、さらに $0.29\,\eta$（$t = 1000$）も残るが、これは補正なしの $\mathbf{v}_t$ が $1/(1-\beta_2) = 1000$ という遅い時間スケールでしか立ち上がらないためである。バイアス補正のないオプティマイザでは、自身のゼロ初期化の影響を打ち消すためだけに 1,000 ステップものウォームアップが必要になってしまう。バイアス補正を備えていれば、わずか 20 ステップの線形な立ち上げで残りの膨張を抑え込め、ウォームアップ曲線は終始 $0.26\,\eta$ 未満に保たれる。

これで数十ステップのウォームアップは説明できるが、大規模訓練でよく使われる数千ステップものウォームアップは説明できない。より長いウォームアップが対処しているのは、このノイズモデルには含まれない第 2 の仕組み、すなわち曲率である。ステップサイズが安定となるのは、最も急な曲率方向が課す上限 $2/\lambda_{\max}$ 未満の領域に限られる（35.1.2 節）。ランダム初期化（random initialization）の直後には、この曲率が異常に大きく、かつ急激に変化し得るため、訓練の後半では安全なステップサイズであっても初期には不安定になることがある。9.7.3 節では実際のニューラルネットワーク上でこれら両方の仕組みを再訪し、序盤のステップサイズを直接計測する。

35.5.5 構造化前処理行列

35.5.5.1 クロネッカー因子

回転した谷は、あらゆる対角前処理行列を打ち負かした。本節で扱ってきた手法はすべて、正定値行列（positive definite matrix）を用いた前処理付きの形式 $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\,\mathbf{P}_t^{-1}\mathbf{g}_t$（$\mathbf{P}_t$ が前処

理行列）を共有している。勾配降下法は $\mathbf{P} = \mathbf{I}$ を使い、ニュートン法はヘッセ行列 $\nabla^2 f$ を使い、Adam は過去の勾配から推定した対角行列を使う。フルサイズの曲率行列であれば、座標軸に沿っているか否かを問わずあらゆる方向の急峻さを捉えられるが、巨大すぎて保持できない。たった 1 枚の 4096×4096 重み行列でさえ 1.7×10^7 個のパラメータをもち、そのフル曲率行列は 2.8×10^{14} 個もの要素を含んで、倍精度（double precision）では約 2 ペタバイトものメモリを必要とする。

この両極端の中間を可能にしているのが、ニューラルネットワークのパラメータが行列の形に編成されているという事実である。線形層 $\mathbf{y} = \mathbf{W}\mathbf{a}$ において、重み行列を $\mathbf{W} \in \mathbb{R}^{m \times n}$、入力を $\mathbf{a} \in \mathbb{R}^n$、出力に関する勾配を $\delta = \partial \ell / \partial \mathbf{y} \in \mathbb{R}^m$ としよう。$\mathbf{W}$ に関する勾配はランク 1 の外積 $\delta \mathbf{a}^\top$ となる。行列の列を縦に並べて得られるベクトルを $\mathrm{vec}(\mathbf{V})$（$\mathbf{V}$ は任意の行列）と書くと、$\mathrm{vec}(\delta \mathbf{a}^\top) = \mathbf{a} \otimes \delta$ が成り立つ。ここで**クロネッカー積**（Kronecker product）$\mathbf{B} \otimes \mathbf{D}$ とは、第 (i, j) ブロックが $b_{ij}\mathbf{D}$ であるブロック行列を表す。

したがって、この層の外積による曲率、すなわちフィッシャー情報行列（Fisher information matrix）$\mathbb{E}[\nabla \log p \, \nabla \log p^\top]$（36.6 節で学ぶ）のうち $\mathbf{W}$ に対応するブロックは $\mathbb{E}\big[(\mathbf{a}\mathbf{a}^\top) \otimes (\delta\delta^\top)\big]$ となる。**K-FAC** (Martens and Grosse 2015) は、この期待値をクロネッカー積 $\mathbf{S}_{\text{in}} \otimes \mathbf{S}_{\text{out}}$ で近似する。ここで $\mathbf{S}_{\text{in}} = \mathbb{E}[\mathbf{a}\mathbf{a}^\top]$ は $n \times n$ 行列であり、$\mathbf{S}_{\text{out}} = \mathbb{E}[\delta\delta^\top]$ は $m \times m$ 行列である。この分解は入力と出力勾配が統計的に独立であれば厳密に成り立つ。訓練中のネットワークでそれが成り立つとは誰も考えていないが、安価に逆行列を計算できるためにこの近似が採用されている。

具体的には、クロネッカー積の逆行列はベクトル化する前の勾配行列に直接作用する。対称かつ正則な因子に対しては、次式が成り立つ。

$$\big(\mathbf{S}_{\text{in}} \otimes \mathbf{S}_{\text{out}}\big)^{-1} \mathrm{vec}(\mathbf{V}) = \mathrm{vec}\big(\mathbf{S}_{\text{out}}^{-1} \, \mathbf{V} \, \mathbf{S}_{\text{in}}^{-1}\big). \tag{35.59}$$

この恒等式を確かめるには、積の第 j 列である $\mathbf{DVB}$ の列を $\sum_k b_{kj} \mathbf{D}\mathbf{v}_k$（$\mathbf{v}_k$ は第 k 列、$\mathbf{V}$ の列ベクトル）と書けばよい。これらの列を縦に並べるとブロック積 $(\mathbf{B}^\top \otimes \mathbf{D})\,\mathrm{vec}(\mathbf{V})$ が得られるため、$\mathrm{vec}(\mathbf{DVB}) = (\mathbf{B}^\top \otimes \mathbf{D})\,\mathrm{vec}(\mathbf{V})$ となる。$(\mathbf{B} \otimes \mathbf{D})^{-1} = \mathbf{B}^{-1} \otimes \mathbf{D}^{-1}$ と各因子の対称性を用い、$\mathbf{B} = \mathbf{S}_{\text{in}}^{-1}$ および $\mathbf{D} = \mathbf{S}_{\text{out}}^{-1}$ と置けば Equation 35.59 が導かれる。このようにして、$(mn)^2$ 個の要素をもつ曲率行列の逆行列計算と適用が、サイズ n と m の手頃な 2 つの逆行列を通じて実行できる。以下のセルでは、先ほどの 4096×4096 の層についてメモリ量と演算回数を比較する。

```
m = n = 4096
print(f'stored entries: full {(m * n)**2:.1e}, two factors {m**2 + n**2:.1e}')
print(f'flops to invert: full {(m * n)**3:.1e}, two factors {m**3 + n**3:.1e}')
```

```
stored entries: full 2.8e+14, two factors 3.4e+07
flops to invert: full 4.7e+21, two factors 1.4e+11
```

因子分解により、保持する要素数は 2.8×10^{14} 個から 3.4×10^7 個へ、逆行列計算のコストは 4.7×10^{21} 回の演算から約 1.4×10^{11} 回へと削減される。

35.5.5.2 K-FAC、Shampoo、Muon

クロネッカー積の恒等式には、どの行列を因子分解するか、そしてそれにどの指数を適用するかという 2 つの選択肢が残されている。現代の構造化オプティマイザは、層の勾配行列に対する両側からの前処理（preconditioning）形式 $\mathbf{G} \mapsto \mathbf{P}_{\text{l}}^{-a} \mathbf{G} \, \mathbf{P}_{\text{r}}^{-a}$ を共有している（ここでの勾配行列 $\mathbf{G}$ は太字であり、リグレット解析におけるスカラーの勾配上界 G とは区別する）。これらの手法を並べると、Adam の対角スケーリングから行列レベルの正規化へと至る自然な発展が見えてくる。

K-FAC は層の入力と出力勾配の二次モーメントから因子を構成し、指数を $a = 1$ とする。これはフィッシャー情報行列による前処理、すなわち**自然勾配**（natural gradient）(Amari 1998) を近似しており、フィッシャー行列を計量として用いた AdaGrad の計量的解釈にあたる。ソフトマックス交差エントロピー（cross-entropy）および二乗損失（squared loss）に対しては、（ラベルをモデルからサンプリングした）フィッシャー行列はガウス・ニュートン行列と一致する。ガウス・ニュートン行列とは、ネットワークの二階導関数を含む項を落としたヘッセ行列の半正定値（positive semidefinite）近似である。したがってこれらの損失に対して、K-FAC は真の曲率を近似している。

これに対し **Shampoo** (Gupta et al. 2018) は、AdaGrad の発想に沿って累積した勾配行列から左右の因子を直接構成し、$\mathbf{P}_{\mathrm{l}} = \sum_t \mathbf{G}_t \mathbf{G}_t^{\top}$ および $\mathbf{P}_{\mathrm{r}} = \sum_t \mathbf{G}_t^{\top} \mathbf{G}_t$ と置いて指数を $a = \frac{1}{4}$ とする。**Muon** (Jordan et al. 2024) は記憶をもたない Shampoo とみなすことができ、現在の行列 $\mathbf{G} = \mathbf{U}\Sigma\mathbf{V}^{\top}$（正方かつ正則と仮定する）のみを用いて次のように計算する。

$$\left(\mathbf{G}\mathbf{G}^{\top}\right)^{-1/4} \mathbf{G} \left(\mathbf{G}^{\top}\mathbf{G}\right)^{-1/4} = \mathbf{U}\Sigma^{-1/2}\,\Sigma\,\Sigma^{-1/2}\mathbf{V}^{\top} = \mathbf{U}\mathbf{V}^{\top},$$

これは $\mathbf{G}$ の極因子（polar factor）である。この変換は $\mathbf{G}$ の特異ベクトルを保ったまま、すべての特異値（singular value）を 1 に揃える。Adam が更新の各座標を均一化するのに対し、Muon は特異方向を均一化する。しかも座標ごとの正規化とは異なり、この写像は行および列の直交回転と可換である（演習問題 5d）。この直交化をモメンタム行列に適用する実用オプティマイザとしての Muon は本編（9.6 節）で扱い、行列積（matrix multiplication）のみを用いて極因子を計算する Newton–Schulz 反復については 33.5.4.2 節で解析する。

これら 3 つの手法はいずれも、フルサイズのニュートン法を再現するものではない。Muon は二次モーメント行列を一切保持せず 1 ステップあたり数回の行列積しか必要としないのに対し、K-FAC と Shampoo は 2 つの因子の逆行列や行列べき根を計算する必要があり、通常はそのコストを数ステップにわたって償却する。さらに、本節で示した肯定的な結果はすべて、曲率が座標軸に沿っているか、個々の層のクロネッカー構造に沿っていることを仮定している。深層ニューラルネットワークの真の曲率は対角でも固定でもないため、これらの適応的手法が実際の訓練でこれほど高い性能を発揮する理由を説明することは、今なお活発に研究されている未解決問題である。

35.5.6 まとめ

AdaGrad、RMSProp、Adam に共通する基本原理である「各座標の更新量を過去の勾配の二乗平均平方根で割る」という操作は、曲率が座標軸に沿っている限り条件数を消去する。$\epsilon = 0$ のとき、Adam の反復点は座標ごとの再スケーリングに対して厳密に不変であり、軸に沿った谷の上ではスケーリング単独で最適調整されたヘビーボール法に 1 桁の差をつけて勝つことができる。しかし谷を 45° 回転させるとその優位性は消え失せ、スケーリング単独では決して収束せず、モメンタム付きの Adam でもヘビーボール法の約 2 倍のステップ数を要する。

過去の勾配を忘却することは、さらに微妙な問題をもたらす。凸な損失の周期的な系列上では、Adam の短い二次モーメント記憶によって定義域の最悪点へと駆動されることがあり、この破綻は AMSGrad の累積最大値によって修正される。重み減衰を伴う場合、ℓ_2 ペナルティを Adam の内部に通すと MAP の不動点は保たれるものの、各座標の縮小量がその勾配の大きさに反比例してスケーリングされてしまう。これに対し AdamW は重み減衰を分離してすべての座標が毎ステップ一様に縮小するようにし、厳密な勾配での不動点においては λ を半径 $1/\lambda$ の ℓ_∞ ボックスへと変える。

最後に、純粋なノイズ座標における Adam の序盤の更新量は、前処理行列ではなくモメンタムの移動平均が原因となって定常時の約 4 倍の大きさになり、バイアス補正を省くと数百ステップに

わたってさらに大きく膨張する。一方、実務で使われるより長いウォームアップは初期の曲率に対処している。対角スケーリングの先には、K-FAC、Shampoo、Muon のようなクロネッカー分解型の前処理行列があり、2 つのコンパクトな因子を用いて重み行列内の座標間の構造を捉え、なかでも Muon は層の更新における特異方向を均一化する。

35.5.7 演習問題

1. **計量の観点からの AdaGrad**。演習問題 1b（35.1 節）では、ノルム $\|\mathbf{d}\|_{\mathbf{P}} = \sqrt{\mathbf{d}^\top \mathbf{P}\mathbf{d}}$ のもとでの最急降下方向が $-\mathbf{P}^{-1}\nabla f$ に比例することを示した。
   1. $\mathbf{P}_t = \operatorname{diag}\big(\sqrt{\mathbf{s}_t}\big) + \epsilon \mathbf{I}$（ただし $\mathbf{s}_t = \sum_{\tau \le t} \mathbf{g}_\tau^2$）と置いて Equation 35.51 を導出せよ。
   2. 目的関数 f を cf（定数 $c > 0$）に置き換え、$\epsilon = 0$ とする。AdaGrad の更新はどう変化するか。もし $\mathbf{P}_t$ を $\mathbf{s}_t$ から構成し、$\sqrt{\mathbf{s}_t}$ を使わなかった場合はどう変化するか。両者を勾配降下法およびニュートン法と比較せよ。
   3. ある 1 つの座標の勾配が独立同分布に従い $\mathbb{E}[g_{t,i}^2] = \sigma^2$ を満たすとする。上記 2 つの選択のそれぞれのもとで、その座標の実効ステップは t に関してどの速さで減衰し、σ にどう依存するか。
2. **集中した勾配と AdaGrad の上界**。定数倍を除けば、調整された大域ステップをもつ射影付きオンライン勾配降下法は、リグレット $D_2 G\sqrt{T}$ をユークリッド直径 D_2 の凸集合上で達成する（$\|\mathbf{g}_t\| \le G$ のとき。演習問題 4）。一方、対角 AdaGrad のリグレットは $D_\infty \sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} g_{t,i}^2}$ であり、ここで D_∞ は ℓ_∞ ノルムにおける集合の直径である (Duchi et al. 2011)。ボックス $[-1, 1]^d$ とワンホット勾配を考え、各ステップで独立に選ばれた 1 つの座標 i（選ばれる確率を p_i とする）が $+1$ または -1 の勾配を受け取り、他のすべての座標は 0 を受け取るものとする。
   1. 両方の上界を計算せよ。AdaGrad については、各平方根の期待値をイェンセンの不等式で上から評価し、勾配降下法の上界に対する AdaGrad の上界の比が高々 $\sum_i \sqrt{p_i}/\sqrt{d}$ であることを示せ。
   2. 一様な p_i の場合、および Zipf 頻度 $p_i \propto i^{-\alpha}$（$d = 10^4$、$\alpha \in \{1, 2, 3\}$）の場合についてこの比を求めよ。
   3. 自然言語テキストにおける単語頻度は $\alpha \approx 1$ の Zipf 則に従い、そのときの比は約 0.63 となる。Bag-of-Words モデルは、係数 $\sqrt{d}$（35.5.1 節）の利得のうちどれだけを実際に得られるか。また、その利得をすべて得るのはどのような勾配分布のときか。
3. **手計算による Reddi の反例**。$C = 4$、$\beta_1 = 0$、命題の設定である $\beta_2 = 1/(1+C^2) = 1/17$ とし、`online` と同様にバイアス補正なしとする。
   1. 固定ステップ η、$[-1, 1]$ の内部にある反復点、および周期パターンに落ち着いた v_t を仮定する。v_a を v が $+C$ のステップ直後にとる値とし、3 ステップの漸化式を解いて v_a を閉じた形で求め、本文中の周期的な値 15.12、1.83、1.05 と 3 回の変位を導出せよ。また SGD の 3 回のステップ $\eta g_t/C$ を計算せよ。Adam は SGD と比べてどのステップを拡大しているか、またそれはなぜか。
   2. AMSGrad は v_t を $\tilde{v}_t = \max(\tilde{v}_{t-1}, v_t)$ に置き換える。実効ステップ $\eta_t/\sqrt{\tilde{v}_t}$ が、t に関して η_t が非増加である限り非増加となることを示せ。周期状態における 3 回の変位を計算し、正味の変位が $-0.51\,\eta$ となることを示せ。
   3. 命題では $\eta < \sqrt{1-\beta_2}$ という条件がステップ $\eta_t = \eta/\sqrt{t}$ に対して必要であった。これがどこで使われるかを示せ。すなわち、$x = 1$ から周期を開始し、$v \ge (1-\beta_2)C^2$ が $+C$ の勾配の直後に成り立つことを用いて、周期の最初のステップが $(0, 1)$ 内に着地し、射影が作用しないことを示せ。
4. **オンライン勾配降下法のリグレット**。●射影付きオンライン勾配降下法は $\mathbf{x}_{t+1} = \Pi_{\mathcal{C}}(\mathbf{x}_t - \eta_t \mathbf{g}_t)$（ただし $\mathbf{g}_t = \nabla f_t(\mathbf{x}_t)$）を実行する。ここで $\Pi_{\mathcal{C}}$ は閉じた凸集合 $\mathcal{C}$（直径 D）へのユークリッド射影であり、$\|\mathbf{g}_t\| \le G$、かつステップサイズ η_t は非増加であるとする

(Zinkevich 2003)。

1. リグレットが次式を満たすことを証明せよ。

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{C}} \sum_{t=1}^{T} f_t(\mathbf{x}) \;\le\; \frac{D^2}{2\eta_T} + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t. \tag{35.60}$$

   $f_t(\mathbf{x}_t) - f_t(\mathbf{x}) \le \mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x})$ から出発し、射影が $\mathcal{C}$ の各点への距離を増加させないことを用いよ。
2. $\eta_t = D/(G\sqrt{t})$ とするとリグレットが高々 $\frac{3}{2}DG\sqrt{T}$ となり、一方で $\eta_t = c/t$ とすると上界が T に関して（任意の $c > 0$ に対して）線形に増大することを示せ。
3. これら 2 つの上界を演習問題 1c の減衰率と組み合わせ、AdaGrad が $\sqrt{\mathbf{s}_t}$ で割り、$\mathbf{s}_t$ では割らない理由を説明せよ。

5. [code] **回転感度**。●`rotated`、`run`、および `#adaptive-stochastic-methods-adam-vs-gd` セルの更新関数ファクトリを用いよ。
   1. $\mathbf{A}_\theta$ 上で $\mathbf{R}_\theta(1,1)^\top$ から出発した勾配降下法およびヘビーボール法の反復点が、軸に沿った谷におけるそれぞれの反復点に $\mathbf{R}_\theta$ を掛けたものになり、したがってステップ数が θ に依存しないことを示せ。
   2. $\beta_1 = 0$、$\theta = \pi/4$ において、f の 40,000 ステップ後の値を表のグリッドの各 η について出力し、本文の値 $250\,\eta^2$ と比較せよ。勾配降下法に対する Adam の優位性のうち、回転後も残る部分をもたらしているのは Adam のどの要素か。
   3. $\theta = \pi/16$ における Adam の座標ごとの実効ステップを出力せよ。これほど小さな回転であっても、軸に沿った谷で Adam が築いたステップサイズ比の大半が失われるのはなぜか。
   4. 演習問題 4（35.1 節）では、$-\operatorname{sign}(\mathbf{g})$ が ℓ_∞ ノルムのもとでの最急降下方向であり、$-\mathbf{U}\mathbf{V}^\top$ がスペクトルノルムのもとでの最急降下方向（ただし $\mathbf{G} = \mathbf{U}\Sigma\mathbf{V}^\top$）であることを確認した。$\beta_1 = \beta_2 = 0$ かつ $\epsilon = 0$ の Adam が $-\operatorname{sign}(\mathbf{g}_t)$ の方向にステップを進めることを示せ。続いて $\mathbf{Q}_1$ と $\mathbf{Q}_2$ を直交行列とし、写像 $\mathbf{G} \mapsto \mathbf{U}\mathbf{V}^\top$ が $\mathbf{G} \mapsto \mathbf{Q}_1\mathbf{G}\mathbf{Q}_2$ と可換であることを示すとともに、符号写像が可換にならない 2×2 の例を 1 つ挙げよ。この結果は、Adam と比較した Muon の回転感度について何を示唆しているか。

6. [extended] **AdamW の暗黙の ℓ_∞ 制約**。●●35.5.3 節の命題から、AdamW の不動点はボックス $|w_i| < 1/\lambda$ 内にあり、$\epsilon \to 0$ では勾配がゼロでないすべての座標でその境界上に乗ることがわかった。Xie and Li (2024) は、$\beta_1 \le \beta_2$ かつ和が発散する非増加ステップスケジュールをもつフルバッチ AdamW が収束するならば、その極限は $\min_{\mathbf{w}} \ell(\mathbf{w})$（制約 $\|\mathbf{w}\|_\infty \le 1/\lambda$）の KKT 点であることを証明している。
   1. 実行可能な $\mathbf{w}$ がこの問題の KKT 点となる（一般的な条件の導出は 35.6 節を参照）のは、すべての座標が $\partial_i \ell(\mathbf{w}) = 0$（$|w_i| < 1/\lambda$ のとき）および $w_i\,\partial_i\ell(\mathbf{w}) \le 0$（$|w_i| = 1/\lambda$ のとき）を満たす場合である。このような点では、いかなる実行可能方向も一次近似において ℓ を減少させないことを示せ。
   2. 定理の設定である $\epsilon = 0$ を AdamW の条件 Equation 35.58 に代入せよ。$\partial_i\ell \neq 0$ である座標が KKT 条件を満たすことを示せ。勾配がゼロになる座標に対して、この条件だけでは $|w_i| \le 1/\lambda$ を示せないのはなぜか。
   3. $n = 100$、$d = 5$、標準正規分布に従う特徴量、およびラベル $y_i = \operatorname{sign}(\mathbf{x}_i^\top \mathbf{w}_{\text{true}})$ をもつ線形分離可能なロジスティック回帰のデータを生成せよ（正則化なしの損失は有限の最小点をもたない）。フルバッチ AdamW を $\epsilon = 10^{-16}$ で 5×10^4 ステップにわたり、η をコサインスケジュールで 10^{-2} から 0 まで減衰させて $\lambda \in \{0.05, 0.1, 0.2\}$ のそれぞれで実行せよ。得られた結果を、射影勾配降下法（ここでは各勾配ステップの後にすべての座標を $[-1/\lambda, 1/\lambda]$ にクリップする）で求めたボックス制約付き問題の解、および $\ell + \frac{\lambda}{2}\|\mathbf{w}\|^2$ 上での Adam の解と比較せよ。
   4. サイズ 10 および 50 のミニバッチ勾配を用いて AdamW の実行を繰り返せ。反復点はボックスに対してどこに落ち着くか、またそれはなぜか。

5. これらの結果から、AdamW において λ は何を制御しているといえるか。またそれは、ガウス事前分布に基づく λ の解釈（36.6 節）とどう異なるか。

7. **クロネッカー恒等式とその計算コスト**。列方向の積み重ね（列スタック）として vec を用いよ。
   1. $\operatorname{vec}(\mathbf{DVB}) = (\mathbf{B}^{\top} \otimes \mathbf{D})\operatorname{vec}(\mathbf{V})$ を積の第 j 列を書き下すことで証明し、さらに $(\mathbf{B} \otimes \mathbf{D})(\mathbf{B}' \otimes \mathbf{D}') = (\mathbf{BB}') \otimes (\mathbf{DD}')$ を証明せよ。
   2. 対称かつ正則な因子に対して Equation 35.59 を導け。因子が対称でない場合にはどの式に置き換わるか、また行方向に並べる（行スタックの）場合にはどの式になるか。
   3. $m \times n$ の層について、フルサイズの $mn \times mn$ 行列の逆行列を求めて適用するのに必要な演算回数と、2 つの因子の逆行列を求めて Equation 35.59 により適用する演算回数（$O(m^3 + n^3 + mn(m+n))$）を比較せよ。両方の回数を $m = n = 4096$ の場合、および $m = 4096$、$n = 16384$ の場合について評価せよ。
   4. 単一の勾配 $\mathbf{G} = \mathbf{U}\Sigma\mathbf{V}^{\top}$ に対して $\mathbf{P}_{\mathrm{l}} = \mathbf{G}\mathbf{G}^{\top}$、$\mathbf{P}_{\mathrm{r}} = \mathbf{G}^{\top}\mathbf{G}$ と置く。Shampoo のステップ $\mathbf{P}_{\mathrm{l}}^{-1/4}\mathbf{G}\,\mathbf{P}_{\mathrm{r}}^{-1/4}$ が $\mathbf{UV}^{\top}$ に等しいことを示せ。$\mathbf{G}$ が正方行列でないとき、逆 4 乗根は何を意味する必要があるか、またその結果はどの特異ベクトルを残すか。

35.6 制約付き最適化

ここまでに最適化したパラメータはすべて任意の実数値をとることができたため、最小点を見つけるとは勾配がゼロになる点を探すことであった。しかし、深層学習に現れる多くの量にはそのような自由度がない。アテンション重みは非負で和が 1 にならなければならず、勾配クリッピングは更新のノルムを閾値以下に抑え、ノルムバジェットは重みベクトルの大きさを制限し、サポートベクターマシンはすべての訓練マージンが少なくとも 1 であることを要求する。変数が実行可能領域（feasible set）の内部に留まらなければならないとき、最適点において勾配がゼロになることはめったになく、前節までのゼロ勾配による判定は破綻する。制約のもとにある最適点をどのように見分け、どのように計算すればよいのだろうか。

勾配クリッピングは最も単純な具体例を与える。ベクトル $\mathbf{y} = (3, 4)$ を、そのノルムが高々 1 になるようにクリップしたいとしよう。単位長に再スケーリングすると $(0.6, 0.8)$ が返り、これは単位円板上で $\mathbf{y}$ に最も近い点である。言い換えれば、$(0.6, 0.8)$ は二乗距離 $f(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ を円板 $\|\mathbf{x}\| \le 1$ 上で最小化し、最適値 $f = 8$ を達成する。ところが、その点での勾配は $(-2.4, -3.2)$ であり、ゼロにはほど遠い。$(0.6, 0.8)$ を最適にしているのは、**許容される**いかなる方向も f を減少させないという事実である。すなわち、$\|\mathbf{x}\| \le 1$ を保つあらゆる移動は、1 次の範囲で f を増加させるか変化させないかのいずれかである。このような問題を扱うには、ゼロ勾配条件に代わる基準と、そのままでは実行可能領域の外へ飛び出してしまう素の勾配降下法に代わる手法の両方が必要になる。

「勾配がゼロである」に代わる条件では、制約ごとに**乗数**（multiplier）と呼ばれる 1 つのスカラーを導入し、最適点がその制約をどれほど強く押しているかを測る。点 $(0.6, 0.8)$ では、境界が外向き法線に沿って強さ 4 で押し返すため、$(-2.4, -3.2) + 4\,(0.6, 0.8) = \mathbf{0}$ が成り立つ。乗数を用いると一階最適性条件（first-order optimality conditions）が得られる。すなわち、等式制約に対するラグランジュ条件と、不等式制約に対するカルーシュ・キューン・タッカー（KKT）条件である。一方、$(3, 4)$ を $(0.6, 0.8)$ へと写した射影（projection）からは、直接的なアルゴリズムが示唆される。通常の勾配ステップを進めたあと、最も近い実行可能点へ射影して戻せばよい。さらに、硬い境界が折れ目を持つペナルティ（たとえば Lasso が各重みに課す絶対値ペナルティ）に置き換わるとき、射影は**近接ステップ**（proximal step）へと一般化され、小さな重みを厳密なゼロへと送り込む。本節全体を通して、凸性（35.3 節）こそがこれらの条件を大域的最適性の保証へと変え、アルゴリズムの収束を保証する鍵となる。

本節を支える柱は 3 つある。KKT 条件とその凸問題における十分性、射影を特徴づける鈍角不等

式（obtuse-angle inequality）、そしてソフト閾値処理（soft-thresholding）である。

35.6.1 等式制約とラグランジュ乗数

35.6.1.1 接条件とラグランジュ条件

等式制約は不等式制約よりも単純であるため、まずはクリッピングの例の中身の詰まった円板をその境界に置き換えるところから始めよう。すなわち、$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$（ただし $\mathbf{y} = (3,4)$）を、$h(\mathbf{x}) = \frac{1}{2}(\|\mathbf{x}\|^2 - 1) = 0$ と表される単位円上で最小化する。より一般に、微分可能な目的関数 $f : \mathbb{R}^n \to \mathbb{R}$ を 1 つの等式制約のもとで最小化する問題を考える。

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \quad \text{subject to} \quad h(\mathbf{x}) = 0.$$

実行可能領域は全次元の領域ではなく曲面（2 次元では曲線）であるため、実行可能な移動はすべてその曲面上に留まらなければならない。

Figure 35.20 に最適点の幾何学的な特徴を示す。いま、f の等位集合（level set）が円と**交差する**実行可能点に立っているとしよう。このとき ∇f は円に沿った非ゼロの成分を持つため、その成分と逆向きに円に沿って滑るように動けば、実行可能領域から出ることなく f を減らせる。したがって、そのような点は最適点ではあり得ない。最適点が現れうるのは等位集合が円に**接する**場所だけであり、そこでは円に沿ったどの移動も 1 次の範囲で f を変化させない。互いに接する曲線は平行な法線ベクトルを共有するため、最適点において ∇f は ∇h のスカラー倍でなければならない。

命題（ラグランジュ条件）$\mathbf{x}^\star$ を f の $\{\mathbf{x} : h(\mathbf{x}) = 0\}$ 上における極小点とする。f は $\mathbf{x}^\star$ で微分可能であり、h は $\mathbf{x}^\star$ の近傍で連続微分可能であり、かつ $\nabla h(\mathbf{x}^\star) \neq \mathbf{0}$ であるとする。このとき、一意なスカラー $\nu^\star$（ラグランジュ乗数）が存在して次式を満たす。

$$\nabla f(\mathbf{x}^\star) + \nu^\star \, \nabla h(\mathbf{x}^\star) = \mathbf{0}. \tag{35.61}$$

言葉で言えば、目的関数の勾配は制約の法線ベクトルの定数倍と釣り合っている。これは 34.2.4 節で証明したラグランジュ条件と同じものであり、そこでは制約を $g(\mathbf{x}) = c$、条件を $\nabla f = \mu \nabla g$ と書いていた。$h = g - c$ および $\nu = -\mu$ とおけば 2 つの形式は一致する。ここでプラスの符号を用いるのは、次に導入するラグランジアンの符号と揃えるためである。

接点 $(0.6, 0.8)$ を Figure 35.20 で確認すると、$\nabla f = \mathbf{x} - \mathbf{y} = (-2.4, -3.2)$ かつ $\nabla h = \mathbf{x} = (0.6, 0.8)$ である。この 2 つのベクトルは実際に平行で $\nabla f = -4\,\nabla h$ を満たしており、これから $\nu^\star = 4$ が得られる。

仮定 $\nabla h(\mathbf{x}^\star) \neq \mathbf{0}$ は**制約想定**（constraint qualification）の最初の例であり、単なる技術的な細目ではなく本質的な条件である。$f(\mathbf{x}) = x_1$ を制約 $h(\mathbf{x}) = x_1^2 + x_2^2 = 0$ のもとで最小化することを考えよう。実行可能領域は $\mathbf{0}$ のただ 1 点のみからなり、これが自動的に最小点となる。ところが $\nabla f(\mathbf{0}) = (1, 0)$ である一方 $\nabla h(\mathbf{0}) = \mathbf{0}$ であるため、いかなる乗数 $\nu^\star$ も Equation 35.61 を満たすことができない。実行可能領域が接線方向を持たない 1 点へと潰れてしまっているため、接条件の議論が足掛かりを失うのである。

35.6.1.2 ラグランジアン

ここで、$\mathbf{x}$ と ν という 2 つの未知数と、勾配が平行であることおよび実行可能性という 2 つの要請があることに注目しよう。これら両方の要請は、$(\mathbf{x}, \nu)$ の 1 つの関数の停留性へとまとめるこ

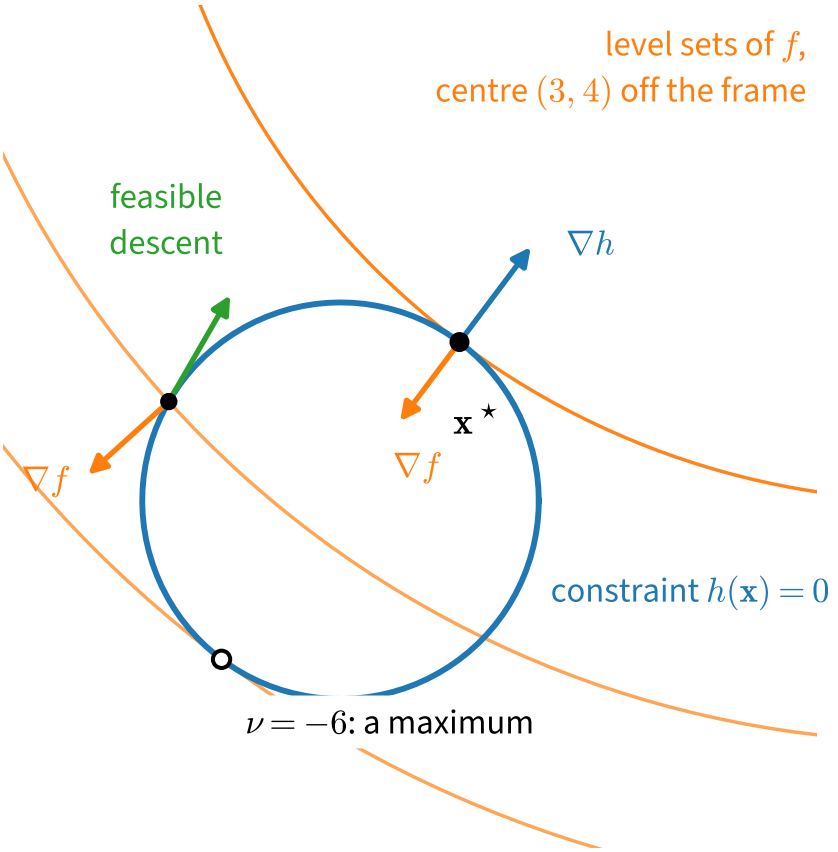


Figure 35.20: $f = \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$ ($\mathbf{y} = (3, 4)$) の単位円 $h(\mathbf{x}) = 0$ 上での最小化。等位集合が円と交差する場所では、∇f が制約に沿った成分を持ち、実行可能な降下方向を与える。$\mathbf{x}^\star = (0.6, 0.8)$ では等位集合が円に接し、$\nabla f = -4\,\nabla h$ となる。対蹠点 $(-0.6, -0.8)$ も $\nu = -6$ で接条件を満たし、最大点となる。

とができる。この関数は**ラグランジアン**と呼ばれ、乗数で重み付けした制約を目的関数に加えたものである。

$$\mathcal{L}(\mathbf{x}, \nu) = f(\mathbf{x}) + \nu\, h(\mathbf{x}). \tag{35.62}$$

$\mathcal{L}$ のすべての偏微分をゼロとおくと次式が得られる。

$$\nabla_{\mathbf{x}} \mathcal{L} = \nabla f + \nu \nabla h = \mathbf{0} \qquad \text{and} \qquad \frac{\partial \mathcal{L}}{\partial \nu} = h(\mathbf{x}) = 0,$$

これにより、Equation 35.61 と等式制約が同時に復元される。このようにして、$\mathbf{x}$ に関する制約付き最適化問題は、$(\mathbf{x}, \nu)$ に関する制約なしの停留問題へと帰着する。

ラグランジアンを使って単位円上の問題を解いてみよう。$\mathbf{x}$ に関する停留性から $\nabla_{\mathbf{x}}\mathcal{L} = (\mathbf{x} - \mathbf{y}) + \nu\mathbf{x} = \mathbf{0}$ となるため、$\mathbf{x} = \mathbf{y}/(1+\nu)$ は原点と $\mathbf{y}$ を通る直線上になければならない。さらに実行可能性 $\|\mathbf{x}\| = 1$ を課すと $|1+\nu| = \|\mathbf{y}\| = 5$ が必要となり、次の 2 つの解が得られる。

- $\nu = 4 : \mathbf{x} = (0.6, 0.8)$ であり、距離 4 だけ $\mathbf{y}$ から離れ、$f = 8$ となる。
- $\nu = -6 : \mathbf{x} = (-0.6, -0.8)$ であり、距離 6 だけ $\mathbf{y}$ から離れ、$f = 18$ となる。

2 番目の解は、円上で**最も遠い**点（$\mathbf{y}$ からの距離が最大となる点）である。ここから重要な注意点が浮かび上がる。ラグランジュ条件は必要条件であって十分条件ではない。というのも、曲面上の f の極大点や鞍点を含むすべての停留点で成り立つからである。この例では、乗数の符号が最小点と最大点を区別している。次節で不等式制約へ進むと、KKT 条件がこの符号を一般的な規則へと昇華させる。

複数の等式制約 $h_j(\mathbf{x}) = 0$（$j = 1, \dots, p$）がある場合は、各制約にそれぞれ乗数を割り当て、ラグランジアンは $\mathcal{L} = f + \sum_j \nu_j h_j$ となる。制約の勾配 $\nabla h_j(\mathbf{x}^\star)$ が線形独立である限り、同じ議論により $\nabla f(\mathbf{x}^\star)$ は制約の法線が**張る空間**（span）に属する。すなわち、目的関数の勾配は制約の勾配の線形結合と釣り合う。

35.6.1.3 超平面上の最近接点

線形制約を持つ最も単純な制約付き問題は、超平面上で原点に最も近い点を求める問題である。すなわち、$f(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{x}\|^2$ を制約 $h(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} - b = 0$ のもとで最小化する。$\mathcal{L} = \tfrac{1}{2}\|\mathbf{x}\|^2 + \nu(\mathbf{a}^\top\mathbf{x} - b)$ の勾配をゼロとおくと $\mathbf{x} + \nu\mathbf{a} = \mathbf{0}$ となるため、最近接点は法線ベクトル $\mathbf{a}$ の方向になければならない。$\mathbf{x} = -\nu\mathbf{a}$ を制約に代入すると $-\nu\|\mathbf{a}\|^2 = b$ となり、$\nu^\star = -b/\|\mathbf{a}\|^2$ と定まる。したがって、解と最適値は次のようになる。

$$\mathbf{x}^\star = \frac{b}{\|\mathbf{a}\|^2}\,\mathbf{a}, \qquad p^\star = \frac{b^2}{2\,\|\mathbf{a}\|^2}.$$

幾何学的には、最近接点は距離 $b/\|\mathbf{a}\|$ だけ単位法線 $\mathbf{a}/\|\mathbf{a}\|$ に沿って進んだ位置にあり、33.1 節の射影の公式と一致する。

$\mathbf{a} = (3, 4)$、$b = 5$ とすると、再び $(0.6, 0.8)$ に到達し、今度は $\nu^\star = -0.2$、$p^\star = \tfrac{1}{2}$ となる。直線 $3x_1 + 4x_2 = 5$ は点 $(0.6, 0.8)$ で単位円に接している。ここで、乗数が単なる代数計算の補助にとどまらないことに注目しよう。b を 5 から 5.1 へとわずかに動かして直線を原点から少し遠ざけると、最適値は約 $0.02 = -\nu^\star \cdot 0.1$ だけ増加する。35.7.3 節で一般の場合について示すように、乗数が常に測っているのはまさにこの量、すなわち制約をずらしたときの最適値の感度である。

35.6.2 不等式制約と KKT 条件

35.6.2.1 有効制約と非有効制約

機械学習に現れる制約の大半は片側のみを制限するものである。たとえば、ノルムが高々 r であること、確率が少なくとも 0 であること、あるいはリソースバジェットを超えないことなどである。不等式制約は、次の標準形で書く。

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \le 0, \; i = 1, \dots, m,$$

このように書くと、各 g_i は領域の真の内部で負、境界上でゼロとなる。解 $\mathbf{x}^\star$ において、すべての不等式は 2 つの状態のいずれかに分類される。$g_i(\mathbf{x}^\star) < 0$ ならば、その制約は**非有効**（inactive）である。解は領域の真の内部にあるため、局所的にはその制約は何の影響も及ぼさない。一方、$g_i(\mathbf{x}^\star) = 0$ ならば、その制約は**有効**（active）である。解はまさに境界上に位置し、そこで不等式は等式のように働く。

勾配クリッピングは、この両方の状態を明快に示してくれる。ベクトル $\mathbf{y}$ を半径 r のユークリッド球へ射影するため、$f(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ を制約 $g(\mathbf{x}) = \tfrac{1}{2}(\|\mathbf{x}\|^2 - r^2) \le 0$ のもとで最小化するとしよう。乗数 λ を導入すると、$f + \lambda g$ の停留性から $(\mathbf{x} - \mathbf{y}) + \lambda \mathbf{x} = \mathbf{0}$ が得られるため、$\mathbf{x} = \mathbf{y}/(1 + \lambda)$ は常に $\mathbf{y}$ を動径方向に再スケーリングした点となる。どちらの状態になるかは $\|\mathbf{y}\|$ によって決まる。

- $\|\mathbf{y}\| \le r$ のとき、$\mathbf{y}$ はすでに実行可能であり、そのまま f を最小化する。制約は非有効で、$\lambda^\star = 0$ かつ $\mathbf{x}^\star = \mathbf{y}$ となる。たとえば $\mathbf{y} = (0.3, 0.4)$ は $r = 1$ のときそのまま変化しない。
- $\|\mathbf{y}\| > r$ のとき、内点が最適になることはあり得ない。内点が最小点であるためには f の制約なし最小点でなければならないが、唯一の制約なし最小点は実行不可能な点 $\mathbf{y}$ だからである。したがって制約は必ず有効となり、$\|\mathbf{x}^\star\| = r$ から $\lambda^\star = \|\mathbf{y}\|/r - 1 > 0$ が定まる。$\mathbf{y} = (3, 4)$、$r = 1$ のときは、$\lambda^\star = 4$、$\mathbf{x}^\star = (0.6, 0.8)$、および $p^\star = \tfrac{1}{2} \cdot 4^2 = 8$ が得られる。

両方のケースは 1 つの式にまとまる。

$$\mathbf{x}^\star = \frac{\mathbf{y}}{\max\left(1, \; \|\mathbf{y}\|/r\right)},$$

この式は、短いベクトルはそのまま残し、長いベクトルはノルム r に再スケーリングする。これを勾配ベクトルに適用したものが**勾配クリッピング**である (Pascanu et al. 2013)。コード上では、この場合分けは 1 回の `max` に集約でき、クリップされた点において f の勾配が $\lambda^\star$ 倍の制約勾配 $\nabla g(\mathbf{x}) = \mathbf{x}$ によって打ち消されることを直接確認できる。

```
y, r = np.array([3.0, 4.0]), 1.0
lam = max(np.linalg.norm(y) / r - 1, 0.0)    # multiplier: 0 if y is inside the ball
x = y / (1 + lam)                            # x* = y / max(1, |y|/r)
print('x* =', x, ' lambda* =', lam, ' f(x*) =', 0.5 * ((x - y)**2).sum())
print('grad f(x*) =', x - y, ' lambda* grad g(x*) =', lam * x,
      ' sum =', (x - y) + lam * x)
```

```
x* = [0.6 0.8]  lambda* = 4.0  f(x*) = 8.0
grad f(x*) = [-2.4 -3.2]  lambda* grad g(x*) = [2.4 3.2]  sum = [0. 0.]
```

∇f がゼロからほど遠くても停留性は成り立っている。目的関数の勾配 $(-2.4, -3.2)$ と境界からの押し返し $(2.4, 3.2)$ の和がゼロになり、乗数 $\lambda^\star = 4$ が両者の長さの比を与えている。

乗数の符号が持つ幾何学的な意味に注目しよう。球の外へ出ると $g(\mathbf{x})$ は増加するため、その勾配 $\nabla g = \mathbf{x}$ は外向きを指す。停留性を $-\nabla f = \lambda \nabla g$（$\lambda > 0$）と書くと、下り方向 $-\nabla f$ が外向きに引っ張る一方で、境界の壁がそれを押し返していることがわかる。仮に $-\nabla f$ が内側を向いていたとすれば、球の内部へ踏み出すことで f を下げられるため、その点は最適ではあり得ない。したがって負の乗数は排除される。実際、$\lambda = -6$ とすると円上の遠い側の点 $(-0.6, -0.8)$ が選ばれ、これは円板上での最小点ではなく最大点になる。等式制約とは異なり、不等式の壁は 1 つの方向にしか押し返さないのである。

35.6.2.2 カルーシュ・キューン・タッカー条件

不等式制約と等式制約を組み合わせると、次の一般的な最適化問題が得られる。

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \le 0,\; i = 1, \dots, m, \qquad h_j(\mathbf{x}) = 0,\; j = 1, \dots, p. \tag{35.63}$$

この問題の**ラグランジアン**は各制約に 1 つずつ乗数を割り当て、不等式制約の乗数には非負であることを課す。

$$\mathcal{L}(\mathbf{x}, \lambda, \nu) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \, g_i(\mathbf{x}) + \sum_{j=1}^{p} \nu_j \, h_j(\mathbf{x}), \qquad \lambda_i \ge 0. \tag{35.64}$$

制約想定（具体的にどの条件を用いるかは次項で述べる）が成り立つ任意の極小点 $\mathbf{x}^\star$ において、次の 4 つの**カルーシュ・キューン・タッカー（KKT）条件**を満たす乗数が存在する (Karush 1939; Kuhn and Tucker 1951)。

$$\begin{aligned}
&\text{stationarity:} && \nabla f(\mathbf{x}^\star) + \textstyle\sum_i \lambda_i^\star \nabla g_i(\mathbf{x}^\star) + \sum_j \nu_j^\star \nabla h_j(\mathbf{x}^\star) = \mathbf{0}, \\
&\text{primal feasibility:} && g_i(\mathbf{x}^\star) \le 0, \qquad h_j(\mathbf{x}^\star) = 0, \\
&\text{dual feasibility:} && \lambda_i^\star \ge 0, \\
&\text{complementary slackness:} && \lambda_i^\star \, g_i(\mathbf{x}^\star) = 0 \quad \text{for every } i.
\end{aligned} \tag{35.65}$$

最初の 3 つの条件、すなわち勾配の釣り合い（停留性）、制約そのものの成立（主実行可能性）、不等式乗数に対する片方向の符号制限（双対実行可能性）は、これまでの例ですでに見てきたとおりである。4 つ目の**相補性条件**（complementary slackness）は、各 i について $\lambda_i^\star$ と $g_i(\mathbf{x}^\star)$ の少なくとも一方がゼロになることを述べている。その結果、すべての非有効制約（$g_i(\mathbf{x}^\star) < 0$）は自動的に $\lambda_i^\star = 0$ となり、停留性の式から脱落する。ただし、微妙な非対称性に留意しておこう。乗数が正であればその制約は確実に有効であるが、逆に有効制約であっても、たまたま力を及ぼさずに $\mathbf{x}^\star$ に接しているだけならば $\lambda_i^\star = 0$ となりうる。球への射影の例では、相補性条件は手計算で行った場合分けをそのまま符号化しており、点 $(0.6, 0.8)$ では $\lambda^\star g(\mathbf{x}^\star) = 4 \cdot 0 = 0$ が成り立っている。

Figure 35.21 は、停留性と双対実行可能性を合わせた幾何学的構造を示している。最適点において、降下方向 $-\nabla f$ は有効制約の外向き法線の非負線形結合でなければならない。言い換えれば、$-\nabla f$ はそれらの外向き法線が張る錐（**法線錐**（normal cone））の内部に含まれ、非有効制約は一切寄与しない。

ラグランジュ条件と同様、KKT 条件は一階条件であり、それ単独では十分条件にならない。$f(x) = -x^2$ を $[-1, 1]$ 上（すなわち $x - 1 \le 0$ かつ $-1 - x \le 0$）で最小化することを考え

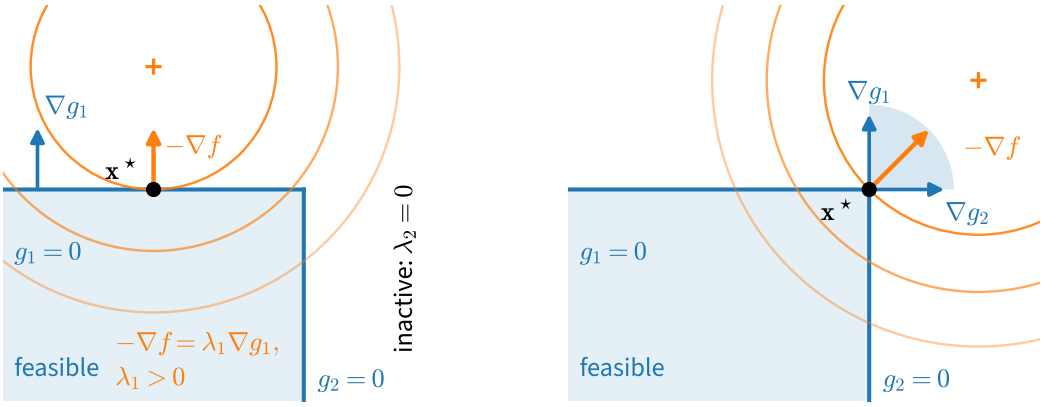


Figure 35.21: 2 つの不等式制約で定義された実行可能領域における KKT 条件の幾何構造。(a) では、一方の制約が $\mathbf{x}^\star$ で有効、もう一方が非有効であり、$-\nabla f$ は $\lambda_1 > 0$ で有効な外向き法線と同じ向きに揃い、$\lambda_2 = 0$ となる。(b) では、角の点で両方の制約が有効となるため、$-\nabla f$ は両方の外向き法線が張る網掛けの法線錐の内部になければならない。オレンジ色の弧は、f の制約なし最小点 (+) のまわりの等位集合を表す。

よう。$x = 0$ では勾配がゼロになり、両方の制約が非有効で、両方の乗数をゼロとおけば 4 つの KKT 条件すべてを満たすが、$x = 0$ は**最大点**である。真の最小点である $x = \pm 1$ も KKT 点であり、それぞれ有効な境界制約に対して乗数 2 を持つ。

35.6.2.3 制約想定と十分性

こうして、互いに補い合う 2 つの問いが残る。第 1 に、極小点が KKT 条件を満たすことが**保証される**のはいつか。第 2 に、KKT 点が最小点であることが**保証される**のはいつか。

1 つ目の問いに対する答えは、等式制約の場合と並行している。すなわち、制約想定が成り立つ限り、極小点は必ず KKT 条件を満たす。最もよく使われる一般的な条件は**線形独立制約想定**（linear independence constraint qualification、LICQ）である。$\mathbf{x}^\star$ を Equation 35.63（ただし f、g_i、h_j は連続微分可能）の極小点としよう。すべての有効な不等式制約の勾配とすべての等式制約の勾配を合わせた集合 $\{\nabla g_i(\mathbf{x}^\star) : g_i(\mathbf{x}^\star) = 0\} \cup \{\nabla h_j(\mathbf{x}^\star) : j = 1, \dots, p\}$ が線形独立であれば、Equation 35.65 を満たす乗数の存在が保証される（Nocedal and Wright (2006) 第 12 章）。LICQ は $\nabla h(\mathbf{x}^\star) \neq \mathbf{0}$ の直接的な一般化であり、これが破れたときに何が起きうるかは演習問題 1 で確認する。

凸問題に対しては、未知の解における勾配を調べる代わりに、1 つの実行可能点を見つけるだけで済むはるかに単純な判定条件がある。問題 Equation 35.63 は、f とすべての g_i が凸関数であり、すべての h_j がアフィン関数であるとき**凸**であるという。凸問題において**スレーター条件**（Slater's condition）(Slater 1950) が成り立つとは、次を満たす点 $\bar{\mathbf{x}}$ が存在することをいう。

$$\begin{aligned} g_i(\bar{\mathbf{x}}) &< 0 \quad \text{for every } g_i \text{ that is not affine}, \\ g_i(\bar{\mathbf{x}}) &\leq 0 \quad \text{for every affine } g_i, \\ h_j(\bar{\mathbf{x}}) &= 0 \quad \text{for every } j. \end{aligned} \tag{35.66}$$

平たく言えば、$\bar{\mathbf{x}}$ は実行可能であり、かつ曲がったすべての不等式制約の真の内部になければならない。すべての制約がアフィンであれば、単に実行可能であるだけでスレーター条件は満た

される。（ここでは扱わないが、f またはいずれかの g_i が $\mathbb{R}^n$ の部分集合上でのみ定義されている場合は、$\bar{\mathbf{x}}$ はその定義域の相対的内部にも属していなければならない。）ユークリッド球の場合は、$\bar{\mathbf{x}} = \mathbf{0}$ とすれば $g(\mathbf{0}) = -r^2/2 < 0$ となるため直ちに条件を満たす。微分可能な f と g_i を持つ凸問題でスレーター条件が成り立つとき、すべての最小点は KKT 点となる (Boyd and Vandenberghe 2004)。その背後にあるより深い理由は強双対性（strong duality）であり、35.7.2 節ではまさに同じ条件からこれを導く。

凸性は 2 つ目の問いにも決着をつけ、本節で最も有用な結果をもたらす。

命題（凸性のもとでの KKT 条件の十分性） 問題 Equation 35.63 において、f とすべての g_i が凸かつ微分可能であり、すべての h_j がアフィンであるとする。$(\mathbf{x}^\star, \lambda^\star, \nu^\star)$ が KKT 条件 Equation 35.65 を満たすならば、$\mathbf{x}^\star$ は大域的最小点である。

証明： $\varphi(\mathbf{x}) = \mathcal{L}(\mathbf{x}, \lambda^\star, \nu^\star)$ とおく。$\lambda_i^\star \geq 0$ であるから、φ は凸関数の非負結合にアフィン項を加えたものであり、したがって φ は凸関数である。停留性より $\nabla \varphi(\mathbf{x}^\star) = \mathbf{0}$ であるから、$\mathbf{x}^\star$ は**大域的な**最小点（φ の $\mathbb{R}^n$ 全体における最小点）となる。ここで任意の実行可能点 $\mathbf{x}$ をとる。$\lambda_i^\star \geq 0$、$g_i(\mathbf{x}) \leq 0$、および $h_j(\mathbf{x}) = 0$ より、次式が成り立つ。

$$f(\mathbf{x}) \;\geq\; f(\mathbf{x}) + \sum_i \lambda_i^\star g_i(\mathbf{x}) + \sum_j \nu_j^\star h_j(\mathbf{x}) = \varphi(\mathbf{x}) \;\geq\; \varphi(\mathbf{x}^\star) = f(\mathbf{x}^\star),$$

ここで最後の等号は、相補性条件（$\sum_i \lambda_i^\star g_i(\mathbf{x}^\star) = 0$）と主実行可能性（$h_j(\mathbf{x}^\star) = 0$）から従う。 ■

この短い証明の中に織り込まれている、より広い原理に注目しよう。**任意の**固定された乗数（$\lambda \geq 0$）に対して、すべての実行可能点 $\mathbf{x}$ で最初の不等式が成り立つため、ラグランジアンを $\mathbf{x}$ に関して最小化すれば、常に制約付き最適値の下界が得られる。そして KKT 乗数においては、その下界は厳密に一致する。35.7 節では、この考察を出発点として双対性（duality）の理論全体を構築する。したがって凸問題においては、KKT 条件は常に十分条件であり、さらにスレーター条件のもとでは必要条件にもなって大域的最小点を完全に特徴づける。

35.6.3 射影と射影勾配降下法

KKT 条件は最適点を**見分ける**方法を教えてくれ、単純な制約であればそれを使って手計算で解くこともできる。より一般的に解を**計算する**には、勾配降下法を最近接の実行可能点を求める操作と組み合わせる。すなわち、通常の勾配ステップを進め、実行可能領域の外へ出るたびにそのまま内側へ射影して戻すのである。

35.6.3.1 凸集合への射影

空でない閉凸集合 $C \subseteq \mathbb{R}^n$ に対し、**射影**とは点 $\mathbf{y}$ を C へ写す操作であり、C 内で $\mathbf{y}$ に最も近い唯一の点として次式で定義される。

$$\Pi_C(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x} \in C} \; \tfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$

なぜ唯一の最小点が常に存在するのだろうか。最適点を除外しないまま、C と $\mathbf{y}$ を中心とする十分大きな閉球との共通部分をとることができるからである。この共通部分は有界閉集合（したがってコンパクト）となるため、連続な目的関数はそこで最小値を達成する。さらに二乗距離の狭義凸性（strict convexity）によって複数の点が同値で並ぶ可能性が排除され、Π_C は一意に定まる写像となる。

すでに超平面への射影（最近接点問題）と球への射影（クリッピングの公式）という 2 つの射影を閉形式で求めた。半空間（half-space）$\{\mathbf{x} : \mathbf{a}^\top \mathbf{x} \geq b\}$ への射影は、超平面の公式から直ちに従う。$\mathbf{y}$ がすでに不等式を満たしていればそのままにし、満たしていなければ $\mathbf{a}$ に沿って境界超平面上までまっすぐ移動する。

$$\Pi(\mathbf{y}) = \mathbf{y} + \frac{\max(0,\ b - \mathbf{a}^\top \mathbf{y})}{\|\mathbf{a}\|^2}\,\mathbf{a}.$$

閉凸集合への射影は、きわめて明快な幾何学的特徴づけを満たす。$\hat{\mathbf{x}} = \Pi_C(\mathbf{y})$ を $\tfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ の C 上での最小点としよう。他の任意の点 $\mathbf{x} \in C$ に対し、凸性により $\hat{\mathbf{x}}$ から $\mathbf{x}$ までの線分全体が C の内部に収まる。したがって $\hat{\mathbf{x}}$ から $\mathbf{x}$ へ向かって動いても 1 次の範囲で目的関数は減少せず、方向微分は非負、すなわち $(\hat{\mathbf{x}} - \mathbf{y})^\top (\mathbf{x} - \hat{\mathbf{x}}) \geq 0$ となる。符号を反転させると、次の**鈍角不等式**（または変分不等式（variational inequality））が得られる。

$$(\mathbf{y} - \Pi_C(\mathbf{y}))^\top (\mathbf{x} - \Pi_C(\mathbf{y})) \ \leq\ 0 \qquad \text{for all } \mathbf{x} \in C. \tag{35.67}$$

幾何学的には、**残差**ベクトル $\mathbf{y} - \Pi_C(\mathbf{y})$ は少なくとも 90° の角を、$\Pi_C(\mathbf{y})$ から任意の点 $\mathbf{x} \in C$ へ向かうベクトルとの間になす（Figure 35.22 の (a)）。言い換えれば、$\Pi_C(\mathbf{y})$ を通り法線 $\mathbf{y} - \Pi_C(\mathbf{y})$ を持つ超平面は C の**支持超平面**（supporting hyperplane）であり、集合 C 全体をその反対側に保つ。

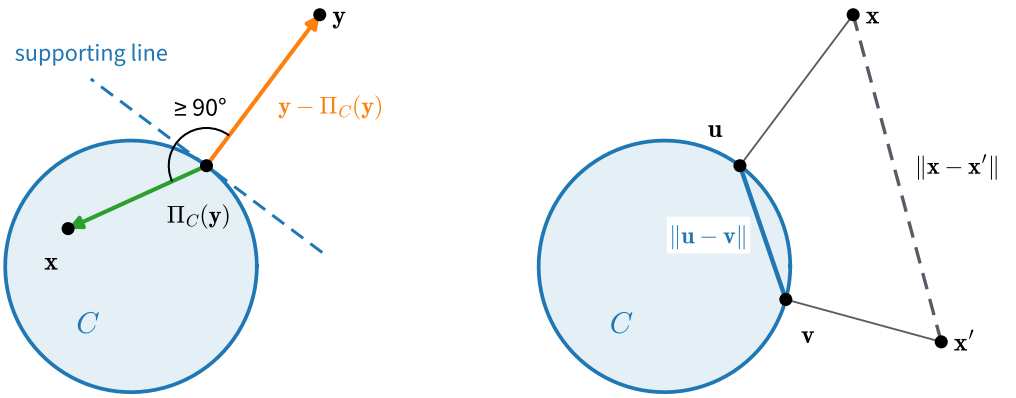


Figure 35.22: (a) 残差 $\mathbf{y} - \Pi_C(\mathbf{y})$ は少なくとも 90° の角を C の内部へ向かうあらゆる方向との間になすため、C は $\Pi_C(\mathbf{y})$ を通る支持直線の反対側に完全に収まる。(b) 閉凸集合への射影は距離を決して増加させない（$\|\mathbf{u} - \mathbf{v}\| \leq \|\mathbf{x} - \mathbf{x}'\|$）。

本節の共通の例でこの不等式を確かめてみよう。単位円板へ射影される $\mathbf{y} = (3, 4)$ に対して、残差は $(2.4, 3.2) = 4\,(0.6, 0.8)$ となり、またしても乗数と外向き法線の積が現れる。任意の $\mathbf{x}$（$\|\mathbf{x}\| \leq 1$）に対して、次式が成り立つ。

$$(2.4, 3.2)^\top (\mathbf{x} - (0.6, 0.8)) = 4\,(0.6 x_1 + 0.8 x_2) - 4 \ \leq\ 4\,\|\mathbf{x}\| - 4 \ \leq\ 0,$$

ここで最初の不等号にはコーシー・シュワルツの不等式 $\mathbf{u}^\top \mathbf{v} \leq \|\mathbf{u}\|\,\|\mathbf{v}\|$ を用いた。ここでの支持直線は $3x_1 + 4x_2 = 5$ であり、まさに先ほどの最近接点の例に現れた超平面そのものである。

同様に重要なこととして、鈍角不等式は**十分条件**でもある。ある点 $\hat{\mathbf{x}} \in C$ が Equation 35.67 を満たすとし、$\varphi(\mathbf{x}) = \tfrac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$ とおこう。その $\hat{\mathbf{x}}$ における勾配は $\hat{\mathbf{x}}-\mathbf{y}$ である。微分可能な凸関数は常にその接平面の上側にあるため（35.3 節）、$\varphi(\mathbf{x}) \geq \varphi(\hat{\mathbf{x}}) + (\hat{\mathbf{x}}-\mathbf{y})^{\top}(\mathbf{x}-\hat{\mathbf{x}})$ が成り立つ。仮定により最後の項はすべての $\mathbf{x} \in C$ に対して非負であるから、$\varphi(\mathbf{x}) \geq \varphi(\hat{\mathbf{x}})$ が C 全体で成り立ち、$\hat{\mathbf{x}} = \Pi_C(\mathbf{y})$ が証明される。したがって、Equation 35.67 は射影を**完全に特徴づける**。射影を用いたアルゴリズムに必要な性質はすべて、射影が決して距離を引き伸ばさないという事実をはじめとして、この不等式から直接導かれる。

命題（射影の非拡大性） 空でない閉凸集合 C とすべての $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ に対して、次式が成り立つ。

$$\|\Pi_C(\mathbf{x}) - \Pi_C(\mathbf{y})\| \;\leq\; \|\mathbf{x}-\mathbf{y}\|.$$

証明： $\mathbf{u} = \Pi_C(\mathbf{x})$ および $\mathbf{v} = \Pi_C(\mathbf{y})$ とおく。Equation 35.67 を 2 回、まず $\mathbf{x}$ に対して $\mathbf{v} \in C$ を用いて、次に $\mathbf{y}$ に対して $\mathbf{u} \in C$ を用いて適用すると、次式が得られる。

$$(\mathbf{x}-\mathbf{u})^{\top}(\mathbf{v}-\mathbf{u}) \leq 0, \qquad (\mathbf{y}-\mathbf{v})^{\top}(\mathbf{u}-\mathbf{v}) \leq 0.$$

この 2 つの不等式を足し合わせると $(\mathbf{x}-\mathbf{y}-(\mathbf{u}-\mathbf{v}))^{\top}(\mathbf{v}-\mathbf{u}) \leq 0$ となり、整理すると次式になる。

$$\|\mathbf{u}-\mathbf{v}\|^2 \;\leq\; (\mathbf{x}-\mathbf{y})^{\top}(\mathbf{u}-\mathbf{v}) \;\leq\; \|\mathbf{x}-\mathbf{y}\|\,\|\mathbf{u}-\mathbf{v}\|$$

ここで後半の不等号はコーシー・シュワルツの不等式による。両辺を $\|\mathbf{u}-\mathbf{v}\|$ で割れば証明が完了する（なお $\|\mathbf{u}-\mathbf{v}\| = 0$ のときは主張は自明である）。$\blacksquare$

パネル (b)（Figure 35.22）はこの性質を視覚的に表している。円板の外にある 2 点間の距離は、常にそれらの射影間の距離以上である。ここでは C の凸性が本質的である。非凸な 2 点集合 $C = \{-1, 1\}$ をとると、$y = 0.01$ は 1 へ射影される一方、$y = -0.01$ は -1 へ射影され、0.02 の入力の差が 2 の出力の差へと拡大してしまう。さらに $y = 0$ では射影が一意ですらなくなる。非凸集合への射影は不連続にも多価にもなりうるのである。

35.6.3.2 射影勾配降下法

射影勾配降下法（projected gradient descent、PGD）は、勾配ステップのたびに射影を施すことで、射影を反復ソルバへと仕立て上げる。

$$\mathbf{x}_{k+1} = \Pi_C\left(\mathbf{x}_k - \eta\,\nabla f(\mathbf{x}_k)\right). \tag{35.68}$$

構成から明らかなように、すべての反復点は実行可能に保たれる。さらに都合のよいことに、PGD の不動点（fixed point）はまさに制約付き停留点と一致する。任意の微分可能な f、閉凸集合 C、およびステップサイズ $\eta > 0$ に対し、不動点方程式 $\mathbf{x}^\star = \Pi_C(\mathbf{x}^\star - \eta\nabla f(\mathbf{x}^\star))$ が成り立つのは、$\mathbf{x}^\star \in C$ が鈍角不等式 Equation 35.67 を $\mathbf{y} = \mathbf{x}^\star - \eta\nabla f(\mathbf{x}^\star)$ に対して満たすとき、かつそのときに限る。$\mathbf{x}^\star$ を相殺して $\eta > 0$ で割ると、この不等式は次の形になる。

$$\nabla f(\mathbf{x}^\star)^{\top}(\mathbf{x}-\mathbf{x}^\star) \;\geq\; 0 \qquad \text{for all } \mathbf{x} \in C.$$

平たく言えば、実行可能な方向の中に降下方向は 1 つも存在しない。これは一般的な制約付き一階条件であり、$\nabla f(\mathbf{x}^\star) = \mathbf{0}$ へと $C = \mathbb{R}^n$ のときに帰着する。f が凸であれば、接平面の不等式からこの条件は直ちに大域的最小点の十分条件となり、f が非凸であれば停留性を保証する。さ

らに $C = \{\mathbf{x} : g_i(\mathbf{x}) \le 0\}$ がスレーター条件を満たす微分可能な凸関数 g_i で定まるとき、この条件は KKT 条件と同値になる。

余分な射影ステップによって勾配降下法は遅くなるのだろうか。驚くべきことに遅くはならず、非拡大性（nonexpansiveness）から 3 行の証明が得られる。

命題（PGD は少なくとも GD と同じ速さで縮小する） C **を空でない閉凸集合とし、** f **は 2 回連続微分可能で** $\mu \mathbf{I} \preceq \nabla^2 f \preceq L\mathbf{I}$ **があるような** $0 < \mu \le L$ **に対して成り立つとし、** $\mathbf{x}^\star$ **を** f **の** C **上での最小点とし、** $\eta > 0$ **とする。このとき、Equation 35.68 の反復列は次式を満たす。**

$$\|\mathbf{x}_{k+1} - \mathbf{x}^\star\| \;\le\; \rho \,\|\mathbf{x}_k - \mathbf{x}^\star\|, \qquad \rho = \max\left(|1 - \eta\mu|,\; |1 - \eta L|\right). \tag{35.69}$$

証明： (i) $\mathbf{x}^\star$ は PGD の不動点であるため、$\mathbf{x}^\star = \Pi_C(\mathbf{b})$（ただし $\mathbf{b} = \mathbf{x}^\star - \eta \nabla f(\mathbf{x}^\star)$）と書ける。(ii) この $\mathbf{b}$ の選び方が急所となるステップであり、非拡大性を $\mathbf{x}_k - \eta\nabla f(\mathbf{x}_k)$ と $\mathbf{b}$ に適用すると、射影がそっくり取り除かれる。

$$\|\mathbf{x}_{k+1} - \mathbf{x}^\star\| \;\le\; \big\|(\mathbf{x}_k - \mathbf{x}^\star) - \eta\left(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^\star)\right)\big\|.$$

(iii) $\mathbf{x}^\star$ と $\mathbf{x}_k$ を結ぶ線分に微分積分学の基本定理を適用すると、$\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^\star) = \bar{\mathbf{H}}(\mathbf{x}_k - \mathbf{x}^\star)$（ただし $\bar{\mathbf{H}} = \int_0^1 \nabla^2 f(\mathbf{x}^\star + t(\mathbf{x}_k - \mathbf{x}^\star))\, dt$）が得られる。この平均化されたヘッセ行列は対称ですべての固有値が $[\mu, L]$ に属するため、$\mathbf{I} - \eta\bar{\mathbf{H}}$ の各固有値は $1 - \eta h$（$h \in [\mu, L]$）の形をしており、その絶対値は高々 ρ である。したがって右辺は高々 $\rho\,\|\mathbf{x}_k - \mathbf{x}^\star\|$ で抑えられる。$\blacksquare$

ステップ (ii) で射影を取り除いてしまえば、あとに残るのは制約なし勾配降下法の縮小だけである。二次関数 $f = \tfrac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x}$ に対しては $\bar{\mathbf{H}} = \mathbf{A}$ となり、ステップ (iii) はモードごとの厳密な縮小率 $|1 - \eta\lambda_i(\mathbf{A})|$（35.1 節）を復元し、その値は 1 より真に小さく保たれる（$0 < \eta < 2/L$ のとき）。強凸ではない凸かつ L 滑らかな f に対しても同様に、$\eta = 1/L$ とした PGD は勾配降下法の $O(1/k)$ の目的関数値の収束率をそのまま引き継ぐ (Nesterov 2018)。

縮小率 ρ が実際にどれほどタイトかを試してみよう。悪条件な二次関数 $f(\mathbf{x}) = \tfrac{1}{2}(x_1^2 + 1000\,x_2^2)$（条件数 $\kappa = 10^3$）を、半空間 $x_1 + x_2 \ge 1$（すなわち $g(\mathbf{x}) = 1 - x_1 - x_2 \le 0$）上で最小化する。KKT 条件は手計算で解ける。制約なし最小点 $\mathbf{0}$ は実行不可能であるため、制約は有効でなければならない。停留性 $(x_1, 1000\,x_2) - \lambda(1,1) = \mathbf{0}$ から $x_1 = \lambda$ および $x_2 = \lambda/1000$ が得られ、これらを $x_1 + x_2 = 1$ に代入すると、$\lambda^\star = 1000/1001 \approx 0.999$、$\mathbf{x}^\star = (1000/1001,\; 1/1001) \approx (0.999, 0.001)$、および $p^\star = 500/1001 \approx 0.4995$ が求まる。

ここで、ステップサイズ $\eta = 0.0015$（安定限界 $2/L = 0.002$ を十分に下回る値）で PGD を実行する。先ほどの命題から、1 ステップあたりの縮小率 $\rho = \max(1 - \eta,\; |1 - 1000\,\eta|) = 0.9985$ が保証される。$(3, 4)$（距離 4.47 だけ $\mathbf{x}^\star$ から離れた点）から出発すると、この上界からは最大 $\log(4.47 \times 10^6)/\log(1/\rho) \approx 10{,}200$ ステップを距離 10^{-6} に達するまでに要すると予測される。以下のコードでは PGD を実行し、1 ステップでの最悪の縮小比、実際のステップ数、毎ステップ境界に当たり始める時点、境界に沿った縮小比、および最終的な勾配を調べる。

```
A = np.diag([1.0, 1000.0])                          # f(x) = 1/2 x^T A x, kappa = 1000
x_star = np.array([1000 / 1001, 1 / 1001])          # KKT point of x1 + x2 >= 1
project = lambda y: y + max(0.0, 1 - y.sum()) / 2   # onto the half-space x1 + x2 >= 1
eta = 1.5e-3
rho = max(abs(1 - eta), abs(1 - 1000 * eta))        # the GD contraction factor
x, dist, projected = np.array([3.0, 4.0]), [], []
for k in range(1000):
```

```
        y = x - eta * A @ x
        projected.append(y.sum() < 1)                  # did the step leave C?
        x = project(y)
        dist.append(np.linalg.norm(x - x_star))
    dist, projected = np.array(dist), np.array(projected)
    ratios = dist[1:] / dist[:-1]
    steps = np.argmax(dist < 1e-6) + 1
    on_line = np.flatnonzero(~projected).max() + 2     # every step from here on is projected
    print(f'bound rho = {rho:.4f}   largest ratio in a step = '
          f'{ratios[dist[:-1] > 1e-12].max():.4f}')
    print(f'steps to |x - x*| < 1e-6: {steps}   every step projected from step {on_line} on')
    print(f'ratio per step on the line: {np.median(ratios[on_line:steps - 1]):.4f}'
          f'   1 - 500.5 eta = {1 - 500.5 * eta:.4f}')
    print(f'grad f at the end = {A @ x}')
```

```
bound rho = 0.9985   largest ratio in a step = 0.9977
steps to |x - x*| < 1e-6: 738   every step projected from step 733 on
ratio per step on the line: 0.2492   1 - 500.5 eta = 0.2492
grad f at the end = [0.999001 0.999001]
```

1 ステップでの最大の比は 0.9977 と $\rho = 0.9985$ に迫っており、1 ステップあたりの最悪ケース評価はほぼタイトである。ところが総ステップ数は嬉しい驚きをもたらす。738 ステップで済み、10,200 ステップには遠く及ばないのである。出力された診断情報からその理由がわかる。軌跡が 2 つの明確な段階に分かれて進むためである。第 1 段階では、最初のステップのあと制約は非有効のままであり、急峻な座標 x_2 は数十回の反復で潰れる一方、平坦な座標 x_1 は制約なし勾配降下法とまったく同様に毎ステップ $1-\eta$ 倍ずつじわじわと内側へ這い進む。$x_1 = 3$ から境界線に到達するまでにおよそ $\log 3/\eta \approx 730$ ステップかかる。ステップ 733 以降は、すべての更新が境界に当たって直線 $x_1 + x_2 = 1$ 上へ射影される。この直線に沿うと二次関数の曲率は $(1+1000)/2 = 500.5$ となるため、$\mathbf{x}^\star$ までの距離は毎ステップ $1 - 500.5\,\eta = 0.249$ 倍に縮まり（直線に沿った 1 次元勾配降下法の厳密な縮小率）、あとほんの数ステップで実行が完了する。ρ は平面上のあらゆる場所における最悪の曲率に備えなければならないのに対し、いったん制約が有効になれば**境界に沿った**曲率だけが効いてくるのである。最後に、終了時点での勾配 $(0.999, 0.999) = \lambda^\star(1,1) = -\lambda^\star \nabla g$ にも注目しよう。ゼロになるどころか、手計算で導いた乗数 $\lambda^\star = 1000/1001$ をそのまま返してくれる。

Π_C が安価に計算できる場面では、PGD は主力となる手法である。非負制約では単に負の成分をゼロにするだけで非負象限（nonnegative orthant）へ射影でき、重み行列の各行に対する最大ノルム制約では更新のたびに各行を球面上へ再スケーリングすればよい。深層学習におけるおそらく最も有名な応用は、入力摂動 δ（$\|\delta\|_\infty \le \epsilon$）に対する射影勾配上昇法によって敵対的サンプルを生成する手法であり、そこでの射影は単に各成分を $[-\epsilon, \epsilon]$ にクリップするだけである (Madry et al. 2018)。なお、勾配クリッピングも球への射影を用いるが、射影するのはパラメータではなく**勾配ベクトル**であるため、重みに対するいかなる制約の PGD でもないことに注意しよう（演習問題 2 でその違いを浮き彫りにする）。クリッピングが達成しているのは、勾配とともに曲率が増大する領域でもステップを安全に保つことである（35.1.1.5 節）。

35.6.3.3 単体への射影

アテンション重みやクラス確率は**確率単体**（probability simplex）$\Delta = \{\mathbf{x} : \mathbf{x} \ge 0,\ \sum_i x_i = 1\}$ の上に乗っている。実数スコアのベクトル $\mathbf{y}$ が与えられたとき、Δ 上で最も近い点をどのように見つければよいだろうか。それには次の問題を考える。

$$\min_{\mathbf{x}} \ \tfrac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2 \quad \text{subject to} \quad \textstyle\sum_i x_i = 1, \ \ -x_i \le 0,$$

これは小規模な**二次計画問題**（quadratic program、QP）、すなわちアフィン制約のもとで凸二次目的関数を最小化する問題であり、KKT 条件を使うと 1 つのスカラーを求める問題に帰着する。和の制約に対する乗数 τ と、乗数 $\lambda_i \ge 0$（非負制約 $-x_i \le 0$ に対するもの）を導入しよう。停留性から $x_i - y_i + \tau - \lambda_i = 0$ が得られ、相補性条件によって各座標は 2 つのグループに分かれる。$x_i > 0$ となる座標では $\lambda_i = 0$ でなければならず、したがって $x_i = y_i - \tau$ となる。一方 $x_i = 0$ となる座標では、停留性から $\lambda_i = \tau - y_i$ となり、双対実行可能性 $\lambda_i \ge 0$ から $y_i \le \tau$ が課される。両方のケースをまとめると次式が得られる。

$$x_i^\star = \max(y_i - \tau,\, 0), \qquad \text{with } \tau \text{ set by } \sum_i \max(y_i - \tau,\, 0) = 1. \tag{35.70}$$

言い換えれば、単体への射影とは、すべてのスコアから同じ閾値 τ を引き、負になった値をゼロにクリップするだけの操作である。注目すべきことに、この閾値の式は**任意の** τ に対して停留性、相補性条件、双対実行可能性、および非負性を自動的に満たしており、和が 1 であるという制約 $\sum_i x_i = 1$ だけが τ を一意に定める。左辺の $\varphi(\tau) = \sum_i \max(y_i - \tau, 0)$ は連続な区分線形関数であり、正の値をとる領域では狭義単調減少であるため、一意な τ（$\varphi(\tau) = 1$ の解）が存在する。

簡単な手計算で τ の見つけ方を確認しよう。すでに降順に並んだベクトル $\mathbf{y} = (0.9, 0.5, -0.2)$ をとり、上位 k 個のスコアがゼロより大きく生き残ると推測してみる。すると和の制約 $\sum_{j\le k}(y_j - \tau) = 1$ から閾値の候補 $\tau_k = (\sum_{j\le k} y_j - 1)/k$ が予測され、これが自己無撞着になるのは k 番目のスコアが τ_k より真に大きく保たれる場合に限られる。

- $k = 1$: $\tau_1 = -0.1$ であり、$y_1 = 0.9 > \tau_1$ を満たす。
- $k = 2$: $\tau_2 = 0.2$ であり、$y_2 = 0.5 > \tau_2$ を満たす。
- $k = 3$: $\tau_3 \approx 0.067$ であるが、$y_3 = -0.2 < \tau_3$ となる。

妥当な最大の選択は $k = 2$ であり、$\tau = 0.2$ および $\mathbf{x}^\star = (0.7, 0.3, 0)$ が得られる（Figure 35.23）。比較として、同じスコアにソフトマックスを適用すると $(0.50, 0.33, 0.17)$ となり、すべての成分が真に正に保たれる。

2 つの単純な考察から、この手計算は一般的なアルゴリズムへと変わる。第 1 に、**サポート** $\{i : x_i^\star > 0\}$ は常に上位のスコアからなる。実際、$y_i > y_j$ かつ $x_j^\star > 0$ ならば $x_i^\star \ge y_i - \tau > y_j - \tau = x_j^\star > 0$ が成り立つ。（**有効な**符号制約は残りのゼロ座標である。）第 2 に、スコアを $u_1 \ge \cdots \ge u_n$ と降順にソートすると、判定条件 $u_k > \tau_k$ は真のサポートサイズ以下のすべての k で成り立ち、それを超えると破綻する。実際、$u_k > \tau_k$ を変形すると $\sum_{j<k}(u_j - u_k) < 1$、すなわち $\varphi(u_k) < 1 = \varphi(\tau)$ となり、φ が狭義単調減少であることから、これは $u_k > \tau$ と同値である。したがって、$\mathbf{y}$ をソートし、最大の k（$u_k > \tau_k$ を満たすもの）を選んで $\tau = \tau_k$ とおけばよく、計算時間は $O(n \log n)$ となる (Held et al. 1974; Duchi et al. 2008)。

アテンションスコアに適用するとき、この単体射影はスコアの低い位置に厳密なゼロの重みを割り当てることから **sparsemax** と呼ばれる (Martins and Astudillo 2016)。ソフトマックスと sparsemax の違いは正則化項 1 つだけである。二乗距離を展開すると、sparsemax は $\mathbf{y}^\top \mathbf{x} - \tfrac{1}{2}\|\mathbf{x}\|^2$ を Δ 上で最大化するのに対し、ソフトマックスは二次のペナルティをシャノンエントロピーに置き換えた $\mathbf{y}^\top \mathbf{x} + H(\mathbf{x})$ を最大化する（35.7 節）。

射影を実装したら、その出力を検証するのがよい習慣である。最初に行う自然なテストは KKT 残差の確認であるが、どの条件が実際に誤りを識別できるかには注意が必要である。以下のセルで

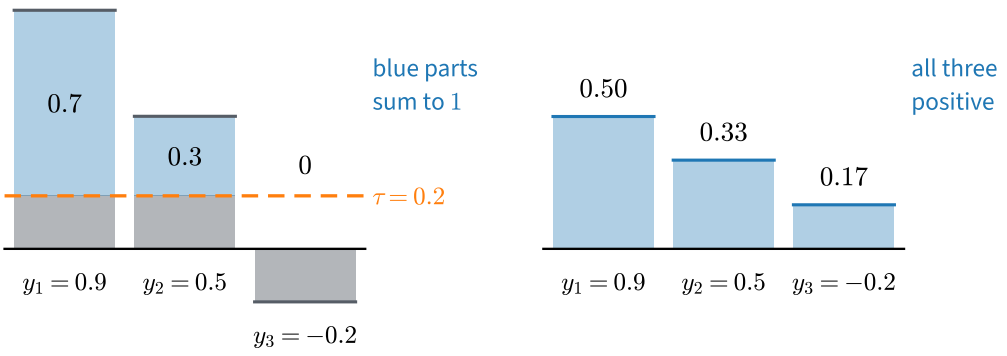


Figure 35.23: $\mathbf{y} = (0.9, 0.5, -0.2)$ の確率単体への射影。(a) ではすべてのスコアが閾値 $\tau = 0.2$ だけ引き下げられてゼロでクリップされ、τ は $\sum_i x_i = 1$ によって一意に定まる。(b) の同じスコアに対するソフトマックスでは、すべての座標が真に正に保たれる。

は、6 個のランダムなスコアに対して、計算された τ と摂動を加えた値 $\tau + 0.3$ の両方でテストし、さらに鈍角不等式 Equation 35.67 による最適性の証明書を確認する。Equation 35.67 の左辺は $\mathbf{x}$ に関してアフィンであるため、Δ 上での最大値は頂点で達成される。したがって、n 個の標準基底ベクトル $\mathbf{e}_i$ を調べるだけで単体全体を検証するのに十分である。

```
def project_simplex(y):
    """Sort-and-threshold projection onto {x >= 0, sum x = 1}; returns x* and tau."""
    u = np.sort(y)[::-1]
    tau_k = (np.cumsum(u) - 1) / np.arange(1, len(y) + 1)   # tau_k for k = 1..n
    k = np.nonzero(u > tau_k)[0][-1]                          # largest k with u_k > tau_k
    return np.maximum(y - tau_k[k], 0.0), tau_k[k]

y = np.random.default_rng().normal(size=6)
x, tau = project_simplex(y)
print('x* =', x.round(3), ' zeros:', int((x == 0).sum()), ' tau =', round(tau, 3))
for t in (tau, tau + 0.3):                       # three conditions hold for every tau ...
    xt, lam = np.maximum(y - t, 0), np.maximum(0, t - y)
    print(f'tau = {t:6.3f}: stationarity {abs(xt - y + t - lam).max():.0e}, '
          f'comp. slack {abs(lam * xt).max():.0e}, dual feas. {lam.min() >= 0}, '
          f'sum x = {xt.sum():.3f}')             # ... only the budget differs
cert = ((y - x) @ (np.eye(6) - x).T).max()      # max_i (y - x*)^T (e_i - x*)
print(f'certificate max_i (y - x*)^T(e_i - x*) = {cert:.0e}  '
      f'(<= 0 up to rounding: {cert <= 1e-12})')
```

```
x* = [0.666 0.334 0.    0.    0.    0.   ]  zeros: 4  tau = 1.662
tau =  1.662: stationarity 0e+00, comp. slack 0e+00, dual feas. True, sum x = 1.000
tau =  1.962: stationarity 0e+00, comp. slack 0e+00, dual feas. True, sum x = 0.400
certificate max_i (y - x*)^T(e_i - x*) = -3e-16  (<= 0 up to rounding: True)
```

通常、6 個の座標のうち 3〜5 個が厳密なゼロに設定される。これらは有効な非負制約であり、それぞれ正の乗数 $\tau - y_i$ を伴う。ここで診断上の落とし穴に注目しよう。停留性、相補性条件、および双対実行可能性は、**両方**の値（正しい τ と誤った値）に対して丸め誤差の範囲内で通って

しまう。誤りを捉えるのは和の制約だけであり、したがってこれら 3 つの残差を調べるだけでは解の正しさを確認できない。一方、頂点による証明書は確実に機能する。丸め誤差の範囲内で $\max_i (\mathbf{y}-\mathbf{x}^\star)^\top(\mathbf{e}_i-\mathbf{x}^\star) \le 0$ であることを確かめれば、単体上のどの点もこれ以上 $\mathbf{y}$ に近づけないことが証明される。

35.6.4 近接作用素と座標降下法

35.6.4.1 ソフト閾値処理と近接作用素

重みを抑制する方法は、硬い境界を設けることだけではない。**Lasso** (Tibshirani 1996) は、硬い壁の代わりに折れ目を持つ ℓ_1 ペナルティを最小二乗損失に加える。

$$F(\mathbf{w}) = \tfrac{1}{2}\|\mathbf{A}\mathbf{w}-\mathbf{b}\|^2 + \lambda\|\mathbf{w}\|_1, \qquad \mathbf{A}\in\mathbb{R}^{n\times d},$$

その最小点は、多くの座標を厳密なゼロに設定するのが特徴である。ここでペナルティの重みをあえて λ と書いたのは、それがノルム制約 $\|\mathbf{w}\|_1 \le c$ に対するラグランジュ乗数のように働くからである。すなわち、F のすべての最小点は、それが消費する ℓ_1 バジェット c のもとで最小二乗損失を最小化しており、逆に任意のバジェット $c>0$ はあるペナルティ λ に対応する。35.7.3.2 節では、Lasso と ℓ_2 重み減衰（weight decay）の両方について、この双方向の対応を証明する。

数値的に何が起きるかを見るため、5 個のゼロ座標を持つスパースなベクトル $\mathbf{w}$ を $\mathbb{R}^8$ 内に用意し、$n=100$ 個のノイズを含む線形観測値を生成して $\lambda=3$ と設定しよう。ステップサイズは $\eta = 1/L$ と選ぶ。ここで $L=\lambda_{\max}(\mathbf{A}^\top\mathbf{A})$ は $\mathbf{A}^\top\mathbf{A}$ の最大固有値であり、最小二乗項の滑らかさの定数である。

```
rng = np.random.default_rng()
w_planted = np.array([2.0, 0, 0, -1.5, 0, 0, 0, 1.0])
A = rng.normal(size=(100, 8))
b = A @ w_planted + 0.05 * rng.normal(size=100)
lam = 3.0
L = np.linalg.eigvalsh(A.T @ A).max()
eta = 1 / L
F = lambda w: 0.5 * ((A @ w - b)**2).sum() + lam * np.abs(w).sum()
soft = lambda z, t: np.sign(z) * np.maximum(np.abs(z) - t, 0.0)  # derived below
print(f'L = lambda_max(A^T A) = {L:.1f};  |a_j|^2 range from '
      f'{(A**2).sum(0).min():.0f} to {(A**2).sum(0).max():.0f}')
```

```
L = lambda_max(A^T A) = 145.6;  |a_j|^2 range from 85 to 109
```

ここで L は通常 130〜190 の範囲に収まるのに対し、列の二乗ノルム $\|\mathbf{a}_j\|^2$（$\mathbf{A}$ の各列）は 100 前後になる。

試すべき最も素朴なアルゴリズムは劣勾配降下法（subgradient descent）である。35.3 節で学んだとおり、$|w|$ の劣勾配（subgradient）は $\operatorname{sign}(w)$（$w\neq 0$ のとき）であり、折れ目では区間 $[-1,1]$ 全体（$w=0$ のとき）となる。各ステップで $\operatorname{sign}(\mathbf{w})$ を選ぶと、更新式 $\mathbf{w} \leftarrow \mathbf{w} - \eta\left(\mathbf{A}^\top(\mathbf{A}\mathbf{w}-\mathbf{b}) + \lambda \operatorname{sign}(\mathbf{w})\right)$ が得られる。これで 5 個のゼロを復元できるだろうか。

```
w_sub, signs = np.zeros(8), []
for k in range(50):                                  # subgradient descent, step 1/L
    w_sub = w_sub - eta * (A.T @ (A @ w_sub - b) + lam * np.sign(w_sub))
    signs.append(np.sign(w_sub))
```

```
null = w_planted == 0
S = np.array(signs)[-20:, null]
print(f'exact zeros: {int((w_sub == 0).sum())} of 8;  sign flips of the null '
      f'coordinates, last 20 steps: {(S[1:] != S[:-1]).sum()} of 95')
print(f'largest null coordinate: {np.abs(w_sub[null]).max():.3f};  '
      f'eta * lam = {eta * lam:.3f}')
```

```
exact zeros: 0 of 8;  sign flips of the null coordinates, last 20 steps: 95 of 95
largest null coordinate: 0.033;  eta * lam = 0.021
```

厳密なゼロを生成することには完全に失敗する。どの座標もゼロに着地せず、5 個の**ヌル**座標（あらかじめゼロに設定した座標）はほぼ毎回の反復で符号を反転させ、約 $\eta\lambda$ の振幅で行ったり来たり振動する。原因はゼロにある折れ目である。真の Lasso 解において、ヌル座標における滑らかな項の勾配は絶対値が λ より真に小さい。そのためゼロの近くでは、ペナルティ項 $-\eta\lambda\,\mathrm{sign}(w_j)$ がステップを支配し、毎回およそ $\eta\lambda$ だけ座標をゼロの反対側へ蹴り出してしまう。劣勾配降下法は各ステップで区間 $[-\lambda, \lambda]$ からただ 1 つの傾きを選んで決め打ちするため、固定ステップサイズでは滑らかな項の勾配をちょうど打ち消す傾きに落ち着くことが決してできないのである。

解決策は単純であり、折れ目のあるペナルティを線形化する代わりに**厳密に**扱えばよい。まずは 1 次元から始め、$\lambda|x| + \frac{1}{2}(x - z)^2$ を $x \in \mathbb{R}$ 上で最小化しよう。目的関数は凸であるため、x が最適であるための必要十分条件は、0 がその劣勾配の集合に含まれることである。

- $x \neq 0$ のとき、目的関数は x で微分可能であるため $\lambda\,\mathrm{sign}(x) + x - z = 0$ となる。$x > 0$ のときはこれから $x = z - \lambda$ が得られ、これには $z > \lambda$ が必要である。一方 $x < 0$ のときは $x = z + \lambda$ が得られ、これには $z < -\lambda$ が必要である。
- $x = 0$ のとき、劣勾配は区間 $[-\lambda, \lambda] - z$ をなし、これが 0 を含むための必要十分条件は $|z| \leq \lambda$ である。

これら 2 つのケースは実数直線を分割し、1 つの式にまとめると**ソフト閾値処理**になる。

$$\mathrm{prox}_{\lambda|\cdot|}(z) \;=\; \mathrm{sign}(z)\,\max\left(|z| - \lambda,\, 0\right). \tag{35.71}$$

絶対値が高々 λ の入力はすべて厳密なゼロへ吸い寄せられ、それより大きい入力はゼロに向かって λ だけ縮小される。たとえば $\lambda = 0.7$ のとき、$z = 0.5$ は 0 へ、$z = 2$ は 1.3 へ、$z = -2$ は -1.3 へと写される（Figure 35.24）。この不感帯（dead zone）$[-\lambda, \lambda]$ こそが、ℓ_1 正則化（regularization）が厳密なスパース性（sparsity）を生み出す仕組みであり、対照的に ℓ_2 正則化は重みをゼロに向かって縮小するだけで決してゼロにはしない。

ソフト閾値処理は、はるかに一般的な操作の 1 つの例にすぎない。**近接作用素**（proximal operator）は、関数 ψ に対して $\psi(\mathbf{x})$ を小さくすることと参照点 $\mathbf{z}$ の近くに留まることのトレードオフをとる。

$$\mathrm{prox}_{\psi}(\mathbf{z}) \;=\; \operatorname*{argmin}_{\mathbf{x}} \left(\psi(\mathbf{x}) + \tfrac{1}{2}\,\|\mathbf{x} - \mathbf{z}\|^2\right). \tag{35.72}$$

この写像はすべての**真閉凸**（proper closed convex）関数 ψ、すなわち凸であり、$\mathbb{R} \cup \{+\infty\}$ に値をとり、少なくとも 1 点で有限値をとり、かつ閉じたエピグラフを持つ関数に対して一意に定まる。これらの条件のもとで、Equation 35.72 の目的関数は 1 強凸となるため、一意な最小点を持つ (Parikh and Boyd 2014)。

3 つの特別な場合を見ることで、近接作用素が既知の概念をいかに統一しているかがわかる。

- $\psi = 0$ のとき、近接作用素は単なる恒等写像である。

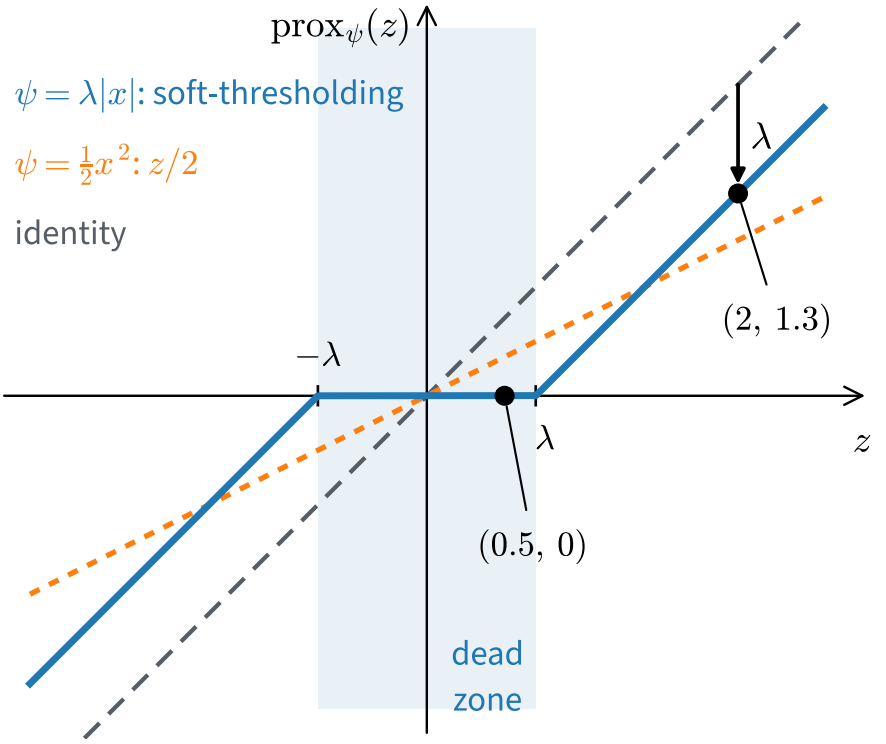


Figure 35.24: ソフト閾値処理 $\mathrm{prox}_{\lambda|\cdot|}(z)$（$\lambda = 0.7$、実線）。不感帯 $[-\lambda, \lambda]$ 内の入力は 0 へ写され（$z = 0.5$ など）、それ以外の入力はすべて 0 に向かって λ だけ移動する（$z = 2$ など）。破線は恒等写像を表す。点線は二乗ペナルティ $\frac{1}{2}x^2$ に対応する写像であり、すべての入力を半分にするが $z = 0$ 以外をゼロにすることはない。

- ψ が閉凸集合 C の**指示関数**（indicator）（値は 0（C 上）および $+\infty$（外部））であるとき、最小点は必ず C に属さなければならないため、近接作用素はユークリッド射影 $\mathrm{prox}_{\psi} = \Pi_C$ に帰着する。近接作用素とは、集合から関数へと一般化された射影なのである。
- $\psi = \lambda\|\cdot\|_1$ のとき、目的関数は座標ごとに完全に分離するため、近接作用素はソフト閾値処理 Equation 35.71 を各座標に適用する操作となる。

演習問題 5 では重みのブロック全体を一括でゼロにするユークリッドノルムの近接作用素を導出し、演習問題 6 ではノルムの近接作用素とその双対球への射影の和が恒等写像になることを示す。

35.6.4.2 近接勾配降下法

Lasso の目的関数は合成形式 $F = s + \psi$ をしており、滑らかな損失 s と、近接作用素を安価に計算できる非平滑なペナルティ ψ を組み合わせたものである。いま、s が L リプシッツ連続な勾配を持つ凸関数であり、ψ が真閉凸であり、$0 < \eta \le 1/L$ であるとする。降下補題（descent lemma）（35.1 節、$1/\eta \ge L$ を使用）より、s は二次モデル $s(\mathbf{x}_k) + \nabla s(\mathbf{x}_k)^{\top}(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2\eta}\|\mathbf{x}-\mathbf{x}_k\|^2$ によって上から抑えられる。ここで ψ まで線形化するのではなく、s のこの二次上界に**厳密な** ψ を加えたものを最小化する。平方完成すると、その最小点は次式で与えられる。

$$\operatorname*{argmin}_{\mathbf{x}} \; \psi(\mathbf{x}) + \frac{1}{2\eta}\left\|\mathbf{x} - (\mathbf{x}_k - \eta\nabla s(\mathbf{x}_k))\right\|^2 \;=\; \mathrm{prox}_{\eta\psi}\left(\mathbf{x}_k - \eta\,\nabla s(\mathbf{x}_k)\right),$$

これは滑らかな部分に対する通常の勾配ステップに続けて、折れ目のある部分に対する近接ステップを施す操作である。この更新を次のように反復する。

$$\mathbf{x}_{k+1} = \mathrm{prox}_{\eta\psi}\left(\mathbf{x}_k - \eta\,\nabla s(\mathbf{x}_k)\right), \tag{35.73}$$

こうして得られる手法が**近接勾配降下法**（proximal gradient descent）であり、特に **ISTA**（iterative shrinkage-thresholding algorithm、反復縮小閾値アルゴリズム）の名で $\psi = \lambda\|\cdot\|_1$ のときに知られている。この手法は射影勾配降下法（ψ として C の指示関数をとる）と通常の勾配降下法（$\psi = 0$ ととる）の両方を包含する。

F が微分可能でなくても、$\eta = 1/L$ とした近接勾配降下法は滑らかな勾配降下法と同じ $O(1/k)$ の収束率 $F(\mathbf{x}_k) - F^\star \le L\|\mathbf{x}_0 - \mathbf{x}^\star\|^2/(2k)$ を達成する（Beck and Teboulle (2009) 定理 3.1）。さらに、s が 2 回連続微分可能かつ μ 強凸（$\mu\mathbf{I} \preceq \nabla^2 s$）であれば、先ほどの PGD の縮小の証明が一字一句そのまま通用する。証明に必要なのは $\mathbf{x}^\star$ が不動点であることと、$\mathrm{prox}_{\eta\psi}$ が Π_C と同様に非拡大であることだけだからである（演習問題 7 でこれを証明する）。

Lasso の場合、近接ステップは閾値 $\eta\lambda$ のソフト閾値処理であり、ゼロから $\eta\lambda$ 以内にあるすべての座標を直接ゼロへと吸い寄せる。この閾値は劣勾配降下法が振動していた振幅とぴったり一致するため、ISTA はヌル座標をきれいにゼロに保てるはずである。実際に確かめてみよう。

```
w = np.zeros(8)
for k in range(50):                                  # ISTA: gradient step, then prox
    w = soft(w - eta * A.T @ (A @ w - b), eta * lam)
print(f'exact zeros: {int((w == 0).sum())} of 8;  support recovered: '
      f'{np.array_equal(w != 0, w_planted != 0)};  '
      f'max deviation on the support: {np.abs(w - w_planted)[~null].max():.3f}')
```

```
exact zeros: 5 of 8;  support recovered: True;  max deviation on the support: 0.029
```

劣勾配降下法と 1 ステップあたり同じ計算コストで同じ 50 回の反復を行うと、ISTA は真のサポートを正確に復元し、5 個のヌル座標すべてをゼロに設定して 3 個の有効な座標を非ゼロのまま残す。これら 3 個の成分は真の値から約 0.05 以内に収まっている。これ以上真の値に大きく近づくことはないが、それは ℓ_1 ペナルティによって真の Lasso 解自体がゼロに向かっておよそ $\lambda/\|\mathbf{a}_j\|^2 \approx 0.03$ だけ縮小されているためである。

35.6.4.3 座標降下法

初読時には本項を読み飛ばしてもよい。座標降下法（coordinate descent）は Lasso に対する標準的なソルバであるが、ニューラルネットワークの訓練に用いられる手法ではないからである。

Lasso において、他のすべての座標を固定したまま 1 つの座標 w_j に沿って最小化する問題は、先ほど解いた 1 次元の問題そのものに帰着する。$\mathbf{r}_{-j} = \mathbf{b} - \sum_{i \neq j} \mathbf{a}_i w_i$ を座標 j を除いて計算した残差としよう。j 番目の部分問題 $\frac{1}{2}\|\mathbf{a}_j w_j - \mathbf{r}_{-j}\|^2 + \lambda |w_j|$ は、定数項を除いて $\|\mathbf{a}_j\|^2 \left(\frac{1}{2}(w_j - c_j/\|\mathbf{a}_j\|^2)^2 + (\lambda/\|\mathbf{a}_j\|^2)|w_j| \right)$ に等しい（ただし $c_j = \mathbf{a}_j^\top \mathbf{r}_{-j}$）。したがって、閾値 $\lambda/\|\mathbf{a}_j\|^2$ のソフト閾値処理によって閉形式で解ける。

$$w_j \leftarrow \frac{\operatorname{sign}(c_j)\,\max(|c_j| - \lambda,\, 0)}{\|\mathbf{a}_j\|^2}, \qquad c_j = \mathbf{a}_j^\top \mathbf{r}_{-j}. \tag{35.74}$$

ある特徴量と現在の残差との相関 c_j の絶対値が高々 λ であれば、その重みは直ちにゼロに設定される。この更新式で座標を順番に巡回する手法こそ、広く使われている `glmnet` パッケージの背後にあるアルゴリズムである (Friedman et al. 2010)。座標を更新するたびに残差を差分更新すれば、全 d 座標を 1 周するスイープのコストは 1 回の勾配計算と同じになる。

同じ原理は、任意の微分可能な目的関数 f が座標ごとの滑らかさの定数 L_j を持ち、次式を満たす場合にも適用できる。

$$f(\mathbf{x} + t\mathbf{e}_j) \le f(\mathbf{x}) + t\,\partial_j f(\mathbf{x}) + \tfrac{1}{2} L_j t^2, \tag{35.75}$$

この二次上界を t に関して最小化すると、次の**座標降下法**の更新式が得られる。

$$x_j \leftarrow x_j - \frac{1}{L_j} \partial_j f(\mathbf{x}). \tag{35.76}$$

L_j は 1 つの座標軸に沿った曲率を測るため、大域的な滑らかさの定数 L より大幅に小さくなり得て、より大きなステップをとることができる。実際、先ほどの Lasso 問題では $L_j = \|\mathbf{a}_j\|^2 \approx 100$ であるのに対し、$L \in [130, 190]$ であった。正定値二次関数 $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x}$ に対しては、$L_j = A_{jj}$ で上界が厳密に一致し、次の更新式が得られる。

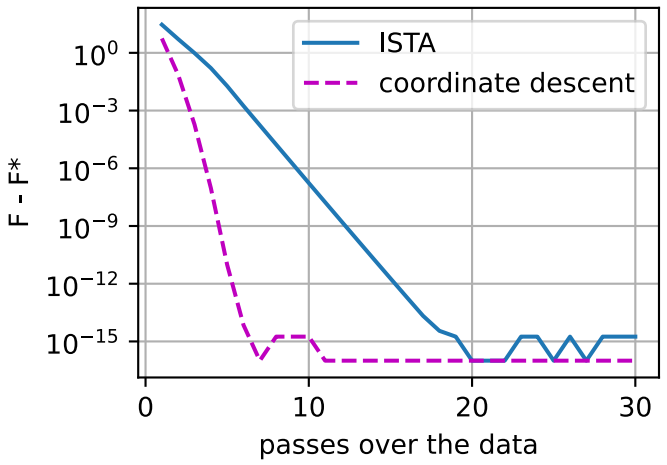
$$x_j \leftarrow x_j - \frac{(\mathbf{A}\mathbf{x} - \mathbf{b})_j}{A_{jj}} \tag{35.77}$$

この更新は f を座標 j に沿って厳密に最小化する。すべての座標を巡回的にスイープすると、$\mathbf{A}\mathbf{x} = \mathbf{b}$ に対する古典的なガウス・ザイデル反復が復元される。

1 回の座標スイープは、同じ計算コストの ISTA の 1 ステップよりどれほど速いだろうか。先ほどの Lasso 問題において、目的関数のギャップ $F(\mathbf{w}) - F^\star$ を 10^{-8} 未満に押し下げるまでに各手法がデータを何回走査するかを数えてみよう。

```
w_ref = np.zeros(8)
for k in range(5000):                               # reference optimum by ISTA
    w_ref = soft(w_ref - eta * A.T @ (A @ w_ref - b), eta * lam)
F_star, col2 = F(w_ref), (A**2).sum(0)              # col2[j] = |a_j|^2
w_i, w_c, r = np.zeros(8), np.zeros(8), b.copy()    # r = b - A w_c, kept up to date
gap_i, gap_c = [], []
for sweep in range(30):
    w_i = soft(w_i - eta * A.T @ (A @ w_i - b), eta * lam)   # one ISTA step
    for j in range(8):                                       # one coordinate sweep
        c_j = A[:, j] @ r + col2[j] * w_c[j]                 # a_j^T r_{-j}
        w_new = soft(c_j, lam) / col2[j]
        r -= A[:, j] * (w_new - w_c[j])
        w_c[j] = w_new
    gap_i.append(F(w_i) - F_star); gap_c.append(F(w_c) - F_star)
passes = lambda gap: next((k + 1 for k, g in enumerate(gap) if g < 1e-8),
                          '> 30')                    # not reached in 30 passes
print(f'passes to gap < 1e-8: ISTA {passes(gap_i)}, coordinate descent {passes(gap_c)}')
d2l.plot(np.arange(1, 31), [np.maximum(gap_i, 1e-16), np.maximum(gap_c, 1e-16)],
         'passes over the data', 'F - F*', legend=['ISTA', 'coordinate descent'],
         yscale='log')                               # gaps floored at rounding level
```

```
passes to gap < 1e-8: ISTA 12, coordinate descent 5
```



対数プロット上では、どちらの手法も直線を描いて減少する（線形収束）が、座標降下法のほうがはるかに急激に減少し、通常はギャップ 10^{-8} に 4〜6 回のスイープで到達するのに対し、ISTA では 12〜25 ステップを要する。本節で証明したどの結果からも、この正確な高速化倍率は予測できない。座標降下法に対する最もシャープな収束率はランダムに座標を選ぶ設定で証明されており (Nesterov 2012)、ここで用いたような巡回的スイープも保証された収束率で収束するものの、その定数はより弱いものになる (Beck and Tetruashvili 2013)。

座標降下法には重要な制約もある。非平滑な折れ目が座標ごとに分離しない場合、座標降下法は最適点から遠く離れた場所で立ち往生しうる。たとえば $f(x, y) = \max(|x|, |y|)$ において点

$(1,1)$ にいるとき、x 単独または y 単独に沿って動いても f を減少させることはできないが、大域的最小値 0 は原点にある。座標降下法が Lasso でうまく機能するのは、その非平滑ペナルティ $\lambda \sum_j |w_j|$ が座標ごとに分離可能だからにほかならない (Tseng 2001)。スパース線形モデルやブロック交互最小化においては、座標降下法は今なお第一選択の手法である。しかし密なニューラルネットワークの訓練では出番がない。現代のアクセラレータは、逐次的なスカラー更新ではなく並列な行列積のために設計されているからである。

本節で行ってきたことすべてを貫く 1 つのパターンに注目しよう。乗数を求めたいときは毎回、まず $\mathbf{x}$ について解いていた。35.7 節ではこの問いを裏返し、乗数を**直接**求める方法、そして任意の乗数の選択が「いかなる実行可能点も所定の下界を下回れない」ことを示す証明書をいかに与えるかを学ぶ。$\lambda^\star = 4$、$p^\star = 8$ であった先ほどのクリッピングの例が、その冒頭を飾ることになる。

35.6.5 まとめ

制約付き最適点では、勾配がゼロになる必要はなく、代わりにいかなる実行可能な方向も目的関数を減少させられないという条件が成り立つ。$\nabla h(\mathbf{x}^\star) \neq \mathbf{0}$ を満たす 1 つの等式制約に対しては、これは $\nabla f + \nu \nabla h = \mathbf{0}$ が、解が制約をどれほど強く押しているかを測るラグランジュ乗数 ν に対して成り立つことを意味する。不等式制約を加えると 4 つの KKT 条件が導かれ、そこでは双対実行可能性（$\lambda_i \geq 0$）が片方向の壁を表し、相補性条件（$\lambda_i g_i = 0$）がすべての非有効制約の乗数をゼロにする。凸問題においてはすべての KKT 点が大域的最小点であり、制約想定（LICQ やスレーター条件など）のもとではすべての最小点が KKT 点となる。

制約付きの解を反復的に計算するため、射影勾配降下法は各勾配ステップに続けてユークリッド射影 Π_C を適用する。閉凸集合への射影は一意であり、鈍角不等式によって特徴づけられ、かつ非拡大であるため、射影勾配降下法は制約なし勾配降下法の収束率をそのまま保つ。球、半空間、および確率単体への射影は、いずれも 1 つのスカラー乗数について解く問題に帰着する。最後に、近接作用素は射影を集合から非平滑な関数へと一般化する。ソフト閾値処理は $\lambda|\cdot|$ の近接作用素であり、ISTA と座標降下法はいずれもこれを用いることで、劣勾配降下法では到達できない厳密なゼロを伴う Lasso の解を求める。

35.6.6 演習問題

1. **退化した制約**。ラグランジュ条件 Equation 35.61 と KKT 条件 Equation 35.65 は、制約想定のもとでのみ最小点において成り立つ。以下の各問題は、いずれかの制約想定に違反している。
    1. $f(\mathbf{x}) = x_1 + x_2^2$ を $\mathbb{R}^3$ 上で制約 $h_1(\mathbf{x}) = x_3 = 0$ および $h_2(\mathbf{x}) = x_3 - x_1^2 = 0$ のもとで最小化せよ。実行可能領域が直線であることを示し、最小点を求め、両方の制約勾配がいたるところで非ゼロであることを示せ。また、最小点においていかなる乗数 ν_1, ν_2 も $\nabla f + \nu_1 \nabla h_1 + \nu_2 \nabla h_2 = \mathbf{0}$ を満たさないことを示せ。さらに、乗数が存在するような別の 2 つの等式制約によって同じ直線を表せ。制約想定とは実行可能領域そのものの性質か、それともそれを記述する関数の性質か。
    2. $f(x) = x$ を $\mathbb{R}$ 上で制約 $x^2 \leq 0$ のもとで最小化せよ。$p^\star$ を求め、最小点においていかなる $\lambda \geq 0$ も KKT 停留条件を満たさないことを示せ。LICQ とスレーター条件のうちどちらが破れているか。それぞれについて理由を説明せよ。
    3. $f(x) = x$ を $\mathbb{R}$ 上で、制約 $-x \leq 0$ を $g_1(x) = g_2(x) = -x$ と 2 回課したもとで最小化せよ。LICQ が破れ、スレーター条件が成り立つことを示せ。最小点において KKT 条件を満たすすべての組 $(\lambda_1^\star, \lambda_2^\star)$ を求めよ。緩和した制約 $-x \leq u_1$ および $-x \leq u_2$ のもとで最適値 $p^\star(u_1, u_2)$ を計算し、すべての KKT 乗数の組が

$p^\star(\mathbf{u}) \geq p^\star(\mathbf{0}) - {\lambda^\star}^\top \mathbf{u}$ をすべての $\mathbf{u}$ に対して満たすことを確かめよ。$p^\star$ は $\mathbf{0}$ で微分可能か。また、関係式 $\lambda^\star = -\nabla p^\star(\mathbf{0})$ に取って代わるものは何か。

2. **射影の不動点。** 閉凸集合 C に対し、射影勾配降下法 Equation 35.68 の不動点は、ステップサイズ $\eta > 0$ の選び方にかかわらず制約付き停留点と一致する。

   1. $C = \{\mathbf{x} : \mathbf{x} \geq 0\}$ とする。$\mathbf{x}^\star$ が不動点であるための必要十分条件は、すべての座標 i について、$\partial f/\partial x_i(\mathbf{x}^\star) = 0$ かつ $x_i^\star \geq 0$ であるか、または $\partial f/\partial x_i(\mathbf{x}^\star) > 0$ かつ $x_i^\star = 0$ であることだと示せ。さらに、これらの条件が制約 $-x_i \leq 0$ に対する KKT 条件に一致することを示し、乗数を $\nabla f(\mathbf{x}^\star)$ を用いて表せ。
   2. 勾配クリッピングは $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\, \Pi_{B_c}(\nabla f(\mathbf{x}_k))$（ただし $B_c = \{\mathbf{z} : \|\mathbf{z}\| \leq c\}$）と更新する。この更新式の不動点を求め、パラメータ球 $\{\mathbf{x} : \|\mathbf{x}\| \leq r\}$ 上への射影勾配降下法の不動点と比較せよ。2 つの更新式のうち、制約付き問題を解いているのはどちらか。
   3. 二値重みはパラメータを非凸集合 $C = \{-1, 1\}$ に制限し、C への射影は近いほうの要素へ丸める（同距離のときは 1 とする）。$f(x) = \tfrac{1}{2}(x - a)^2$（$0 < a < 1$）とせよ。どのようなステップサイズのとき $x = 1$ が Equation 35.68 の不動点になり、どのようなステップサイズのとき $x = -1$ が不動点になるか。η が両方の範囲を超えると反復列はどう振る舞うか。f を C 上で最小化するのはどちらの点か。また、どちらかの点は $f'(x^\star)(x - x^\star) \geq 0$ をすべての $x \in C$ に対して満たすか。この集合に対しては、不動点の議論のどのステップが破綻するか。
   4. 半空間上の PGD のセルを、$\eta = 2.2/1000$（0.0015 の代わり）で実行せよ。射影は非拡大であるにもかかわらず、実行が発散するのはなぜか。縮小に関する命題のどの因子が 1 を超えているか。

3. [code] **Sparsemax と ℓ_1 球。** 関数 `project_simplex` は単体射影 Equation 35.70、すなわち sparsemax を計算し、その第 1 出力が射影後のベクトルである。

   1. すべてのスコアに同じ定数を加えても射影が変わらないことを示せ。スコアを温度 $T > 0$ で割り、$u_1 > u_2$ を $\mathbf{y}$ の上位 2 つの成分とする。$\mathbf{y}/T$ の射影がワンホットになる必要十分条件が $T \leq u_1 - u_2$ であることを示し、$T \to \infty$ としたときの極限を求めよ。10 個の標準正規分布に従うスコアについて、$\mathbf{y}/T$ の射影の非ゼロ成分の個数を T に対してプロットせよ。
   2. S を $\mathbf{x}^\star$ のサポートとし、どの y_i も閾値 τ に等しくないと仮定する。ヤコビ行列 $\partial \mathbf{x}^\star / \partial \mathbf{y}$ を Equation 35.70 から導出し (Martins and Astudillo 2016)、中心有限差分と比較して確かめよ。sparsemax を通る逆伝播は、長さ n の入力に対して何回の算術演算を必要とするか。
   3. ℓ_1 球 $\{\mathbf{x} : \|\mathbf{x}\|_1 \leq z\}$ への射影が $\mathbf{y}$ の各成分の符号を保ち、$\|\mathbf{y}\|_1 > z$ のときには $\operatorname{sign}(\mathbf{y}) \odot z\, \Pi_\Delta(|\mathbf{y}|/z)$ に等しくなることを示せ (Duchi et al. 2008)。これを `project_simplex` を用いて実装せよ。得られた結果を変分不等式 $(\mathbf{y} - \hat{\mathbf{x}})^\top (\mathbf{x} - \hat{\mathbf{x}}) \leq 0$ を $2n$ 個の点 $\pm z\, \mathbf{e}_i$ で調べることで検証せよ。なぜこれらの点だけで十分なのか。

4. **球面上の二階条件。●** ラグランジュ条件は一階条件であり、極小点と鞍点を区別できない。二階の判定法では制約の曲率を考慮に入れる必要がある。

   1. f と h を 2 回連続微分可能とし、$\mathbf{x}^\star$ が Equation 35.61 を乗数 $\nu^\star$ で満たすとする。任意の曲線 $\mathbf{x}(t)$ が $\{h = 0\}$ 上にあり、$\mathbf{x}(0) = \mathbf{x}^\star$ かつ $\dot{\mathbf{x}}(0) = \mathbf{v}$ を満たすとき、次式を示せ。

$$\frac{d^2}{dt^2} f(\mathbf{x}(t))\bigg|_{t=0} = \mathbf{v}^\top \nabla_{\mathbf{x}}^2 \mathcal{L}(\mathbf{x}^\star, \nu^\star)\, \mathbf{v},$$

ここでラグランジアンには Equation 35.62 を用いる。これから、極小点では $\mathbf{v}^\top \nabla^2_{\mathbf{x}} \mathcal{L}\, \mathbf{v} \geq 0$ がすべての接線方向 $\mathbf{v}$ に対して成り立つと結論せよ。この判定に f のヘッセ行列単独を用いるのが誤りであるのはなぜか。なお逆に、接空間上で正定値であれば狭義の極小点となる（Nocedal and Wright (2006) 第 12 章）。

2. $\mathbf{A}$ を相異なる固有値 $a_1 < a_2 < \cdots < a_n$ と単位固有ベクトル $\mathbf{q}_1, \dots, \mathbf{q}_n$ を持つ対称行列とし、$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x}$ を制約 $\mathbf{x}^\top \mathbf{x} = 1$ のもとで最小化せよ。ラグランジュ条件を満たす点が $\pm \mathbf{q}_i$（乗数 $\nu^\star = -a_i$）であることを示し、$\nabla^2_{\mathbf{x}} \mathcal{L}$ の $\mathbf{q}_i$ における接空間上での固有値を求めよ。これらの点のうち、極小点、極大点、鞍点はそれぞれどれか。また、$\mathbf{q}_i$ は互いに独立な降下方向を何本持つか。

3. 球面上への射影勾配降下法 $\mathbf{x} \leftarrow (\mathbf{x} - \eta \nabla f(\mathbf{x}))/\|\mathbf{x} - \eta \nabla f(\mathbf{x})\|$ は非凸集合への射影を行う。この f に対して、この更新が $\mathbf{I} - 2\eta \mathbf{A}$ に対するべき乗法になっていることを示せ。$a_1 + a_n > 0$ と仮定する。ランダムな初期点から出発したとき、反復列 $\mathbf{x}_k$ と関数値 $f(\mathbf{x}_k)$ はどう振る舞うか。その答えは $\eta > 0$ にどう依存するか。

5. **ブロックソフト閾値処理（グループ Lasso）**。ベクトル全体に課すペナルティ $\lambda \|\mathbf{w}\|_2$ は $\mathbf{w} = \mathbf{0}$ でのみ折れ目を持つ。その劣勾配は、$\mathbf{w}/\|\mathbf{w}\|$（$\mathbf{w} \neq \mathbf{0}$ のとき）および単位球内のすべてのベクトル（$\mathbf{0}$ のとき）に λ を掛けたものである。

   1. $\operatorname{prox}_{\lambda \|\cdot\|_2}(\mathbf{z}) = \max\left(0,\ 1 - \lambda/\|\mathbf{z}\|\right) \mathbf{z}$ を示せ。
   2. ゼロが生じるパターンを座標ごとのソフト閾値処理 Equation 35.71 と比較せよ。ブロック近接作用素が厳密に $\mathbf{0}$ を返すのはいつか。また、ブロック内の 1 つの座標だけをゼロにすることはあり得るか。
   3. ペナルティ $\lambda \sum_u \|\mathbf{w}_u\|_2$（出力側の重みベクトル $\mathbf{w}_u$ をネットワークの隠れユニット u にわたって足し合わせたもの）は、ネットワークにどのような作用をもたらすか。
   4. [code] (a) の結果を 2 次元グリッド上の総当たり最小化と比較して、$\mathbf{z} = (0.3, 0.4)$ および $\mathbf{z} = (3, 4)$（$\lambda = 1$）について検証せよ。答えはそれぞれ $\mathbf{0}$ と $(2.4, 3.2)$ になるが、$(2.4, 3.2)$ は本節のどこに現れたか。

6. **モロー分解**。●**凸共役**とは、ψ に対して $\psi^*(\mathbf{y}) = \sup_{\mathbf{x}} \left(\mathbf{y}^\top \mathbf{x} - \psi(\mathbf{x})\right)$ で定義される関数である。真閉凸な ψ に対しては、2 つの事実がこれを近接作用素 Equation 35.72 と結びつける。すなわち、$\mathbf{x} = \operatorname{prox}_{\psi}(\mathbf{z})$ が成り立つ必要十分条件は $\mathbf{z} - \mathbf{x} \in \partial \psi(\mathbf{x})$（劣勾配の集合）であり、かつ $\mathbf{y} \in \partial \psi(\mathbf{x})$ が成り立つ必要十分条件は $\mathbf{x} \in \partial \psi^*(\mathbf{y})$ である (Parikh and Boyd 2014)。

   1. モロー分解 $\mathbf{z} = \operatorname{prox}_{\psi}(\mathbf{z}) + \operatorname{prox}_{\psi^*}(\mathbf{z})$ を導出せよ。
   2. $\psi = \lambda \|\cdot\|_1$ の共役は直方体 $\{\mathbf{y} : \|\mathbf{y}\|_\infty \leq \lambda\}$ の指示関数である。この分解により、ソフト閾値処理 Equation 35.71 が $\mathbf{z}$ からクリップされた $\mathbf{z}$ のコピーを引いた形に書けることを示せ。
   3. $\psi = \lambda \|\cdot\|_2$ の共役は ℓ_2 球（半径 λ）の指示関数であり、その近接作用素は球への射影、すなわち勾配クリッピングにおけるノルムによるクリップ写像である。この分解を用いて、クリッピングの公式からグループ Lasso の演習問題で導いたブロックソフト閾値処理の公式を復元せよ。$\mathbf{z} = (3, 4)$、$\lambda = 1$ のとき、本節に登場したどの 2 つのベクトルの和が $\mathbf{z}$ になるか。

7. [code] **加速近接勾配法**。●FISTA (Beck and Teboulle 2009) は、ISTA の更新式 Equation 35.73（$F = s + \psi$ に対するもの）にモメンタムを加えた手法である。$\mathbf{v}_1 = \mathbf{x}_0$、$t_1 = 1$ とおき、$k \geq 1$ に対して次式を反復する。

$$\begin{aligned} \mathbf{x}_{k} &= \operatorname{prox}_{\eta\psi}\!\left(\mathbf{v}_{k} - \eta \nabla s(\mathbf{v}_{k})\right), \\ t_{k+1} &= \tfrac{1}{2}\Big(1 + \sqrt{1 + 4t_{k}^{2}}\Big), \\ \mathbf{v}_{k+1} &= \mathbf{x}_{k} + \frac{t_{k} - 1}{t_{k+1}}\,(\mathbf{x}_{k} - \mathbf{x}_{k-1}). \end{aligned} \tag{35.78}$$

$\eta = 1/L$ とすると、その目的関数のギャップは $F(\mathbf{x}_k) - F^{\star} \le 2L\|\mathbf{x}_0 - \mathbf{x}^{\star}\|^2/(k+1)^2$ を満たし、ISTA の $O(1/k)$ を上回る。

1. 射影の証明と同様に、近接作用素が非拡大であること、すなわち $\|\operatorname{prox}_{\eta\psi}(\mathbf{u}) - \operatorname{prox}_{\eta\psi}(\mathbf{v})\| \le \|\mathbf{u}-\mathbf{v}\|$ を、$\operatorname{prox}_{\eta\psi}(\mathbf{u})$ と $\operatorname{prox}_{\eta\psi}(\mathbf{v})$ を特徴づける 2 つの劣勾配不等式を足し合わせることで示せ。
2. 同じ Lasso 問題について ISTA のセルに FISTA を追加し、両手法の $F(\mathbf{x}_k) - F^{\star}$ を 100 ステップにわたって対数軸上にプロットせよ。ただし $F^{\star}$ には 10^4 ステップの ISTA で得られた値を用いよ。10 ステップ後ではどちらの手法が先行しており、30 ステップ後ではどちらか。PGD の縮小に関する命題と (a) を用いて、ISTA が直線状に減少する理由を説明せよ。
3. FISTA の反復列に沿って $F(\mathbf{x}_k)$ は単調減少するか。
4. $\mathbf{A}$ を 30×100 のガウスランダム行列に置き換え、同じ 3 個の非ゼロ係数を埋め込んで両手法を 1000 ステップ実行せよ。今度は k のどの範囲で FISTA が先行するか。また、$\mathbf{A}^{\top}\mathbf{A}$ が特異であるにもかかわらず、ISTA が依然として終盤に高速な線形収束段階に入るのはなぜか。

35.7　双対性

反復ソルバの進行が止まり収束が報告されたとき、返された点は本当に最適なのだろうかという自然な疑問が残る。制約のない凸関数であれば、勾配がゼロであることによって直ちに決着がつく（35.3 節）。しかし制約がある場合、最適解は通常、勾配がゼロでない境界上に位置するため（35.6 節）、この単純な判定法は使えなくなる。したがって、数値ソルバにも数学的な証明にも、最適値に対する信頼できる下界という別の道具が必要になる。本節では、そのような下界を体系的に構成する方法を展開し、KKT 条件におけるラグランジュ乗数が実際に何を測っているのかを明らかにする。

この難しさを具体的に見るため、35.6.2 節のクリッピング問題に立ち返ろう。そこでは、二乗距離 $\tfrac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2$（目標点 $\mathbf{x}_0 = (3, 4)$ までの距離）を単位円板 $\|\mathbf{x}\| \le 1$ 上で最小化し、最適解 $\mathbf{x}^{\star} = (0.6, 0.8)$ と最適値 8 を求めた（ここでは目標点を $\mathbf{x}_0$ と書き、$\mathbf{y}$ は使わない。$\mathbf{y}$ は後でラベルおよび凸共役（convex conjugate）の引数として用いるためである）。いま、反復ソルバが $\mathbf{x} = (0.55, 0.835)$ で停止したとしよう。この点は $\|\mathbf{x}\|^2 = 0.9997 \le 1$ を満たすため実行可能点（feasible point）であり、目的関数値 8.01 を達成する。この 8.01 がほぼ最適なのか、それとも円板内の別の点で 7 に到達するのかを、どうすれば判定できるだろうか。真の最適解における勾配は $(-2.4, -3.2)$ とゼロにはほど遠く、円板内には無限に多くの点が含まれるため、1 つずつ調べることは問題外である。

中心となる発想は、制約を目的関数に折り込むことである。非負の係数 $\lambda \ge 0$ を制約関数 $\tfrac{1}{2}(\|\mathbf{x}\|^2 - 1)$ に掛けて目的関数に加える。すべての実行可能点においてこの追加項は非正となるため、修正した目的関数を $\mathbb{R}^2$ **全体**にわたって最小化すれば、その値は必ず制約付き最適値以下になる。このように $\lambda \ge 0$ を選ぶごとに制約なし最小化を通じて下界が得られ、とくに $\lambda = 4$ とすると下界 8 が得られるため、ソルバが返した 8.01 は最適値から 0.01 以内にあることが直ちに

証明される。このような下界のうち最も厳しいものを探すことで、対になる最適化問題である**双対問題**（dual problem）が定義される。注目すべきことに、元の問題が非凸であっても、双対問題は常に凸問題となる。狭義実行可能点（strictly feasible point）を持つ凸問題では、双対問題の最適値は制約付き最適値と一致し、最適乗数は各制約を緩和したときに最適値がどれだけ速く改善するかを表す。

以下ではまず、この下界の考え方を用いてクリッピング問題の最適性を証明し、あらゆる最適化問題に対して成り立つ 2 行の議論である**弱双対性**（weak duality）を述べる。次に、本節で唯一繊細な幾何学的ステップを含む証明によって凸問題に対する**強双対性**（strong duality）を確立し、最適乗数を**シャドープライス**として解釈することで、重み減衰（weight decay）をノルム制約に直接結びつける。続いて、線形制約付き問題の双対を計算する便利な道具として**凸共役**を導入した後（初読時には流し読みしてもよい）、サポートベクターマシン（support vector machine、SVM）、注水法（water-filling）による電力配分、および最良の双対下界が最適値を真に下回る非凸問題という 3 つの示唆に富む例を順に見ていく。計算はすべて NumPy で扱えるほど小規模であり、プロットには `d21` を用いる。

35.7.1 双対問題

最初の目標は、クリッピング問題のいかなる実行可能点も 8 未満の値をとらないことを、実行可能点を 1 つずつ調べることなく証明することである。本節のすべての下界は、任意の実行可能点においてすべての不等式制約関数は非正である、というただ 1 つの基本的な事実に基づいている。

35.7.1.1 1 つの乗数から得られる下界

クリッピング問題を振り返ろう。目的関数 $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2$（ただし $\mathbf{x}_0 = (3, 4)$）を、制約 $g(\mathbf{x}) = \frac{1}{2}(\|\mathbf{x}\|^2 - 1) \le 0$ のもとで最小化する問題である。係数 $\frac{1}{2}$ を g に含めているのは、単にその勾配を $\mathbf{x}$ に一致させるためである。KKT 条件から、解はすでにわかっている。最適解は $\mathbf{x}^\star = \mathbf{x}_0/\|\mathbf{x}_0\| = (0.6, 0.8)$、最適値は $p^\star = \frac{1}{2}(5-1)^2 = 8$ であり、最適乗数は $\lambda^\star = 4$ となる。この乗数は停留条件 $\nabla f(\mathbf{x}^\star) + \lambda^\star \mathbf{x}^\star = (-2.4, -3.2) + 4\,(0.6, 0.8) = \mathbf{0}$ から定まる。

ここで、任意の乗数 $\lambda \ge 0$ と任意の実行可能点 $\mathbf{x}$ をとる。$g(\mathbf{x}) \le 0$ かつ $\lambda \ge 0$ であるから、その積 $\lambda g(\mathbf{x})$ は非正であり、したがって $f(\mathbf{x}) + \lambda g(\mathbf{x}) \le f(\mathbf{x})$ が成り立つ。左辺を $\mathbb{R}^2$ 全体にわたって最小化すれば値はさらに小さくなるか変わらないため、その制約なし最小値は高々 $f(\mathbf{x})$ である。これはすべての実行可能な $\mathbf{x}$ に対して成り立つため、最良の実行可能な $\mathbf{x}$ に対しても成り立つ。

$$\min_{\mathbf{x} \in \mathbb{R}^2} \big[f(\mathbf{x}) + \lambda g(\mathbf{x}) \big] \;\le\; \min_{\|\mathbf{x}\| \le 1} f(\mathbf{x}) \;=\; p^\star \qquad \text{for every } \lambda \ge 0.$$

左辺を具体的に計算してみよう。最小化する関数はラグランジアン $\mathcal{L}(\mathbf{x}, \lambda) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 + \frac{\lambda}{2}(\|\mathbf{x}\|^2 - 1)$ である。任意の $\lambda \ge 0$ に対し、これは $\mathbf{x}$ に関する凸二次関数でヘッセ行列は $(1+\lambda)\mathbf{I}$ となるため、一意な最小点（minimizer）は勾配 $(\mathbf{x} - \mathbf{x}_0) + \lambda \mathbf{x}$ がゼロになる点、すなわち $\mathbf{x}(\lambda) = \mathbf{x}_0/(1+\lambda)$ で達成される。λ を大きくするほど、$\mathbf{x}_0$ は原点に向かってより強く縮小される。この $\mathbf{x}(\lambda)$ をラグランジアンに代入すると（$\mathbf{x}(\lambda) - \mathbf{x}_0 = -\lambda \mathbf{x}_0/(1+\lambda)$ および $\|\mathbf{x}_0\|^2 = 25$ を用いる）、$\frac{1}{2}\|\mathbf{x}(\lambda) - \mathbf{x}_0\|^2 = \frac{25}{2}\lambda^2/(1+\lambda)^2$ および $\frac{\lambda}{2}\|\mathbf{x}(\lambda)\|^2 = \frac{25}{2}\lambda/(1+\lambda)^2$ となる。これら 2 つの分数を合わせると因子 $(1+\lambda)$ が約分され、最小値は λ の単純な関数として残る。

$$q(\lambda) \;=\; \frac{25}{2}\,\frac{\lambda}{1+\lambda} \;-\; \frac{\lambda}{2}. \tag{35.79}$$

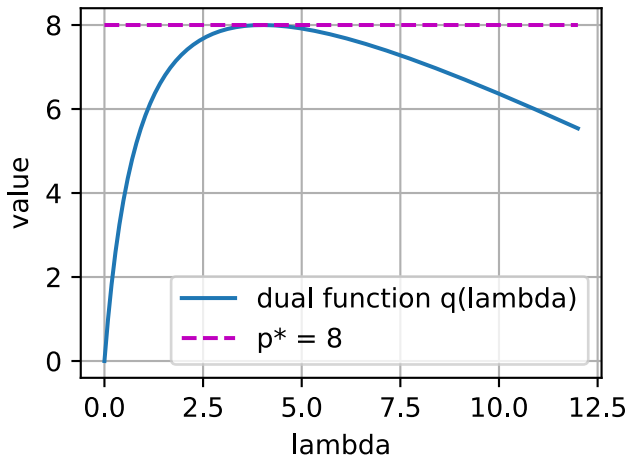
これら 2 つの項は互いに逆方向へ引っ張り合う。λ が大きくなるにつれて、第 1 項は 12.5 に向かって増加する一方、第 2 項（定数 $-\frac{1}{2}$ が g の中に含まれることに由来する項）は際限なく線形に減少する。最も厳しい下界は、これら 2 つの効果が釣り合う点にある。

いくつか数値を代入してみると、この釣り合いが具体的によくわかる。$q(0) = 0$、$q(1) = 5.75$、$q(2) = 7.33$、$q(3) = 7.875$、$q(4) = 8$、$q(5) = 7.92$、$q(8) = 7.11$ である。これらの値はどれ 1 つをとっても $p^\star$ に対する正当な下界となっている。微分すると $q'(\lambda) = \frac{25}{2(1+\lambda)^2} - \frac{1}{2}$ となり、これは $(1+\lambda)^2 = 25$、すなわち $\lambda = 4$ のときにゼロになる。さらに $q''(\lambda) = -25/(1+\lambda)^3 < 0$ であるから、$\lambda = 4$ が一意な最大点であり、$q(4) = 10 - 2 = 8 = p^\star$ を達成する。最良の下界は真の最適値に一致し、それを達成する乗数は KKT 乗数にほかならない。これにより、ソルバが得た値 8.01 は最適値から 0.01 以内にあることが保証される。最適化していない $\lambda = 3$ のような適当な値でさえ、すでに最適値から 0.135 以内にあることを保証してくれる。

なぜ $\lambda = 4$ で最適値に到達するのだろうか。ラグランジアンの最小点における制約値 $g(\mathbf{x}(\lambda)) = \frac{1}{2}(25/(1+\lambda)^2 - 1)$ に注目しよう。$\lambda < 4$ のときは乗数が弱すぎて最小点を単位円板の内側へ引き込めず（たとえば $\mathbf{x}(1) = (1.5, 2)$ のノルムは 2.5 である）、$g(\mathbf{x}(\lambda)) > 0$ となって下界は実行不可能な点を反映したものになる。逆に $\lambda > 4$ のときは乗数が最小点を内側へ引き込みすぎてしまい（たとえば $\mathbf{x}(8)$ のノルムは $\frac{5}{9}$ である）、$\lambda g(\mathbf{x}(\lambda)) < 0$ が下界を引き下げてしまう。ちょうど $\lambda = 4$ のときに最小点 $\mathbf{x}(4) = (0.6, 0.8)$ が境界円上に乗り、ペナルティ項 $\lambda g(\mathbf{x}(\lambda))$ がゼロとなって、下界が実行可能点の目的関数値と一致するのである。

$q(\lambda)$ と $p^\star$ を λ の値のグリッド上で並べてプロットし、これらの下界がどのように振る舞うかを確認しよう。

```
x0 = np.array([3.0, 4.0])                          # the point to clip
f = lambda x: 0.5 * np.sum((x - x0) ** 2)          # objective
g = lambda x: 0.5 * (np.sum(x ** 2) - 1.0)         # constraint g(x) <= 0
x_of = lambda lam: x0 / (1 + lam)                  # minimizer of L(., lam)
q = lambda lam: f(x_of(lam)) + lam * g(x_of(lam))  # dual function
p_star = f(x0 / np.linalg.norm(x0))                # value at the clipped point
lams = np.linspace(0.0, 12.0, 1201)
qs = np.array([q(l) for l in lams])
d2l.plot(lams, [qs, np.full_like(lams, p_star)], 'lambda', 'value',
         legend=['dual function q(lambda)', 'p* = 8'])
```



得られた曲線は凹関数であり、至るところで水平線 $p^\star = 8$ の下側に留まりつつ、$\lambda = 4$ で 1 度だけ接する。驚くべきことに、ここでの凹性は円板の幾何形状とは何の関係もない。以下で示すように、双対関数（dual function）は**あらゆる**最適化問題に対して凹関数となる。

35.7.1.2 ラグランジュ双対関数

上の下界の議論には、クリッピング問題に固有の性質は何ひとつ使われていなかった。同じ構成法を 35.6.2 節の一般の制約付き問題に適用しよう。

$$\min_{\mathbf{x}\in\mathcal{D}} \; f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \le 0,\; i = 1,\dots,m, \qquad h_j(\mathbf{x}) = 0,\; j = 1,\dots,p,$$

この問題の最適値を $p^\star$ とし、すべての実行可能点にわたる f の下限として定義する（実行可能点が存在しない場合は $p^\star = +\infty$ とする）。ここで $\mathcal{D}$ は f およびすべての制約関数が定義される定義域である。これは通常単なる $\mathbb{R}^n$ であるが、本節末尾の双対ギャップ（duality gap）の例で $[0,1]$ に対して行うように、ある集合をハード制約として残す場合はその集合となる。ラグランジアンは各制約に対して 1 つずつ乗数を導入し、次式で定義される。

$$\mathcal{L}(\mathbf{x}, \lambda, \nu) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \, g_i(\mathbf{x}) + \sum_{j=1}^{p} \nu_j \, h_j(\mathbf{x}),$$

ここで、不等式制約には $\lambda_i \ge 0$ を対応させ、等式制約には符号を制限しない $\nu_j \in \mathbb{R}$ を対応させる。また、$\lambda \succeq 0$ と書いて、$\lambda_i \ge 0$ がすべての座標 i について成り立つことを表す。

乗数を固定してラグランジアンを $\mathbf{x}$ について最小化すると、下界の構成は (λ, ν) の関数になる。**ラグランジュ双対関数**（Lagrange dual function）は次式で定義される。

$$q(\lambda, \nu) = \inf_{\mathbf{x}\in\mathcal{D}} \; \mathcal{L}(\mathbf{x}, \lambda, \nu). \tag{35.80}$$

言葉で言えば、$q(\lambda,\nu)$ とは、f がどこまで小さくなりうるかを、ハード制約を取り除いて λ と ν で重み付けした線形ペナルティに置き換えたもとで問うものである。クリッピング問題では、これにより Equation 35.79 が得られた。一般に、この下限は達成されるとは限らず、$-\infty$ になることさえある。たとえば、x を制約 $-x \le 0$ のもとで最小化する場合を考えよう。ラグランジアン $(1-\lambda)x$ は $\lambda = 1$ でない限り下に有界でないため、$q(1) = 0$ であり、$q(\lambda) = -\infty$ がそれ以外のすべての λ に対して成り立つ。$-\infty$ という下界は自明に正しいが何の役にも立たない。$q(\lambda,\nu) > -\infty$ となる乗数の集合を q の**実効定義域**（effective domain）と呼ぶ。後ほどサポートベクターマシンの双対問題を導出する際、この暗黙の定義域制限は明示的な等式制約として現れる。

q の 2 つの基本的な性質は、f、g_i、h_j に凸性や微分可能性すら含むいかなる仮定も置かずに成り立つ。

命題（双対関数は凹関数である）。 **任意の主問題（primal problem）に対し、$q(\lambda,\nu)$（Equation 35.80）は凹関数である。**

証明。 任意の $\mathbf{x} \in \mathcal{D}$ を固定する。乗数は $\mathcal{L}$ に線形に入るため、写像 $(\lambda,\nu) \mapsto \mathcal{L}(\mathbf{x},\lambda,\nu)$ はアフィンである。双対関数 q は、このアフィン関数族の各点ごとの下限である。したがって、その**ハイポグラフ**（グラフ上またはその下側にある点の集合）は、これらのアフィン関数の下側にある半空間の共通部分となる。凸集合の任意の共通部分は凸集合であるため、q は凹関数である。■

Figure 35.25 に、クリッピング問題に対するこの構成を示す。各破線は固定された 1 つの $\mathbf{x}$ に対するラグランジアンを λ の関数として描いたものであり、q はそれらすべての直線の下側に留まる曲線である。クリッピング問題の目的関数はたまたま凸関数であるが、上の議論ではその性質を一切使っていない。直線群の下側包絡線（lower envelope）は、f と g がどのような形状であっても凹関数になる。

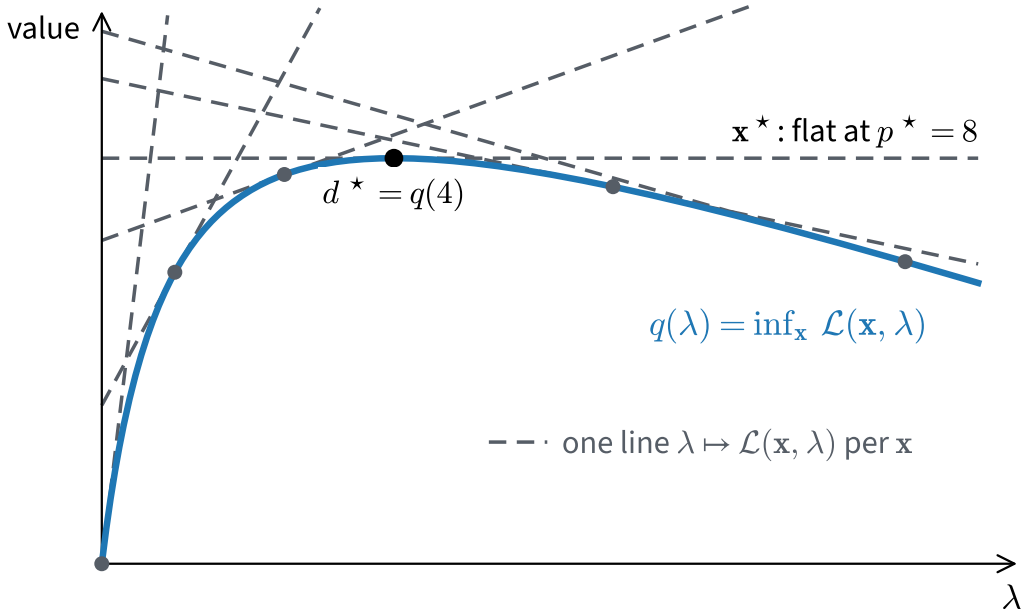


Figure 35.25: 下側包絡線としてのクリッピング問題の双対関数。固定された各 $\mathbf{x}$ は 1 本の直線 $\lambda \mapsto \mathcal{L}(\mathbf{x}, \lambda)$（破線）を与え、$q(\lambda)$ はそれらの直線が λ でとる最小値となる。描かれている直線は最小点 $\mathbf{x}(\lambda) = \mathbf{x}_0/(1+\lambda)$（いくつかの λ の値に対するもの）に対応し、それぞれが q に自身の λ で接している（黒丸）。$\mathbf{x}^\star = \mathbf{x}(4)$ の直線は $g(\mathbf{x}^\star) = 0$ のため水平であり、q に最大値 $d^\star = q(4) = p^\star = 8$ で接する。

命題（弱双対性）。$\mathbf{x}$ が実行可能であり $\lambda \succeq 0$ ならば、$q(\lambda, \nu) \le f(\mathbf{x})$ が成り立つ。したがって、次式を得る。

$$d^\star \;=\; \sup_{\lambda \succeq 0,\, \nu} q(\lambda, \nu) \;\le\; \inf_{\mathbf{x} \text{ feasible}} f(\mathbf{x}) \;=\; p^\star. \tag{35.81}$$

証明。任意の実行可能な $\mathbf{x}$ と任意の $\lambda \succeq 0$ に対し、$\lambda_i g_i(\mathbf{x}) \le 0$（$\lambda_i \ge 0$ かつ $g_i(\mathbf{x}) \le 0$ であるため）および $\nu_j h_j(\mathbf{x}) = 0$ が成り立つ。したがって、次式が成り立つ。

$$q(\lambda, \nu) \;\le\; \mathcal{L}(\mathbf{x}, \lambda, \nu) \;=\; f(\mathbf{x}) + \sum_i \lambda_i g_i(\mathbf{x}) + \sum_j \nu_j h_j(\mathbf{x}) \;\le\; f(\mathbf{x}).$$

左辺で (λ, ν)（ただし $\lambda \succeq 0$）にわたる上限をとり、右辺で実行可能な $\mathbf{x}$ にわたる下限をとれば、Equation 35.81 が得られる。■

弱双対性は、クリッピング問題におけるすべての $q(\lambda)$ を保証付きの下界に変えた 2 行の不等式にほかならない。またこれは、35.6.2 節における KKT 条件の十分性を支える仕組みその

ものでもある。そこでは同じ不等式の連鎖を KKT 乗数において評価することで、相補性条件（complementary slackness）により両方の不等号が等号で成立する。

$q(\lambda, \nu)$ を $\lambda \succeq 0$ および ν について最大化することを**双対問題**と呼ぶ。これは凸集合上で凹関数を最大化する問題であるため、たとえ主問題がどれほど強い非凸性を持っていても、双対問題は**常に**凸最適化問題となる。その最適値 $d^\star$ はこの構成法が与えうる最も厳しい下界であり、双対実行可能なすべての (λ, ν) が正当な下界を与える。非負の差 $p^\star - d^\star \geq 0$ を**双対ギャップ**と呼ぶ。クリッピング問題ではこのギャップはゼロであったが、本節最後の具体例では真に正のギャップが生じる。となると次に生じる自然な問いは、ギャップがゼロになることを事前に保証できるのはいつか、ということである。

35.7.2 強双対性

クリッピング問題では、最良の双対下界が主最適値と厳密に一致した。しかし非凸問題では、双対下界が主最適値に真に届かず、解の一部しか保証できないことがある。幸い、凸性とただ 1 つの狭義実行可能点という 2 つの穏やかな条件さえあれば、このギャップを埋めるには十分である。達成可能な制約値と目的関数値を単純な幾何学的視点から眺めると、なぜこの両方が必要なのかがよくわかる。

35.7.2.1 スレーター条件

命題（スレーター条件は強双対性を導く）。問題が凸であり（f とすべての g_i が凸、すべての h_j がアフィン）、スレーター条件（35.6.2.3 節）を満たすとする。すなわち、実行可能点 $\bar{\mathbf{x}}$ が $\mathcal{D}$ の相対的内部（relative interior）に存在し、$g_i(\bar{\mathbf{x}}) < 0$ がアフィンでないすべての不等式制約 g_i に対して成り立つとする。このとき $d^\star = p^\star$ が成り立ち、さらに $p^\star$ が有限であれば双対最適解が達成される。

これは 35.6.2 節で凸問題に対して導入した制約想定（constraint qualification）と同じものである。すなわち、湾曲したすべての不等式制約の真に内側に少なくとも 1 つの点が存在しなければならない（多次元における完全な証明については Boyd and Vandenberghe (2004) の 5.3.2 節を参照）。クリッピング問題はスレーター条件を容易に満たす。f と g はともに凸であり、原点 $\mathbf{0}$ は $g(\mathbf{0}) = -\tfrac{1}{2} < 0$ を満たすからである。したがって、導関数を 1 度も計算する前から $q(4) = p^\star$ は保証されていたのである。

凸性と狭義実行可能性が**なぜ**ギャップを埋めるのかを理解するため、単一の不等式制約 $g(\mathbf{x}) \leq 0$ を持つ問題を考えよう。すべての点 $\mathbf{x} \in \mathcal{D}$ をその制約値と目的関数値の組 $(g(\mathbf{x}), f(\mathbf{x}))$ に写し、これらの組の上側および右側にあるすべての点を集めた集合を作る。

$$\mathcal{A} = \big\{(u, t) : u \geq g(\mathbf{x}),\ t \geq f(\mathbf{x}) \text{ for some } \mathbf{x} \in \mathcal{D}\big\}.$$

この (u, t) 平面において実行可能点は $u \leq 0$ に対応するため、主最適値 $p^\star$ とは単に高さ t のうち $\mathcal{A}$ が縦軸 $u = 0$ 上で到達する最小値にほかならない。いま $\lambda \geq 0$ を固定する。すべての (u, t) のうち与えられた $\mathbf{x}$ から生じるものの上では、線形関数 $t + \lambda u$ は $(g(\mathbf{x}), f(\mathbf{x}))$ で最小となるため、$q(\lambda) = \inf_{(u,t)\in\mathcal{A}} (t + \lambda u)$ となる。幾何学的には、直線 $t + \lambda u = c$（傾き $-\lambda$）を下から押し上げ、ちょうど $\mathcal{A}$ を下から支えるところで止めたときの縦軸 $u = 0$ 上の切片が $q(\lambda)$ である。集合 $\mathcal{A}$ 全体がこの支持直線（supporting line）の上または上側に位置するため、切片 $q(\lambda)$ が $p^\star$ を超えることは決してない。これが弱双対性を視覚化した姿である（Figure 35.26）。

この幾何学的描像があれば、単一の不等式制約に対する強双対性の証明は見通しのよいものになる。

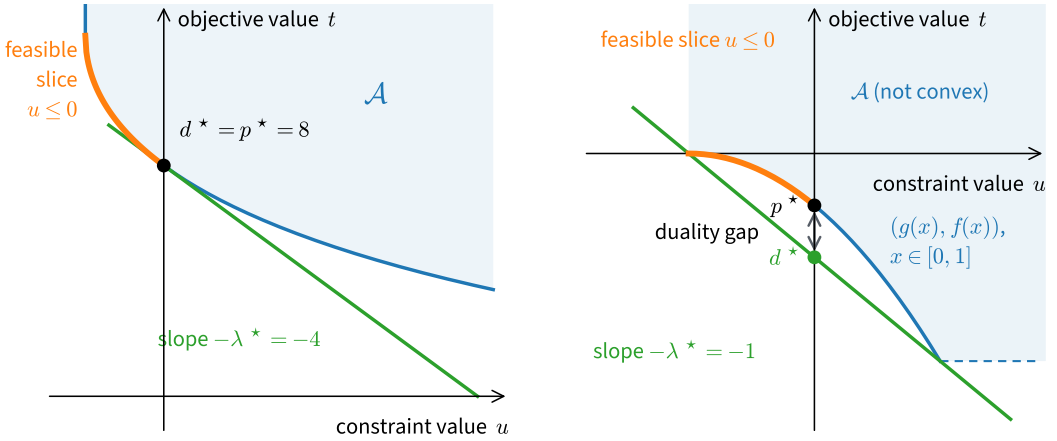


Figure 35.26: 双対性の支持直線による幾何構造。各 $\mathbf{x}$ は $(g(\mathbf{x}), f(\mathbf{x}))$ に写され、網掛け領域 $\mathcal{A}$ はこれらの組と、その上側および右側にあるすべての点を含む。双対関数を λ で評価することは、傾き $-\lambda$ の直線を $\mathcal{A}$ を下から支えるまで押し上げることに相当し、$u = 0$ におけるその切片が $q(\lambda)$ となる。(a) クリッピング問題：$\mathcal{A}$ は凸集合で $\mathbf{0}$ は狭義実行可能点であるため、最適な支持直線は $(0, p^\star)$ に傾き -4 で接し、$d^\star = p^\star = 8$ を与える。(b) $-x^2$ を $[0,1]$ 上で制約 $x \le \frac{1}{2}$ のもとで最小化する問題：$\mathcal{A}$ が非凸であるため、最も高い支持直線でも $(0, p^\star)$ の真に下側を通り、$d^\star = -\frac{1}{2} < -\frac{1}{4} = p^\star$ となる。

証明 ($g(\bar{\mathbf{x}}) < 0$ かつ $p^\star$ が有限である単一不等式制約の場合)。 第 1 に、$\mathcal{A}$ は凸集合である。任意の 2 点 (u_1, t_1) と (u_2, t_2) を $\mathcal{A}$ からとり（それぞれ $\mathbf{x}_1$ と $\mathbf{x}_2$ に対応するとする）、任意の $\theta \in [0,1]$ をとる。g の凸性より、内分点 $\mathbf{x}_\theta = \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2$ は $g(\mathbf{x}_\theta) \le \theta g(\mathbf{x}_1) + (1-\theta) g(\mathbf{x}_2) \le \theta u_1 + (1-\theta) u_2$ を満たし、同じ不等式が f に対しても成り立つ。ゆえに (u_1, t_1) と (u_2, t_2) の凸結合（convex combination）は $\mathcal{A}$ に属する。

第 2 に、$(0, p^\star)$ は $\mathcal{A}$ の境界上にある。すなわち、この点は $\mathcal{A}$ 内の点列の極限である一方、$(0, p^\star - \varepsilon) \notin \mathcal{A}$ がすべての $\varepsilon > 0$ に対して成り立つ。凸集合の任意の境界点には支持直線が存在する。すなわち、ゼロでないベクトル $(\lambda, \gamma) \neq (0,0)$ が存在して、$\gamma t + \lambda u \ge \gamma p^\star$ がすべての $(u,t) \in \mathcal{A}$ に対して成り立つ（Boyd and Vandenberghe (2004) の 2.5.2 節を参照。これは 35.3 節で劣勾配（subgradient）を導いた幾何学的事実と同じものである）。$\mathcal{A}$ は上方向および右方向に限りなく広がっているため、$\gamma \ge 0$ かつ $\lambda \ge 0$ でなければならない。

では、$\gamma = 0$ となることはありうるだろうか。もしそうだとすると、支持直線は垂直で $\lambda > 0$ となり、$\lambda u \ge 0$（したがって $u \ge 0$）が $\mathcal{A}$ 全体で成り立つことになる。しかしこれは、狭義実行可能点 $\bar{\mathbf{x}}$ が $u = g(\bar{\mathbf{x}}) < 0$ を与えることと矛盾する。したがって $\gamma > 0$ である。両辺を γ で割ると、$t + (\lambda/\gamma)\, u \ge p^\star$ がすべての $(u,t) \in \mathcal{A}$ に対して得られ、これは $q(\lambda/\gamma) \ge p^\star$ を意味する。弱双対性と合わせれば $q(\lambda/\gamma) = p^\star$ となり、双対最適解が達成されてその値は $p^\star$ に一致する。■

この証明において、2 つの仮定がいかにきれいに役割分担しているかに注目しよう。凸性は $\mathcal{A}$ を凸集合にし、$(0, p^\star)$ における支持直線の存在を保証するために 1 度だけ使われる。スレーター条件は、その支持直線が（無限大の乗数に対応する）垂直線になるのを防ぐために 1 度だけ使われる。Figure 35.26 のパネル (a) のクリッピング問題では、ノルム $\|\mathbf{x}\| = r$ の点は最小目的関数値 $\frac{1}{2}(5-r)^2$ と制約値 $u = \frac{1}{2}(r^2 - 1)$ を持つため、$\mathcal{A}$ の下側境界は曲線 $t = \frac{1}{2}\big(5 - \sqrt{1+2u}\big)^2$ となる。その $u = 0$ における傾きは -4 であり、傾き $-\lambda^\star = -4$ の最適な支持直線と一致する。パネル (b) は本節末尾で扱う非凸の例を先取りしたものである。そこでは $\mathcal{A}$ が逆向きに湾曲して

いるため、$\mathcal{A}$ の下側にあるどの直線も $(0, p^\star)$ の真に下を通過してしまう。

35.7.2.2 鞍点

強双対性は、双対関数に言及することすらなく、ラグランジアンのみを用いて言い換えることもできる。実際、主問題は min–max 問題として書き直せる。任意の $\mathbf{x} \in \mathcal{D}$ に対して、次式が成り立つ。

$$\sup_{\lambda \succeq 0,\, \nu} \mathcal{L}(\mathbf{x}, \lambda, \nu) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \text{ is feasible,} \\ +\infty & \text{otherwise.} \end{cases}$$

もし $\mathbf{x}$ がいずれかの制約に違反していれば、対応する乗数を大きくすることで $\mathcal{L}$ を $+\infty$ へ発散させられる。一方、$\mathbf{x}$ が実行可能であれば、乗数側にできる最善の選択は余裕のあるすべての制約に対して $\lambda_i = 0$ と置くことであり、結果として $f(\mathbf{x})$ が残る。したがって $p^\star = \inf_{\mathbf{x}} \sup_{\lambda \succeq 0, \nu} \mathcal{L}$ となる一方、定義より $d^\star = \sup_{\lambda \succeq 0, \nu} \inf_{\mathbf{x}} \mathcal{L}$ である。弱双対性とは、任意の 2 変数関数に対して成り立つ普遍的な不等式 $\sup \inf \le \inf \sup$ にほかならない。強双対性は、最小化と最大化の順序を入れ替えても値が変わらないことを主張する。両方の最適解が達成されるとき、この順序交換可能性はラグランジアンに鞍点が存在することと同値である。

命題（強双対性と鞍点）。 強双対性が成り立ち、主最適解 $\mathbf{x}^\star$ と双対最適解 $(\lambda^\star, \nu^\star)$ の両方が達成されるとする。このとき $(\mathbf{x}^\star, \lambda^\star, \nu^\star)$ は、ラグランジアンの 鞍点（saddle point）である。すなわち、すべての $\mathbf{x} \in \mathcal{D}$、すべての $\lambda \succeq 0$、およびすべての ν に対して、次式が成り立つ。

$$\mathcal{L}(\mathbf{x}^\star, \lambda, \nu) \;\le\; \mathcal{L}(\mathbf{x}^\star, \lambda^\star, \nu^\star) \;\le\; \mathcal{L}(\mathbf{x}, \lambda^\star, \nu^\star). \tag{35.82}$$

逆に、$\mathcal{L}$ の $\mathbf{x} \in \mathcal{D}$ および $(\lambda \succeq 0, \nu)$ に関する任意の鞍点は双対ギャップをゼロにし、その鞍点値は $p^\star = d^\star$ となる。

証明。 q の定義と弱双対性より、次式が成り立つ。

$$d^\star = q(\lambda^\star, \nu^\star) = \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda^\star, \nu^\star) \;\le\; \mathcal{L}(\mathbf{x}^\star, \lambda^\star, \nu^\star) \;\le\; f(\mathbf{x}^\star) = p^\star,$$

ここで最後の不等号は、$\lambda_i^\star \ge 0$ かつ $g_i(\mathbf{x}^\star) \le 0$ によりすべてのペナルティ項が非正となることから従う。強双対性 $(d^\star = p^\star)$ により、この不等式の連鎖はすべて等号で成り立たなければならない。左側の等号は、$\mathbf{x}^\star$ が $\mathcal{L}(\cdot, \lambda^\star, \nu^\star)$ を最小化することを意味し、Equation 35.82 の右側の不等式を与える。右側の等号からは $\sum_i \lambda_i^\star g_i(\mathbf{x}^\star) = 0$ が従い、これは相補性条件にほかならない。ゆえに、任意の $\lambda \succeq 0$ と ν に対して $\mathcal{L}(\mathbf{x}^\star, \lambda, \nu) = f(\mathbf{x}^\star) + \sum_i \lambda_i g_i(\mathbf{x}^\star) \le f(\mathbf{x}^\star) = \mathcal{L}(\mathbf{x}^\star, \lambda^\star, \nu^\star)$ が成り立ち、Equation 35.82 の左側の不等式が得られる。

逆に、$(\mathbf{x}^\star, \lambda^\star, \nu^\star)$ が鞍点であれば、次式が成り立つ。

$$\begin{aligned} \inf_{\mathbf{x}} \sup_{\lambda \succeq 0, \nu} \mathcal{L} &\le \sup_{\lambda \succeq 0, \nu} \mathcal{L}(\mathbf{x}^\star, \lambda, \nu) = \mathcal{L}(\mathbf{x}^\star, \lambda^\star, \nu^\star) \\ &= \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda^\star, \nu^\star) \le \sup_{\lambda \succeq 0, \nu} \inf_{\mathbf{x}} \mathcal{L}, \end{aligned}$$

これより $p^\star \le d^\star$ がわかる。逆向きの不等式は弱双対性から得られるため、$p^\star = d^\star$ が成り立つ。 ■

クリッピングの例では、鞍点は $(\mathbf{x}^\star, \lambda^\star) = ((0.6, 0.8), 4)$ である。$\mathbf{x}$ を $\mathbf{x}^\star$ からずらすと $\mathcal{L}(\cdot, 4)$ は増加し、一方で λ を動かしても $\mathcal{L}(\mathbf{x}^\star, \lambda) = 8$ は $g(\mathbf{x}^\star) = 0$ のため一定のまま保たれる。

強双対性は実用上 3 つの利点をもたらす。

- **双対問題は主問題より単純または小規模になりうる。**サポートベクターマシンの双対問題はデータ点ごとに 1 つの変数を持ち、単純な符号制約と高々 1 つの等式制約しか持たない。また、演習問題 8 のリッジ回帰の双対問題は、特徴量の次元ではなくデータ例の数に等しいサイズの連立一次方程式（linear system）に帰着する。
- **任意の双対実行可能点が停止基準を与える。**任意の主実行可能な $\mathbf{x}$ と双対実行可能な (λ, ν) に対し、差 $f(\mathbf{x}) - q(\lambda, \nu)$ は $f(\mathbf{x})$ が $p^\star$ からどれだけ離れうるかの上界を与える。
- **双対解から主問題の解を復元できる。**@eq-mdl-opt-saddle-point より、$\mathbf{x}^\star$ は $\mathcal{L}(\cdot, \lambda^\star, \nu^\star)$ の最小点集合の**中に**含まれる。その最小点が一意であるとき（たとえば f が狭義凸のとき）、直ちに $\mathbf{x}^\star$ が得られる（クリッピング問題で $\mathbf{x}(4) = \mathbf{x}_0/5 = \mathbf{x}^\star$ となったのと同様である）。最小点が一意でない場合は、実行可能性と相補性条件によって最小点集合の中から $\mathbf{x}^\star$ が特定される。線形計画問題では $\mathcal{L}(\cdot, \lambda^\star)$ は $\mathbf{x}$ に関して平坦になり、後述するオフセットにペナルティを課さない SVM では、ラグランジアンの最小化だけではオフセットが定まらず、相補性条件がそれを決定する。

鞍点の命題はまた、35.6.2 節で凸問題に対する十分条件として示した KKT 条件の必要性にも決着をつける。微分可能な f、g_i、h_j を $\mathcal{D} = \mathbb{R}^n$ 上に持つ凸問題がスレーター条件を満たし、最適解を $\mathbf{x}^\star$ で達成するとしよう。このとき強双対性が成り立ち、達成された双対最適解 $(\lambda^\star, \nu^\star)$ が存在する。$\mathbf{x}^\star$ は $\mathcal{L}(\cdot, \lambda^\star, \nu^\star)$ を $\mathbb{R}^n$ 全体にわたって最小化するため、その点において勾配 $\nabla_{\mathbf{x}} \mathcal{L}$ はゼロになり、さらに今見たとおり相補性条件も成り立つ。主実行可能性および双対実行可能性と合わせれば、これらはまさに 4 つの KKT 条件そのものである。すなわちスレーター条件のもとで、KKT 条件は必要かつ十分である。

最後に、この min–max 構造は明示的な制約を持たない深層学習の問題にも現れる。敵対的生成ネットワーク（16.1 節）や敵対的訓練（adversarial training）は、直接 $\min_\theta \max_\phi$ ゲームとして定式化される (Goodfellow et al. 2014; Madry et al. 2018)。鞍点が存在しなければ、min max 値と max min 値の間にギャップが生じうる。さらに鞍点が存在する場合であっても、最小化変数に対する勾配降下ステップと最大化変数に対する勾配上昇ステップを同時に行うと収束しないことがある。実際、原点を鞍点とする単純な双線形目的関数 xy に対しては、同時降下上昇法は任意の正のステップサイズで外側へ渦巻状に発散する（演習問題 5）。

35.7.3 価格としての乗数

制約にはどれほどの価値があるだろうか。クリッピング問題において、単位円板をわずかに広げたとき、$\mathbf{x}_0$ にどれだけ近づけるだろうか。最適なラグランジュ乗数は、この問いに直接答えてくれる。さらに都合のよいことに、同じ感度解析の議論から、損失が凸である場合には重み減衰と重みに対するノルム制約がまったく同じ解の族を描くことも明らかになる。

35.7.3.1 シャドープライス

各制約の価値を測るため、右辺に摂動を加えて最適目的関数値がどのように変化するかを追跡しよう。摂動ベクトル $\mathbf{u} \in \mathbb{R}^m$ と $\mathbf{v} \in \mathbb{R}^p$ に対し、摂動後の最適値を次式で定義する。

$$p^\star(\mathbf{u}, \mathbf{v}) = \inf \{ f(\mathbf{x}) : g_i(\mathbf{x}) \le u_i,\ h_j(\mathbf{x}) = v_j \}.$$

$(\mathbf{0}, \mathbf{0})$ では元の最適値 $p^\star(\mathbf{0}, \mathbf{0})$ が復元され、一方で $u_i > 0$ とすることは第 i 不等式制約を緩和することに相当する。実はこの関数はすでに幾何学的に目にしている。単一の不等式制約の場合、

$p^\star(u)$ は高さ t のうち Figure 35.26 の集合 $\mathcal{A}$ が横座標 u で到達する最小値にほかならない。$\mathcal{A}$ の下側境界が $p^\star(u)$ のグラフであり、傾き $-\lambda^\star$ の支持直線は至るところでその下側にある。次の命題は、この幾何学的不等式を任意の個数の制約へと拡張するものである。

命題（乗数はシャドープライスである）。 強双対性が成り立ち、双対最適解 $(\lambda^\star, \nu^\star)$ が存在するとする。このとき、すべての $\mathbf{u}$ と $\mathbf{v}$ に対して次式が成り立つ。

$$p^\star(\mathbf{u},\mathbf{v}) \;\geq\; p^\star(\mathbf{0},\mathbf{0}) - {\lambda^\star}^{\top}\mathbf{u} - {\nu^\star}^{\top}\mathbf{v}, \tag{35.83}$$

さらに $p^\star$ が $(\mathbf{0},\mathbf{0})$ で微分可能であれば、その点において $\nabla_{\mathbf{u}} p^\star = -\lambda^\star$ および $\nabla_{\mathbf{v}} p^\star = -\nu^\star$ が成り立つ。

証明。 $\mathbf{x}$ を摂動後の問題に対する任意の実行可能点とすると、$g_i(\mathbf{x}) \leq u_i$ および $h_j(\mathbf{x}) = v_j$ が成り立つ。強双対性、下限としての q の定義、$\lambda_i^\star \geq 0$、および摂動後の制約を用いると、次式が得られる。

$$\begin{aligned} p^\star(\mathbf{0},\mathbf{0}) = q(\lambda^\star,\nu^\star) &\leq f(\mathbf{x}) + \sum_i \lambda_i^\star g_i(\mathbf{x}) + \sum_j \nu_j^\star h_j(\mathbf{x}) \\ &\leq f(\mathbf{x}) + {\lambda^\star}^{\top}\mathbf{u} + {\nu^\star}^{\top}\mathbf{v}. \end{aligned}$$

摂動後の問題で実行可能なすべての $\mathbf{x}$ にわたる下限をとれば、Equation 35.83 が得られる。さらに $p^\star$ が原点で微分可能であれば、非負関数 $p^\star(\mathbf{u},\mathbf{v}) - p^\star(\mathbf{0},\mathbf{0}) + {\lambda^\star}^{\top}\mathbf{u} + {\nu^\star}^{\top}\mathbf{v}$ は大域的最小値 0 を $(\mathbf{0},\mathbf{0})$ で達成するため、その点において勾配はゼロでなければならない。$\blacksquare$

最適乗数 $\lambda_i^\star$ は制約 i の**シャドープライス**と呼ばれ、その制約を緩和したときに最適値が改善する変化率を測る。なお、Equation 35.83 は大域的な不等式であり、**いかなる**大きさの摂動に対しても成り立つ。制約 i を $u_i > 0$ だけ緩和しても最適値は高々 $\lambda_i^\star u_i$ しか下がらない一方、同じ量だけ厳しくすると最適値は少なくとも $\lambda_i^\star u_i$ だけ上がる。一次近似では、$p^\star$ が微分可能なとき変化量は $-\lambda_i^\star u_i$ となる。余裕のある制約では相補性条件により $\lambda_i^\star = 0$ となるが、これも直感と一致する。有効になっていない制約を緩めても最適値は改善しないからである。

クリッピング問題でこれを確認してみよう。$g(\mathbf{x}) \leq 0$ を $g(\mathbf{x}) \leq u$ に緩和すると円板は半径 $r = \sqrt{1+2u}$ へ拡大し、$p^\star(u) = \tfrac{1}{2}\bigl(5 - \sqrt{1+2u}\bigr)^2$（先ほど求めた $\mathcal{A}$ の下側境界）が得られる。その $u = 0$ における導関数は確かに $-4 = -\lambda^\star$ となっている。制約を $u = \tfrac{1}{2}$ だけ緩和すると（半径 $\sqrt{2}$ の円板）、新しい最適値は $p^\star(\tfrac{1}{2}) = \tfrac{1}{2}(5-\sqrt{2})^2 \approx 6.43$ となり、Equation 35.83 が保証する大域的下界 $8 - 4\cdot\tfrac{1}{2} = 6$ をしっかり上回っている。

1 つ留意しておきたい繊細な点がある。乗数は制約のスケーリングの仕方に依存するということである。ここでの $\lambda^\star = 4$ が価格付けしているのは u、すなわち二乗半径の**半分**であり、半径 r そのものではない。半径 1 単位あたりの変化率を連鎖律で求めると $\mathrm{d}p^\star/\mathrm{d}r = -(5-r)$ となり、これは r に u に関する導関数を掛けたものに等しく、$-\lambda^\star$ と一致するのはたまたま $r = 1$ だからにすぎない。制約を定数倍すれば乗数もそれに応じてスケーリングされる。たとえば $\|\mathbf{x}\|^2 - 1 \leq 0$ のように係数 $\tfrac{1}{2}$ を付けずに書けば、乗数は半分の 2 になる。

35.6.1.3 節の最近接点問題は等式制約の場合の例となっている。そこでは、$\tfrac{1}{2}\|\mathbf{x}\|^2$ の超平面 $\mathbf{a}^{\top}\mathbf{x} = b$ 上での最小値は $p^\star(b) = b^2/(2\|\mathbf{a}\|^2)$ であり、$\partial p^\star/\partial b = -\nu^\star$ は Equation 35.83 を摂動 $h(\mathbf{x}) = \mathbf{a}^{\top}\mathbf{x} - b = v$（超平面を $b + v$ へ平行移動させるもの）に適用した結果そのものである。これにより、同節で数値的に確認した $0.02 = 0.2 \cdot 0.1$ という増加量が説明される。

35.7.3.2 重み減衰とノルム制約

深層学習では、モデルの複雑さを制御するために、重み減衰（2.7 節）を訓練損失 $\ell(\mathbf{w})$ に加えるか、あるいは重みにハードなノルム制約を課すことがよく行われる。本書の記法に従えば、これら 2 つの問題は次のように書ける。

$$\min_{\mathbf{w}} \ \ell(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2 \qquad \text{and} \qquad \min_{\mathbf{w}} \ \ell(\mathbf{w}) \quad \text{subject to} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 \le \tfrac{1}{2}r^2.$$

ペナルティ付き問題の停留点（stationary point）では $\nabla \ell(\mathbf{w}) + \lambda \mathbf{w} = \mathbf{0}$ が成り立つ。一方、制約付き問題では、乗数 $\lambda \ge 0$ を $g(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 - \frac{1}{2}r^2$ に対して導入すると、KKT 停留条件 $\nabla \ell(\mathbf{w}) + \lambda \mathbf{w} = \mathbf{0}$ が得られる。重み減衰係数がラグランジュ乗数の役割を果たす形で、まったく同じ方程式が現れるのである。

損失 ℓ が**凸**であるとき、この同値性は双方向に成り立つ。まずペナルティから制約へ向かう方向として、$\mathbf{w}_\lambda$ をペナルティ付き目的関数の最小点とし、ノルムバジェットを $r = \|\mathbf{w}_\lambda\|$ と選ぶ。すると $\mathbf{w}_\lambda$ は実行可能であり、制約は有効になっており（相補性条件を満たす）、KKT 停留条件も自動的に成り立つ。凸問題では KKT 条件が十分条件であるため（35.6.2 節）、$\mathbf{w}_\lambda$ は制約付き問題の解となる。

次に制約からペナルティへ向かう方向として、任意のバジェット $r > 0$ をとる。原点 $\mathbf{w} = \mathbf{0}$ が狭義実行可能点であるため、スレーター条件が成り立つ。制約付き最小値が $\mathbf{w}^\star$ で達成されるなら、KKT 必要条件から乗数 $\lambda^\star(r) \ge 0$ が存在し、$\mathbf{w}^\star$ は凸のペナルティ付き目的関数 $\ell(\mathbf{w}) + \frac{\lambda^\star}{2}\|\mathbf{w}\|^2$ を最小化する。$\lambda^\star > 0$ である限り、二次項によってこの目的関数は狭義凸となるため、$\mathbf{w}^\star$ がその一意な最小点となる。一方、バジェットに余裕があるときは $\lambda^\star = 0$ となり、ペナルティ付き問題は正則化なしの最小化に帰着する。シャドープライスの命題より $\lambda^\star(r) = -\partial p^\star / \partial(\frac{1}{2}r^2)$ が成り立つ。すなわち重み減衰係数とは、二乗ノルムの半分 1 単位あたりの訓練損失の限界削減量にほかならない。

実際、どちらの方向の議論も ℓ_2 ノルムに固有のものではない。$\frac{1}{2}\|\mathbf{w}\|^2$ を任意の凸ペナルティ $\Omega(\mathbf{w})$、たとえば Lasso（35.6.4.1 節）で用いられる ℓ_1 ノルム $\|\mathbf{w}\|_1$ に置き換えてみよう。任意の最小点 $\mathbf{w}_\lambda$（$\ell + \lambda\Omega$ の最小点）は、バジェット $\Omega(\mathbf{w}) \le \Omega(\mathbf{w}_\lambda)$ の制約付き問題を解く。実際、そのバジェット内のすべての $\mathbf{w}$ に対して $\ell(\mathbf{w}) \ge \ell(\mathbf{w}_\lambda) + \lambda\,(\Omega(\mathbf{w}_\lambda) - \Omega(\mathbf{w})) \ge \ell(\mathbf{w}_\lambda)$ が成り立つからである。逆に、凸な ℓ と $\inf \Omega$ を真に上回る任意のバジェットに対してはスレーター条件が成り立ち、鞍点の命題によって制約付き問題の任意の最小点が $\ell + \lambda^\star\Omega$ も最小化することが保証される。

ただし、完全な同値性が成り立つためには ℓ の凸性が不可欠である。非凸損失の場合、ペナルティ付き問題の停留点は依然として対応する制約付き問題の KKT 点になるものの、大域的最適解の集合は食い違うことがあり、**いかなる**ペナルティ重み λ を用いても到達できないノルムバジェットが生じうる（演習問題 6）。

同じ二次項に対する第 3 の解釈にも触れておこう。ノイズ分散（noise variance）σ^2 を持つ二乗損失（squared loss）と、重みに対する等方的なガウス事前分布（Gaussian prior）$\mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$ を仮定すると、ペナルティ $\frac{\lambda}{2}\|\mathbf{w}\|^2$（ただし $\lambda = \sigma^2/\tau^2$）は定数項と全体のスケールを除いて負の対数事前分布と一致する（36.6 節）。このように、単一の数値 λ は、重み減衰係数、ノルムバジェットに対するラグランジュ乗数、およびノイズ分散と事前分布の分散の比という 3 通りの読み方ができる。

この同値性を数値的に検証してみよう。ランダムなデータ上の凸最小二乗損失 $\ell(\mathbf{w}) = \frac{1}{2}\|\mathbf{A}\mathbf{w} - \mathbf{b}\|^2$ に対し、まず $(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})\mathbf{w} = \mathbf{A}^\top\mathbf{b}$ によってペナルティ付き問題を閉形式で解き、$r = \|\mathbf{w}_\lambda\|$ とおく。次に、ステップサイズ $1/\lambda_{\max}(\mathbf{A}^\top\mathbf{A})$ の射影勾配降下法（projected gradient descent）を用い、クリッピングによって半径 r の球へ射影しながら制約付き問題を解く。制約付きソルバ

は同じ重みを復元できるだろうか。また、停留条件から計算した乗数（$\nabla \ell(\mathbf{w}) + \lambda \mathbf{w} = \mathbf{0}$ と $\mathbf{w}$ の内積をとって得られる $\lambda = -\mathbf{w}^\top \nabla \ell(\mathbf{w}) / r^2$）は λ と一致するだろうか。

```
rng = np.random.default_rng()
A, b = rng.normal(size=(30, 5)), rng.normal(size=30)
grad = lambda w: A.T @ (A @ w - b)                    # gradient of 1/2 |Aw - b|^2
step = 1.0 / np.linalg.eigvalsh(A.T @ A).max()        # 1 / lambda_max(A^T A)
print(' lambda   r = |w_pen|   |w_con - w_pen|   recovered multiplier')
for lam in [0.1, 1.0, 10.0]:
    w_pen = np.linalg.solve(A.T @ A + lam * np.eye(5), A.T @ b)  # penalty form
    r = np.linalg.norm(w_pen)                         # its norm becomes the budget
    w = np.zeros(5)                                   # constraint form: PGD
    for _ in range(5000):
        w -= step * grad(w)
        w *= min(1.0, r / np.linalg.norm(w))          # project onto the r-ball
    lam_rec = -(w @ grad(w)) / r ** 2                 # KKT: grad l + lam w = 0
    print(f'{lam:7.2f}  {r:10.4f}  {np.linalg.norm(w - w_pen):15.2e}  {lam_rec:20.6f}')
```

```
 lambda   r = |w_pen|   |w_con - w_pen|   recovered multiplier
   0.10      0.4138         1.28e-16             0.100000
   1.00      0.3860         1.58e-16             1.000000
  10.00      0.2501         5.72e-17            10.000000
```

どの行においても、制約付き問題の解はペナルティ付き問題の解と機械精度で一致し、停留条件から復元した乗数も表示された小数点以下 6 桁すべてで λ と一致している（具体的なバジェット r の値は乱数の抽出ごとに変わる）。ペナルティと制約は同じ解の経路に対する 2 通りのパラメータ化にすぎず、ラグランジュ乗数が両者の間を変換しているのである。

35.7.4 凸共役

双対関数 q を評価するには、通常、乗数の組を選ぶたびに新たな制約なし最小化問題を解き直す必要がある。しかし制約が線形である場合には、共通する計算をくくり出して、その最小化を最初に 1 度だけ済ませておくことができる。制約 $\mathbf{A}\mathbf{x} \preceq \mathbf{b}$ に対してラグランジアンは $f(\mathbf{x}) + \lambda^\top (\mathbf{A}\mathbf{x} - \mathbf{b})$ であるから、次式が成り立つ。

$$q(\lambda) = \inf_{\mathbf{x}} \left[f(\mathbf{x}) + (\mathbf{A}^\top \lambda)^\top \mathbf{x} \right] - \mathbf{b}^\top \lambda .$$

この下限は f と単一のベクトル $\mathbf{A}^\top \lambda$ のみに依存し、$\mathbf{A}$ や $\mathbf{b}$ に個別には依存しないことに注目しよう。よく使われる各目的関数 f に対して $\inf_{\mathbf{x}} [f(\mathbf{x}) - \mathbf{y}^\top \mathbf{x}]$ の値を $\mathbf{y}$ の関数として 1 度だけ計算しておけば、目的関数 f を持つ**いかなる**線形制約付き問題の双対問題も、対応表からそのまま読み取れるようになる。その対応表こそが凸共役である。

35.7.4.1 支持直線

まず 1 次元で $f(x) = \frac{1}{2} x^2$ を考え、傾き $y = 2$ をとろう。直線 $t = 2x + c$（傾き 2）のうち、至るところで放物線以下にあるものはどれだろうか。$2x + c \le \frac{1}{2} x^2$ がすべての x で成り立つ必要があり、これは $c \le \min_x (\frac{1}{2} x^2 - 2x) = -2$（$x = 2$ で達成）を意味する。そのような直線のうち最も高い $t = 2x - 2$ は点 $(2, 2)$ で放物線に接し、そこでは $f'(2) = 2$ である。傾き 2 の直線をグラフに接するまで下げていくと縦軸切片は -2 に落ち着き、数値 $\max_x (2x - \frac{1}{2} x^2) = 2$ はその切片の符号を反転させたものになる。この様子を Figure 35.27(a) に示す。

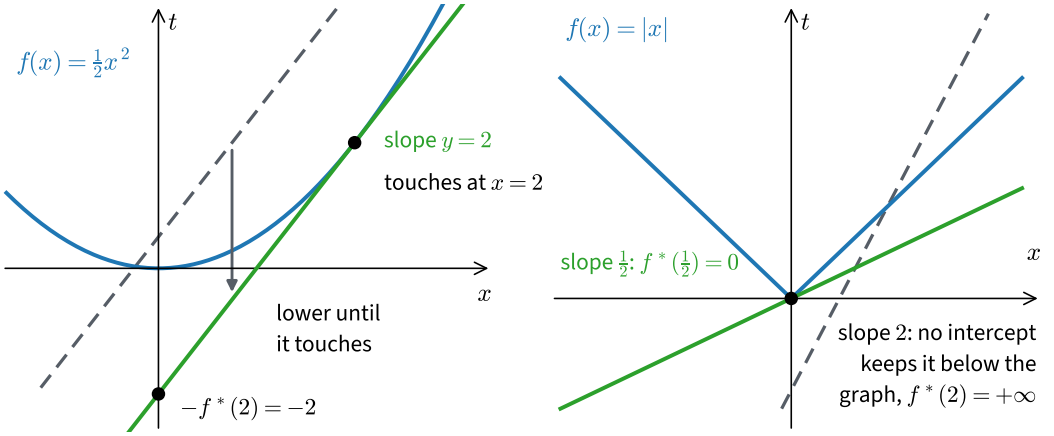


Figure 35.27: 支持直線としての凸共役。(a) $f(x) = \frac{1}{2}x^2$ と傾き 2 の場合、傾き 2 の破線をグラフに接するまで下げると $x = 2$ で接して切片は -2 となり、$f^*(2) = 2$ を与える。(b) $f(x) = |x|$ の場合、傾き $\frac{1}{2}$ の直線は原点でグラフを下から支えて切片 0 となる一方、傾き 2 の直線はどのような切片をとっても $|x|$ より速く上昇するため、$f^*(\frac{1}{2}) = 0$ および $f^*(2) = +\infty$ となる。このように、凸共役が有限になるのは、グラフの下側に支持直線が存在する傾きに限られる。

関数 $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ の**凸共役**（フェンシェル・ルジャンドル変換とも呼ばれる）は、すべての傾きベクトル $\mathbf{y}$ に対してこの符号反転した切片を記録したものである（本書では共役にアスタリスク $*$ を、最適解・最適値に星印 $\star$ を用いる）。

$$f^*(\mathbf{y}) \;=\; \sup_{\mathbf{x}} \; \left(\mathbf{y}^\top \mathbf{x} - f(\mathbf{x})\right). \tag{35.84}$$

幾何学的には、$f^*(\mathbf{y})$ は傾き $\mathbf{y}$ を持ち至るところで f のグラフ以下にあるアフィン関数のうち最も高いものの縦軸切片にマイナスを付けた値である。凸共役は、f を各点での値によってではなく、f の支持直線の族によって記述する。$f(x) = \frac{1}{2}x^2$ と任意の傾き y に対して上限は $x = y$ で達成され、$f^*(y) = \frac{1}{2}y^2$ となる。すなわち、この二次関数は自分自身の共役（自己共役）である。

f が値 $+\infty$ をとることを許すのは、制限された定義域を表現する便利な方法である。$f(\mathbf{x}) = +\infty$ となる点は上限に一切寄与せず、このような拡大実数値関数の凸性は単にそのエピグラフの凸性を意味する。

上限そのものも $+\infty$ になりうる。$f(x) = |x|$ と傾き $y = 2$ を考えよう。$x > 0$ では $2x - |x| = x$ が際限なく大きくなる。つまり、切片をどれほど低くしても、傾き 2 の直線は V 字型のグラフの下側に留まることができない。一方、$|y| \le 1$ のときは常に $yx - |x| \le 0$ が成り立ち、$x = 0$ で等号となるため、原点を通る傾き y の直線がグラフを下から支える。したがって、$f^*(y) = 0$（$|y| \le 1$ のとき）であり、それ以外では $f^*(y) = +\infty$ となる（Figure 35.27(b)）。言い換えれば、絶対値関数の共役は区間 $[-1, 1]$ の**指示関数**（indicator function）、すなわち集合の内部で 0、外部で $+\infty$ をとる関数である。

Equation 35.84 から直ちに 2 つの性質が従う。第 1 に、f^* は f が凸であるかどうかにかかわらず**常に**凸関数である。固定された各 $\mathbf{x}$ に対して式 $\mathbf{y}^\top \mathbf{x} - f(\mathbf{x})$ は $\mathbf{y}$ に関してアフィンであり、アフィン関数族の各点ごとの上限はそのエピグラフがそれらの上側半空間の共通部分となるため凸関数になる（これは q が凹関数になることを示した議論のちょうど鏡像である）。

第 2 に、$f^*(\mathbf{y})$ はすべての $\mathbf{x}$ にわたる上限であるから、$f^*(\mathbf{y}) \geq \mathbf{y}^\top \mathbf{x} - f(\mathbf{x})$ がすべての $\mathbf{x}$ と $\mathbf{y}$ に対して成り立つ。項を整理すると**フェンシェル・ヤングの不等式**が得られる。

$$\mathbf{y}^\top \mathbf{x} \;\leq\; f(\mathbf{x}) + f^*(\mathbf{y}) \qquad \text{for all } \mathbf{x}, \mathbf{y}. \tag{35.85}$$

等号が成り立つのは、$\mathbf{x}$ が Equation 35.84 の上限を達成するとき、かつそのときに限る。f が凸かつ微分可能であれば、これは $\mathbf{y} = \nabla f(\mathbf{x})$ のとき、かつそのときに限って生じる。というのも、このとき $\mathbf{x} \mapsto \mathbf{y}^\top \mathbf{x} - f(\mathbf{x})$ は凹関数となり、勾配がゼロになる点で最大化されるからである。ここでは凸性が重要である。非凸の二重井戸関数（double well）$\phi_0(x) = (x^2 - 1)^2$ では $\phi_0'(0) = 0$ であるにもかかわらず、$\phi_0^*(0) = -\min \phi_0 = 0$ かつ $\phi_0(0) + \phi_0^*(0) = 1 > 0$ となる。これは $x = 0$ がグラフを下から水平な直線で支える点ではなく極大点だからである。自己共役対 $\frac{1}{2}\|\mathbf{x}\|^2$ に対しては、フェンシェル・ヤングの不等式はおなじみの不等式 $\mathbf{y}^\top \mathbf{x} \leq \frac{1}{2}\|\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{y}\|^2$ に帰着する。

共役へ移ることで何か情報が失われるだろうか。任意の凸関数 f が**固有**（proper）（決して $-\infty$ をとらず、恒等的に $+\infty$ でもない）かつ**閉**（closed）（エピグラフが閉集合である）ならば、支持直線族はグラフを完全に決定し、共役をもう 1 度とると元の f が復元される。すなわち $f^{**} = f$ が成り立つ (Rockafellar 1970)。一方、f が非凸である場合（かつ少なくとも 1 つのアフィン関数で下から抑えられる場合）、f^{**} は下側から抑える最も大きい閉凸関数、すなわち f の凸包絡関数（convex envelope）となる（演習問題 1d）。

本節および演習問題で用いる 4 つの共役対を以下にまとめる。

- $\frac{1}{2}\|\mathbf{x}\|^2$ と $\frac{1}{2}\|\mathbf{y}\|^2$（自己共役対）。
- $\frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x}$ と $\frac{1}{2}\mathbf{y}^\top \mathbf{A}^{-1}\mathbf{y}$（ただし $\mathbf{A} \succ 0$ であり、上限は $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ で達成される）。
- e^x と、$y \log y - y$（$y > 0$ のとき。さらに値 0 を $y = 0$ で、$+\infty$ を $y < 0$ でとる。上限は $x = \log y$ で達成され、たとえば $f^*(1) = -1$、$f^*(e) = 0$ となる）。
- $\lambda\|\mathbf{x}\|_1$ とボックス $\{\mathbf{y} : \|\mathbf{y}\|_\infty \leq \lambda\}$ の指示関数（上の絶対値関数のベクトル版）。

35.7.4.2 線形制約付き問題の双対

ここで、凸共役を導入する動機となった線形制約付き問題、すなわち $f(\mathbf{x})$ を制約 $\mathbf{A}\mathbf{x} \preceq \mathbf{b}$ のもとで最小化する問題に戻ろう。下限からマイナス符号をくくり出すと、そのまま f^* を定義する上限へと変形できる。

$$\begin{aligned} q(\lambda) &= \inf_{\mathbf{x}} \left[f(\mathbf{x}) + (\mathbf{A}^\top \lambda)^\top \mathbf{x} \right] - \mathbf{b}^\top \lambda \\ &= -\sup_{\mathbf{x}} \left[(-\mathbf{A}^\top \lambda)^\top \mathbf{x} - f(\mathbf{x}) \right] - \mathbf{b}^\top \lambda \\ &= -f^*(-\mathbf{A}^\top \lambda) - \mathbf{b}^\top \lambda. \end{aligned} \tag{35.86}$$

等式制約 $\mathbf{A}\mathbf{x} = \mathbf{b}$ の場合も、まったく同じ 3 行の変形で $q(\nu) = -f^*(-\mathbf{A}^\top \nu) - \mathbf{b}^\top \nu$（ただし ν の符号は制約なし）が得られる。なお、これは q が凹関数である理由に対する第 2 の説明にもなっている。f^* は凸関数でその引数は λ に関して線形であり、マイナス符号が凸性を凹性へと反転させるからである。

Equation 35.86 を考えられる最小の例で試してみよう。半平面内で原点に最も近い点を求める問題、すなわち $\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}\|^2$（制約は $\mathbf{a}^\top \mathbf{x} \leq b$、ただし $\mathbf{a} = (1, 2)$、$b = -1$）である。原点は制約に違反するため（$\mathbf{a}^\top \mathbf{0} = 0 > -1$）、最適解において制約は有効になっていなければならない。$f^*(\mathbf{y}) = \frac{1}{2}\|\mathbf{y}\|^2$ と $\|\mathbf{a}\|^2 = 5$ を Equation 35.86 に代入すると、1 ステップで双対関数が得られる。

$$q(\lambda) = -\tfrac{1}{2}\lambda^2\|\mathbf{a}\|^2 - b\lambda = \lambda - \tfrac{5}{2}\lambda^2.$$

$q'(\lambda) = 1 - 5\lambda = 0$ とおくと $\lambda^\star = \tfrac{1}{5}$ および $d^\star = \tfrac{1}{5} - \tfrac{1}{10} = \tfrac{1}{10}$ が得られる。$\lambda^\star$ におけるラグランジアンの最小点は共役の上限を達成する点であり、$\mathbf{x} = -\lambda^\star \mathbf{a} = (-0.2, -0.4)$ となる。直接確かめると $\mathbf{a}^\top \mathbf{x} = -1 = b$ かつ $f(\mathbf{x}) = \tfrac{1}{2}(0.04 + 0.16) = \tfrac{1}{10} = d^\star$ であり、弱双対性によって $\mathbf{x}$ が最適解であることが保証される。これは 35.6.1.3 節の射影公式 $(b/\|\mathbf{a}\|^2)\,\mathbf{a}$ を別の超平面に対して再現したものであり、主問題が 2 変数であったのに対し、双対問題は 1 変数となってわずか 1 行で解けたことになる。

35.7.4.3 log-sum-exp とエントロピー

深層学習のあらゆる場面で、ソフトマックスは生のスコアを指数変換して正規化することで確率へと変換する。なぜ他の正値関数ではなく指数関数なのだろうか。log-sum-exp 関数 $\mathrm{lse}(\mathbf{x}) = \log \sum_i e^{x_i}$ の凸共役がその明快な答えを与えてくれる。確率単体（probability simplex）を $\Delta = \{\mathbf{p} : \mathbf{p} \succeq 0,\ \mathbf{1}^\top \mathbf{p} = 1\}$、シャノンエントロピーを $H(\mathbf{p}) = -\sum_i p_i \log p_i$ と表そう。

$\mathrm{lse}^*(\mathbf{p}) = \sup_{\mathbf{x}} \big(\mathbf{p}^\top \mathbf{x} - \mathrm{lse}(\mathbf{x})\big)$ を計算するため、まず上限がいつ有限になりうるかを調べる。もし負の座標 p_i が 1 つでもあれば、$x_i \to -\infty$ とすることで $\mathbf{p}^\top \mathbf{x}$ は $+\infty$ へ発散する一方、$\mathrm{lse}(\mathbf{x})$ は有界に留まるため、上限は $+\infty$ となる。また $\mathbf{1}^\top \mathbf{p} \neq 1$ であれば、すべての座標を定数だけシフトさせて $\mathbf{x} = c\mathbf{1}$ とおくと $c(\mathbf{1}^\top \mathbf{p} - 1) - \log n$（$\mathrm{lse}(c\mathbf{1}) = c + \log n$ であるため）となり、やはり $c \to \pm\infty$ としたときに発散する。ゆえに $\mathrm{lse}^*(\mathbf{p})$ が有限となるのは確率単体 Δ 上に限られる。

$\mathbf{p}$ が Δ の内部にあるとき（すべての $p_i > 0$）、$\mathbf{x}$ に関する勾配をゼロとおくと $\mathrm{softmax}(\mathbf{x}) = \mathbf{p}$ が得られ、その解は $\mathbf{x} = \log \mathbf{p}$（加法定数の自由度を除いて一意）となって値 $\sum_i p_i \log p_i$ を与える。一方、$\mathbf{p}$ が Δ の境界上にあり一部の $p_i = 0$ となっている場合、ソフトマックスの出力は常に真に正であるため有限な最大点は存在しない。しかし、該当する座標で $x_i \to -\infty$ とすれば同じ値に収束するため、この式は Δ 全体へ連続的に拡張される（ただし標準的な規約 $0 \log 0 = 0$ を用いる）。lse は凸かつ至るところ有限であるため $\mathrm{lse}^{**} = \mathrm{lse}$ が成り立ち、次の共役対が得られる。

$$\begin{aligned} \mathrm{lse}^*(\mathbf{p}) &= \begin{cases} \sum_i p_i \log p_i = -H(\mathbf{p}), & \mathbf{p} \in \Delta, \\ +\infty, & \text{otherwise}, \end{cases} \\ \mathrm{lse}(\mathbf{x}) &= \max_{\mathbf{p} \in \Delta} \big(\mathbf{p}^\top \mathbf{x} + H(\mathbf{p})\big) \quad \text{(dually)}. \end{aligned} \tag{35.87}$$

Equation 35.87 の 2 行目を言葉で読んでみよう。log-sum-exp とは、期待スコア $\mathbf{p}^\top \mathbf{x}$ にエントロピーボーナス $H(\mathbf{p})$ を加えた量を、n 個の結果上のすべての確率分布にわたって最大化したものである。どの分布がこの最大値を達成するだろうか。フェンシェル・ヤングの不等式の等号成立条件より、それは $\mathbf{p} = \nabla \mathrm{lse}(\mathbf{x})$ であり、その第 i 成分は $\partial\,\mathrm{lse}/\partial x_i = e^{x_i} / \sum_j e^{x_j}$、すなわちソフトマックスそのものである（そのヘッセ行列は 35.3 節で計算した）。言い換えれば、同じ期待スコア $\mathbf{p}^\top \mathbf{x}$ を達成するすべての確率分布の中で、ソフトマックスは最大エントロピーを持つ分布であり、36.4.2 節における最大エントロピー法からの導出と一致する（演習問題 2 ではその制約側からソフトマックスを導出する）。エントロピー項がなければ、$\mathbf{p}^\top \mathbf{x}$ を Δ 上で最大化するとすべての確率質量が $\arg\max_i x_i$ に集中するが、エントロピーを加えることでそのハードな選択が滑らかな確率分布へと軟化される。これとまったく同じ共役性の構成を f ダイバージェンスの凸生成関数に適用すると、f-GAN の訓練に用いられる変分下界が得られる（37.3.4 節）。

35.7.5 具体例

ここからは、双対性が実際の問題で何を明らかにするかを浮き彫りにする 3 つの具体例を順に見ていこう。サポートベクターマシンでは、相補性条件によって解がごく少数のデータ点に集中し、さらに双対問題ではデータが内積のみを通じて現れる。注水法では、ラグランジュ乗数が電力 1 単位あたりの通信レートという具体的な工学的単位を持つ。最後に、1 次元の非凸問題を通じて、凸性が防いでいた真に正の双対ギャップを具体的に確認する。

35.7.5.1 サポートベクターマシンの双対問題

線形分離可能な訓練データ $(\mathbf{x}_i, y_i)$（ラベルは $y_i \in \{\pm 1\}$）が与えられたとき、サポートベクターマシン (Boser et al. 1992) は可能な限り最大のマージン（margin）を持つ分離超平面 $\mathbf{w}^\top \mathbf{x} + b = 0$ を求める。その主問題は次式で表される。

$$\min_{\mathbf{w}, b} \ \tfrac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i\,(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \ \ i = 1, \dots, n. \tag{35.88}$$

点 $\mathbf{x}$ から超平面までの距離は $|\mathbf{w}^\top \mathbf{x} + b| / \|\mathbf{w}\|$ であるから（33.1 節）、2 つのマージン超平面 $\mathbf{w}^\top \mathbf{x} + b = \pm 1$ は決定境界（decision boundary）の両側にそれぞれ距離 $1/\|\mathbf{w}\|$ の位置にあり、互いの間隔は $2/\|\mathbf{w}\|$ となる。したがって $\tfrac{1}{2}\|\mathbf{w}\|^2$ を最小化することはマージンを最大化することに相当する。これは訓練点ごとに 1 つのアフィン不等式制約を持つ凸二次計画問題である。制約がアフィンであるため、スレーター条件には実行可能点が 1 つ存在すれば十分であり、データを分離する任意の $(\mathbf{w}, b)$ を定数倍してすべての点が少なくとも 1 のマージンを持つようにスケーリングすればよい。ゆえに強双対性が保証される。

各制約に非負の乗数 $\alpha_i \geq 0$ を 1 つずつ導入し、ラグランジアンを次式で作る。

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \tfrac{1}{2}\|\mathbf{w}\|^2 + \sum_{i} \alpha_i \left(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right).$$

双対関数 $q(\alpha) = \inf_{\mathbf{w}, b} \mathcal{L}$ を計算するため、まず $\mathcal{L}$ がペナルティの課されていないオフセット b に関してアフィンであり、その係数が $-\sum_i \alpha_i y_i$ であることに注目しよう。この係数がゼロでない限り、$b \to \pm\infty$ とすることで $\mathcal{L}$ は $-\infty$ へ発散するため、$q(\alpha) = -\infty$ が $\sum_i \alpha_i y_i \neq 0$ のときは常に成り立つ。ここに Equation 35.80 の実効定義域が実際に機能している姿が見て取れる。有限性の条件 $\sum_i \alpha_i y_i = 0$ が、双対問題における明示的な等式制約になるのである。この定義域上では b の項が消え、$\mathbf{w}$ に関する狭義凸な二次関数が残るため、その一意な最小値は $\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0}$ となる点で達成される。

$$\mathbf{w} = \sum_i \alpha_i y_i \, \mathbf{x}_i, \qquad \text{on the domain} \qquad \sum_i \alpha_i y_i = 0.$$

したがって、最適な重みベクトルは訓練入力をそれぞれの乗数で重み付けした線形結合となる。この式を $\mathcal{L}$ に代入すると、**SVM の双対問題**が得られる。

$$\max_{\alpha \geq 0, \ \sum_i \alpha_i y_i = 0} \ \sum_i \alpha_i \ - \ \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j \, y_i y_j \, \mathbf{x}_i^\top \mathbf{x}_j. \tag{35.89}$$

ソルバを選ぶ前から、2 つの際立った帰結が導かれる。第 1 に、相補性条件 $\alpha_i(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) = 0$ より、$\alpha_i > 0$ となる任意の点は $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ を満たし、いずれかのマージン超平面上に

ぴったり乗っていなければならない。これらの点を**サポートベクター**と呼ぶ。Figure 35.28 では、丸で囲まれた 4 点の中にしかサポートベクターは存在しえない。それらの点では制約が有効になっており、Figure 35.21 と同様にその法線ベクトルが ∇f と釣り合う一方、$\alpha_i = 0$ である点はすべて $\mathbf{w}$ から完全に脱落する。第 2 に、訓練入力は Equation 35.89 の中に内積 $\mathbf{x}_i^\top \mathbf{x}_j$ のみを通じて現れる。これらの内積をカーネル関数（kernel function）$k(\mathbf{x}_i, \mathbf{x}_j)$ に置き換えれば直ちに非線形分類器が得られるが、これは主問題の定式化では隠れていた構造である。

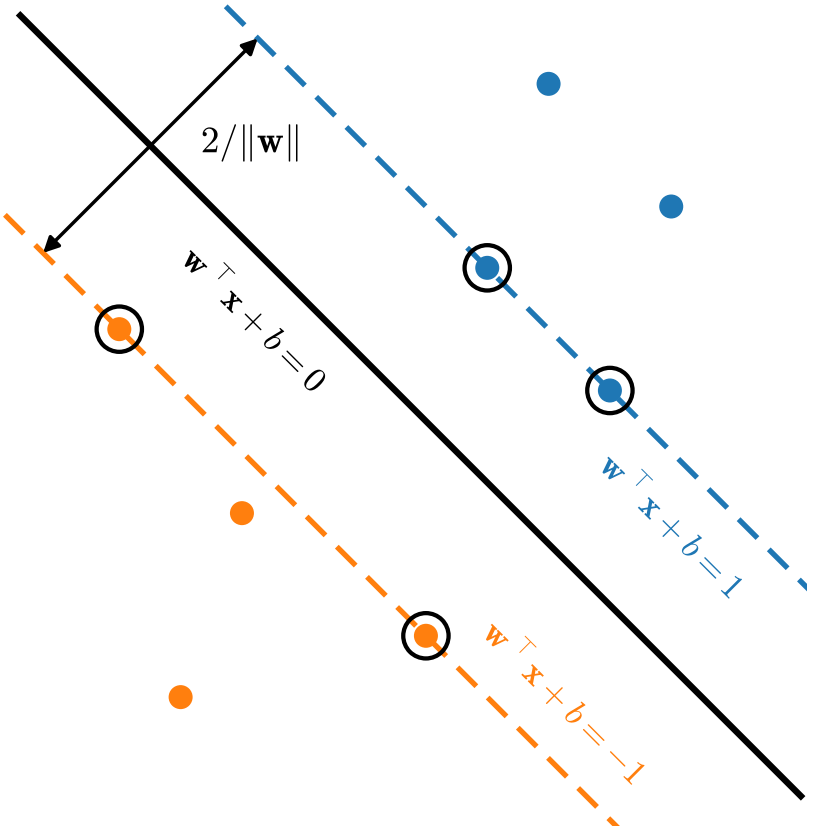


Figure 35.28: 下のコードセルで用いる 8 点の最大マージン分離。実線は決定境界 $\mathbf{w}^{\star\top}\mathbf{x} + b^\star = 0$（$\mathbf{w}^\star = (\frac{4}{7}, \frac{4}{7})$、$b^\star = -\frac{3}{7}$）であり、破線はマージン超平面 $\mathbf{w}^{\star\top}\mathbf{x} + b^\star = \pm 1$（互いの間隔は $2/\|\mathbf{w}^\star\|$）である。マージン超平面上にある丸で囲まれた 4 点のみが $\alpha_i > 0$ を持ちうる。

等式制約 $\sum_i \alpha_i y_i = 0$ が生じたのは、ひとえにペナルティなしのオフセット b によってラグランジアンがある方向にアフィンとなったためである。双対問題の実行可能領域（feasible region）を非負象限（負の成分をゼロにクリップするだけで射影できる単純な集合）に保つため、コードではオフセットを重みに折り込んだ密接に関連する変種を解くことにしよう。すなわち、各入力に定数特徴量を加えて $\tilde{\mathbf{x}}_i = (\mathbf{x}_i, 1)$ とし、$\tilde{\mathbf{w}} = (\mathbf{w}, b)$ とおいて、$\frac{1}{2}\|\tilde{\mathbf{w}}\|^2$ を制約 $y_i \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_i \geq 1$ のもとで最小化する。この変種は b にもペナルティを課すため決定境界は Equation 35.88 とわずかに異なりうるが、引き換えにラグランジアンがすべての主変数に関して狭義凸となり、等式制約が不要になる。$\mathbf{Q}_{ij} = y_i y_j \tilde{\mathbf{x}}_i^\top \tilde{\mathbf{x}}_j$ と定義すると、双対問題は次式となる。

$$\max_{\alpha \succeq 0} \quad \mathbf{1}^\top \alpha - \tfrac{1}{2} \alpha^\top \mathbf{Q}\, \alpha, \tag{35.90}$$

これは非負象限上の凹二次関数の最大化問題である。射影勾配上昇法（35.6.3 節の更新則のステップ符号を反転させたもの）を適用すると、$\alpha \leftarrow \max(\mathbf{0},\, \alpha + \eta\,(\mathbf{1} - \mathbf{Q}\alpha))$ を安全なステップサイズ $\eta = 1/\lambda_{\max}(\mathbf{Q})$ で反復することになる。平面上の 8 点に対してこの上昇法を実行し、復元された超平面がデータを分離するか、正の乗数がすべてマージン 1 の点に乗っているか、そして双対値が主問題の値と一致するかという 3 つの予測を検証しよう。

```
X = np.array([[1.0, 1.5], [2.0, 0.5], [2.5, 2.0], [1.5, 3.0],
              [-1.0, -0.5], [-2.0, 1.0], [-1.5, -2.0], [0.5, -1.5]])
y = np.array([1.0, 1.0, 1.0, 1.0, -1.0, -1.0, -1.0, -1.0])
Xa = np.hstack([X, np.ones((len(X), 1))])     # fold the offset b into w
G = y[:, None] * Xa                           # rows y_i x~_i
Q = G @ G.T                                   # Q_ij = y_i y_j x~_i^T x~_j
alpha = np.zeros(len(X))
eta = 1.0 / np.linalg.eigvalsh(Q).max()       # 1 / lambda_max(Q)
for _ in range(5000):                         # projected gradient ascent
    alpha = np.maximum(0.0, alpha + eta * (1.0 - Q @ alpha))
w = G.T @ alpha                               # w~* = sum_i alpha_i y_i x~_i
margins = y * (Xa @ w)
dual, primal = alpha.sum() - 0.5 * alpha @ Q @ alpha, 0.5 * w @ w
print('alpha* =', alpha.round(4))
print('rank of Q =', np.linalg.matrix_rank(Q), '  multipliers > 0:', int((alpha >
 ↪  1e-8).sum()))
print('w* =', w[:2].round(4), ' b* =', w[2].round(4),
      ' separates data:', bool((np.sign(Xa @ w) == y).all()))
print('margins y_i (w^T x_i + b) =', margins.round(4))
print(f'primal = {primal:.6f}, dual = {dual:.6f}, gap = {abs(primal - dual):.1e}, '
      f'comp. slack = {np.abs(alpha * (margins - 1.0)).max():.1e}')
```

```
alpha* = [0.1309 0.0732 0.     0.     0.     0.2442 0.     0.3885]
rank of Q = 3   multipliers > 0: 4
w* = [0.5714 0.5714]  b* = -0.4286  separates data: True
margins y_i (w^T x_i + b) = [1.     1.     2.1429 2.1429 1.2857 1.     2.4286 1.    ]
primal = 0.418367, dual = 0.418367, gap = 4.4e-16, comp. slack = 3.0e-16
```

3 つの予測はすべて成り立っている。4 つの乗数が正であり、まさにその 4 点がマージン 1 に位置している一方、残りの 4 点はマージンが 1.29 から 2.43 の間にあり $\alpha_i = 0$ となっている。出力された数値は厳密解 $\tilde{\mathbf{w}}^\star = (\tfrac{4}{7}, \tfrac{4}{7}, -\tfrac{3}{7})$（すなわち $\mathbf{w}^\star = (\tfrac{4}{7}, \tfrac{4}{7})$、$b^\star = -\tfrac{3}{7}$）と一致し、$p^\star = d^\star = \tfrac{41}{98} \approx 0.418367$、双対ギャップは 10^{-16} 付近となっている（主問題の値と双対値が等しいことは、最適解における恒等式 $\sum_i \alpha_i^\star = \|\tilde{\mathbf{w}}^\star\|^2$ からも直接従う。演習問題 4a を参照）。

この出力結果は、双対解の解釈に関する繊細な点も浮き彫りにしている。主問題の目的関数 $\tfrac{1}{2}\|\tilde{\mathbf{w}}\|^2$ は狭義凸であるため、主最適解 $\tilde{\mathbf{w}}^\star$ は一意である。しかし双対最適解 $\alpha^\star$ は一意では**ない**。ここでは $\mathbf{Q} = \mathbf{G}\mathbf{G}^\top$ であり、$\mathbf{G}$ は 8×3 行列（各行が $y_i \tilde{\mathbf{x}}_i^\top$）であるため、$\mathbf{Q}$ のランクは高々 3 である。4 つのサポートベクターは $\mathbb{R}^3$ 内の 4 本のベクトルに対応するため線形従属であり、それら 4 つの添字上にのみ非ゼロ成分を持つ非ゼロベクトル $\mathbf{d}$ で $\mathbf{G}^\top \mathbf{d} = \mathbf{0}$ を満たすものが存在する。$\alpha^\star$ を $\mathbf{d}$ の方向に動かしても二次項は変化せず（$\mathbf{Q}\mathbf{d} = \mathbf{0}$ のため）、またすべてのサポートベクターがマージン 1 を持つことから $\mathbf{1}^\top \mathbf{d} = \sum_i d_i\, y_i \tilde{\mathbf{w}}^{\star\top} \tilde{\mathbf{x}}_i = \tilde{\mathbf{w}}^{\star\top} \mathbf{G}^\top \mathbf{d} = 0$ となるため一次項も変化しない。したがって乗数ベクトルからなる線分全体が最適解となり、いずれも同じ $\tilde{\mathbf{w}}^\star$ と同じ双対値を与える。$\mathbf{0}$ から出発する射影勾配上昇法は、単にその線分上の 1 点を返したにすぎない。この線分の各端点では、マージン 1 を持つ 4 点のうち 1 点で実際に $\alpha_i = 0$ となる（演習問題 4c）。相補性条件は一方向の含意であることに注意しよう。すなわち、$\alpha_i > 0$ ならばマージンが 1 であることは保証されるが、マージンが 1 だからといって $\alpha_i > 0$ が強制されるわけではない。

35.7.5.2 注水法

2 つ目の具体例は、閉形式解（closed-form solution）と忘れがたい物理的描像の両方を備えている。送信機が総電力バジェット P を n 個の独立な通信チャネルに配分しなければならないとしよう。チャネル i は背景雑音電力 $N_i > 0$ を持ち、そこに電力 $P_i \geq 0$ を割り当てると 1 回の使用あたり $\log(1 + P_i/N_i)$ nat の伝送レートが得られる (Cover and Thomas 1999)（Cover と Thomas は実ガウスチャネルに対して係数 $\frac{1}{2}$ を含めているが、これは以下のラグランジュ乗数を定数倍するだけで最適配分は変えない）。総伝送レートを最大化するため、次の問題を解く。

$$\max_{\mathbf{P}} \; U(\mathbf{P}) = \sum_{i=1}^{n} \log\left(1 + \frac{P_i}{N_i}\right) \quad \text{subject to} \quad P_i \geq 0, \;\; \sum_i P_i = P. \tag{35.91}$$

凹関数 U の最大化は、アフィン制約のもとでの凸関数 $-U$ の最小化と同値である。すべての成分が真に正である実行可能な配分が存在するためスレーター条件が成り立ち、KKT 条件は必要十分条件となる。バジェットの等式制約に用いる乗数 ν（$\sum_i P_i - P = 0$ に対応）と、非負制約に用いる乗数 $\lambda_i \geq 0$（$-P_i \leq 0$ に対応）を導入すると、$-U$ のラグランジアンの停留条件は次式となる。

$$\frac{1}{N_i + P_i} = \nu - \lambda_i.$$

ここから 2 つの場合を読み取ろう。チャネル i に正の電力が割り当てられるとき（$P_i > 0$）、相補性条件により $\lambda_i = 0$ が強制されるため $N_i + P_i = 1/\nu$ となり、電力を受けるすべての稼働チャネルはまったく同じ合計レベル $1/\nu$ に揃う。一方、チャネル i に電力が割り当てられないとき（$P_i = 0$）、双対実行可能性 $\lambda_i = \nu - 1/N_i \geq 0$ から $N_i \geq 1/\nu$ が要求される。この共通のしきい値を $\theta = 1/\nu$ と書けば、最適配分は次の簡潔な形で表される。

$$P_i^{\star} = \max(\theta - N_i,\, 0), \qquad \text{with } \theta \text{ set by } \sum_i \max(\theta - N_i,\, 0) = P.$$

これが古典的な**注水法**の規則である。各チャネルを底の高さが N_i にある器と見立て、器の並び全体へ体積 P の水を注ぎ込む場面を想像してほしい。Figure 35.29 に示すように、水面は一様な水位 θ に落ち着き、最も雑音の小さい静かなチャネルに最も多くの電力が注がれる一方、底の高さが θ を超える器は完全に乾いたまま残る。$\sum_i \max(\theta - N_i, 0)$ は連続な区分線形関数であり、$\theta \leq \min_i N_i$ でゼロ、$\theta > \min_i N_i$ で狭義単調増加であるため、任意のバジェット $P > 0$ に対して一意な水位 θ が定まり、二分法によって容易に求められる（35.6.3 節の単体への射影は、各座標から共通のレベルを引いてゼロでクリップするという、このしきい値構造をちょうど反転させた形になっている）。

シャドープライスの命題から、$\nu^{\star}$ には直ちに工学的な意味が与えられる。U の最大化は $-U$ の最小化であるから、バジェットを P から $P + \upsilon$ へ緩和すると $-U^{\star}$ は変化率 $-\nu^{\star}$ で変化し、したがって $\mathrm{d}U^{\star}/\mathrm{d}P = \nu^{\star} = 1/\theta$ となる。すなわち最適乗数は、追加電力 1 単位あたりに得られる追加の通信レートを表す。5 つのチャネルに対する注水法の配分を計算し、このシャドープライスによる予測を $P \pm 10^{-4}$ における有限差分（finite difference）と比較してみよう。

```
N = np.array([0.1, 0.4, 0.8, 1.6, 2.5])       # channel noise levels
P = 3.0                                       # total power budget

def water_fill(P, N, steps=60):
```

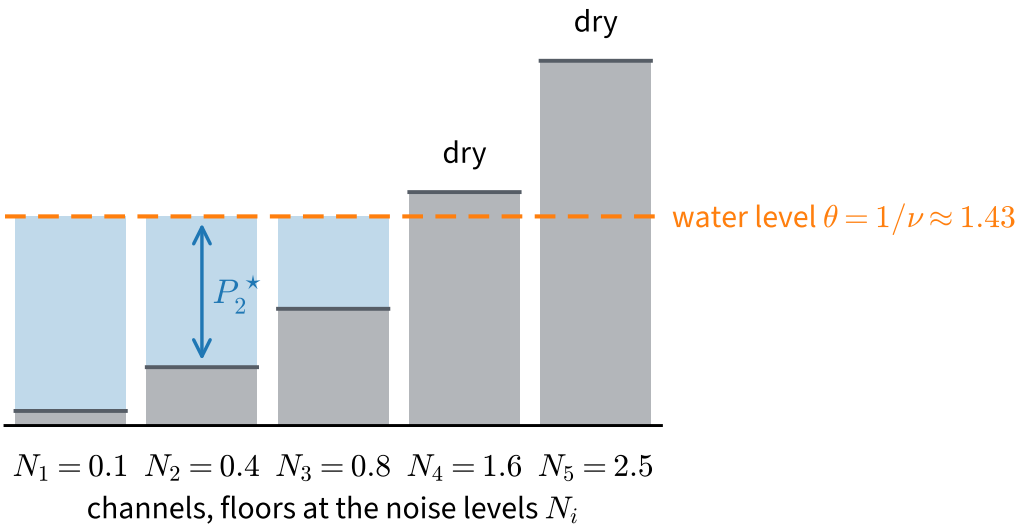


Figure 35.29: 下のコードセルの雑音フロアとバジェットに対する注水法。電力 $P = 3$ を底の高さ N_i の器に注ぐと、雑音が小さい上位 3 つのチャネルが共通の水位 $\theta = 1/\nu^\star \approx 1.43$ まで満たされてそれぞれ $P_i^\star = \theta - N_i$ が割り当てられる一方、底が θ より高い 2 つの器には相補性条件の要請どおり何も割り当てられない。

```
    lo, hi = N.min(), N.min() + P            # bracket for the water level
    for _ in range(steps):                   # bisection on the level theta
        mid = 0.5 * (lo + hi)
        lo, hi = (mid, hi) if np.maximum(0, mid - N).sum() < P else (lo, mid)
    theta = 0.5 * (lo + hi)
    return theta, np.maximum(0.0, theta - N)

theta, power = water_fill(P, N)
print('water level theta =', f'{theta:.6f}', ' P* =', power.round(4),
      ' total =', f'{power.sum():.6f}')
print('wet channels at the common level:', (N + power)[power > 0].round(6))
print('dry channels above the level:    ', bool((N[power == 0] >= theta).all()))
rate = lambda power: np.log(1.0 + power / N).sum()
dP = 1e-4                                    # shadow price, two ways
nu = 1.0 / theta
sens = (rate(water_fill(P + dP, N)[1]) - rate(water_fill(P - dP, N)[1])) / (2 * dP)
print(f'nu = 1/theta = {nu:.6f} vs finite-difference dU*/dP = {sens:.6f}')
```

```
water level theta = 1.433333  P* = [1.3333 1.0333 0.6333 0.     0.     ]  total = 3.000000
wet channels at the common level: [1.433333 1.433333 1.433333]
dry channels above the level:     True
nu = 1/theta = 0.697674 vs finite-difference dU*/dP = 0.697674
```

電力バジェットは機械精度で満たされており、稼働している 3 つのチャネルはすべて共通の水位 $\theta = (3 + 0.1 + 0.4 + 0.8)/3 \approx 1.4333$ に揃い、雑音の大きい 1.6 と 2.5 のチャネルは双対実行可能性の要請どおり乾いたまま留まっている。さらに、乗数 $\nu^\star = 1/\theta = 0.697674$ は有限差分による感度 $\mathrm{d}U^\star/\mathrm{d}P$ と表示された 6 桁すべてで一致している。

35.7.5.3 明示的な双対ギャップ

弱双対性はあらゆる問題に対して成り立つが、強双対性はそうではない。1 次元の非凸の例を見ると、そのギャップが具体的によくわかる。定義域 $\mathcal{D} = [0, 1]$ 上で次の問題を考えよう。

$$\min_{x \in [0,1]} \; f(x) = -x^2 \quad \text{subject to} \quad g(x) = x - \tfrac{1}{2} \le 0.$$

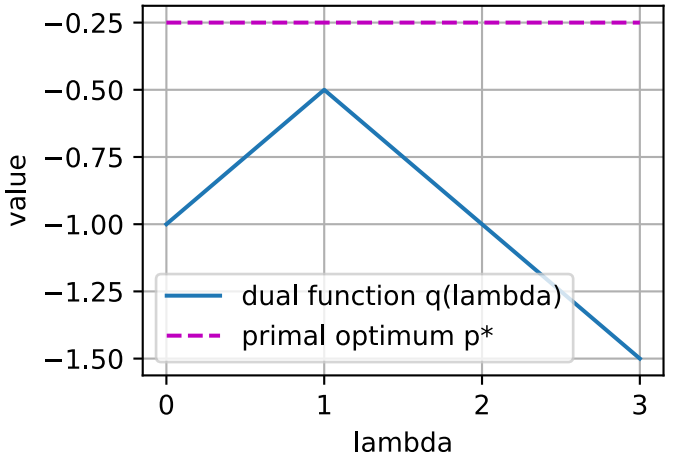
$f(x) = -x^2$ は凸ではなく凹であるため、この問題は非凸問題である。実行可能領域は $[0, \frac{1}{2}]$ であり、最小値は右端点で達成されて $p^\star = f(\frac{1}{2}) = -\frac{1}{4}$ となる。ここでは制約 $g(x) \le 0$ のみを双対化し、$\mathcal{D} = [0, 1]$ はハードな定義域制約として残すことで、Equation 35.80 の下限を $[0, 1]$ 上でとることにする（弱双対性はそのまま変わらず成り立つ）。任意の $\lambda \ge 0$ に対し、ラグランジアン $\mathcal{L}(x, \lambda) = -x^2 + \lambda(x - \frac{1}{2})$ は x に関して凹関数であるため、$[0, 1]$ 上での最小値は必ず両端点のいずれか、すなわち $\mathcal{L}(0, \lambda) = -\lambda/2$ または $\mathcal{L}(1, \lambda) = \lambda/2 - 1$ で達成される。したがって、次式が得られる。

$$q(\lambda) = \min\left(-\frac{\lambda}{2},\; \frac{\lambda}{2} - 1\right),$$

保証されているとおり、これは区分線形な凹関数である。2 本の直線は $\lambda^\star = 1$ で交わり、そこでは $d^\star = -\frac{1}{2} < -\frac{1}{4} = p^\star$ となって、大きさ $\frac{1}{4}$ の真に正の双対ギャップが生じる。$q(\lambda)$ を $p^\star$ と並べてプロットし、この非凸問題の双対曲線を本節冒頭のクリッピング問題のプロットと比較してみよう。

```
xs = np.linspace(0.0, 1.0, 4001)              # the domain D = [0, 1]
fx, gx = -xs**2, xs - 0.5                     # objective and constraint on D
p_star = fx[gx <= 0.0].min()                  # primal optimum: -1/4
lams = np.linspace(0.0, 3.0, 601)
q = np.array([(fx + lam * gx).min() for lam in lams])  # dual function on D
d_star, lam_star = q.max(), lams[q.argmax()]
print(f'p* = {p_star:.4f},  d* = {d_star:.4f} at lambda = {lam_star:.2f},'
      f'  gap = {p_star - d_star:.4f}')
d2l.plot(lams, [q, np.full_like(lams, p_star)], 'lambda', 'value',
         legend=['dual function q(lambda)', 'primal optimum p*'])
```

```
p* = -0.2500,  d* = -0.5000 at lambda = 1.00,  gap = 0.2500
```



滑らかに上昇して $p^\star$ に接したクリッピング問題の曲線とは異なり、このテント型の曲線は主最適値より丸々 4 分の 1 だけ低い $-\frac{1}{2}$ でピークを迎える。狭義実行可能点が確実に存在することにも注目しよう。実際、$x = \frac{1}{4}$ は $\mathcal{D}$ の内部にあり $g(\frac{1}{4}) = -\frac{1}{4} < 0$ を満たすため、このギャップはスレーター条件の不成立ではなく、もっぱら凸性の欠如によって引き起こされている。Figure 35.26 のパネル (b) に立ち返ると、像の曲線 $\{(u, -(u+\frac{1}{2})^2) : u \in [-\frac{1}{2}, \frac{1}{2}]\}$ が下向きに湾曲しているため、$\mathcal{A}$ の下側にあるいかなる支持直線も $(0, p^\star)$ に届くことができず、最も高い支持直線（傾き -1）でも $-\frac{1}{2}$ までしか保証できないのである。

未解決の課題が 2 つ残っている。第 1 に、非凸問題では双対問題は最適値の下界を与えるものの、最適値に到達するとは限らない。第 2 に、たとえ鞍点が存在する場合でも、$\mathbf{x}$ と λ を同時に更新すると収束しないことがあり、これがミニマックス訓練における中心的な難しさとなっている。深層ネットワーク全体の訓練は非凸であるが、その構成要素の多く（射影、クリップ付き更新、最終層の当てはめ、凸信頼領域部分問題など）は凸であり、本節の内容はそれらにそのまま適用できる。もっとも、ここまでの本章の導出はすべて厳密な実数演算を仮定してきた。次節では、最適化アルゴリズムそのものではなく有限の浮動小数点精度（floating-point precision）が結果を左右するときに何が起きるかを見ていく。

35.7.6 まとめ

任意の制約付き最適化問題に対し、ラグランジュ双対関数 $q(\lambda, \nu) = \inf_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \lambda, \nu)$ は凹関数であり、弱双対性 $q(\lambda, \nu) \le p^\star$ を満たすため、$\lambda \succeq 0$ を選ぶたびに主最適値の下界が保証される。主問題が凸でありスレーター条件（アフィンでないすべての不等式制約に対する狭義実行可能点の存在）を満たすとき、強双対性によってギャップはゼロになり（$d^\star = p^\star$）、達成された主・双対最適解の組はラグランジアンの鞍点をなす。線形制約の場合、双対関数は凸共役を用いて $q(\lambda) = -f^*(-\mathbf{A}^\top \lambda) - \mathbf{b}^\top \lambda$ に帰着し、log-sum-exp と負のエントロピーのような共役対から直ちに双対定式化が得られる。強双対性のもとでは、最適乗数はシャドープライスとして働き（$\lambda_i^\star = -\partial p^\star / \partial u_i$ および $\nu_j^\star = -\partial p^\star / \partial v_j$ が $p^\star$ の微分可能な点で成り立つ）、凸損失において重み減衰がノルム制約と一致する理由、SVM の乗数がサポートベクター以外でゼロになる理由、そして注水法の乗数 $1/\theta$ が電力 1 単位あたりの通信レートを測る理由を説明する。凸性がなければ、$-x^2$ を $[0, 1]$ 上で制約 $x \le \frac{1}{2}$ のもとで最小化する例が示したように、真に正の双対ギャップが残りうる。

35.7.7 演習問題

1. **共役対**。@eq-mdl-opt-conjugate から以下の各凸共役を計算せよ。
   1. $f(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$（ただし $\lambda > 0$）。
   2. $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x}$（ただし $\mathbf{A} \succ 0$）。フェンシェル・ヤングの不等式 Equation 35.85 はこの対に対して何を主張するか。
   3. ソフトプラス $f(t) = \log(1+e^t)$（定義域 $\mathbb{R}$）。ロジット t のソフトラベル $p \in [0,1]$ に対する二値交差エントロピーは $\log(1+e^t) - pt$ である。フェンシェル・ヤングの不等式を用いて、これが少なくとも二値エントロピー $H(p) = -p\log p - (1-p)\log(1-p)$ 以上であり、等号が成り立つのは $p = 1/(1+e^{-t})$（t のシグモイド）のとき、かつそのときに限ることを示せ。さらに、両者の差を KL ダイバージェンスとして表せ。
   4. 二重井戸関数 $\phi_0(x) = (x^2-1)^2$（35.3.4 節の関数 ϕ から傾き項 $x/2$ を除いたもの）は凸ではない。その凸包絡関数 ϕ_0^{**} と、$\phi_0^{**} < \phi_0$ となる集合を求めよ。

   Boyd および Vandenberghe の『Convex Optimization』例 3.22、3.26 を改変。

2. **最大エントロピーと温度**。確率ベクトル $\mathbf{p} \in \mathbb{R}^n$ のエントロピーは $H(\mathbf{p}) = -\sum_i p_i \log p_i$ である。本問では制約側からソフトマックスを導出する。
   1. H を制約 $\sum_i p_i = 1$ のもとで 1 つのラグランジュ乗数を用いて最大化せよ（符号制約 $p_i \geq 0$ は無視する）。唯一の停留点が一様分布であることを示せ。符号制約を無視しても答えが変わらないのはなぜか。
   2. すべてが等しいわけではないスコア $s_1, \dots, s_n$ と、目標値 $\bar{s}$（$\min_i s_i$ と $\max_i s_i$ の真に間にある値）を固定し、制約 $\sum_i p_i s_i = \bar{s}$ を追加せよ。最大点が $\mathbf{p} = \mathrm{softmax}(\beta \mathbf{s})$（スカラー β はスコア制約の乗数から定まる）となることを示せ。また、$\mathrm{softmax}(\beta\mathbf{s})$ のもとでの平均スコアが β に関して狭義単調増加であり、目標値を満たす β がただ 1 つ存在することを示せ。
   3. $H^\star(\bar{s})$ を最大エントロピーとする。$\mathrm{d}H^\star/\mathrm{d}\bar{s} = -\beta$ であり、35.6.1.3 節の乗数がオフセット b を価格付けするのと同様に、この乗数がスコア制約を価格付けすることを示せ。どのような目標値 $\bar{s}$ に対して β は負になるか。また、ソフトマックス温度 $1/\beta$ は $\bar{s}$ が $\max_i s_i$ に近づくときどうなるか。

3. **ℓ_1 最小化に対する双対保証**。基底追跡法

$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{35.92}$$

   （ただし $\mathbf{A} \in \mathbb{R}^{m\times n}$、$m < n$）は、劣決定な連立一次方程式からスパースな解を選び出す。制約はアフィンであるため、スレーター条件には実行可能点が 1 つあれば十分であり、方程式が解を持つ限り $d^\star = p^\star$ が成り立つ。また $\|\cdot\|_1$ の凸共役はボックス $\{\mathbf{y} : \|\mathbf{y}\|_\infty \leq 1\}$ の指示関数である。
   1. Equation 35.86 の等式制約版を用い、ν を $-\nu$ に置き換えると、双対問題が $\max_\nu \mathbf{b}^\top\nu$（制約 $\|\mathbf{A}^\top\nu\|_\infty \leq 1$）になることを示せ。
   2. $\mathbf{A} = \begin{pmatrix}1 & 2\end{pmatrix}$ および $b = 2$ に対して両方の問題を手計算で解き、$d^\star = p^\star$ を確認せよ。
   3. $\mathbf{x}^\star$ と $\nu^\star$ を最適解とする。$x_i^\star = 0$ が $|(\mathbf{A}^\top\nu^\star)_i| < 1$ となるすべての成分で成り立ち、$(\mathbf{A}^\top\nu^\star)_i = \mathrm{sign}(x_i^\star)$ が $x_i^\star \neq 0$ となるすべての成分で成り立つことを示せ。候補解 $\mathbf{x}$（$\mathbf{A}\mathbf{x} = \mathbf{b}$ を満たす）が与えられたとする。問題を解くことなくベクトル ν によって $\mathbf{x}$ が最適解であることを保証するには、どのような条件を満たせばよいか。

4. **サポートベクターと双対値**。オフセットを折り込んだ SVM 双対問題 Equation 35.90 において、$\alpha^\star$ を最適解とし、$\tilde{\mathbf{w}}^\star = \sum_i \alpha_i^\star y_i \tilde{\mathbf{x}}_i$ とおく。

1. 非負象限上の双対問題の KKT 条件を書き下すことで、$\sum_i \alpha_i^\star = \|\tilde{\mathbf{w}}^\star\|^2$ を示せ。さらに、双対最適値が主問題の値 $\frac{1}{2}\|\tilde{\mathbf{w}}^\star\|^2$ に等しいことを導け。
2. $\alpha_j^\star = 0$ である訓練点を削除しても、残りの乗数は縮小された問題に対して最適なままであり、$\tilde{\mathbf{w}}^\star$ も変化しないことを示せ。1 つ抜き検証では、n 個の各点を残りの $n-1$ 点で訓練した SVM によって分類する。1 つ抜き誤り率は高々 $\alpha_i^\star > 0$ となる点の割合以下であることが、すべての最適解 $\alpha^\star$ に対して成り立つことを導け。どの最適解 $\alpha^\star$ が最良の上界を与えるか。
3. [code] SVM のコードセルの行列 $\mathbf{Q}$ のランクは 3 である。S を $\alpha_i^\star > 0$ となる 4 つの添字の集合とし、$\mathbf{G}_S$ を `G` の対応する行からなる部分行列とする。非ゼロの $\mathbf{d} \in \mathbb{R}^4$ で $\mathbf{G}_S^\top \mathbf{d} = \mathbf{0}$ を満たすものを求め（`np.linalg.svd` を $\mathbf{G}_S^\top$ に適用せよ）、$\alpha_S^\star + t\mathbf{d}$ をいずれかの成分がゼロに達するまで両方向へ動かせ。両方の端点が双対実行可能であり、双対値 $\frac{41}{98}$ を持ち、同じ $\tilde{\mathbf{w}}^\star$ を与えることを検証せよ。各端点において、マージン 1 のどの点が $\alpha_i = 0$ になるか。このことは、乗数から「サポートベクター」を読み取ることについて何を意味するか（端点は $(0, 0.204, 0, 0, 0, 0.192, 0, 0.441)$ および $(0.204, 0, 0, 0, 0, 0.273, 0, 0.359)$ となるはずである）。

5. [code] **勾配降下上昇法。**●主双対法や GAN の訓練では、最小化変数を勾配降下法で、最大化変数を勾配上昇法で同時に更新する。

1. 原点を鞍点とする双線形ゲーム $\min_x \max_y \, xy$ に対し、同時降下上昇法 $x \leftarrow x - \eta y$ および $y \leftarrow y + \eta x$ を線形変換として表せ。任意の $\eta > 0$ に対し、ゼロでないあらゆる初期点から鞍点を離れて渦巻状に発散することを示せ。また、y の更新に新しい x を用いる交互更新版が、行列式 1 の線形変換であることを示せ。$\eta < 2$ のとき、その反復点はどのように振る舞うか。
2. 35.6.1.3 節の最近接点問題のラグランジアン $\mathcal{L}(\mathbf{x}, \nu) = \frac{1}{2}\mathbf{x}^\top \mathbf{x} + \nu(\mathbf{a}^\top \mathbf{x} - b)$ に対し、ここでは $\mathbf{a} = (1, 2)$、$b = 1$、初期値 $\mathbf{x} = \mathbf{0}$、$\nu = 0$ として同時降下上昇法を実行せよ。反復点が $(\mathbf{x}^\star, \nu^\star)$ に収束するステップサイズの範囲を求めよ。さらにその範囲を、組 $(\mathbf{a}^\top \mathbf{x}/\|\mathbf{a}\|, \nu)$ と、$\mathbf{x}$ のうち $\mathbf{a}$ に直交する成分のそれぞれに作用する更新の固有値から導出せよ。
3. 双線形ゲームの目的関数に $\frac{\epsilon}{2}x^2$（ただし $0 < \epsilon \leq 2$）を加えよ。このとき同時降下上昇法はどのステップサイズに対して収束するか。また、前問のラグランジアンのどの項が ϵ の役割を果たしているか。

6. [code] **凸性がない場合のペナルティと制約の比較。**●凸損失に対しては、35.7.3.2 節で重み減衰とノルム制約が同じ解を描くことを示した。代わりに非凸の 1 次元損失 $\ell(w) = (w-1)^2(w-3)^2 - w$（極小点は $w = 1.16$ と $w = 3.11$ の付近にある）を考えよ。

1. w の細かいグリッド上で、対数グリッド上の各ペナルティ付き最小点 $w_\lambda = \operatorname{argmin}_w \ell(w) + \frac{\lambda}{2}w^2$（各 λ に対するもの）と、制約付き最適値 $p^\star(r) = \min\{\ell(w) : w^2 \leq r^2\}$（$r \in [0, 3.2]$）を計算せよ。点 $(w_\lambda^2, \ell(w_\lambda))$ を曲線 $(r^2, p^\star(r))$ の上に重ねてプロットせよ。バジェット r のうち、いかなるペナルティ付き最小点のノルム $|w_\lambda|$ にも決して一致しないものはどの範囲か。
2. w_λ が $\ell(w) + \frac{\lambda}{2}w^2$ を最小化することは、傾き $-\frac{\lambda}{2}$ で $(w_\lambda^2, \ell(w_\lambda))$ を通る直線が集合 $\{(w^2, \ell(w)) : w \in \mathbb{R}\}$ のすべての点の下側にあることと同値であることを示せ。この事実と Figure 35.26 の支持直線の描像を用いて、欠落するバジェットが生じる理由を説明せよ。
3. 欠落する範囲から制約が有効となるバジェット r を 1 つ選べ。本文と同様に制約を $\frac{1}{2}w^2 \leq \frac{1}{2}r^2$ と書いたときの制約付き解の KKT 乗数 λ を計算し、この λ を用いたペナルティ付き目的関数がより小さい値をとる点を求めよ。さらに、この制約付き問題の双対ギャップ $p^\star(r) - d^\star(r)$ を計算せよ。

7. [extended] **エントロピー正則化付き最適輸送。●●**確率ベクトル $\mathbf{p} \in \mathbb{R}^n$ と $\mathbf{r} \in \mathbb{R}^m$、およびコスト行列 $\mathbf{C} \in \mathbb{R}^{n \times m}$ が与えられたとき、最適輸送は最も低コストな輸送計画 Γ で質量 $\mathbf{p}$ を質量 $\mathbf{r}$ へ移すものを求める。

$$\mathrm{OT} = \min_{\Gamma \geq 0} \sum_{i,j} C_{ij}\Gamma_{ij} \quad \text{subject to} \quad \Gamma\mathbf{1} = \mathbf{p}, \;\; \Gamma^\top \mathbf{1} = \mathbf{r}, \tag{35.93}$$

これは nm 変数の線形計画問題である。エントロピー正則化 (Cuturi 2013) では、目的関数に $\varepsilon \sum_{i,j} \Gamma_{ij}(\log \Gamma_{ij} - 1)$ (ただし $\varepsilon > 0$) を加える。正則化後の最適解を Γ_ε とする。シンクホーン反復によるその解法は、分布間の比較に広く用いられる (Peyré and Cuturi 2019)。

   1. 周辺分布の制約に対する項 $\phi^\top(\mathbf{p} - \Gamma\mathbf{1}) + \psi^\top(\mathbf{r} - \Gamma^\top\mathbf{1})$ を用いてラグランジアンを作れ。$\Gamma_\varepsilon = \mathrm{diag}(\mathbf{u})\,\mathbf{K}\,\mathrm{diag}(\mathbf{v})$ (ただし $K_{ij} = e^{-C_{ij}/\varepsilon}$、$u_i = e^{\phi_i/\varepsilon}$、$v_j = e^{\psi_j/\varepsilon}$) となることを示せ。符号制約 $\Gamma_{ij} \geq 0$ をラグランジアンから省いてよいのはなぜか。
   2. 双対関数が

$$q(\phi, \psi) = \mathbf{p}^\top\phi + \mathbf{r}^\top\psi - \varepsilon \sum_{i,j} e^{(\phi_i + \psi_j - C_{ij})/\varepsilon}, \tag{35.94}$$

      となること、およびこれを ϕ について（ψ を固定して）厳密に最大化すると要素ごとのシンクホーン更新 $\mathbf{u} = \mathbf{p}/(\mathbf{K}\mathbf{v})$ が得られることを示せ。これより、シンクホーンのアルゴリズムは凹関数上のブロック座標上昇法であり、双対値が決して減少しないことを導け。すべての双対値が OT の下界にもなるのはなぜか。
   3. $H(\Gamma) = -\sum_{i,j}\Gamma_{ij}\log\Gamma_{ij}$ とおき、すべての実行可能な輸送計画が $\max(H(\mathbf{p}), H(\mathbf{r})) \leq H(\Gamma) \leq H(\mathbf{p}) + H(\mathbf{r})$ を満たすことを認めるものとする。このとき $0 \leq \sum_{i,j} C_{ij}(\Gamma_\varepsilon)_{ij} - \mathrm{OT} \leq \varepsilon \min(H(\mathbf{p}), H(\mathbf{r}))$ を示せ。
   4. コストが $[0, 1]$ の範囲にあるとする。float64 では、e^{-x} は x が約 745 を超えるとゼロに丸められる。どのような ε に対して $\mathbf{K}$ の成分がゼロになりうるか、またそのとき乗法的な更新 $\mathbf{u} = \mathbf{p}/(\mathbf{K}\mathbf{v})$ に何が起きるか。ϕ と ψ を直接更新し、$\mathbf{K}$ との積の代わりに log-sum-exp を用いればこの問題を回避できる理由を説明せよ。

8. [code] **リッジ回帰の双対問題。** $\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{A}\mathbf{w} - \mathbf{b}\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$（ただし $\mathbf{A} \in \mathbb{R}^{n \times d}$）を、$\min_{\mathbf{w}, \mathbf{z}} \frac{1}{2}\|\mathbf{z}\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$（制約 $\mathbf{A}\mathbf{w} - \mathbf{b} - \mathbf{z} = \mathbf{0}$、乗数 $\alpha \in \mathbb{R}^n$）と書き換えよ。

   1. ラグランジアンを $\mathbf{z}$ と $\mathbf{w}$ について最小化し、$\mathbf{z} = \alpha$、$\mathbf{w} = -\mathbf{A}^\top\alpha/\lambda$、および双対関数 $q(\alpha) = -\frac{1}{2}\|\alpha\|^2 - \frac{1}{2\lambda}\alpha^\top\mathbf{A}\mathbf{A}^\top\alpha - \mathbf{b}^\top\alpha$ を導け。なぜ強双対性が成り立つのか。
   2. q を最大化して $\alpha^\star = -\lambda(\mathbf{A}\mathbf{A}^\top + \lambda\mathbf{I})^{-1}\mathbf{b}$ および $\mathbf{w}^\star = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \lambda\mathbf{I})^{-1}\mathbf{b}$ を求めよ。SVM の重みベクトルがデータ点の線形結合になるのと同様に、なぜ解が $\mathbf{A}$ の行ベクトルの線形結合になるのか、またなぜデータが $\mathbf{A}\mathbf{A}^\top$ のみを通じて現れるのかを説明せよ。
   3. データ例数 $n = 20$、特徴量数 $d = 50$ に対し、主問題の解法 $(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{I})\mathbf{w} = \mathbf{A}^\top\mathbf{b}$ と双対公式が約 10^{-14} の精度で一致すること、および主問題の値と双対値が一致することを数値的に検証せよ。$n \ll d$ のとき計算コストが低いのはどちらの解法か。また、$\mathbf{A}\mathbf{A}^\top$ をカーネル行列に置き換えると何が得られるか。

35.8 数値安定性と条件性

本章のこれまでの計算はすべて浮動小数点（floating point）で実行し、その結果をあたかも厳密な実数であるかのように読み取ってきた。大抵の場合、この便利な仮定はそのまま通用する。しかし時折、訓練損失が NaN になったり、線形ソルバが一見まったく妥当に見えながら大きく誤った解を返したりすることがある。本節では、最適化アルゴリズムではなく有限精度演算が計算結果を左右するのはどのような場合か、そしてそれにどう対処できるかを考える。

2 つの小さな例から、数値的な問題が生じる 2 通りの経路がわかる。第 1 に、3 つのロジット $(1000, 1001, 1002)$ は確率 $(0.09, 0.24, 0.67)$ を持つごく普通のソフトマックスを定める。ところが教科書通りの式をそのまま評価すると、倍精度（fp64）でさえ 3 つの NaN が返る。第 2 に、`np.linalg.solve` をある 12×12 の連立一次方程式に適用すると、与えられた方程式と 10 進法で 17 桁目しか違わない方程式の厳密解であるにもかかわらず、真の解から 30% もずれた答えが返る。前者の失敗は NaN の損失を出して訓練を停止させるが、後者は気づかれないまま通り過ぎてしまう。

これら 2 つの失敗の原因は根本的に異なる。素朴なソフトマックスの式は、いかなる標準的な浮動小数点形式（floating-point format）でも保持できないほど巨大な数 e^{1000} を経由するが、e^{1000} を作らない代数的に同値な書き換えを行えば、最終桁の数単位まで正確な値が得られる。対照的に、この 12×12 の連立一次方程式はきわめて敏感であり、データの 16 桁目に生じた丸め誤差（rounding error）が解の最初の桁を動かしてしまうため、ソルバをどのように書き換えてもその感度を取り除くことはできない。

これら 2 つの状況を見分けるため、まずコンピュータが実数をどのように格納し、1 回の丸めがどれだけの誤差を生むかを調べる。次に、ソフトマックスと交差エントロピー（cross-entropy）を形式が保持できない数を決して経由しないように書き換え、さらに分散についてもほぼ等しい大きな数同士を決して減算しないように書き換える。続いて、感度の尺度である関数の条件数（condition number）を導入し、不適切なアルゴリズムと本質的に難しい問題を切り分ける。最後に、連立一次方程式の条件数が、最小二乗法による求解で保持できる桁数と、35.1 節で見た勾配降下法に必要なステップ数の双方をいかに支配するかを確認する。5.5 節では現代の訓練が実際に低精度形式をどう活用するかを扱ったが、本節ではそれらの形式での計算がなぜ成功し、なぜ失敗するのかを導出する。

個々の公式以上に重要な考え方が 3 つある。浮動小数点演算の相対丸めモデル、不安定なアルゴリズムとは良条件（well-conditioned）な問題を悪条件（ill-conditioned）な中間ステップへ迂回させるものだという原理、そしてソルバの解の誤差を問題の感度に結びつける上界である。総和の順序とリッジ正則化に関する項は、初読時には飛ばしてもよい。

```
from fractions import Fraction
```

35.8.1 浮動小数点演算

35.8.1.1 表現と間隔

コンピュータに 1.1 を格納させると、実際には何が格納されるのだろうか。この仕組みは、2 進小数点以下を 2 ビットだけ保持するトイ 2 進形式で考えると最もわかりやすい。1 と 2 の間では、この形式は $(1.00)_2 = 1$、$(1.01)_2 = 1.25$、$(1.10)_2 = 1.5$、$(1.11)_2 = 1.75$ の 4 つの値しか保持できない。これら 4 つの値に 2 のべき乗を掛けると他の区間も埋まり、$2, 2.5, 3, 3.5$ が 2 と 4 の間に、$0.5, 0.625, 0.75, 0.875$ が 0.5 と 1 の間に並ぶ。隣り合う 2 のべき乗に挟まれた区間 $[2^e, 2^{e+1})$ である各**オクターブ**にはちょうど 4 つの値が入るため、大きさが 2 倍になるたびに隣り合う数の絶対間隔も 2 倍になる。

グリッド点の間に落ちる実数は、最も近いグリッド点へ丸められる。たとえば $1.1 \mapsto 1$（相対誤差 -0.091）、$1.3 \mapsto 1.25$（誤差 -0.038）、$3.3 \mapsto 3.5$（誤差 $+0.061$）となる。ちょうど中間に位置する点は同距離（タイ）となり、標準的な演算では最下位ビットが 0 である側の隣接値へ丸める。したがって中点 1.125 は 1 へ丸められ、誤差は -0.111 となる。どの相対誤差も $1/8$ を超えないことに注目しよう。丸めによって数が動く量は高々間隔の半分であり、どのオクターブでも間隔の半分はそのオクターブの最小値の $1/8$ だからである。Figure 35.30 に実数直線上のこのグリッドを示す。

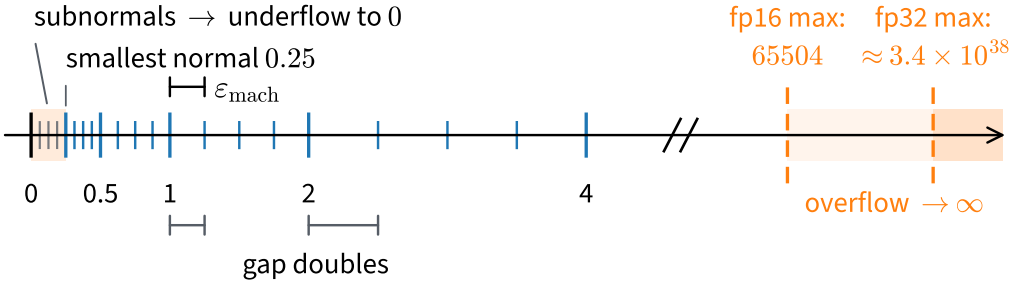


Figure 35.30: 仮数部 2 ビットのトイ形式（1 オクターブあたり 4 値、$\varepsilon_{\text{mach}} = 1/4$、最小の正規化数 0.25）における実数直線上の浮動小数点数。隣り合う値との絶対間隔は 2 のべき乗ごとに 2 倍になる一方、相対間隔は $\varepsilon_{\text{mach}}$ 付近に保たれる。最小の正規化数より下では非正規化数が同じ間隔のままゼロまで続き、軸の省略記号の先では実際の形式がそれぞれのオーバーフロー閾値（fp16 では 65504、fp32 では約 3.4×10^{38}）で終わる。

実際の浮動小数点形式もこれとまったく同じ仕組みであり、単にビット数が多いだけである。浮動小数点数とは、2 進小数点以下に固定長 p 桁の 2 進数字を持つ基数 2 の科学記数法であり、次の式で表される。

$$x = (-1)^{s} \cdot (1.b_1 b_2 \ldots b_p)_2 \cdot 2^{e}, \tag{35.95}$$

ここで、s は符号ビット、$b_1, \ldots, b_p$ は**仮数部**（mantissa または significand）のビット、e は固定された範囲から選ばれる整数の指数部である。指数部はその形式が表現できる大きさの範囲を定め、仮数部は相対精度を定める。オクターブ $[2^{e}, 2^{e+1})$ の内部では、隣り合う値は 2^{e-p} ずつ離れている。1 からその次に表現可能な数までの間隔 $\varepsilon_{\text{mach}} = 2^{-p}$ は**計算機イプシロン**（machine epsilon）と呼ばれ、先ほどのトイ形式では $p = 2$、$\varepsilon_{\text{mach}} = 1/4$ である。

実数 x を最も近い表現可能な値へ丸めると、その移動量は高々間隔の半分である 2^{e-p-1} に収まる。オクターブ全体で $|x| \geq 2^{e}$ であるから、相対誤差は高々 2^{-p-1} である。許容範囲内の指数部を持ち Equation 35.95 の形で表される値を**正規化数**（normal number）と呼ぶことにすると、正規化数へ丸められるすべての実数 x は次を満たす。

$$\operatorname{fl}(x) = x\,(1+\delta), \qquad |\delta| \leq u = \tfrac{1}{2}\,\varepsilon_{\text{mach}} = 2^{-p-1}. \tag{35.96}$$

言い換えれば、1 回の丸めが数に与える摂動は高々 2^{p+1} 分の 1 であり、トイ形式では 8 分の 1、fp32（$p = 23$）では約 1700 万分の 1、bf16（$p = 7$）では 256 分の 1 である。この上界 u を**丸めの単位**（unit roundoff）と呼ぶ。標準の IEEE 演算はすべての基本演算の結果を最も近い表現可能な値へ丸めるため、計算された和 $x \oplus y$ は $(x+y)(1+\delta)$（ただし $|\delta| \leq u$）に等しくなり、同じ保証が $-$、$\times$、$/$ に対しても成り立つ (IEEE 2019; Goldberg 1991)。

この明快な相対モデルには 3 つの例外がある。形式が保持できる上限を超える結果は ∞ へと**オーバーフロー**する。最小の正規化数より小さい結果は**アンダーフロー**し、ゼロまでの残りの隙間を埋める**非正規化数**（subnormal number）を段階的に辿って減少する。$0/0$ や ∞/∞ のような未定義の演算は NaN を生じる。これら 3 つはいずれも相対的ではなく絶対的なスケールの現象であり、標準的な形式を一覧した直後に改めて取り上げる。

35.8.1.2 形式

深層学習では、倍精度（fp64）より幅の狭い 5 つの形式が用いられる。各形式の計算機イプシロン $\varepsilon_{\text{mach}}$、最小の正規化数、最大の有限値をライブラリから直接確認してみよう。

```
print(f'{"dtype":>14} {"eps":>10} {"smallest normal":>16} {"max":>10}')
for dt in [torch.float16, torch.float32, torch.bfloat16,
           torch.float8_e4m3fn, torch.float8_e5m2]:
    fi = torch.finfo(dt)
    print(f'{str(dt).removeprefix("torch."):>14} {fi.eps:10.3e} '
          f'{fi.smallest_normal:16.3e} {fi.max:10.3e}')
```

```
         dtype        eps  smallest normal        max
       float16  9.766e-04        6.104e-05  6.550e+04
       float32  1.192e-07        1.175e-38  3.403e+38
      bfloat16  7.812e-03        1.175e-38  3.390e+38
  float8_e4m3fn  1.250e-01        1.562e-02  4.480e+02
    float8_e5m2  2.500e-01        6.104e-05  5.734e+04
```

fp32 は仮数部に $p = 23$ ビット（10 進で約 7 桁）を保持し、最大 3.4×10^{38} まで達する。**fp16** は $p = 10$ ビット（10 進で約 3 桁）を保持するが、指数部の範囲が狭く、65504 を超えるとオーバーフローし、$2^{-14} \approx 6.1 \times 10^{-5}$ 未満では非正規化数になる。**bf16**（ライブラリでは `bfloat16`）はこれとは逆のトレードオフをとる。fp32 の 8 個の指数部ビットをすべて維持してその広大なダイナミックレンジを保つ代わりに、仮数部をわずか $p = 7$ ビット（$\varepsilon_{\text{mach}} = 2^{-7} \approx 0.0078$、10 進で約 2〜3 桁）に抑えている。なお、bf16 についてよく引用される 2^{-8} という値は、丸めの単位 $u = \varepsilon_{\text{mach}}/2$ である。2 つの fp8 形式はこのトレードオフをさらに推し進め、E4M3 は仮数部 3 ビット（10 進で約 1 桁）を保持して上限が 448 となり、E5M2 は fp16 と同じ指数部範囲で仮数部 2 ビットを保持する。これほどダイナミックレンジが狭いため、fp8 が実用になるのはテンソルごと、あるいは値のブロックごとに独立したスケール係数を組み合わせた場合に限られる。

bf16 は単に fp32 の上位 16 ビット（同じ符号ビット、同じ 8 個の指数部ビット、および先頭 7 個の仮数部ビット）であるため、fp32 から bf16 への変換は下位 16 ビットを丸め落とすことに帰着する。この丸めは、生のビットパターンに対する整数演算で直接実装できる。切り捨てられる 16 ビットの中点は `0x8000` であるから、`0x7FFF`（中点より 1 小さい値）を加え、さらに残す最下位ビットが 1 であればもう 1 を加えてから、下位 16 ビットをゼロにクリアする。この加算で残すビット側へ桁上がりが生じるのは、切り捨て部分が中点より真に大きいか、あるいはちょうど中点にあって残す最下位ビットが奇数であるとき、かつそのときに限る。これがわずか 1 行のビット操作による最近傍偶数丸め（round-to-nearest, ties-to-even）であり、演習問題 5 ではこの上に確率的丸めを構築する。中点より上の入力 1 つと、厳密な中点（タイ）2 つで動作を確認しよう。

```
def to_bf16(x):
    """Round float32 to the nearest bfloat16, ties to even."""
    bits = np.atleast_1d(np.asarray(x, np.float32)).view(np.uint32)
    bits = (bits + 0x7FFF + ((bits >> 16) & 1)) & 0xFFFF0000
    return bits.astype(np.uint32).view(np.float32)
```

```
    for x, expected in [(1 + 2**-8 + 2**-10, 1 + 2**-7),   # above the midpoint
                        (1 + 2**-8, 1.0),                  # tie, 1 is even
                        (1 + 3 * 2**-8, 1 + 2**-6)]:       # tie, 1 + 2^-6 is even
        print(f'{x:.8f} -> {to_bf16(x).item():.8f}   (expected {expected:.8f})')
```

```
1.00488281 -> 1.00781250   (expected 1.00781250)
1.00390625 -> 1.00000000   (expected 1.00000000)
1.01171875 -> 1.01562500   (expected 1.01562500)
```

予想通り、最初の入力は 1 と $1 + 2^{-7}$ の中点より上にあるため切り上げられる。2 番目の入力はちょうどその中点上にあり、偶数の隣接値である 1 へ切り捨てられるのに対し、3 番目の入力は $1 + 2^{-7}$ と $1 + 2^{-6}$ の中点上にあり、偶数の隣接値である $1 + 2^{-6}$ へ切り上げられる。次に、fp32 でも同じタイ規則を確認し、大きさに応じて間隔がどう広がるかを観察するとともに、各形式でオーバーフローせずに指数関数を計算できる最大のロジットを求めてみよう。

```
    eps = np.finfo(np.float32).eps
    one = np.float32(1.0)
    print('1 + eps   != 1 :', one + eps != one)
    print('1 + eps/2 == 1 :', one + np.float32(eps / 2) == one)
    print('gap between float32 neighbors at 1    :', np.spacing(one))
    print('gap between float32 neighbors at 2^20 :', np.spacing(np.float32(2.0**20)))
    for dt in [np.float16, np.float32, np.float64]:
        print(f'{np.dtype(dt).name}: exp(x) overflows for x >',
              f'{np.log(np.finfo(dt).max):.2f}')
```

```
1 + eps   != 1 : True
1 + eps/2 == 1 : True
gap between float32 neighbors at 1    : 1.1920929e-07
gap between float32 neighbors at 2^20 : 0.125
float16: exp(x) overflows for x > 11.09
float32: exp(x) overflows for x > 88.72
float64: exp(x) overflows for x > 709.78
```

ここで $1 + \varepsilon_{\text{mach}}/2$ は 1 とその上側の隣接値との中点であるため、偶数丸めによって 1 に戻される。間隔の半分という最悪ケースの丸め誤差は実際に発生し、相対誤差は $u/(1+u)$ となって上界 $|\delta| \le u$ のわずかに内側に収まる。一方、2^{20} における間隔は 1 における間隔の 2^{20} 倍（約 100 万倍）である。そして最後の 3 行は e^x がオーバーフローする境界を特定しており、fp16 では 11.09、fp32 では 88.72、fp64 では 709.78 を超えるとオーバーフローする。

35.8.1.3 オーバーフロー、アンダーフロー、微小な更新

指数関数は加法的なスケールを乗法的なスケールに変えるため、さほど大きくないロジットでもオーバーフローしうる。fp16 では、上限 $\log 65504 \approx 11.09$ は通常の未正規化な分類器スコアがとりうる範囲の十分に内側にある。

その反対側では、e^{-x} がアンダーフローする。最小の正規化数を下回ると、形式は同じ絶対間隔を保ちながら有効ビットを 1 つずつ失う非正規化数を辿り、やがて完全に 0 へ丸められる。後から $\log 0$ をとると $-\infty$ が生じ、逆伝播は `inf` や `NaN` で埋め尽くされる。fp16 では $2^{-14} \approx 6.1 \times 10^{-5}$ 未満で非正規化数が始まり、最小の非正規化数は $2^{-24} \approx 6 \times 10^{-8}$ であり、$2^{-25} \approx 3 \times 10^{-8}$ 未満の値はすべてそのままゼロへ丸められる。

加算からは、訓練に直接関わる第 2 の絶対閾値が生じる。勾配ステップは重み w を $w - \eta g$ に置き換えるが、更新量 ηg が w 周辺の間隔に比べて小さいと、加算結果は単に元の w に丸め戻され、ステップは跡形もなく消えてしまう。

命題（微小な更新の消失）。$w > 0$ を、$2^e \le w < 2^{e+1}$ を満たす仮数部 p ビット形式の浮動小数点数とし、Δ を実数とする。$|\Delta| < 2^{e-p-1}$ ならば $\mathrm{fl}(w + \Delta) = w$ が成り立つ。ただし $w = 2^e$ かつ $\Delta < 0$ の場合は例外であり、このときは $|\Delta| < 2^{e-p-2}$ に対して同じ結論が成り立つ。

証明。 $[2^e, 2^{e+1}]$ では表現可能な数は 2^{e-p} 間隔で並ぶため、w の隣接値は $w \pm 2^{e-p}$ である。ただし $w = 2^e$ より下では 1 つ下のオクターブ $[2^{e-1}, 2^e)$ が始まり、間隔は半分の 2^{e-p-1} になる。最近傍丸めが w を返すのは、$w + \Delta$ がその側にある隣接グリッド点よりも w に近いとき、すなわち $|\Delta|$ がその間隔の半分より真に小さいときである。■

重みの更新 $\Delta = -\eta g$ において $2^e < w < 2^{e+1}$ のとき、この命題は次のようになる。

$$\mathrm{fl}(w - \eta g) = w \qquad \text{whenever } |\eta g| < 2^{e-p-1} = u\,2^e. \tag{35.97}$$

要するに、およそ $u|w|$ より小さい更新はすべて失われる。相対モデル Equation 35.96 は積や商をよく記述するが、大きさの大きく異なる 2 つの数を足し合わせる場合には、この鋭い絶対的なカットオフが見えなくなる。

$w = 1$ のようにちょうど 2 のべき乗の上にある重みは、命題の例外ケースに該当する。1 より下では間隔が半分になるため、増加方向では u 未満で消える（5.5 節で実験した通りである）のに対し、減少方向ではすでに $u/2$ 未満で消えてしまう。したがって bf16 では、1 に等しい重みは $2^{-9} \approx 0.002$ より小さいあらゆる減少を無視するため、学習率が 10^{-3} でオーダー 1 の勾配が与えられても重みは完全に凍結されたままになる。対応する閾値は fp16 では $2^{-12} \approx 2.4 \times 10^{-4}$、fp32 では $2^{-25} \approx 3 \times 10^{-8}$ であり、混合精度（mixed precision）訓練で重みのマスターコピーを fp32 で保持するのはこのためである。これらの閾値を跨ぐステップを試してみよう（bf16 の更新は fp32 で計算し、`to_bf16` で丸める）。

```
one32, one16 = np.float32(1.0), np.float16(1.0)
print('bfloat16: 1 - 1e-3 ->', to_bf16(one32 - np.float32(1e-3)).item())
print('bfloat16: 1 - 2e-3 ->', to_bf16(one32 - np.float32(2e-3)).item())
print('fp16    : 1 - 1e-4 ->', one16 - np.float16(1e-4))
print('fp16    : 1 - 3e-4 ->', one16 - np.float16(3e-4))
print('fp32    : 1 - 1e-3 ->', one32 - np.float32(1e-3))
```

```
bfloat16: 1 - 1e-3 -> 1.0
bfloat16: 1 - 2e-3 -> 0.99609375
fp16    : 1 - 1e-4 -> 1.0
fp16    : 1 - 3e-4 -> 0.9995
fp32    : 1 - 1e-3 -> 0.999
```

確かに、10^{-3} のステップでは bf16 の重みは 1 に張り付いたままであるのに対し、2×10^{-3} では 0.00195 の閾値を超えて隣接値 $1 - 2^{-8} = 0.99609375$ へ跳ぶ。同様に、fp16 は 10^{-4} を飲み込んでしまうが、3×10^{-4} に対しては $1 - 2^{-11} \approx 0.9995$ へ進み、fp32 は更新をそのまま保持する。

一方、勾配が直面するのはアンダーフローの閾値である。fp16 では、6.1×10^{-5} 未満の勾配は精度を失い、3×10^{-8} 未満の勾配はゼロに消える。逆伝播の前に損失へ定数 S を掛けると、すべての勾配が S 倍に拡大されて正規化数の範囲へ引き上げられ、その後 fp32 で S で割れば本来のスケールに戻る（実践的な手順は 5.5.3.3 節で扱い、演習問題 1 では S の許容範囲を求める）。

確率的丸め（stochastic rounding）(Gupta et al. 2015) は、これとは異なる方法で更新消失のカットオフを取り除く。常に最も近いグリッド点へ丸めるのではなく、格納される値の期待値が厳密な結果に一致するよう選ばれた確率でランダムに切り上げまたは切り捨てを行う。すると、最近傍丸めでは小さすぎて消えてしまう更新であっても、多数のステップにわたって期待値の意味で蓄積されるようになる（演習問題 5）。

35.8.2 安定なソフトマックスと交差エントロピー

ソフトマックス、log-sum-exp、交差エントロピーはいずれもロジットの指数をとるため、先ほど計算したオーバーフローとアンダーフローの閾値に正面から突き当たる。議論を具体的に保つため、本項を通して 1 つの通し例を用いる。ロジット $\mathbf{z} = (0, t)$ を持ち、正解ラベルが起こりにくい方のクラス 0 である 2 クラス分類器である。その確率は $\mathrm{softmax}(\mathbf{z}) = \big(1/(1+e^{t}),\, e^{t}/(1+e^{t})\big)$、log-sum-exp は $\log(1+e^{t})$ であり、クラス 0 に対する交差エントロピー損失（cross-entropy loss）は $\log(1+e^{t}) \approx t$（大きな t のとき）となる。これらの出力自体はいずれも極端な値ではなく、$t = 100$ であっても確率は 3.7×10^{-44} と 1、損失は単に 100 である。

35.8.2.1 最大値シフト

ロジット $\mathbf{z} \in \mathbb{R}^{n}$ が与えられたとき、ソフトマックスベクトルは次の式で定義される。

$$\mathrm{softmax}(\mathbf{z})_{i} = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}.$$

$t = 100$ でこの式をそのまま評価すると e^{100} が作られ、fp32 のオーバーフロー上限である 88.72 を超えてしまう。第 2 成分は `inf/inf`（`NaN`）になり、第 1 成分は $1/\infty = 0$ へ潰れる。真の確率自体は fp32 に余裕をもって収まるのであり、収まらないのは中間の指数関数だけであることに注目しよう。

幸い、分子と分母に同じ正の定数 e^{-c} を掛けても比は変わらないため、ソフトマックスは**シフト不変**（shift-invariant）である。

$$\mathrm{softmax}(\mathbf{z} - c\mathbf{1}) = \mathrm{softmax}(\mathbf{z}) \qquad \text{for every } c \in \mathbb{R}. \tag{35.98}$$

対数の中の和から e^{c} を括り出すと、ソフトマックスの分母の対数である **log-sum-exp** $\mathrm{lse}(\mathbf{z}) = \log \sum_{j} e^{z_j}$ についても対になる恒等式が得られる。

$$\mathrm{lse}(\mathbf{z}) = c + \log \sum_{j=1}^{n} e^{z_j - c}. \tag{35.99}$$

どちらの恒等式も任意の $c \in \mathbb{R}$ に対して成り立ち、$c = \max_{j} z_j$ と選べば両方の計算が安全になる。シフト後の指数 $z_j - c$ はすべて高々 0 となるため、どの指数関数もオーバーフローしない。同時に最大の項が $e^{0} = 1$ に等しくなるため、和は必ず $[1, n]$ に収まり、その対数が $-\infty$ へアンダーフローすることもない。これを fp32 の $(0, 100)$、冒頭の fp64 の $(1000, 1001, 1002)$、そして両方式とも有限に留まる $(0, 10)$ で試してみよう。最後のケースではビット単位まで完全に同じ値が返るだろうか。

```python
def softmax_naive(z):
    e = np.exp(z)
    return e / e.sum()

def softmax_stable(z):
    e = np.exp(z - z.max())        # shift by the max: largest exponent is 0
    return e / e.sum()

z = np.array([0.0, 100.0], dtype=np.float32)
with np.errstate(over='ignore', invalid='ignore'):
    print('naive  softmax(0, 100):', softmax_naive(z))
print('stable softmax(0, 100):', softmax_stable(z))
z = np.array([1000.0, 1001.0, 1002.0])       # the opening's logits, fp64
with np.errstate(over='ignore', invalid='ignore'):
    print('naive  softmax(1000, 1001, 1002):', softmax_naive(z))
print('stable softmax(1000, 1001, 1002):', softmax_stable(z).round(4))
z = np.array([0.0, 10.0], dtype=np.float32)
a, b = softmax_naive(z), softmax_stable(z)
print('softmax(0, 10), naive minus stable, in ulps:',
      (a - b) / np.spacing(b))
```

```
naive  softmax(0, 100): [ 0. nan]
stable softmax(0, 100): [3.8e-44 1.0e+00]
naive  softmax(1000, 1001, 1002): [nan nan nan]
stable softmax(1000, 1001, 1002): [0.09   0.2447 0.6652]
softmax(0, 10), naive minus stable, in ulps: [2. 0.]
```

$t = 100$ では、シフト版ソフトマックスは $(3.8 \times 10^{-44}, 1)$（真の 3.7×10^{-44} が有効ビット 5 ビットを残して非正規化数のグリッドに丸められた値）を返すのに対し、直接評価する式は 0 と NaN を返す。精度を高くしても問題は解決しない。冒頭のロジット $(1000, 1001, 1002)$ は fp64 でさえ e^{1000} がオーバーフローし、直接評価する式は 3 つの NaN になるが、シフト版は問題なく $(0.09, 0.24, 0.67)$ を返す。$t = 10$ では、実数上の恒等式は丸めの順序まで一致することを保証しないため、2 つの式は最下位ビットまで一致するか、あるいは最終桁単位（unit in the last place、**ulp**）で 1〜2 単位だけ異なる。なお、境界ケースが 1 つ残る。最大値シフトは非有限のロジットを救うことはできず、行全体が $-\infty$ であると $c = -\infty$ となって $z_j - c$ が未定義になる。

35.8.2.2 log-sum-exp の上下界

log-sum-exp はあらゆる指数分布族（exponential family）モデルの対数正規化項であり（36.4 節）、1 回の減算を除けば交差エントロピー損失そのものである。シフト恒等式 Equation 35.99 から、この値が最大のロジットにどれほど近いかが直ちにわかる。

命題（log-sum-exp のサンドイッチ不等式）。 任意の $\mathbf{z} \in \mathbb{R}^n$ に対して次が成り立つ。

$$\max_j z_j \;\le\; \mathrm{lse}(\mathbf{z}) \;\le\; \max_j z_j + \log n.$$

証明。 $c = \max_j z_j$ を Equation 35.99 に代入する。各指数関数 $e^{z_j - c}$ は $(0, 1]$ に属し、最大値を達成する項は 1 に等しいため、和は $[1, n]$ に、その対数は $[0, \log n]$ に属する。これに c を加えれば上下両側の評価が得られる。■

したがって log-sum-exp は**ソフト最大値**（soft maximum）である。2 クラスの通し例では、

$\mathrm{lse}(0,t)=\log(1+e^{t})$ は $\log 2\approx 0.69$ 以内の差で $\max(0,t)$ に追随し、2 つのロジットが並ぶ $t=0$ で最大の差に達し、$|t|$ が大きくなるにつれて最大値にぴたりと漸近する（Figure 35.31）。

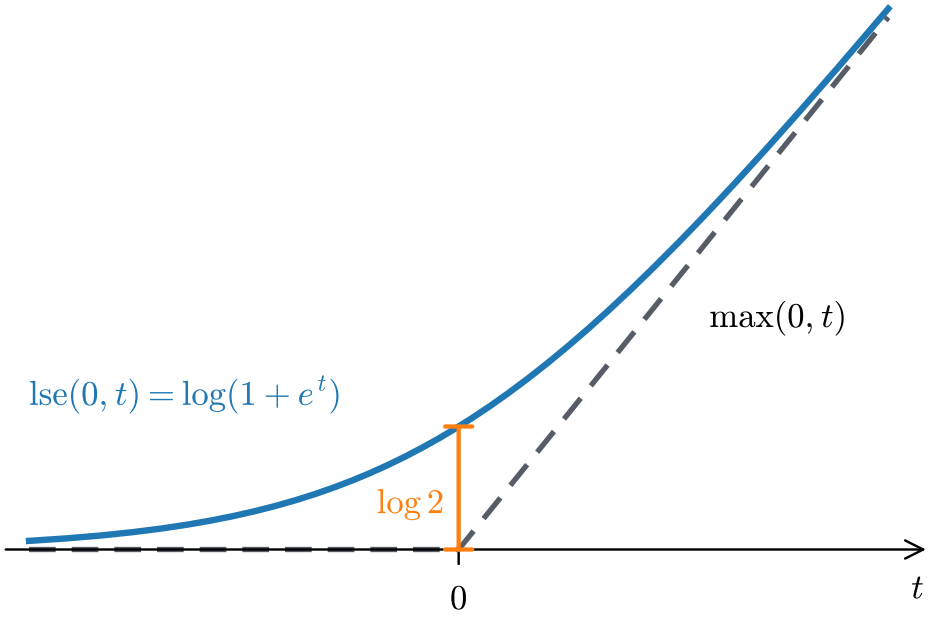


Figure 35.31: ソフト最大値としての log-sum-exp。$\mathrm{lse}(0,t)=\log(1+e^{t})$ は $\log 2$ 以内の差で $\max(0,t)$ に収まり、$n=2$ におけるサンドイッチ不等式と一致する。差は 2 つのロジットが並ぶ $t=0$ で最大となり、$|t|$ が大きくなるにつれて消える。

さらに、その勾配はソフトマックスそのもの、すなわち $\partial\,\mathrm{lse}(\mathbf{z})/\partial z_i = e^{z_i}/\sum_j e^{z_j} = \mathrm{softmax}(\mathbf{z})_i$ である。ラベル y に対するロジットに関する交差エントロピーの勾配が $\mathrm{softmax}(\mathbf{z})-\mathbf{e}_y$ へと簡約されるのはこのためであり、また 35.3 節では log-sum-exp が凸関数であることも示される。

シフトしたロジットからシフト後の log-sum-exp を引くと**対数ソフトマックス**（log-softmax）が得られ、確率そのものを一度も作らずに対数確率（log probability）を直接計算できる。

$$\log\mathrm{softmax}(\mathbf{z})_i = (z_i - c) - \log\sum_{j=1}^{n} e^{z_j-c}, \qquad c=\max_j z_j. \tag{35.100}$$

最終結果の指数をとらないため、微小な確率が対数をとる前にゼロへアンダーフローすることはない。ここで Equation 35.100 の微妙な点に注目しよう。代数的に同値な式 $z_i-\mathrm{lse}(\mathbf{z})$ は、まず大きな数 $\mathrm{lse}=c+\log\sum_j e^{z_j-c}$ を作ってからそれを大きな数 z_i から引くため、浮動小数点では目立って精度が落ちる。この括弧の付け方によってどれほどの差が生じるだろうか。ロジット $(1000, 1001, 1002)$ に対して fp32 で両方の形を比較してみよう。

```
z = np.array([1000.0, 1001.0, 1002.0], dtype=np.float32)
c = z.max()
s = np.log(np.exp(z - c).sum())          # log sum exp(z - c), in [0, log 3]
```

```
    with np.errstate(over='ignore'):
        print('naive log(sum(exp(z))):', np.log(np.exp(z).sum()))
    lse = c + s
    print('lse = c + s           :', lse, '  float32 gap at 1002:', np.spacing(c))
    log_p_sub = z - lse                       # forms lse, then subtracts it
    log_p = (z - c) - s                       # never forms c + s
    print(f'sum of exp(z - lse)    : {np.exp(log_p_sub).sum():.8f}')
    print(f'sum of exp((z - c) - s): {np.exp(log_p).sum():.8f}')
```

```
naive log(sum(exp(z))): inf
lse = c + s           : 1002.4076   float32 gap at 1002: 6.1035156e-05
sum of exp(z - lse)    : 1.00001323
sum of exp((z - c) - s): 1.00000000
```

直接計算は `inf` へオーバーフローする。$1000 > 709.78$ であるため fp64 でも同様であり、冒頭で `inf/inf` が 3 つの `NaN` になった原因もまさにこれである。シフトを用いた値 $\mathrm{lse} = 1002.4076$ は、サンドイッチ区間 $[1002, 1002 + \log 3]$ の内側に問題なく収まっている。さらに興味深いことに、対数ソフトマックスの 2 つの式は食い違う。$z_i - \mathrm{lse}$ の指数をとって得た確率の和は 1.0000132 になるのに対し、Equation 35.100 では和が 1.0000000 になる。

なぜ $z_i - \mathrm{lse}$ は精度を失ったのだろうか。意外なことに、減算自体は一切の誤差を生んでいない。2 つの浮動小数点数が互いに 2 倍以内の範囲にあるとき、両者はともに小さい方の数のグリッド間隔の整数倍であり、その差を表すのに余分なビットを必要としないため、差は厳密に表現可能だからである（Sterbenz の補題、演習問題 6）。

誤差が混入したのはそれより前、lse を格納した時点である。1002 付近では fp32 の数は $2^{-14} \approx 6.1 \times 10^{-5}$ 間隔で並んでいるため、真の値 1002.4076060 は 1.32×10^{-5} だけ切り捨てられる。その結果、$z_i - \mathrm{lse}$ として計算されたすべての対数確率は 1.32×10^{-5} だけ大きくなりすぎ、得られる各確率は $e^{1.32 \times 10^{-5}}$ 倍に膨らみ、それらの和は 1 を 1.32×10^{-5} だけ上回る。指数の絶対誤差は指数関数を通すと同程度の相対誤差に変わり、減算は入力にすでに潜んでいた誤差を露呈させたにすぎない。これが 35.8.3 節の中心テーマである。

35.8.2.3 ロジットからの交差エントロピー

ラベル y に対する交差エントロピー損失は負の対数ソフトマックスにほかならないため、Equation 35.100 からロジットを用いて直接表せる。

$$-\log \mathrm{softmax}(\mathbf{z})_y = \mathrm{lse}(\mathbf{z}) - z_y. \tag{35.101}$$

この方法で損失を計算すれば、必要なのは安定な log-sum-exp が 1 回と減算が 1 回だけであり、確率を一度も作る必要がない。通し例では、クラス 0 に対する損失は $\mathrm{lse}(0, t) - 0 = \log(1 + e^t)$ となる。これに対し、先に確率を計算すると、正解クラスの確率 $e^{-t}/(1 + e^{-t}) \approx e^{-t}$ は $t > 87.3$ で fp32 の正規化数の範囲を下回り、$t = 104$ 付近で完全にゼロへアンダーフローして、その負の対数を ∞ へ飛ばしてしまう。この崖を跨ぐ $t = 103$ と 104 を含めて、ライブラリの組み込み関数で 2 つの経路を比較してみよう。

```
    print('    t  CE from logits    CE via probabilities')
    for t in [20.0, 60.0, 103.0, 104.0]:
        logits = torch.tensor([[0.0, t]])
        y = torch.tensor([0])                       # label = the unlikely class
```

```
    from_logits = F.cross_entropy(logits, y)
    probs = F.softmax(logits, dim=1)            # stable softmax, then...
    from_probs = -torch.log(probs[0, y])        # ...take the log yourself
    print(f'{t:5.0f}  {from_logits.item():15.4f}  {from_probs.item():15.4f}')
```

```
    t  CE from logits    CE via probabilities
   20          20.0000              20.0000
   60          60.0000              60.0000
  103         103.0000             103.2789
  104         104.0000                  inf
```

ロジットから直接計算すると、どの行でも損失は t に等しくなり、確率を経由する計算も $t = 20$ と 60 ではこれと一致する。しかし $t = 103$ では確率 e^{-103} が非正規化数の範囲に落ち込み、その先の振る舞いはライブラリが非正規化数をどう扱うかによって分かれる。

PyTorch は非正規化数を保持するため、$e^{-103} \approx 1.32 \times 2^{-149}$ は fp32 の最小の非正規化数である 2^{-149} へ丸められる。有効ビットが 1 ビットしか残らないため、損失は $149 \log 2 = 103.2789$ となり、何の警告もないまま小数第 1 位から誤った値になる。$t = 104$ では $e^{-104} < 2^{-150}$ が 0 へ丸められ、損失は `inf` になる。

実践上の教訓は明快である。実際に確率が必要になるまでは、損失、尤度（likelihood）、事後分布（posterior）をすべて対数空間に保っておくことである。3.3.5.2 節の融合損失が Equation 35.101 を実装しているのもこのためであり、37.1 節では交差エントロピーが何を測っているかを解釈し、36.11 節では同じアンダーフローを避けるために画素ごとの対数確率を数百個足し合わせている。

35.8.3 桁落ち

35.8.3.1 関数の条件数

fp32 では、$\log(1 + x)$ を $x = 10^{-8}$ で評価すると 0（相対誤差 100%）が返るのに対し、ライブラリ関数 `log1p` は 10^{-8} を返す。$\log(1 + x)$ は $x = 0$ 付近で本質的に計算が難しいのだろうか、それとも式の形に非があるのだろうか。こうした問いに体系的に答えるには、特定のアルゴリズムに依らず、関数が入力の誤差をどれほど強く増幅するかを測る量が必要である。

微分可能な関数 f の、点 x（ただし $f(x) \neq 0$）における**相対条件数**（relative condition number）とは、x に対する微小な相対摂動が $f(x)$ の相対摂動へと何倍になって現れるかを表す係数である。x を $x(1 + \delta)$ へ摂動させると、出力は一次近似で $f(x(1 + \delta)) \approx f(x) + x f'(x)\,\delta$ へ変化する。この変化量を $f(x)$ で割ると、次の増幅率が得られる。

$$\kappa_f(x) = \left| \frac{x\, f'(x)}{f(x)} \right|. \tag{35.102}$$

たとえば指数関数では $\kappa_f(x) = |x|$ であり、$x = 88$ において丸めの単位 1 つ分の入力誤差は出力で 88 単位分の誤差になる。これは先ほど対数ソフトマックスのコードセルで見た仕組みそのものであり、指数の絶対誤差が確率における同程度の相対誤差に変わった理由である。なお、κ_f は厳密な数学関数の一次の性質である。f をどのように評価するかについては何も語らず、また $f(x) = 0$ となる点では相対誤差が定義されないため破綻する。

減算は 2 つの入力をとり、その両方に同じ一次の評価が成り立つ。$\hat{a} = a(1+\delta_a)$ と $\hat{b} = b(1+\delta_b)$ が相対誤差 $|\delta_a|, |\delta_b| \le u$ を持ち、減算自体は厳密であるとする。このとき $\hat{a} - \hat{b} = (a - b) + (a\delta_a - b\delta_b)$ であるから、三角不等式（triangle inequality）より差の相対誤差は次のように抑えられる。

$$\frac{|(\hat{a}-\hat{b})-(a-b)|}{|a-b|} = \frac{|a\delta_a - b\delta_b|}{|a-b|} \le \frac{|a|+|b|}{|a-b|}\,u. \tag{35.103}$$

係数 $(|a|+|b|)/|a-b|$ は減算の条件数であり、$a \approx b$ となると急激に増大する。すでに丸め誤差を含んでいるほぼ等しい 2 つの数を引く現象は**桁落ち**（catastrophic cancellation）と呼ばれる。正確な上位桁同士が打ち消し合い、下位桁にあったノイズが答えの上位桁へと繰り上がるのである。参考として、代表的な問題の条件数を以下の表にまとめる。

問題	κ_f	性質・領域
$a-b$	$(\lvert a\rvert+\lvert b\rvert)/\lvert a-b\rvert$	$a \approx b$ のとき大
$\log(1+x)$	$x/((1+x)\log(1+x))$	1.000（$x=10^{-8}$）、0.82（$x=0.5$）
e^x	$\lvert x\rvert$	大きな $\lvert x\rvert$ で大
$1-\cos x$	$x\sin x/(1-\cos x)$	2 に収束（$x\to 0$）
$\sqrt{x}$	$1/2$	常に一定
データの分散（平均 m、ばらつき s）	高々 $2(1+\lvert m\rvert/s)$	後述

2 行目から冒頭の問いに決着がつく。$\log(1+x)$ は $x=0$ 付近で完全に良条件である（$\kappa_f \approx 1$）。素朴なアルゴリズム `log(1 + x)` が破綻するのは、悪条件な中間ステップへ迂回するためである。この式はまず $y = 1+x$ を作り、続いて $\log y$ を $y \approx 1$ で評価するが、そこでは対数関数が $\kappa_{\log}(y) = 1/|\log y| \approx 1/x = 10^8$ を持つ。$1+x$ を丸めることでオーダー u の相対誤差が y に生じ、それを $\log$ が 10^8 倍に拡大するため、fp32 では 100% を超える誤差になる。対照的に、`log1p` は $1+x$ の丸め誤差が結果に入り込まないよう書かれており、ごく小さい x に対しては単に x を返す。

ここから本節全体を貫く基本原理が得られる。すなわち、**不安定なアルゴリズムとは、良条件な問題を悪条件な中間ステップへ迂回させるものである。** κ_f が小さいにもかかわらず計算結果が悪いなら、原因はアルゴリズムにあるので式を書き換えればよい。逆に κ_f 自体が大きいなら、いかなるアルゴリズムでも答えを救うことはできず、代わりに問題そのものを変えなければならない。これが 35.8.4 節の主題である。

35.8.3.2 ほぼ等しい数の減算

これをコードで確かめよう。以下のセルでは $\log(1+x)$ の実験を再現したあと、10 進法の先頭 7 桁が一致する 2 つの fp32 の数 1.0002344 と 1.0002341（真の差は 3.0×10^{-7}）を減算する。

```
x = np.float32(1e-8)
print('float32 rounds 1 + x to:', np.float32(1.0) + x)
print('log(1 + x) =', np.log(np.float32(1.0) + x), '   log1p(x) =', np.log1p(x))
a, b = np.float32(1.0002344), np.float32(1.0002341)
print(f'stored a = {float(a):.10f}, b = {float(b):.10f}')
print('a - b =', a - b, '=', (a - b) / np.spacing(np.float32(1.0)), 'ulps of 1')
print('float64: log(1 + 1e-20) =', np.log(1 + 1e-20),
      '  log1p(1e-20) =', np.log1p(1e-20))
```

```
float32 rounds 1 + x to: 1.0
log(1 + x) = 0.0    log1p(x) = 1e-08
stored a = 1.0002343655, b = 1.0002341270
```

```
a - b = 2.3841858e-07 = 2.0 ulps of 1
float64: log(1 + 1e-20) = 0.0   log1p(1e-20) = 1e-20
```

1 と 2 の間では、fp32 におけるあらゆる差は $\varepsilon_{\text{mach}} = 1.19 \times 10^{-7}$ の整数倍であり、真の差 3.0×10^{-7} はそのグリッド上に乗らない。減算の結果は 2 ulp にあたる 2.38×10^{-7}（20% 小さい）となったが、今回も減算自体は誤差を一切生んでいない。2 つの入力は 1.0002343655 と 1.0002341270 として格納されており、セルが出力したのはそれらの厳密な差であって、誤差は格納された入力に最初から焼き込まれていたのである。ここでの桁落ち係数 Equation 35.103 は $2.0005/(3 \times 10^{-7}) \approx 7 \times 10^{6}$ である。これに $u \approx 6 \times 10^{-8}$ を掛けると最大 0.4 の相対誤差が許容され、観測された 0.2 はまさにその上界の内側に収まっている。

同じ桁落ちは、$\log(1+x)$ や $e^{x}-1$ の 0 付近（`log1p` と `expm1` がこれを回避する）、$1-\cos x$ の 0 付近（$2\sin^2(x/2)$ と書き換えられる）、および二次方程式の解の公式で $b^2 \gg 4|ac|$ のとき（演習問題 2）にも現れる。これらのどのケースでも、代数的な書き換えによって問題となる減算を取り除くことができる。

減算を消去できず妥協を受け入れるしかない場面が有限差分（finite difference）である。数値勾配チェックでは、自動微分（automatic differentiation）を前進差分 $(f(x+h)-f(x))/h$ と比較する。ステップ幅 h はどれほど小さくすべきだろうか。テイラーの定理より、数学的な打ち切り誤差（truncation error）は約 $\frac{h}{2}|f''(x)|$ であり、$h \to 0$ とともに縮小する。一方、各関数評価にはおよそ $u|f(x)|$ の丸め誤差が伴い、その差を h で割ると誤差は約 $2u|f(x)|/h$ に拡大され、$h \to 0$ とともに発散する。総誤差が最小になるのはこれら 2 項が釣り合う次の点である。

$$h^{\star} = 2\sqrt{\frac{u\,|f(x)|}{|f''(x)|}}, \qquad \text{with error about } 2\sqrt{u\,|f(x)f''(x)|}.$$

最適な誤差は $\sqrt{u}$ に比例するため、最良の前進差分であっても形式が持つ桁数の約半分しか維持できない。$f = \sin$、$x = 1$ の場合、ここから fp64 では $h^{\star} \approx 2 \times 10^{-8}$（誤差 1.8×10^{-8}）、fp32 では $h^{\star} \approx 5 \times 10^{-4}$ と予測される。両方の形式で h を 10^{-2} から 10^{-12} までスイープし、このトレードオフを直接確認しよう。

```
print('      h   error in fp64   error in fp32')
for h in 10.0 ** -np.arange(2, 13):
    err = [abs(float((np.sin(t(1) + t(h)) - np.sin(t(1))) / t(h)) - np.cos(1.0))
           for t in (np.float64, np.float32)]  # forward difference of sin at 1
    print(f'{h:7.0e}  {err[0]:14.1e}  {err[1]:14.1e}')
```

```
      h   error in fp64   error in fp32
  1e-02         4.2e-03         4.2e-03
  1e-03         4.2e-04         4.0e-04
  1e-04         4.2e-05         2.8e-04
  1e-05         4.2e-06         3.9e-03
  1e-06         4.2e-07         6.3e-02
  1e-07         4.2e-08         5.6e-02
  1e-08         3.0e-09         5.4e-01
  1e-09         5.3e-08         5.4e-01
  1e-10         5.8e-08         5.4e-01
  1e-11         1.2e-06         5.4e-01
  1e-12         4.3e-05         5.4e-01
```

fp64 では、誤差は h に比例して 4×10^{-3}（打ち切り誤差が支配的な $h = 10^{-2}$）から約 10^{-8}

($h = 10^{-8}$ 付近；個々の h における正確な値は丸めノイズで変動する）まで減少し、その後は桁落ちが支配的になって 4×10^{-5}（$h = 10^{-12}$）へと再び上昇する。fp32 では、$h = 10^{-3}$〜10^{-4} にある最適領域でも正しい桁は約 3 桁しか残らず、$h = 10^{-8}$ 以下では $1 + h$ がそのまま 1 に丸められて差分商が 0 となり、誤差は導関数（derivative）全体の値 $\cos 1 = 0.54$ に一致してしまう。数値勾配チェックを常に fp64 で実行すべき理由はここにある。

また、精度を高くしても桁落ちは決して治らず、破綻が起きる入力がより小さい領域へ押しやられるだけであることにも注意しよう。先ほどの `log1p` のセルの最終行では、fp64 でも `log(1 + 1e-20)` が 0 を返し（fp32 で 10^{-8} のときに起きたのと同様）、一方で `log1p` は両方の形式で正確さを保っていた。式を定式化し直すことで、悪条件なステップそのものが取り除かれるのである (Higham 2002)。

35.8.3.3 1 パス分散

桁落ちは、一見まったく無害に見える公式の中にも潜んでいる。標本分散（sample variance）の計算はその古典的な例であり、正規化層やデータ前処理パイプラインが絶えず分散を計算する深層学習にも直結する問題である。

データ $x_1, \dots, x_N$ が平均 m と標準偏差 s を持つとき、教科書でおなじみの次の恒等式がある。

$$\frac{1}{N}\sum_{i=1}^{N}(x_i - m)^2 = \frac{1}{N}\sum_{i=1}^{N} x_i^2 - m^2,$$

この式を見ると、x_i と x_i^2 の和だけを累積して 1 パスで分散を計算したくなる。厳密な算術ではこの恒等式に何の欠陥もないが、浮動小数点では罠になる。$s \ll |m|$ のとき、右辺の 2 項はいずれも m^2 付近にある一方でその差は s^2 であるから、桁落ち係数 Equation 35.103 はおよそ $2m^2/s^2$ になる。$m = 10^9$、$s = 1$ の場合、この係数は 2×10^{18} に達し、fp64 が吸収できる 10^{16} を 2 桁も上回る。

この精度喪失のうち、問題自体によって避けられない部分はどれだけだろうか。分散の条件数を調べてみよう。各入力 x_i を相対誤差 δ_i（$|\delta_i| \le u$）で摂動させ、x_i を $x_i + x_i\delta_i$ に置き換える。x_i に関する分散の微分は $\frac{2}{N}(x_i - m)$ となる。誘導される m の変化には $\sum_i (x_i - m) = 0$ が掛かるからである。$x_i = (x_i - m) + m$ と書けば、分散の一次の変化量は次のようになる。

$$\frac{2}{N}\sum_{i=1}^{N}(x_i - m)\, x_i\delta_i = \frac{2}{N}\sum_{i=1}^{N}(x_i - m)^2\delta_i + \frac{2m}{N}\sum_{i=1}^{N}(x_i - m)\,\delta_i.$$

第 1 の和の大きさは高々 $2s^2 u$ である。コーシー・シュワルツの不等式より、第 2 の和の大きさは高々 $2|m|u \cdot \frac{1}{N}\sum_i |x_i - m| \le 2|m| s\, u$ である。s^2 で割ると、分散の相対条件数は高々 $2(1 + |m|/s) \approx 2|m|/s$、今の例では約 2×10^9 であることがわかる（$\delta_i = u\,\mathrm{sign}(x_i - m)$ と選べば、ガウス分布に従うデータでは $1.6\,|m|/s$ となる）。したがって問題自体が許容する誤差は $2 \times 10^9 u \approx 2 \times 10^{-7}$ にすぎず、約 7 桁の精度が残るはずである。これに対し、素朴な 1 パス公式はオーダー $(m/s)^2 u$ の誤差を被る。35.8.4 節で正規方程式が行列の条件数を 2 乗するのと同様に、この公式は条件数を **2 乗**してしまうのである。

安定な 1 パスのアルゴリズムは、すべての偏差をスケール s に保ち、m^2 のスケールには持ち込まない。Welford (1962) は逐次平均 m_k と**中心化**（centered）二乗の逐次和 $M_k = \sum_{i \le k}(x_i - m_k)^2$ を保持し、新しい各サンプルと直前の平均との偏差 $d_k = x_k - m_{k-1}$ から両者を次のように更新する。

$$m_k = m_{k-1} + \frac{d_k}{k}, \qquad M_k = M_{k-1} + d_k\,(x_k - m_k). \tag{35.104}$$

M_k の更新における非対称性に注目しよう。**古い**平均からの偏差と、**新しい**平均からの偏差を掛け合わせている。まさにこの非対称性によって、漸化式が厳密に成り立つのである。

命題（Welford の漸化式）。 $m_0 = M_0 = 0$ **とするとき、漸化式 Equation 35.104 はすべての** $k \ge 1$ **に対して厳密な算術のもとで次を満たす。**

$$m_k = \frac{1}{k}\sum_{i=1}^{k} x_i, \qquad M_k = \sum_{i=1}^{k} (x_i - m_k)^2.$$

証明。 平均の恒等式 $k\,m_k = (k-1)\,m_{k-1} + x_k$ は第 1 の漸化式から直ちに導かれる。M_k については k に関する帰納法で示す。$m_k - m_{k-1} = d_k/k$ および $x_k - m_k = d_k\,(k-1)/k$ を用い、第 k 項を切り離して最初の $k-1$ 項を m_{k-1} の周りで中心化し直す。

$$\begin{aligned}\sum_{i=1}^{k}(x_i - m_k)^2 &= \sum_{i=1}^{k-1}\Big((x_i - m_{k-1}) - \tfrac{d_k}{k}\Big)^2 + (x_k - m_k)^2 \\ &= M_{k-1} + (k-1)\tfrac{d_k^2}{k^2} + \tfrac{(k-1)^2}{k^2} d_k^2 .\end{aligned}$$

交差項は $\sum_{i\le k-1}(x_i - m_{k-1}) = 0$ により消え、帰納法の仮定より最初の和は M_{k-1} に一致する。残る 2 項をまとめると $\tfrac{k-1}{k}\,d_k^2 = d_k \cdot d_k \tfrac{k-1}{k} = d_k(x_k - m_k)$ となり、漸化式が加える項とちょうど一致する。$\blacksquare$

Welford の漸化式が浮動小数点でも精度を保つことの証明はより難しく、ここでは直感的なあらすじのみを述べる（完全な解析は Higham (2002) を参照）。一見すると、x_k と m_{k-1} はともに 10^9 付近にあるため、Welford の更新でも大きな数同士を減算しているように見える。しかし Sterbenz の補題によりこの減算は厳密であり、x_k も厳密な入力データであるため、d_k に含まれる誤差は逐次平均に由来するものだけ、すなわち $|m|$ のスケールで 10^5 ステップにわたり高々数十 ulp にとどまる。したがって M_k に加えられる量はすべてスケール s の偏差となり、更新時の丸め誤差は s^2 に対する相対誤差として入り、m^2 に対するものにはならない。平均に一定のオフセット ε が加わっても、中心化二乗和のずれは $N\varepsilon^2$ にとどまる。一次の項には $\sum_i (x_i - m) = 0$ が掛かるからである。実際、Welford 法の誤差は問題自体の感度 $u|m|/s \approx 1.1 \times 10^{-7}$ に達するのに対し、素朴な 1 パス公式は $(m/s)^2 u \approx 100$ もの誤差を被る。

素朴な 1 パス法、Welford 法、2 パスの基準値という 3 つの手法すべてを、10^5 個の fp64 サンプル（平均 10^9、分散 1）で比較してみよう。

```
rng = np.random.default_rng()
x = 1e9 + rng.normal(0.0, 1.0, size=100_000)    # mean 1e9, variance 1

naive = (x**2).mean() - x.mean()**2             # one pass, cancels
two_pass = ((x - x.mean())**2).mean()           # subtract the mean first

m, M = 0.0, 0.0                                 # Welford: one pass
for k, xk in enumerate(x, start=1):
    d = xk - m
    m += d / k                                  # m_k
```

```
    M += d * (xk - m)                          # M_k
welford = M / len(x)

print(f'naive E[x^2] - E[x]^2 : {naive:12.6f}   (= {naive / 128:.0f} x 128)')
print(f'Welford, one pass     : {welford:12.6f}')
print(f'two-pass reference    : {two_pass:12.6f}')
print(f'Welford vs two-pass   : {abs(welford - two_pass) / two_pass:.1e} relative')
print('float64 gap near 1e18 :', np.spacing(1e18))
```

```
naive E[x^2] - E[x]^2 :  -128.000000   (= -1 x 128)
Welford, one pass     :     0.999287
two-pass reference    :     0.999287
Welford vs two-pass   : 6.8e-09 relative
float64 gap near 1e18 : 128.0
```

素朴な 1 パス推定値は厳密に 128 の整数倍（典型的には 0、± 128、± 256）として出力され、真の分散 1 には遠く及ばず、頻繁に負の値にさえなる。なぜ 128 なのだろうか。$\frac{1}{N}\sum x_i^2$ と m^2 はともに $[2^{59}, 2^{60})$ に属し、そこでは隣り合う fp64 の数が $2^{59-52} = 128$ 離れているため、両者の差は 128 の倍数にしかなりえない。予測された約 100 の誤差がグリッドに量子化された姿である。一方、Welford の漸化式は 2 パスの基準値と少なくとも 7 桁一致し、逆符号の丸め誤差が和全体で部分的に打ち消し合うため 8〜9 桁一致することも多い。

データセット全体のストリーミング統計量を計算する場合など、1 パスで分散を累積する場面では常に同じ注意が必要になる。（なお、バッチ正規化が推論用に保持する移動モーメントは指数移動平均（exponential moving average、7.3 節）であり、大きな逐次和同士を減算することはない。）

35.8.3.4 総和の順序

素朴な分散も 2 パスの基準値も 10^5 個の数を足し合わせるが、浮動小数点では加算の順序が結果に影響する。各加算は $(1+\delta_k)$ の因子を導入するため、単純な左から右への総和では最初の項が $n-1$ 回すべての丸めを通過する。u の一次までで、左から右への総和は次を満たす。

$$\left|\mathrm{fl}\Big(\sum_{i=1}^{n} x_i\Big) - \sum_{i=1}^{n} x_i\right| \le (n-1)\,u \sum_{i=1}^{n} |x_i|.$$

初期の項がその後のすべての加算の丸め誤差を蓄積するため、最悪ケースの誤差は n に対して線形に増大する。**ペアワイズ**総和法は配列の前半と後半を再帰的に足し合わせることでこれを回避し、どの項も $\log_2 n$ 回の加算しか通過しないため、上界は $u \log_2 n \sum_i |x_i|$ へと縮小する。NumPy の `sum` はこの考え方のブロック版を用いている。Kahan の補償付き総和法はさらに一歩進め、各加算の丸め誤差を逐次補正変数に保持することで n に依存しない誤差上界を達成する（演習問題 8）。10^7 個の fp32 の 0.1（格納されている値は 0.100000001490 ...）を左から右への順序とペアワイズの両方で足し合わせてみよう。ペアワイズ用の配列はゼロでパディングして 2^{24} 要素にし、隣り合うペアの加算を繰り返す。

```
x = np.full(10_000_000, 0.1, dtype=np.float32)
exact = len(x) * float(x[0])                # x[0] = 0.100000001490...
s = np.concatenate([x, np.zeros(2**24 - len(x), np.float32)])  # pad to 2^24
while len(s) > 1:                           # pairwise: add neighbors, halve
    s = s[0::2] + s[1::2]
```

```
    print(f'in order (cumsum): {np.cumsum(x)[-1]:16.4f}')
    print(f'pairwise         : {s[0]:16.4f}')
    print(f'exact            : {exact:16.4f}')
```

```
in order (cumsum):     1087937.0000
pairwise         :     1000000.0000
exact            :     1000000.0149
```

左から右への総和は $1{,}087{,}937$ で終わり、上界 $(n-1)u\sum_i |x_i| \approx 6\times 10^5$ の内側にはあるものの、ほぼ 9% も大きすぎる。ペアワイズ総和法は $1{,}000{,}000$ に着地し、厳密な和との差は 0.015 以内であり、その上界 $u\log_2 n\sum_i |x_i| \approx 1.4$ の十分内側に収まっている。

左から右への総和におけるドリフトの犯人は、微小な重み更新を消し去ったのと同じグリッド効果である。各被加数は、その時点の逐次和のグリッド間隔へ丸められるのである。和が $2^{19}=524{,}288$ を超えると fp32 の数は $1/16$ 間隔になるため、0.1 は 1.6 目盛り分に相当し、毎回の加算が 2 目盛り分の 0.125（25% 過大）へと切り上げられる。2^{20} を超えると間隔は倍の $1/8$ になり、各被加数は再び 0.125 へ丸められる。それより前の 2^{17} から 2^{19} の間では、逆に 0.09375 へ切り捨てられていた。浮動小数点の加算は結合法則を満たさないため、GPU スレッド間の並列リダクションはデバイスごと、あるいは実行ごとにビット単位まで同一の結果を生むとは限らない。幸い、Welford の漸化式もペアワイズリダクションに対応している。演習問題 7 の統合則は 2 つの部分集合の統計量を結合するものであり、分散訓練においてデバイス間で逐次モーメントを統合する際にもこの方法が用いられている。

35.8.4 条件性

行列の条件数とリッジ回帰（ridge regression）のフィルタ係数は 33.5 節で展開した。ここで新たに扱うのは、前進誤差と後退誤差、後退誤差を測る公式、そして正規方程式（normal equations）が桁数の面で支払う代償である。

35.8.4.1 ほぼ平行な 2 直線

ここまでは計算の答えが 1 つの数であった。連立一次方程式では答えはベクトルになり、33.5.3.2 節では次の連立一次方程式を取り上げた。

$$x_1 + x_2 = 2, \qquad x_1 + 1.0001\,x_2 = 2.0001,$$

この例は、そのような答えがいかに脆くなりうるかを示している。解は $(1,1)$ であるが、第 2 式の右辺を 10^{-4}（データの相対変化にして 3.5×10^{-5}）だけ増やすと、解は $(0,2)$ へ移動する。各方程式は平面上の直線であり、解は 2 直線の交点である。これら 2 直線は傾き -1 と -0.9999 を持ち、0.003° の角度で交わるため、一方をわずかに平行移動させるだけで交点が他方の直線に沿って大きく滑ってしまう（Figure 35.32）。

同じ数値を逆向きに読んでみよう。あるソルバが元の方程式に対して $\hat{\mathbf{x}} = (0,2)$ を返したとする。真の解までの距離である**前進誤差**（forward error）は 100% である。しかし $(0,2)$ は、b_2 を 10^{-4} だけ変えた近傍の問題の厳密解でもある。このようなデータの変更量のうち最小のものが**後退誤差**（backward error）である。ごく小さな後退誤差を持つソルバは近傍の問題を厳密に解いているが、それでも真の答えからは大きく離れうる。

どれほど離れうるかはソルバではなく問題の性質である。ここで再び 35.8.3.1 節の条件数が、今度は関数 $\mathbf{b}\mapsto \mathbf{A}^{-1}\mathbf{b}$ に対して現れる。すなわち、あらゆる右辺とあらゆる変化の方向にわたってとった、入力の相対変化が出力の相対変化へと拡大されうる最大の倍率である。特異値

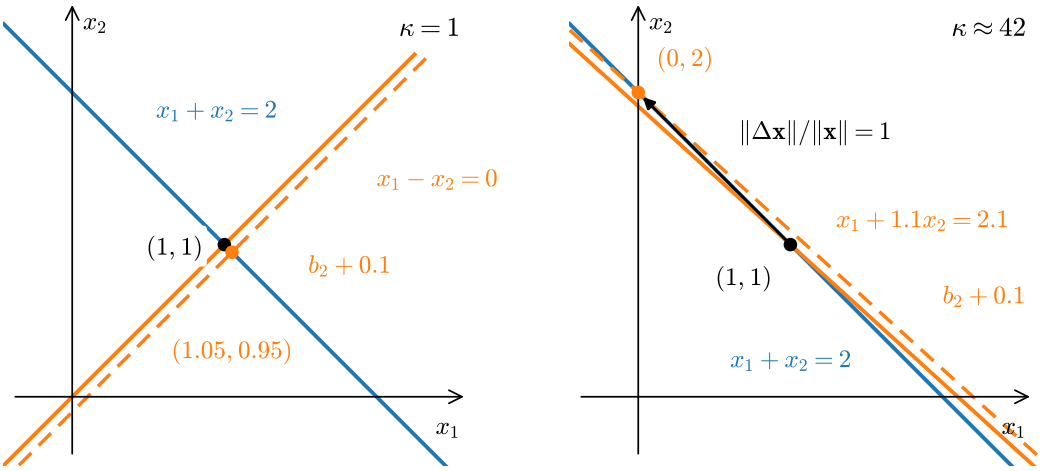


Figure 35.32: 連立一次方程式は、表す直線がほぼ平行なときに敏感になる。(a) 直交する 2 直線が $(1, 1)$ で交わる場合。第 2 の直線を 0.1 ずらすと交点は 5% 移動する（増幅率 1、$\kappa = 1$）。(b) 2 直線 $x_1 + x_2 = 2$ と $x_1 + 1.1x_2 = 2.1$ も $(1, 1)$ で交わるが、同じだけずらすと交点は $(0, 2)$ へ移動し、100% の変化が 3.4% の $\mathbf{b}$ の変化から生じる（増幅率 29、上界は $\kappa \approx 42$）。本文の方程式は 1.0001 を 1.1 の代わりに置いたものであり、なす角 $0.003°$、$\kappa \approx 4 \times 10^4$ で同じ幾何構造を持つ。

$\sigma_1 \geq \cdots \geq \sigma_d > 0$ を持つ行列では、その倍率は $\kappa(\mathbf{A}) = \sigma_1/\sigma_d$ であり（33.5.3.2 節）、先ほどの 2 直線では約 4×10^4 になる。シフト $(0, 10^{-4})$ は最も増幅される方向と一致していないため、そのうちの 2.8×10^4 倍を実現している。

35.8.4.2 後退誤差と前進誤差

連立一次方程式 $\mathbf{A}\mathbf{x} = \mathbf{b}$ と計算解 $\hat{\mathbf{x}}$ に対し、前進誤差は $\|\hat{\mathbf{x}} - \mathbf{x}\|/\|\mathbf{x}\|$ であり、後退誤差は $\hat{\mathbf{x}}$ を厳密解にする最小のデータ変化の大きさである。アルゴリズムの後退誤差が u のオーダーにあるとき、すなわち与えられた問題からデータの丸め約 1 回分だけ異なる問題の厳密解を返すとき、そのアルゴリズムは**後退安定**（backward stable）であるという。特異値分解（SVD）や QR 分解は後退安定であり、`np.linalg.solve` が用いるピボット選択付き LU 分解も実用上そうである。一方、1 パス分散は後退安定ではない。いかなるデータセットの分散も非負であるため、負の出力はいかなるデータの厳密な分散にもなりえないからである。

この定義はデータのあらゆる変化にわたる最小化を要求する。幸い、行列のみの変化に対しては残差（residual）から計算できる閉じた形の式が存在する。

命題（残差からの後退誤差）。 $\hat{\mathbf{x}} \neq \mathbf{0}$ とし、$\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$ とおく。**最小の $\|\Delta\mathbf{A}\|_2$（ただし $(\mathbf{A} + \Delta\mathbf{A})\hat{\mathbf{x}} = \mathbf{b}$ を満たすもの）は $\|\mathbf{r}\|/\|\hat{\mathbf{x}}\|$ であり、$\Delta\mathbf{A} = \mathbf{r}\hat{\mathbf{x}}^{\top}/\|\hat{\mathbf{x}}\|^2$ によって達成される。**

証明。 そのような任意の $\Delta\mathbf{A}$ は $\Delta\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} = \mathbf{r}$ を満たすため、$\|\mathbf{r}\| \leq \|\Delta\mathbf{A}\|\,\|\hat{\mathbf{x}}\|$ が成り立つ。ランク 1 行列 $\mathbf{r}\hat{\mathbf{x}}^{\top}/\|\hat{\mathbf{x}}\|^2$ は $\hat{\mathbf{x}}$ を $\mathbf{r}$ に写し、そのノルムは $\|\mathbf{r}\|\,\|\hat{\mathbf{x}}\|/\|\hat{\mathbf{x}}\|^2 = \|\mathbf{r}\|/\|\hat{\mathbf{x}}\|$ である。■

$\mathbf{A}$ の大きさに対する相対的な後退誤差は次のようになる。

$$\omega = \frac{\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\|}{\|\mathbf{A}\|\,\|\hat{\mathbf{x}}\|}. \tag{35.105}$$

残差が小さければ、問題の条件がどうであれ後退誤差が小さいことが保証される。2 直線の方程式において $\hat{\mathbf{x}} = (0, 2)$ とすると、$\mathbf{r} = (0, -10^{-4})$ であり、$\omega = 10^{-4}/(2.00005 \cdot 2) = 2.5 \times 10^{-5}$ となる。$\mathbf{b}$ にも摂動を許す定式化は Rigal と Gaches によるものである（Higham (2002) を参照）。浮動小数点では残差自体も丸め誤差を伴って計算されるため、出力された ω の値のうち u 程度より小さいものはノイズである。

条件数は後退誤差を前進誤差へと変換する。これはぜひ覚えておくべき結果であり、その証明に必要なのは 2 つのノルム $\|\mathbf{A}\| = \sigma_1$ と $\|\mathbf{A}^{-1}\| = 1/\sigma_d$ のみである（33.5 節）。

命題（前進誤差 $\le$ 条件数 $\times$ 後退誤差）。 *$\mathbf{A}$ を正則行列とし、$\mathbf{A}\mathbf{x} = \mathbf{b}$ とする。また、計算解 $\hat{\mathbf{x}}$ が近傍の方程式 $(\mathbf{A} + \Delta\mathbf{A})\,\hat{\mathbf{x}} = \mathbf{b}$（ただし $\|\Delta\mathbf{A}\| \le \omega \|\mathbf{A}\|$）を厳密に解くとする。このとき次が成り立つ。*

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\hat{\mathbf{x}}\|} \;\le\; \kappa(\mathbf{A})\,\omega. \tag{35.106}$$

証明。 $\mathbf{A}\mathbf{x} = \mathbf{b}$ を $(\mathbf{A} + \Delta\mathbf{A})\hat{\mathbf{x}} = \mathbf{b}$ から引くと $\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}) = -\Delta\mathbf{A}\,\hat{\mathbf{x}}$ となり、したがって $\hat{\mathbf{x}} - \mathbf{x} = -\mathbf{A}^{-1}\Delta\mathbf{A}\,\hat{\mathbf{x}}$ から次が得られる。

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \le \|\mathbf{A}^{-1}\|\,\|\Delta\mathbf{A}\|\,\|\hat{\mathbf{x}}\| \le \frac{1}{\sigma_d}\,\sigma_1\,\omega\,\|\hat{\mathbf{x}}\| = \kappa(\mathbf{A})\,\omega\,\|\hat{\mathbf{x}}\|. \qquad \blacksquare$$

ここでの誤差は、手元にあるベクトル $\hat{\mathbf{x}}$ に対する相対値として測られている。真の解 $\mathbf{x}$ に対する相対誤差については、$\|\hat{\mathbf{x}}\| \le \|\mathbf{x}\| + \|\hat{\mathbf{x}} - \mathbf{x}\|$ を代入して $\|\hat{\mathbf{x}} - \mathbf{x}\|$ について解くと、上界 $\kappa\omega/(1 - \kappa\omega)$（$\kappa\omega < 1$ のとき）が得られ、一次近似では同じである。2 直線の方程式では $\kappa\omega = 4 \times 10^{4} \cdot 2.5 \times 10^{-5} = 1.0$ であり、$\hat{\mathbf{x}}$ に対する相対誤差は $\|(-1, 1)\|/\|(0, 2)\| = 0.71$ となって上界の内側に収まる。行列ではなく右辺の誤差に対する対応する上界は 33.5.3.2 節に示されている。

後退安定なソルバは $\omega \approx u$ を満たし、この上界の $\log_{10}$ をとると次の経験則が得られる。

$$\text{correct digits in } \hat{\mathbf{x}} \;\approx\; \text{digits carried by the format} \;-\; \log_{10}\kappa(\mathbf{A}).$$

この経験則は最悪ケースの評価である。$\kappa = 10^{3}$ の行列では、fp32 が持つ 7 桁のうち 3 桁が失われうる。一方、本章の通し例である $\tfrac{1}{2}(x_1^2 + 1000x_2^2)$ は同じ κ を持つにもかかわらず 1 桁も失わない。行列が対角行列であり、各未知数が正しく丸められる 1 回の除算だけで求まるからである。

35.8.4.3 例：ヒルベルト行列

ヒルベルト行列（Hilbert matrix）$\mathbf{H}_{ij} = 1/(i + j - 1)$（$i, j = 1, \dots, n$）は、$n$ に対して指数関数的に増大する条件数を持ち、その各行は先ほどの 2 直線と同様にしてほぼ平行になっていく。ここでは $\mathbf{H}\mathbf{x} = \mathbf{b}$ を $\mathbf{b} = \mathbf{H}\mathbf{1}$（厳密解は $\mathbf{1}$）として解き、前進誤差を先ほどの経験則および後退誤差 Equation 35.105 と比較する。ソルバを評価する前に 1 つ注意点がある。$\mathbf{b} = \mathbf{H}\mathbf{1}$ を fp64 で計算した時点で $\mathbf{b}$ の各成分が丸められ、ソルバが動き出す前に相対サイズ u のデータ変化が生じているのである。（$1/3$ や $1/7$ などの行列成分の丸め自体は、格納された行列から $\mathbf{b}$ を作っているためここでは悪影響を与えない。）この丸めだけでどれほどの誤差が生じるだろうか。すべての fp64 の数は有理数であるから、格納された連立一次方程式は `Fraction` 型の成分に対するガウス・ジョルダン消去法によって丸め誤差を一切伴わずに解くこともできる。以下の表では、その厳密求解の誤差を `np.linalg.solve` の誤差と並べて示す。

```
def solve_exact(A, b):
    """Gauss-Jordan elimination in exact rational arithmetic."""
    M = [[Fraction(a) for a in row] + [Fraction(c)] for row, c in zip(A, b)]
    n = len(M)
    for j in range(n):              # no pivoting: A is positive definite here
        for i in range(n):
            if i != j:
                r = M[i][j] / M[j][j]
                M[i] = [a - r * c for a, c in zip(M[i], M[j])]
    return np.array([float(M[i][n] / M[i][i]) for i in range(n)])
```

次元 $n \in \{4, 6, 8, 10, 12\}$ に対して $n \times n$ のヒルベルト行列 `H` と右辺 `b = H @ x_true` を構成し、浮動小数点ソルバ `np.linalg.solve(H, b)` と `solve_exact(H, b)` を条件数、前進誤差、保持桁数、後退誤差にわたって比較しよう。

```
print(' n     kappa  forward error  digits kept  16 - log10 kappa  backward error'
      '  exact solve error')
for n in [4, 6, 8, 10, 12]:
    i = np.arange(n)
    H = 1.0 / (1.0 + i[:, None] + i[None, :])   # Hilbert matrix
    x_true = np.ones(n)
    b = H @ x_true
    x_hat = np.linalg.solve(H, b)
    kappa = np.linalg.cond(H)
    fwd = np.linalg.norm(x_hat - x_true) / np.linalg.norm(x_true)
    omega = (np.linalg.norm(b - H @ x_hat)
             / (np.linalg.norm(H, 2) * np.linalg.norm(x_hat)))
    exact = np.linalg.norm(solve_exact(H, b) - x_true) / np.linalg.norm(x_true)
    print(f'{n:2d}  {kappa:8.1e}  {fwd:13.1e}  {-np.log10(fwd):11.1f}'
          f'  {16 - np.log10(kappa):16.1f}  {omega:14.1e}  {exact:17.1e}')
```

n	kappa	forward error	digits kept	16 - log10 kappa	backward error	exact solve error
4	1.6e+04	4.1e-14	13.4	11.8	0.0e+00	1.1e-13
6	1.5e+07	1.4e-10	9.8	8.8	1.3e-16	1.0e-10
8	1.5e+10	6.1e-08	7.2	5.8	5.7e-17	7.0e-08
10	1.6e+13	8.7e-05	4.1	2.8	1.0e-16	4.5e-05
12	1.8e+16	3.2e-01	0.5	-0.2	9.8e-17	1.4e-01

保持された桁数は、約 13 桁（$n = 4$）から 1 桁を大きく下回る水準（$n = 12$）まで落ち込む一方、後退誤差は約 10^{-16} 以下にとどまり続ける。すなわち、計算された各ベクトルは、格納された方程式と 16 桁目または 17 桁目しか違わない方程式を解いている。では、桁数が失われた責任はソルバにあるのだろうか。右端の列がその潔白を証明している。丸め誤差を一切犯さない格納済み方程式の厳密求解でも、**1** に対してすべての行で同じオーダーの誤差（約 10^{-7}、10^{-4}、10^{-1} がそれぞれ $n = 8$、10、12 で発生）が生じている。桁数はソルバが走る前、**b** を格納した時点ですでに失われていたのである。誤差が最も増幅される方向と完全には揃っていないため、経験則は約 1 桁（行によって約 0.5 桁から約 1.5 桁）だけ悲観的になっている。$n = 12$ では $\kappa u \approx 2$ となり、上界はもはや何も縛らない。末尾の数値は LAPACK のビルドによって多少変わるが、全体の傾向は変わらない。

35.8.4.4 正規方程式の条件性

最小二乗問題 $\min_{\mathbf{w}} \|\mathbf{A}\mathbf{w}-\mathbf{b}\|^2$（ただし $\mathbf{A}\in\mathbb{R}^{n\times d}$ は列フルランク）は、正規方程式 $\mathbf{A}^\top\mathbf{A}\mathbf{w}=\mathbf{A}^\top\mathbf{b}$ によって解かれる。これは代数的に同値でありながら、より悪条件な行列を経由する経路である（33.5.3.2 節）。

$$\kappa(\mathbf{A}^\top\mathbf{A}) = \kappa(\mathbf{A})^2. \tag{35.107}$$

これと同じ現象を 1 パス分散でもすでに見た。手軽な公式が問題の条件数を 2 乗してしまうのである。経験則によれば、正規方程式は約 $2\log_{10}\kappa$ 桁を失いうる。一方、QR 分解（$\mathbf{A}=\mathbf{Q}\mathbf{R}$、ただし $\mathbf{Q}$ は正規直交列を持ち $\mathbf{R}$ は三角行列）や特異値分解（SVD）による解法は $\mathbf{A}$ に直接作用するため、**残差 $\mathbf{b}-\mathbf{A}\mathbf{w}$ が小さいときには**失う桁数が約 $\log_{10}\kappa$ で済む。残差が大きい場合には、最小二乗問題自体がより敏感になる。その条件数はおよそ $\kappa+\kappa^2\tan\theta$ であり、ここで θ は $\mathbf{b}$ と $\mathbf{A}$ の列空間（column space）がなす角である。このとき QR 分解や SVD も $2\log_{10}\kappa$ 桁近くを失うが、問題自体が強制する以上に失うことは決してない。これに対し正規方程式は、残差の大小にかかわらず常にそれだけの桁数を失う。

本章の通し例は良条件すぎてこの差が見えない。その計画行列は $\mathbf{A}^\top\mathbf{A}=\mathrm{diag}(1,1000)$ を満たすため、$\kappa(\mathbf{A})=\sqrt{1000}\approx 32$ にすぎないからである。そこで以下のセルでは、ランダムな正規直交因子と列空間内の右辺（残差ゼロ）から、$\kappa(\mathbf{A})=10^3$ および 10^6 を持つより難しい計画行列を構成する。3 桁および 6 桁の差が現れるはずである。

```
rng = np.random.default_rng()
n, d = 100, 10
for kappa in [1e3, 1e6]:
    U, _ = np.linalg.qr(rng.normal(size=(n, d)))    # orthonormal columns
    V, _ = np.linalg.qr(rng.normal(size=(d, d)))
    sigma = np.logspace(0, -np.log10(kappa), d)     # kappa(A) as chosen
    A = U * sigma @ V.T                             # A = U diag(sigma) V^T
    w_true = rng.normal(size=d)
    b = A @ w_true                                  # zero residual
    w_ne = np.linalg.solve(A.T @ A, A.T @ b)        # normal equations
    w_ls = np.linalg.lstsq(A, b, rcond=None)[0]     # SVD-based solve
    digits = [-np.log10(np.linalg.norm(w - w_true) / np.linalg.norm(w_true))
              for w in (w_ne, w_ls)]
    print(f'kappa(A) = {kappa:.0e}: normal equations keep {digits[0]:4.1f} '
          f'digits, lstsq keeps {digits[1]:4.1f}')
```

```
kappa(A) = 1e+03: normal equations keep 10.3 digits, lstsq keeps 14.3
kappa(A) = 1e+06: normal equations keep  5.7 digits, lstsq keeps 11.7
```

$\kappa(\mathbf{A})=10^3$ では正規方程式が保持する桁数は約 $10\sim12$ 桁であるのに対し `lstsq` は約 14 桁であり、10^6 では約 $4\sim6$ 桁に対して約 $11\sim13$ 桁となる。正確な桁数は実行ごとに 1〜2 桁変動するが、両者の差は $\log_{10}\kappa$ の周りに分布し、大半がそこから 1 桁以内に収まる。`lstsq` や 33.5.3.1 節の擬似逆行列（pseudoinverse）が特異値分解（SVD）に基づいて計算され、$(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$ を用いない理由はここにある。

35.8.4.5 リッジ正則化と条件性

特徴量が強い多重共線性を持つか計画行列がランク落ちしているために $\kappa(\mathbf{A})$ 自体が大きい場合、いかなるアルゴリズムを選んでも助けにはならず、問題そのものを変えなければならない。答え

を変えずに問題を変える方法が 1 つある。**前処理行列**（preconditioner）は変数変換によって新しい問題をより良条件にし、その解を元の問題の解へと写し戻す。$\mathbf{A}$ の各列をそのノルムで割る操作はその一例であり（演習問題 4）、35.5 節の座標ごとのステップサイズもオプティマイザの内部で同じ働きをしている。

リッジ正則化は答えそのものも変化させる。これは $\frac{1}{2}\|\mathbf{A}\mathbf{w}-\mathbf{b}\|^2+\frac{\lambda}{2}\|\mathbf{w}\|^2$ を最小化する手法であり、その最小点（minimizer）は次の式で与えられる。

$$\mathbf{w}_\lambda=(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{b}, \tag{35.108}$$

これは 33.5.3.1 節で導いたリッジ解である。$\sigma_1\geq\cdots\geq\sigma_d\geq 0$ を行列 $\mathbf{A}\in\mathbb{R}^{n\times d}$ の特異値とし、ランクを超える添字では $\sigma_i=0$ とする。行列 $\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I}$ は $\mathbf{A}^\top\mathbf{A}$ と同じ固有ベクトルを持ち、固有値は $\sigma_i^2+\lambda\geq\lambda>0$ となる。したがって正定値であり、$\mathbf{A}$ がランク落ちしていても正則となって、条件数は次を満たす。

$$\kappa(\mathbf{A}^\top\mathbf{A}+\lambda\mathbf{I})=\frac{\sigma_1^2+\lambda}{\sigma_d^2+\lambda}. \tag{35.109}$$

$\sigma_1>\sigma_d$ である限り、この比は λ に関して狭義単調減少し（導関数は $(\sigma_d^2-\sigma_1^2)/(\sigma_d^2+\lambda)^2<0$）、$1$ へと $\lambda\to\infty$ で収束する。λ を加えることはスペクトルの床を押し上げる働きをする。$\lambda\ll\sigma_1^2$ の間は天井がほとんど動かないため、κ はほぼ $1/\lambda$ に比例して（$\lambda\gg\sigma_d^2$ となった領域で）減少する。二次オプティマイザのダンピング項や、ガウス過程回帰（第 40 章（英語版））でカーネル行列に加えるジッタは、別の名前で呼ばれているこの $\lambda\mathbf{I}$ にほかならない。

その代償は最小点が変化することである。現在の記法で、$\mathbf{v}_i$ を行列 $\mathbf{A}$ の右特異ベクトル、$\mathbf{w}_0=\mathbf{A}^+\mathbf{b}$ を最小ノルム最小二乗解とすると、33.5.3.1 節のフィルタ係数は次の形をとる。

$$\mathbf{w}_\lambda=\sum_{i=1}^{d}\frac{\sigma_i^2}{\sigma_i^2+\lambda}\,(\mathbf{v}_i^\top\mathbf{w}_0)\,\mathbf{v}_i. \tag{35.110}$$

リッジ正則化が最も強く減衰させるのは小さな σ_i を持つ方向であり、それはまさに正則化なしの求解がデータ誤差を $1/\sigma_i$ 倍に増幅してしまう方向である。Figure 35.33 に等位集合への効果を示す。各パネルはそれぞれの最小点を中心に描かれているため、最小点が原点へ向かって移動する様子自体は描画されていない。

この同じ κ が、リッジ目的関数に対する勾配降下法のステップ数をも支配する。最良の固定ステップサイズのもとで、条件数 $\kappa>1$ の二次関数に対する勾配降下法は 1 ステップごとに誤差を $\rho=(\kappa-1)/(\kappa+1)$ 倍に縮小させる（35.1 節）。誤差が $-\mathbf{w}^\star$ である原点から出発すると、相対誤差は $\|\mathbf{w}_k-\mathbf{w}^\star\|/\|\mathbf{w}^\star\|\leq\rho^k$（$k$ ステップ後）となるため、許容誤差 $0<\mathrm{tol}<1$ に達するまでのステップ数は次で抑えられる。

$$k\leq\left\lceil\frac{\log(1/\mathrm{tol})}{\log\frac{\kappa+1}{\kappa-1}}\right\rceil\leq\left\lceil\frac{\kappa}{2}\,\log\frac{1}{\mathrm{tol}}\right\rceil\quad\text{steps.} \tag{35.111}$$

第 2 の不等式は $\log\frac{\kappa+1}{\kappa-1}=2\,\mathrm{artanh}(1/\kappa)\geq 2/\kappa$ から従う。6 桁の精度を得るには $\frac{1}{2}\log 10^6=6.9$ ステップ（κ の 1 単位あたり）が上界となるが、3 次元以上での実行ではそれより早く収束しうる（演習問題 4）。

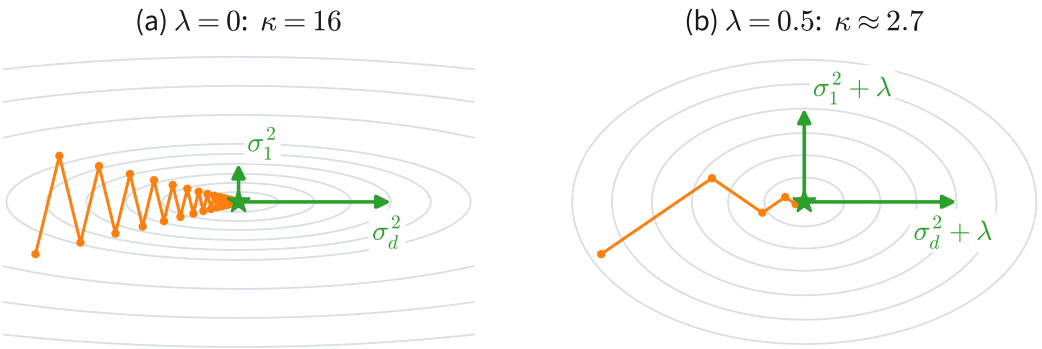


Figure 35.33: リッジ項 $\frac{\lambda}{2}\|\mathbf{w}\|^2$ を加える前後の最小二乗目的関数の等位集合。両パネルが見やすいよう $\kappa = 16$ で描いている（通し例の $\kappa = 1000$ では線のように潰れてしまう）。ペナルティは $\mathbf{A}^\top\mathbf{A}$ のすべての固有値を λ だけ押し上げる。$1/\sqrt{\sigma_i^2 + \lambda}$ に比例する半軸は曲率の小さい方向で長く、細長い谷が丸みを帯びたボウルへと変わる。オレンジ色：各パネルの最良の固定ステップサイズを用いた同じ初期点からの勾配降下法の反復点、緑色の矢印：曲率を添えた主軸。

与えられた λ によって何ステップ節約でき、そのコストはどれだけになるだろうか。通し例で両方を測定してみよう。目的関数 $\frac{1}{2}(w_1^2 + 1000w_2^2) - w_1 - 1000w_2$ は、$\mathbf{A}^\top\mathbf{A} = \mathrm{diag}(1, 1000)$ および $\mathbf{A}^\top\mathbf{b} = (1, 1000)$ を持つ最小二乗問題である。その最小点は $(1, 1)$ であるため、リッジ解の最小点には 2 つのフィルタ係数がそのまま現れる。関数 `gd_iterations` は最良の固定ステップサイズで勾配降下法を実行し、ステップ数とリッジ解の最小点を返す。

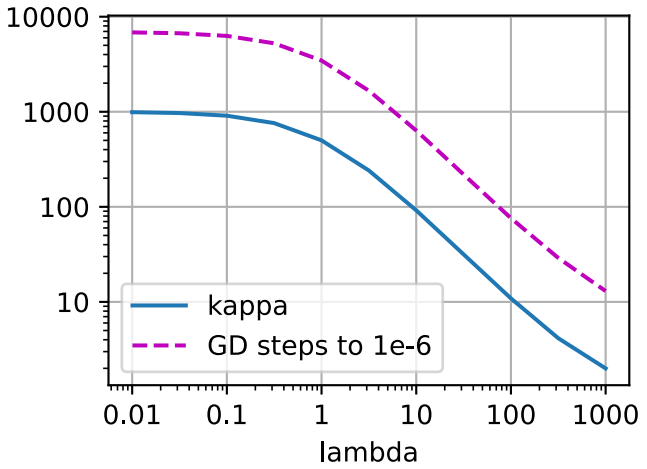
```
def gd_iterations(AtA, Atb, lam, tol=1e-6):
    """GD steps at step 2/(L+mu) to reach relative error tol on ridge."""
    M = AtA + lam * np.eye(len(Atb))
    w_star = np.linalg.solve(M, Atb)
    mu, L = np.linalg.eigvalsh(M)[[0, -1]]
    eta, w, k = 2 / (L + mu), np.zeros(len(Atb)), 0
    while np.linalg.norm(w - w_star) > tol * np.linalg.norm(w_star):
        w, k = w - eta * (M @ w - Atb), k + 1
    return k, w_star

AtA, Atb = np.diag([1.0, 1000.0]), np.array([1.0, 1000.0])
print('lambda   kappa  steps  steps/kappa  ridge minimizer')
for lam in [0, 1, 10, 100, 1000]:
    kappa = (1000 + lam) / (1 + lam)
    k, w_star = gd_iterations(AtA, Atb, lam)
    print(f'{lam:6d} {kappa:7.1f} {k:6d} {k / kappa:12.2f}  {w_star.round(4)}')
```

```
lambda   kappa  steps  steps/kappa  ridge minimizer
     0  1000.0   6908         6.91  [1. 1.]
     1   500.5   3458         6.91  [0.5   0.999]
    10    91.8    635         6.92  [0.0909 0.9901]
   100    10.9     76         6.98  [0.0099 0.9091]
  1000     2.0     13         6.51  [0.001 0.5  ]
```

条件数は 1000 から 2 へと（λ が 0 から 1000 へ動くにつれて）低下し、それに伴ってステップ数も 6,908 から 13 へと減少する。ステップ数と κ の比は 6.9 付近に保たれる（小さな κ では近似

$2/\kappa$ が緩くなり、整数ステップへの切り上げも効いてくる）。右端の列は節約されたステップ数の代償を示している。$\lambda = 1$ では、$\sigma^2 = 1$ の遅い方向が半分の 0.5 に縮小する一方、$\sigma^2 = 1000$ の硬い方向は 0.1% しか動かない。$\lambda = 1000$ では硬い方向も半分に縮小する。λ をスイープすると、κ とステップ数の関係を同時に確認できる。

```
lams = np.logspace(-2, 3, 11)
kappas = (1000 + lams) / (1 + lams)
steps = [gd_iterations(AtA, Atb, lam)[0] for lam in lams]
d2l.plot(lams, [kappas, steps], 'lambda', None,
         legend=['kappa', 'GD steps to 1e-6'], xscale='log', yscale='log')
```



両対数軸上では 2 本の曲線は 6.9 倍の間隔を保って平行に並び、傾きは -1 に近い（λ が固有値 1 と 1000 の十分内側にある間）。このようにリッジ正則化は連立一次方程式の求解と勾配降下法の双方の条件性を改善し、いずれに対しても解を縮小させることでその代償を支払う。35.7.3.2 節で示した通り、ペナルティ $\frac{\lambda}{2}\|\mathbf{w}\|^2$ は $\|\mathbf{w}\|$ に対する制約に対応しているため、統計的な理由から選ばれた λ が計算の条件性をも同時に定めることになる。

ここまでの道具立てはいずれも、100 層を通過する勾配のように反復の多数ステップにわたって増大する誤差は扱っていない。それについては 4.4 節で取り上げる。

35.8.5 まとめ

浮動小数点形式は、約 p ビットの相対精度と有限の指数部範囲で数を格納する。各丸めは高々 $u = 2^{-p-1}$ の相対誤差を生じる一方、オーバーフロー、アンダーフロー、微小な更新の消失は相対モデルには現れない絶対スケールの現象である。ソフトマックスと交差エントロピーは形式が保持できない値を経由するために破綻し、1 パス分散はほぼ等しい大きな数同士を減算するために破綻するが、いずれの真の答えも極端な値ではない。最大値シフト、ロジットから計算する交差エントロピー、Welford の漸化式はこうした迂回を避ける。相対条件数 $\kappa_f = |xf'/f|$ は不適切なアルゴリズムを難しい問題から切り分け、この値が小さいのに計算結果が悪ければアルゴリズムに原因がある。連立一次方程式では、$\kappa(\mathbf{A})$ が前進誤差を後退誤差（残差によって保証される）の κ 倍で抑えるため、約 $\log_{10}\kappa$ 桁が失われ、正規方程式を経由するとその 2 倍の桁数が失われる。リッジ正則化はすべての特異値の 2 乗に λ を加えることで求解と勾配降下法の双方に対して κ を引き下げるが、その代償として解はフィルタ係数 $\sigma_i^2/(\sigma_i^2 + \lambda)$ 倍に縮小する。

35.8.6 演習問題

1. **損失スケールの許容範囲**。損失スケーリングは損失に定数 S を掛け（これによりすべての勾配が S 倍になる）、勾配を fp32 に変換したあとで S で割る。fp16 では、最小の正規化数は $2^{-14} \approx 6.1 \times 10^{-5}$、最小の非正規化数は $2^{-24} \approx 6.0 \times 10^{-8}$（$2^{-25}$ 未満の値はゼロに丸められる）、最大の有限数は 65504 である。あるネットワークの厳密な非ゼロ勾配成分の大きさが 10^{-9} から 10^{-1} の間にあり、逆伝播ではスケールされた勾配を fp16 で計算するとする。
   1. スケールされたすべての勾配が fp16 の正規化数となるような 2 のべき乗 S をすべて求めよ。
   2. より深いネットワークの勾配の大きさが 10^{-12} から 10^{-1} の間にあるとする。いかなるスケールを用いてもそのすべてを正規化数に保つことはできないことを示せ。オーバーフローを避ける最大の 2 のべき乗 S をとるとき、大きさ 10^{-12} の勾配はスケール後に有効ビットを何ビット保持し、その相対丸め誤差はいくらになるか。
   3. 動的損失スケーリング（5.5.3.3 節）はオーバーフローを避ける最大のスケールを選び、それによって最小側の勾配を犠牲にする。そちら側を犠牲にする方が望ましいのはなぜか。
   4. 勾配を bf16 で計算する場合、最初の 2 つの小問の答えはどう変わるか。

2. [code] **問題とアルゴリズムの切り分け**。以下の各式を fp32 で評価する。それぞれについて、指定された領域における**関数**の相対条件数 Equation 35.102 と、**書かれた通りの式**の桁落ち係数 Equation 35.103 を計算せよ。問題とアルゴリズムのどちらに原因があるかを判定し、誤差が問題自体の感度に一致するよう式を書き換えよ。さらに両方の形を fp64 の結果と照合せよ。
   1. $ax^2 + bx + c = 0$（ただし $b > 0$ かつ $b^2 \gg 4|ac|$）の絶対値が小さい方の根を c の関数とみなしたもの（たとえば $a = c = 1$、$b = 10^4$）。（ヒント：共役な式を掛けよ。）
   2. 関数 $\log(1 - e^{-x})$（ただし $x > 0$；$\log(1 - q)$ を $\log q = -x$ から計算する式）を $x \in \{10^{-6}, 10^{-3}, 1, 20\}$ で評価したもの。小さい x で正確な定式化と大きい x で正確な定式化をそれぞれ 1 つずつ見つけ、切り替え点を選べ。
   3. ロジット t（ラベル 1）の二値交差エントロピー $\log(1 + e^{-t})$ を $t \in \{-800, -50, 0, 50\}$ で評価したもの。これらのうち、書かれた通りの式がオーバーフローするのはどの値か、また 0 を返すのはどの値か。$\max(-t, 0) + \operatorname{log1p}(e^{-|t|})$ がこの関数に等しく、決してオーバーフローせず、桁も失わないことを示せ。これは log-sum-exp のサンドイッチ不等式のどの部分にあたるか。
   4. $\sqrt{x+1} - \sqrt{x}$（$x = 10^8$ において）。

   小問 2 は Martin Mächler「Accurately Computing log(1 −exp(−|a|))」第 2 節を改変。

3. [code] **浮動小数点における正規方程式**。正規方程式は条件数を 2 乗する（Equation 35.107）。
   1. $\tau > 0$ に対し、$(m+1) \times m$ 行列 $\mathbf{A}$（Läuchli (1961)）は第 1 行がすべて 1 であり、その下に $\tau\mathbf{I}$ を並べた行列である。$\mathbf{A}^\top\mathbf{A}$、$\mathbf{A}$ の特異値、および $\kappa(\mathbf{A})$ を求めよ。$\tau = 10^{-8}$ のとき、行列 $\mathbf{A}^\top\mathbf{A}$ が fp64 で特異行列へ丸められることを示せ。35.8.4 節の経験則は、この $\mathbf{A}$ を用いた QR 分解ベースの最小二乗求解に対し、$\mathbf{b}$ が $\mathbf{A}$ の値域に属するとき正しい桁が何桁残ると予測するか。残差が大きいとき、その予測はどう変わるか。
   2. `#numerical-stability-conditioning-normal-equations` セルを変更し、$\kappa(\mathbf{A}) = 10^k$ を $k = 2, \dots, 10$ にわたってスイープせよ。正規方程式と `lstsq` の正しい桁数を k に対して、経験則の 2 直線 $16 - 2k$ および $16 - k$ と重ねてプロットせよ。経験則が正規方程式に正しい桁が 1 桁も残らないと予測するのはどの k か、またそこでは実際に何が観測されるか。

4. [code] **前処理としての特徴量スケーリング**。異なる単位で測られた特徴量は、計画行列の

各列に大きく異なるノルムを与える。$\mathbf{X}$ を標準正規分布に従う成分を持つ 200×5 行列とし、$\mathbf{A} = \mathbf{X}\,\mathrm{diag}(1, 3, 10, 30, 100)$ とおき、$\mathbf{b} = \mathbf{A}\mathbf{w}^\star$（$\mathbf{w}^\star$ はランダム）とおく。

1. `#numerical-stability-conditioning-ridge` セルの `gd_iterations`（$\mathbf{A}^\top\mathbf{A}$ と $\mathbf{A}^\top\mathbf{b}$ を引数にとる）をこの計画行列に $\lambda = 0$ で適用せよ。$\kappa(\mathbf{A}^\top\mathbf{A})$ を計算し、ステップ数を上界 Equation 35.111 と比較せよ。ステップ数が上界を下回る理由を、初期点の誤差が $\mathbf{A}^\top\mathbf{A}$ の両端の固有ベクトル方向に持つ成分から説明せよ。
2. $\mathbf{A}$ のすべての列をそのノルムで割り、同じ測定を繰り返せ。再スケーリング後の問題の最小点は元の $\mathbf{w}^\star$ へどのように写し戻されるか。
3. 演習問題 1（35.2 節）の小問 2 では、新しい座標 $\mathbf{u}$（$\mathbf{w} = \mathbf{T}\mathbf{u}$）における勾配降下法を、正定値行列 $\mathbf{P}$ で前処理された元の問題に対する勾配降下法（更新式 $\mathbf{w} \leftarrow \mathbf{w} - \eta\,\mathbf{P}^{-1}\nabla f(\mathbf{w})$）として表した。列スケーリングに対応するのはどの $\mathbf{T}$ とどの $\mathbf{P}$ か。また、$\mathbf{P}$ は $\mathbf{A}^\top\mathbf{A}$ の対角成分とどのように関係するか。

5. [code] **確率的丸め**。●確率的丸め (Gupta et al. 2015) は、実数 x（隣り合う表現可能な値 $x_- < x_+$ の間にある）を、x_+ へ確率 $q = (x - x_-)/(x_+ - x_-)$ で丸め、それ以外の場合は x_- へ丸める。

   1. 丸め後の値が期待値 x と分散 $q(1-q)\Delta^2 \le \Delta^2/4$（ただし $\Delta = x_+ - x_-$）を持つことを示せ。
   2. bf16 の重みが $w = 1$ から出発し、$K = 1000$ 回の -10^{-4} の更新を受け、毎回その直後に確率的丸めを行うとする。重みは $[0.5, 1]$ の範囲にとどまり、そこでの bf16 の間隔は $\Delta = 2^{-8}$ である。異なるステップの丸め誤差が互いに無相関であることを示し、最終的な重みの平均と標準偏差を予測せよ。最近傍丸めを用いた場合、最終的な重みはいくらになるか。
   3. 正の fp32 の数に対し、そのビットパターンに $\{0, \dots, 2^{16} - 1\}$ から一様ランダムに選んだ整数を加えてから下位 16 ビットをクリアすると、bf16 への確率的丸めになる。その理由を説明せよ。35.8.1 節の `to_bf16` 関数の横にこの丸めを実装し、1000 回の独立な実行によって前小問の予測を検証せよ。さらに、同じ更新を受ける fp32 のマスターコピーと比較せよ。

6. **厳密な減算**。●減算そのものが厳密であっても、不正確な差が返ることがある。

   1. 上界 Equation 35.103 が、適切な δ_a と δ_b（$|\delta_a|, |\delta_b| \le u$）に対して達成されることを示せ。
   2. Sterbenz の補題 (Sterbenz 1974) を証明せよ。すなわち、a と b が形式 Equation 35.95 の浮動小数点数であり $b/2 \le a \le 2b$ を満たすならば、$a - b$ も浮動小数点数となり、したがって計算された差は厳密である。ただし、その形式は非正規化数を含むと仮定せよ。
   3. `#numerical-stability-conditioning-log1p` セルでは、fp32 の差 $a - b$ が真の値 3.0×10^{-7} から約 20% ずれている。Sterbenz の補題と出力された ulp 数を用いてこの減算が厳密であったことを確認し、格納された a と b の相対誤差を計算したうえで、最初の小問の上界がその誤差を覆っていることを確かめよ。

7. [code] **逐次モーメントの統合**。●データ並列訓練では各シャード上で統計量を計算し、それらを結合する。シャード A が k_A 個の値を持ち、その平均を m_A、中心化二乗和を M_A とする。同様にシャード B が k_B 個の値を持ち、m_B と M_B を持つとする。

   1. 2 つのシャードの和集合における平均 m と中心化二乗和 M の厳密な公式を、k_A、m_A、M_A、k_B、m_B、M_B を用いて導出せよ。
   2. $k_B = 1$ の場合が Welford の漸化式 Equation 35.104 に一致することを示せ。
   3. 10^5 個のサンプル（`#numerical-stability-conditioning-welford` セルのもの）を、ほぼ等サイズの 2^j 個のシャードへ分割せよ（`np.array_split` を用い、$j \in \{1, 4, 8\}$ と

する）。各シャードの統計量を Welford の漸化式で計算し、二分木状にペアワイズ統合して、得られた分散を 2 パスの基準値と比較せよ。続いて、各シャードの和 $\sum_i x_i$ と $\sum_i x_i^2$ を足し合わせて 1 パス公式を適用する方法でもう一度統合し、両者の違いを説明せよ。

8. [code] **総和の誤差増大。**●$S_n = \sum_{i=1}^{n} x_i$ とし、$\hat{S}_n$ をその計算値とする。順次総和法は最悪ケースの上界 $|\hat{S}_n - S_n| \le (n-1)\,u \sum_i |x_i| + O(u^2)$ を持つ。

   1. $n = 2^k$ に対し、再帰的なペアワイズ総和法は項を半分ずつに分け、それぞれの半分を再帰的に足し合わせてから 2 つの結果を加算する。これが $|\hat{S}_n - S_n| \le u \log_2(n) \sum_i |x_i| + O(u^2)$ を満たすことを証明せよ。
   2. $n \in \{2^4, 2^8, 2^{12}, 2^{16}, 2^{20}\}$ 個の $[0, 1)$ 上の一様乱数を fp32 で格納し、順次総和法（`np.cumsum(x)[-1]`）、`np.sum`（ブロック化されたペアワイズ総和法）、および Kahan の補償付き総和法 (Kahan 1965) の相対誤差を float64 の基準値に対して測定せよ。Kahan の手法は補正値 c を保持し、各加算 $t = s + y$ のあとに $c = (t - s) - y$ とおいて次の項から c を引く。3 つの誤差を 2 つの上界とともに n に対してプロットせよ。
   3. 順次総和法の実測誤差は、最悪ケースの上界と比べて n に対しどのように増大するか、またそれはなぜか。本文における 10^7 個の 0.1 の総和が最悪ケースにはるかに近いのはなぜか。
   4. 大きな n において、高精度な 2 つの手法の誤差を律速している要因は何か。

   Steven G. Johnson「MIT 18.335 Introduction to Numerical Methods, Problem Set 1」問題 4 を改変。

9. [code] **混合精度反復改良法。**●●混合精度訓練は演算の大半を低精度で実行し、微小な補正を蓄積する量を高精度に保つ。反復改良法 (Moler 1967; Carson and Higham 2018) は同じ役割分担を $\mathbf{A}\mathbf{x} = \mathbf{b}$ に適用する。すなわち $O(n^3)$ の行列分解を低精度で実行し、$O(n^2)$ の残差計算だけを高精度で実行する。fp32 の解 $\mathbf{x}_0$ から出発し、次を繰り返す。

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k, \qquad (\mathbf{A} + \Delta\mathbf{A}_k)\,\mathbf{d}_k = \mathbf{r}_k, \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k,$$

   ここで残差と更新は fp64 で計算し、補正ベクトル $\mathbf{d}_k$ は fp32 の求解から得る。

   1. fp32 の求解を後退安定としてモデル化し、$\|\Delta\mathbf{A}_k\| \le c\,u_{32}\,\|\mathbf{A}\|$（ただし $u_{32} = 2^{-24}$、c は適度な定数）とおき、fp64 の丸め誤差は無視する。誤差 $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}$ が $\mathbf{e}_{k+1} = (\mathbf{A} + \Delta\mathbf{A}_k)^{-1}\Delta\mathbf{A}_k\,\mathbf{e}_k$ を満たすことを示せ。さらに縮小率を $c\,\kappa(\mathbf{A})\,u_{32}$ を用いて導出し、それが 1 未満となる条件を述べよ。
   2. `#numerical-stability-conditioning-hilbert` セルのヒルベルト行列に対し、$n = 3, \dots, 10$ でこの反復を実装せよ。補正の計算には、`np.linalg.solve` を $\mathbf{H}$ と $\mathbf{r}_k$ の fp32 コピーに適用すること。10 反復にわたって前進誤差を記録し、最終値を fp64 による求解の誤差と比較せよ。反復改良法が fp64 の求解精度に到達するのはどの n か。収束はするものの 10 反復以内ではその精度に届かないほど遅いのはどの n か。誤差が増大するのはどの n か。
   3. 各 n について 1 反復あたりの実測縮小率を推定し、それを $\kappa(\mathbf{H})\,u_{32}$ および $\mathbf{I} - \hat{\mathbf{H}}^{-1}\mathbf{H}$ のスペクトル半径（ただし $\hat{\mathbf{H}}$ は $\mathbf{H}$ の fp32 コピーであり、逆行列の適用は fp64 で行う）と比較せよ。この実験において $\Delta\mathbf{A}_k$ を支配している摂動は何か、また最初の小問の上界はどれほど悲観的か。

35.9 関連文献

本節では本章の範囲を超えた話題を紹介するが、そのすべてが同じ緊急度を持つわけではない。そのうち 5 つはニューラルネットワークの訓練に直接関わるため、それぞれの見出しの先頭に配置している。すなわち、安定性の縁（edge of stability）、反復点の平均化（iterate averaging）、補間（interpolation）、フィッシャー行列、および鞍点法（saddle-point method）である。残りの話題は古典的な最適化理論および数値解析からのものであり、最後に一般的な教科書、講義、サーベイをまとめる。なお、訓練の実践的な側面については 9.11 節で概観している。本書の他の箇所は本節の内容に一切依存しないため、関心のある項目を選んで自由に拾い読みするとよい。

35.9.1 勾配法

35.9.1.1 安定性の縁における自己安定化

ニューラルネットワークをフルバッチ勾配降下法で訓練すると、驚くべき現象が起こる。損失の鋭さ（sharpness、ヘッセ行列の固有値のうち最大のもの）が安定限界 $2/\eta$ まで上昇したのち、発散するのではなくちょうどその付近にとどまり続けるのである (Cohen et al. 2021)。35.1.2.4 節の二次モデルは、鋭さが $2/\eta$ を超えた瞬間に最大固有モードが増大する理由を示すことでこの振る舞いの前半を説明するが、純粋な二次モデルでは際限のない発散が予測されるため、何が鋭さを再び引き下げるのかを説明できない。Damian et al. (2023) は、テイラー展開にもう 1 つだけ項、すなわち最大固有ベクトル方向の 3 次の項を残すことでこの謎を解明した。その方向に沿って振動が大きくなると、この 3 次の結合が反復点を鋭さそのものの勾配と逆向きに押し出す。それによって鋭さが再び $2/\eta$ 未満へと下がり、次のサイクルが始まるまで振動が減衰する。この 3 次の項がなければ、振動は歯止めなく増大し続けるだろう。

35.9.1.2 最適多項式と共役勾配法

二次関数 $f = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x}$（固有値が $[\mu, L]$ に含まれ、最小点が $\mathbf{0}$ であるもの）の最小化を考えよう。k ステップ目の更新がそれまでに得られた勾配の張る空間（span）にとどまる任意の一次手法は、$\mathbf{x}_k = p_k(\mathbf{A})\mathbf{x}_0$ という形の反復点を生成する。ここで、p_k は次数が高々 k で $p_k(0) = 1$ を満たす多項式である。このように見れば、二次関数に対する高速な勾配法を設計することは、区間 $[\mu, L]$ 全体にわたって可能な限り小さくなる多項式を選ぶことと等価である。区間全体での最大値を最小にする多項式はスケーリングされたチェビシェフ多項式であり、その最大値は $2\rho^k/(1+\rho^{2k}) \approx 2\rho^k$ となる。ここで縮小率 $\rho = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ は、ヘビーボール法で現れたものと完全に一致する（35.1.3 節）。次元 d が k に比べて大きいとき、固有値は区間全体に広がり得るため、この族に属するいかなる手法もその最悪ケースの収束率（convergence rate）を上回ることはできない。共役勾配法（conjugate gradient method）は実用上さらに優れた振る舞いを示す。各ステップにおいて、f を**実際の**行列 $\mathbf{A}$ と初期点 $\mathbf{x}_0$ に対して最小化する多項式を、μ や L を一切知ることなく自動的に見つけ出し、厳密演算のもとでは高々 d ステップで終了するのである。Nocedal and Wright (2006)（第 5 章）は、標準的な共役勾配法とその前処理付き版の両方を導出している。

35.9.1.3 半正定値計画による収束率の保証

タイトな収束率を手計算で証明するのは、非常に繊細な作業として知られている。35.3.3.3 節における手計算の証明ではステップごとの縮小率として $1-\mu/L$ が得られたが、同項で引用した厳密な縮小率は $(1-\mu/L)^2$ であり、必要なステップ数は半分になる。このよりタイトな評価は、**性能推定**（performance estimation）と呼ばれる技法から得られる。Drori and Teboulle (2014) は、巧みな代数的変形を手作業で探す代わりに、すべての L 滑らかな凸関数にわたるアルゴリズムの最悪ケース性能を、反復点と勾配がなすグラム行列（すべてのペアの内積を並べた表）上の最適化問題として定式化した。この問題を上から評価すると小さな半正定値計画問題（semidefinite

program、SDP）に帰着し、その双対解から人間が検証可能な証明が文字通り得られる。Taylor et al. (2018) はまさにこの方法で二乗の縮小率を導出した。また制御理論の視点に立つ Lessard et al. (2016) は、勾配を μ と L のみで制約された線形力学系内部の未知の非線形性とみなし、ステップ数によらずサイズが一定の半正定値計画問題によって収束率を保証している。注目すべきことに、同じ枠組みは反例の発見にもつながり、ヘビーボール法が収束しない関数（35.1.3 節）もこうして発見された。

35.9.2 確率的手法と適応的手法

35.9.2.1 定数ステップによる平均化

35.4.1 節で見たように、定数ステップの SGD はステップサイズ η に比例するノイズフロアに落ち着く。時間とともに η を縮小すればこのノイズフロアは解消されるが、終盤の進行が遅くなるうえ、スケジュールが調整に対して敏感になってしまう。反復点を平均化すれば、より明快な解決策が得られる。個々の反復点は最小点の周囲で跳ね回り続けるものの、それらの移動平均をとることで平均ゼロのノイズが相殺されるからである。Polyak and Juditsky (1992) は、$1/k$ より緩やかに減衰するステップサイズで得られた反復点を平均化すると、漸近的に最適な共分散が達成されることを示した。最小二乗法に対しては、Bach and Moulines (2013) がさらに踏み込み、**定数**のステップサイズですでに十分であることを示している。すなわち、平均化された反復点は $1/n$ のオーダーの誤差に n 個のデータ例で到達し、これには強凸性すら必要ない。

35.9.2.2 補間

古典的な確率的最適化では、最適点においても個々の訓練例がパラメータを互いに相反する方向へ引っ張り合うため、避けられないノイズフロアが生じると仮定する。これに対し、現代の過剰パラメータ化されたニューラルネットワークは、しばしば大きく異なるレジームで動作する。モデルがすべての訓練例に厳密に適合するとき、最小点では各訓練例の勾配がすべてゼロになる。するとノイズフロアは消失し、強凸な損失のもとでは、定数ステップの SGD がいかなる分散低減（variance reduction）も用いずに線形収束する（35.4.4 節）。Ma et al. (2018) は最小二乗法についてこの補間レジームを解析し、それ以上ミニバッチを大きくしても訓練が高速化しなくなる臨界バッチサイズ（critical batch size）を特定した。興味深いことに、このしきい値を支配しているのは勾配ノイズではなく、訓練例ごとの曲率とフルバッチの曲率の比である。また、勾配の分散が全勾配（full gradient）のノルムの二乗に比例して縮小するという関連条件である**強増大条件**（strong growth condition）のもとでは、SGD を加速してモメンタムと同等の収束率を達成することさえ可能になる (Vaswani et al. 2019)。

35.9.2.3 フィッシャー行列とガウス・ニュートン行列

K-FAC（35.5.5 節）などの二次近似は、フィッシャー情報行列 $\mathbf{F} = \mathbb{E}[\nabla \log p \nabla \log p^\top]$ の近似を目的とする。ここで期待値は、モデル自身の予測分布（predictive distribution）からラベルを抽出してとる (Amari 1998; Martens and Grosse 2015)。ソフトマックス交差エントロピーや二乗誤差のような標準的な損失に対しては、$\mathbf{F}$ はガウス・ニュートン行列 $\mathbf{J}^\top \mathbf{H}_{\text{out}} \mathbf{J}$ と一致する（$\mathbf{H}_{\text{out}}$ はネットワークの出力に関する損失のヘッセ行列である）。より一般に、この恒等式はネットワークの出力が指数分布族（exponential family）の尤度の自然パラメータ（natural parameter）を表すときには常に成り立つ（36.4 節）。実務上よくある落とし穴は、モデルからのサンプリングの代わりにデータセットの正解ラベルにわたって同じ外積を平均した**経験**フィッシャー行列（empirical Fisher）で代用してしまうことである。これは計算が安価であり、コードベース上でも単に「Fisher」と表記されることが多いが、一般には真のフィッシャー情報行列にもガウス・ニュートン行列にも一致しない (Martens 2020)。

35.9.2.4 ミラー降下法

AdaGrad を計量の観点から見ると（35.5.1 節）、ステップ長は二次のユークリッドノルムで測られる。この幾何構造は、あらゆる方向が同等に見える $\mathbb{R}^d$ のような制約なし空間には適しているが、各座標が非負かつ総和が 1 でなければならず乗法的な更新の方がはるかに自然である確率単体（probability simplex）の上では扱いにくい。ミラー降下法（mirror descent）は、勾配ステップの近接形式における二乗距離 $\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$ をブレグマンダイバージェンス（Bregman divergence）$D_\phi(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \nabla\phi(\mathbf{y})^\top(\mathbf{x}-\mathbf{y})$ に置き換えることで、勾配降下法の幾何構造を適応させる (Nemirovski and Yudin 1983; Beck and Teboulle 2003)。ϕ として負のエントロピーを選ぶと、更新は指数勾配則（exponentiated gradient rule）$x_i \leftarrow x_i e^{-\eta g_i} / \sum_j x_j e^{-\eta g_j}$ となる。勾配が座標ごとに有界であるとき、この選択によって 35.5.1 節で定義したリグレットは、d 次元単体上において射影勾配降下法（projected gradient descent）の $\sqrt{d}$ スケーリングからはるかに緩やかな $\sqrt{\log d}$ へと改善される。

35.9.3 制約と双対性

35.9.3.1 鞍点法

ラグランジュ双対性（35.7.2 節）で登場した鞍点は、GAN の訓練（16.1 節）をはじめとする 2 人ゲームの均衡としても直接現れる。しかし残念ながら、一方のプレイヤーには勾配降下法を、もう一方のプレイヤーには勾配上昇法を同時に適用する素朴な戦略はうまく機能しない。単純な双線形ゲーム xy においてさえ、任意の正のステップサイズに対して鞍点から外側へと渦を巻いて発散してしまうのである（演習問題 5）。外挿勾配法（extragradient method）は、一歩先を覗き見ることでこの渦を断ち切る。すなわち、試行ステップを進め、その先読み点で勾配を評価し、得られた勾配を元の反復点から適用する。リプシッツ勾配を持つ凸凹（convex–concave）問題では、その平均化された反復点は $O(1/k)$ の速さで収束する (Nemirovski 2004)。また楽観的手法（optimistic method）は、直前のステップの勾配を試行勾配の安価な代用として再利用することで、半分の勾配コストで同様の安定化効果を達成しており、Daskalakis et al. (2018) はこの発想を GAN の訓練の安定化に応用して成果を上げている。

35.9.3.2 問題クラスと内点法

35.7.5 節で手計算により解いた双対問題は二次計画問題（quadratic program）であり、これは線形計画問題、二次計画問題、二次錐計画問題（second-order cone program）、半正定値計画問題という凸最適化クラスの標準的な入れ子階層の中に位置づけられる。現代の内点法（interior-point method）は、これらすべてのクラスを多項式時間で高い精度まで解くことができる。内点法は不等式 $g_i(\mathbf{x}) \le 0$ を直接扱う代わりに、滑らかな対数バリア（logarithmic barrier）$-\frac{1}{s}\sum_i \log(-g_i(\mathbf{x}))$ を目的関数に加え、s を大きくしながら制約なし最小点を追跡することで、境界へと向かう**中心パス**（central path）を描き出す。このパス上の各点において、量 $-1/(s\,g_i(\mathbf{x}))$ は自動的に双対実行可能な乗数をなし、双対ギャップ（duality gap）m/s（m は制約の個数）を保証する。これにより、35.7 節で手作業により構成した最適性の証明書が自動的に得られる (Boyd and Vandenberghe 2004; Nesterov 2018)。

35.9.3.3 作用素分割と ADMM

35.6.4.1 節の近接勾配ステップは、滑らかな損失と単一の単純な非滑らかペナルティの和を扱う。これに対し、構造を持つ多くの問題は合成形式 $\min_{\mathbf{x}} f(\mathbf{x}) + h(\mathbf{A}\mathbf{x})$ をとる。ここで両方の項は凸であり、ともに非滑らかであってもよく、そのフェンシェル双対（Fenchel dual）は $\max_{\mathbf{y}} -f^*(-\mathbf{A}^\top\mathbf{y}) - h^*(\mathbf{y})$ となる (Rockafellar 1970)。2 つの項を切り離して個別に扱えるようにするため、補助変数 $\mathbf{z} = \mathbf{A}\mathbf{x}$ を導入し、二次ペナルティ $\frac{\tau}{2}\|\mathbf{A}\mathbf{x}-\mathbf{z}\|^2$ をラグランジアンに加

える。まず $\mathbf{x}$ について最小化し、次に $\mathbf{z}$ について最小化し（これは h に対する近接ステップに帰着する）、最後に $\mathbf{y}$ について双対上昇ステップをとる操作を交互に繰り返すと、交互方向乗数法（alternating direction method of multipliers、ADMM）が得られ、これは任意の $\tau > 0$ に対して収束する。Boyd et al. (2011) はこの分割によって大規模な最適化問題を複数の計算機へ自然に分散できることを示しており、Parikh and Boyd (2014) は代表的な正則化項や制約に対する閉形式の近接作用素をまとめた便利なカタログを提供している。

35.9.3.4 フランク・ウルフ法

射影勾配降下法では、反復ごとに制約集合への射影が必要になる。これはボックス制約や ℓ_2 球に対しては容易であるが、核ノルム球（特異値の総和が与えられた上界以下である行列の集合）の上では、1 回の射影に完全な SVD のコストがかかる。フランク・ウルフ法（Frank–Wolfe method）(Frank and Wolfe 1956) は、制約集合上で目的関数の線形近似を代わりに最小化することで射影を完全に回避する。具体的には、$\mathbf{s}_k = \operatorname{argmin}_{\mathbf{s} \in C} \nabla f(\mathbf{x}_k)^{\top} \mathbf{s}$ を求め、凸結合（convex combination）$\mathbf{x}_{k+1} = (1 - \gamma_k)\mathbf{x}_k + \gamma_k \mathbf{s}_k$ をステップサイズ $\gamma_k = 2/(k+2)$ でとる。核ノルム球上ではこの線形部分問題に必要なのは勾配の最大特異ベクトル対だけであり、確率単体上では最小の座標を 1 つ見つける操作に帰着する。この手法は $O(LD^2/k)$ の速さで収束し（D は制約集合の直径）、さらに利点として、量 $\nabla f(\mathbf{x}_k)^{\top}(\mathbf{x}_k - \mathbf{s}_k)$ が追加の計算なしに毎ステップ $f(\mathbf{x}_k) - f^{\star}$ の上界を与える (Jaggi 2013)。

35.9.4 浮動小数点

35.9.4.1 確率的丸め誤差解析

35.8.3 節における総和の最悪ケースの上界は nu として線形に増大する（n 個の項を順次加算する場合）が、実際に計測される丸め誤差は通常それよりはるかに小さい。なぜなら最悪ケース解析では、個々の丸め誤差がすべて同じ最悪の方向へ積み重なると仮定しているからである。これに対し Higham and Mary (2019) は、各ステップの丸め誤差を独立で平均ゼロの確率変数としてモデル化し、高い確率で成り立つオーダー $\sqrt{n}\,u$ の上界を証明した。これは実験で典型的に見られるランダムウォーク的な相殺と一致する。同様に重要な点として、この解析は楽観的な $\sqrt{n}\,u$ の振る舞いがいつ破綻するかも明らかにしている。すなわち、10^7 個の 0.1 を足し合わせる同項の例のように、長い区間にわたって加算がすべて同じ方向へ丸められ、丸め誤差どうしが強く相関する場合である。

35.9.4.2 増大係数と反復改良法

35.8.4 節では、部分ピボット選択付き LU 分解が実用上は後退安定であることを述べた。厳密に言えば、その後退誤差は n の低次多項式と $\rho_n u$ の積で上から抑えられる。ここで**増大係数**（growth factor）ρ_n は、消去の過程で生じる最大の中間成分と $\mathbf{A}$ の最大成分との比を測る量である。最悪ケースでは、Wilkinson の有名な病的行列によって ρ_n は 2^{n-1} にまで達する。しかし実際の計算で現れるほぼすべての行列では ρ_n は穏やかな値にとどまり、万全の保証が必要な場合でも、QR 分解を用いれば約 2 倍の演算コストで指数関数的な増大を完全に排除できる (Higham 2002; Trefethen and Bau 1997)。すべての計算に倍精度のコストを払うことなく高い精度を回復するため、混合精度の反復改良法（iterative refinement）では、$\mathbf{A}$ を fp32 で安価に分解し、残差を fp64 で評価して、その低精度の因子を用いて補正量を求める。また 35.8 節の最後の演習問題では、この基本的なループが停滞するまでに $\mathbf{A}$ がどこまで悪条件になり得るかを調べる。この低精度の因子を、一般の連立一次方程式に対する反復ソルバである GMRES の内部で前処理行列として代わりに用いれば、混合精度の反復改良法をはるかに悪条件な行列へも適用できるようになる (Carson and Higham 2018)。

35.9.5 最適化全般の参考資料

書籍

- 『Convex Optimization』（Boyd、Vandenberghe 著）：凸集合と凸関数、双対性、および内点法を扱っており、無料の PDF が公開されている。35.3 節、35.6 節、35.7 節の基本文献である。
- 『Numerical Optimization』（Nocedal、Wright 著、第 2 版、Springer）：直線探索、信頼領域法、準ニュートン法、共役勾配法、および制約付き最適化を、35.1 節、35.2 節、35.6 節よりも踏み込んで解説している。
- 『Lectures on Convex Optimization』（Nesterov 著、第 2 版、Springer）：計算量限界と加速一次手法を扱う凸最適化の講義録であり、35.1 節と 35.3 節が引用する一般的な証明を収録している。
- 『Optimization for Data Analysis』（Wright、Recht 著、Cambridge、2022 年）：データ解析向けの勾配法、加速法、確率的手法、および座標降下法による最適化を扱っており、35.1 節から 35.5 節までの副読本となる。
- 『Accuracy and Stability of Numerical Algorithms』（Higham 著、第 2 版、SIAM）：丸め誤差解析、総和計算、ならびに連立一次方程式と最小二乗法の条件性を扱っている（35.8 節）。
- 『Mathematics for Machine Learning』（Deisenroth、Faisal、Ong 著）：勾配降下法、制約付き最適化、および凸性に関する自己完結した無料の章を収録している。

講義と講義動画

- 「Stanford EE364a: Convex Optimization I」（Boyd 担当）：Boyd、Vandenberghe の凸最適化の教科書に対応するスライド、課題、ソフトウェアであり、2023 年の講義動画も公開されている（35.3 節、35.7 節）。
- 「CMU 10-725: Convex Optimization」（Tibshirani 担当）：劣勾配、近接手法、確率的手法、双対性、および ADMM に関する機械学習向けの凸最適化スライドと講義録であり、2016 年秋の講義動画も利用できる（35.4 節、35.6 節、35.7 節）。

チュートリアル、講義ノート、サーベイ

- 「Why Momentum Really Works」（Goh 著、Distill）：凸二次関数におけるモメンタムを対話的な図で解説している（35.1.3 節）。
- 「Optimization Methods for Large-Scale Machine Learning」（Bottou、Curtis、Nocedal 著）：大規模学習のための確率的勾配法による最適化理論をまとめている（35.4 節）。
- 「An overview of gradient descent optimization algorithms」（Ruder 著）：モメンタム、ネステロフの加速法、AdaGrad、RMSProp、Adam といった勾配降下法の最適化アルゴリズムを並べて比較している（35.5 節）。
- 「What Every Computer Scientist Should Know About Floating-Point Arithmetic」（Goldberg 著）：浮動小数点演算における IEEE 形式、丸め、および桁落ちを解説している（35.8 節）。
- 「Optimization (Chapter 5)」（Hardt、Recht 著『Patterns, Predictions, and Actions』）：教師あり学習における最適化を簡潔に解説している（35.1 節、35.4 節）。

Chapter 36

確率と統計学習

本編全体を通して、回帰モデルは二乗誤差（squared error）で、分類器は交差エントロピー（cross-entropy）で訓練し（2.1 節、3.1 節）、有限のテストセット上で正解率を報告し、検証スコアを比較してモデルを選択してきた（2.6 節）。これらのステップはいずれも、有限の標本（sample）から推論（inference）を引き出している。コイン投げで表が 9 回（13 回中）出た場合を考えてみよう。経験的な表の割合は $9/13 \approx 0.69$ であるが、公正なコインであっても 8 回に 1 回を超える頻度で少なくとも 9 回は表が出る。テスト正解率とは、まさにその同じコインをテスト例ごとに 1 回投げたものにほかならない。本章では、有限の標本がいつ信頼できる結論を支え、いつ判断を誤らせるのかを説明する数学を展開する。

標本に伴う不確実性を整理するには、4 つの異なる概念が土台となる。結果がどのようにばらつくかを記述する**分布**、標本を数値へ写す**推定量**（estimator）、仮想的な標本間にわたる推定量の広がりとは異なる、手元の単一の標本で条件付けた**事後分布**（posterior）、そして有限の標本における小さな誤差が新たなデータにおける小さな誤差をいつ保証するかを裏づける**汎化限界**（generalization bound）である。

本章では、これら 4 つの概念を順に取り上げる。まず、連続密度（36.1 節）、同時分布とヤコビ行列（Jacobian）による変換（36.2 節）、代表的な名前付き分布（36.3 節）、そしてそれらを統一する指数分布族（exponential family、36.4 節）を確立する。これにより、線形回帰、ロジスティック回帰、ソフトマックス回帰がいずれも「予測マイナス観測」という同じ形の勾配を共有する理由が明らかになる。次に、36.5 節ではエラーバー、仮説検定、信頼区間、ブートストラップ法を展開し、36.6 節では最尤法（maximum likelihood）から訓練損失を導出してパラメータの分散をフィッシャー曲率へと結びつける。続いて、隠れ変数を動機としてエビデンス下界（evidence lower bound）と期待値最大化法（expectation–maximization）を導入し（36.7 節）、事後分布が閉じた形を持たない場合に備えてラプラス近似、モンテカルロ法、マルコフ連鎖モンテカルロ法、変分近似を扱う（36.8 節）。最後に、36.9 節で有限標本における裾の上界と高次元における球殻への集中を証明し、36.10 節でそれらを選択後の一様収束（uniform convergence）へと拡張し（第 3.5 章）、36.11 節で分布、平滑化、較正（calibration）を画像分類へと応用する。

1.6 節の内容に馴染みのある読者は冒頭のモーメントの代数計算を流し読みしてよく、また 36.8 節と 36.11 節は最初の通読時には読み飛ばしても後続の節の理解には支障がない。エントロピーとダイバージェンスの本格的な理論は第 37 章に引き継がれ、測度論（measure theory）、時系列（time series）、およびノンパラメトリックな拡張については 36.12 節で概観する。

36.1 確率変数

1.6 節では、数え上げによって確率を構築した。サイコロには 6 つの面があり、ラベルは 10 個のクラスのいずれかに属し、事象の確率はそこに含まれる結果の確率を足し合わせたものにすぎない。しかし、ニューラルネットワークが扱う数値の大半はそのようなものではない。ピクセルの強度、初期化時に抽出される重み、あるいは拡散モデル（diffusion model）が画像に加えるガウスノイズは、いずれも区間内の任意の値を取り得るため、値を 1 つずつ列挙するには多すぎる。では、そのうちのたった 1 つの値が起こる確率はいくらだろうか。

丸いボードに向かって無造作に投げられ、ボード上のどこに刺さる確率も等しいダーツを考えよう。そして、そのダーツが中心からちょうど半径の半分の位置に刺さったとする。その起こりやすさを尋ねられたら、距離を丸めて（たとえば半径の 0.45 倍から 0.55 倍の間として）答えるだろう。これはおおよそ 10 回に 1 回の割合で起こる。精度をもう 1 桁求めれば確率は 10 分の 1 になり、さらにもう 1 桁求めればさらに 10 分の 1 になる。桁数を無限に増やすと確率は 0 になるが、それでもダーツがどこかに刺さったことは確かである。

この難点を抜け出すには、1 点の確率を問うのをやめ、その近くで確率がどれだけの速さで蓄積するかを問えばよい。区間を 10 分の 1 に狭めるたびに確率も 10 分の 1 に減るため、確率と幅の比は一定に保たれる。この単位長さあたりの確率が**密度**（density）である。密度を用いれば、和を積分に置き換えることで、離散確率のあらゆる道具を連続的に変化する量へとそのまま引き継ぐことができる。

以下の内容の一部は 1.6 節の再訪であり、連続変数へと拡張して証明を与える。平均と分散の意味づけはすでにそこで述べたため、ここではただちに連続版の形と代数的な性質へ進む。それらと並行して、密度そのもの、累積分布関数からサンプルを抽出する汎用的な手法、Markov と Chebyshev による裾の上界、そして裾があまりに重いために両方の上界が破綻する釣鐘型の密度を導入する。本節から結果を 2 つだけ持ち帰るなら、密度と累積分布関数の関係 Equation 36.6 とマルコフの不等式 Equation 36.14 を押さえておきたい。本章における残りの裾の上界はすべてここから導かれる。本節全体を通してダーツの例を追い、中心からの距離という 1 つの数値に注目する。2 つの座標や密度の変換については 36.2 節に譲る。

36.1.1 密度

密度の考え方を 3 つの段階で展開しよう。まず 1 点において、限りなく狭くなるビンの極限として捉える。次に区間上において、曲線の下の面積として捉える。最後に、その面積を蓄積する関数を通じて捉える。この関数からは、乱数サンプルを抽出する実用的な方法も得られる。

36.1.1.1 ビンから密度へ

「単位長さあたりの確率」を厳密にするため、まずダーツが指定された距離の範囲内に刺さる確率を求めよう。R を、半径 1 のボードの中心からダーツまでの距離とする。ダーツが中心から距離 r 以内に刺さるのは、半径 r の円盤に当たるときに限られる。ボードのどの部分も等しく起こりやすいため、この確率は単にボードの面積に占めるその円盤の割合となる。

$$P(R \le r) = \frac{\pi r^2}{\pi} = r^2, \qquad 0 \le r \le 1.$$

半径の半分の内側には面積の 4 分の 1 しかないため、確率も 4 分の 1 しか含まれない。したがって、幅 ϵ で中心が 0.5 のビンに入る確率は $(0.5 + \epsilon/2)^2 - (0.5 - \epsilon/2)^2 = \epsilon$ となり、冒頭の例の数値が得られる。

ビン	幅 ϵ	確率	確率 / ϵ
$[0.45, 0.55]$	0.1	0.1	1
$[0.495, 0.505]$	0.01	0.01	1
$[0.4995, 0.5005]$	0.001	0.001	1

ビンを 10 分の 1 に狭めるたびに確率も 10 分の 1 になり、比は変わらない。この極限の比こそ名前をつけたい量である。確率変数（random variable）X の点 x における**確率密度**（probability density）とは、x の近傍における単位長さあたりの確率であり、次式で表される。

$$p(x) = \lim_{\epsilon \downarrow 0} \frac{P(x \le X < x + \epsilon)}{\epsilon},$$

ただし、この極限が存在するものとする。言い換えれば、短い区間に対する確率は、おおよそ区間の長さと密度の積になる。

$$P(x \le X < x + \epsilon) \approx \epsilon\, p(x), \tag{36.1}$$

ここで誤差は ϵ よりも速く 0 に近づく。これは密度を 1 点ずつ記述したものであり、当面はこれで十分である。区間を用いた厳密な定義は次の項で与える。ダーツの例では、任意の距離 r（区間 $(0, 1)$ 内）に対して、区間 $[r, r + \epsilon)$ の確率は $(r + \epsilon)^2 - r^2 = 2r\epsilon + \epsilon^2$ である。これを幅 ϵ で割ると $2r + \epsilon$ となり、$\epsilon \to 0$ とすると次式が得られる。

$$p(r) = 2r, \qquad 0 \le r \le 1. \tag{36.2}$$

密度は中心から外縁に向かって線形に増大する。これは半径 r の円の円周の長さをそのまま反映しており、外側に行くほどボードの面積が広いという単純な理由による。$r = 0.5$ における密度は 1 であり、上の表の比と一致する。

なお、密度は確率ではない。$r = 0.9$ ではダーツの密度は 1.8 であり、$[0, 0.1]$ のどこにも等しい起こりやすさで落ちる変数はその区間上で密度 10 を持つ。合計 1 の確率を長さ 0.1 に詰め込むからである。密度は任意の非負の値をとり得るが、10 を超えられるのは合計の長さが $1/10$ 未満の集合上に限られる。さもなければ、その集合の確率が 1 を超えてしまうからである。密度は速さと同じく「変化率」だと考えるとよい。自動車の速さは距離がどれだけの速さで積み上がるかを表すものであり、すでに走った距離を表すのではない。同様に、密度が 1.8（$r = 0.9$ において）であるとは、r が 0.9 を超えて増えるときに確率がどれだけの速さで蓄積するかを表しており、点 0.9 そのものにどれだけの確率が乗っているかを表すものではない。

1 つの点は長さが 0 であるため、確率も 0 になる。任意の $\epsilon > 0$ に対して点 x は $[x, x + \epsilon)$ に含まれるため、$P(X = x)$ は高々おおよそ $\epsilon\, p(x)$ であり、ϵ が 0 に近づくにつれて消える。したがって、密度を持つ任意の確率変数に対して次が成り立つ。

$$P(X = x) = 0 \qquad \text{for every } x. \tag{36.3}$$

これにより冒頭のパズルが解ける。ダーツが距離 0.5 に刺さる確率は、他のあらゆる厳密な距離と同じく 0 であるが、それでもダーツはどこかには必ず刺さる。連続確率においては、確率 0 の事象は不可能を意味しない。

正の確率を持つ 1 つの点は**アトム**と呼ばれる。密度を持つ変数はアトムを持たない一方、サイコロの出目のような離散変数はすべてアトムから成り立つ。両方を持つ変数もある。たとえば、標

準正規分布に従う入力を受け取る ReLU ユニットの出力は、0 を確率 $\frac{1}{2}$ でとり（アトム）、残りの半分の確率を密度によって $(0, \infty)$ 上に広げる（詳細は演習問題 2 で扱う）。

上の表は厳密な面積の公式から求めたが、シミュレーションには公式は一切不要である。円形のボードにダーツを一様に投げるには、ボードを囲む正方形 $[-1, 1]^2$ に一様に投げ、円の内側に刺さったものだけを残せばよい。これでボードのどの部分も等しい確率になる。以下では、3 つのビンのそれぞれに入ったダーツの割合を数え、ビンの幅で割る。

```
rng = np.random.default_rng()
xy = rng.uniform(-1, 1, size=(4 * 10**6, 2))          # darts on the square
on_board = (xy**2).sum(axis=1) <= 1                   # keep those on the board
r_dart = np.linalg.norm(xy[on_board], axis=1)         # distances R
for width in [0.1, 0.01, 0.001]:
    prob = np.mean(np.abs(r_dart - 0.5) <= width / 2)  # fraction in the bin
    print(f'width {width:<5}  probability {prob:.5f}  ratio {prob / width:.3f}')
```

```
width 0.1    probability 0.09983  ratio 0.998
width 0.01   probability 0.01002  ratio 1.002
width 0.001  probability 0.00099  ratio 0.990
```

予想どおり、ビンを 10 分の 1 に狭めるたびに割合も 10 分の 1 に減り、比は $1 = p(0.5)$ の近くに保たれる。しかし、厳密な表からは見えなかったのは、極端に狭いビンが実際にはどれほどノイズの影響を受けるかという点である。400 万本のダーツのうち約 3.1 百万本がボードに刺さり、そのうち約 3,000 本しか幅 0.001 のビンに入らないため、最後の比は実行ごとに数パーセント変動する。

36.1.1.2 面積としての確率

ここまでは、微小な区間の確率から密度を読み取ってきた。今度は逆向きに考えてみよう。密度が与えられたとき、ある区間全体の確率はどう求めればよいだろうか。実数直線をビン $[\epsilon i, \epsilon(i+1))$（幅 ϵ）に分割しよう。Equation 36.1 より、各ビンの確率はおおよそ $\epsilon\, p(\epsilon i)$ であり、ビンは互いに素であるため確率は足し合わせられる。幅と高さの積を足し合わせるとリーマン和（Riemann sum）になり、これは曲線の下の面積、すなわち定積分（34.6 節）へと ϵ が 0 に近づくにつれて収束する。X は必ず何らかの実数値をとるため、すべてのビンを合わせた確率は 1 になる。

$$1 = P(X \in \mathbb{R}) \approx \sum_i \epsilon\, p(\epsilon i) \;\xrightarrow[\epsilon \to 0]{}\; \int_{-\infty}^{\infty} p(x)\, dx.$$

a と b の間にあるビンだけを足し合わせれば、区間の確率はその区間上での密度の下の面積にほかならないことが示唆される。厳密に言えば、この議論は証明ではなく動機づけである。ビンはおおよそ $1/\epsilon$ 個あり、個々の誤差が ϵ に比べて小さくても、その合計が無視できるほど小さくなるのは誤差が x にわたって一様に縮む場合に限られるからである。議論を確かな土台に載せるため、この面積の公式を正式な定義としよう。

定義　確率変数 X が p を**密度に持つ**とは、$p \geq 0$ であり、かつすべての $a < b$ に対して次式が成り立つことをいう。

$$P(X \in (a, b]) = \int_a^b p(x)\, dx. \tag{36.4}$$

$a \to -\infty$ および $b \to \infty$ とすれば、いかなる密度もその下の全面積が 1 であることが確かめられる。実際、非負性（$p \geq 0$）と全面積が 1 であるという 2 つの条件さえ満たせば、任意の関数が有効な密度となる。Figure 36.1 に、2 つの峰を持つ密度に対するこの定義を示す。ダーツの例もこの条件を満たしている。$0 \leq a < b \leq 1$ に対して、積分 $\int_a^b 2r\,dr = b^2 - a^2$ は面積から求めた確率 $P(a < R \leq b)$ を再現し、$2r$ が確かにその密度であることが確認できる。とくに、半径の外側半分に刺さる確率は $\int_{0.5}^1 2r\,dr = 1 - 0.25 = 0.75$ であり、実にボードの 4 分の 3 が中心から半径の半分より外側にある。

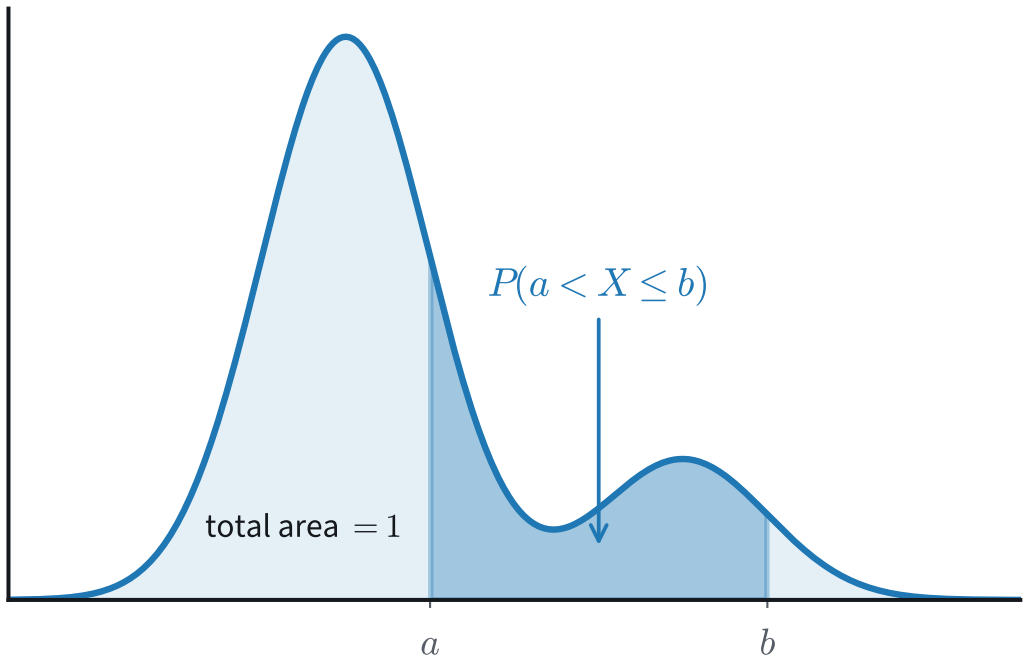


Figure 36.1: 確率密度 $p(x)$。曲線の下の全面積は 1 であり、X が a から b までの区間に入る確率は、その上の網掛け部分の面積 $\int_a^b p(x)\,dx$ である。密度そのものは確率ではなく（1 を超えることもある）、区間上の積分だけが確率となる。

36.1.1.3 累積分布関数

値そのものが真の確率であり、離散変数と連続変数を同じように扱える関数を用いると便利なことが多い。X の**累積分布関数**（cumulative distribution function、CDF）は、X が x 以下の値をとる確率を与える。X が密度を持つとき、これは単に x より左側にある密度の下の面積である。

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(t)\,dt. \tag{36.5}$$

言い換えれば、F は実数直線を左から右へ走査しながら確率を蓄積していく。その値は 0（$x \to -\infty$ の極限）から始まり、1（$x \to \infty$ の極限）へと上昇する。非負の密度 p は面積を増やすことしかしないため減少することは決してなく、X が密度を持つときは常に F は連続である。ダーツの例については冒頭ですでに計算したとおり、$F(r) = 0$（$r < 0$）、$F(r) = r^2$

$(0 \le r \le 1)$、$F(r) = 1$（$r > 1$）である。この関数こそが出発点であり、密度 $2r$ はそこから直接導かれた。

密度とは異なり、累積分布関数はあらゆる確率変数に対して定義される。0 または 1 を返す公平なコインの累積分布関数は次のようになる。

$$F(x) = \begin{cases} 0 & x < 0, \\ \frac{1}{2} & 0 \le x < 1, \\ 1 & x \ge 1, \end{cases}$$

これは各アトムの確率の分だけ上方にジャンプする**階段関数**である。$F(x) = P(X \le x)$ はちょうど x の上にあるアトムも含むため、この関数は各ジャンプにおいて上側の値をとる。より一般に、すべての累積分布関数は**右連続**（right-continuous）であり、すなわち $\lim_{y \downarrow x} F(y) = F(x)$ を満たす。混合型の変数は両方の振る舞いを併せ持つ。先ほど挙げた ReLU の出力の累積分布関数は、値が 0（$x < 0$）から始まり、$\frac{1}{2}$ へと $x = 0$ でジャンプし、そこから 1 に向かって滑らかに上昇する。したがって、単一の関数によって離散・連続・混合の確率変数が統一される。Figure 36.2 にそれぞれの例を示す。

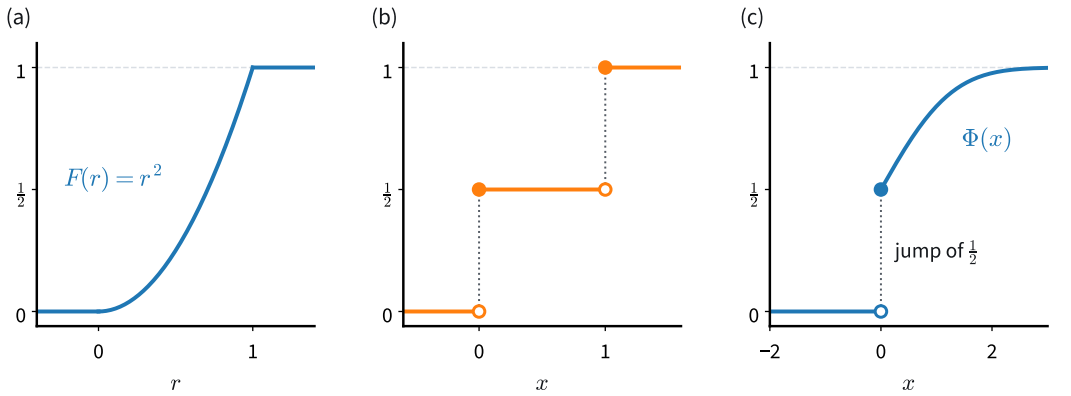


Figure 36.2: 3 つの累積分布関数。(a) 中心からダーツまでの距離：$F(r) = r^2$ は 0 から 1 まで連続的に上昇する。(b) 0 または 1 を返す公平なコイン：各アトムで $\frac{1}{2}$ ずつジャンプし、そこで上側の値をとる階段関数（黒丸）。(c) 標準ガウス分布の ReLU：F は 0（$x < 0$）から始まり、$\frac{1}{2}$ へと $x = 0$ でジャンプし、そこからはガウス分布の累積分布関数 $\Phi(x)$ に従う。

密度が存在するとき、累積分布関数と密度は同じ情報を持つ。p を積分すれば F が得られ、F を微分すれば p が復元される。これにより、冒頭で見た縮小するビンの極限が定理として確立される。

命題（密度は累積分布関数の導関数）　**X が密度 p（点 x で連続）を持つならば、F は x で微分可能であり、次式が成り立つ。**

$$F'(x) = p(x). \tag{36.6}$$

とくに、単位長さあたりの確率 $P(x \le X < x + \epsilon)/\epsilon$ は、$p(x)$ へと $\epsilon \downarrow 0$ の極限で収束する。

証明　Equation 36.5 より、F は p の累積面積を表す関数である。微分積分学の基本定理によれば、累積面積を表す関数の導関数（derivative）は、被積分関数が連続なすべての点で被積分関数

そのものに一致する（34.6 節）。後半の主張については、1 点 x の確率は高々 $\int_{x-\delta}^{x} p(t)\,dt$（任意の $\delta > 0$ に対して）であり、これらの積分は δ とともに 0 に収束する。したがって端点を区間に入れるか除くかで確率は変わらず、$P(x \le X < x+\epsilon) = F(x+\epsilon) - F(x)$ となるため、この比は単に F の差分商である。■

ダーツの例では、$F'(r) = 2r$ が $(0,1)$ 上で成り立ち、ビンから求めた密度が復元される。Figure 36.3 は 2 つの関数を並べて示したものであり、左図の網掛け面積 $\int_a^b p$ は右図の縦方向の上昇量 $F(b) - F(a)$ に等しい。

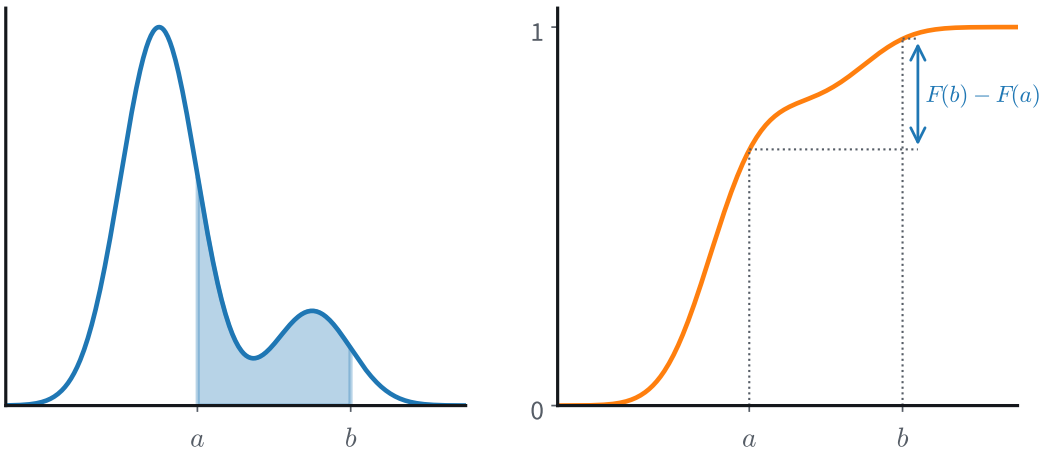


Figure 36.3: 左：密度 p と、a から b までの区間上で網掛けされた面積。右：対応する累積分布関数 $F(x) = \int_{-\infty}^{x} p$。0 から 1 へ増加する曲線であり、上昇量 $F(b) - F(a)$ は左図の網掛け面積に等しい。密度は累積分布関数の傾き $(F' = p)$ であり、累積分布関数は密度の下の累積面積である。

36.1.1.4 分位点と逆変換サンプリング

コンピュータを使えば、0 と 1 の間のどこにも等しい確率で落ちる乱数を手軽に利用できる。ダーツをはじめ任意の連続分布をシミュレーションするには、そうした一様乱数を目的の分布からのサンプルへと変換する方法が必要になる。

構成要素となるのは、あらゆる密度のなかで最も単純なものである。確率変数 U が $[0,1]$ **上で一様**であるとは、その密度が値 1 を $[0,1]$ 上でとり、それ以外では 0 であることをいう。このとき $[0,1]$ の任意の部分区間の確率はその長さに等しく、累積分布関数は単に $F_U(u) = u$ $(0 \le u \le 1)$ となる。

累積分布関数は値 x を確率水準 u へと写す。この写像を逆転させ、指定した割合 u の確率がそれ以下に収まるような値を問うことで、**分位点関数**（quantile function）が得られる。F が連続であり、区間 I（$0 < F < 1$ を満たす範囲）上で狭義単調増加であるとき、この関数は I を $(0,1)$ の上へ 1 対 1 に写し、その逆関数 $F^{-1} : (0,1) \to I$ が分位点関数となる。$u = 0.25, 0.5, 0.75$ における値は**四分位点**（quartile）と呼ばれ、真ん中の四分位点 $F^{-1}(0.5)$ は**中央値**（median）と呼ばれる。ダーツの例では $F(r) = r^2$ の逆関数は $F^{-1}(u) = \sqrt{u}$ となり、四分位点は 0.5、0.707、0.866 となる。ダーツの中央値は 0.707 であり、半径の半分よりかなり外側にある。ボードの 4 分の 3 が 0.5 より外側にあるためである。

注目すべきことに、分位点関数は汎用的なサンプラでもある。

命題（逆変換サンプリング） F **を連続な累積分布関数とし、区間** I（$0 < F < 1$ **を満たす範囲）上で狭義単調増加であるとする。また、**U **を** $[0,1]$ **上で一様な確率変数とする。このとき、**$X = F^{-1}(U)$ **は累積分布関数** F **を持つ。**

証明 確率 1 で U は $(0,1)$ に属するため、X は定義され I に属する。点 x が I に属するとき、事象 $F^{-1}(U) \le x$ と $U \le F(x)$ は一致する。なぜなら F は I 上で増加関数だからである。また、U は $[0,1]$ 上で一様であるため、2 つ目の事象の確率は区間 $[0, F(x)]$ の長さに等しい。

$$P(F^{-1}(U) \le x) = P(U \le F(x)) = F(x).$$

点 x が I より左側にあるときは $F(x) = 0$ であり常に $X > x$ が成り立つ。一方、点 x が I より右側にあるときは $F(x) = 1$ であり常に $X \le x$ が成り立つ。■

なお、この仮定のもとでは、区間の外側で F が平坦であってもよい。実際、ダーツの累積分布関数もそうなっている（$F = 0$（$r < 0$ のとき）かつ $F = 1$（$r > 1$ のとき））。F^{-1} を**一般化逆関数**（generalized inverse）$F^{-}(u) = \min\{x : F(x) \ge u\}$ に置き換えれば、この仮定自体が不要となり、階段状の累積分布関数からもサンプリングできるようになる。

ダーツに対して逆変換サンプリング（inverse-transform sampling）を適用すると $R = \sqrt{U}$ となる。近道をして $R = U$ と置きたくなるかもしれないが、それがなぜ誤りかはコードを 1 行も実行せずにすぐわかる。$R = U$ とするとダーツの半分が $r < 0.5$ の内側に入ってしまうが、本来そこに入るべき割合は 4 分の 1 にすぎない。Figure 36.4 に、2 つの峰を持つ密度におけるこの構成の幾何学的な意味を示す。密度が高い場所では累積分布関数が急勾配で立ち上がるため、縦軸上で等間隔に並んだ水準は、横軸上では密集したサンプルの塊へと圧縮される。

これで、ダーツの距離を生成するまったく異なる 2 つの方法が得られた。正方形にダーツを投げてボード上のものだけを残す方法と、単に $\sqrt{U}$ を計算する方法である。以下のセルでは、素朴な近道である $R = U$ も並べて両者を比較する。

```
u = rng.uniform(size=10**6)
r = np.sqrt(u)                                          # R = F^{-1}(U)
print(f'P(R < 0.5): darts {np.mean(r_dart < 0.5):.3f}, sqrt(U) '
      f'{np.mean(r < 0.5):.3f}, naive U {np.mean(u < 0.5):.3f}')
print(f'mean distance: darts {r_dart.mean():.3f}, sqrt(U) {r.mean():.3f}')
```

```
P(R < 0.5): darts 0.250, sqrt(U) 0.250, naive U 0.500
mean distance: darts 0.667, sqrt(U) 0.667
```

2 つの正しい手法は一致する。どちらも距離の 4 分の 1 が 0.5 未満に入り、平均距離は次節で導出する 0.667 の近くになる。素朴な近道である U はサンプルの半分を 0.5 未満に入れてしまう。中心付近より外縁付近のほうがボードの面積がはるかに広いという事実を無視して、距離を $[0,1]$ 全体に均等に広げてしまうからである。

36.1.2 平均と分散

密度は分布全体を完全に規定するが、分布の中心がどこにあり、どれほど広く散らばっているかという 2 つの要約値だけで、知りたいことの大半がわかる場合が多い。1.6 節では離散変数に対してこの両者を導入し、その動機づけを与えた。ここではそれらを密度へと拡張し、本節以降で依拠する代数的な規則を証明する。

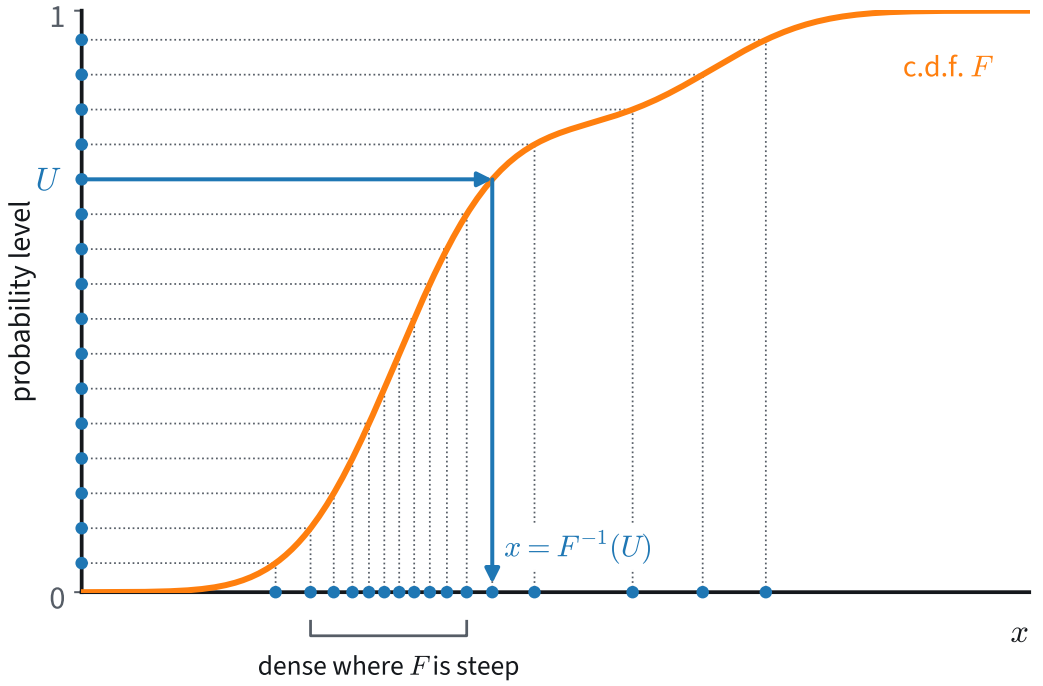


Figure 36.4: 逆変換サンプリング。縦軸（左）上の一様な水準 U が、累積分布関数 F で折り返されて横軸上の値 $x = F^{-1}(U)$ へと写される。F の傾きが急な場所、すなわち密度 $p = F'$ が大きい場所では、一様に広がった水準が密なサンプルの集まりへと圧縮されるため、折り返された点のヒストグラムは p を再現する。

36.1.2.1 平均

平均は X の各値をその確率で重み付けして平均したものである。密度を持つ変数では、先ほどと同様に実数直線を幅 ϵ のビンに切り分けると、位置 ϵi のビンはその値 ϵi と確率（おおよそ $\epsilon\, p(\epsilon i)$）の積だけ寄与し、その和は積分へと収束する。したがって、X の**平均**、すなわち**期待値**（expectation）は次式で与えられる。

$$\mu_X = \mathbb{E}[X] = \sum_i x_i\, p_i \quad \text{(discrete)}, \qquad \mu_X = \mathbb{E}[X] = \int_{-\infty}^{\infty} x\, p(x)\, dx \quad \text{(density)}. \tag{36.7}$$

離散の場合と同様、平均は釣り合いの点として働く。長さに沿って線密度 p を持つ細い棒は μ_X で釣り合う。この定義が意味を持つためには、離散の場合には $\sum_i |x_i|\, p_i < \infty$、密度の場合には $\int |x|\, p(x)\, dx < \infty$ が成り立つ必要がある。この条件が破れると平均は存在しない。実際、そのような具体的な密度にまもなく出会うことになる。同じ確率の重みを用いれば、任意の関数 $g(X)$ の平均も直接求めることができる。

$$\mathbb{E}[g(X)] = \sum_i g(x_i)\, p_i, \qquad \mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)\, p(x)\, dx,$$

ここでも $|g|$ の和または積分が有限であることを条件とする。これがいかに便利であるかに注目してほしい。g の値を X の確率で重み付けするだけでよく、変換後の変数 $g(X)$ の分布をあらかじめ求める必要がないのである。この規則はあまりにも自然に使えるため、$\mathbb{E}[g(X)]$ の定義ではなく定理であることをつい忘れてしまうことから、**無意識な統計学者の法則**（law of the unconscious statistician）という有名な呼び名がついている。なお、$\mathbb{E}[g(X)]$ は一般に $g(\mathbb{E}[X])$ とは異なる。g が凸関数であるとき、両者のどちらが大きいかを教えてくれるのがイェンセンの不等式（35.3.3.1 節）である。

ダーツの例では次のようになる。

$$\mathbb{E}[R] = \int_0^1 r \cdot 2r\, dr = \tfrac{2}{3}.$$

平均距離は 0.667 であり、0.5 ではない。これもやはり中心付近より外縁付近のほうがボードの面積が広いためである。また、平均が中央値 0.707 よりわずかに小さいことにも注目しよう。中心付近に刺さる比較的少数のダーツは、真ん中の順位の値をずらす以上に平均値を下方向へ引き下げるのである。

期待値の性質のうち最も頻繁に用いるのが線形性である。1.6 節でもすでに用いたが、ここで証明を与えよう。

命題（期待値の線形性）　平均が存在する確率変数 X, Y と定数 c, d に対して、次式が成り立つ。

$$\mathbb{E}[cX + d] = c\,\mathbb{E}[X] + d, \qquad \mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]. \tag{36.8}$$

証明　どちらの恒等式も、平均を定義する和（または積分）の線形性から従う。1 つ目の式については、$\mathbb{E}[cX+d] = \sum_i (cx_i + d)p_i = c\sum_i x_i p_i + d\sum_i p_i = c\,\mathbb{E}[X] + d$ が $\sum_i p_i = 1$ を用いて得られる。2 つ目の式については、p_{ij} を $(X, Y) = (x_i, y_j)$ となる確率、すなわち 1.6 節と

同様の組の同時分布（joint distribution）とする。和を分割し、周辺確率 $\sum_j p_{ij} = P(X = x_i)$ と $\sum_i p_{ij} = P(Y = y_j)$ に着目すれば、次式が得られる。

$$\mathbb{E}[X + Y] = \sum_{i,j} (x_i + y_j)\, p_{ij} = \sum_{i,j} x_i\, p_{ij} + \sum_{i,j} y_j\, p_{ij} = \mathbb{E}[X] + \mathbb{E}[Y].$$

密度の場合も、同時密度（p_{ij} の連続版であり 36.2 節で定義する）の上での積分によってまったく同じ計算が成り立つ。■

決定的に重要なのは、2 つ目の恒等式には独立性が一切不要だという点である。X と Y がどれほど強く互いに従属していても、和の平均は平均の和に等しい。データセット上の平均として期待損失をとるたびに、この事実に依拠している。

36.1.2.2 分散と標準偏差

平均は分布がどこに位置するかを示すが、その周りにどれほど密に集まっているかについては何も語らない。1.6 節と同様、広がりの程度は平均からの二乗距離の平均で測り、これを**分散**（variance）と呼ぶ。

$$\sigma_X^2 = \operatorname{Var}(X) = \mathbb{E}\big[(X - \mu_X)^2\big], \tag{36.9}$$

分散は $\mathbb{E}[X^2] < \infty$ である限り定義される。二乗を展開し、線形性 Equation 36.8 を μ_X を定数とみなして適用すると、通常はるかに計算しやすい形、すなわち「二乗の平均から平均の二乗を引いた式」が得られる。

$$\operatorname{Var}(X) = \mathbb{E}\big[X^2 - 2\mu_X X + \mu_X^2\big] = \mathbb{E}[X^2] - 2\mu_X\, \mathbb{E}[X] + \mu_X^2 = \mathbb{E}[X^2] - \mu_X^2. \tag{36.10}$$

したがって、密度を持つ変数の平均と分散は次の 2 つの積分で与えられる。

$$\mu_X = \int_{-\infty}^{\infty} x\, p(x)\, dx, \qquad \sigma_X^2 = \int_{-\infty}^{\infty} x^2 p(x)\, dx - \mu_X^2. \tag{36.11}$$

ダーツの例では $\mathbb{E}[R^2] = \int_0^1 r^2 \cdot 2r\, dr = \tfrac{1}{2}$ であるため、$\operatorname{Var}(R) = \tfrac{1}{2} - \tfrac{4}{9} = \tfrac{1}{18} \approx 0.056$ となる。また、一様変数 U（$[0,1]$ 上）では $\mu_U = \int_0^1 u\, du = \tfrac{1}{2}$ であり、$\operatorname{Var}(U) = \tfrac{1}{3} - \tfrac{1}{4} = \tfrac{1}{12}$ となる。

単位のリスケーリングやデータの中心化はアフィン変換（affine transformation）である。アフィン写像 $cX + d$ のもとでは、平行移動 d は分散から消え、スケール因子 c は二乗されて外に出る。実際、線形性より $\mu_{cX+d} = c\mu_X + d$ となるため、平均からの偏差は $(cX + d) - \mu_{cX+d} = c(X - \mu_X)$ となり、これを二乗して期待値をとれば次式が得られる。

$$\operatorname{Var}(cX + d) = c^2 \operatorname{Var}(X). \tag{36.12}$$

さらに 2 つの性質を心に留めておくとよい。第 1 に、分散は決して負にならず、分散が 0 になるのは X が確率 1 で定数であるとき、かつそのときに限る。非負の量 $(X - \mu_X)^2$ の期待値が 0 になり得るのは、その量が確率 1 で 0 である場合だけだからである。第 2 に、互いに独立な確率変

数の分散は足し合わせられる（1.6 節）。この事実は、変数同士が相互作用するときに独立性に代わって現れる共分散（covariance）の項とともに、36.2.2.1 節で証明する。

分散は偏差を二乗するため、ドルで測った利益に対して「ドルの二乗」になるように、単位も二乗になる。その平方根をとれば元の単位に戻り、**標準偏差**（standard deviation）が得られる。

$$\sigma_X = \sqrt{\operatorname{Var}(X)}, \tag{36.13}$$

標準偏差はアフィン写像のもとで $\sigma_{cX+d} = |c|\,\sigma_X$ と線形にスケールする。ダーツの例では、ボードの半径を単位として $\sigma_R = \sqrt{1/18} \approx 0.236$ である。

36.1.3 分布の裾

標準偏差は、平均の周りでの典型的な変動に対する自然なスケールを定める。次に生じる自然な問いは、平均だけ、あるいは平均と分散だけがわかっているとき、遠く離れた裾にどれほどの確率が乗り得るかということである。2 つの古典的な不等式が、あらゆる分布に対して一度にこの問いに答える。どちらの上界もかなり大まかであるが、驚くべきことに追加の仮定なしにはこれ以上鋭くできない最良の評価であることが、単純な 3 点分布によって示される。続いて、平均も分散も持たない釣鐘型の密度を取り上げ、これら 2 つの仮定が何を食い止めていたのかを明らかにする。

36.1.3.1 マルコフの不等式とチェビシェフの不等式

最初の上界は平均のみを用い、任意の非負の確率変数に適用できる。証明はわずか 1 行で済むにもかかわらず、本章におけるほぼすべての裾の上界は、うまく選んだ変数の関数にこの不等式を適用することで導かれる。

命題（マルコフの不等式） $X \geq 0$ **が平均を持ち、**$t > 0$ **であるとき、次式が成り立つ。**

$$P(X \geq t) \leq \frac{\mathbb{E}[X]}{t}. \tag{36.14}$$

証明 $\mathbf{1}_{X \geq t}$ を、値が 1（$X \geq t$ のとき）または 0（それ以外のとき）をとる指示変数とする。このとき、すべての結果に対して $X \geq t\,\mathbf{1}_{X \geq t}$ が成り立つ。実際、$X \geq t$ のときは右辺が $t \leq X$ となり、それ以外のときは $0 \leq X$ となる。両辺の期待値をとり、$\mathbb{E}[\mathbf{1}_{X \geq t}] = P(X \geq t)$ を用いれば $\mathbb{E}[X] \geq t\,P(X \geq t)$ が得られる。■

幾何学的には、$[0, \infty)$ 上において 0（t 未満）および t（t 以上）をとる階段関数は常に直線 $y = x$ の下側にあるため（Figure 36.5）、その平均値 $t\,P(X \geq t)$ も X の平均値の下側になければならない。t を平均の 10 倍に設定すれば、ただちに便利な経験則が得られる。所得、ファイルサイズ、勾配ノルムのいずれを扱う場合でも、非負の母集団において平均の 10 倍を超えられるのは高々 10 分の 1 である。

マルコフの不等式は平均しか使わないため、必然的に大まかな評価となる。これを二乗偏差 $(X - \mu_X)^2$（その平均は分散である）に適用すれば、両側の裾を同時に抑えられる。この結果は Bienaymé と Chebyshev によるものである (Chebyshev 1867)。

命題（チェビシェフの不等式） **有限かつ非ゼロの分散を持つ任意の** X **と任意の** $\alpha > 0$ **に対して、次式が成り立つ。**

$$P(|X - \mu_X| \geq \alpha \sigma_X) \leq \frac{1}{\alpha^2}. \tag{36.15}$$

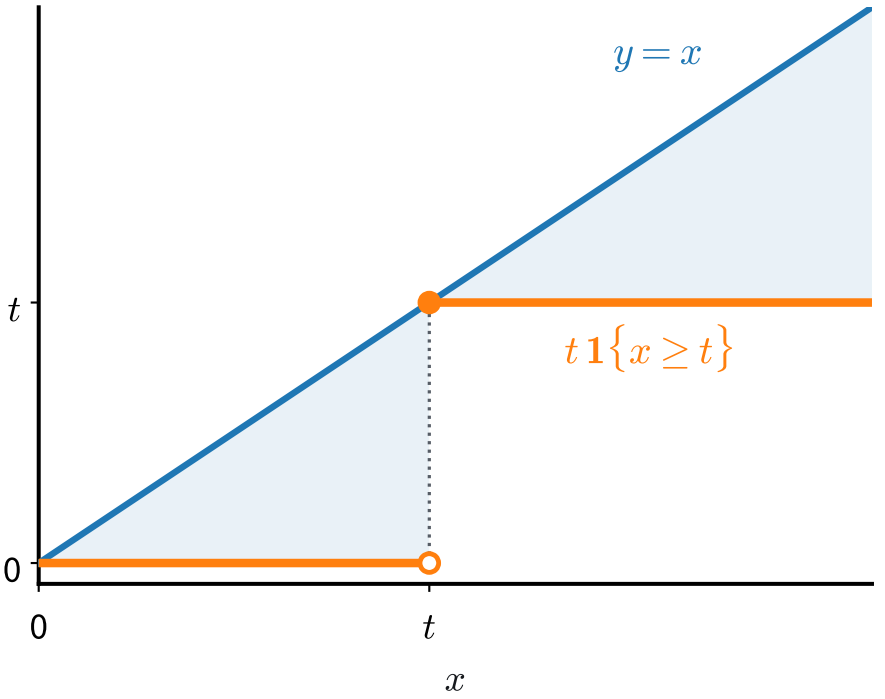


Figure 36.5: マルコフの不等式の幾何学的解釈。$x \geq 0$ において階段関数 $t\,\mathbf{1}\{x \geq t\}$（オレンジ）は直線 $y = x$（青）の下側にあるため、その平均 $t\,P(X \geq t)$ は $\mathbb{E}[X]$ を超えられない。網掛けされた隙間が、この上界で捨てられている余裕分に相当する。

証明　変数 $Z = (X - \mu_X)^2$ は非負で平均 $\mathrm{Var}(X) = \sigma_X^2$ を持ち、事象 $|X - \mu_X| \geq \alpha\sigma_X$ は事象 $Z \geq \alpha^2\sigma_X^2$ と同一である。$\sigma_X > 0$ より、しきい値 $t = \alpha^2\sigma_X^2$ は正であるため、マルコフの不等式 Equation 36.14 から $P(|X - \mu_X| \geq \alpha\sigma_X) \leq \sigma_X^2/(\alpha^2\sigma_X^2) = 1/\alpha^2$ が得られる。■

とくに、いかなる分布であっても、平均から標準偏差の 10 倍以上離れられるのは高々 1% である。同じ証明を絶対的なしきい値 $\epsilon > 0$（$\alpha\sigma_X$ の代わり）で実行すれば、有限の分散のみを仮定し、退化したケース $\sigma_X = 0$ も含む同値な形が得られる。

$$P(|X - \mu_X| \geq \epsilon) \leq \frac{\mathrm{Var}(X)}{\epsilon^2}.$$

ダーツの例では $\mu_R = 0.667$、$\sigma_R = 0.236$ である。平均から標準偏差の 2 倍下側は 0.195 であり、2 倍上側はボードの外に出てしまうため、$P(|R - \mu_R| \geq 2\sigma_R) = P(R \leq 0.195) = 0.195^2 \approx 0.038$ となり、チェビシェフの上界である $\frac{1}{4}$ よりはるかに小さい。

ダーツに対して $\frac{1}{4}$ がこれほど緩いことを踏まえると、あらゆる分布に成り立つ上界をもっと鋭くできるだろうか。さらなる仮定なしには不可能であり、わずか 3 点のみに台を持つ分布がそれを証明する。X が値 $a - 2$ と $a + 2$ をそれぞれ確率 q でとり、値 a を残りの確率 $1 - 2q$（ただし $0 < q \leq \frac{1}{2}$）でとるとしよう。これを **3 アトムの例**と呼ぶ。対称性より平均は a であり、分散は $\mathrm{Var}(X) = \mathbb{E}[(X - a)^2] = 4q + 4q = 8q$ となるため、$\sigma_X = 2\sqrt{2q}$ である。ここで $\alpha = 2$ とする。q が小さくなると、しきい値 $2\sigma_X = 4\sqrt{2q}$ もそれに伴って縮むが、外側の 2 つのアトムは距離 2（中心 a から）の位置に固定されたままである。Figure 36.6 に、その結果生じる 3 つの領域を示す。$q > \frac{1}{8}$ のとき、しきい値は 2 を超えるため、外側のアトムはどちらも裾に入らず確率は 0 となる。$q = \frac{1}{8}$ では、しきい値がちょうど 2 に一致して $\sigma_X = 1$ となり、$a \pm 2$ が平均から標準偏差の 2 倍の位置に並ぶ。このとき外側の両アトムが対象となり、合計確率は $2q = \frac{1}{4}$ となってチェビシェフの上界で等号が成立する。$q < \frac{1}{8}$ のときも外側の両アトムはしきい値の外側にあるが、合わせた確率質量 $2q$ は $\frac{1}{4}$ を下回る。

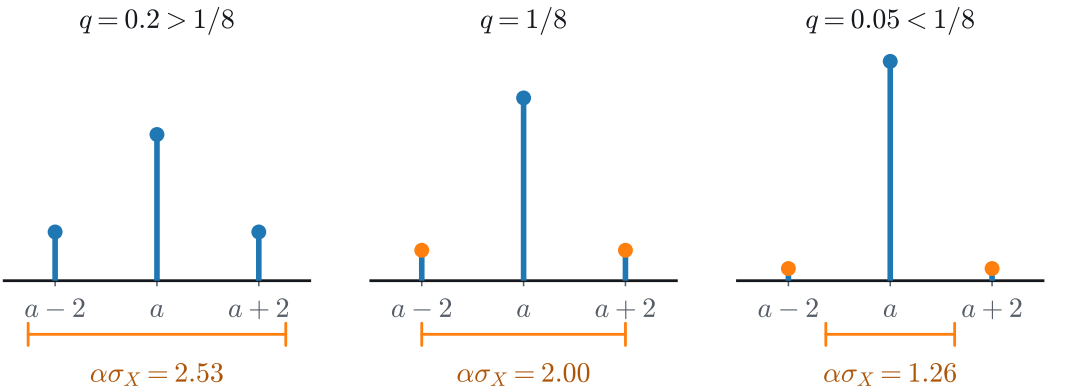


Figure 36.6: 3 アトムの例におけるチェビシェフの上界：確率質量 q、$1 - 2q$、q が点 $a - 2$、a、$a + 2$ の縦棒として描かれており、しきい値距離 $\alpha\sigma_X = 4\sqrt{2q}$（$\alpha = 2$ の場合）が $a - \alpha\sigma_X$ から $a + \alpha\sigma_X$ までの水平ブラケットで示されている。3 つのパネルはそれぞれ $q > 1/8$、$q = 1/8$（上界で等号が達成される場合）、$q < 1/8$ の領域を表す。オレンジ色のアトムはしきい値以上にあるため、事象 $|X - a| \geq \alpha\sigma_X$ に数えられる。

同じ構成ですべての $\alpha \geq 1$ に対しても決着がつく。$q = 1/(2\alpha^2)$（高々 $\frac{1}{2}$ である）を選ぶと $\sigma_X = 2\sqrt{2q} = 2/\alpha$ となるため、しきい値 $\alpha\sigma_X = 2$ が外側のアトムの真上に来て

$P(|X-a| \geq \alpha\sigma_X) = 2q = 1/\alpha^2$ となる。ここで等号が成立するため、$1/\alpha^2$ より小さい定数ではすべての分布に対して成り立たないことが証明される（$\alpha < 1$ のときは上界が 1 を超えるため何も主張しない）。

したがって、チェビシェフの不等式は与えられた分散を持つあらゆる分布のなかでの最悪ケースを捉えている。普段目にする分布に対してはかなり緩い評価になる。平均 μ、標準偏差 σ のガウス分布、すなわち密度 $e^{-(x-\mu)^2/(2\sigma^2)}/(\sigma\sqrt{2\pi})$ を持つ釣鐘曲線（詳細は 36.3.2.3 節で学ぶ）では、標準偏差の 2 倍や 3 倍の外側にある確率はチェビシェフの上界が許す値よりはるかに小さい。

平均からの距離	チェビシェフの上界	ガウス分布
2σ 以上	25%	4.6%
3σ 以上	11%	0.27%

最悪ケースの分布は、3 アトムの例がそうであるように、裾の確率質量をすべて単一の距離に集中させる。典型的な分布では裾が広がりながら急速に薄くなっていく。

36.1.3.2 重い裾

チェビシェフの不等式には有限の分散が必要であり、平均そのものにも $\int |x|\,p(x)\,dx < \infty$ が必要である。驚くべきことに、一見すると滑らかで無害に見える釣鐘曲線でも、この両方の条件が破れることがある。**コーシー分布**は次の密度を持つ。

$$p(x) = \frac{1}{\pi(1+x^2)}.$$

$\int_{-\infty}^{\infty} \frac{dx}{1+x^2} = [\arctan x]_{-\infty}^{\infty} = \pi$ であるため、これは有効な密度である。裾が $e^{-x^2/2}$ のように急落するガウス分布とは異なり、コーシー分布の裾は $1/(\pi x^2)$ のように代数的にしか減衰しない。まず二次モーメント（second moment）が破綻する。$[-L, L]$ 上で積分すると次式が得られる。

$$\int_{-L}^{L} x^2 p(x)\,dx = \frac{2}{\pi}(L - \arctan L),$$

この値は 5.43、62.7、636（それぞれ $L = 10$、100、1000 のとき）となり、被積分関数 $x^2/(\pi(1+x^2))$ が非ゼロの定数 $1/\pi$ に近づくため限りなく増大する。したがって $\mathbb{E}[X^2] = \infty$ となり、コーシー分布は有限の分散を持たない。

平均はさらに微妙な形で破綻する。被積分関数 $x\,p(x)$ は奇関数であるため、対称性から平均は 0 だと結論づけたくなるし、実際 $\int_{-L}^{L} x\,p(x)\,dx = 0$ がすべての L に対して成り立つ。しかし、期待値の定義には絶対収束が必要であり、

$$\int_{-L}^{L} |x|\,p(x)\,dx = \frac{1}{\pi}\log(1+L^2)$$

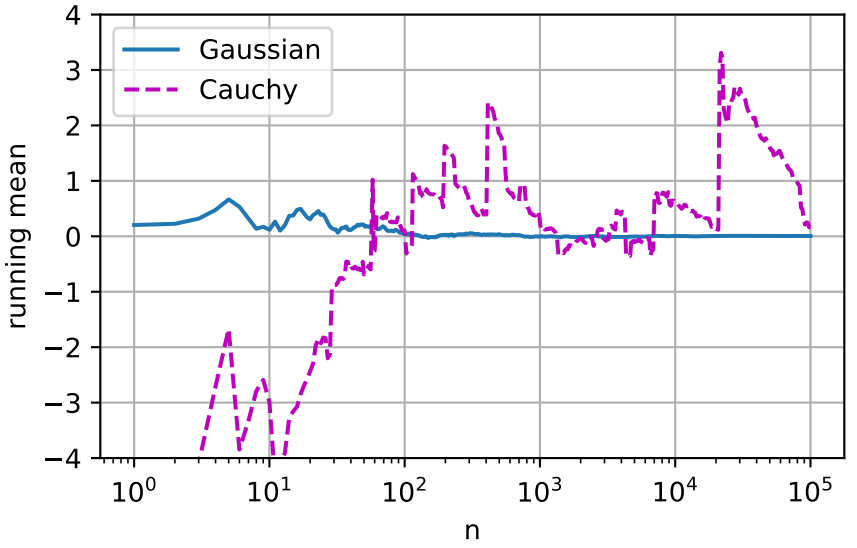
は $L \to \infty$ とするにつれて限りなく増大する。その結果、コーシー分布の平均は存在しない。絶対収束が成り立たない場合、積分の値は極限のとり方に依存してしまう。たとえば非対称な区間 $[-L, 2L]$ 上で積分すると、次のようになる。

$$\int_{-L}^{2L} x\,p(x)\,dx = \frac{1}{2\pi}\log\frac{1+4L^2}{1+L^2} \xrightarrow[L\to\infty]{} \frac{\log 2}{\pi} \approx 0.221.$$

落ち着くべき平均が存在しないとき、独立なコーシー分布からのサンプルを平均するとどうなるだろうか。それでも 1000 個のサンプルを平均すれば、1 回の抽出よりも 0 の近くに落ち着くと期待したくなるかもしれない（1 回の抽出が $[-1,1]$ の外に出る確率は $P(|X|>1) = 1-\frac{2}{\pi}\arctan 1 = \frac{1}{2}$ である）。以下のセルでは、10^5 個のコーシー乱数の逐次平均を 10^5 個の標準ガウス乱数の逐次平均と並べてプロットし、さらに 1000 組の独立な平均（それぞれ 1000 個のコーシー乱数の平均）のうち 1 より大きく 0 から離れたままのものがいくつあるかを数える。

```
n = 10**5
idx = np.unique(np.logspace(0, 5, 500).astype(int)) - 1  # log-spaced n
running_mean = lambda x: (np.cumsum(x) / np.arange(1, n + 1))[idx]
d2l.plot(idx + 1, [running_mean(rng.standard_normal(n)),
                   running_mean(rng.standard_cauchy(n))],
         'n', 'running mean', legend=['Gaussian', 'Cauchy'],
         xscale='log', ylim=[-4, 4], figsize=(5, 3))
averages = rng.standard_cauchy((1000, 1000)).mean(axis=1)
print(f'averages of 1000 Cauchy draws with |average| > 1: '
      f'{np.mean(np.abs(averages) > 1):.2f}')
```

```
averages of 1000 Cauchy draws with |average| > 1: 0.49
```



ガウス分布の逐次平均は 0 に向かって滑らかに収束する。一方、コーシー分布の逐次平均は決して落ち着かない。たった 1 つの極端なサンプルが引き金となって突然大きく跳ね上がり、オーダー 1 のジャンプがどの n においても起こり得る。平均をとることによる改善はまったくない。1000 個のサンプルの平均のうち約半分が依然として 1 より大きく 0 から離れており、1 回の抽出における割合と変わらない。実際、証明抜きで述べると、独立な n 個の標準コーシー変数の平均は、単一の標準コーシー変数とまったく同じ分布に従う。

任意の指数関数 $e^{-\lambda t}$（$\lambda > 0$）よりも裾の減衰が遅いとき、その分布は**裾が重い**（heavy-tailed）

という。$P(|X| > t)$ が $t^{-\alpha}$ のように減衰するべき乗則の裾（power-law tail）はその代表例であり、すべてのモーメント $\mathbb{E}[|X|^k]$（$k \geq \alpha$）を無限大にする。コーシー分布は $\alpha = 1$ に対応する。ただし、裾が重いからといって必ずしもモーメントが無限大になるとは限らない。36.2.3.1 節で導出する対数正規分布（log-normal distribution）は、裾が重いにもかかわらずすべてのモーメントが有限である。深層学習において重い裾が問題になるのは、その上での標本平均の収束が遅くなるか、あるいはまったく収束しなくなるからである。たとえばミニバッチの勾配ノイズは裾が重いと論じられており (Simsekli et al. 2019)、この仮説は演習問題 5 で検証できる。

本節で扱ったのは一度に 1 つの数値だけであったが、実際のダーツは平面上に刺さり 2 つの座標を持つ。この 2 つの座標は無相関でありながら独立とはほど遠いことがわかる。それを理解するには平面上の密度が必要となるため、36.2 節で取り上げる。

36.1.4 まとめ

連続確率変数は個々の値すべてに確率 0 を割り当てるため、その分布は単位長さあたりの確率である密度を通じて記述する。密度を区間上で積分すればその区間の確率が得られ、累積分布関数はこの面積を左から右へと蓄積して $F' = p$ を満たし、累積分布関数の逆関数をとれば一様ノイズがその分布からのサンプルへと変換される。マルコフの不等式とチェビシェフの不等式は、平均または分散のみを用いて任意の分布の裾を上から抑える。チェビシェフの上界は最悪ケースでは厳密である一方、典型的な分布に対しては緩い。コーシー分布のような裾の重い分布は平均も分散も持たず、そのサンプルを平均しても広がりはまったく縮まらない。何よりもまず、マルコフの不等式を次節以降へ持ち帰ってほしい。36.9 節におけるすべての裾の上界は、この不等式から導かれる。

36.1.5 演習問題

1. **べき乗則の裾**。X が密度 $p(x) = 1/x^2$（$x \geq 1$ のとき）および $p(x) = 0$（それ以外のとき）を持つとする。
    1. p が有効な密度であることを確かめ、$P(X > 2)$ と累積分布関数 $F(x)$ をすべての実数 x に対して計算せよ。
    2. 分位点関数 $F^{-1}(u)$（$u \in (0,1)$）、中央値、および他の 2 つの四分位点を求めよ。さらに、この分布に対する逆変換サンプラ $X = F^{-1}(U)$（Figure 36.4）を書き下せ。
    3. $\mathbb{E}[X] = \infty$ を示せ。勾配クリッピングが勾配ノルムに対して行うように、X を水準 $c > 1$ でクリッピングすると $\min(X, c)$ が得られる。$\mathbb{E}[\min(X, c)]$ を計算し、それが c とともにどのように増大するか述べよ。
    4. コーシー密度 $p_C(x) = 1/(\pi(1 + x^2))$ に対して、本文で挙げた $\int_{-L}^{L} x\, p_C(x)\, dx$ と $\int_{-L}^{2L} x\, p_C(x)\, dx$ の値、および $L \to \infty$ としたときのそれぞれの極限を導出せよ。この 2 つの極限はコーシー分布の平均について何を示しているか。また、(a) の非負の変数 X の平均ではなぜそのような曖昧さが生じ得ないのか。
2. **整流ガウス分布**。$X \sim \mathcal{N}(0,1)$ が標準ガウス密度 $p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ と累積分布関数 $\Phi(x) = \int_{-\infty}^{x} p_X(t)\, dt$ を持つとし、$Y = \max(0, X)$ とする。これは活性化前の値が標準正規分布に従う ReLU ユニットの出力であり、初期化時のユニットの一般的なモデルである。
    1. 累積分布関数 F_Y を Φ を用いて表し、その概形を描け。Y はどこにアトムを持ち、その確率質量はいくらか。Y は離散型か、密度を持つか、それとも混合型か。
    2. Y が密度を持たない理由を説明せよ。狭義単調かつ連続微分可能な変換 $Y = g(X)$ に対しては、36.2.3.1 節の変数変換の公式により密度 $p_Y(y) = p_X(g^{-1}(y))\, |(g^{-1})'(y)|$ が与えられる。ReLU ではこの公式のどの仮定が破れているか。

3. $[-c, c]$（しきい値 $c > 0$）へクリッピングすると（hard-tanh 活性化関数が $c = 1$ で行う操作に相当する）、$C = \min(\max(X, -c), c)$ が得られる。C の累積分布関数を書き下し、しきい値 c として、C のアトムが合わせて 10% の確率を持つような値を求めよ。この方程式には閉形式解がないため、Φ の数表を用いるか、`math.erf` と $\Phi(z) = \frac{1}{2}(1 + \mathrm{erf}(z/\sqrt{2}))$ を用いよ。
4. $\mathbb{E}[Y^2] = \frac{1}{2}$ および $\mathbb{E}[Y] = 1/\sqrt{2\pi}$ を示せ。ヒント：$X^2 = Y^2 + \min(0, X)^2$ と表し、X の対称性を用いよ。4.4.2.2 節の He 初期化は重みの分散を 2 倍にするが、この係数はこれら 2 つのモーメントのどちらを補償しているか。

3. [code] **ロジスティック分布と較正**。ロジスティック分布は累積分布関数 $F(x) = \mathrm{sigmoid}(x) = 1/(1 + e^{-x})$ を持つ。
   1. Equation 36.6 を用いてその密度を求め、さらに分位点関数を求めよ。四分位点はいくらか。
   2. X が連続かつ狭義単調増加な累積分布関数 F を持つとする。$F(X)$ が $[0, 1]$ 上で一様であることを示せ。これは逆変換サンプリングの逆命題である。
   3. ある確率的回帰モデルが、各入力 i に対して目的変数 y_i の予測分布として、正しい位置 μ_i を中心としスケール s を持つロジスティック分布、すなわち累積分布関数 $\mathrm{sigmoid}\big((y - \mu_i)/s\big)$ を出力するとする。実際の誤差 $y_i - \mu_i$ は標準ロジスティック分布に従うものとする。(a) の分位点関数を用いて 10^4 個の誤差を抽出し、$u_i = \mathrm{sigmoid}\big((y_i - \mu_i)/s\big)$ を $s \in \{0.5, 1, 2\}$ に対して計算して、u_i の 3 つのヒストグラムをプロットせよ。予測分布が狭すぎることを示すのはどの形状であり、広すぎることを示すのはどの形状か。このようなヒストグラムは、確率的予測の較正を確かめる標準的な手法である (Gneiting et al. 2007)。

4. **片側チェビシェフ不等式**。●チェビシェフの不等式 Equation 36.15 は両側の裾を一度に抑える。一方、カンテリの不等式は片側の裾を抑えるものであり、X（平均 μ、分散 σ^2）および任意の $t > 0$ に対して次式が成り立つ。

$$P(X - \mu \geq t) \leq \frac{\sigma^2}{\sigma^2 + t^2}. \tag{36.16}$$

   1. 任意の $u \geq 0$ に対して、マルコフの不等式 Equation 36.14 を $(X - \mu + u)^2$ に適用することで $P(X - \mu \geq t) \leq (\sigma^2 + u^2)/(t + u)^2$ を示せ。さらに u を適切に選んでカンテリの不等式を証明せよ。
   2. $t = \alpha\sigma$ としたときの上界を書き下し、Equation 36.15 と $\alpha = 1$ および $\alpha = 2$ において比較せよ。
   3. 与えられた μ、σ、t に対して、平均 μ と分散 σ^2 を持ち、この上界で等号を達成する 2 点上の分布を求めよ。

5. [extended] **重い裾の平均**。●●Simsekli et al. (2019) は、深層ネットワークにおける確率的勾配降下法（SGD）の勾配ノイズは裾が重く、ガウス分布よりも α 安定分布（$\alpha < 2$ で分散が無限大になる）でよく記述されると論じた。この課題では、その領域でモンテカルロ平均がどのように振る舞うかを測定し、続いて小規模なモデルでその前提を検証する。パレート密度 $p(x) = \alpha x^{-\alpha-1}$（$x \geq 1$、$\alpha > 0$）は、問題 1 の密度を $\alpha = 1$ の場合として含んでいる。
   1. 問題 1 の (b) と (c) をこの分布族へ一般化せよ。すなわち、$\mathbb{E}[X^k] < \infty$ となるのがちょうど $k < \alpha$ のときに限ることを示し、平均 μ（$\alpha > 1$ のとき）を計算し、逆変換サンプラを書き下せ。
   2. $1 < \alpha < 2$ とし、平均は有限だが分散は無限大であるとする。独立な n 個のサンプルの最大値が t を超える確率は、$nt^{-\alpha}$ が小さいとき $nt^{-\alpha}$ に近いことを示し、最大

値が典型的に到達するスケール t_n を求めよ。さらに、標本平均の誤差 $|\bar{X}_n - \mu|$ が $n^{1/\alpha - 1}$ のオーダーになること（$n^{-1/2}$ ではなく）を発見的に論じよ。Simsekli らが援用する一般化中心極限定理は、この議論を厳密にするものである。

3. $\alpha \in \{1.5, 2.5, 3\}$ および $n \in \{10, 10^2, \dots, 10^5\}$ に対して、$|\bar{X}_n - \mu|$ の中央値を 400 回の反復実験から推定し、中央値誤差の対数を $\log n$ に対して直線回帰して傾きを求めよ。$\alpha = 1.5$ における傾きを (b) のレート $n^{1/\alpha - 1}$ と比較せよ。また、$\alpha = 2.5$ と $\alpha = 3$ における傾きを、分散が有限のときに成り立つレート $n^{-1/2}$ と比較せよ。

4. 各反復において区間 $\bar{X}_n \pm 1.96\, \hat{\sigma}_n / \sqrt{n}$ を構成せよ。ここで $\hat{\sigma}_n$ は標本標準偏差であり、ガウス分布の平均に対しては μ を約 95% の確率で被覆する。この区間が実際にどれだけの頻度で μ を被覆するか記録せよ。$\alpha = 1.5$ かつ $n = 10^5$ における被覆率がなぜそのようになるのか、どの実行でも $\hat{\sigma}_n$ が有限であることを踏まえて説明せよ。この結果は、ミニバッチ勾配のような裾の重いサンプルから計算されたエラーバーに対して何を示唆するか。

5. Hill 推定量 (Hill 1975) は、上位 k 個のサンプル（m 個の正のサンプル $X_{(1)} \geq X_{(2)} \geq \cdots \geq X_{(m)}$ のうち）から裾指数を次式で推定する。

$$\hat{\alpha}_k = \Big(\frac{1}{k} \sum_{i=1}^{k} \log \frac{X_{(i)}}{X_{(k+1)}} \Big)^{-1}. \tag{36.17}$$

10^5 個のパレート乱数に対して $\alpha \in \{1.5, 3\}$ および $k \in \{10^2, 10^3, 10^4\}$ でこの推定量を確かめよ。次に、3.3.5 節のソフトマックス回帰を訓練せよ。訓練後のパラメータにおいて、ミニバッチ勾配とフルバッチ勾配の差の各座標を 1000 個のミニバッチにわたって収集せよ（バッチサイズは 32 と 256 の両方で行う）。それらの絶対値に対する $\hat{\alpha}_k$ を両方のバッチサイズについて k に対してプロットせよ。推定値は 2 を下回るか。また、結論は k の選び方にどれほど依存するか。

36.2 同時分布と変換

36.1 節では、半径 1 のボードに刺さったダーツを、中心からの距離 R（密度は $2r$）という 1 つの数値に還元した。しかし実際には、ダーツは水平座標 X と垂直座標 Y の両方を持つ点に着地する。機械学習でモデル化する量のほとんどは、このように複数の変数の組として現れる。画像は互いに値が依存し合う何千ものピクセルからなり、教師あり学習の問題はどれも入力とターゲットの対を扱う。個々の変数の密度を別々に知るだけでは不十分である。なぜなら、別々の密度からは変数同士がどのように連動するかが何もわからないからである。

本節では 2 つの問いを軸に議論を進める。1 つ目は、ある変数から別の変数について何がわかるかという問いである。ダーツがボードの上端に近い $Y = 0.9$ に着地したとすると、X は中心線から 0.44 以内になければならないため、Y を観測すれば X について確かに何らかの情報が得られる。ところが驚くべきことに、X に対する最良の単一の予測値は依然として 0 であり（Y が何であっても変わらない）、2 つの座標の共分散も 0 であることがわかる。従属性、予測可能性、相関は 3 つの異なる概念であり、これらを区別するには 2 変数の同時密度が必要になる。2 つ目は、ニューラルネットワークの層が入力に作用するときや、サンプラが一様ノイズを変形するときのように、確率変数を変換すると密度がどう変化するかという問いである。ダーツの距離が $R = \sqrt{U}$ として一様変数 U から生成できることはすでに見たとおりであり、平方根をとることで平坦な密度は線形な密度 $2r$ へと変わる。明らかに、変換が密度に及ぼす作用は、単に軸のラベルを付け替えるだけにとどまらない。

本節の 3 つの結果は、本書の後半でとりわけ重要な役割を果たす。全分散の法則 Equation 36.22、最良の二乗誤差予測器としての条件付き平均 Equation 36.23、そして変数変換の公式 Equation 36.31 である。共分散行列と、相関がコサインであるという幾何学的な補足については、初読時には読み飛ばしても差し支えない。

```
rng = np.random.default_rng()
```

36.2.1 同時密度と条件付き密度

連続確率変数の対について知り得るすべての情報は、2 変数の単一の関数に集約される。まずはその関数を定義し、そこから 3 つの本質的な量を取り出す方法を見ていこう。すなわち、一方の座標単独の分布、他方が観測されたもとでの一方の座標の分布、そして他方が与えられたもとでの一方の平均（回帰モデルが推定する対象）である。

36.2.1.1 同時密度と周辺密度

1 次元の密度が短い区間における単位長さあたりの確率を測るのと同様に、その 2 次元版は小さな正方形における単位面積あたりの確率を測る。**同時密度**（joint density）$p_{X,Y}$（確率変数の対 (X, Y) に対する密度）を用いると、一辺 ϵ の (x, y) にある小さな正方形に着地する確率は、おおよそその面積と密度の積として次のように表される。

$$P\big(X \in [x, x+\epsilon],\, Y \in [y, y+\epsilon]\big) \approx \epsilon^2\, p_{X,Y}(x, y).$$

1 次元の場合と同様に、平面の領域 $\mathcal{D}$ をこのような正方形で敷き詰めることで、正式な定義として採用する性質が導かれる。すなわち $p_{X,Y} \geq 0$ かつ $P((X, Y) \in \mathcal{D}) = \iint_{\mathcal{D}} p_{X,Y}(x, y)\, dx\, dy$ であり、したがって平面全体にわたる同時密度の下の全体積は 1 に等しい。n 変数の場合も、n 次元の直方体と n 重積分を用いれば同じ定義が成り立つ。

ダーツの例では、単位円板のどの部分も等しく確からしく、ボードの面積は π であるため、同時密度は円板の内部で $p_{X,Y}(x, y) = 1/\pi$、外部で 0 となる。この密度は円板上では平坦であるにもかかわらず、x の関数と y の関数の積としては書けないことに注意しよう。ボードの円形の縁が 2 つの座標を結びつけているのである。

同時密度がわかっていても、一方の座標単独の分布にしか関心がないことは多い。これを**周辺分布**（marginal distribution）と呼ぶ。細い縦の帯 $X \in [x, x+\epsilon]$ の確率はおおよそ $\epsilon\, p_X(x)$ である。この帯を y に沿って小さな正方形に切り分け、それらの確率を足し合わせると次式を得る。

$$\epsilon\, p_X(x) \approx \sum_i P\big(X \in [x, x+\epsilon],\, Y \in [\epsilon i, \epsilon(i+1)]\big) \approx \sum_i \epsilon^2\, p_{X,Y}(x, \epsilon i).$$

因子 ϵ を 1 つ約分し、残った和をリーマン和（Riemann sum）として y について極限をとれば、次の式が得られる。

$$p_X(x) = \int_{-\infty}^{\infty} p_{X,Y}(x, y)\, dy. \tag{36.18}$$

言葉で言えば、x における周辺密度とは、x における縦の帯に沿って同時密度を積分したものである（Figure 36.7）。すなわち、不要な変数を単に**積分消去**（integrate out）すればよい。ダーツ

の場合、x における縦の帯はボードを $-\sqrt{1-x^2}$ から $\sqrt{1-x^2}$ まで横切るため、周辺密度は次のようになる。

$$p_X(x) = \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \frac{1}{\pi}\, dy = \frac{2}{\pi}\sqrt{1-x^2}, \qquad -1 \le x \le 1,$$

これは半円を描く。簡単に確かめておくと、$\int_{-1}^{1} \sqrt{1-x^2}\, dx$ は単位円板の半分の面積 $\pi/2$ であるから、p_X を積分すると確かに 1 になる。同時密度が平坦であっても、周辺密度は曲線になる。ボードの中央を通る縦の帯のほうが、縁の近くを通る帯よりも長いからである。

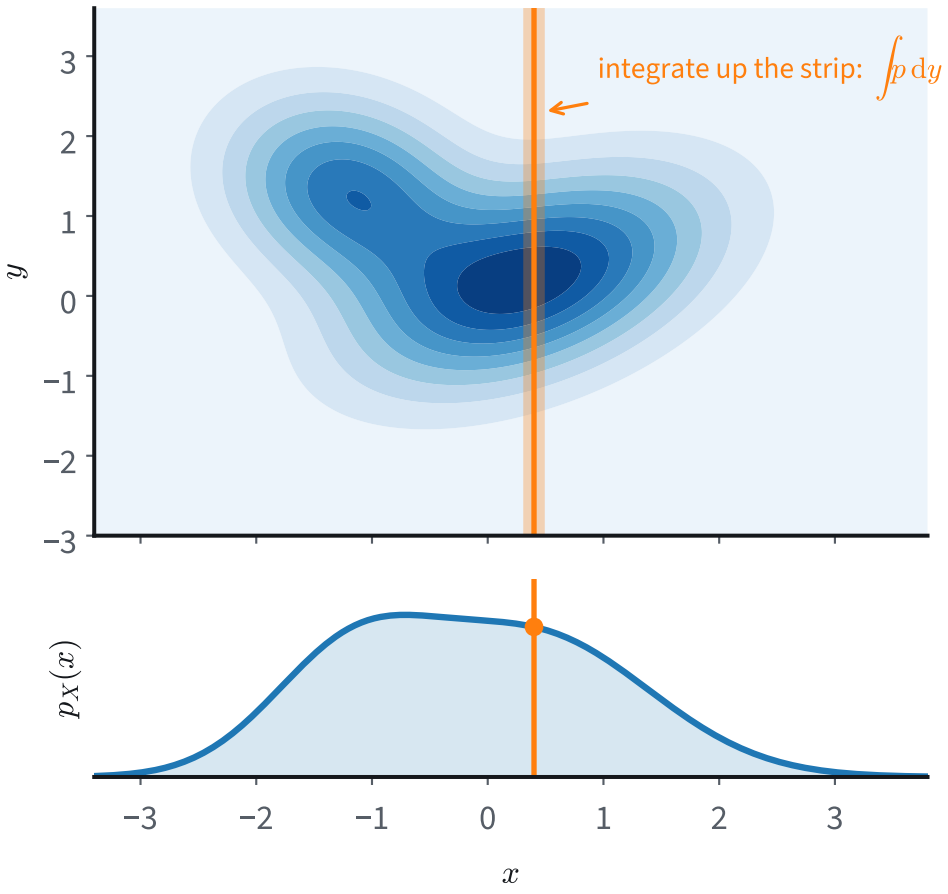


Figure 36.7: 等高線で示した連続同時密度 $p_{X,Y}(x,y)$ と、位置 x で強調表示した 1 本の縦の帯。この帯に沿ってすべての y にわたり同時密度を積分すると、下の周辺密度 $p_X(x)$（x 軸に沿ってプロット）の高さが得られる。この帯を x 全体に走査すると周辺密度全体が描かれる。

36.2.1.2 条件付き密度と独立性

$Y = y$ であることが観測されたとき、X の密度はどう変わるべきだろうか。離散変数であれば、答えは単に $P(X = x \mid Y = y) = P(X = x, Y = y)/P(Y = y)$ である。しかし連続密度の

場合、この式は $P(Y = y) = 0$ のため 0 を 0 で割ることになってしまう。解決策は、そもそも密度を定義したときと同じ極限の議論である。正の確率を持つ細い水平な帯 $Y \in [y, y + \epsilon]$ で条件付け、その帯の幅を縮めていくのである。分子に正方形近似を、分母に 1 変数の規則を適用すると、次のようになる。

$$P(X \in [x, x + \epsilon] \mid Y \in [y, y + \epsilon]) \approx \frac{\epsilon^2 \, p_{X,Y}(x, y)}{\epsilon \, p_Y(y)} = \epsilon \, \frac{p_{X,Y}(x, y)}{p_Y(y)}.$$

この帯の内部では、短い x 区間の確率はその長さ ϵ と $p_{X,Y}(x, y)/p_Y(y)$ の積になる。この比は、X について $Y = y$ が与えられたときの単位長さあたりの確率にあたるため、**条件付き密度**（conditional density）を次のように定義する。

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x, y)}{p_Y(y)}, \qquad p_Y(y) > 0. \tag{36.19}$$

固定した任意の y（ただし $p_Y(y) > 0$）に対して、これは x に関する正当な密度である。非負であり、周辺化の規則 Equation 36.18 によって積分値が $p_Y(y)/p_Y(y) = 1$ となるからである。幾何学的には、同時密度を水平線 $Y = y$ に沿ってスライスし、その断面の面積が 1 になるようスケールし直している。ダーツの場合、水平線 $Y = y$ はボードを弦 $[-\sqrt{1 - y^2}, \sqrt{1 - y^2}]$ 上で横切るため、条件付き密度は次のようになる。

$$p_{X|Y}(x \mid y) = \frac{1/\pi}{(2/\pi)\sqrt{1 - y^2}} = \frac{1}{2\sqrt{1 - y^2}} \qquad \text{for } |x| \le \sqrt{1 - y^2}.$$

$Y = y$ が与えられたもとで、水平位置 X は高さ y の弦上で一様に分布する。この弦は $|y|$ が大きくなるにつれて短くなるため、Y を観測すると X の分布は変化する。

Equation 36.19 を変形すると、**連鎖律**（chain rule）$p_{X,Y}(x, y) = p_{X|Y}(x \mid y) \, p_Y(y)$ が得られる。同時密度を $p_{X|Y} \, p_Y = p_{Y|X} \, p_X$ と 2 通りに書き、$p_{X|Y}$ について解くと、次の**密度に対するベイズの定理**（Bayes' rule）が得られる。

$$p_{X|Y}(x \mid y) = \frac{p_{Y|X}(y \mid x) \, p_X(x)}{p_Y(y)}. \tag{36.20}$$

1.6 節で離散変数に対してベイズの定理が果たしたのと同様に、この恒等式は、Y が X からどのように生成されるかを表すモデルを、X の（Y を観測した後の）分布へと変換する。

Y を観測しても X について新しい情報が何も得られず、$p_{X|Y}(x \mid y) = p_X(x)$ が $p_Y(y) > 0$ である限り常に成り立つ特別な場合、2 つの変数は**独立**（independent）であるといい、$X \perp Y$ と書く。連鎖律により、独立性は同時密度が $p_{X,Y}(x, y) = p_X(x) \, p_Y(y)$ のように因数分解できることと同値である（確率 0 の集合を除く）。一方から他方を導くには、$p_Y(y)$ を掛けるか、それで割ればよい。

直感的には、独立性とは、X の条件付き密度が位置も形状も同じ x の関数となり、どの高さ y でも変わらないことを意味する。単位正方形上、すなわち $p_{X,Y}(x, y) = 4xy$（$[0, 1]^2$ 上）という同時密度を考えてみよう。その周辺密度は $p_Y(y) = \int_0^1 4xy \, dx = 2y$ であり、これより $p_{X|Y}(x \mid y) = 4xy/(2y) = 2x$ となって y によらないため、$X \perp Y$ である。ダーツはその逆の例であり、条件付き密度はどの y でも一様であるが、その台となる弦の長さは y に依存する。

Figure 36.8 が示すように、形状が同じだけでは独立性には十分ではない。相関のある対では、3 つの高さにおける条件付き密度は同じ形状を共有するものの中心がずれているため従属であり、同時密度が因数分解できるときに限って 3 つのスライスがすべて一致する。

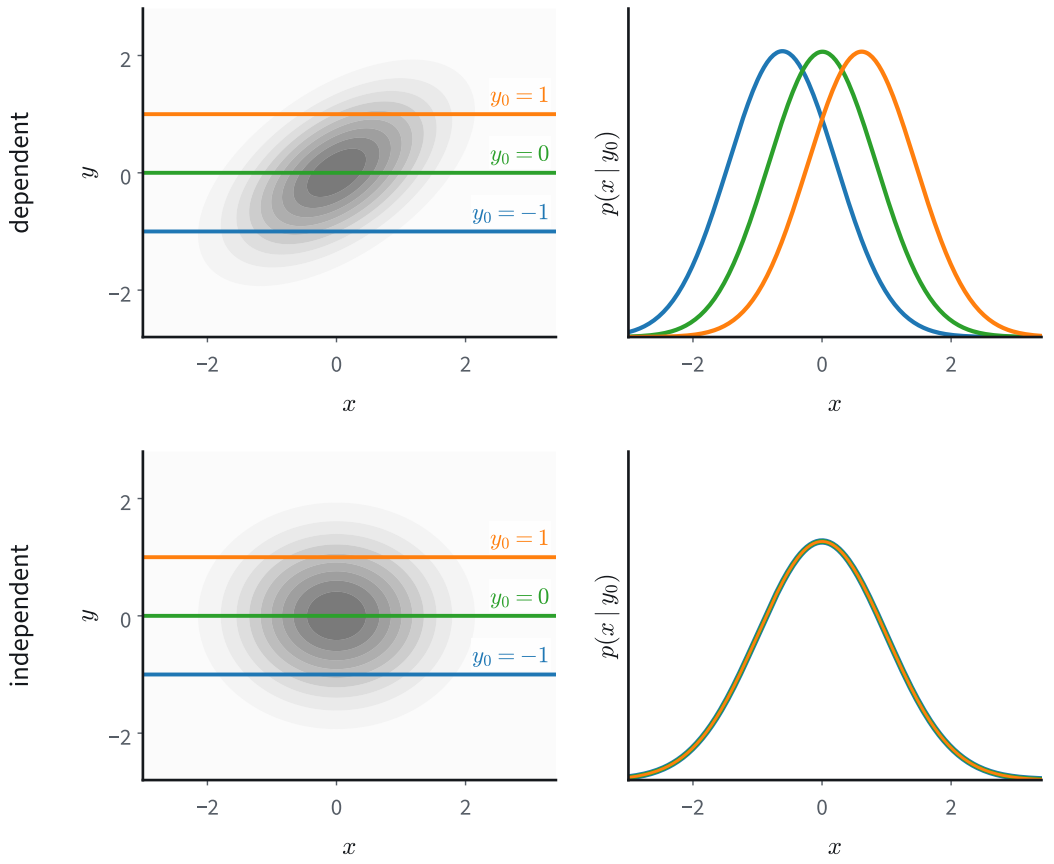


Figure 36.8: スライスとしての条件付け。左：同時密度と $y_0 = -1, 0, 1$ における 3 本の水平スライス。右：各スライスを面積 1 に再正規化したものが条件付き密度 $p(x \mid y_0)$ である。上段：相関のある対。3 つの条件付き密度は同じ形状だが中心がずれているため、条件付き密度は y_0 に依存し、対は従属である。下段：同時密度が因数分解できる対。3 つの条件付き密度が一致し、これが独立性である。

独立性からは、和の密度に対する簡潔な公式も得られる。$X \perp Y$ のとき、対 $(X, X+Y)$ は同時密度 $p_X(x)\,p_Y(s-x)$ を点 (x, s) で持ち、x を Equation 36.18 によって積分消去すると**畳み込み**（convolution）$p_{X+Y}(s) = \int p_X(x)\,p_Y(s-x)\,dx$ が得られる。

36.2.1.3　条件付き期待値

回帰モデルは入力からターゲットを予測する。前項との整合性を保つため、ここでは X をターゲット、Y を観測される入力とする（本編の他章では逆の記号割り当てを用いる）。二乗損失による訓練では、関数 h のうち $\mathbb{E}[(X - h(Y))^2]$ をできるだけ小さくするものを探索する。どの関数が最良だろうか。任意の細い帯 $Y \approx y$ の中で、モデルは広がりを持つターゲットの候補全体に対して 1 つの数値を予測しなければならない。任意の確率変数 X に対し、平均二乗距離の意味で最も近い定数 c はその平均 μ_X である。実際、μ_X の周りで二乗を展開すると

$\mathbb{E}[(X-c)^2] = \mathrm{Var}(X) + (\mu_X - c)^2$ となる。このことは、各帯の内部における X の平均を予測すべきであることを示唆している。そこで、X について $Y=y$ が与えられたときの**条件付き期待値**（conditional expectation）を、条件付き密度の平均として次のように定義する。

$$\mathbb{E}[X \mid Y{=}y] = \int x\, p_{X|Y}(x \mid y)\, dx. \tag{36.21}$$

これは単に、X の値を $Y=y$ となるすべての結果にわたって平均したものである。y を動かすと $\mathbb{E}[X \mid Y{=}y]$ は y の関数を描き、そこに確率変数 Y を代入すると、$\mathbb{E}[X \mid Y]$ という Y のみに依存する確率変数になる。同じ帯内での平均により、$\mathbb{E}[g(X,Y) \mid Y{=}y] = \int g(x,y)\, p_{X|Y}(x \mid y)\, dx$ が任意の関数 g に対して定義される。因子 $h(Y)$ は各帯の内部では定数であるため、条件付き期待値の外へそのままくくり出せる。

$$\mathbb{E}[h(Y)\, g(X,Y) \mid Y] = h(Y)\, \mathbb{E}[g(X,Y) \mid Y].$$

同様に、1 つの帯の内部における X の広がりは**条件付き分散**（conditional variance）$\mathrm{Var}(X \mid Y) = \mathbb{E}[X^2 \mid Y] - (\mathbb{E}[X \mid Y])^2$ であり、これは 36.1 節の分散の計算公式を各帯の内部に適用したものにほかならない。

直感的には、帯ごとの平均をさらに平均すれば全体の平均に戻り、全体の分散は帯内部のばらつきと帯間のばらつきの和に分解できるはずである。実際そのどちらも成り立ち、とりわけ 2 つ目の公式は非常に有用である。

命題（全期待値と全分散の法則） *X, Y が同時密度を持ち、X が有限の分散を持つとする（離散変数の組についても、積分を和に置き換えれば同じ恒等式が成り立つ）。このとき、次式が成り立つ。*

$$\begin{aligned} \mathbb{E}[X] &= \mathbb{E}\big[\mathbb{E}[X \mid Y]\big], \\ \mathrm{Var}(X) &= \mathbb{E}\big[\mathrm{Var}(X \mid Y)\big] + \mathrm{Var}\big(\mathbb{E}[X \mid Y]\big). \end{aligned} \tag{36.22}$$

証明 **塔性質**（tower property）として知られる 1 つ目の恒等式については、内側の期待値を書き下し、連鎖律 $p_{X|Y}(x \mid y)\, p_Y(y) = p_{X,Y}(x,y)$ を適用して帯を再び同時密度へとまとめればよい。

$$\mathbb{E}\big[\mathbb{E}[X \mid Y]\big] = \int \Big(\int x\, p_{X|Y}(x \mid y)\, dx \Big) p_Y(y)\, dy = \iint x\, p_{X,Y}(x,y)\, dx\, dy = \mathbb{E}[X].$$

条件付き密度が定義されない y（すなわち $p_Y(y) = 0$ となる値）には Y が確率 0 でしか着地しないため、両辺のどちらにも寄与しない。$g(x,y)$ を x の代わりに用いて同じ計算を行えば、$\mathbb{E}\big[\mathbb{E}[g(X,Y) \mid Y]\big] = \mathbb{E}[g(X,Y)]$ が得られる。2 つ目の恒等式については、右辺の両項を展開する。塔性質を X^2 に適用すると次式を得る。

$$\begin{aligned} \mathbb{E}\big[\mathrm{Var}(X \mid Y)\big] &= \mathbb{E}\big[\mathbb{E}[X^2 \mid Y]\big] - \mathbb{E}\big[(\mathbb{E}[X \mid Y])^2\big] \\ &= \mathbb{E}[X^2] - \mathbb{E}\big[(\mathbb{E}[X \mid Y])^2\big]. \end{aligned}$$

さらに $\mathrm{Var}(Z) = \mathbb{E}[Z^2] - (\mathbb{E}[Z])^2$（36.1 節）を確率変数 $\mathbb{E}[X \mid Y]$（塔性質によりその平均は $\mathbb{E}[X]$ である）に適用すると、次式を得る。

$$\operatorname{Var}\big(\mathbb{E}[X \mid Y]\big) = \mathbb{E}\big[(\mathbb{E}[X \mid Y])^2\big] - \mathbb{E}[X]^2.$$

これら 2 式を足し合わせると、中央の項 $\mathbb{E}[(\mathbb{E}[X \mid Y])^2]$ が打ち消し合い、$\mathbb{E}[X^2] - \mathbb{E}[X]^2 = \operatorname{Var}(X)$ が残る。■

2 つ目の恒等式は、右辺に現れる E、V、V、E の並びにちなんで、しばしば**イブの法則**（Eve's law）と呼ばれる。この式は X の全分散を 2 つの部分に分解する。1 つは各帯の内部における分散の平均であり、これは Y を知っても決して取り除けない。もう 1 つは帯ごとの平均それ自体の分散であり、これが Y によって説明される部分である。統計学では、データの変動をグループに帰属させるために、この分解を分散分析（analysis of variance）という名で用いる。次に示すとおり、同じ分解によって先ほどの回帰の問いにも決着がつく。

命題（条件付き平均は最良の二乗誤差予測器である） **X が有限の分散を持ち、h を $\mathbb{E}[h(Y)^2] < \infty$ を満たす任意の関数とする。このとき、次式が成り立つ。**

$$\mathbb{E}\big[(X - h(Y))^2\big] = \mathbb{E}\big[\operatorname{Var}(X \mid Y)\big] + \mathbb{E}\big[(\mathbb{E}[X \mid Y] - h(Y))^2\big]. \tag{36.23}$$

証明 $m(Y) = \mathbb{E}[X \mid Y]$ とおき、予測誤差を $X - h(Y) = (X - m(Y)) + (m(Y) - h(Y))$ の 2 つの部分に分解する。二乗すると 2 つの二乗項と 1 つの交差項が現れるが、注意を要するのは交差項だけである。1 つ目の二乗項の期待値は、塔性質により $\mathbb{E}[\operatorname{Var}(X \mid Y)]$ となる。交差項 $2(X - m(Y))(m(Y) - h(Y))$ の条件付き平均は 0 である。なぜなら、因子 $m(Y) - h(Y)$ は Y のみに依存するため帯内の平均からくくり出せ、残りが $\mathbb{E}[X - m(Y) \mid Y] = m(Y) - m(Y) = 0$ となるからである。したがって塔性質により、その全体の平均も 0 になる。■

右辺の 2 つの項に注目しよう。第 1 項は h にまったく依存せず、一方で第 2 項は非負であり $h(Y) = \mathbb{E}[X \mid Y]$ のときに 0 になる。したがって、条件付き平均が二乗誤差のもとでの最適な予測器であり、いかなる予測器も帯内分散の平均より小さい期待二乗誤差を達成することはできない。ゆえに、二乗損失で訓練されたネットワークは $\mathbb{E}[\text{target} \mid \text{input}]$ を推定しているのであり、これは 2.1 節におけるモデルの解釈（そこでは $\mathbb{E}[y \mid \mathbf{x}]$ と表記した）そのものである。

2 つの例でこの分解を具体的に見てみよう。まずダーツの例に戻り、X を Y から予測してみる。$Y = y$ が与えられたとき、水平座標 X は 0 に関して対称な弦上で一様に分布するため、$\mathbb{E}[X \mid Y = y] = 0$ となる。半幅 w の区間上の一様変数は分散 $w^2/3$ を持つため、$\operatorname{Var}(X \mid Y = y) = (1 - y^2)/3$ となる。対称性により $\mathbb{E}[X^2] = \mathbb{E}[Y^2] = \tfrac{1}{2}\mathbb{E}[R^2] = \tfrac{1}{4}$ であるから、イブの法則より次式が得られる。

$$\operatorname{Var}(X) = \mathbb{E}\Big[\frac{1 - Y^2}{3}\Big] + \operatorname{Var}(0) = \frac{1 - 1/4}{3} + 0 = \frac{1}{4}.$$

分散はすべて説明不能な項に残る。Y を知ることで X の範囲は狭まるものの中心はまったく動かないため、X を Y から予測する最良の二乗誤差予測器は定数 0 であり、これは X と Y が従属であっても変わらない。

2 つ目の例として、2 値ラベル Z を、値 1（確率 0.2）または値 0（確率 0.8）をとるように抽出する。$Z = 1$ のときは X を正規分布 $\mathcal{N}(3, 1)$（平均 3、分散 1）から抽出し、$Z = 0$ のときは X を $\mathcal{N}(-1, 1)$ から抽出する。この混合分布は、Figure 36.1 の 2 つの峰を持つ密度を生み出す。全体の平均は $0.2 \cdot 3 + 0.8 \cdot (-1) = -0.2$ である。各成分の内部での分散は 1 であるため、$\mathbb{E}[\operatorname{Var}(X \mid Z)] = 1$ となる。成分間では、$\mathbb{E}[X \mid Z]$ は値 3 と -1 をとり、その分散は $0.2 \cdot 0.8 \cdot (3 - (-1))^2 = 2.56$ である。したがって全分散は 3.56 であり、ラベル Z はそのうちの $2.56/3.56 = 72\%$ を説明する。100 万個のサンプルを抽出すれば、経験的な成分内分散と成

分間分散の項はそれぞれ 1 と 2.56 の近くに落ち着くだろう。コードを実行する前に考えてみてほしい。両者の和は標本分散とおおよそ一致するだけだろうか、それとも厳密に一致するだろうか。

```
n = 10**6
z = rng.uniform(size=n) < 0.2                          # component label Z
x = np.where(z, rng.normal(3, 1, n), rng.normal(-1, 1, n))
w = np.array([np.mean(~z), np.mean(z)])                # P(Z=0), P(Z=1)
cond_mean = np.array([x[~z].mean(), x[z].mean()])      # E[X | Z]
within = w @ np.array([x[~z].var(), x[z].var()])       # E[Var(X | Z)]
between = w @ (cond_mean - x.mean())**2                # Var(E[X | Z])
print(f'Var(X) {x.var():.3f} = within {within:.3f} + between {between:.3f}')
```

```
Var(X) 3.562 = within 1.000 + between 2.562
```

厳密に一致する。それぞれの項は理論値（1 と 2.56）の近くに落ち着き、小数第 3 位にサンプリングノイズが現れるが、両者の和はどの実行でも出力の最終桁の丸め誤差を除いて標本分散に一致する。イブの法則は、経験的なグループの割合とグループ内分散に確率と条件付き分散の役割を担わせることで、標本そのものに対しても厳密な恒等式として成り立つのである。

36.2.2 共分散と相関

条件付き平均は、X の平均が Y に伴ってどう動くかを記述するため、それ自体が 1 つの関数になる。これに対し、単一の要約値で済ませたいことも多い。すなわち、X と Y はともに大きくなる傾向があるのか、それとも一方が大きいときにもう一方は小さくなる傾向があるのか、という指標である。その数値こそが共分散である。ただし共分散には、本節の残りで対処または明らかにする 2 つの限界がある。両変数の単位を引き継いでしまうこと（これは相関によって解消される）と、直線的な傾向しか捉えられないこと（これはどのようなスケール変換でも解消できない）である。

36.2.2.1 共分散

1.6 節では、**共分散**（covariance）を、それぞれの平均からの偏差の積の期待値として定義した。離散変数 X, Y が値 (x_i, y_j) を確率 p_{ij} でとるとき、共分散は次の式で与えられる。

$$\sigma_{XY} = \operatorname{Cov}(X,Y) = \sum_{i,j} (x_i - \mu_X)(y_j - \mu_Y)\, p_{ij} = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y], \tag{36.24}$$

ここで 2 番目の形は、分散の計算公式と同様に積を展開することで得られる。特に、$\operatorname{Cov}(X,X) = \operatorname{Var}(X)$ である。本節を通じて両変数の分散は有限であると仮定する。このとき $|xy| \le (x^2 + y^2)/2$ であるから、XY の期待値の存在が保証される。

具体例として、$X \in \{1, 3\}$、$Y \in \{-1, 3\}$ とし、同時確率が次で与えられるとする。

$$\begin{aligned} P(X{=}1, Y{=}{-1}) &= \tfrac{\theta}{2}, & P(X{=}1, Y{=}3) &= \tfrac{1-\theta}{2}, \\ P(X{=}3, Y{=}{-1}) &= \tfrac{1-\theta}{2}, & P(X{=}3, Y{=}3) &= \tfrac{\theta}{2}, \end{aligned}$$

ここで $\theta \in [0,1]$ はパラメータである。これを **4 セルの例**と呼ぶことにする。各変数はそれぞれの 2 つの値を確率 $\tfrac{1}{2}$ でとるため、$\mu_X = 2$、$\mu_Y = 1$ である。積の期待値は次のように計算できる。

$$\mathbb{E}[XY] = \tfrac{1}{2}(-\theta + 3(1-\theta) - 3(1-\theta) + 9\theta) = 4\theta,$$

これより $\mathrm{Cov}(X,Y) = 4\theta - 2$ が得られる。すなわち、共分散は $+2$（$\theta = 1$ のとき）、-2（$\theta = 0$ のとき）、そして 0（$\theta = \tfrac{1}{2}$ のとき）となる。

同時密度を持つ連続変数の場合、和は二重積分に置き換わる。

$$\sigma_{XY} = \iint (x - \mu_X)(y - \mu_Y)\, p_{X,Y}(x,y)\, dx\, dy, \tag{36.25}$$

すなわち、同時密度にわたって偏差の積を平均する。ダーツの例では、両座標の平均は 0 であり、$\mathrm{Cov}(X,Y) = \mathbb{E}[XY] = \frac{1}{\pi}\iint_{\text{disk}} xy\, dx\, dy = 0$ となる。被積分関数が x に関して奇関数である一方、円板が対称だからである。それにもかかわらず、弦が縮むことから見たように X と Y は従属である。共分散は線形な傾向しか見ないが、ダーツの円形の従属関係には線形な傾向がないため、共分散が 0 であっても独立性は導かれない（1.6 節では離散の例として $Y = X^2$ を挙げた）。Figure 36.9 は円板上でこれら両方の事実を示している。

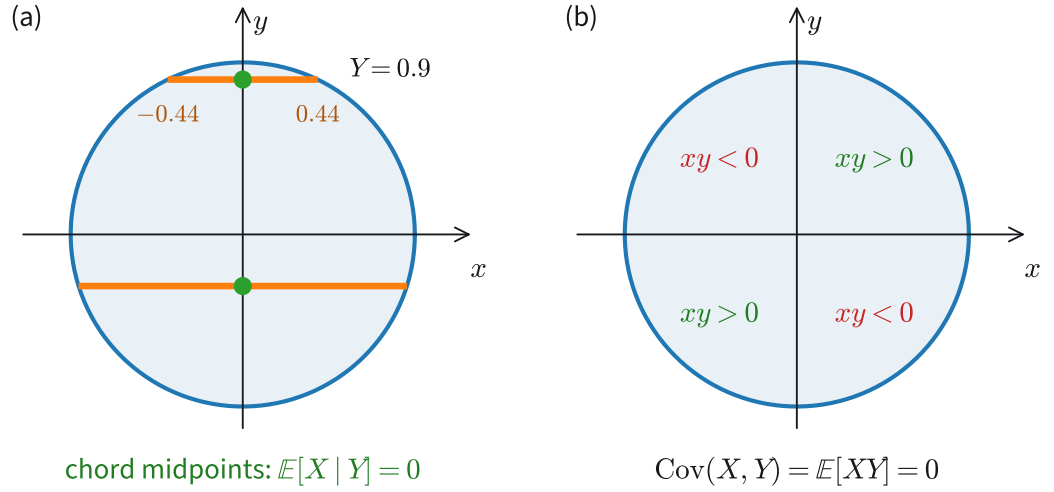


Figure 36.9: ダーツの座標は従属であるが、平均においては予測不能であり無相関である。(a) 高さ $Y = 0.9$ の弦は X を $|X| \le 0.44$ に閉じ込めるのに対し、中央付近の弦ははるかに長いため、X は Y に従属している。どの弦も中心は $x = 0$（緑の点）にあるため、$\mathbb{E}[X \mid Y] = 0$ となる。(b) 積 xy は 2 つの象限で正、残り 2 つの象限で負となり、いずれも面積が等しいため $\mathrm{Cov}(X,Y) = 0$ となる。

共分散は両変数の単位も引き継ぐ。Figure 36.10 の下段は上段とまったく同じサンプルをプロットしたものだが、Y の単位をドルではなくセントで表している。どの点群の形状もまったく変わらないにもかかわらず、標本共分散はすべて 100 倍に跳ね上がる。単位の情報がなければ共分散で意味を持つのは符号だけであり、以下で導入する相関係数は単位を完全に取り除く。

共分散は各引数に関して線形である。定数を足しても変数とその平均が同じだけ平行移動するため偏差からは消え、定数倍の因子はそのまま外に出るため $\mathrm{Cov}(cX + d, Y) = c\,\mathrm{Cov}(X,Y)$ となる。同様に、和の偏差は偏差の和であるから、期待値の線形性により $\mathrm{Cov}(X + Z, Y) = \mathrm{Cov}(X,Y) + \mathrm{Cov}(Z,Y)$ が成り立ち、第 2 引数についても同じ規則が成り立つ。さらに、独

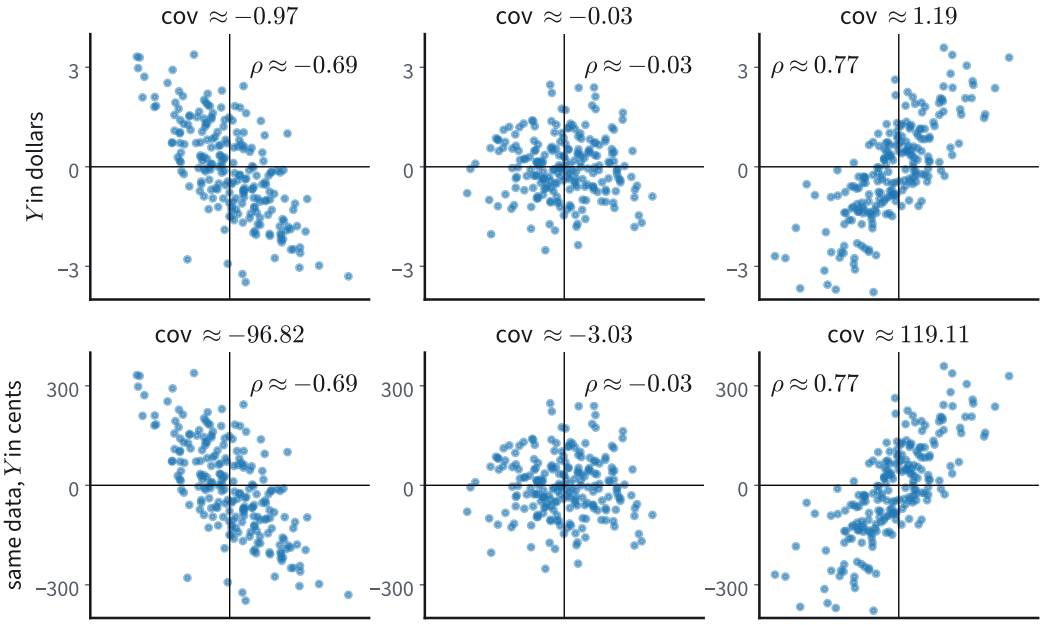


Figure 36.10: 上段：共分散を負、0、正（左から右）に調整した (X, Y) の散布図（Y の単位はドル）。負の共分散では点群が右下がりに傾き、0 では丸くなり、正では右上がりに傾く。下段：まったく同じサンプルの Y をセント単位で表したもの。標本共分散がすべて 100 倍になる一方、各パネルの相関 ρ（上部隅）は変わらない。共分散の大きさは単位を持ち、スケールに依存しないのはその符号だけである。

立変数の共分散は常に 0 である。同時密度が因数分解できるときは常に、$\mathbb{E}[XY]$ の二重積分が $\mathbb{E}[X]\,\mathbb{E}[Y]$ へと分解するからである。

共分散を用いると、和の分散の公式も完成する。$\tilde{X} = X - \mu_X$、$\tilde{Y} = Y - \mu_Y$ とおく。線形性により $X + Y$ の平均は $\mu_X + \mu_Y$ となり、したがって偏差は $\tilde{X} + \tilde{Y}$ となる。$\mathbb{E}[(\tilde{X} + \tilde{Y})^2] = \mathbb{E}[\tilde{X}^2] + 2\,\mathbb{E}[\tilde{X}\tilde{Y}] + \mathbb{E}[\tilde{Y}^2]$ を展開すると、次の式が得られる。

$$\operatorname{Var}(X + Y) = \operatorname{Var}(X) + \operatorname{Var}(Y) + 2\operatorname{Cov}(X, Y). \tag{36.26}$$

X と Y が独立であるとき、共分散は消え、分散は単純に足し合わされる。

この公式から導かれる最も重要な帰結は、多数の抽出値を平均したときに何が起きるかである。n 個の偏差の和の二乗を同様に展開すると、$\operatorname{Var}(\sum_i X_i) = \sum_i \sum_j \operatorname{Cov}(X_i, X_j)$ が得られる。いま $X_1, \dots, X_n$ が独立で、共通の分散 σ^2 を持つとする。すべての交差項が消えるため和の分散は $n\sigma^2$ となり、さらに $c = 1/n$ 倍することで分散は n^2 で割られる。したがって、平均 $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ の分散は次のようになる。

$$\operatorname{Var}(\bar{X}_n) = \frac{1}{n^2} \cdot n\sigma^2 = \frac{\sigma^2}{n}. \tag{36.27}$$

独立な n 個の抽出値を平均すると、分散は n 分の 1 に、標準偏差は $\sqrt{n}$ 分の 1 になる。つまり、典型的な誤差を半分にするには 4 倍の抽出数が必要になる。この $\sigma/\sqrt{n}$ のスケーリングは、シミュレーションで生成したサンプルの平均で期待値を近似するあらゆる**モンテカルロ**推定の基礎となっている（36.5.1.7 節）。ダーツの例では、独立に n 回投げたときの平均距離の分散は $1/(18n)$ であるため、556 本投げれば標準偏差を 0.01 まで下げられる。なお、$\sigma/\sqrt{n}$ の収束率には有限の分散が必要である。一方、$\bar{X}_n$ が平均に収束することを主張する**大数の法則**（law of large numbers）そのものに必要なのは、有限の平均だけである（36.5 節）。コーシー分布はそのどちらも持たず、n をどれほど大きくしても抽出値の平均が落ち着くことはない。

36.2.2.2 相関

4 セルの例における共分散 $4\theta - 2$ が大きいかどうかは、X と Y の単位に依存する。単位を取り除くには、まったく同じ単位を持つ 2 つの標準偏差で割ればよい。**相関係数**（correlation coefficient）は次式で定義される。

$$\rho(X, Y) = \frac{\operatorname{Cov}(X, Y)}{\sigma_X \sigma_Y}. \tag{36.28}$$

1.6 節では、ρ が常に -1 と 1 の間に収まることを述べた。この評価を証明し、等号がいつ成り立つかを正確に確かめよう。

命題（相関係数の範囲）有限かつ非ゼロの分散を持つ任意の X, Y に対して、$-1 \le \rho(X, Y) \le 1$ が成り立つ。等号は、$Y = cX + d$ が定数 $c \neq 0$ と d に対して確率 1 で成り立つとき、かつそのときに限り成立し、$\rho = 1$（$c > 0$ のとき）、$\rho = -1$（$c < 0$ のとき）となる。

証明分散は決して負にならないため、$\operatorname{Var}(tX + Y) \ge 0$ がすべての実数 t に対して成り立つ。和の分散の公式とアフィンスケーリングの公式を適用すると、これは t に関する二次関数になる。

$$\operatorname{Var}(tX + Y) = \operatorname{Var}(X)\,t^2 + 2\operatorname{Cov}(X, Y)\,t + \operatorname{Var}(Y) \;\ge\; 0 \quad \text{for every } t.$$

最高次の係数が正で決して 0 を下回らない二次関数は実数解を高々 1 つしか持てないため、その判別式は非正でなければならない。

$$\big(2\,\mathrm{Cov}(X,Y)\big)^2 - 4\,\mathrm{Var}(X)\,\mathrm{Var}(Y) \le 0, \qquad \text{that is} \qquad \mathrm{Cov}(X,Y)^2 \le \mathrm{Var}(X)\,\mathrm{Var}(Y).$$

両辺を $\sigma_X^2\sigma_Y^2$ で割ると $\rho^2 \le 1$ が得られる。$\rho^2 = 1$ のときは判別式が 0 となり、この二次関数は重解 $t^\star$ を持って $\mathrm{Var}(t^\star X + Y) = 0$ を満たす。分散が 0 の確率変数は確率 1 で定数であるため、確率 1 で $Y = -t^\star X + d$ となり、また $t^\star \neq 0$ である（$\mathrm{Var}(Y) > 0$ のため）。逆に、$Y = cX + d$（$c \neq 0$）であれば $\mathrm{Cov}(X,Y) = c\,\mathrm{Var}(X)$ かつ $\sigma_Y = |c|\sigma_X$ となるため、$\rho = c/|c| = \pm 1$ が得られる。■

相関は X と Y がどれだけ密接に直線に沿っているかをその直線の傾きの符号とともに測るが、傾きの大きさは無視する。実際、$Y = 0.5X$ と $Y = 2X$ はどちらも $\rho = 1$ である。4 セルの例では $\sigma_X = 1$、$\sigma_Y = 2$ であるため、$\rho = (4\theta - 2)/2 = 2\theta - 1$ となり、-1 から 1 まで（θ が 0 から 1 まで動くにつれて）変化する。Figure 36.11 は ρ の 2 つの死角を示している。同じ相関を持ちながら傾きが異なる 2 つの点群と、放物線上に並びながら相関が 0 になる点群である。

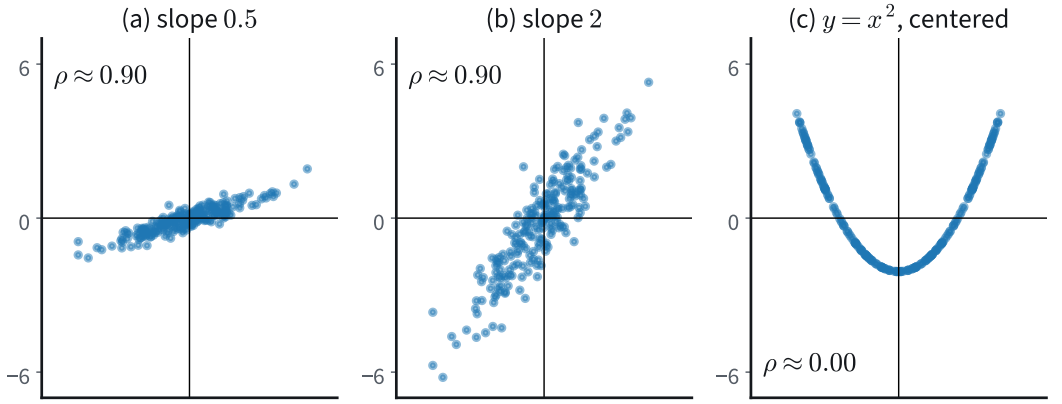


Figure 36.11: 相関は線形関係の密着度と符号を測り、傾きは測らない。(a) と (b) はともに $\rho \approx 0.9$ だが、傾きはそれぞれ 0.5 と 2 である。また非線形な関係も見逃す。(c) では Y は X の関数であるが、$\rho \approx 0$ である。

上の証明がコーシー・シュワルツの不等式 $|\mathbf{u}^\top \mathbf{v}| \le \|\mathbf{u}\|\,\|\mathbf{v}\|$ の古典的な証明そのものであることに気づくだろう。これは偶然ではなく、相関は文字通りコサインなのである。平均 0 で有限の分散を持つ確率変数の空間では、積の期待値が内積として働く（確率 1 で一致する 2 つの変数は同一視する）。中心化された変数 $\tilde{X}$ と $\tilde{Y}$ に対して、この内積は $\langle \tilde{X}, \tilde{Y} \rangle = \mathbb{E}[\tilde{X}\tilde{Y}] = \mathrm{Cov}(X,Y)$ であり、誘導される長さは標準偏差 $\|\tilde{X}\| = \sqrt{\mathbb{E}[\tilde{X}^2]} = \sigma_X$、そして相関はコサイン $\langle \tilde{X}, \tilde{Y} \rangle / (\|\tilde{X}\|\,\|\tilde{Y}\|)$ となる。幾何学的には、無相関な変数は直交しており、$\rho = \pm 1$ はそれらが平行であることを意味する。

36.2.2.3 共分散行列

n 個の変数を同時に扱う場合、すべての変数の対に共分散が存在するため、それらを行列にまとめるのが自然である。確率ベクトル $\mathbf{X} = (X_1, \dots, X_n)^\top$（平均は μ）に対し、**共分散行列**（covariance matrix）は偏差ベクトル自身の外積の期待値として次のように定義される。

$$\Sigma = \mathbb{E}\big[(\mathbf{X}-\mu)(\mathbf{X}-\mu)^{\top}\big], \qquad \Sigma_{ij} = \mathrm{Cov}(X_i, X_j). \tag{36.29}$$

対角成分には個々の分散が並び、非対角成分には対ごとの共分散が並ぶ。$\mathrm{Cov}(X_i, X_j) = \mathrm{Cov}(X_j, X_i)$ であるため、この行列は対称である。また半正定値でもある。任意の固定ベクトル $\mathbf{a}$ に対し、スカラー射影 $\mathbf{a}^{\top}\mathbf{X} = \sum_i a_i X_i$ の偏差は $\mathbf{a}^{\top}(\mathbf{X}-\mu)$ となるため、次式が成り立つ。

$$\mathrm{Var}(\mathbf{a}^{\top}\mathbf{X}) = \mathbb{E}\big[\mathbf{a}^{\top}(\mathbf{X}-\mu)(\mathbf{X}-\mu)^{\top}\mathbf{a}\big] = \mathbf{a}^{\top}\Sigma\,\mathbf{a} \;\geq\; 0.$$

言葉で言えば、二次形式 $\mathbf{a}^{\top}\Sigma\,\mathbf{a}$ はデータを方向 $\mathbf{a}$ に射影したときの分散にほかならず、分散が負になることは決してない。ダーツの例では $\Sigma = \mathrm{diag}(\frac{1}{4}, \frac{1}{4})$ であり、各座標は無相関で同じ広がりを持つ。本書の後半では、Σ の固有ベクトルが主成分分析（principal component analysis）の方向（33.5 節）として、また多変量ガウス分布の楕円状の等高線の主軸（36.3 節）として登場する。

36.2.3 確率変数の変換

機械学習では確率変数の変換が絶えず行われる。活性化関数は活性化前の値に作用し、サンプラは一様ノイズを変形し、正規化フロー（normalizing flow）は単純なガウス分布を複雑なデータ分布へと歪める。X が既知の密度を持ち、$Y = g(X)$ とおくとき、Y の密度はどうなるだろうか。

36.2.3.1 1 次元の規則

最初に思いつきそうな $p_Y(y) = p_X(g^{-1}(y))$ という推測は誤りであり、その理由はダーツの例からすでにわかる。ダーツの距離は $R = \sqrt{U}$（ただし U は $[0,1]$ 上の一様変数）として生成できるが、この写像は U の平坦な密度を傾いた密度 $2r$ へと変える。変換のもとで保存されるのが密度の高さではなく確率質量だからである。幅 Δx を持つ x の周りの狭い区間はおおよそ $p_X(x)\,\Delta x$ の確率を持ち、g によって幅おおよそ $\Delta y = |g'(x)|\,\Delta x$ の $y = g(x)$ の周りの区間へと写されるが、この区間も同じ確率を持たなければならない。

$$p_Y(y)\,|g'(x)|\,\Delta x \approx p_X(x)\,\Delta x.$$

g が数直線を引き伸ばす箇所では、同じ確率がより広い区間に広がるため、密度は拡大率 $|g'(x)|$ の分だけ薄くならなければならない。逆に g が数直線を圧縮する箇所では、密度は高くならなければならない。Figure 36.12 は $Y = e^X$ に対して面積の等しいこれら 2 つの帯を示している。

命題（1 次元の変数変換） **X が密度 p_X を持ち、この密度が開区間 I（ただし $P(X \in I) = 1$）上で連続であるとする。また $g : I \to \mathbb{R}$ は連続微分可能で $g' \neq 0$ が I 上で成り立ち、したがって g は狭義単調かつ単射であるとする。このとき $Y = g(X)$ は次の密度を持つ。**

$$p_Y(y) = p_X\big(g^{-1}(y)\big)\left|\frac{dg^{-1}}{dy}(y)\right| \tag{36.30}$$

これは区間 $g(I)$ 上での式であり、その外側では 0 である。

証明 まず g が増加関数であるとする。このとき $g(X) \leq y$ が成り立つのは $X \leq g^{-1}(y)$ のとき、かつそのときに限るため、累積分布関数は $F_Y(y) = F_X(g^{-1}(y))$ を $y \in g(I)$ において満たす。連鎖律および $F_X' = p_X$（36.1 節）を用いて微分すると、$p_Y(y) = p_X(g^{-1}(y))\,\frac{dg^{-1}}{dy}(y)$

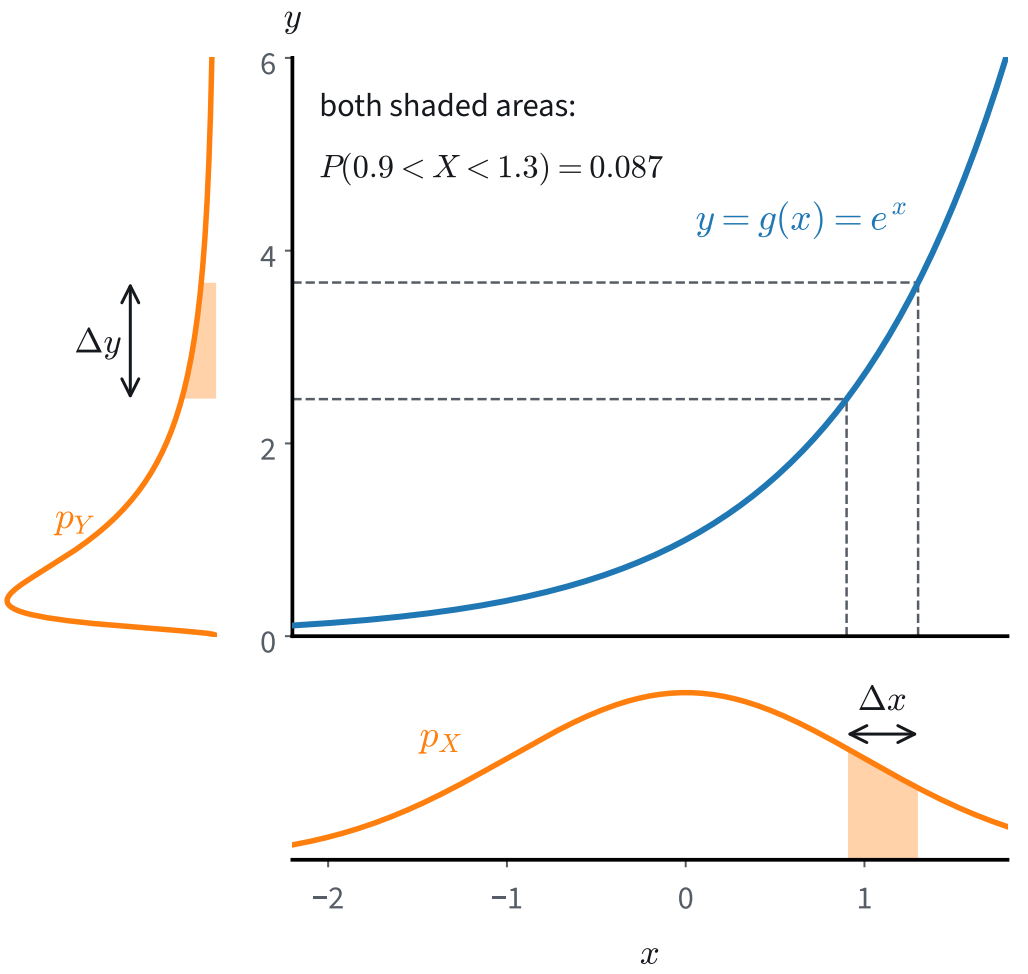


Figure 36.12: 変数変換。幅 Δx の p_X の下の区間（下）は、g によって幅 $\Delta y = g'(x)\,\Delta x$ の p_Y の下の区間（左）に写される。網掛けされた 2 つの面積は同じ確率を表すため、g が引き伸ばす箇所（$\Delta y > \Delta x$）では、密度 p_Y は因子 $|g'(x)|$ の分だけ低くならなければならない。ここでは $X \sim \mathcal{N}(0, 1)$、$Y = e^X$ としている。

が得られる。一方、g が減少関数であれば、$g(X) \le y$ は $X \ge g^{-1}(y)$ のときに成り立つため、$F_Y(y) = 1 - F_X(g^{-1}(y))$ となる。この導関数には負号が現れるが、それが dg^{-1}/dy の負号と打ち消し合うため、絶対値をとれば両方の場合を一度に扱える。いずれの場合も、F_Y は全確率を担う $g(I)$ 上で連続微分可能であるから、微分積分学の基本定理によりその導関数は Y の密度となる。$\blacksquare$

言葉で読み解くと、y における新しい密度は、y に写される点での元の密度に、逆写像の局所的な圧縮率を掛けたものである。$dg^{-1}/dy = 1/g'(x)$（ただし $x = g^{-1}(y)$）であるから、これは引き伸ばしの議論から導いた $1/|g'(x)|$ の因子とまったく同じである。

逆変換サンプリングは変数変換の一種であり、この公式を使えばダーツの距離のサンプラからその密度がただちに復元できる。$X = U$ を $I = (0,1)$ 上の一様変数とし、$g(u) = \sqrt{u}$ とおくと、$g^{-1}(r) = r^2$ かつ $|dg^{-1}/dr| = 2r$ となる。Equation 36.30 より、ダーツの距離の密度 $p_R(r) = 1 \cdot 2r$ が得られる。

g が単射でない場合、複数の入力点が同じ出力 y に写り得るため、それぞれの確率の寄与を単純に足し合わせればよい。

$$p_Y(y) = \sum_{x:\, g(x)=y} \frac{p_X(x)}{|g'(x)|},$$

これは、値 y の逆像が有限個しかなく、その各点で $g'(x) \neq 0$ を満たす場合に成り立つ。$Y = X^2$ かつ $y > 0$ のとき、2 つの逆像 $\pm\sqrt{y}$ はどちらも $|g'| = 2\sqrt{y}$ を満たすため、$p_Y(y) = [p_X(\sqrt{y}) + p_X(-\sqrt{y})]/(2\sqrt{y})$ となる。ここで $g' \neq 0$ という仮定は本質的である。ReLU 活性化関数は $g' = 0$ となる負の半直線全体 $(-\infty, 0)$ の確率質量をすべて 1 点 0 に潰してしまい、純粋な密度ではなく点質量（point atom）を生じさせる（演習問題 2）。

具体例として、$X \sim \mathcal{N}(0,1)$ が標準正規密度 $p_X(x) = e^{-x^2/2}/\sqrt{2\pi}$ を持つとし、$Y = e^X$ とおく。このとき $g^{-1}(y) = \log y$ および $|dg^{-1}/dy| = 1/y$（$y > 0$）であるから、変数変換の公式により次の**対数正規**（log-normal）密度が得られる。

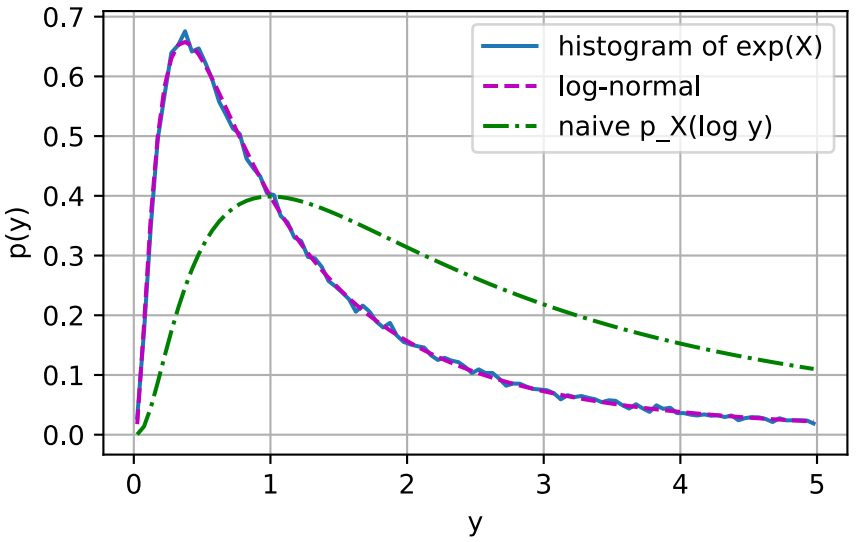
$$p_Y(y) = \frac{1}{y\sqrt{2\pi}} \exp\!\Big(-\tfrac{1}{2}(\log y)^2\Big), \qquad y > 0.$$

先頭の因子 $1/y$ が変数変換による補正項である。$\log p_Y(y) = -\log y - \tfrac{1}{2}(\log y)^2 + \text{const}$ の導関数を 0 とおくと $\log y = -1$ となるため、真の密度は $y = e^{-1} \approx 0.368$ でピークをとる。対照的に、素朴な推測 $p_X(\log y)$ は $y = 1$ でピークをとり、そもそも正当な密度ですらない。$y = e^x$ を代入すると全体の面積は $\int p_X(x)\, e^x\, dx = \mathbb{E}[e^X] = e^{1/2} \approx 1.65$ となってしまうからである（平方完成により $x - x^2/2 = \tfrac{1}{2} - \tfrac{1}{2}(x-1)^2$ となる）。

同じ計算からすべてのモーメント $\mathbb{E}[Y^k] = \mathbb{E}[e^{kX}] = e^{k^2/2}$ が求まり、そのどれもが有限である。それにもかかわらず、上側の裾 $P(Y > t) = P(X > \log t)$ は t のいかなる指数関数よりも遅く減衰する。したがって対数正規分布は裾が重い（heavy-tailed）が、コーシー分布とは異なりあらゆる次数の有限なモーメントを持つ。

10^5 個の e^X のサンプルのヒストグラムを、2 つの候補曲線と並べてプロットしてみよう。素朴な曲線が面積を 0.65 だけ余分に囲んでしまうことはすでにわかっている。図を見る前に、その余分な面積がどこに現れるか予想してみてほしい。

```
y = np.exp(rng.standard_normal(10**5))                # Y = e^X
edges = np.linspace(0, 5, 101)
mids = (edges[:-1] + edges[1:]) / 2
hist = np.histogram(y, edges)[0] / (len(y) * (edges[1] - edges[0]))
p_X = lambda x: np.exp(-x**2 / 2) / np.sqrt(2 * np.pi)
d2l.plot(mids, [hist, p_X(np.log(mids)) / mids, p_X(np.log(mids))],
         'y', 'p(y)', figsize=(5, 3),
         legend=['histogram of exp(X)', 'log-normal', 'naive p_X(log y)'])
```



ヒストグラムはサンプリングノイズの範囲内で対数正規密度によく一致しており、ピークは $y = 1$ よりずっと左側の e^{-1} 付近にある。素朴な曲線はピークの位置を誤っている。真の密度に y を掛けたものに等しいため、$y < 1$ ではデータを下回り、$y > 1$ では裾に向かうにつれて増大し続ける倍率でデータを上回るのである。

36.2.3.2 多変量の規則

多次元では、変換は長さではなく面積や体積を引き伸ばすため、拡大率は行列式になる。最も単純な場合として、線形変換 $\mathbf{Y} = \mathbf{A}\mathbf{X}$（ただし $\mathbf{A} = \operatorname{diag}(2, 3)$）から始めよう。この写像は単位正方形を、大きさ 2×3 で面積 $6 = \det \mathbf{A}$ の長方形へと写す。この長方形は元の正方形と同じ確率を持たなければならないため、$\mathbf{Y}$ の密度は対応する点における $\mathbf{X}$ の密度の 6 分の 1、すなわち $p_{\mathbf{Y}}(\mathbf{y}) = p_{\mathbf{X}}(\mathbf{A}^{-1}\mathbf{y})/6$ にならなければならない。

非線形写像では局所的な拡大率が点ごとに変化するが、これは重積分の変数変換の定理によって扱える（34.6 節）。g を $\mathbb{R}^n$ の 2 つの領域間の連続微分可能な全単射とし、その逆写像も連続微分可能であるとする。また、逆写像の偏微分を並べたヤコビ行列を $J_{g^{-1}}$ と書く。局所的に g^{-1} は体積を $\left|\det J_{g^{-1}}\right|$ 倍にスケールするため、$\mathbf{x} = g^{-1}(\mathbf{y})$ を領域 $\mathcal{D}$ の逆像上の積分に代入すると次式が得られる。

$$\int_{g^{-1}(\mathcal{D})} f(\mathbf{x})\, d\mathbf{x} = \int_{\mathcal{D}} f(g^{-1}(\mathbf{y}))\, \left|\det J_{g^{-1}}(\mathbf{y})\right|\, d\mathbf{y}.$$

ここで $\mathbf{Y} = g(\mathbf{X})$ とおく。事象 $\mathbf{Y} \in \mathcal{D}$ は事象 $\mathbf{X} \in g^{-1}(\mathcal{D})$ と同一であるため、$f = p_{\mathbf{X}}$ としてこの定理を適用すると次のようになる。

$$P(\mathbf{Y} \in \mathcal{D}) = \int_{g^{-1}(\mathcal{D})} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} = \int_{\mathcal{D}} p_{\mathbf{X}}(g^{-1}(\mathbf{y}))\, \left|\det J_{g^{-1}}(\mathbf{y})\right| d\mathbf{y}.$$

したがって、右辺の被積分関数が $\mathbf{Y}$ の密度となる。

$$\begin{aligned} p_{\mathbf{Y}}(\mathbf{y}) &= p_{\mathbf{X}}(g^{-1}(\mathbf{y}))\, \left|\det J_{g^{-1}}(\mathbf{y})\right|, \\ \log p_{\mathbf{Y}}(\mathbf{y}) &= \log p_{\mathbf{X}}(\mathbf{x}) - \log \left|\det J_g(\mathbf{x})\right| \\ &\qquad \text{(equivalently, at } \mathbf{x} = g^{-1}(\mathbf{y})). \end{aligned} \tag{36.31}$$

2 行目では $J_{g^{-1}}(\mathbf{y}) = J_g(\mathbf{x})^{-1}$ を用いた。サンプルを g に通すと、その対数密度から局所的な体積拡大率の対数が引かれるだけである。$\log|\det(J_2 J_1)| = \log|\det J_1| + \log|\det J_2|$ が成り立つため、写像を合成するとそれぞれの対数行列式の和が差し引かれる。これこそが、**正規化フロー**が厳密な尤度を計算する仕組みそのものである。すなわち、対数行列式を安価に評価できる可逆な層を積み重ねるのであり (Rezende and Mohamed 2015)、38.3.3 節では微分方程式のステップを層とするフローを構築する。また、演習問題 5 ではカップリング層をゼロから実装する。

最後に、ダーツの例でもう一度これを確認しよう。ダーツの位置を極座標である距離 R と角度 Θ により $(x, y) = (r\cos\theta, r\sin\theta)$ と表す。(r, θ) から (x, y) へのこの写像のヤコビ行列式は $\cos\theta \cdot r\cos\theta - (-r\sin\theta)\sin\theta = r$ であるため、多変量の規則から次式が得られる。

$$p_{R,\Theta}(r, \theta) = \frac{1}{\pi} \cdot r = 2r \cdot \frac{1}{2\pi}, \qquad 0 \le r \le 1,\ 0 \le \theta < 2\pi.$$

同時密度はきれいに因数分解される。角度 Θ は $[0, 2\pi)$ 上の一様分布に従い、距離 R は再び密度 $2r$ を（今度はヤコビ行列から直接）復元し、2 つの変数は独立である。デカルト座標ではダーツの X と Y は従属であったが、適切な座標系をとれば、まったく同じダーツが 2 つの独立な変数によって記述される。

本節では、平均に関する中心的な問いが 1 つ未解決のまま残されている。独立な n 個の抽出値の平均が分散 σ^2/n を持つこと（Equation 36.27）はわかったが、その分布はまだわかっていない。チェビシェフの不等式（36.1 節）を使えば、この分散から $\sigma^2/(n\epsilon^2)$ という上界（平均が ϵ 以上の誤差を生じる確率に対する上界）が得られるが、有界な抽出値やガウス分布からの抽出値に対してはこの上界はあまりに悲観的であり、真の誤差確率は n に関して指数関数的に減少する。36.3.2.3 節では平均がどの分布に近づくかを特定し、36.9 節ではその指数減衰率を確立する。ただし、そのどちらに取り組むにも、まずはダーツだけにとどまらない豊かな分布のカタログが必要である。

36.2.4 まとめ

周辺密度は不要な変数を積分消去することで得られ、条件付き密度は同時密度を細い帯に沿ってスライスして再正規化することで得られる。条件付き平均は最適な二乗誤差予測器であり、全分散の法則はある変数の分散を、条件付ける変数が説明する部分と説明できない部分とに分解する。共分散と相関は一般の従属性ではなく線形な関連を測るものであり、有限の分散を持つ独立な n 個の抽出値を平均すると分散は σ^2/n に減少する。滑らかな可逆写像のもとでは、密度はヤコビ行列式の絶対値で与えられる局所的な体積拡大率で割られる。ダーツの例はこれらすべての概念

を 1 つに結びつける。すなわち、デカルト座標では従属でありながら無相関であるが、極座標に移ると 2 つの独立な変数になる。

36.2.5 演習問題

1. **無相関だが従属。**白色化（たとえば 33.5 節の主成分を用いたもの）は、各座標を無相関にする。次の変数の対は、各座標がガウス分布に従い共分散が 0 になる例である。$X \sim \mathcal{N}(0,1)$ とし、S を X と独立で $P(S=1)=P(S=-1)=\tfrac{1}{2}$ を満たすランダムな符号とし、$Y = SX$ とおく。
   1. 累積分布関数を計算することで $Y \sim \mathcal{N}(0,1)$ であることを示し、さらに $\mathrm{Cov}(X,Y)=0$ であることを示せ。
   2. $P(|X|\le 1, |Y|\le 1)$ と $P(|X|\le 1)\,P(|Y|\le 1)$ を比較せよ。X と Y は独立か。
   3. $X+Y$ が 0 に点質量を持つため、$X+Y$ はガウス分布に従わないことを示せ。確率 1 で点 (X,Y) は 2 本の直線 $y=\pm x$ のいずれかの上に乗るが、直線は面積を持たないため直線上の二重積分は 0 である。各座標がそれぞれ密度を持つにもかかわらず、組 (X,Y) は同時密度を持たないことを結論づけよ。

2. **相関のある推定量の平均化。**本文では独立な抽出に対して $\mathrm{Var}(\bar{X}_n)=\sigma^2/n$ を導出した。従属な推定量の平均はこれとは異なる振る舞いをする。
   1. $Y_1,\dots,Y_n$ がそれぞれ分散 σ^2 を持ち、任意の対の間の相関が ρ であるとする（同じデータで訓練された n 個のアンサンブルメンバーの予測などがこれにあたる）。次式を示せ。

$$\mathrm{Var}\Big(\frac{1}{n}\sum_{i=1}^{n} Y_i\Big) = \sigma^2\Big(\rho + \frac{1-\rho}{n}\Big). \tag{36.32}$$

      $n\to\infty$ のとき、平均化によって何が達成されるか。
   2. $\mathbf{a}^\top \Sigma\, \mathbf{a} \ge 0$ を共分散行列 Σ（$Y_1,\dots,Y_n$ の共分散行列）に適用し、$\rho \ge -1/(n-1)$ であることを示せ。また、この下界を達成する変数の例を挙げよ。

3. [code] **ガウス変数のシグモイド変換。**$X\sim\mathcal{N}(0,s^2)$ をシグモイドユニットの活性化前の値とし、$Y=\mathrm{sigmoid}(X)$（その逆関数は $\mathrm{logit}(y)=\log(y/(1-y))$）とおく。
   1. Equation 36.30 を用いて、密度 p_Y を区間 $(0,1)$ 上で求めよ。
   2. どのような s に対して、$y=\tfrac{1}{2}$ で p_Y が極小値をとるか。$\log p_Y$ を $z=\mathrm{logit}(y)$ の関数として書くと計算が簡単になる。
   3. `#random-variables-change-of-variables` セルを修正し、Y のヒストグラムと p_Y を $s\in\{1,\sqrt{2},3\}$ のそれぞれについて比較せよ。$s=3$ におけるヒストグラムは、誤差逆伝播法がこれらのユニットの勾配を伝播させる際に通る導関数 $\mathrm{sigmoid}'(X)=Y(1-Y)$ について何を示唆するか。

4. **説明分散。**●回帰モデル h がターゲット X を入力 Y から予測するとする。このモデルが説明する分散の割合は $R^2 = 1 - \mathbb{E}\big[(X-h(Y))^2\big]/\mathrm{Var}(X)$ である。
   1. Equation 36.23 と全分散の法則 Equation 36.22 を用い、任意の予測器について $R^2 \le \mathrm{Var}(\mathbb{E}[X\mid Y])/\mathrm{Var}(X)$ が成り立つことを示せ。本文の 2 成分混合分布において X をラベル Z から予測する場合と、ダーツにおいて X を Y から予測する場合について、この上限の値を求めよ。

2. 入力が与えられたもとで、ターゲットは -1 または $+1$ をそれぞれ確率 $\frac{1}{2}$ でとるとする。期待二乗誤差を最小化する予測値は何か。また、ターゲットがその値をとる頻度はどれだけか。期待絶対誤差 $\mathbb{E}\big[|X - h(Y)|\big]$ を最小化する予測値は何か。

5. [code] **アフィンカップリング層とボックス・ミュラー法。●**正規化フローは、Equation 36.31 における対数行列式を安価に計算できる可逆写像を積み重ねる。RealNVP (Dinh, Sohl-Dickstein, et al. 2017) のアフィンカップリング層は、$\mathbf{x} \in \mathbb{R}^2$ を次式へと写す。

$$y_1 = x_1, \qquad y_2 = x_2 \, e^{s(x_1)} + t(x_1), \tag{36.33}$$

ここで s と t は任意の関数であり、たとえば小さなニューラルネットワークである。

1. 逆写像とヤコビ行列 $J_g(\mathbf{x})$ を書き下せ。また、$\log \left|\det J_g(\mathbf{x})\right| = s(x_1)$ が、s と t がどれほど複雑であっても成り立つことを示せ。
2. X_1 と X_2 を独立な標準正規変数とし、$s(x_1) = \tanh(x_1)$、$t(x_1) = x_1^2/2$ とおく。Equation 36.31 を用いて $\log p_{\mathbf{Y}}(\mathbf{y})$ を閉じた形で表せ。NumPy を用い、$p_{\mathbf{Y}}$ を領域 $[-6, 6] \times [-12, 24]$ 上の間隔 0.02 のグリッドで評価し、そのリーマン和が 1 に近いことを確かめよ。また、いくつかのランダムな点で、対数行列式を有限差分によるヤコビ行列の行列式と照合して確認せよ。
3. 第 1 座標はこの層をそのまま通過する。Y_1 の分布は、s と t が何であってもどうなるか。合成写像が両方の周辺分布を変化させられるようにするには、2 つ目のカップリング層をどのように配置すべきか。また、その合成写像の対数行列式はどうなるか。
4. フローは一様ノイズからガウス分布へ向かう向きにも構成できる。U_1, U_2 を独立な $(0, 1)$ 上の一様変数とし、$R = \sqrt{-2 \log U_1}$、$\Theta = 2\pi U_2$、$(Z_1, Z_2) = (R \cos\Theta, R \sin\Theta)$ とおく。1 次元の規則を用いて R の密度を求め、次に (R, Θ) の同時密度を求め、本文の極座標のヤコビ行列を使ってデカルト座標へ変換することで、Z_1 と Z_2 が独立な標準正規変数であることを示せ。これがボックス・ミュラー法である。10^5 個のサンプルを用いて、(Z_1, Z_2) の標本共分散行列が単位行列に近いことを確かめよ。

36.3 分布

36.1 節と 36.2 節では、確率質量関数、確率密度、期待値、分散といったランダムな量を扱うための一般的な言語を整備した。深層学習におけるあらゆる予測モデルは、明示しているかどうかにかかわらず、この言語を話している。二値分類器は 1 つの数値、すなわちラベルが 1 である確率を出力する。言語モデルは語彙に含まれるすべてのトークンに対して、総和が 1 になるよう制約された確率を出力する。回帰ネットワークは実数値を予測し、二乗誤差を最小化するように訓練される。いずれの場合も、モデルはラベルに対する確率分布を暗黙のうちに選択しており、その選択が最小化すべき損失関数を決定している。では、それらはどのような分布であり、どこから現れ、その平均と分散はどのように計算できるのだろうか。

驚くべきことに、たった 1 回のコイン投げから出発するだけで、この問いのほぼすべてに答えることができる。コインを 13 回投げて 9 回表が出たとする。このコインに偏りがない（公正である）としたら、この結果はどれほど珍しいのだろうか。この問いに答えるには、繰り返しコインを投げたときに表が出る回数の分布が必要になるが、実はこれは分類器がテスト例を何個正しく分類できるかを支配する分布とまったく同じである。すぐに確かめるように、9 回表が出る確率は約 9% である。

本節の中心的な考え方は、少数の基本的な分布がすべてその 1 回のコイン投げから生まれるという点にある。決まった回数のコイン投げで表の数を数えれば二項分布（binomial distribution）が得られる。めったに表が出ないコイン投げを多数回行う極限をとればポアソン分布（Poisson

distribution）が導かれ、次の希少な事象が起こるまでの待ち時間は指数分布（exponential distribution）を与える。通常のコイン投げを多数回足し合わせてスケールを調整すれば、ガウス分布（Gaussian distribution）が現れる。どの分布もより単純な要素から組み立てられているため、その平均と分散は構成法から直接、毎回短く本質を突いた議論によって導かれる。

適切な分布を選ぶことは、単なる記法の問題ではない。たとえば分散が小さすぎるカウントモデルを使うと、そこから計算されるエラーバーはどれも実際より見かけ上狭くなってしまう。本節の 3 つの結果、すなわちポアソン極限とその待ち時間、中心極限定理（central limit theorem）、そしてガウスベクトルに対する条件付けの公式は、本節の範囲を超えて広く重要となる。対照的に、離散一様分布（discrete uniform distribution）、ガンマ分布（Gamma distribution）、ラプラス分布（Laplace distribution）は初読時には流し読みしても構わない。

Figure 36.13 に、これから学ぶ各分布が互いにどのように構成されるかをまとめる。

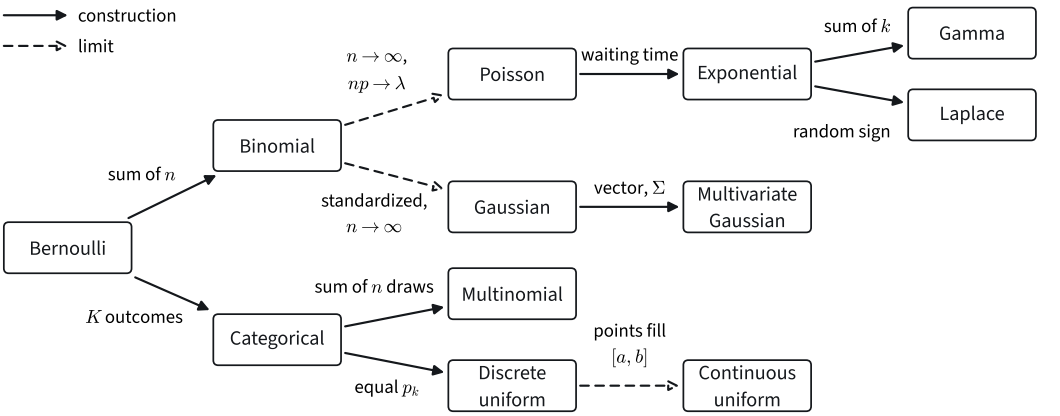


Figure 36.13: 本節で扱うすべての分布は 1 回のコイン投げから派生する。実線の矢印は（和、待ち時間、ランダムな符号、座標のベクトル化、または等確率の割り当てによって）新たな分布を構成する操作を表し、破線の矢印は試行回数や点の数が増大する極限を表す。ポアソン分布と指数分布は同一の事象のストリームを、それぞれ時間枠内の個数および到着間隔の時間として記述する。

36.3.1 離散分布

離散確率変数は、$\{0, 1\}$ や $\{0, 1, 2, \dots\}$ のような可算集合に値をとる。その分布は確率質量関数（probability mass function、pmf）$P(X = k) \geq 0$（ただし $\sum_k P(X = k) = 1$）によって定まる。平均は $\mathbb{E}[X] = \sum_k k\, P(X = k)$ であり、分散は $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ である。Figure 36.14 に、ここで登場する離散分布を通し例で評価した概観を示す。

36.3.1.1 ベルヌーイ分布

最も単純なランダムな量は、結果が 2 通りしかないものである。コインが表になるか裏になるか、ユーザーが広告をクリックするか無視するか、分類器がある例に正解するか不正解になるかといった具合である。この試行は、1 とラベル付けした結果が生じる確率 p というたった 1 つの数値で完全に記述される。

ベルヌーイ確率変数 $X \sim \mathrm{Bernoulli}(p)$ は、値 1 を確率 p でとり、値 0 を確率 $1 - p$ でとる（ただし $0 \leq p \leq 1$）。これら 2 つの場合は 1 つの式にすっきりとまとめられる。

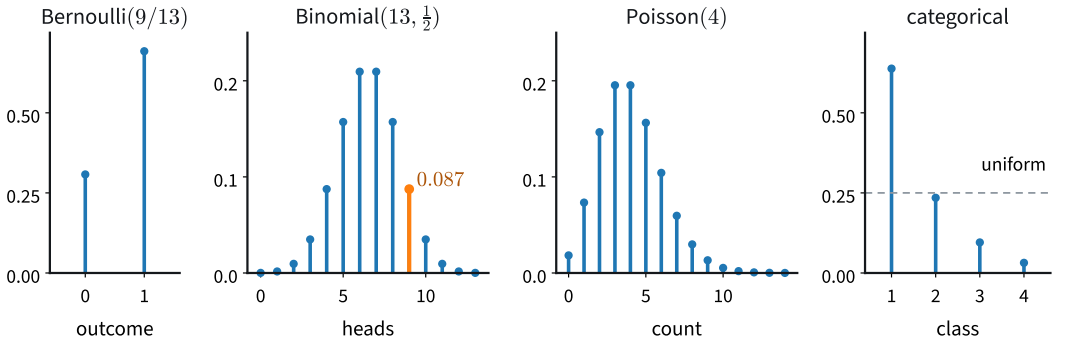


Figure 36.14: 通し例における 4 つの確率質量関数。Bernoulli$(9/13)$：観測された表の頻度に一致するコインの 1 回投げ。Binomial$(13, \frac{1}{2})$：公正なコインを 13 回投げたときの表の回数であり、オレンジ色の棒は $P(X = 9) = 715/8192 \approx 0.087$ を強調している。Poisson(4)：平均レートが毎分 4 件であるサーバーへのリクエスト数。ロジット $(2, 1, 0.1, -1)$ から生成された 4 クラスのカテゴリカル分布（一様確率 $1/4$ と対比して表示）。二項分布とポアソン分布はそれぞれの平均（6.5 と 4）付近でピークをとり共通の縦軸スケールを用い、結果の数が少ない残り 2 つのパネルは別の縦軸スケールを共有している。

$$P(X = x) = p^x (1 - p)^{1-x}, \qquad x \in \{0, 1\}. \tag{36.34}$$

ここで指数はスイッチの役割を果たす。$x = 1$ のときこの式は p になり、$x = 0$ のときは $1 - p$ になる。

モーメントの計算は容易である（1.6 節と同様）。平均は単に $\mathbb{E}[X] = 1 \cdot p + 0 \cdot (1 - p) = p$ となる。X は 0 と 1 の値しかとらないため $X^2 = X$ が成り立ち、$\mathbb{E}[X^2] = p$ でもある。したがって分散は $\mathrm{Var}(X) = p - p^2 = p(1 - p)$、すなわち 2 つの結果の確率の積となる。分散は結果が確定的であるとき（$p = 0$ または $p = 1$）に 0 となり、最大値 $\frac{1}{4}$ に達するのは公正なコイン（$p = \frac{1}{2}$）のときである。

深層学習では、二値分類器はラベル y が 1 に等しい確率の推定値 $\hat{p}$ を出力する。Equation 36.34 より、観測されたラベルに割り当てられる負の対数確率は $-\log P(y) = -y \log \hat{p} - (1 - y) \log(1 - \hat{p})$ であり、これが**二値交差エントロピー**（binary cross-entropy）損失である。したがって、二値交差エントロピーで分類器を訓練することは、各ラベルにベルヌーイ分布を当てはめることと等価である。

以下では 2 種類の異なるコインが登場するため、両者を明確に区別しておくとよい。冒頭の問いで扱ったのは $p = \frac{1}{2}$ の公正なコインである。これに対し、観測データと確率が一致するコインは $p = 9/13 \approx 0.69$ をもち、その 1 回投げの分散は $\frac{9}{13} \cdot \frac{4}{13} \approx 0.21$ である。

36.3.1.2 二項分布

冒頭のパズルに戻ろう。知りたいのは、公正なコインを 13 回投げたときに表が出る回数の分布、とりわけ 9 回表が出る確率がどれほどかということである。

まずは自明でない最小のケースである 2 回のコイン投げから始めよう。起こり得る 4 通りの結果は TT、HT、TH、HH であり、次を与える。

$$P(X=0)=(1-p)^2, \qquad P(X=1)=2p(1-p), \qquad P(X=2)=p^2.$$

$P(X=1)$ に現れる因子 2 は表 1 回と裏 1 回の並べ方が 2 通りあることを表しており、二項展開により 3 つの確率の和は $(p+1-p)^2=1$ となる。

一般に、n 回のコイン投げで表が出る回数は、各試行に対応する n 個の指示変数の和で表される。各コイン投げ $X_1,\dots,X_n$ が独立であり、それぞれが Bernoulli(p)（共通の p）に従うとき、それらの和は**二項**分布に従う。

$$X=\sum_{i=1}^{n} X_i \sim \text{Binomial}(n,p). \tag{36.35}$$

組み合わせは $\binom{n}{k}$ 通り（どの k 回が n 回のコイン投げのうち表になるかの選び方）あり、独立性により特定のどの並びも確率 $p^k(1-p)^{n-k}$ をもつ。並びの総数に 1 つの並びの確率を掛ければ、二項分布の確率質量関数が得られる。

$$P(X=k)=\binom{n}{k}p^k(1-p)^{n-k}, \qquad k=0,1,\dots,n. \tag{36.36}$$

$n=2$ のとき、これは先ほどの 3 つの確率を再現する。公正なコインの場合、13 回投げて 9 回表が出る並びは $\binom{13}{9}=715$ 通りあり、それぞれの確率は $2^{-13}=1/8192$ なので、次のようになる。

$$P(X=9)=\frac{715}{8192}\approx 0.087.$$

表が 9 回**以上**出る確率は $(715+286+78+13+1)/8192$ すなわち $1093/8192\approx 0.133$ である。裾確率（tail probability）が 13.3% であることからコインの公平性を疑うのに十分かどうかは統計学の問題であり、ここではひとまず脇に置いておく。

平均と分散。確率質量関数から直接 $\sum_k k\,P(X=k)$ を計算するのは手間がかかる。代わりに、和による表現 Equation 36.35 を使えば両方のモーメントが 1 行で求まる。期待値の線形性は確率変数が独立か従属かを問わず成り立つため、次を得る。

$$\mathbb{E}[X]=\sum_{i=1}^{n}\mathbb{E}[X_i]=np.$$

分散については、36.2 節で見たように独立な確率変数の共分散は 0 であるため、**独立な**変数の和の分散はそれぞれの分散の和になる。

$$\text{Var}(X)=\sum_{i=1}^{n}\text{Var}(X_i)=np(1-p).$$

このように、n 回の独立なコイン投げは、1 回投げの n 倍の平均と n 倍の分散をもつ。公正なコインを 13 回投げる場合、平均は 6.5、分散は 3.25、標準偏差は $\sqrt{3.25}\approx 1.80$ である。9 回表が出るという結果は、平均より標準偏差の $(9-6.5)/1.80\approx 1.4$ 倍だけ上に位置している。

2 つのモーメントの公式が異なる仮定に依拠している点は見落としやすいため注意しておきたい。平均 np はコイン投げが互いに従属していても成り立ち、i 回目のコイン投げが固有の確率 p_i をもつ場合でも平均は単に $\sum_i p_i$ となる。しかし分散の公式には、独立性が決定的に必要である。もし 13 回の「コイン投げ」が実際には公正なコインを 1 回投げた結果を 13 回コピーしたものだったとすれば、X は 0 または 13 の値しかとらず、その分散は $13^2 \cdot \frac{1}{4} = 42.25$ と、$np(1-p)$ の 13 倍にもなってしまう。

隣接項の比。@eq-mdl-binomial-pmf を直接計算しようとすると階乗が現れ、n が大きいときにはすぐにオーバーフローしてしまう。幸いなことに、隣り合う確率の比をとれば階乗がすべて約分されるため、階乗を 1 つも使わずに確率質量関数全体を計算できる。

$$\frac{\binom{n}{k+1}}{\binom{n}{k}} = \frac{k!\,(n-k)!}{(k+1)!\,(n-k-1)!} = \frac{n-k}{k+1}, \qquad \text{so} \qquad \frac{P(X=k+1)}{P(X=k)} = \frac{n-k}{k+1}\cdot\frac{p}{1-p}.$$

この比には直接的な組み合わせ論的意味がある。k 個の表からなる集合から出発して、残りの $n-k$ 個の裏のどれか 1 つを選んで表に裏返すことを考えると、得られる $k+1$ 個の表の各集合には $k+1$ 通りの経路で到達でき、因子 $p/(1-p)$ が裏の確率を表の確率へと置き換える。$P(X=0) = (1-p)^n$ から出発してこの比を繰り返し掛けていけば、確率質量関数全体が生成される。さらに、この比は 1 を超えるのが $k+1 < (n+1)p$ の間であり、その後は 1 を下回るため、二項分布の確率質量関数は平均付近の単一のピークまで滑らかに上昇してから減少することがわかる。

この漸化式は Python でわずか 3 行で書ける。以下でポアソン極限とガウス分布の釣鐘曲線の両方をこれと比較するため、再利用可能な関数にまとめておこう。

```
def binomial_pmf(n, p, kmax):
    """P(X=k) for k = 0..kmax, built from P(X=0) by the successive ratio."""
    pmf = [(1 - p) ** n]
    for k in range(kmax):
        pmf.append(pmf[-1] * max(n - k, 0) / (k + 1) * p / (1 - p))
    return np.array(pmf)

n, p = 13, 0.5
pmf = binomial_pmf(n, p, n)
print(f'P(X=9): by the ratio {pmf[9]:.4f}, by counting {math.comb(13, 9) / 2**13:.4f}')
print(f'P(X>=9) = {pmf[9:].sum():.4f}')
print(f'mean np = {n * p}, variance np(1-p) = {n * p * (1 - p)}')
```

```
P(X=9): by the ratio 0.0873, by counting 0.0873
P(X>=9) = 0.1334
mean np = 6.5, variance np(1-p) = 3.25
```

期待どおり、比の漸化式から $715/8192 \approx 0.0873$ と裾確率 0.1334 が再現される。

36.3.1.3 ポアソン分布

現実世界のカウントデータの多くには、自然な上限 n が存在しない。ある Web サーバーが毎分平均 $\lambda = 4$ 件のレートでリクエストを受け取っており、到着数に決まった上限はないとしよう。リクエストが 1 件も来ない静かな 1 分間や、10 件以上のリクエストが突発的に押し寄せるバーストは、どれほどの確率で起こるだろうか。

これをコイン投げの枠組みに結びつけるため、1 分間を n 個の微小な部分区間に分割してみよう。ここでは 2 つの単純化の仮定をおく。各部分区間は十分に短く同じスライスに 2 件以上のリクエストが到着することはないとし、各スライスは到着確率が等しく $4/n$ である独立な試行として振る舞うものとする。これらの仮定のもとで、1 分間の合計件数は $\mathrm{Binomial}(n, 4/n)$ に従う。その平均は $n \cdot \frac{4}{n} = 4$ となってどの n でも一定であり、分散 $4(1-\frac{4}{n})$ は着実に 4 へと近づいていく。リクエストが 0 件である静かな 1 分間の確率は $(1-\frac{4}{n})^n$ であり、これは 0.0115（$n=20$）、0.0169（$n=100$）、0.0182（$n=1000$）となって $e^{-4} \approx 0.0183$ に収束する。スライスを細かくするにつれて確率は安定し、きれいな極限分布の存在を示唆している。

極限の導出。 $\mathrm{Binomial}(n, \lambda/n)$ においてカウント k を固定し、$n \to \infty$ としたときの極限を考えよう。二項分布の確率質量関数を書き直すと 4 つの因子が現れ、そのうち 3 つは初等的な極限をもつ。

$$\binom{n}{k}\left(\frac{\lambda}{n}\right)^k\left(1-\frac{\lambda}{n}\right)^{n-k} = \underbrace{\frac{n(n-1)\cdots(n-k+1)}{n^k}}_{\to 1}\cdot\frac{\lambda^k}{k!} \cdot \underbrace{\left(1-\frac{\lambda}{n}\right)^{n}}_{\to e^{-\lambda}} \cdot \underbrace{\left(1-\frac{\lambda}{n}\right)^{-k}}_{\to 1}.$$

第 1 の因子はそれぞれ 1 に近づく k 個の分数の積であり、第 3 の因子は指数関数を定義する古典的な極限 $(1-\lambda/n)^n \to e^{-\lambda}$ であり、最後の因子は k が固定されているため 1 に近づく。成功確率の任意の列 p_n（ただし $np_n \to \lambda$）に対しても $(1-p_n)^n \to e^{-\lambda}$ が成り立つため、まったく同じ議論が通用する。この結果は**希少事象の法則**（law of rare events）として知られ、残った因子からレート $\lambda > 0$ の**ポアソン**確率質量関数が得られる。

$$P(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}, \qquad k = 0, 1, 2, \ldots \tag{36.37}$$

これを $X \sim \mathrm{Poisson}(\lambda)$ と書く。個々には起こりにくい多数の独立な機会からカウントが生じ、全体の期待値が一定の λ であるときにはいつでもポアソン分布が現れる。確率の総和が 1 になることは、テイラー級数 $\sum_{k\ge 0} \lambda^k/k! = e^{\lambda}$ から直ちにわかる。

平均と分散。 両方のモーメントは確率質量関数から直接読み取れる。$\mathbb{E}[X]$ の和では $k=0$ の項が消え、因子 k が $k!$ と約分され、添字を $j = k-1$ にずらせばポアソン確率変数の全確率が残る。

$$\mathbb{E}[X] = \sum_{k\ge 1} k\,\frac{e^{-\lambda}\lambda^k}{k!} = \lambda \sum_{k\ge 1} \frac{e^{-\lambda}\lambda^{k-1}}{(k-1)!} = \lambda \sum_{j\ge 0} \frac{e^{-\lambda}\lambda^{j}}{j!} = \lambda.$$

まったく同じ手法を $k(k-1)$ に適用すると $\mathbb{E}[X(X-1)] = \lambda^2$ が得られ、そこから分散が導かれる。

$$\mathrm{Var}(X) = \mathbb{E}[X(X-1)] + \mathbb{E}[X] - \mathbb{E}[X]^2 = \lambda^2 + \lambda - \lambda^2 = \lambda.$$

ポアソン分布を特徴づける性質は、平均と分散が等しいことである。先ほどの Web サーバーでは $P(X=0) = e^{-4} \approx 0.018$、$P(X=4) = \frac{4^4}{4!}e^{-4} \approx 0.195$、$P(X \ge 10) \approx 0.008$ となり、リクエストが来ない静かな 1 分間は 1 時間に約 1 回、10 件以上のバーストは約 2 時間に 1 回起こる。平均と分散が等しいというこの性質は、鋭い診断基準にもなる。多数の分にわたって観測したリ

クエスト数の分散が平均の数倍に達しているなら、モデル化の仮定の少なくとも 1 つが破綻しており（時間帯によってトラフィックが変動しているか、リクエストが連鎖的な後続リクエストを引き起こしているのかもしれない）、ポアソンモデルから求めたエラーバーは危険なほど狭くなってしまう。ポアソン分布のレート自体を確率変数とする標準的な対処法は、演習問題 1 で扱う。

ポアソン過程。単位時間あたりレート λ で区間を細分化する描像を時間軸全体に広げると、連続時間で事象が到着するモデルが得られる。長さ t の任意の時間枠には nt 個のスライスが含まれるため、その事象数は $\mathrm{Binomial}(nt, \lambda/n)$ となり、これは $\mathrm{Poisson}(\lambda t)$ に収束する。重ならない時間枠は互いに素なスライスから構成されるため、それぞれのカウントは独立になる。この連続時間の事象ストリームが、レート λ の**ポアソン過程**（Poisson process）である。次節では、事象間の待ち時間が指数分布に従うことを見る。

二項分布と同様に、ポアソン分布の確率質量関数も $P(X=k+1)/P(X=k) = \lambda/(k+1)$ という単純な隣接項の比を満たす。これを使ってサーバーの確率を計算し、シミュレーションによる 100,000 分間のデータとモーメントを照合してみよう。

```
rng = np.random.default_rng()

def poisson_pmf(lam, kmax):
    """P(X=k) for k = 0..kmax, built from P(X=0) = e^-lam by the ratio lam/(k+1)."""
    pmf = [math.exp(-lam)]
    for k in range(kmax):
        pmf.append(pmf[-1] * lam / (k + 1))
    return np.array(pmf)

pmf = poisson_pmf(4.0, 60)
print(f'P(0) = {pmf[0]:.3f}, P(4) = {pmf[4]:.3f}, P(X>=10) = {pmf[10:].sum():.3f}')
counts = rng.poisson(4.0, 100000)
print(f'100,000 simulated minutes: mean {counts.mean():.2f}, '
      f'variance {counts.var(ddof=1):.2f}')
```

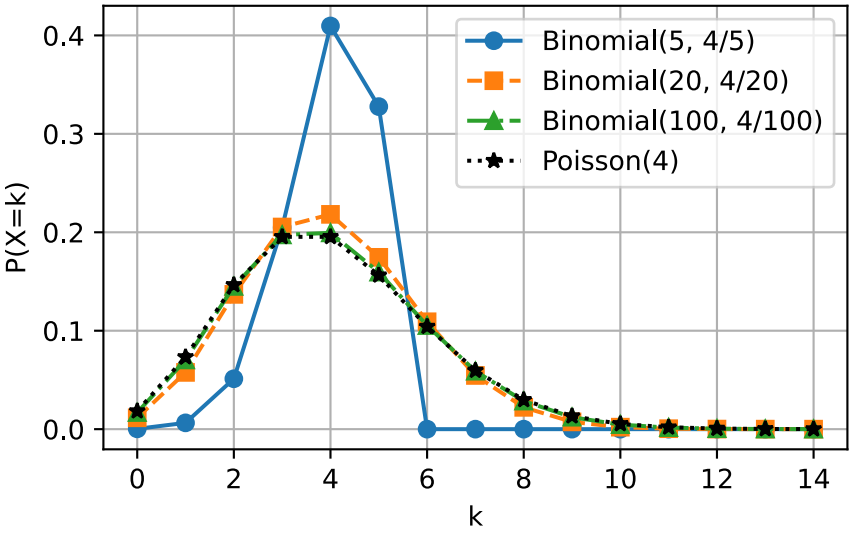
```
P(0) = 0.018, P(4) = 0.195, P(X>=10) = 0.008
100,000 simulated minutes: mean 4.00, variance 4.01
```

シミュレーションで得られたカウントは、平均も分散も 4 のすぐ近くに収まっている。

二項分布からポアソン分布への極限。$\mathrm{Binomial}(n, 4/n)$ はどれほど速く $\mathrm{Poisson}(4)$ に収束するのだろうか。両者の確率質量関数を重ねて描き、**全変動距離**（total variation distance）$\frac{1}{2}\sum_k |P_{\mathrm{Bin}}(k) - P_{\mathrm{Poi}}(k)|$ を用いてずれを測ってみよう。この距離は、任意の事象に対して 2 つの分布が割り当てる確率の差の最大値に等しい。

```
ns, ks = (5, 20, 100), np.arange(15)
d2l.plot(ks, [binomial_pmf(n, 4 / n, 14) for n in ns] + [poisson_pmf(4.0, 14)],
         xlabel='k', ylabel='P(X=k)', fmts=('o-', 's--', '^-.', 'k*:'),
         legend=[f'Binomial({n}, 4/{n})' for n in ns] + ['Poisson(4)'],
         figsize=(5, 3))
for n in (20, 100, 1000):
    tv = 0.5 * np.abs(binomial_pmf(n, 4 / n, 60) - poisson_pmf(4.0, 60)).sum()
    print(f'n = {n:4d}: total-variation distance {tv:.3f}')
```

```
n =   20: total-variation distance 0.056
n =  100: total-variation distance 0.010
n = 1000: total-variation distance 0.001
```



$n = 5$ では、二項分布は 5 件を超えることができず、ピークも目立って高すぎる。$n = 100$ になると、2 つの確率質量関数は見た目では区別がつかなくなる。全変動距離は 0.056（$n = 20$）から 0.010（$n = 100$）、さらに 0.001（$n = 1000$）へと低下し、ほぼ $1/n$ に比例して縮小する。

36.3.1.4 カテゴリカル分布と多項分布

K クラスの分類器や、語彙から次のトークンを予測する言語モデルは、3 つ以上の結果の中から 1 つを選ぶ。カテゴリカル分布とは、要するに K 面のコインである。

カテゴリカル確率変数は $\{1, \dots, K\}$ に値をとり、その確率はベクトル $\mathbf{p} = (p_1, \dots, p_K)$（ただし $p_k \geq 0$ かつ $\sum_k p_k = 1$）で指定される。

$$P(X = k) = p_k. \tag{36.38}$$

$K = 2$ のとき、これはベルヌーイ分布に帰着する。ニューラルネットワークでは確率を直接予測することはまれであり、代わりにネットワークは**ロジット**と呼ばれる制約のない実数スコア $z_1, \dots, z_K$ を出力し、**ソフトマックス**関数によってカテゴリカル分布へと変換する。

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}. \tag{36.39}$$

指数関数をとることで全成分が正になることが保証され、総和で割ることで和が 1 になる。たとえばロジット $\mathbf{z} = (2, 1, 0.1, -1)$ からは $\mathbf{p} \approx (0.638, 0.235, 0.095, 0.032)$ が得られる。真のクラスをワンホットベクトル $\mathbf{y}$ と書けば、観測されたラベルに割り当てられる負の対数確率は $-\log p_y = -\sum_k y_k \log p_k$ となり、これは 3.1.2 節の**ソフトマックス交差エントロピー**（softmax cross-entropy）損失にほかならない。二値交差エントロピーがベルヌーイ分布を当てはめるのとまったく同様に、ソフトマックス交差エントロピーは各ラベルにカテゴリカル分布を当てはめる。

離散一様分布。最も単純なカテゴリカル分布は、すべての結果に同じ確率を与えるものである。$\{1, \dots, n\}$ 上で $p_k = 1/n$ としたものが**離散一様**分布 Unif$\{1, \dots, n\}$ であり、公正なサイコロ

や一様ランダムに抽出されたインデックスを表す。その平均と分散は、よく知られた和の公式 $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ と $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ から導かれる。

$$\begin{aligned}\mathbb{E}[X] &= \frac{1}{n}\sum_{i=1}^{n} i = \frac{n+1}{2},\\ \operatorname{Var}(X) &= \frac{(n+1)(2n+1)}{6} - \frac{(n+1)^2}{4}\\ &= \frac{(n+1)\bigl(2(2n+1) - 3(n+1)\bigr)}{12} = \frac{n^2-1}{12}.\end{aligned}$$

平均はちょうど中点に位置し、分散は結果が長さ $n-1$ の範囲に広がるため n^2 のオーダーで増大する。公正な 6 面サイコロの場合、平均は 3.5、分散は $35/12 \approx 2.92$ である。

多項分布。ベルヌーイ試行を n 回繰り返すと二項分布が得られるのと同じように、カテゴリカル分布から n 個の独立なサンプルを引き、各クラスの出現回数を集計すると**多項**分布（multinomial distribution）が得られる。カウントベクトル $\mathbf{x} = (x_1, \dots, x_K)$（制約 $\sum_k x_k = n$ を満たす）の確率質量関数は次で与えられる。

$$P(\mathbf{x}) = \frac{n!}{x_1! \cdots x_K!} \prod_{k=1}^{K} p_k^{x_k}. \tag{36.40}$$

二項分布のときと同様に、先頭の多項係数は n 回の抽出でカウント $\mathbf{x}$ を生じる並びの総数を数え上げており、後ろの積はいずれか 1 つの並びが生じる確率を与える。$K = 2$ のとき係数は $\binom{n}{x_1}$ に帰着し、二項分布が復元される（$13!/(9!\,4!) = 715$）。任意の 1 つのクラス k だけに注目すれば、各抽出はクラス k に入るか入らないかのいずれかであるため、$x_k \sim \text{Binomial}(n, p_k)$ となり、$\mathbb{E}[x_k] = np_k$、$\operatorname{Var}(x_k) = np_k(1-p_k)$ が成り立つ。全クラスを同時に考えると、カウントベクトルの平均は $n\mathbf{p}$、共分散行列は $n\bigl(\operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^{\top}\bigr)$ となる（演習問題 7 で導出する）。非対角成分がすべて負であることに注目しよう。抽出の総数 n が固定されているため、あるクラスの回数が増えれば必ず他のクラスの回数が減るからである。$n = 1$ のとき、カウントベクトルは単にカテゴリカル分布からの 1 回の抽出をワンホットエンコーディングで表したものになる。

例のロジットに対するソフトマックス確率を計算し、得られたカテゴリカル分布から 10,000 個のサンプルを抽出して経験頻度を確認してみよう。

```
z = np.array([2.0, 1.0, 0.1, -1.0])          # logits for K = 4 classes
p_cat = np.exp(z) / np.exp(z).sum()           # softmax
draws = rng.choice(4, size=10000, p=p_cat)
print('softmax p   =', p_cat.round(3))
print('frequencies =', (np.bincount(draws, minlength=4) / 10000).round(3))
```

```
softmax p   = [0.638 0.235 0.095 0.032]
frequencies = [0.639 0.234 0.095 0.033]
```

経験頻度は約 1 パーセントポイント以内の誤差で $\mathbf{p}$ と一致している。

36.3.2 連続分布

連続確率変数は、確率密度関数 $p(x) \geq 0$（ただし $\int p(x)\,dx = 1$）によって記述される。36.1 節で見たとおり、確率は密度曲線の下の面積であり、期待値と分散は積分で与えられる。まずは指

数分布から始めよう。ポアソン過程によってすでにその準備が整っているからである。

36.3.2.1 指数分布

先ほどの Web サーバーに戻ろう。ポアソン分布からは、1 分間に何件のリクエストが到着するかがわかる。一方、タイムアウトを設定するエンジニアは、同じトラフィックのストリームに対して密接に関連する別の問いを立てる。次のリクエストが到着するまでどれだけの時間待つことになるだろうか、という問いである。

興味深いことに、これら 2 つの問いは同じ答えを共有している。次のリクエストまで 1 分より長く待つことは、その 1 分間に到着するリクエストが 0 件であることとまったく同じ事象であるため、$P(\text{wait} > 1) = P(N = 0) = e^{-4} \approx 0.018$ となり、すでに計算したポアソン確率と一致する。Figure 36.15 に、両方の分布が単一の事象ストリームをどのように記述するかを示す。

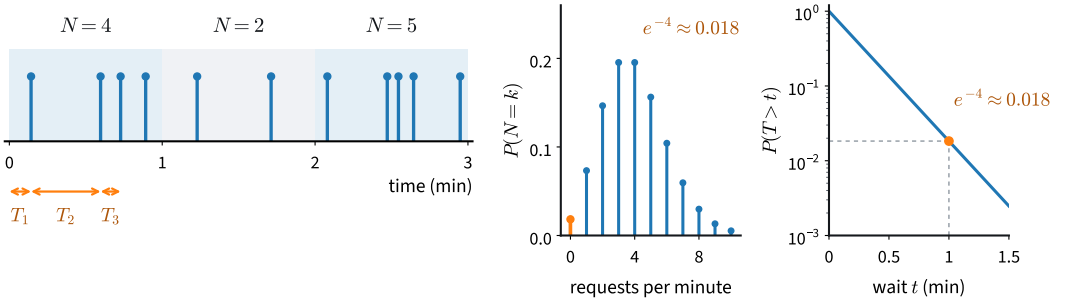


Figure 36.15: 毎分平均 4 件のレートで到着する 1 本のリクエストストリームを 2 通りの見方で捉えた図。(a) 3 分間にわたる事象発生時刻。長さ 1 の各時間枠には $N = 4$、2、5 件のリクエストが含まれ、T_1、T_2、T_3 は最初の 3 つの待ち時間を表す。(b) 1 分間のリクエスト数は $\text{Poisson}(4)$ に従うため、静かな 1 分間の確率は $e^{-4} \approx 0.018$ である。(c) 待ち時間が t 分を超える確率は e^{-4t} となり、対数軸上で直線を描く。1 分より長く待つことと静かな 1 分間を観測することは同じ事象である。

まったく同じ推論が任意の時間長 $t \geq 0$ に当てはまる。レート λ のポアソン過程において、待ち時間 T（時刻 0 から最初の事象が起こるまでの時間）が t を超えるのは、区間 $[0, t]$ に事象が 1 つも含まれないとき、かつそのときに限る。$[0, t]$ 内のカウントは $\text{Poisson}(\lambda t)$ に従うため、次が得られる。

$$P(T > t) = P(N(t) = 0) = e^{-\lambda t}, \qquad t \geq 0.$$

したがって累積分布関数（cumulative distribution function）は $F(t) = 1 - e^{-\lambda t}$ であり、これを微分するとレート $\lambda > 0$ の**指数**分布の確率密度が得られる。

$$p(t) = \lambda e^{-\lambda t} \quad (t \geq 0), \qquad p(t) = 0 \quad (t < 0), \qquad F(t) = 1 - e^{-\lambda t}, \qquad T \sim \text{Exp}(\lambda). \tag{36.41}$$

言い換えれば、この密度は空の時間枠が生じる確率が減衰する速さを測っている。ポアソン過程では、任意の到着時刻より未来のスライスは新たな独立試行であるため、各事象の後の待ち時間も同じ分布に従う。すなわち、連続する到着の間隔は互いに独立な $\text{Exp}(\lambda)$ 確率変数となる。

平均と分散。部分積分（integration by parts）を 2 回行えばモーメントはすぐに求まる。$\int_0^\infty u\,e^{-u}\,du = [-ue^{-u}]_0^\infty + \int_0^\infty e^{-u}\,du = 1$ および $\int_0^\infty u^2 e^{-u}\,du = [-u^2e^{-u}]_0^\infty + 2\int_0^\infty u\,e^{-u}\,du = 2$ を用い、$u = \lambda t$ と変数変換すると次を得る。

$$\mathbb{E}[T] = \int_0^\infty t\,\lambda e^{-\lambda t}\,dt = \frac{1}{\lambda}, \qquad \mathbb{E}[T^2] = \frac{2}{\lambda^2}, \qquad \mathrm{Var}(T) = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}.$$

直感のとおり、到着レートが高いほど待ち時間は短くなる。毎分 4 件のリクエストがある場合、平均待ち時間は $\frac{1}{4}$ 分すなわち 15 秒である。中央値（median）m は $F(m) = \frac{1}{2}$ の解として $m = \log 2/\lambda \approx 10.4$ 秒と求まる。つまり平均は 15 秒であるにもかかわらず、全待ち時間の半分は約 10 秒未満で終わる。密度が $t = 0$ でピークをとり右側に長い裾を引くため、平均が中央値より大きく引き上げられるのである。

無記憶性。リクエストが来ないまま、すでに s 分間待ったとしよう。そこからさらに少なくとも t 分間待たなければならない確率はいくらだろうか。

$$P(T > s + t \mid T > s) = \frac{P(T > s + t)}{P(T > s)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} = e^{-\lambda t} = P(T > t).$$

すでに待った時間は、残りの待ち時間について何の情報ももたらさない。$(0, \infty)$ 上のすべての連続分布のうち、この**無記憶**性（memoryless property）をもつのは指数分布だけである（演習問題 2 で証明する）。無記憶性はまた、このモデルが適用できる限界をも示している。使用に伴ってフィラメントが摩耗する電球は、古くなるほど次の 1 時間に切れる確率が高まるため、その寿命は指数分布には従わない。

ガンマ分布。では、時刻 T_k すなわち k 番目のリクエストが到着するまでの時間を問うたらどうなるだろうか。T_k は k 個の独立な指数待ち時間の和であるため、期待値の線形性と独立性から直ちに平均 k/λ と分散 k/λ^2 がわかる。その密度もポアソン過程から直接導かれる。k 番目の到着が時刻 t より後に起こるのは、高々 $k-1$ 個の事象しか $[0, t]$ 内に起こらないとき、かつそのときに限るので、次が成り立つ。

$$P(T_k > t) = P(N(t) \le k - 1) = \sum_{j=0}^{k-1} e^{-\lambda t}\frac{(\lambda t)^j}{j!}.$$

第 j 項を t に関して微分すると、積の微分法則により $-\lambda e^{-\lambda t}\frac{(\lambda t)^j}{j!} + \lambda e^{-\lambda t}\frac{(\lambda t)^{j-1}}{(j-1)!}$ が現れる。この第 2 項が 1 つ前の項の第 1 項と打ち消し合うことに注目しよう。和全体が望遠鏡和として相殺され、最後（$j = k-1$）の項の第 1 項だけが残る。

$$p(t) = -\frac{d}{dt}P(T_k > t) = \frac{\lambda^k\, t^{k-1} e^{-\lambda t}}{(k-1)!}, \qquad t > 0.$$

$k = 1$ のとき、指数分布の密度が復元される。この分布族を整数 k 以外にも拡張するため、$(k-1)!$ を**ガンマ関数**（gamma function）$\Gamma(z) = \int_0^\infty u^{z-1}e^{-u}\,du$（$z > 0$）に置き換える。部分積分により $\Gamma(z+1) = z\Gamma(z)$ が示され、$\Gamma(1) = 1$ であるから、すべての非負整数 n に対して $\Gamma(n+1) = n!$ が成り立つ。これにより、形状 $\alpha > 0$、レート $\beta > 0$ の**ガンマ分布**が次のように定義される。

$$p(t) = \frac{\beta^{\alpha}\, t^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)}, \qquad t > 0, \qquad T \sim \text{Gamma}(\alpha, \beta). \tag{36.42}$$

$u = \beta t$ と置換すると $p(t)$ の積分は $\Gamma(\alpha)/\Gamma(\alpha) = 1$ に帰着し、同じ置換から $\mathbb{E}[T] = \Gamma(\alpha+1)/(\beta\,\Gamma(\alpha)) = \alpha/\beta$ および $\text{Var}(T) = \alpha/\beta^2$ が得られる。$\alpha = k$、$\beta = \lambda$ のとき、これらは k 番目の待ち時間のモーメントと一致する。Web サーバーの例では、4 番目のリクエストは平均 1 分、標準偏差 0.5 分で到着し、$P(T_4 > 1) = P(N(1) \le 3) \approx 0.43$ となる。つまり平均は 4 件であっても、全体の 43% の分では到着するリクエストが 4 件未満になる。

サンプリング。 36.1 節で見たように、逆変換法（inverse-transform method）では一様乱数 U に対して可逆な累積分布関数 F の逆関数を計算することで、任意の連続分布からのサンプル $X = F^{-1}(U)$ に変換できる。指数分布では $F^{-1}(q) = -\log(1-q)/\lambda$ であるため、$T = -\log(1-U)/\lambda$ となる。$1-U$ と U は同じ分布に従うが、実装上は $1-U$ と書くことが重要である。乱数生成器（random number generator）は $[0,1)$ の値を返すため、$U = 0$ が起こり得て $-\log U$ が無限大へオーバーフローしてしまうからである。

ポアソン過程と指数分布の結びつきが寸分の隙もないものであれば、この構成を逆向きに動かせるはずである。すなわち、独立な指数分布の間隔を生成し、それを累積して到着時刻を作り、1 分あたりのポアソンカウントを復元できるはずだ。60,000 個の間隔をレート 4 で生成し、最初の 10,000 分間を調べて検証してみよう。

```
lam = 4.0                                       # requests per minute
gaps = -np.log(1 - rng.random(60000)) / lam     # inverse transform: Exp(lam)
times = np.cumsum(gaps)                         # event times, about 15,000 min
counts = np.bincount(times[times < 10000].astype(int), minlength=10000)
print(f'requests per minute: mean {counts.mean():.2f}, variance {counts.var(ddof=1):.2f}')
print(f'silent minutes {np.mean(counts == 0):.4f},   waits over 1 min '
      f'{np.mean(gaps > 1):.4f},   e^-4 = {math.exp(-4):.4f}')
print(f'mean wait {gaps.mean():.3f} min,   minutes with at most 3 requests '
      f'{np.mean(counts <= 3):.3f}')
```

```
requests per minute: mean 3.99, variance 4.00
silent minutes 0.0204,   waits over 1 min 0.0191,   e^-4 = 0.0183
mean wait 0.252 min,   minutes with at most 3 requests 0.434
```

シミュレーションによるカウントは確かに 4 に近い平均と分散をもっている。静かな 1 分間の割合と 1 分を超える待ち時間の割合はどちらも e^{-4} のすぐ近くにあり、リクエストが高々 3 件である分の割合もガンマ分布による予測 $P(T_4 > 1) \approx 0.43$ と一致している。指数分布に従う間隔を生成することで、ポアソンカウントが再現されるのである。

36.3.2.2 一様分布

本節のあらゆるサンプラは、$[0,1)$ から一様ランダムに抽出された 1 つの数値という共通の素材から出発する。この分布を任意の有限区間へと一般化したものが、連続一様分布（continuous uniform distribution）である。

$a < b$ を満たす区間 $[a,b]$ 上の**一様**分布 $U(a,b)$ は一定の確率密度をもち、その値は積分すると 1 になるという条件から定まる。

$$p(x) = \begin{cases} \dfrac{1}{b-a} & x \in [a,b], \\ 0 & \text{otherwise.} \end{cases} \tag{36.43}$$

同じ長さの部分区間にはすべて同じ確率、すなわちその長さを $b-a$ で割った値が割り当てられる。両方のモーメントは直接積分すれば求まる。分散については、区間の中点が原点になるよう中心化し、$-c$ から c まで（ただし $c=\frac{b-a}{2}$）の区間とすると計算がすっきりする。

$$\mathbb{E}[X]=\int_a^b \frac{x}{b-a}\,dx=\frac{b^2-a^2}{2(b-a)}=\frac{a+b}{2},\qquad \mathrm{Var}(X)=\int_{-c}^{c}\frac{y^2}{2c}\,dy=\frac{c^2}{3}=\frac{(b-a)^2}{12}.$$

特に、$U(0,1)$ は平均 $\frac{1}{2}$、分散 $\frac{1}{12}$ をもち、$U(-\sqrt{3},\sqrt{3})$ は分散 $(2\sqrt{3})^2/12=1$ をもつ。この公式が離散一様分布の連続極限になっていることに注目しよう。間隔 h で等間隔に並ぶ等確率の n 個の点の分散は $h^2(n^2-1)/12$ であり、$nh=b-a$ を保ちながら点が $[a,b]$ を埋め尽くす極限で $(b-a)^2/12$ に収束する。逆変換を通じれば、先ほど指数分布で行ったように $U(0,1)$ を任意の連続分布へと変形できる。見た目には、一様分布は平坦な密度と不連続な端をもち、裾をまったくもたないため、ガウス分布とは似ても似つかない。だからこそ、以下で中心極限定理が一様分布の和を釣鐘曲線へと変えていく様子を観察するうえで理想的なテストケースとなる。

36.3.2.3 ガウス分布

隣接項の比を使えば任意の二項確率を高精度に計算できるが、それだけでは分布全体がどのような形をしているかは見えてこない。ピーク付近の形状を表す単純な閉じた式は存在するのだろうか。驚くべきことにそれは存在し、しかもそこにはコインの平均と分散しか現れない。ここから、確率論と機械学習において最も重要な分布へと導かれる。

平均付近での対数比。$q=1-p$ とおき、カウントを平均からのずれ x によって測ることにすると $k=np+x$ となる。これを二項分布の隣接項の比に代入すると次を得る。

$$\frac{P(X=k+1)}{P(X=k)}=\frac{(n-k)\,p}{(k+1)\,q}=\frac{(nq-x)\,p}{(np+x+1)\,q}=\frac{1-x/(nq)}{1+(x+1)/(np)}.$$

x が標準偏差 $\sqrt{npq}$ 程度の大きさであり n が大きいとき、$x/(nq)$ と $(x+1)/(np)$ はどちらも微小である。対数をとり、線形近似 $\log(1+\epsilon)\approx\epsilon$ を適用すると次のようになる。

$$\log\frac{P(X=k+1)}{P(X=k)}\approx-\frac{x}{nq}-\frac{x+1}{np}=-\frac{x}{npq}-\frac{1}{np},$$

ここで $\frac{1}{nq}+\frac{1}{np}=\frac{p+q}{npq}=\frac{1}{npq}$ を使って分数をまとめた。主要項を見ると、$\log P(X=k)$ から $\log P(X=k+1)$ への 1 ステップの変化量は、平均からの距離 x に対して傾き $-1/(npq)$ で線形に変化する。1 ステップの差分が線形に変化するのはどのような関数だろうか。放物線である。実際、放物線 $-x^2/(2npq)$ の変化量は

$$-\frac{(x+1)^2-x^2}{2npq}=-\frac{x}{npq}-\frac{1}{2npq}$$

と表される（x が 1 増えるとき）。これは微小な定数 $(p-q)/(2npq)$ の違いを除けば $\log P$ のステップ変化と一致する。この定数項は公正なコインでは完全に消え、一般の場合でもピーク周辺の $\sqrt{npq}$ ステップにわたって累積してなお $O(1/\sqrt{n})$ にしかならない。以上から、次の結論が得られる。

$$\log P(X = np + x) \approx C - \frac{x^2}{2npq}$$

これはある定数 C に対して、平均から数標準偏差以内にあるすべてのカウント $k = np + x$ で成り立ち、np が整数であるかどうかによらない（13 回のコイン投げでは $np = 6.5$ である）。両辺の指数をとれば、二項分布はピーク付近で釣鐘曲線 $e^{C} e^{-x^2/(2npq)}$ の形をとることがわかる。その中心は平均にあり、幅は標準偏差 $\sqrt{npq}$ で決まる。高次の剰余項を厳密に評価せず落としたため、これは厳密な証明ではなく導出のスケッチであるが、これを厳密化したものが中心極限定理の最古の形態である**ド・モアブル・ラプラスの定理**（de Moivre–Laplace theorem）である。

正規化定数。定数 C を決めるには、確率の総和が 1 にならなければならないという事実を使う。整数のカウントは間隔 1 で並んでいるため、すべての k にわたる釣鐘曲線の和は積分 $e^{C}\int_{-\infty}^{\infty} e^{-x^2/(2npq)}\,dx$ でよく近似される。したがって問題はすべて $I = \int_{-\infty}^{\infty} e^{-x^2/2}\,dx$ の評価に帰着する。一見すると、$e^{-x^2/2}$ は初等的な原始関数をもたないため、この積分は絶望的に思える。ここで使われる古典的な技巧は、代わりに I^2 を計算することである。平面上の二重積分とみなして極座標 $x = r\cos\theta$、$y = r\sin\theta$（面積要素 $dx\,dy = r\,dr\,d\theta$）に移れば、被積分関数は初等的に積分できるようになる。

$$\begin{aligned} I^2 &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(x^2+y^2)/2}\,dx\,dy = \int_0^{2\pi}\int_0^{\infty} e^{-r^2/2}\,r\,dr\,d\theta \\ &= 2\pi\left[-e^{-r^2/2}\right]_0^{\infty} = 2\pi. \end{aligned}$$

ゆえに $I = \sqrt{2\pi}$ である。極座標変換のヤコビ行列から生じる余分な因子 r のおかげで、$r\,e^{-r^2/2}$ は初等的な原始関数 $-e^{-r^2/2}$ をもつようになる（$e^{-x^2/2}$ 単体にはこれがない）。続いて $x = \sqrt{npq}\,u$ と変数変換（change of variables）すれば $\int_{-\infty}^{\infty} e^{-x^2/(2npq)}\,dx = \sqrt{2\pi npq}$ となるため、$e^{C} = 1/\sqrt{2\pi npq}$ と定まり、次式が得られる。

$$P(X = np + x) \approx \frac{1}{\sqrt{2\pi npq}}\, e^{-x^2/(2npq)}.$$

公正なコインを 13 回投げる例でこれを試してみよう。$npq = 3.25$ であり、$k = 9$ は $x = 2.5$ に対応するため、釣鐘曲線による近似値は $e^{-2.5^2/6.5}/\sqrt{2\pi\cdot 3.25} \approx 0.0846$ となり、これほど小さな n にもかかわらず厳密値 0.0873 に驚くほど近い。以下では、n が大きくなるにつれて釣鐘曲線による近似がどれほど速く精密になるかを測定する。

```
    for n in (13, 100, 1000):
        k = np.arange(n + 1)
        bell = np.exp(-(k - n / 2)**2 / (n / 2)) / np.sqrt(np.pi * n / 2)
        exact = binomial_pmf(n, 0.5, n)
        peak = n // 2
        print(f'n = {n:4d}: relative error at the peak {abs(bell[peak] / exact[peak] - 1):.2%}')
    print(f'n = 13, k = 9: bell {np.exp(-2.5**2 / 6.5) / np.sqrt(np.pi * 6.5):.4f},'
          f' exact {binomial_pmf(13, 0.5, 13)[9]:.4f}')
```

```
n =   13: relative error at the peak 1.66%
n =  100: relative error at the peak 0.25%
```

```
n = 1000: relative error at the peak 0.03%
n = 13, k = 9: bell 0.0846, exact 0.0873
```

$n = 13$、$k = 9$ での釣鐘曲線の近似値は約 3% 低いだけであり、ピークにおける相対誤差は 1.7%（$n = 13$）から 0.25%（$n = 100$）、0.03%（$n = 1000$）へと低下し、ほぼ $1/n$ のオーダーで縮小する。

標準ガウス分布。指数の放物線の形から、自然なスケール変換が示唆される。標準化されたカウント $Y^{(n)} = (X - np)/\sqrt{npq}$ は、平均 0、分散 1 をどの n に対してももち、そのとり得る値は間隔 $1/\sqrt{npq}$ で並んでいる。この単位で書き直すと、釣鐘近似は $P(X = k) \approx \phi(y)/\sqrt{npq}$（ただし $y = (k - np)/\sqrt{npq}$）となる。これは滑らかな密度 $\phi(y)$ にグリッド間隔を掛けたものであり、この関数は次で定義される。

$$\phi(y) = \frac{1}{\sqrt{2\pi}}\, e^{-y^2/2}.$$

これが標準化された二項分布の極限分布である**標準ガウス分布**（standard Gaussian）$\mathcal{N}(0,1)$ の密度である。$I = \sqrt{2\pi}$ より全積分は 1 となり、原点周りの対称性から平均は 0 である。分散については、$u = x$、$dv = x\, e^{-x^2/2}\, dx$ とおいて部分積分すると次が得られる。

$$\int_{-\infty}^{\infty} x^2\, \frac{e^{-x^2/2}}{\sqrt{2\pi}}\, dx = \frac{1}{\sqrt{2\pi}}\Big(\big[-x\, e^{-x^2/2}\big]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} e^{-x^2/2}\, dx \Big) = 0 + 1 = 1.$$

したがって標準ガウス分布の分散は 1 である。Y を μ だけ平行移動し、$\sigma > 0$ 倍にスケール変換すると $X = \mu + \sigma Y$ が作られる。36.2 節の変数変換の公式により、その密度は $\phi((x-\mu)/\sigma)/\sigma$ となり、これが一般の**ガウス**（または正規）分布を定義する。

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, \exp\!\Big(-\frac{(x-\mu)^2}{2\sigma^2} \Big), \qquad X \sim \mathcal{N}(\mu, \sigma^2). \tag{36.44}$$

$X = \mu + \sigma Y$ であるから、平均は μ、分散は σ^2 となる。つまり 2 つのパラメータがそのまま 1 次と 2 次のモーメントになっており、σ が釣鐘の幅を制御する。

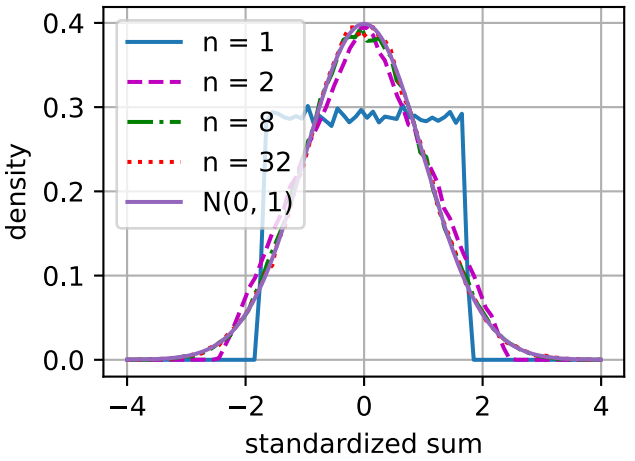
中心極限定理。ここまでの導出を振り返ると、最終的な分布の形にはコイン投げ固有の性質は平均と分散以外ほとんど入り込んでいなかった。実際、一般の定理に必要なのはそれだけである。$X_1, X_2, \ldots$ を平均 μ と有限の分散 $0 < \sigma^2 < \infty$ をもつ独立同分布（independent and identically distributed）の確率変数とし、$S_n = \sum_{i=1}^{n} X_i$ とおく。このとき、すべての実数 y に対して次が成り立つ。

$$\lim_{n \to \infty} P\Big(\frac{S_n - n\mu}{\sigma\sqrt{n}} \le y \Big) = \Phi(y) = \int_{-\infty}^{y} \phi(u)\, du. \tag{36.45}$$

言葉で述べれば、有限の分散をもつ**いかなる**分布であっても、その独立なコピーを多数足し合わせて標準化した和は標準ガウス分布に収束する。各項がベルヌーイ変数の場合がド・モアブル・ラプラスの定理であり、一般の場合の証明については (Wasserman 2013) を参照されたい。極限において生き残るのは個々の項の平均と分散だけであり、センサーノイズや測定ノイズのように多数の小さな独立した影響の積み重ねから生じる量がこれほど確実にガウス分布に従う理由はここにある。

一様分布の和。中心極限定理は、実際に動く様子を観察するとさらに印象的である。完全に平坦で鋭い端をもち裾がまったくない $U(0,1)$ から出発し、n 個の独立なサンプルの和をとり、平均 $n/2$ と標準偏差 $\sqrt{n/12}$ で標準化して、得られるヒストグラムを ϕ と比較してみよう。

```
ns, x = (1, 2, 8, 32), np.arange(-4, 4, 0.01)
curves = []
for n in ns:                                    # standardized sums of n uniforms
    s = (rng.random((100000, n)).sum(1) - n / 2) / np.sqrt(n / 12)
    hist, edges = np.histogram(s, bins=80, range=(-4, 4), density=True)
    curves.append(np.interp(x, (edges[:-1] + edges[1:]) / 2, hist))
curves.append(np.exp(-x**2 / 2) / np.sqrt(2 * np.pi))     # N(0, 1) density
d2l.plot(x, curves, xlabel='standardized sum', ylabel='density',
         legend=[f'n = {n}' for n in ns] + ['N(0, 1)'])
```



$n = 1$ では一様分布の平坦な台地が見え、$n = 2$ では鋭い三角形になる。$n = 8$ の時点ですでにヒストグラムはガウス分布の釣鐘曲線をよく捉えており、$n = 32$ になるとプロットの解像度では 2 つの曲線の見分けがつかなくなる。

ただし、分散が有限であるという仮定を外すことはできない。36.1 節で見たように、コーシー分布からのサンプルの平均はどれほど多くのサンプルを平均しても決して落ち着かないのに対し、n 個の標準ガウス変数の平均は $\mathcal{N}(0, 1/n)$ に従い 0 の周りに鋭く集中する。

ガウスノイズと二乗誤差。回帰ネットワークが予測値 $\hat{y}$ を出力し、観測ラベルをガウスノイズを伴う $y \sim \mathcal{N}(\hat{y}, \sigma^2)$ としてモデル化するとき、y に割り当てられる負の対数確率は $-\log p(y) = (y-\hat{y})^2/(2\sigma^2) + \frac{1}{2}\log(2\pi\sigma^2)$ となる。正のスケール因子と定数項の違いを除けば、これは二乗誤差損失である（2.1.3 節）。

36.3.2.4 ラプラス分布

二乗誤差は二次関数的に増大するため、残差（residual）10 には残差 1 の 100 倍もの重いペナルティが課され、少数の外れ値（outlier）によって当てはめが大きく引きずられてしまう。線形にしか増大しない損失である絶対誤差は、より裾の重いノイズ分布に対応する（2.1.3.3 節）。

これを構成するには、$P(S = 1) = P(S = -1) = \frac{1}{2}$ を満たす偏りのないランダムな符号 S と、それと独立な平均 $b > 0$（レート $1/b$）の指数確率変数 E をとり、$X = \mu + SE$ とおけばよ

い。$S=1$ のときは常に X は μ より大きく $X-\mu=E$ となるため、$x>\mu$ における密度は $\frac{1}{2}\cdot\frac{1}{b}e^{-(x-\mu)/b}$ となる。対称性から、μ より小さい側でも $x-\mu$ を $|x-\mu|$ に置き換えた同じ式が成り立つ。こうして**ラプラス**分布が得られる。

$$p(x)=\frac{1}{2b}\,e^{-|x-\mu|/b},\qquad X\sim\mathrm{Laplace}(\mu,b). \tag{36.46}$$

分布の左右それぞれの半分が、指数分布の半分の確率質量を担っている。密度は μ の周りで対称であり、その点で鋭く尖ったピークをとり、両側に向かって指数関数的に減衰する。

平均と分散。両方のモーメントは $X=\mu+SE$ から直ちに導かれる。独立性により $\mathbb{E}[SE]=\mathbb{E}[S]\,\mathbb{E}[E]=0$ となるため、$\mathbb{E}[X]=\mu$ である。また常に $S^2=1$ であるから、分散は単に指数分布の二次モーメント（second moment）$2/\lambda^2$（ただし $\lambda=1/b$）に一致する。

$$\mathrm{Var}(X)=\mathbb{E}\big[S^2E^2\big]=\mathbb{E}[E^2]=2b^2.$$

ラプラスノイズと絶対誤差。回帰ネットワークがラベルを $y\sim\mathrm{Laplace}(\hat{y},b)$ とモデル化するとき、負の対数確率は $-\log p(y)=|y-\hat{y}|/b+\log 2b$ となり、スケールと定数の違いを除いて絶対誤差に一致する。モデルの重みに対する事前分布（prior distribution）としてラプラス密度を用いれば、LASSO の ℓ_1 ペナルティ $\|\mathbf{w}\|_1$ が導かれる (Tibshirani 1996)。さらに、位置パラメータ μ をデータ $x_1,\dots,x_m$ に当てはめる際、ラプラス確率の最大化は $\sum_i|x_i-\mu|$ の最小化となり、これは標本中央値で達成される。たった 1 つの異常点が標本平均（sample mean）をいくらでも遠くへ引きずり得るのに対し、中央値は高々 1 順位しかずれない。

裾の比較。@fig-mdl-prob-continuous-pdfs に、同じ分散をもつようスケールを揃えた一様分布、ガウス分布、ラプラス分布の密度を比較する。分散を $\sigma^2=2b^2$ で揃えたとき、ラプラス確率変数が中心から標準偏差の 3 倍を超えて離れる確率は $P(|X-\mu|>3\sigma)=e^{-3\sigma/b}=e^{-3\sqrt{2}}\approx 0.014$ であり、ガウス分布の 0.0027 と比べて約 5 倍もの確率質量が裾に残る。

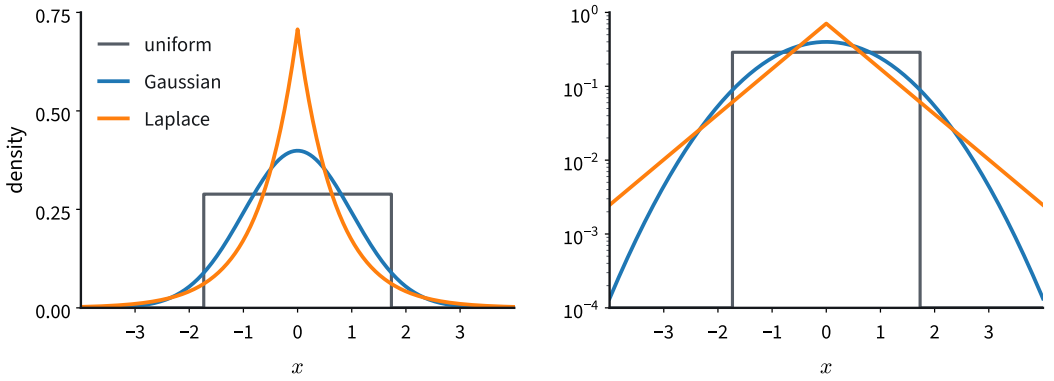


Figure 36.16: 分散 1 をもつ 3 つの密度（$[-\sqrt{3},\sqrt{3}]$ 上の一様分布、標準ガウス分布、$b=1/\sqrt{2}$ のラプラス分布）。(a) 線形スケールでの密度。(b) 対数スケールでの同じ密度。ガウス分布は放物線、ラプラス分布は 2 本の直線、一様分布は $\pm\sqrt{3}$ で途切れる平坦な台地となる。標準偏差の 3 倍より外側では、ラプラス分布はガウス分布の約 5 倍の裾確率をもつ。

ラプラス分布をその構成法（ランダムな符号と指数乱数の積）から直接サンプリングし、標準偏差の 3 倍を超えるサンプルがどれほどの頻度で現れるかを数えて、この裾の比較を確かめてみよう。

```
b, m = 1.0, 200000
lap = rng.choice([-1.0, 1.0], m) * rng.exponential(b, m)   # S * E
sigma = math.sqrt(2) * b                                    # matched sd
gau = rng.normal(0.0, sigma, m)
print(f'sample variances: Laplace {lap.var(ddof=1):.2f}, '
      f'Gaussian {gau.var(ddof=1):.2f}   (2b^2 = 2)')
print(f'P(|X| > 3 sd): Laplace {np.mean(np.abs(lap) > 3 * sigma):.4f}, '
      f'Gaussian {np.mean(np.abs(gau) > 3 * sigma):.4f}')
```

```
sample variances: Laplace 1.98, Gaussian 2.00   (2b^2 = 2)
P(|X| > 3 sd): Laplace 0.0140, Gaussian 0.0027
```

どちらの標本分散（sample variance）もほぼ 2 になり、標準偏差の 3 倍を超える裾の割合はそれぞれ 0.014 と 0.003 の近くに収まっている。

36.3.3 多変量ガウス分布

深層学習において、ランダムな量がスカラーであることはまれである。画像のノイズ、生成モデルにおける潜在表現、入力のバッチ全体にわたる予測は、いずれも各座標が相互作用するベクトルである。多変量ガウス分布（multivariate Gaussian）は、相関をもつベクトル上の分布でありながら、必要となるほぼすべての操作のもとで解析的に扱いやすい唯一の分布である。通し例として、それぞれの分散が 2、共分散が 1 である 2 座標の組を考えよう。

$$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

36.3.3.1 等高線と固有ベクトル

$\mathbb{R}^d$ 上の**多変量ガウス分布** $\mathcal{N}(\mu, \Sigma)$ は、平均ベクトル μ と対称な正定値（positive definite）共分散行列 Σ によってパラメータ化される。その密度は次の式で与えられる。

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\!\Big(-\tfrac{1}{2}(\mathbf{x}-\mu)^{\top}\Sigma^{-1}(\mathbf{x}-\mu)\Big), \tag{36.47}$$

ここで $|\Sigma|$ は Σ の行列式（determinant）を表す。$d = 1$ のとき、これは Equation 36.44 に帰着する。$\mathbf{x}$ は指数部の二次形式（quadratic form）のみを通じて密度に現れるため、その二次形式が一定となる等位集合上で密度は一定になる。これらの等高線はどのような幾何学的形状をしているのだろうか。

通し例の手計算。 $\mu = \mathbf{0}$ とし、座標軸を 45° 回転して和と差の座標 $u = (x_1 + x_2)/\sqrt{2}$ および $w = (x_1 - x_2)/\sqrt{2}$ に移ろう。これらの分散と共分散は Σ の成分から直接求まる。

$$\operatorname{Var}(u) = \frac{2 + 2 + 2 \cdot 1}{2} = 3, \qquad \operatorname{Var}(w) = \frac{2 + 2 - 2 \cdot 1}{2} = 1, \qquad \operatorname{Cov}(u, w) = \frac{2 - 2}{2} = 0.$$

この回転後の座標系では、共分散行列は対角行列になる。新しい軸は単に Σ の固有ベクトル（eigenvector）$\mathbf{v}_1 = (1, 1)/\sqrt{2}$ と $\mathbf{v}_2 = (1, -1)/\sqrt{2}$ であり、対応する固有値（eigenvalue）は $\gamma_1 = 3$ と $\gamma_2 = 1$ である（実際、$\Sigma \mathbf{v}_1 = (3, 3)/\sqrt{2} = 3\mathbf{v}_1$ が成り立つ）。

一般の場合。スペクトル定理により、任意の対称行列（symmetric matrix）は固有ベクトルからなる正規直交基底（orthonormal basis）をもち、$\Sigma = \mathbf{V}\,\mathrm{diag}(\gamma_1, \dots, \gamma_d)\mathbf{V}^\top$（ただし $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$）と表せる。また正定値性から、すべての固有値は $\gamma_i > 0$ を満たす。固有座標 $\mathbf{u} = \mathbf{V}^\top(\mathbf{x} - \mu)$ に移ると、二次形式は単純な重み付き二乗和に変わる。

$$(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu) = \mathbf{u}^\top\,\mathrm{diag}(\gamma_1^{-1}, \dots, \gamma_d^{-1})\,\mathbf{u} = \sum_{i=1}^{d} \frac{u_i^2}{\gamma_i}.$$

この和が r^2 に等しい等位集合は、固有ベクトル $\mathbf{v}_i$ に沿った楕円体であり、その半軸長は $r\sqrt{\gamma_i}$ となる。したがって、任意の多変量ガウス分布の等高線は、Σ の固有ベクトルを主軸とし、標準偏差 $\sqrt{\gamma_i}$ に比例して引き伸ばされた楕円体になる。通し例では、45° 傾いた軸比 $\sqrt{3} : 1$ の楕円になる（Figure 36.17 左）。また $|\Sigma| = \prod_i \gamma_i$ であるから、密度は $\mathbf{u}$ において d 個の 1 次元ガウス分布の積へと分解する。すなわち固有ベクトルの方向に沿って、各座標は互いに独立で分散 γ_i をもつ。特別な場合として $\Sigma = \sigma^2\mathbf{I}$ のときは、等高線は球面となり、元の座標が最初から互いに独立である。

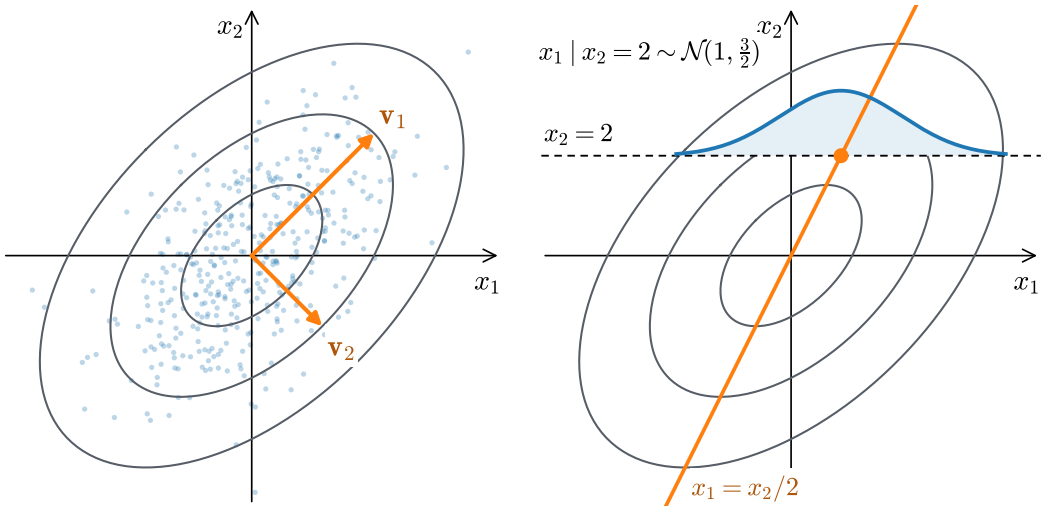


Figure 36.17: 行が $(2, 1)$ と $(1, 2)$ である共分散行列 Σ をもつ通し例の 2 変量ガウス分布。(a) 400 個のサンプルと、指数部の二次形式が 1、4、9 に等しい 3 本の密度等高線。楕円の軸は Σ の固有ベクトルに沿い、半軸長は $\sqrt{\gamma_1} = \sqrt{3}$ と $\sqrt{\gamma_2} = 1$ に比例する。矢印 $\mathbf{v}_1$ と $\mathbf{v}_2$ は中央の等高線まで達している。(b) $x_2 = 2$ を観測すると破線に沿って密度が切り出され、x_1 の条件付き密度 $\mathcal{N}(1, \frac{3}{2})$ が得られる。すべてのスライスにわたり、条件付き平均は直線 $x_1 = x_2/2$ 上に乗り、条件付き分散は $3/2$ で一定に保たれる。

1 つ注意点を早めに挙げておこう。高次元では、平均付近に質量が集中した釣鐘という低次元の直感は誤解を招きやすい。$\mathbb{R}^d$ における標準ガウスベクトルのノルムはほぼ常に $\sqrt{d}$ の近くにあり、独立に引いた 2 つのサンプルはほぼ直交（orthogonal）する。これら 2 つの事実は 36.9 節で証明する。

36.3.3.2 線形変換とサンプリング

実際には、既存のガウスベクトルから座標を混ぜ合わせたり部分空間（subspace）へ射影（projection）したりして、新たなガウスベクトルを日常的に構成する。これらはいずれも線形変

換（linear map）であり、ガウス分布が非常に扱いやすい大きな理由の 1 つは、線形変換がガウス性を保つことにある。まずはサンプリングに必要な可逆正方行列の場合についてこれを証明しよう。長方形行列による射影については、ガウスベクトルの任意の部分ベクトルがそれ自体ガウス分布に従うことを次項で示してから導く。

命題（可逆な線形変換）。 $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ を $\mathbb{R}^d$ 上の確率ベクトルとし、$\mathbf{A}$ を可逆な $d \times d$ 行列、$\mathbf{b} \in \mathbb{R}^d$ とする。このとき次が成り立つ。

$$\mathbf{A}\mathbf{x} + \mathbf{b} \sim \mathcal{N}\big(\mathbf{A}\mu + \mathbf{b},\ \mathbf{A}\Sigma\mathbf{A}^{\top}\big).$$

証明。 期待値の線形性により、上記の平均と共分散は有限の二次モーメントをもつ任意の分布で成り立つ。実際、$\mathbb{E}\big[(\mathbf{A}\mathbf{x} - \mathbf{A}\mu)(\mathbf{A}\mathbf{x} - \mathbf{A}\mu)^{\top}\big]$ は $\mathbf{A}\,\mathbb{E}\big[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^{\top}\big]\,\mathbf{A}^{\top} = \mathbf{A}\Sigma\mathbf{A}^{\top}$ に等しい。あとは変換後の密度がガウス密度であることだけを確かめればよい。36.2 節の変数変換の公式により、$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ の密度は $p_{\mathbf{y}}(\mathbf{y}) = p_{\mathbf{x}}(\mathbf{A}^{-1}(\mathbf{y}-\mathbf{b}))/|\det \mathbf{A}|$ となる。$\mathbf{c} = \mathbf{y} - \mathbf{A}\mu - \mathbf{b}$ とおけば、指数部は $-\tfrac{1}{2}\mathbf{c}^{\top}\mathbf{A}^{-\top}\Sigma^{-1}\mathbf{A}^{-1}\mathbf{c} = -\tfrac{1}{2}\mathbf{c}^{\top}(\mathbf{A}\Sigma\mathbf{A}^{\top})^{-1}\mathbf{c}$ と書き直され、正規化因子も $|\det \mathbf{A}|\,|\Sigma|^{1/2} = |\mathbf{A}\Sigma\mathbf{A}^{\top}|^{1/2}$ を満たす。したがって $p_{\mathbf{y}}$ は主張どおりの平均と共分散をもつガウス密度である。$\blacksquare$

ここでは可逆性が不可欠である。もし $\mathbf{A} = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$ であれば、出力の 2 つの座標は常に同じ値になり、$\mathbf{A}\Sigma\mathbf{A}^{\top} = \left(\begin{smallmatrix} 6 & 6 \\ 6 & 6 \end{smallmatrix}\right)$ は特異行列になる。結果として得られる確率ベクトル（random vector）は完全に直線 $y_1 = y_2$ 上に集中し、平面上の 2 次元密度をもたない。

行列因子によるサンプリング。 手元の乱数生成器が独立な標準正規乱数しか生成できないとき、どのようにして $\mathcal{N}(\mu, \Sigma)$ からサンプリングすればよいだろうか。独立に引いた d 個の乱数を並べて $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ を作れば、その同時密度は ϕ の d 個の積になる。$\mathbf{L}\mathbf{L}^{\top} = \Sigma$ を満たす任意の正方行列 $\mathbf{L}$ に対し、$\mathbf{L}$ は必ず可逆行列（invertible matrix）になり（Σ が正定値行列であるため）、先ほどの命題から $\mathbf{x} = \mu + \mathbf{L}\mathbf{z}$ は平均 μ、共分散 $\mathbf{L}\,\mathbf{I}\,\mathbf{L}^{\top} = \Sigma$ のガウス分布に従うことが保証される。

要するに、行列因子を掛けることで独立なノイズが相関をもつノイズへと変換される。乱数 $\mathbf{z}$ を固定したとき $\mathbf{x}$ は μ と $\mathbf{L}$ の滑らかで微分可能な関数になるため、この同じ構成法は変分オートエンコーダ（variational autoencoder、VAE）が確率的な層を通して逆伝播を行うための**再パラメータ化トリック**（reparameterization trick）そのものでもある（34.7.3 節）。最も高速に計算できる因子は**コレスキー因子**（Cholesky factor）である。任意の対称正定値行列は、対角成分が正の下三角行列 $\mathbf{L}$ を用いて $\Sigma = \mathbf{L}\mathbf{L}^{\top}$ と一意に分解できる。通し例では、$\mathbf{L}\mathbf{L}^{\top}$ の各成分を Σ に順番に一致させることで、まず $L_{11} = \sqrt{2}$、次に $L_{21} = \Sigma_{21}/L_{11} = 1/\sqrt{2}$、最後に $L_{22} = \sqrt{\Sigma_{22} - L_{21}^2} = \sqrt{3/2}$ が求まる（実際、$L_{21}^2 + L_{22}^2 = \tfrac{1}{2} + \tfrac{3}{2} = 2$ である）。

固有因子 $\mathbf{V}\,\mathrm{diag}(\sqrt{\gamma_1}, \dots, \sqrt{\gamma_d})$ など、他のどの平方根因子を使っても同様にうまくいく。演習問題 5 で示すように、同じ Σ に対する任意の 2 つの因子は直交行列（orthogonal matrix）を掛ける違いしかもたない。実際のライブラリのサンプラは実装ごとに異なり、多くはコレスキー因子を用いる一方、NumPy の `multivariate_normal` はデフォルトで特異値分解（singular value decomposition）から得られる因子を用いている。

ライブラリのサンプラを呼び出す代わりに、仕組みが動く様子を確認できるようコレスキー分解の手順を直接実装し、標本共分散とその固有値の両方を検証してみよう。

```
Sigma = np.array([[2.0, 1.0], [1.0, 2.0]])
L = np.linalg.cholesky(Sigma)                  # Sigma = L L^T
x = rng.standard_normal((400000, 2)) @ L.T     # each row is x = L z
C = np.cov(x.T)
```

```
    print('L =', L.round(3).tolist())
    print('sample covariance =', C.round(2).tolist())
    print('its eigenvalues   =', np.linalg.eigvalsh(C).round(2))
```

```
L = [[1.414, 0.0], [0.707, 1.225]]
sample covariance = [[1.99, 0.99], [0.99, 1.99]]
its eigenvalues   = [1.   2.99]
```

コレスキー因子は手計算の結果と一致し、経験共分散は Σ と約 0.01 以内で一致し、その固有値も 1 と 3 の近くに収まっている。

36.3.3.3 条件付け

ガウスベクトルの一部の座標を観測し、残りの座標を推論したい場面は頻繁にある。通し例において $x_2 = 2$ を観測したとき、x_1 はどのあたりにあると期待すべきだろうか。ベクトルを 2 つのブロック $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ に分割し、それに合わせて平均と共分散もブロック分割しよう。

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}.$$

ガウス分布の条件付けの背後にある仕掛けは幾何学的で明快である。$\mathbf{x}_1$ から、$\mathbf{x}_2$ に基づく $\mathbf{x}_1$ の線形予測を引けばよい。すると残るのは、$\mathbf{x}_2$ についての情報を一切もたない残差である。

命題（残差と周辺分布）。 $\mathbf{B} = \Sigma_{12}\Sigma_{22}^{-1}$ とおく。このとき残差 $\mathbf{r} = \mathbf{x}_1 - \mathbf{B}\mathbf{x}_2$ は $\mathbf{x}_2$ と独立であり、第 2 ブロックの周辺分布（marginal distribution）は $\mathbf{x}_2 \sim \mathcal{N}(\mu_2, \Sigma_{22})$ となる。

証明。 Σ_{22} は正定値行列の主対角ブロックであるため正定値であり、したがって可逆である。構成法より、残差は $\mathbf{x}_2$ と無相関になる。実際、$\operatorname{Cov}(\mathbf{r}, \mathbf{x}_2) = \Sigma_{12} - \mathbf{B}\Sigma_{22} = \mathbf{0}$ である。線形変換 $\mathbf{x} \mapsto (\mathbf{r}, \mathbf{x}_2)$ は正方ブロック行列 $\left(\begin{smallmatrix} \mathbf{I} & -\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{smallmatrix}\right)$ に対応し、その行列式は 1 であるため、可逆な線形変換に関する命題より $(\mathbf{r}, \mathbf{x}_2)$ は同時ガウス分布に従う。共分散の非対角ブロックが消えているため（$\operatorname{Cov}(\mathbf{r}, \mathbf{x}_2) = \mathbf{0}$）、その共分散行列はブロック対角である。結果として共分散の逆行列もブロック対角となり、指数部の二次形式は $\mathbf{r}$ の関数と $\mathbf{x}_2$ の関数の和へと綺麗に分離し、行列式も 2 つのブロックの行列式の積に分解する。したがって同時密度は $\mathbf{r}$ のガウス密度と $\mathbf{x}_2$ のガウス密度の積に因数分解される。ゆえに 2 つのブロックは独立であり、$\mathbf{x}_2$ は平均 μ_2、共分散 Σ_{22} をもつ。$\blacksquare$

証明全体の鍵は、残差が $\mathbf{x}_2$ と無相関になるように $\mathbf{B}$ を選んだ点にある。副産物として、後半の主張からガウスベクトルの任意の周辺部分ベクトルがそれ自体ガウス分布に従うことも証明された（第 1 ブロックについては、可逆な線形変換であるブロックの順序交換を行えばよい）。これにより、先ほど後回しにした長方形行列による線形変換も直ちに解決する。

系（横長行列）。 $\mathbf{A}$ を行フルランク $m \le d$ の $m \times d$ 行列とし、$\mathbf{b} \in \mathbb{R}^m$ とすると、$\mathbf{A}\mathbf{x} + \mathbf{b} \sim \mathcal{N}(\mathbf{A}\mu + \mathbf{b}, \mathbf{A}\Sigma\mathbf{A}^{\top})$ が成り立つ。実際、$\mathbf{A}$ に $d - m$ 個の行を追加して可逆な $d \times d$ 行列を作れば、それによる $\mathbf{x}$ の像はガウス分布に従い、$\mathbf{A}\mathbf{x} + \mathbf{b}$ は単にその像の第 1 ブロックとして得られる。

ここまで来れば、$\mathbf{x}_2$ による条件付けは造作もない。$\mathbf{x}_1 = \mathbf{r} + \mathbf{B}\mathbf{x}_2$ であり $\mathbf{r}$ は $\mathbf{x}_2$ と独立であるため、$\mathbf{x}_2$ を固定することはガウス変数 $\mathbf{r}$ を既知のベクトル $\mathbf{B}\mathbf{x}_2$ だけ平行移動するにすぎない。残差の平均は $\mu_1 - \mathbf{B}\mu_2$、共分散は $\Sigma_{11} - \mathbf{B}\Sigma_{21} - \Sigma_{12}\mathbf{B}^{\top} + \mathbf{B}\Sigma_{22}\mathbf{B}^{\top}$ すなわち $\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ であるから、条件付き分布 $\mathbf{x}_1 \mid \mathbf{x}_2$ は次のパラメータをもつガウス分布となる。

$$\begin{aligned} \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2), \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \end{aligned} \tag{36.48}$$

Equation 36.48 には際立った特徴が 2 つある。条件付き平均は観測ブロック $\mathbf{x}_2$ のアフィン関数であり、条件付き共分散は観測値に一切依存しない。通し例では $\mathbf{B} = \frac{1}{2}$ なので、$x_2 = 2$ を観測すると条件付き平均は $\frac{1}{2} \cdot 2 = 1$、条件付き分散（conditional variance）は $2 - 1 \cdot \frac{1}{2} \cdot 1 = \frac{3}{2}$ となる。したがって $x_1 \mid x_2 = 2 \sim \mathcal{N}(1, \frac{3}{2})$ となり、$x_2 = 2$ における水平断面（Figure 36.17 右）と一致する。

$\mathbf{B}$ が $\mathbf{x}_2$ から $\mathbf{x}_1$ を予測する線形回帰（linear regression）の行列、すなわちそれを差し引くと相関（correlation）が 0 になるような $\mathbf{x}_2$ の係数行列になっていることに注目しよう。多変量ガウス分布においては、条件付き期待値（conditional expectation）は共分散行列から直接読み取れる線形回帰**そのもの**である。一方、線形代数（linear algebra）で Σ_{22} の**シューア補行列**（Schur complement）と呼ばれる条件付き共分散（33.2.3.5 節）は、差し引かれる項 $\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ が半正定値（positive semidefinite）であるため、決して Σ_{11} を超えない。x_2 を観測することで、x_1 に関する不確実性は分散 2 から $\frac{3}{2}$ へと縮小する。$\mathbf{x}_2$ が未知関数の観測値を表し、$\mathbf{x}_1$ が新しいテスト入力における関数値を表すとき、この公式はガウス過程回帰（Gaussian process regression）となる（第 40 章（英語版））。

注意深い読者は、共分散が 0 であっても一般には独立性（independence）を意味しないのではないかと疑問に思うかもしれない。そのとおりであり、上の証明はより強い性質、すなわち $(\mathbf{r}, \mathbf{x}_2)$ が**同時**ガウス分布に従うことに依拠していた。同時ガウス性（joint Gaussianity）がなければ、この結論は成り立たない。たとえば $Z \sim \mathcal{N}(0,1)$ とし、S をそれと独立で偏りのないランダムな符号としよう。すると SZ も周辺分布としては $\mathcal{N}(0,1)$ に従い、$\mathrm{Cov}(SZ, Z) = \mathbb{E}[S]\,\mathbb{E}[Z^2] = 0$ を満たすが、$|SZ| = |Z|$ であるため 2 つの変数は強く従属している。

行列の公式を一切使わずに、コレスキー分解で作ったサンプル上でガウス分布の条件付けを直接確かめることもできる。x_2 が幅 0.05 以内で 2 の近くに入るサンプルだけを残し、その x_1 の値の分布を調べるとともに、残差 $r = x_1 - \frac{1}{2}x_2$ を確認してみよう。

```
r = x[:, 0] - 0.5 * x[:, 1]                    # residual x1 - B x2, B = 1/2
print(f'Cov(r, x2) = {np.cov(r, x[:, 1])[0, 1]:.3f},  Var(r) = {r.var(ddof=1):.3f}')
near = x[np.abs(x[:, 1] - 2) < 0.05, 0]        # x1 of samples with x2 near 2
print(f'{near.size} samples with x2 near 2: mean of x1 {near.mean():.2f}, '
      f'variance {near.var(ddof=1):.2f}')
```

```
Cov(r, x2) = -0.003,  Var(r) = 1.499
4091 samples with x2 near 2: mean of x1 0.97, variance 1.52
```

予測どおり、残差の共分散はほぼ 0（x_2 との間）であり、分散はほぼ $\frac{3}{2}$ となる。約 4,000 個のサンプルがこのスライスに入り、それらの x_1 の値は平均が 1 付近、分散が $\frac{3}{2}$ 付近となって $\mathcal{N}(1, \frac{3}{2})$ と一致している。

自然な問いが 2 つ残されている。第 1 に、ここまでに出会ったほぼすべての分布は単一の代数的なひな型を共有しており、36.4 節ではこれを利用して損失、勾配、共役更新を統一的に扱う。第 2 に、冒頭の $715/8192$ という計算はコインが公正であることを仮定していたが、13 回のコイン投げから未知の表の確率（偏り）を推論し、その不確実性を定量化することは 36.5 節の主題となる。

36.3.4 まとめ

以下の表に、本節で導入した分布を登場順にまとめる。カテゴリカル分布と多項分布の行では、交差エントロピー損失（cross-entropy loss）がクラスラベル（class label）を符号化する方式に合わせて、各抽出をワンホットベクトルとして表している。

分布	確率質量関数または確率密度関数	平均	分散	モデルの仮定
Bernoulli(p)	$p^x(1-p)^{1-x}$	p	$p(1-p)$	1 回の yes/no の結果
Binomial(n,p)	$\binom{n}{k}p^k(1-p)^{n-k}$	np	$np(1-p)$	固定回数の n 回の独立試行、共通の p
Poisson(λ)	$\lambda^k e^{-\lambda}/k!$	λ	λ	多数の希少な独立機会、平均と分散が一致
Categorical$(\mathbf{p})$	p_k	$\mathbf{p}$	$\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$	K 通りの結果のうちの 1 つ
Multinomial$(n,\mathbf{p})$	Equation 36.40	$n\mathbf{p}$	$n(\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top)$	固定回数の n 回の独立抽出
Unif$\{1,\dots,n\}$	$1/n$	$(n+1)/2$	$(n^2-1)/12$	等確率の値
Exp(λ)	$\lambda e^{-\lambda t}$、$t \geq 0$	$1/\lambda$	$1/\lambda^2$	一定の事象発生レート（経年劣化なし）
Gamma(α,β)	$\beta^\alpha t^{\alpha-1}e^{-\beta t}/\Gamma(\alpha)$	α/β	α/β^2	α 番目の事象の発生時刻（α が整数の場合）
$U(a,b)$	$1/(b-a)$（$[a,b]$ 上）	$(a+b)/2$	$(b-a)^2/12$	同じ長さの区間がすべて等確率
$\mathcal{N}(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/(2\sigma^2)}$	μ	σ^2	多数の小さな独立した影響の和、有限の分散
Laplace(μ,b)	$\frac{1}{2b}e^{-\lvert x-\mu\rvert/b}$	μ	$2b^2$	対称で指数関数的に減衰する裾
$\mathcal{N}(\mu,\Sigma)$	Equation 36.47	μ	Σ	同時ガウス分布に従う座標群

本章の他の節でも重要な分布がいくつか登場する。コーシー分布は 36.1 節、対数正規分布（log-normal distribution）は 36.2 節、ベータ分布とディリクレ分布は 36.4 節で扱い、さらに負の二項分布（negative binomial distribution）、幾何分布（geometric distribution）、スチューデントの t 分布が演習問題 1、2、4 に登場する。

少数のまとまった分布族によって大半の機械学習システムのラベルとノイズがモデル化され、各分布の負の対数確率が標準的な訓練損失（training loss）を与える。これらはいずれも、和、極限、待ち時間、ランダムな符号、あるいは相関をもつベクトルを通じて、たった 1 回のコイン投げから派生する。そうした構成法のおかげで、各分布の平均、分散、サンプリングアルゴリズムがわずか数行で導かれる。中心極限定理は、有限の分散をもつ多数の小さな独立した影響が足し合わされるときに必ずガウス分布が現れる理由を説明してくれる。最後に、多変量ガウス分布は線形変換と条件付けのもとで閉じており、あるブロックを別のブロックで条件付ける操作は直接線形回帰へと帰着する。

36.3.5 演習問題

1. **ガンマ・ポアソン混合分布**。ポアソン分布のレート自体を確率変数とし、密度 Equation 36.42 をもつ $\lambda \sim \mathrm{Gamma}(\alpha,\beta)$ に従うものとし、$X \mid \lambda \sim \mathrm{Poisson}(\lambda)$ とする。
   1. 全期待値の法則 $\mathbb{E}[X] = \mathbb{E}\big[\mathbb{E}[X \mid \lambda]\big]$ と全分散の法則
   $$\mathrm{Var}(X) = \mathbb{E}\big[\mathrm{Var}(X \mid \lambda)\big] + \mathrm{Var}\big(\mathbb{E}[X \mid \lambda]\big)$$
   を用いて、$\mathbb{E}[X]$ と $\mathrm{Var}(X)$ を求めよ。分散を平均 $m = \alpha/\beta$ と形状 α を用いて表せ。α と β のどのような極限でポアソン分布が再び現れるか。
   2. X が**負の二項分布**の確率質量関数

$$P(X=k)=\frac{\Gamma(k+\alpha)}{k!\,\Gamma(\alpha)}\Big(\frac{\beta}{1+\beta}\Big)^{\alpha}\Big(\frac{1}{1+\beta}\Big)^{k},\qquad k=0,1,2,\dots, \tag{36.49}$$

をもつことを、ガンマ分布の正規化定数を用いて λ を積分消去することで示せ。

3. あるカウントデータのデータセットが標本平均 2、標本分散 6 をもつとする。モデルが両方に一致するように α と β を選べ。このとき、どのような名前付きの分布が得られるか。

2. **無記憶性の逆**。●指数分布は無記憶である。この問題ではその逆を証明する。
   1. $X>0$ を連続確率変数とし、その生存関数 $G(t)=P(X>t)$ がすべての $s,t\ge 0$ に対して $G(s+t)=G(s)\,G(t)$ を満たすとする。ある $\lambda>0$ に対して $G(t)=e^{-\lambda t}$ となることを示せ。まず $G(1)$ から出発し、有理数 t へ式を拡張したうえで、G が非増加であることを用いよ。なぜ $G(1)$ は 0 にも 1 にもなり得ないのか。
   2. X が $\{1,2,\dots\}$ に値をとり、すべての整数 $s,t\ge 0$ に対して $P(X>s+t\mid X>s)=P(X>t)$ を満たすとする。ある $p\in(0,1]$ に対して $P(X>t)=(1-p)^t$ となり、X が幾何分布に従うことを示せ。

3. [extended] **単語出現数のバースト性**。●●Church and Gale (1995) は、テキスト中の単語出現数が文書間でポアソンモデルの許容範囲を超えて変動すること、そして負の二項分布のようなポアソン混合分布のほうがよく当てはまることを発見した。この知見を『The Time Machine』（8.2 節）上で小規模に再現せよ。`d2l.download(d2l.DATA_URL + 'timemachine.txt')` でテキストをダウンロードし、`d2l.TimeMachine._preprocess` と同様に英字のみを残して単語に分割し、その単語系列を文書の役割を果たす長さ $L=500$ 語の連続したチャンクに区切れ。
   1.「the」「and」「of」「time」「traveller」「weena」「morlocks」の各単語について、すべてのチャンクでの出現回数を数え、カウントの標本平均に対する標本分散の比を計算せよ。ポアソンモデルはどのような比を予測するか。そこから最も大きく外れるのはどの単語か。
   2. 各単語のカウントに対してポアソンモデルと負の二項分布モデル（問題 1b）を最尤法によって当てはめ、最大化された対数尤度を比較せよ。負の二項分布は平均 m と形状 α でパラメータ化せよ。最尤推定量 m は標本平均に一致するため、たとえば対数グリッド上の 1 次元探索が必要なのは α のみである。
   3. 各単語について、その単語を含まないチャンクの経験比率および経験的な**適応性** $P(X\ge 2\mid X\ge 1)$ を、当てはめた両モデルの予測値と比較せよ。適応性は何を測っており、なぜポアソンモデルは「traveller」のような名前に対してこれを過小評価するのか。
   4. チャンク長 $L\in\{100,250,500,2000\}$ に対して (a) を繰り返せ。チャンク内の単語のレートが、固定された形状をもつガンマ分布から抽出された 1 語あたりレートの L 倍であると仮定する。問題 1a を用いて分散平均比が L に伴いどのように増大するかを予測し、実測値と比較せよ。

4. **ガンマ尺度混合分布としてのスチューデントの t 分布**。●精度（分散の逆数）自体がランダムであるガウス分布は、いかなるガウス分布よりも重い裾をもつ。その結果として得られるのがスチューデントの t 分布であり、36.5 節では小標本の平均に対してこれを用いる。密度 Equation 36.42 をもつ $\tau\sim\text{Gamma}(\nu/2,\nu/2)$（ある $\nu>0$ に対し形状とレートがともに $\nu/2$）とし、$X\mid\tau\sim\mathcal{N}(\mu,1/\tau)$ とする。
   1. τ を積分消去することで、X が次の密度

$$p(x)=\frac{\Gamma\big(\frac{\nu+1}{2}\big)}{\sqrt{\pi\nu}\,\Gamma\big(\frac{\nu}{2}\big)}\Big(1+\frac{(x-\mu)^2}{\nu}\Big)^{-(\nu+1)/2}, \tag{36.50}$$

   すなわち自由度 ν の**スチューデントの t 分布**をもつことを示せ。**ヒント：**$a,c>0$ に

対して $\int_0^\infty \tau^{a-1} e^{-c\tau}\,d\tau = \Gamma(a)/c^a$ が成り立つ。

2. $p(x)$ が $|x|^{-(\nu+1)}$ のオーダーで減衰し、したがって $\mathbb{E}|X-\mu|^k$ が有限になるのは $k < \nu$ のときに限ることを示せ。36.1 節に登場した分布のうち、$\nu = 1$ の場合に対応するのはどれか。
3. $\mathrm{Var}(X) = \nu/(\nu-2)$（$\nu > 2$）となることを、全分散の法則

$$\mathrm{Var}(X) = \mathbb{E}\big[\mathrm{Var}(X \mid \tau)\big] + \mathrm{Var}\big(\mathbb{E}[X \mid \tau]\big).$$

を用いて示せ。
4. [code] 混合分布の手順（まず τ を引き、次に X を引く）によって $\nu = 3$、$\mu = 0$ のサンプルを 10^5 個生成し、$|X| > 3$ となる割合をガウス分布の 0.0027 と比較せよ。続いて、20 個ずつのサンプルの標本平均を 10^4 個生成し、その標準偏差を $\sqrt{3/20}$ と比較せよ。

5. **1 つの共分散行列に対する 2 つの因子。**●$\Sigma = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ とする。
   1. コレスキー因子 $\mathbf{L}$ を手計算で求め、さらに本文で求めた固有ベクトルと固有値から固有因子 $\mathbf{M} = \mathbf{V}\,\mathrm{diag}(\sqrt{\gamma_1}, \sqrt{\gamma_2})$ を計算せよ。$\mathbf{L}\mathbf{L}^\top = \mathbf{M}\mathbf{M}^\top = \Sigma$ を確かめよ。
   2. $\mathbf{M}_1\mathbf{M}_1^\top = \mathbf{M}_2\mathbf{M}_2^\top$ を満たす任意の 2 つの可逆な因子は、直交行列 $\mathbf{Q}$ を用いて $\mathbf{M}_2 = \mathbf{M}_1\mathbf{Q}$ を満たすことを示せ（**ヒント：**$\mathbf{Q} = \mathbf{M}_1^{-1}\mathbf{M}_2$ とおく）。$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ のとき、$\mathbf{Q}\mathbf{z}$ の分布はどうなるか。ここから $\mu + \mathbf{M}_1\mathbf{z}$ と $\mu + \mathbf{M}_2\mathbf{z}$ が同じ分布をもつことを結論せよ。
   3. **白色化。**$\mathbf{z} = \mathbf{L}^{-1}(\mathbf{x} - \mu)$ が共分散 $\mathbf{I}$ をもつこと（$\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ のとき）を示せ。また、$\mathbf{z}$ を 1 つの $\mathbf{x}$ に対して求める計算が約 d^2 回の演算で済む（$\mathbf{L}$ が既知の場合）のに対し、約 $d^3/3$ 回の演算が $\mathbf{L}$ の計算には必要となる理由を説明せよ。
   4. [code] 各因子を用いて 10,000 点をサンプリングして両方の経験共分散を表示し、三角行列ソルバを用いてコレスキーサンプルを白色化したうえで、結果の共分散を表示せよ。

6. **重ね合わせ、間引き、到着順序。**●レート λ_1 と λ_2 のポアソン過程である 2 つの独立な発生源からサーバーにリクエストが届くとする。
   1. 合流したストリームがレート $\lambda_1 + \lambda_2$ のポアソン過程になることを示せ。区間細分化の描像を用いるか、二項定理によって独立な $\mathrm{Poisson}(\lambda_1)$ と $\mathrm{Poisson}(\lambda_2)$ のカウントの和が $\mathrm{Poisson}(\lambda_1 + \lambda_2)$ に従うことを示せ。
   2. レート λ の過程の各リクエストが確率 q で独立にログに記録されるとする。記録されたリクエストがレート $q\lambda$ のポアソン過程をなすことを、リクエスト数で条件付けて二項分布をポアソン分布にわたって足し合わせることで示せ。毎分 $\lambda = 4$、$q = \frac{1}{4}$ のとき、ある 1 分間に何もログに記録されない確率はいくらか。
   3. ちょうど n 個の事象が $[0, t]$ 内に生じたという条件のもとで、それらの発生時刻は n 個の独立な $U(0,t)$ からのサンプルを昇順に並べたものと同じ分布に従うことを示せ。どのスライスも等確率で事象を含むという細分化の描像を用いよ。
   4. [code] 指数分布の間隔によってレート 4 の過程を 1,000 分間シミュレーションし、$q = \frac{1}{4}$ で間引きを行って、間引き後の毎分カウントの平均と分散が 1 に近いことを確認せよ。ちょうど 3 個の事象が生じた分における各事象の分内の位置を集め、そのヒストグラムが平坦であることを確認せよ。

7. **多項分布の共分散。**●多項分布のカウントベクトルを $\mathbf{x} = \sum_{i=1}^{n} \mathbf{e}_{Y_i}$、すなわち n 個の独立なワンホットベクトルの和（$P(Y_i = k) = p_k$）として表す。
   1. $\mathbb{E}[\mathbf{e}_Y] = \mathbf{p}$ および $\mathrm{Cov}(\mathbf{e}_Y) = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$ を示し、そこから $\mathbb{E}[\mathbf{x}] = n\mathbf{p}$ および $\mathrm{Cov}(\mathbf{x}) = n\big(\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top\big)$ を導け。さらに $\mathrm{Var}(x_k) = np_k(1-p_k)$ と $\mathrm{Cov}(x_j, x_k) = -np_jp_k$（$j \neq k$）を読み取れ。すべての共分散が負になるのはなぜか。
   2. $\mathrm{Cov}(\mathbf{x})\mathbf{1} = \mathbf{0}$ を示し、これを制約 $\sum_k x_k = n$ によって説明せよ。

3. 公正なサイコロを 12 回振るとき、Var(x_k)、Cov(x_j, x_k)、および 2 つのカウント間の相関を求めよ。
4. 残りの $K-1$ 個のクラスを 1 つにまとめることで、x_k 単体が Binomial(n, p_k) に従うことを示せ。

36.4 指数分布族

線形回帰、ロジスティック回帰、ソフトマックス回帰は、それぞれ異なる損失関数を最小化する。しかし、それらの損失をモデルの重みに関して微分すると、3 つともまったく同じ訓練規則に帰着する。すなわち、予測と観測の差に入力を掛けたものになる。一見別々にみえる 3 つの導出が 1 つの式に収束するとき、その背後には共通の構造が潜んでいるのが常である。36.3 節の分布カタログでも、各分布の平均や分散を求めるたびに独自の巧妙な代数的工夫が必要にみえたが、やはり同じ疑問に行き着く。

本章を通して登場する、13 回のコイントスで 9 回表が出たコインを考えよう。このコインのバイアスに対する自然な推定値は $9/13 \approx 0.69$ である。同じ 13 回のコイントスにバイアス項のみを持つロジスティック回帰モデルを当てはめると、モデルは間接的な経路をたどって $9/13$ に到達する。最終層が対数オッズ（log-odds）$\log(9/4) \approx 0.81$ を計算し、シグモイドがその数値を $9/13$ へと写し戻すのである。確率を直接扱うのではなく、なぜわざわざ対数オッズを経由する回り道をとるのだろうか。

1 つの考え方が、この両方の謎を解き明かす。カタログに載っているほぼすべての分布は、少数のデータ統計量の線形結合の指数関数を通してのみパラメータが現れる、単一の正準形（canonical form）で書き表せる。この構造を持つ分布は**指数分布族**（exponential family）をなし、コインの場合、指数部に現れるパラメータこそが対数オッズにほかならない。驚くべきことに、この形式だけから 4 つの中心的な性質が導かれる。第 1 に、平均と分散は単一の関数の 1 階および 2 階の導関数にすぎない。第 2 に、最尤法は勾配が常に「予測マイナス観測」となる凸最適化問題になる。第 3 に、クラスラベルをモデル化するとソフトマックスが自動的に現れる。第 4 に、同じ形式の事前分布はカウントを足し合わせるだけで更新される。

深層学習においては 2 番目の性質が中心的な結果であり、1 番目の性質がその証明を与える（36.4.3.2 節および 36.4.3.3 節）。この 2 つを合わせれば、カウントのような新しい種類のラベルにどの損失関数を組み合わせるべきか、またコインがすべての試行で表になったときのように最尤法による当てはめが有限の解を持たないのはいつか、といったことがわかる。なお、本節のうち 2 つの箇所は初読時には読み飛ばしてもよい。指数形式がどこから生じるかを説明する最大エントロピーによる導出（36.4.2 節）と、ノイズを伴う選択および統計物理学からのソフトマックスの別導出である。

36.4.1 多様な分布を統一する単一の形式

まずはカタログの中で最も単純な分布である 1 回のコイントスから始め、対数オッズが自然に姿を現すまで式を書き換えてみよう。このわずかな式変形から、一般形がどのような姿になるべきかが見えてくる。

36.4.1.1 指数形式によるベルヌーイ分布

バイアス p のコインでは、表（$x=1$）が確率 p で、裏（$x=0$）が確率 $1-p$ で出る。前節ではこの両方のケースを 1 つの式 $p^x(1-p)^{1-x}$ にまとめた。結果 x とパラメータ p がどのように関わるかを見るため、対数をとって x を含む項をくくり出そう。

$$\log P(X=x) = x\log p + (1-x)\log(1-p) = x\log\frac{p}{1-p} + \log(1-p).$$

2 つの役割がいかにきれいに分離しているかに注目してほしい。結果は x と対数オッズ $\log\frac{p}{1-p}$ の積を通してのみ現れ、第 2 項は p のみに依存する。ここで対数オッズに $\eta = \log\frac{p}{1-p}$ という名前を付け、第 2 項を η で書き直そう。p について解くと $p = e^{\eta}/(1+e^{\eta})$ となるため、$1-p = 1/(1+e^{\eta})$ であり、$\log(1-p) = -\log(1+e^{\eta})$ となる。両辺の指数をとると次式が得られる。

$$P(X=x) = \exp\big(x\eta - \log(1+e^{\eta})\big), \qquad x \in \{0,1\}.$$

これでコインの確率は、x の線形関数から x にまったく依存しない項を引いたものの指数関数となった。この第 2 項は単なる正規化定数に対応する。実際、正規化前の 2 つの重みは $e^{0}=1$ と e^{η} であり、その和は $1+e^{\eta}$ である。

この式に現れた 3 つの構成要素は、本節を通して繰り返し登場する。**統計量**（statistic）$T(x)=x$ は、結果のうち本質的に効いてくる部分である。そこに掛かるのが**自然パラメータ**（natural parameter）η であり、ベルヌーイ分布の場合は対数オッズ、すなわち**ロジット**とも呼ばれる量にあたる。最後に、**対数分配関数**（log-partition function）$A(\eta) = \log(1+e^{\eta})$ は正規化定数の対数であり、**ソフトプラス**として見覚えのある関数だろう。先ほどの変数変換を逆にたどると、次のようになる。

$$p = \frac{e^{\eta}}{1+e^{\eta}} = \operatorname{sigmoid}(\eta), \qquad \operatorname{sigmoid}(\eta) = \frac{1}{1+e^{-\eta}}.$$

したがって、ロジスティック回帰のシグモイドとは、コインの自然パラメータをその平均へと写し戻す写像にほかならない。誰かが恣意的に発明する必要はなかったのである。

本章のコインの例では、経験的なバイアス $9/13$ に対応する自然パラメータは次のようになる。

$$\hat{\eta} = \log\frac{9/13}{4/13} = \log\frac{9}{4} \approx 0.811,$$

そしてシグモイドはこれをそのまま元の値へと戻す。すなわち $\operatorname{sigmoid}(\log\frac{9}{4}) = \frac{9/4}{1+9/4} = \frac{9}{13}$ である。なぜ $\hat{\eta}$ が最尤法による対数オッズの値になるのか、さらに $\hat{\eta}$ の周りでの損失の曲率がその推定値の不確実性について何を教えてくれるのかは、まもなく確認する。

36.4.1.2 定義と属する分布

ベルヌーイ分布の書き換えには、データの統計量、それに掛かるパラメータ、そして正規化定数という 3 つの要素があった。これを複数の統計量へと同時に一般化し、さらに各結果に対してパラメータに依存しない固定の重みを許容すると、カタログの大半を収める単一のテンプレートが得られる。

ある分布の族は、その確率質量関数または確率密度関数が次の形で書けるとき、**指数分布族**であるという。

$$p(\mathbf{x} \mid \eta) = h(\mathbf{x})\,\exp\big(\eta^{\top}T(\mathbf{x}) - A(\eta)\big). \tag{36.51}$$

この形式では、データとパラメータは内積 $\eta^\top T(\mathbf{x})$ を通してのみ相互作用する。**基底測度**（base measure）$h(\mathbf{x}) \geq 0$ はパラメータとは独立に各結果へ固定の重みを割り当て、$h > 0$ となる領域が分布の台（support）を定める。**自然パラメータ** $\eta \in \mathbb{R}^d$ は調整可能なパラメータであり、**十分統計量**（sufficient statistic）$T(\mathbf{x}) \in \mathbb{R}^d$ は自然パラメータが掛かるデータの特徴量である。**対数分配関数**は正規化定数（normalizing constant）の対数であり、次式で定義される。

$$A(\eta) = \log \int h(\mathbf{x})\, \exp(\eta^\top T(\mathbf{x}))\, d\mathbf{x}, \tag{36.52}$$

連続密度の場合は積分をとり、離散的な $\mathbf{x}$ の場合は和をとる。平たく言えば、A とは Equation 36.51 の積分または和を 1 にするための定数であり、有効な自然パラメータとは $A(\eta)$ が有限になるようなパラメータのことである。ベルヌーイ分布は $h(x) = 1$、$T(x) = x$、$A(\eta) = \log(1 + e^\eta)$ とした特別な場合にあたる。

なぜ特徴量 $T(\mathbf{x})$ は**十分**と呼ばれるのだろうか。それは、η についてデータから知り得る情報のすべてを保持しているからである。独立な n 個の観測に対する同時尤度は、次のように因数分解される。

$$\prod_{i=1}^{n} p(\mathbf{x}_i \mid \eta) = \Big(\prod_{i=1}^{n} h(\mathbf{x}_i)\Big) \exp\Big(\eta^\top \sum_{i=1}^{n} T(\mathbf{x}_i) - n A(\eta)\Big).$$

パラメータは和 $\sum_i T(\mathbf{x}_i)$ を通してのみデータと関わる。この和が等しい 2 つのデータセットは、η に依存しない定数倍だけが異なる尤度を生じるため、まったく同じパラメータ推定値を与える。コインの場合、この和は単なる表の合計回数である。13 回のコイントスのどの並べ替えもあらゆるバイアスのもとで同じ確率を持つため、9 回表が出たことさえわかれば、その出現順序はコインについてそれ以上の情報を何も付け加えない。

各項を Equation 36.51 と照らし合わせれば、この族に属する他の分布もすぐに特定できる。レート λ のポアソン分布では、事象が k 回起きる確率は $\frac{\lambda^k e^{-\lambda}}{k!} = \frac{1}{k!}\exp(k\log\lambda - \lambda)$ であるから、$h(k) = 1/k!$、$T(k) = k$、$\eta = \log\lambda$、$A(\eta) = e^\eta = \lambda$ と読み取れる。平均と分散がともに未知のガウス分布には 2 つの統計量が必要になる。指数部の二乗を展開して x と x^2 の項に整理すると、次のようになる。

$$\frac{1}{\sqrt{2\pi}} \exp\Big(\underbrace{\tfrac{\mu}{\sigma^2}}_{\eta_1}\, x + \underbrace{\big(-\tfrac{1}{2\sigma^2}\big)}_{\eta_2}\, x^2 - \underbrace{\big(\tfrac{\mu^2}{2\sigma^2} + \log\sigma\big)}_{A(\eta)}\Big),$$

したがって、$T(x) = (x, x^2)^\top$ および $h(x) = 1/\sqrt{2\pi}$ となる。第 2 の自然パラメータが厳密に負でなければならない点に注目しよう。$\eta_2 < 0$ のとき、因子 $e^{\eta_2 x^2}$ がガウス分布の急速な裾の減衰をもたらすのに対し、$\eta_2 \geq 0$ では A の正規化積分が発散する。Table 36.1 に、これらを含む代表的な指数分布族の分布をまとめる。

Table 36.1: 指数分布族に属する分布。各行の確率または密度は $h(\mathbf{x})\exp(\eta^\top T(\mathbf{x}) - A(\eta))$ であり、二項分布では試行回数 n、ラプラス分布では位置 μ を固定する必要がある。

分布	$h(\mathbf{x})$	$T(\mathbf{x})$	η	$A(\eta)$
ベルヌーイ分布（p）	1	x	$\log\frac{p}{1-p}$	$\log(1+e^\eta)$
二項分布（n, p）	$\binom{n}{x}$	x	$\log\frac{p}{1-p}$	$n\log(1+e^\eta)$

分布	$h(\mathbf{x})$	$T(\mathbf{x})$	η	$A(\eta)$
ポアソン分布（λ）	$1/k!$	k	$\log\lambda$	e^{η}
カテゴリカル分布（one-hot $\mathbf{y}$）	1	$\mathbf{y}$	ロジット $\mathbf{z}$	$\log\sum_k e^{z_k}$
指数分布（λ）	1（$x \geq 0$ のとき）	x	$-\lambda$	$-\log(-\eta)$
$\mathcal{N}(\mu, \sigma^2)$	$1/\sqrt{2\pi}$	(x, x^2)	$\left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right)$	$-\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\log(-2\eta_2)$
ラプラス分布 (μ, b)	1	$\lvert x-\mu\rvert$	$-1/b$	$\log(-2/\eta)$

カテゴリカル分布の行はロジットを用いてクラス確率を表しており、ここからソフトマックスが生じる。その起源については 36.4.4 節で詳しく探る。さらに高次元にはもう 2 つの重要な分布がある。十分統計量 $\mathbf{x}$ と $\mathbf{x}\mathbf{x}^{\top}$、自然パラメータ $\Sigma^{-1}\mu$ と $-\frac{1}{2}\Sigma^{-1}$ を持つ多変量ガウス分布、そして試行回数を固定した多項分布である。なお、h、η、T への分解は一意ではないことに留意しておこう。のちに重要となる唯一の冗長性はカテゴリカル分布の行にある。one-hot ベクトルの K 個の成分の和は常に 1 になるため、K 個のロジットのうち 1 つは冗長なのである（36.4.3.2 節）。

36.4.1.3 指数分布族に属さない分布

すべての分布が Equation 36.51 に収まるわけではなく、2 つの古典的な例外はそれぞれ示唆に富む異なる理由から指数分布族からはみ出す。

まず、一様分布 $U(0, \theta)$ を考えよう。その密度は $1/\theta$（区間 $[0, \theta]$ 上）であり、その外側では 0 である。ここでは、台そのものがパラメータ θ とともに移動する。対照的に、指数分布族では指数因子が決して 0 にならないため、台は $h > 0$ となる集合に固定されている。台が移動すると、推定の性質は根本から変わる。たとえば 3 個のサンプル 0.1、0.2、0.9 を観測したとする。尤度は θ^{-3}（$\theta \geq 0.9$ のとき）および 0（$\theta < 0.9$ のとき）となる（それより小さな θ からは 0.9 が決して生成されないからである）。したがって、最尤推定量は標本最大値 $\hat{\theta} = \max_i x_i = 0.9$ になる。

まもなく証明するように、指数分布族の内部では、最尤法は常に十分統計量のモデル平均をその標本平均と等しくおく。その処方箋をここにそのまま当てはめると、平均 $\theta/2$ を $\bar{x} = 0.4$ と等しくおくことになり、$\theta = 0.8$ が得られるが、この値のもとでは実際に観測されたサンプル 0.9 が起こり得ないことになってしまう。一様分布の尤度はデータの和ではなく最大値を通してデータに依存するため、本節の仕組みは適用できないのである。演習問題 7（36.6 節）では、標本サイズが増えるにつれてこれら 2 つの推定量がどう振る舞うかを比較する。

裾の重い分布は、第 2 の理由から指数分布族の外に出る。**スチューデントの** t 分布は、精度 $1/\sigma^2$ 自体がランダムなガウス分布をとることで生じる。その精度について平均をとると、急速に減衰する e^{-x^2} の裾が多項式減衰の裾に置き換わる（密度の導出は演習問題 4 を参照）。スケール 1、自由度 ν の場合、n 個の観測の尤度は因子 $\left(1 + (x_i - \mu)^2/\nu\right)^{-(\nu+1)/2}$ の積となり、データの有限個の和へと圧縮することはできない。実際、Pitman、Koopman、Darmois による古典的な定理によれば、台が固定された正則な分布族のうち、固定次元の十分統計量を持つのは指数分布族に限られる (Darmois 1935; Koopman 1936; Pitman 1936)。

実用上の帰結として、すべての指数分布族で成り立つことをのちに証明する負の対数尤度の凸性は、その外側では崩れ得る。スチューデントの t 分布では、データがよく離れた 2 つのクラスタに分かれているとき、位置パラメータ μ に関する負の対数尤度は各クラスタの近くに別々の極小値を持つようになる。

36.4.2 最大エントロピー

ここまでは、なじみのある分布を書き換えるだけで指数形式が現れた。しかしさらに踏み込んで、そもそもこの形式はどこから生じるのかと問うてみよう。ある確率的な量についてわかっているのが、少数の統計量の平均値だけだとする。そのとき、どの分布を選ぶべきだろうか。驚くべきことに、それらの平均値以外の仮定を一切置かない分布は、常に指数分布族に属する。

36.4.2.1 エントロピーと偏りのあるサイコロ

「平均値以外の仮定を一切置かない」ということを厳密にするには、分布がどれほど広がっているか、言い換えれば特定の値に偏らず余計な決めつけをしていないかを測る尺度が必要である。有限集合上の分布 $p = (p_1, \dots, p_K)$ に対して、**エントロピー**は次式で定義される。

$$H[p] = -\sum_{k=1}^{K} p_k \log p_k,$$

単位は nat とし、慣例として $0 \log 0 = 0$ と定める。各項は驚き $-\log p_k$（結果 k に対する値）をその確率で重み付けしているため、エントロピーとは平均的な驚きにほかならない。公正な 6 面サイコロではどの面にも $\log 6$ の驚きが割り当てられ、エントロピーは $\log 6 \approx 1.79$ nat となる。常に 6 の目しか出ないサイコロには驚きがまったくなく、エントロピーは 0 である。K 個の結果上のあらゆる分布の中で、一様分布が最大可能なエントロピー $\log K$ を達成する。この事実はすぐあとで証明しよう。

E. T. Jaynes はこの問題を具体的なパズルとして定式化した (Jaynes 1957)。ある 6 面サイコロの出目の平均が 4.5（公正な場合の 3.5 ではなく）であるとだけ知らされ、それ以外は何もわからないとする。$\{1, \dots, 6\}$ 上で平均 4.5 を持つ分布は数多く存在する。確率 $1/2$ を 3 に、確率 $1/2$ を 6 に割り当てる分布もその 1 つだが、この分布は 1、2、4、5 が決して出ないと大胆に断定しており、知らされた情報にはそのような根拠はどこにもない。この分布のエントロピーはわずか $\log 2 \approx 0.69$ である。

Jaynes の**最大エントロピー原理**（principle of maximum entropy）は、既知の平均と一致するすべての分布の中からエントロピーが最大の分布、すなわち手持ちの知識と整合する範囲で最も余計な仮定を置かない選択を採用する。証明に先立って答えを述べよう。一意な最大点は、一様分布の**指数傾斜**（exponential tilt）で与えられる。

$$p_k = \frac{e^{\eta k}}{\sum_{j=1}^{6} e^{\eta j}}, \qquad k = 1, \dots, 6,$$

ここで η は平均が 4.5 になるよう調整する。3 と 6 の目に集中した分布は、この平均を達成するために公正なサイコロの $\log 6 - \log 2 \approx 1.10$ nat（1.79 nat のうち）ものエントロピーを手放していた。下のコードを実行する前に、指数傾斜がどれだけのエントロピーを手放すことになるか少し予想してみてほしい。指数傾斜の平均は η に関して単調に増加するため、単純な二分法で正確な η の値が求まる。

```
k = np.arange(1, 7)
def tilt(eta):                                  # p_k proportional to exp(eta k)
    w = np.exp(eta * k)
    return w / w.sum()
```

```
lo, hi = -5.0, 5.0
for _ in range(50):                                  # bisection: the mean increases with eta
    eta = (lo + hi) / 2
    lo, hi = (eta, hi) if (tilt(eta) * k).sum() < 4.5 else (lo, eta)
p = tilt(eta)
print('eta =', round(eta, 3), ' p =', p.round(3))
print('H =', round(-(p * np.log(p)).sum(), 3), ' log 6 =', round(np.log(6), 3))
```

```
eta = 0.371  p = [0.054 0.079 0.114 0.165 0.24  0.347]
H = 1.614  log 6 = 1.792
```

得られる分布は各面にわたって等比的に変化し、どの面も 1 つ下の面より $e^{0.371} \approx 1.45$ 倍出やすくなる。そのエントロピーは 1.61 nat であり、公正なサイコロと比べてわずか 0.18 nat しか失われていない。また $\eta = 0$ とおけば公正なサイコロに戻る。Figure 36.18 では、3 通りの目標平均に対する指数傾斜を比較している。

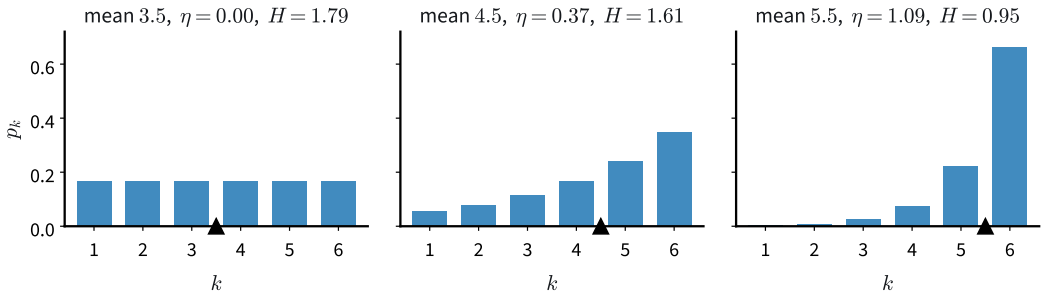


Figure 36.18: 出目の平均を 3.5、4.5、5.5（黒三角）に固定したサイコロの最大エントロピー分布。いずれも一様分布の指数傾斜 $p_k \propto e^{\eta k}$ であり、要求される平均が高くなるにつれて傾斜パラメータ η は大きくなり、エントロピー H は低下する。

この指数傾斜がすでに指数分布族の形式 Equation 36.51（6 つの面上で $h \equiv 1$、統計量 $T(k) = k$、自然パラメータ η）になっていることに注目しよう。統計量の平均を固定してエントロピーを最大化すると、その統計量の定数倍の指数関数へとまっすぐ導かれた。これはサイコロに限った偶然ではない。

36.4.2.2 最大エントロピーの命題

一般の定理を述べるには、結果の空間が自然な基準重みを備えている場合を扱えるよう、エントロピーを少しだけ拡張しておく必要がある。分布 q と基底測度 h に対し、**基準測度に対する相対エントロピー**（以下、h に対する相対エントロピー）を次式で定義する。

$$H_h[q] = -\sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{h(\mathbf{x})}$$

これは離散的な結果に対する式であり、連続密度の場合は和を積分に置き換える。$h \equiv 1$ とおけば、上で定義した通常のエントロピーに戻る。

証明では、任意の候補分布 q を指数分布族の分布 p と直接比較する。同じ空間上の 2 つの分布 p と q について、$p(\mathbf{x}) > 0$ が $q(\mathbf{x}) > 0$ となるすべての点で成り立つとする。このとき、q の p に対する**カルバック・ライブラー・ダイバージェンス**は、$D_{\mathrm{KL}}(q\|p) = \sum q \log(q/p)$（$q > 0$ とな

る結果にわたる和、密度の場合は対応する積分）で定義される (Cover and Thomas 1999)。エントロピーとこのダイバージェンスについては 37.1 節で詳しく学ぶが、ここで必要なのは 1 本の接線だけで証明できる初等的な事実 1 つだけである。対数関数は凹関数であるため、$\log u \le u - 1$ がすべての $u > 0$ に対して成り立ち、等号は $u = 1$ のときに限られる（Figure 36.19）。

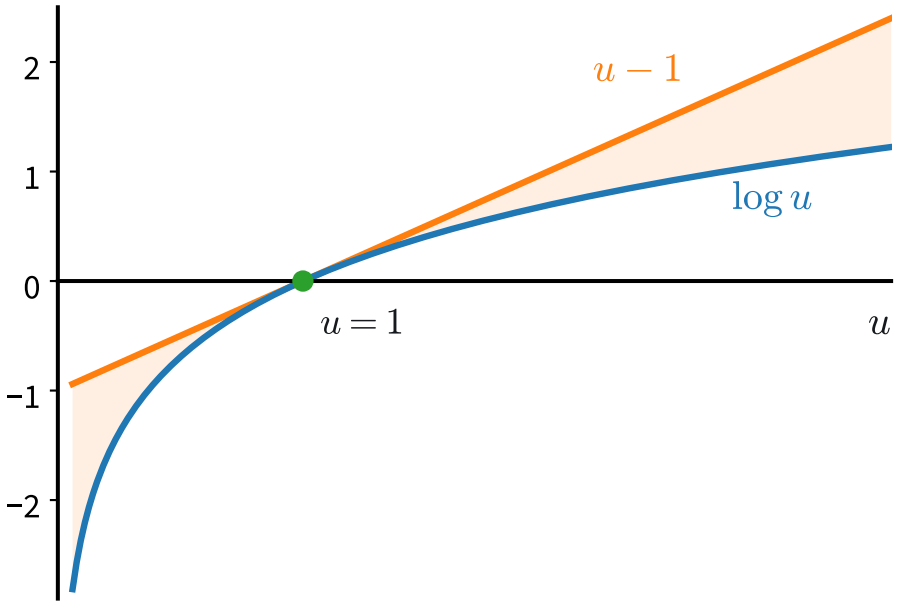


Figure 36.19: 接線による上界 $\log u \le u - 1$。凹関数である対数（青）は、$u = 1$ での接線（オレンジ）よりもすべての $u > 0$ において下側に位置し、その 1 点でのみ接する。

補題（ギブスの不等式）：カルバック・ライブラー・ダイバージェンスは決して負にならない。すなわち、

$$D_{\mathrm{KL}}(q\|p) = \sum_{\mathbf{x}:\, q(\mathbf{x})>0} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \ge 0, \tag{36.53}$$

であり、等号が成り立つのは $q = p$ のとき、かつそのときに限る。

証明：接線による上界 $\log u \le u - 1$ を $u = p(\mathbf{x})/q(\mathbf{x})$ に適用し、$q(\mathbf{x}) > 0$ となるすべての結果について和をとる。

$$-D_{\mathrm{KL}}(q\|p) = \sum_{q>0} q \log \frac{p}{q} \le \sum_{q>0} q\Big(\frac{p}{q} - 1\Big) = \sum_{q>0} p - 1 \le 0.$$

最初の不等式で等号が成り立つには、$p = q$ が $q > 0$ となるすべての点で成り立つ必要がある。すると $\sum_{q>0} p = 1$ が強制され、p には q の台の外側に残る確率質量がなくなるため、いたるところで $p = q$ となる。■

命題（最大エントロピー）：$p(\mathbf{x}) = h(\mathbf{x}) \exp(\eta^\top T(\mathbf{x}) - A(\eta))$ を指数分布族に属する分布とし、q を（連続の場合は密度を持ち）h が 0 になるすべての点で 0 となり、有限の平均統計量 $\mathbb{E}_q[T(\mathbf{x})]$ を持ち、かつその平均において p と一致する（すなわち $\mathbb{E}_q[T(\mathbf{x})] = \mathbb{E}_p[T(\mathbf{x})]$ を満たす）任意の

分布とする。このとき $H_h[q] \le H_h[p]$ が成り立ち、等号は $q = p$ のとき、かつそのときに限り成り立つ。

証明： $H_h[q]$ の中の対数に p を挟んで $\log \frac{q}{h} = \log \frac{q}{p} + \log \frac{p}{h}$ と分解する。第 1 項からは $-D_{\mathrm{KL}}(q\|p) \le 0$ がギブスの不等式より得られる（$p > 0$ が $h > 0$ のすべての点で成り立つため適用できる）。第 2 項は十分統計量に関して線形であり、$\log \frac{p}{h} = \eta^{\top} T - A(\eta)$ となるため、q は T の平均を通してのみ現れる。

$$-\sum_{\mathbf{x}} q \log \frac{p}{h} = -\eta^{\top} \mathbb{E}_q[T] + A(\eta) = -\eta^{\top} \mathbb{E}_p[T] + A(\eta) = H_h[p].$$

中央の等号では平均の制約を用い、最後の等号では p を q の代わりに置いて同じ計算を繰り返した。2 つの項を合わせると $H_h[q] = H_h[p] - D_{\mathrm{KL}}(q\|p) \le H_h[p]$ が得られ、等号は $q = p$ の場合に限り成り立つ。$\blacksquare$

鍵となるステップは 2 番目にある。$\log(p/h)$ が T に関して線形であるため、交差項は q に $\mathbb{E}_q[T]$ のみを通して依存し、その値は制約によって固定されている。平たく言えば、**指数分布族のある分布が平均の制約を満たすなら、その分布は同じ制約を満たすすべての分布の中でエントロピーを最大にする一意な分布である。**注目すべきことに、ラグランジュ乗数は一切不要であり、この証明は単なる停留点ではなく大域的最大値を直接保証している。

偏りのあるサイコロでは、$\eta \approx 0.371$ とした指数傾斜が $h \equiv 1$、$T(k) = k$ の指数分布族に属し、かつ平均 4.5 を持つため、これが一意な最大点となる。さらに、なじみ深い 3 つの分布も系として直ちに従う。

- **制約なし**（$T \equiv 0$）：すべての q が許容され、最大点は $p \propto h$ となる（ただし h の和が有限であるとする）。$h \equiv 1$ とした有限集合上では、これはエントロピー $\log K$ の一様分布であり、公正なサイコロは $K = 6$ の場合にあたる。
- $\mathbb{R}$ **上での平均と分散**：$T(x) = (x, x^2)$ かつ定数の h をとると、最大点はガウス分布になる。定数の h は H_h を一定値だけ平行移動させるにすぎないため、所定の平均と分散を持つすべての密度関数の中でガウス分布が最大のエントロピーを持つ。
- **カウント上の平均**（$\{0, 1, 2, \dots\}$ 上で $T(k) = k$）：$h(k) = 1/k!$ とれば、H_h を平均 m のもとで最大化する最大点はレート $\lambda = m$ のポアソン分布になる一方、$h \equiv 1$ とれば幾何分布 $p_k = (1 - r) r^k$（ただし $r = m/(1 + m)$）が得られる。演習問題 2 で詳しく調べるように、どの分布が現れるかは基底測度によって決まる。

指数分布族のいずれかの分布が制約を満たすという仮説を落とさないよう注意してほしい。たとえば平均 m のみを $\mathbb{R}$ 全体の上で固定し、$h \equiv 1$ とした場合を考えよう。候補となる密度 $e^{\eta x}$ は $\mathbb{R}$ 上でいかなる η に対しても可積分ではないため、該当する分布は存在しない。それどころか最大点自体が存在しない。実際、ガウス分布 $\mathcal{N}(m, s^2)$ は平均 m とエントロピー $\frac{1}{2} \log(2\pi e s^2)$ を持ち、これは $s \to \infty$ とすると際限なく大きくなる。

36.4.3 対数分配関数

36.3 節の分布カタログでは、平均と分散を求めるたびに別々の工夫が必要にみえた。ベルヌーイ分布では $X^2 = X$、二項分布では独立なコイントスの和、ポアソン分布では級数の添字ずらし、ガウス分布では部分積分といった具合である。しかし指数分布族の内部では、それらの計算はすべて対数分配関数 A を微分するという単一の操作に帰着する。1 階導関数が平均を、二階導関数が分散を与え、しかも同じ関数がそのまま最尤法を支配する。

36.4.3.1 モーメントの性質

$A(\eta)$ は指数関数 $\exp(\eta^\top T(\mathbf{x}))$ の積分または和の対数であった。このような対数を微分すると、因子 $T(\mathbf{x})$ が $\mathbf{x}$ の確率で重み付けされた形で降りてくる。それはまさに T の期待値にほかならない。

命題（十分統計量の平均）：指数分布族の分布 Equation 36.51 に対して、次式が成り立つ。

$$\nabla_\eta A(\eta) = \mathbb{E}[T(\mathbf{x})]. \tag{36.54}$$

証明：@eq-mdl-log-partition を積分記号の下で微分する。この微分と積分の交換は、$A(\eta) < \infty$ となる自然パラメータの集合の内部で正当化される (Folland 1999)。この集合はヘルダーの不等式により凸であり、以降は η をその内部にとる。$Z(\eta) = e^{A(\eta)} = \int h(\mathbf{x}) e^{\eta^\top T(\mathbf{x})} d\mathbf{x}$ と書けば、連鎖律より $\nabla A = (\nabla Z)/Z$ となり、ここで分子は次のようになる。

$$\nabla_\eta Z = \int h(\mathbf{x})\, T(\mathbf{x})\, e^{\eta^\top T(\mathbf{x})}\, d\mathbf{x}.$$

これを $Z = e^A$ で割ると因子 e^{-A} が積分の中に入り、密度 Equation 36.51 が完成する。

$$\begin{aligned} \nabla_\eta A &= \frac{1}{Z} \int h(\mathbf{x})\, T(\mathbf{x})\, e^{\eta^\top T(\mathbf{x})}\, d\mathbf{x} \\ &= \int T(\mathbf{x}) \underbrace{h(\mathbf{x})\, e^{\eta^\top T(\mathbf{x}) - A(\eta)}}_{p(\mathbf{x}\mid\eta)}\, d\mathbf{x} = \mathbb{E}[T(\mathbf{x})]. \quad \blacksquare \end{aligned}$$

同様の積分と微分の交換のもとで $\nabla A = (\nabla Z)/Z$ をもう 1 回微分すると、十分統計量の共分散行列が得られる。

$$\nabla^2 A = \frac{\nabla^2 Z}{Z} - \frac{(\nabla Z)(\nabla Z)^\top}{Z^2} = \mathbb{E}[T\,T^\top] - \mathbb{E}[T]\,\mathbb{E}[T]^\top = \mathrm{Cov}(T).$$

要するに、A の傾きは T の平均であり、その曲率は T の共分散である。あらゆる共分散行列は半正定値であるため（$\mathbf{v}^\top \mathrm{Cov}(T)\mathbf{v} = \mathrm{Var}(\mathbf{v}^\top T) \geq 0$ がすべての $\mathbf{v}$ に対して成り立つ、33.3.2.2 節）、対数分配関数 A は常に凸関数となる。

なじみのある 2 つの例で確認してみよう。コインの場合、$A(\eta) = \log(1 + e^\eta)$ からベルヌーイ分布の平均 $A'(\eta) = \mathrm{sigmoid}(\eta) = p$ と、その分散 $A''(\eta) = \mathrm{sigmoid}(\eta)(1 - \mathrm{sigmoid}(\eta)) = p(1-p)$ が得られる。$\hat{\eta} = \log \frac{9}{4}$ において、これらはそれぞれ $9/13 \approx 0.692$ および $\frac{9}{13} \cdot \frac{4}{13} = \frac{36}{169} \approx 0.213$ となる。ポアソン分布では $A(\eta) = e^\eta$ は微分しても自分自身のままであるから、$\lambda = 4$（$\eta = \log 4$）において A' と A'' はともに 4 になる。モーメントは単なる導関数であるため、コードで書ける任意の A に対して自動微分で直接計算できる。両方のケースを確かめてみよう。

```
def derivatives(A, eta):
    """First and second derivative of the scalar function A at eta."""
    eta = torch.tensor(eta, dtype=torch.float64, requires_grad=True)
    (dA,) = torch.autograd.grad(A(eta), eta, create_graph=True)
    (d2A,) = torch.autograd.grad(dA, eta)
```

```
    return dA.item(), d2A.item()

cases = [('Bernoulli', torch.nn.functional.softplus, math.log(9 / 4), 9 / 13, 36 / 169),
         ('Poisson', torch.exp, math.log(4), 4, 4)]
for name, A, eta, mean, var in cases:
    dA, d2A = derivatives(A, eta)
    print(f"{name:9s} A' = {dA:.4f} (mean {mean:.4f})  A'' = {d2A:.4f} (variance {var:.4f})")
```

```
Bernoulli A' = 0.6923 (mean 0.6923)  A'' = 0.2130 (variance 0.2130)
Poisson   A' = 4.0000 (mean 4.0000)  A'' = 4.0000 (variance 4.0000)
```

どちらの導関数も、表示された 4 桁すべてにおいて解析的な平均および分散と一致している。分布カタログではべき級数の計算を要した「ポアソン分布の平均と分散が等しい」という有名な性質が、ここでは $A = A' = A''$（$A(\eta) = e^{\eta}$ に対して）という 1 行に凝縮されていることに注目してほしい。

36.4.3.2 モーメントマッチングとしての最尤法

モーメントの性質により、最尤法は期待値に関する驚くほど明快な条件へと姿を変える。すなわち、当てはめたモデルは十分統計量の標本平均を再現しなければならない。独立な観測 $\mathbf{x}_1, \dots, \mathbf{x}_n$ が与えられたとき、Equation 36.51 の積からなる尤度の負の対数をとると次式が得られる。

$$-\sum_{i=1}^{n} \log p(\mathbf{x}_i \mid \eta) = n\,A(\eta) - \eta^{\top} \sum_{i=1}^{n} T(\mathbf{x}_i) - \sum_{i=1}^{n} \log h(\mathbf{x}_i). \tag{36.55}$$

ここでデータは和 $\sum_{i=1}^{n} T(\mathbf{x}_i)$ を通してのみ現れる。A は凸関数であり第 2 項は η に関して線形であるため、負の対数尤度は η に関して凸関数となる。その勾配を 0 とおき、モーメントの性質 Equation 36.54 を適用すると次式が得られる。

$$\nabla A(\hat{\eta}) = \bar{T}, \qquad \bar{T} = \frac{1}{n} \sum_{i=1}^{n} T(\mathbf{x}_i).$$

したがって最尤法は、モデルのもとでの十分統計量の期待値を観測された標本平均に一致させる。この原理を**モーメントマッチング**と呼ぶ。しかも目的関数が凸であるため、この方程式の解は自動的に大域的な最小点となる。

13 回のコイントスで 9 回表が出たコインで、これが何を与えるか見てみよう。負の対数尤度は $13A(\eta) - 9\eta = 13\log(1 + e^{\eta}) - 9\eta$ である。導関数 $13\,\mathrm{sigmoid}(\eta) - 9$ が 0 になるのはまさに $\mathrm{sigmoid}(\hat{\eta}) = 9/13$ のときであり、ここから $\hat{\eta} = \log \frac{9}{4} \approx 0.811$ が得られる。この最小点において、損失の値は次のようになる。

$$13 \log \frac{13}{4} - 9 \log \frac{9}{4} \approx 8.02 \text{ nats},$$

これはバイアス $9/13$ のコインのエントロピー $-\frac{9}{13}\log\frac{9}{13} - \frac{4}{13}\log\frac{4}{13} \approx 0.617$ の 13 倍である。最適点において、コイントス 1 回あたりの平均損失は当てはめられたコインのエントロピーに一致する。

この恒等式は $h \equiv 1$ であるすべての指数分布族で成り立つ。$\hat{\eta}$ における平均の負の対数尤度は $A(\hat{\eta}) - \hat{\eta}^\top \bar{T}$ である一方、当てはめられた分布のエントロピーは $A(\hat{\eta}) - \hat{\eta}^\top \nabla A(\hat{\eta})$ である。$\nabla A(\hat{\eta}) = \bar{T}$ であるから、この 2 つの数値は一致する。

最小点における損失の二階導関数は $13A''(\hat{\eta}) = 13 \cdot \frac{36}{169} = \frac{36}{13} \approx 2.77$ である。曲率が大きいということは、$\hat{\eta}$ から離れると損失が急激に増加し、対数オッズが狭い範囲に絞り込まれることを意味する。13 回のコイントスではまだかなりの不確定性が残っており、$1/\sqrt{2.77} \approx 0.60$ だけ $\hat{\eta} \approx 0.81$ から動かしても、損失は約 0.5 nat しか増えない。36.6 節ではこの曲率をフィッシャー情報量として位置づけ、エラーバーへと変換する。Figure 36.20 に、A の $\hat{\eta}$ における接線と負の対数尤度の形状を示す。

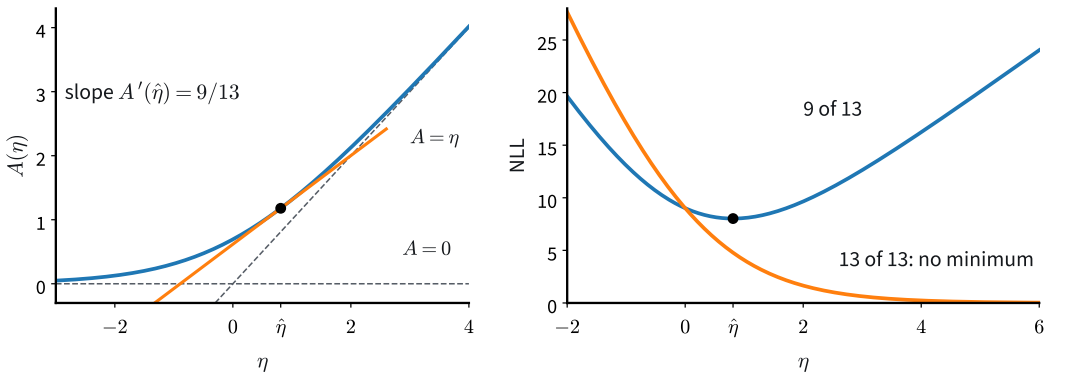


Figure 36.20: 左：ベルヌーイ分布の対数分配関数 $A(\eta) = \log(1 + e^\eta)$ とその漸近線、および $\hat{\eta} = \log(9/4)$ での接線（傾きは平均 $9/13$）。右：負の対数尤度 $13A(\eta) - x\eta$ のグラフ。13 回中 $x = 9$ 回表が出た場合は最小点 $\hat{\eta}$ を持ち、$x = 13$ 回表が出た場合はすべての η に対して減少し続け、有限の最小点を持たない。

ここで重要な注意点がある。モーメントマッチング方程式 $\nabla A(\hat{\eta}) = \bar{T}$ が解を持つのは、$\bar{T}$ がその分布族のいずれかの分布によって実際に達成可能な平均である場合に限られる。ベルヌーイ分布のコインでは、達成可能な平均は $\mathrm{sigmoid}(\eta)$（η は実数）、すなわち開区間 $(0, 1)$ である。もし 13 回のコイントスがすべて表になったなら、$\bar{T} = 1$ は境界上にあり達成できない。このとき負の対数尤度は $13A(\eta) - 13\eta = 13\log(1 + e^{-\eta})$ となり、あらゆる η に対して単調に減少する。すなわち、値は 9.01（$\eta = 0$ のとき）、0.087（$\eta = 5$ のとき）、0.0006（$\eta = 10$ のとき）と下がっていく。0 に漸近するが決して到達しないため、有限の最小点は存在せず、当てはめられた確率は 1 へと近づいていく（$\hat{\eta} \to \infty$ のとき）。

ロジスティック回帰も、訓練セット内の 2 つのクラスを超平面で完全に分離できるときにはまったく同じ壁に突き当たる。重みを大きくするほど損失が減り続け、重みが無限大へと発散してしまうのである。パラメータに事前分布を入れる（本節の最後で導入する）か、重み減衰を用いれば有限の最適解が回復する。詳細は演習問題 4（36.6 節）で確認する。

最小点が存在する場合、負の対数尤度が狭義凸であればその最小点は一意である。これは $\mathrm{Cov}(T)$ が正定値であるとき、すなわち十分統計量の自明でないいかなる線形結合も台全体で定数にならないときに成り立つ。このようなパラメータ化は**最小**（minimal）と呼ばれる。K 個すべてのロジットでパラメータ化したカテゴリカル分布は最小ではない。K 個の one-hot 座標の和は常に 1 になるため、すべてのロジットに同じ定数を加えても確率は一切変化しないからである。その結果、負の対数尤度は $\mathbf{1}$ の方向に平坦になり、最小点は 1 点ではなく直線をなす。それでも深層学習ライブラリはあえて K 個すべてのロジットを保持し、このシフト不変性を利用して指数をとる前に最大のロジットを引くことで数値安定性を確保している（3.3.5 節）。

36.4.3.3 一般化線形モデル

教師あり学習のモデルは、ラベル y の入力 $\mathbf{x}$ が与えられたもとでの条件付き分布を予測する。**一般化線形モデル**（generalized linear model）は、その分布を指数分布族 Equation 36.51 から選び、入力を線形変換 $\mathbf{W}^{\top}\mathbf{x}$ を通してモデルに組み込む（バイアス項は定数の特徴量に吸収させる）。最も明快な選択は、この線形変換をそのまま自然パラメータに等しくおく $\eta = \mathbf{W}^{\top}\mathbf{x}$ であり、この組み合わせは**正準リンク**（canonical link）と呼ばれる。モデルが予測する十分統計量の平均を $\mu(\mathbf{x}) = \nabla A(\eta) = \mathbb{E}[T(y) \mid \mathbf{x}]$ と書くと、1 つの例に対する負の対数尤度は $A(\eta) - \eta^{\top} T(y) - \log h(y)$ となる。モーメントの性質 Equation 36.54 より、η に関する勾配は単に $\mu(\mathbf{x}) - T(y)$ である。$\eta = \mathbf{W}^{\top}\mathbf{x}$ に連鎖律を適用すると、直ちに次式が得られる。

$$\nabla_{\mathbf{W}} \big(-\log p(y \mid \mathbf{x}) \big) = \mathbf{x} \, \big(\mu(\mathbf{x}) - T(y) \big)^{\top}. \tag{36.56}$$

どの指数分布族を選んでも、勾配は常に入力と予測誤差（予測平均マイナス観測統計量）の積になる。モーメントの性質を手にした今、本節の冒頭で提示した謎はわずか 2 行で解決する。さらに、損失は $\mathbf{W}$ に関して凸である。凸関数 A と $\mathbf{W}$ の線形関数を合成し、そこに線形項を加えたものだからである。ただし、このすっきりした勾配は正準リンクに固有の性質であることに留意しよう。36.4.4 節で扱うプロビットモデルは同じ線形スコアをガウス分布の累積分布関数に入力するため、その勾配には予測値に応じて変化する余分な因子が掛かる。指数分布族を入れ替えるだけで、4 つのなじみ深いモデルが現れる。

ラベル y	分布族	$T(y)$	平均 $\mu(\mathbf{x})$	モデル
実数	ガウス分布（分散 1）	y	η	線形回帰
はい・いいえ	ベルヌーイ分布	$y \in \{0, 1\}$	$\mathrm{sigmoid}(\eta)$	ロジスティック回帰
K クラスのうち 1 つ	カテゴリカル分布	one-hot $\mathbf{y}$	$\mathrm{softmax}(\eta)$	ソフトマックス回帰
カウント	ポアソン分布	y	e^{η}	ポアソン回帰

あまり見慣れないラベルでこの規則を確かめるため、カウントデータに対するポアソンモデルを考えよう。ある入力に対してモデルが平均 $e^{\eta} = 3$ を予測し、観測されたカウントが $y = 5$ だったとすると、η に関する損失の勾配は $3 - 5 = -2$ となる。したがって勾配降下法のステップは η を増加させ、予測平均を 5 に向けて押し上げる。

表の 1 行目では、ガウス分布の分散を 1 に固定することで二次項 $e^{-y^2/2}$ が基底測度 $h(y)$ に組み込まれるため、負の対数尤度は定数項を除いて 2.1 節の二乗損失に一致する。深層ニューラルネットワークは、この最終層（自然パラメータへの線形変換とそれに続く指数分布族の負の対数尤度）をそのまま保ちつつ、生の入力 $\mathbf{x}$ を学習された表現に置き換える。ネットワークの重み全体に関して損失はもはや凸ではないが、出力層の自然パラメータに関する勾配は依然として「予測マイナス観測」のままである (Bishop 2006)。

36.4.4 ソフトマックスの起源

本編を通して、分類器は K 個のロジットからなるベクトル $\mathbf{z}$ を**ソフトマックス**によってクラス確率へと変換してきた。

$$\mathrm{softmax}(\mathbf{z})_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}, \qquad k = 1, \dots, K. \tag{36.57}$$

この写像が指数分布族の表のカテゴリカル分布の行から現れることは、すでに見たとおりである。ソフトマックスには明らかに望ましい性質がある。各確率は対応するロジットとともに増加し、すべての確率は正で、その和は 1 になる。しかし、そうした性質を持つ写像は他にも数多くある。たとえば $\log(1+e^{z_k})$ を e^{z_k} の代わりに正規化しても、ロジットのランキングを保つ正の確率が得られる。ソフトマックスを際立たせているのは、ロジットの差が対数オッズ $z_i - z_j = \log(p_i/p_j)$ になるという点である。驚くべきことに、異なる 3 つの分野に由来する 3 つの独立した議論がすべてこの同じ式へと収束する。すなわち、人間のノイズを伴う選択のモデル、統計物理学のボルツマン分布、そして指数分布族である。3 番目の議論が最も短く、本節の残りの部分が依拠するのもこれだけである。

36.4.4.1 ノイズを伴う選択とガンベルノイズ

第 1 の議論は、そもそも離散的なラベルがどのように生成されるかをモデル化する。商品を選ぶ買い物客や画像にラベルを付けるアノテータのような意思決定者を想像してみよう。各選択肢 k に潜在的なスコア z_k を割り当てているものの、ときおり誤りを犯すとする。ノイズを伴う選択をモデル化する自然な方法は、各スコアに独立なランダムノイズ ϵ_k を加え、摂動後のスコアが最も高い選択肢を選ぶことである。

$$y = \operatorname*{argmax}_k \, (z_k + \epsilon_k).$$

このモデルのもとでは、クラス k の確率は選択肢 k がノイズ込みの競争に勝つ確率となるため、その具体的な式はノイズの分布によって決まる。たいていの人が最初に試すのはガウスノイズだろう。選択肢が 2 つで $\epsilon_k \sim \mathcal{N}(0, s^2)$ のとき、選択肢 1 が勝つのは $\epsilon_2 - \epsilon_1 < z_1 - z_2$ の場合である。差 $\epsilon_2 - \epsilon_1$ 自体が分散 $2s^2$ のガウス分布に従うため、勝つ確率は次のようになる。

$$P(y=1) = \Phi\Big(\frac{z_1 - z_2}{\sqrt{2}\, s}\Big),$$

ここで Φ は標準ガウス分布の累積分布関数（cumulative distribution function、CDF）を表す（密度は ϕ と書く）。これが**プロビット**モデルであり、選択確率はノイズのスケールを単位として測ったスコア差のみに依存する。

残念ながら、プロビットモデルは選択肢が 3 つ以上になった途端に閉じた形を失う。選択肢 k が残りの $K-1$ 個の競合すべてに勝つ確率は $K-1$ 個のガウス分布の CDF の積の積分となり、初等的な公式が存在しないため、訓練の毎ステップですべての訓練例に対して勾配ともども数値積分しなければならなくなる。

ガウスノイズからガンベルノイズへ切り替えると、その積分は完全に消え去る。標準**ガンベル**分布の CDF は $G(x) = \exp(-e^{-x})$ である。この分布は極値理論において、ガウス分布や指数分布のように指数関数的に減衰する裾を持つ分布から独立に引いた多数のサンプルの最大値を適切にシフト・スケーリングしたときの極限分布として自然に現れる (Gumbel 1954)。さらに McFadden (1974) は、経済的選択をモデル化するためにこの分布を導入した。

命題（ガンベルノイズはソフトマックスを与える）： $\epsilon_1, \dots, \epsilon_K$ **が独立な標準ガンベル確率変数であるならば、ノイズ付き argmax はソフトマックス分布に従う。**

$$P\Big(\operatorname*{argmax}_j \, (z_j + \epsilon_j) = k\Big) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}.$$

証明：$S = \sum_j e^{z_j}$ とおく。摂動後の各スコア $z_j + \epsilon_j$ の CDF は $G(t - z_j) = \exp(-e^{z_j} e^{-t})$ である。選択肢 k がスコア $t = z_k + \epsilon_k$ をとったという条件のもとで、独立性より他の $K - 1$ 個の選択肢すべてが t を下回る確率は $\prod_{j \neq k} \exp(-e^{z_j} e^{-t}) = \exp\big(-(S - e^{z_k}) e^{-t}\big)$ となる。これに密度 $e^{z_k} e^{-t} \exp(-e^{z_k} e^{-t})$（$t = z_k + \epsilon_k$ の密度）を掛けて t について積分し、$v = e^{-t}$ と置換すると次式が得られる。

$$P(y = k) = \int_{-\infty}^{\infty} e^{z_k} e^{-t} \exp\!\left(-S e^{-t}\right) dt = e^{z_k} \int_0^{\infty} e^{-Sv}\, dv = \frac{e^{z_k}}{S}. \quad \blacksquare$$

この計算がうまくいくのは、ガンベル分布の CDF の積がやはりガンベル分布の CDF になるからである。全 K 個の選択肢にわたる最大スコアは $P\big(\max_j (z_j + \epsilon_j) \le t\big) = \exp(-S e^{-t}) = G(t - \log S)$ を満たす。つまり勝者のスコアは、標準ガンベル確率変数を $\log\text{-sum-exp} \log \sum_j e^{z_j}$ だけシフトしたものにすぎない。この見方に立てば、log-sum-exp はロジットの滑らかな最大値として働き、その勾配であるソフトマックスは各選択肢がその最大値を達成する確率を与える。演習問題 1 では、競合する指数分布の待ち時間を用いた同じ恒等式の別証明を扱う。

この命題を逆向きに読めば、そのままサンプリングアルゴリズムにもなる。すなわち、$\mathrm{softmax}(\mathbf{z})$ からカテゴリを 1 つサンプリングするには、各ロジットに独立な標準ガンベルノイズを加えて argmax をとればよい。これが広く使われている **Gumbel-max トリック**である。

36.4.4 節の残りでは本編にならい、T を十分統計量ではなく温度（ソフトマックス温度）を表す記号として用いる。$\mathrm{argmax}_k (z_k + T\epsilon_k) = \mathrm{argmax}_k (z_k / T + \epsilon_k)$ が任意の $T > 0$ に対して成り立つため、ガンベルノイズを T 倍にスケーリングすると $\mathrm{softmax}(\mathbf{z}/T)$ からのサンプリングになる。ノイズの大きい意思決定は高い温度に対応するわけである。4 つのロジットに対して 10^5 回の選択を $T = 1$ と $T = 2$ で実際にサンプリングし、経験的な勝率を厳密なソフトマックス確率と比較して Gumbel-max トリックを確かめてみよう。

```
z = np.array([2.0, 1.0, 0.1, -1.0])                    # four logits
softmax = lambda v: np.exp(v) / np.exp(v).sum()
rng = np.random.default_rng()
n = 100_000
for T in (1.0, 2.0):                                    # noise scale = temperature
    winners = np.argmax(z + T * rng.gumbel(size=(n, 4)), axis=1)
    freq = np.bincount(winners, minlength=4) / n
    print(f'T = {T:.0f}: argmax frequencies {freq.round(3)}')
    print(f'      softmax(z / T)       {softmax(z / T).round(3)}'
          f'  max diff {np.abs(freq - softmax(z / T)).max():.3f}')
```

```
T = 1: argmax frequencies [0.638 0.234 0.095 0.033]
      softmax(z / T)       [0.638 0.235 0.095 0.032]  max diff 0.001
T = 2: argmax frequencies [0.451 0.274 0.173 0.102]
      softmax(z / T)       [0.451 0.274 0.174 0.101]  max diff 0.002
```

経験頻度は $\mathrm{softmax}(\mathbf{z}/T)$ と数千分の 1 以内で一致しており、これは 10^5 回のサンプルにおけるモンテカルロ法のノイズ下限ちょうどである。ノイズのスケールを 2 倍にすると、ロジットを半分にしたときとまったく同様に勝率が平坦化される。

選択肢が 2 つの場合、独立な 2 つの標準ガンベル確率変数の差はロジスティック分布に従う。その CDF は $\mathrm{sigmoid}(x) = 1/(1 + e^{-x})$ であるから、この**ロジット**モデルは $P(y = 1) = \mathrm{sigmoid}(z_1 - z_2)$ を与える。シグモイドはスケール調整したガウス分布の CDF に極めてよく追随するため（$|\mathrm{sigmoid}(1.7x) - \Phi(x)| < 0.01$ がすべての x で成り立つ）、当てはめたプロビッ

トモデルとロジットモデルは通常の訓練例に対してほぼ同一の確率を出力する。両者が大きく異なるのは裾の振る舞いである。確信度高く誤ラベルが付いた外れ値に対し、ロジットモデルの損失はスコア差に関して線形にしか増えないのに対し、プロビットモデルの損失は二次関数的に増大するため、外れ値がプロビットモデルの当てはめをはるかに強く引きずってしまう（演習問題4）。

36.4.4.2 温度とボルツマン分布

第 2 の議論は統計物理学に由来する。統計物理学でも、まったく同じ指数法則がエネルギー状態を支配している。気体中の分子のように温度 T で熱平衡にある物理系は、状態 k（エネルギー E_k）を次の確率でとる。

$$p_k = \frac{\exp(-E_k/(k_{\mathrm{B}}T))}{\sum_j \exp(-E_j/(k_{\mathrm{B}}T))},$$

ここで k_{B} はボルツマン定数である。エネルギーの低い状態ほど出現しやすく、温度 T は確率が低エネルギー状態へどれほど鋭く集中するかを制御する。Boltzmann は気体分子に対してこの分布を発見し、Gibbs (1902) がその上に統計力学を築き上げた。温度をエネルギーの単位で測り（$k_{\mathrm{B}} = 1$ とおき）、エネルギーを負のロジットと同一視すれば（$E_k = -z_k$）、ボルツマン分布は $\mathrm{softmax}(\mathbf{z}/T)$ になる。この言葉で言えば、ロジットとは負のエネルギーであり、最も起こりやすいクラスとは最低エネルギー状態にほかならない。

前節の最大エントロピーの命題を使えば、物理学を一切持ち出さずに同じ式を導ける。物理学者は平均エネルギーを固定してそれ以外には何も仮定しないが、分類における対応する制約は平均ロジット $\sum_k p_k z_k$ を固定することである。K 個のクラスを $h \equiv 1$ の結果とし、ロジット z_k を統計量とみなそう。K 個のクラス上のすべての分布のうち、所定の平均ロジット（$\mathbf{z}$ の最小成分と最大成分の間に厳密に入る値）を達成する最大エントロピー分布は $p_k \propto e^{\eta z_k}$ である。目標平均がロジットの単純平均を上回るときは常に $\eta > 0$ となり、$T = 1/\eta$ とおけば $\mathrm{softmax}(\mathbf{z}/T)$ が復元される。T が ∞ から 0 へと冷えるにつれて、平均ロジットはロジットの単純平均から最大値へと滑らかに上昇する。シミュレーションで用いたロジット $\mathbf{z} = (2, 1, 0.1, -1)$ では、標準のソフトマックス（$T = 1$）は平均ロジット 1.49 を達成し、単純平均 0.525 と最大値 2 の間に位置している。

36.4.4.3 指数分布族としてのカテゴリカル分布

第 3 の議論は、本節の前半ですでに明らかにしたものである。クラスラベルを one-hot ベクトル $\mathbf{y}$ として符号化すると、カテゴリカル分布は $P(\mathbf{y}) = \exp(\mathbf{z}^{\top}\mathbf{y} - A(\mathbf{z}))$、対数分配関数は $A(\mathbf{z}) = \log \sum_k e^{z_k}$ となり、ロジットを自然パラメータとする指数分布族になる。モーメントの性質より、その平均は $\nabla A(\mathbf{z}) = \mathrm{softmax}(\mathbf{z}) = \mathbb{E}[\mathbf{y}]$ である。負の対数尤度は $\mathbf{z}$ に関して自動的に凸となり、Equation 36.56 よりその勾配は単に $\mathrm{softmax}(\mathbf{z}) - \mathbf{y}$ となる。

3 つの議論はすべて同じ関数に行き着くが、それぞれが異なる側面に光を当てている。ノイズを伴う選択は厳密なサンプラを与え、ボルツマン分布は温度スケーリングの意味を説明し、指数分布族は凸な損失と簡潔な勾配をもたらす。

36.4.5 共役事前分布

最尤法は、13 回のコイントスで 9 回表が出たという結果を $9/13$ という 1 つの数値に圧縮する。しかし点推定だけでは、その値にどれほど確信を持てるかはわからない。13 回中 9 回の表でも、1300 回中 900 回の表でも、比率はまったく同じになるからである。知識をコインのバイアスに関

する確率分布として表せばその不確実性を直接捉えられるし、理想的には新しいコイントスが得られるたびにゼロからやり直すのではなく単純な更新で取り込めることが望ましい。

ここからは次節以降の記法に合わせ、バイアス自体を未知の確率変数として扱い θ と書く。データを見る前の θ に関する分布を**事前分布**（prior）と呼ぶ。ベイズ則は事前分布に尤度を掛けて正規化し、$p(\theta \mid x) \propto p(\theta)\,P(x \mid \theta)$ によって**事後分布**（posterior）を作り出す。事後分布が事前分布とまったく同じ分布族に属し、データを観測してもパラメータを更新するだけで済むとき、その事前分布は尤度に対して**共役**（conjugate）であるという。

36.4.5.1 コインに対するベータ事前分布

コインのバイアスに対する事前分布は、$[0,1]$ 上に台を持つ密度でなければならない。θ の関数として見ると、表が x 回（n 回のコイントス中）出る尤度は $\theta^{x}(1-\theta)^{n-x}$ であるから、同じ代数形式を持つ事前分布を選べば指数を足し合わせるだけで尤度と結合できる。その正規化定数には 36.3 節で導入したガンマ関数 $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt$ を用いる。これは $\Gamma(z+1) = z\,\Gamma(z)$ および $\Gamma(n+1) = n!$ によって階乗を一般化する関数である。パラメータ $a, b > 0$ を持つ**ベータ**分布は、次の密度を持つ。

$$\operatorname{Beta}(\theta \mid a, b) = \frac{\theta^{a-1}(1-\theta)^{b-1}}{B(a,b)}, \qquad 0 \le \theta \le 1, \tag{36.58}$$

ここで正規化定数 $B(a,b) = \int_0^1 \theta^{a-1}(1-\theta)^{b-1}d\theta$ は $\Gamma(a)\Gamma(b)/\Gamma(a+b)$ に等しい。密度に θ を掛けると a が 1 増えるだけなので、この正規化定数から平均がたちどころに求まる。

$$\mathbb{E}[\theta] = \frac{1}{B(a,b)}\int_0^1 \theta^{a}(1-\theta)^{b-1}\,d\theta = \frac{B(a+1,b)}{B(a,b)} = \frac{a}{a+b},$$

ここでは $\Gamma(a+1) = a\,\Gamma(a)$ と $\Gamma(a+b+1) = (a+b)\Gamma(a+b)$ を使ってガンマ関数の因子を約分した。同じ工夫を 2 回適用すれば分散も求まり、平均を $m = a/(a+b)$ と書けば次のようになる。

$$\operatorname{Var}(\theta) = \frac{ab}{(a+b)^2(a+b+1)} = \frac{m(1-m)}{a+b+1}.$$

$a+b$ の値が大きいほど密度は平均の周りにより鋭く集中し、$a = b = 1$ のときは $[0,1]$ 上の平坦な一様密度になる。

ここで、表が x 回（n 回のコイントス中）観測されたとしよう。事前分布に尤度を掛けると、指数がそのまま足し合わされる。

$$\underbrace{\theta^{a-1}(1-\theta)^{b-1}}_{\text{prior}} \cdot \underbrace{\theta^{x}(1-\theta)^{n-x}}_{\text{likelihood}} = \theta^{(a+x)-1}(1-\theta)^{(b+n-x)-1},$$

これを正規化すれば、観測されたカウントの分だけパラメータがシフトしたベータ密度に再びなる。

$$\theta \mid x \;\sim\; \operatorname{Beta}\bigl(a+x,\; b+n-x\bigr). \tag{36.59}$$

この更新には明快な**擬似カウント**（pseudo-count）としての解釈がある。すなわち、a と b は表と裏の事前のカウントとして働き、表が x 回、裏が $n-x$ 回観測されると、それらが集計値にそのまま加算される。事後平均は次式で与えられる。

$$\mathbb{E}[\theta \mid x] = \frac{a + x}{a + b + n},$$

これは擬似カウントを織り込んだうえでの表の割合にほかならない。この値は事前平均 $a/(a+b)$ と最尤推定量 x/n の間に位置し、x/n への収束は n が大きくなるにつれて進む。

ただし、1 つ微妙な違いに注意しておこう。事前分布が何回のコイントスに相当するかは、どの要約量を計算するかによって変わる。事後分布の両パラメータが 1 を超えるとき、事後最頻値は $(a+x-1)/(a+b+n-2)$ となるため、最頻値に対して事前分布が寄与するのは表 $a-1$ 回と裏 $b-1$ 回である。混乱しないようここで明確に整理しておくと、**a と b は事後平均における擬似カウントであるのに対し、$a-1$ と $b-1$ は事後最頻値における擬似カウントである**。一様事前分布 $\mathrm{Beta}(1,1)$ のもとでは、平均には仮想的な表 1 回と裏 1 回が含まれるが、最頻値には一切含まれない。

一様事前分布のもとで本章のコインに戻ると、13 回のコイントスで 9 回表が出た結果により、$\mathrm{Beta}(1,1)$ は $\mathrm{Beta}(10,5)$ へと更新される。その事後平均は $10/15 = 2/3$ であり、これは**ラプラスの継起則**（Laplace's rule of succession）として知られる古典的な公式 $(x+1)/(n+2)$ である (Laplace 1814)。最頻値 $9/13$ は最尤推定量と一致し、標準偏差は $\sqrt{(2/3)(1/3)/16} \approx 0.118$ となる。

もし代わりに 130 回のコイントスで 90 回表が出たなら、事後分布はより鋭い $\mathrm{Beta}(91,41)$ となり、平均は 0.689、標準偏差は 0.040 になる。10 倍の回数のコイントスを集めると、標準偏差は $2.9 \approx \sqrt{133/16}$ 分の 1 に縮小する。すなわち事後分布の不確実性は $1/\sqrt{n}$ のレートで狭まっていき、これは 36.5 節で見る標準誤差と同じスケーリングである。

次のコイントスで表が出る確率には、どの値を割り当てるべきだろうか。θ が与えられたもとではその確率は θ であるから、事後分布の信念にわたって平均をとると $P(\text{heads} \mid x) = \int_0^1 \theta\, p(\theta \mid x)\, d\theta = \mathbb{E}[\theta \mid x]$ となる。先ほどのコインでは、この予測確率は $2/3$ である。ラプラスの継起則は、次に表が出る確率をそのまま与えているのである。

共役性のおかげで、ベイズ則の正規化積分を計算せずに済んだ。では、事前分布が共役でない場合はどうすればよいだろうか。1 次元であれば、事前分布と尤度の積を θ の細かいグリッド上で評価し、その和で正規化するだけでよい。たとえば自然パラメータ上のガウス事前分布 $\eta \sim \mathcal{N}(0,1)$ を考えよう。これは 36.8 節でロジスティック回帰の重みに置くのと同じ種類の事前分布である。この事前分布はコインがおそらく公正に近いという信念を表しており、ベルヌーイ尤度に対して共役ではない。実際、変数変換によって θ に関する密度を求めると $\phi(\eta)/(\theta(1-\theta))$（ただし ϕ は標準ガウス密度）となり、ベータ密度にはならない。同じグリッド上で両方の事前分布のもとでの事後分布を計算し、要約量を比較してみよう。

```
x, n = 9, 13                                  # 9 heads in 13 flips
theta = np.linspace(0, 1, 100_001)[1:-1]      # interior grid points
eta = np.log(theta) - np.log1p(-theta)        # natural parameter
priors = {'Beta(1, 1)': np.ones_like(theta),  # densities in theta
          'eta ~ N(0, 1)': np.exp(-eta**2 / 2) / (theta * (1 - theta))}
for name, prior in priors.items():
    post = prior * theta**x * (1 - theta)**(n - x)
    post /= post.sum()                        # grid weights sum to one
```

```
    mean = post @ theta
    sd = math.sqrt(post @ (theta - mean)**2)
    print(f'{name:13s}: mean {mean:.3f}, sd {sd:.3f}, '
          f'mode {theta[post.argmax()]:.3f}')
```

```
Beta(1, 1)   : mean 0.667, sd 0.118, mode 0.692
eta ~ N(0, 1): mean 0.644, sd 0.111, mode 0.665
```

一様事前分布のもとでは、グリッド計算によって Beta$(10, 5)$ の厳密なモーメント、すなわち平均 0.667、標準偏差 0.118、最頻値 $9/13 \approx 0.692$ が再現される。対数オッズにガウス事前分布を置くと事後分布は公正なコインへと引き寄せられ、平均は 0.644、最頻値は 0.665 に下がり、標準偏差もわずかに縮んで 0.111 となる。グリッド計算なら非共役な事前分布も共役な場合と同じくらい容易に扱えるが、それは θ がスカラーだからにすぎない。高次元のパラメータベクトルにスケールする手法は 36.8 節で展開する。Figure 36.21 は、コイントスが積み重なるにつれてベータ事後分布がどのように鋭くなるかを可視化したものである。

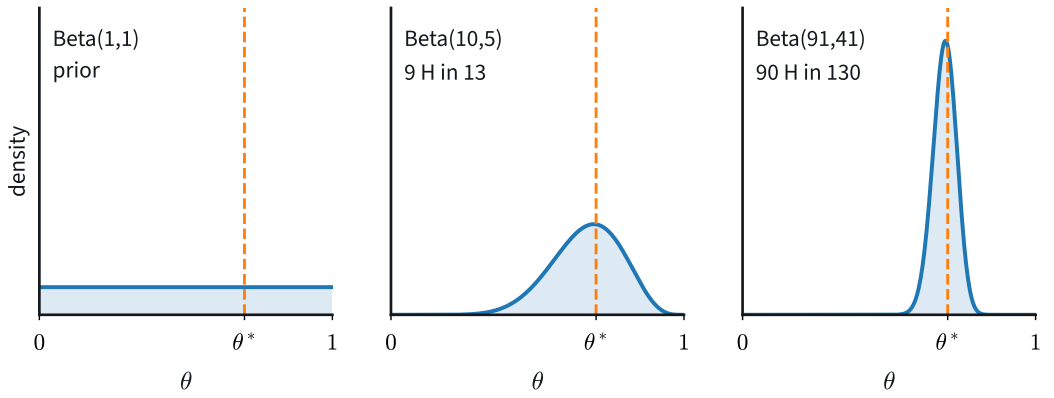


Figure 36.21: コインのバイアス θ のベイズ更新。左：平坦な事前分布 Beta$(1, 1)$。中央：事後分布 Beta$(10, 5)$（真のバイアス $\theta^* = 0.7$（破線）を持つコインを 13 回投げて 9 回表が出た後、平均 $2/3$、標準偏差 0.12）。右：事後分布 Beta$(91, 41)$（130 回投げた後、標準偏差 0.04）。各更新では観測された表と裏の回数が擬似カウントに加算され、事後分布の幅は $1/\sqrt{n}$ のオーダーで縮小する。

36.4.5.2 すべての指数分布族における共役性

一見すると、ベータ分布の更新は都合のよい代数的偶然のようにみえる。しかし実は、これは指数分布族の普遍的な性質である。すべての指数分布族は共役事前分布を持ち、その更新は常に十分統計量の加算に帰着する。

命題（共役事前分布）：尤度 Equation 36.51 と、その自然パラメータ上の次の事前分布を考える。

$$p(\eta) \propto \exp\!\big(\nu^\top \eta - \kappa\, A(\eta)\big), \tag{36.60}$$

ここで $\nu \in \mathbb{R}^d$ と $\kappa \in \mathbb{R}$ は、密度が正規化可能となるように選ぶ。n 個の独立な観測 $\mathbf{x}_1, \dots, \mathbf{x}_n$ を得た後、事後分布はまったく同じ形式を保ち、ν は $\nu + \sum_i T(\mathbf{x}_i)$ に、κ は $\kappa + n$ に更新される。

証明：η に依存しない因子を除くと、尤度からの寄与は $\exp\!\big(\eta^{\top}\sum_i T(\mathbf{x}_i) - nA(\eta)\big)$ である。これに事前分布を掛けると、指数部が 1 行でまとまる。

$$e^{\nu^{\top}\eta - \kappa A(\eta)} \cdot e^{\eta^{\top}\sum_i T(\mathbf{x}_i) - nA(\eta)} = e^{(\nu + \sum_i T(\mathbf{x}_i))^{\top}\eta - (\kappa + n)A(\eta)}.$$

この積を正規化すれば事後分布が得られるため、事後分布は更新後のパラメータ $\nu + \sum_i T(\mathbf{x}_i)$ と $\kappa + n$ を持つ同じ分布族に属する。■

ハイパーパラメータと呼ばれる事前分布のパラメータ ν と κ は、**擬似データ**（pseudo-data）のように振る舞う。すなわち ν は十分統計量の事前の和として、κ は事前の標本サイズとして働く (Bishop 2006)。ベルヌーイ分布のコインでは、この事前分布はロジット $\eta = \log\frac{\theta}{1-\theta}$ で書かれており、η から θ へ変数変換すると $\mathrm{Beta}(\nu, \kappa - \nu)$ に変換される。そして一般の更新則 $\nu \to \nu + x$、$\kappa \to \kappa + n$ は、ベータ分布の更新 $a \to a + x$、$b \to b + n - x$ をそのまま再現する。演習問題 7 では、ヤコビ行列を含めてこの変数変換を具体的に計算する。

36.4.5.3 ガンマ・ポアソン共役とディリクレ・多項共役

広く使われるさらに 2 つの共役ペアも同じパターンに従う。1 つは事象のレートに対するもの、もう 1 つは単語やクラス上のカテゴリカル確率に対するものである。レート λ の関数として見ると、n 個のポアソンカウント $x_1, \dots, x_n$ の尤度は $\lambda^{\sum_i x_i} e^{-n\lambda}$ に比例する。これに対応する共役事前分布は、形状パラメータ $a > 0$ とレートパラメータ $b > 0$ を持つ**ガンマ**分布であり、その密度は $\mathrm{Gamma}(\lambda \mid a, b) = \frac{b^a}{\Gamma(a)}\lambda^{a-1}e^{-b\lambda}$（$\lambda > 0$、平均 a/b）で与えられる。事前分布と尤度を掛け合わせると事後分布 $\mathrm{Gamma}\big(a + \sum_i x_i,\ b + n\big)$ が得られるため、事前分布は仮想的な a 回の事象が仮想的な b 単位期間にわたって生じたことを表している。演習問題 7 では、一般の命題からこのペアを直接導出する。

ベータ事前分布を 2 つの結果から K 個のカテゴリへと一般化するには、確率ベクトル $\mathbf{p}$（ただし $p_k \geq 0$ かつ $\sum_k p_k = 1$）上の密度である**ディリクレ**分布を用いる。集中度パラメータ $a_k > 0$ を持つディリクレ分布の密度は次式で与えられる。

$$\mathrm{Dir}(\mathbf{p} \mid \mathbf{a}) = \frac{\Gamma(\sum_k a_k)}{\prod_k \Gamma(a_k)} \prod_{k=1}^{K} p_k^{a_k - 1},$$

この分布の平均は $\mathbb{E}[p_k] = a_k / \sum_j a_j$ であり、$K = 2$ のときにはベータ分布に帰着する。各カテゴリのカウント $x_1, \dots, x_K$ を観測すると尤度は $\prod_k p_k^{x_k}$ に比例するため、事後分布は $\mathrm{Dir}(\mathbf{a} + \mathbf{x})$ となる。すなわち、各カテゴリの観測カウントがそれぞれの擬似カウントに加算される。

これが実用上なぜ重要かを見るため、3 語の語彙でカウント $(3, 0, 1)$ が得られたとしよう。最尤推定量 $(0.75, 0, 0.25)$ は未出現の 2 番目の単語に確率 0 を割り当ててしまうため、将来その単語を含む文書が現れるとあり得ない文書だと判定してしまう。そこで一様事前分布 $\mathrm{Dir}(1, 1, 1)$ を $\mathbf{p}$ に置くと、事後分布は $\mathrm{Dir}(4, 1, 2)$ となり、その平均は $(4/7, 1/7, 2/7) \approx (0.571, 0.143, 0.286)$ となる。正規化の前にすべてのカウントに 1 を加える**プラス 1 平滑化**（add-one smoothing）は、このディリクレ事前分布のもとでの事後平均にほかならない。36.11 節のナイーブベイズ分類器では、その 2 クラス版を用いる。

閉じた形で事後分布が求まるのは原則というより例外であり、共役ペアの外側ではベイズ推論に数値近似が必要になる（36.8 節）。しかしそれに取り組む前に、点推定に関するより基本的な問いに答えておく必要がある。13 回のコイントスからエラーバーなしの単一の数値 $9/13$ を得たが、この $9/13$ は真のバイアスからどれほど離れ得るのだろうか。また、13 回中 11 回表が出たコインは 9 回表が出たコインと本当に異なるのだろうか。36.5 節ではこれらの問いを取り上げる。

36.4.6 まとめ

分布カタログのほぼすべての分布は指数分布族の形式 $h(\mathbf{x})\exp(\eta^{\top}T(\mathbf{x})-A(\eta))$ を共有しており、そこではデータとパラメータが内積 $\eta^{\top}T(\mathbf{x})$ のみを通して関わり合い、統計量の平均を固定してエントロピーを最大化するとまさにこの形式が現れる。対数分配関数 A を微分するとモーメントが生成される。すなわち ∇A は T の平均であり、$\nabla^2 A$ はその共分散であることから、A が凸であることが直ちに証明される。その結果、最尤法はモデルの十分統計量の期待値を標本平均に一致させる凸最適化問題となり、その標本平均が達成可能である限り有限の解を持つ。指数分布族に正準リンクを組み合わせれば、勾配が常に「予測マイナス観測に入力を掛けたもの」になる一般化線形モデルが得られ、線形回帰、ロジスティック回帰、ソフトマックス回帰、ポアソン回帰が統一されると同時に、プロビットのような非正準リンクが異なる振る舞いを示す理由も説明される。ソフトマックスはカテゴリカル分布族の平均写像として現れる（ガンベルノイズやボルツマン分布からも同様に導かれる）。最後に、すべての指数分布族は十分統計量を加算するだけで更新される共役事前分布を持ち、コインの場合、$\mathrm{Beta}(a,b)$ は事後平均に対して a と b の擬似カウントを、最頻値に対しては $a-1$ と $b-1$ の擬似カウントを寄与する。

36.4.7 演習問題

1. **指数分布の競合：** $X_1,\dots,X_m$ を $X_i\sim\mathrm{Exp}(\lambda_i)$ に従う互いに独立な確率変数とする（たとえば m 本の独立な事象ストリームにおける次の事象までの待ち時間）。
   1. $\min_i X_i\sim\mathrm{Exp}\big(\sum_i\lambda_i\big)$ となることを示せ。
   2. j 番目のストリームが最初に発火する確率が $P(X_j=\min_i X_i)=\lambda_j/\sum_i\lambda_i$ となることを示せ。
   3. $V_1,\dots,V_K$ を独立な $\mathrm{Exp}(1)$ の確率変数とし、$\mathbf{z}\in\mathbb{R}^K$ をロジットのベクトルとする。(b) を用いて、$\operatorname{argmax}_k\,(z_k-\log V_k)$ がソフトマックス確率 Equation 36.57 に従うことを示せ。$-\log V_k$ は標準ガンベル分布に従うため、これはガンベルノイズがソフトマックスを与えることの 2 つ目の証明になっている。
2. **非負整数上の最大エントロピー：** 分布 $(p_0,p_1,p_2,\dots)$（$\{0,1,2,\dots\}$ 上）であって平均 $m>0$ が固定されたものを考える。
   1. シャノンエントロピー $H[p]=-\sum_k p_k\log p_k$ を、制約 $\sum_k p_k=1$ および $\sum_k k\,p_k=m$ のもとでラグランジュ乗数（35.6.1 節）を用いて最大化せよ。停留点が幾何分布 $p_k=(1-r)\,r^k$ となることを示し、r を m で表せ。
   2. $h(k)=1/k!$ に対する相対エントロピー $H_h[p]=-\sum_k p_k\log\big(p_k/h(k)\big)$ について同じ導出を繰り返し、停留点が $\mathrm{Poisson}(m)$ となることを示せ。
   3. $m=4$ において、両方の分布のシャノンエントロピーを数値的に計算せよ。どちらが大きいか、またなぜそうでなければならないか。2 つの分布のもとで $P(X\ge 15)$ を比較せよ。基底測度の選択は、大きなカウントに関してどのような仮定を織り込んでいるか。
   4. ラグランジアンの停留点は最大点であるとは限らない。(a) と (b) の 2 つの分布がそれぞれ平均 m を持つ指数分布族に属することを具体的に示すことで、最大エントロピーの命題から両者が大域的な最大点であることを証明せよ。
3. **対数分配関数からのモーメント計算：** モーメントの性質 Equation 36.54 とその 2 次の形式 $\nabla^2 A=\mathrm{Cov}(T)$ は、モーメントの計算を微分へと帰着させる。
   1. ポアソン分布では $A(\eta)=e^{\eta}$ である。A から平均と分散を導出せよ。関数 A のどの性質によって両者が等しくなるか。
   2. ガウス分布では、自然パラメータは $\eta_1=\mu/\sigma^2$ および $\eta_2=-1/(2\sigma^2)$ である。$A=-\eta_1^2/(4\eta_2)-\tfrac{1}{2}\log(-2\eta_2)$ となることを示し、これを微分して $\mathbb{E}[x]$、$\mathbb{E}[x^2]$、$\mathrm{Var}(x)$、$\mathrm{Cov}(x,x^2)$ を復元せよ。
   3. 位置 μ を固定したスケール b のラプラス分布は、十分統計量 $T(x)=|x-\mu|$ を持つ。

これを Equation 36.51 の形で書き、A を用いて $\mathbb{E}\,|x-\mu|$ と $\mathrm{Var}|x-\mu|$ を計算せよ。得られた結果をラプラス分布の分散 $2b^2$ と照合して確かめよ。

4. ガウス分布からの n 個の観測に対し、2 つのモーメントマッチング方程式 $\nabla A(\hat{\eta}) = \bar{T}$ を書き下し、$\hat{\mu}$ と $\hat{\sigma}^2$ について解け。$n=1$ のときはどうなるか。その答えを、$\bar{T}$ が達成可能な平均であるという条件と関連づけて説明せよ。

4. [code] **プロビットとロジット：** 2 つのクラスに対するスコア差を $m = z_1 - z_2$ とするとき、ロジットモデルは $P(y=1) = \mathrm{sigmoid}(m)$ を与え、ノイズスケール s のプロビットモデルは $P(y=1) = \Phi\big(m/(\sqrt{2}\,s)\big)$ を与える。
   1. 独立な 2 つの標準ガンベル確率変数の差が CDF $\mathrm{sigmoid}(x)$ を持つことを示せ。標準ガンベル分布の分散は $\pi^2/6$ である。標準ガウス分布と同じく分散が 1 になるようなスケール c（CDF が $\mathrm{sigmoid}(cx)$ となる分布に対する値）を求めよ。
   2. スケール c として $\max_x \big|\,\mathrm{sigmoid}(cx) - \Phi(x)\big|$ を最小化する値、およびそのときの最大誤差の値を数値的に求めよ。なぜこのスケールは (a) の値より小さくなるのか。
   3. クラス 2 の例が決定境界の誤った側（$m>0$）にあるとする。ロジットモデルのもとでの負の対数尤度 $-\log \mathrm{sigmoid}(-m)$ が m と $m + e^{-m}$ の間に収まり、かつ m に関する導関数が 1 未満であることを示せ。さらに不等式 $\frac{x}{1+x^2}\,\phi(x) \le 1-\Phi(x) \le \frac{\phi(x)}{x}$ $(x>0)$ を用いて、プロビットモデルのもとでの負の対数尤度 $-\log\Phi\big(-m/(\sqrt{2}\,s)\big)$ が $m^2/(4s^2)$ のオーダーで増大し、その導関数が m に関して線形に増大することを示せ。
   4. 100 個の入力 $x \sim \mathcal{N}(1,1)$（クラス 1）と 100 個の入力 $x \sim \mathcal{N}(-1,1)$（クラス 2）をサンプリングし、$m = wx + b$ とおく。両方のモデルのもとで最尤法により (w,b) を当てはめよ。次に $x=6$ にクラス 2 とラベル付けされた例を 3 個追加し、再び当てはめよ。各モデルについて、これらの誤ラベル例がある場合とない場合の傾き w の比を計算し、モデル間の違いを (c) を用いて説明せよ。なお $\log\Phi$ の計算には `scipy.special.log_ndtr` のような数値的に安定なルーチンを用いよ。

5. **最小カテゴリカル分布族 :●** カテゴリカルな結果を one-hot ベクトル $\mathbf{x} \in \{0,1\}^K$ として表し、クラス確率 $\mathbf{p} = \mathrm{softmax}(\mathbf{z})$ を $T(\mathbf{x}) = \mathbf{x}$、自然パラメータ $\mathbf{z}$、$A(\mathbf{z}) = \log\sum_k e^{z_k}$ の指数分布族として書く。
   1. $\nabla^2 A(\mathbf{z})$ を計算し、これが共分散 $\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$（one-hot ベクトル $\mathbf{x}$ の共分散）に等しいことを示せ。なお、任意の one-hot ベクトルは $\mathbf{x}\mathbf{x}^\top = \mathrm{diag}(\mathbf{x})$ を満たす。$\nabla^2 A(\mathbf{z})\,\mathbf{1} = \mathbf{0}$ となること、およびデータセットに対する負の対数尤度の勾配が $\mathbf{1}$ と任意の $\mathbf{z}$ において直交することを示せ。小さなステップサイズの勾配降下法を初期点 $\mathbf{z}_0$ からロジットに対して実行し、データにはすべてのクラスが含まれているとする。勾配降下法が決して変化させない $\mathbf{z}$ の関数は何か。したがって、無数にある最小点のうちどれに近づくか。
   2. $z_K = 0$ に固定する。すると $\eta = (z_1, \dots, z_{K-1})$ および $T(\mathbf{x}) = (x_1, \dots, x_{K-1})$ を持つ最小表現が得られる。$\nabla^2 A(\eta)$ が、すべての $p_k > 0$ である限り正定値となることを示せ。各クラスのカウント $n_1, \dots, n_K$ が与えられたとき、モーメントマッチング条件 $\nabla A(\hat{\eta}) = \bar{T}$ を最尤法による推定値 $\hat{\eta}$ について解け。あるクラスで $n_k = 0$ のときはどうなるか。

6. [code] **ポアソン回帰 :●** カウントに対する一般化線形モデルは、ポアソン分布の自然パラメータを特徴量のアフィン関数とする。すなわち $y_i \mid \mathbf{x}_i \sim \mathrm{Poisson}(e^{\eta_i})$、$\eta_i = \mathbf{w}^\top \mathbf{x}_i + b$ とおく。
   1. 負の対数尤度 Equation 36.55 を用いて損失を $(\mathbf{w}, b)$ の関数として書き、その勾配とヘッセ行列を計算せよ。損失が凸であることを示せ。勾配が 0 であるという条件は、当てはめられた平均 $\hat{m}_i = e^{\hat{\eta}_i}$ にどのような方程式を課すか。
   2. $n = 1000$ 個の例を $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$、$\mathbf{w} = (-0.3, 1.0)$、$b = 0.5$ により生成せよ。0 を初期値として $\mathbf{u} = (\mathbf{w}, b)$ をニュートン法（更新ステップ $\mathbf{u} \leftarrow \mathbf{u} - \big(\nabla^2 L(\mathbf{u})\big)^{-1} \nabla L(\mathbf{u})$

を損失 L に対して繰り返す手法）で当てはめ、得られた解において (a) の方程式が成り立つことを確認せよ。

3. 各レートに独立な $g_i \sim \text{Gamma}(1, 1)$（平均 1）を掛け、条件付き平均を e^{η_i} のまま保ちつつカウントが過分散になるようにせよ。そのようなデータセット 200 個と、乗数を掛けないデータセット 200 個のそれぞれにポアソンモデルを当てはめよ。データセット間にわたる推定値の平均と標準偏差を比較し、平均の振る舞いを (a) の方程式を用いて説明せよ。

7. **自然パラメータからの共役事前分布：●**スカラーの自然パラメータ η に対し、共役事前分布 Equation 36.60 は $p(\eta) \propto \exp(\nu\,\eta - \kappa\,A(\eta))$ となり、n 個の観測によって $\nu \to \nu + \sum_i T(x_i)$ および $\kappa \to \kappa + n$ と更新される。
   1. ポアソン分布では $\eta = \log\lambda$、$A(\eta) = e^{\eta}$ である。ヤコビ行列（ヤコビアン）を含めて η から λ へ変数変換し、λ 上の事前分布がガンマ分布になることを示せ。その形状パラメータとレートパラメータを特定し、この更新が 36.4.5.3 節のガンマ事後分布を再現することを確かめよ。
   2. ベルヌーイ分布では $\eta = \log\frac{\theta}{1-\theta}$、$A(\eta) = \log(1 + e^{\eta})$ である。θ 上の事前分布が $\text{Beta}(\nu, \kappa - \nu)$ になることを示し、これが正規化可能となる (ν, κ) の値の範囲を求めよ。
   3. ロジット上の平坦な事前分布 $\nu = \kappa = 0$ をとる。これは θ 上にどのような密度を誘導するか、またその密度は正規化可能か。表が x 回（n 回のコイントス中）出た後、事後分布が正規化可能となるのはいつか、またその平均はいくらか。これを、一様事前分布のもとでの事後分布 Equation 36.59（$\text{Beta}(1, 1)$ を θ 上においた場合）の平均と比較せよ。この比較は「平坦な事前分布」という表現について何を示唆しているか。

36.5 統計学

本書のすべてのモデルは、最終的にホールドアウトデータ上で計算される数値（テスト正解率、検証損失、あるいはベンチマークスコア）によって評価される。この数値はモデル固有の性質として扱われがちだが、実際にはたまたまテストセットに入った特定のデータ例も反映している。ある分類器がテストの 1,000 問中 912 問に正解し、91.2% の正解率を出したとする。同じ情報源から得られた別の 1,000 問に対しては、スコアはわずかに高くも低くもなるだろう。つまり、91.2% という値は実のところ確率変数の 1 つの実現値にすぎない。では、2 つ目のモデルが同じ問題群のうち 921 問に正解したとしたらどうだろうか。このモデルは本当に優れているのか、それともたまたま今回の問題のバッチがこのモデルに有利だっただけなのだろうか。

統計学は、実際には抽出されなかった他のあらゆる標本にわたって推定値がどのように変動するかを問うことで、このような疑問に答える。この変動はある分布、すなわち推定値の**標本分布**（sampling distribution）に従い、本節で扱うすべての道具はこの分布を記述し、それに基づいて推論するために作られている。その中心と広がりから、推定値が系統的に的を外しているか、そしてどれほど広くばらつくかがわかる。仮説検定（hypothesis testing）は、観測された差が偶然だけで生じるばらつきの範囲内に無理なく収まるかどうかを調べる。信頼区間（confidence interval）は、データから棄却できないパラメータ値をすべて集めたものである。そして、標本分布を閉じた式で表せないときには、ブートストラップ法によって手元のデータから直接シミュレーションする。

深層学習においてこれらの道具が重要なのは、どのモデルを運用するか、どのハイパーパラメータが優れているか、新しいアイデアが有効だったかといったエンジニアリング上の意思決定のほとんどが、ノイズを含んだ数値の比較に基づいているからである。それ自体のノイズより小さな差に基づいて行動するのは、コイン投げで決めるのと変わらない。本章のこれまでの節では背後の分布が既知であるときにデータがどう振る舞うかを記述してきたが、本節では逆向きに進み、観測データから誠実なエラーバー付きの推定値を導く。データは外界から得られたものである必

要さえない。コンピュータ上で生成したサンプルを平均して期待値を近似するとき（モンテカルロ推定）、その結果もまた 1 つの標本平均（sample mean）にほかならないため、同じ理論からエラーバーが得られる（36.5.1.7 節）。同様に、テスト正解率は 1 問ごとに 1 回投げられるコインのように振る舞うため、前の節で登場したコイン（9 回の表が 13 回の試行で出た例）も妥当性確認の例として随所に再登場する。

本節から 3 つの考え方だけを覚えるとしたら、次の点に注目してほしい。すなわち、平均二乗誤差（mean squared error）は統計的バイアス（bias）の二乗と分散にきれいに分解されること、p 値は帰無仮説（null hypothesis）が真であるときに一様分布に従うこと、そして信頼区間とは仮説検定が棄却しないパラメータ値の集合にほかならないことである。最初に読む際は、一致性（consistency）とデルタ法（delta method）に関する議論は軽く読み流してよい。ただし、本節全体を通じて 1 つだけ注意点を心に留めておく必要がある。以下に示すあらゆる保証は、推定量（estimator）や検定をデータを**見る前に**選んだことを前提としている。そうでない場合にこれらの保証がいかに容易に崩れるかは、本節の最後で確認する。

```
from scipy.stats import norm
```

```
rng = np.random.default_rng()
```

36.5.1 推定量とその標本分布

先ほどの 91.2% という正解率をどこまで信頼できるかを判断する前に、その数値を生み出した規則に正確な名前を与え、テスト問題がどのように得られたかを表す単純な確率モデルを用意する必要がある。

36.5.1.1 推定量と標本比率

推定量とは、標本 $x_1, \dots, x_n$ を受け取り、数値 $\hat{\theta}_n$ によって未知のパラメータ θ を近似するあらゆる規則のことである。標本平均、標本中央値（sample median）、テストセット上の正答割合はいずれも推定量である。データはランダムであるため $\hat{\theta}_n$ もまた確率変数であり、サイズ n の標本抽出を繰り返したときの分布がその**標本分布**である。

この分類器のテスト正解率をモデル化するため、$x_i = 1$ をモデルが問題 i に正解した場合とし、それ以外の場合は $x_i = 0$ とする。分類器の真の正解率が θ である大きな問題プールからテスト問題が独立に抽出されると仮定すると、指示変数 x_i は平均 θ の独立同分布（i.i.d.）なベルヌーイ確率変数となる。推定値は標本比率（sample proportion）$\hat{\theta}$、すなわち n 問中の正答割合である。期待値の線形性より $\mathbb{E}[\hat{\theta}] = \theta$ であり、1.6 節と同様にその分散は $\mathrm{Var}(\hat{\theta}) = \theta(1-\theta)/n$ となる。言い換えれば、標本比率はまさに真の正解率を中心としており、典型的な誤差は $1/\sqrt{n}$ の速さで縮小する。$\theta = 0.912$ と $n = 1{,}000$ を代入すると、標準偏差は $\sqrt{0.912 \cdot 0.088 / 1000} \approx 0.0090$、すなわち約 0.9 パーセントポイントとなる。中心極限定理（central limit theorem）により、この標本分布はガウス分布でよく近似されるため、約 95% のテストセットにおいて、正解率は幅 ± 1.8 パーセントポイントで θ の周囲に収まる。

数式だけに頼るのではなく、この標本分布を直接観察してみよう。10,000 個の独立なテストセット（各 1,000 問）を、真の正解率が 0.912 である分類器についてシミュレーションし、得られる正解率の平均と広がりを確認する。

```
theta, n = 0.912, 1000
acc = rng.binomial(n, theta, 10000) / n      # accuracy on 10,000 test sets
print(f'mean {acc.mean():.4f}, std {acc.std():.4f}, '
      f'formula {np.sqrt(theta * (1 - theta) / n):.4f}')
```

```
mean 0.9119, std 0.0089, formula 0.0090
```

数式の予測どおり、シミュレーションされた正解率は 0.912 の周りに集まり、標準偏差は 0.0090 となる。実際には抽出しなかった仮想的なテストセット上の正解率を表すこの 10,000 個の数値の集まりこそが、本節のすべての道具が推論の対象とするものである。

ここで、921 問に正解した 2 つ目のモデルに戻ろう。もしそのスコアが別の独立な 1,000 問のテストセットで測定されていたならば、2 つの正解率の間に観測された 0.009 という差の標準偏差は $\sqrt{0.0090^2 + 0.0085^2} \approx 0.012$ となる（ここで 0.0085 は同じ公式を 0.921 で評価した値である）。0.009 の差はそれ自体の標準誤差（standard error）の約 0.7 倍にすぎず、2 つのモデルを区別するには到底足りない。しかし今回の例では、両方のモデルが**同じ** 1,000 問で評価されている。36.5.2.6 節では、この共有テストセットを活用することで比較がどれほど鋭敏になるかを確認する。

36.5.1.2 統計的バイアスと分散

推定量が真の値から外れる仕方は、大きく 2 通りある。平均的に中心からずれた場所を狙ってしまうか、自身の平均の周りに大きくばらつくかである。$\hat{\theta}_n$ の**統計的バイアス**（以下では単にバイアスとも呼ぶ）は 1 つ目の誤差、すなわち推定量の平均値と真のパラメータとの系統的なずれを測るものであり、次式で定義される。

$$\mathrm{Bias}(\hat{\theta}_n) = \mathbb{E}[\hat{\theta}_n] - \theta, \tag{36.61}$$

ここで期待値は標本の無作為抽出に関してとる。起こり得るすべての θ の値に対してバイアスが 0 であるとき、その推定量を**不偏**（unbiased）と呼ぶ。**分散**は 2 つ目の誤差、すなわち $\hat{\theta}_n$ が自身の中心の周りでどれほど広く変動するかを測るものであり、その平方根が**標準誤差**である。

$$\mathrm{Var}(\hat{\theta}_n) = \mathbb{E}\big[(\hat{\theta}_n - \mathbb{E}[\hat{\theta}_n])^2\big], \qquad \mathrm{se}(\hat{\theta}_n) = \sqrt{\mathrm{Var}(\hat{\theta}_n)}. \tag{36.62}$$

端的に言えば、バイアスは推定値が平均してどこに着地するかを表し、標準誤差は 1 回の抽出で得られる典型的な値がその平均からどれほど離れるかを表す。先ほどの正解率の例では、標本比率のバイアスは 0、標準誤差は 0.009 である。分散は θ の周りではなく推定量自身の平均の周りで測られる点に注意しよう。データを完全に無視して常に 0.5 を出力する頑固な規則は分散が 0 であり、その誤差はすべてバイアスから生じる。

36.5.1.3 標本分散

標本比率が不偏推定量（unbiased estimator）であることから、不偏性は自然に成り立つもののように思えるかもしれない。しかし、データから標準誤差を計算しようとした途端に、バイアスを持つ推定量が現れる。標本平均の標準誤差は母分散（population variance）σ^2 に依存するが、これはほとんどの場合未知であり、同じ標本から推定しなければならない。最も自然な選択は**プラグイン分散**（plug-in variance）、すなわち標本平均 $\bar{x}$ からの二乗偏差の平均であり、次式で与えられる。

$$s_0^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

驚くべきことに、s_0^2 は系統的に小さくなりすぎる。それはなぜだろうか。$\bar{x}$ は、**この**特定の標本に対して二乗偏差和を最小化する唯一の点だからである。それとは異なる点である真の平均 μ からの偏差を測れば和は常に大きくなるため、μ を $\bar{x}$ に置き換えると分子が縮小するのである。

自明でない最小の標本サイズである $n = 2$ を考えると、この縮小の大きさは一目瞭然となる。このとき標本平均は単に x_1 と x_2 の中点なので、両方の偏差は $\pm(x_1 - x_2)/2$ に等しくなり、次式が得られる。

$$\sum_{i=1}^{2}(x_i - \bar{x})^2 = \frac{(x_1 - x_2)^2}{2}.$$

x_1 と x_2 は互いに独立で平均 μ、分散 σ^2 を持つため、その差 $x_1 - x_2$ は平均 0、分散 $2\sigma^2$ を持つ。したがって右辺の期待値は σ^2 となる。これを $n = 2$ で割ると平均して $\sigma^2/2$ となり、真の分散の半分にしかならない。代わりに $n - 1 = 1$ で割れば、平均して σ^2 が復元される。驚くべきことに、これとまったく同じ $n - 1$ の補正によって、あらゆる標本サイズ n でバイアスが取り除かれる。

命題（不偏標本分散）。 **$x_1, \dots, x_n$ を平均 μ、有限の分散 σ^2 を持つ独立同分布の確率変数とし、$n \geq 2$ とする。このとき、次が成り立つ。**

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad \text{satisfies} \qquad \mathbb{E}[s^2] = \sigma^2. \tag{36.63}$$

証明。 $\bar{x}$ の周りの偏差と μ の周りの偏差を比較するため、$x_i - \bar{x} = (x_i - \mu) - (\bar{x} - \mu)$ と書いて二乗を展開する。$\sum_i (x_i - \mu) = n(\bar{x} - \mu)$ を用いて交差項を整理すると、次式が得られる。

$$\begin{aligned}\sum_{i=1}^{n}(x_i - \bar{x})^2 &= \sum_{i=1}^{n}(x_i - \mu)^2 - 2(\bar{x} - \mu)\cdot n(\bar{x} - \mu) + n(\bar{x} - \mu)^2 \\ &= \sum_{i=1}^{n}(x_i - \mu)^2 - n(\bar{x} - \mu)^2.\end{aligned}$$

交差項は最後の項の 2 倍に逆符号をつけたものになるため、中心を $\bar{x}$ にとると、μ にとった場合に比べて二乗和から $n(\bar{x} - \mu)^2$ が引かれる。ここで両辺の期待値をとる。各項 $\mathbb{E}[(x_i - \mu)^2]$ は σ^2 に等しいため、第 1 項の和の期待値は $n\sigma^2$ である。一方、標本平均の分散は σ^2/n であるから、$\mathbb{E}[n(\bar{x} - \mu)^2] = \sigma^2$ となる。したがって二乗偏差和の期待値は $(n-1)\sigma^2$ であり、$n-1$ で割ることで $\mathbb{E}[s^2] = \sigma^2$ が得られる。■

この係数を覚えるには、**自由度**（degrees of freedom）という考え方が役に立つ。n 個の偏差 $x_i - \bar{x}$ には和が 0 になるという制約があるため、自由に動けるのはそのうち $n - 1$ 個だけである。すなわち、データから $\bar{x}$ を推定することで自由度を 1 つ消費している。プラグイン分散 s_0^2 のバイアスは $\frac{n-1}{n}\sigma^2 - \sigma^2 = -\sigma^2/n$ であり、n が小さいときには無視できないが、標本が大きくなるにつれて消えていく。

ここで 1 つ注意点がある。不偏性は非線形変換を施すと保たれない。凹関数である平方根関数にイェンセンの不等式（Jensen's inequality）を適用すると $\mathbb{E}[s] \le \sqrt{\mathbb{E}[s^2]} = \sigma$ となり、s が定数でない限り狭義の不等号が成り立つ。$n = 2$ のガウスデータでは標本標準偏差は $s = |x_1 - x_2|/\sqrt{2}$ であり、その期待値は $\mathbb{E}[s] = \sigma\sqrt{2/\pi} \approx 0.80\,\sigma$ となる。つまり、不偏分散推定量の平方根をとっても、標準偏差は平均して 20% 過小評価される。

数値計算ライブラリでは、これら 2 つの分母を「自由度の補正値（delta degrees of freedom）」フラグで切り替える。`np.var(x, ddof=0)` は n で割り、`np.var(x, ddof=1)` は $n-1$ で割る。

36.5.1.4　バイアス・分散分解

バイアスと分散がどのように相互作用するかを見るため、一方を犠牲にしてもう一方を抑える推定量を意図的に作ってみよう。$n = 30$ 個のノイズを含む測定値 $x_i \sim \mathcal{N}(\theta, \sigma^2)$（ただし $\theta = 1$、$\sigma = 4$）を観測したとする。通常の標本平均 $\bar{x}$ は不偏であり、分散は $\sigma^2/n = 16/30 \approx 0.533$（標準誤差 0.73）となる。ここで**縮小平均**（shrunk mean）$c\,\bar{x}$（ただし係数は $0 < c < 1$）を考える。これは標本平均を 0 の方向へ引き寄せる推定量である。0 へ引き寄せることで $c\theta - \theta = (c-1)\theta$ のバイアスが生じるが、同時に分散は $c^2\sigma^2/n$ へと圧縮される。$c = 0.65$ のときバイアスは -0.35 になる一方、標準誤差は 0.73 から 0.47 へと大きく低下する。Figure 36.22 に両者の標本分布を並べて示す。

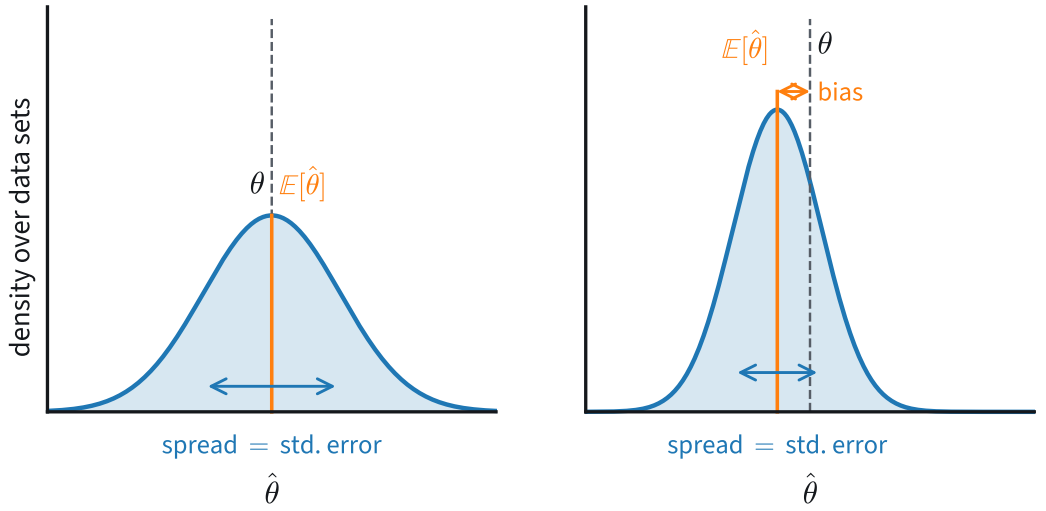


Figure 36.22: $\theta = 1$ に対する 2 つの推定量の標本分布（30 個の観測値を $\mathcal{N}(1, 16)$ から抽出した場合）。左：標本平均 $\bar{x}$（バイアス 0、標準誤差 0.73）。右：縮小平均 $0.65\,\bar{x}$（バイアス -0.35、標準誤差 0.47）。バイアスは中心の θ からのずれであり、標準誤差は分布の幅である。

どちらの推定量が実際には優れているのだろうか。縮小平均は系統的に的を外しているが、ばらつきがずっと小さいため、1 回の抽出で得られる値は多くの場合 θ に対して $\bar{x}$ よりも近くに着地する。この比較に決着をつけるには、両方の誤差を同時に考慮する単一のスコアが必要である。**平均二乗誤差**（MSE）とは、推定量から真のパラメータまでの二乗距離の平均であり、次式で定義される。

$$\mathrm{MSE}(\hat{\theta}_n) = \mathbb{E}\big[(\hat{\theta}_n - \theta)^2\big]. \tag{36.64}$$

都合のよいことに、MSE は先ほど定義した 2 つの量へと綺麗に分解される。

命題（バイアス・分散分解）。 $\hat{\theta}_n$ **が有限の分散を持つならば、次が成り立つ。**

$$\mathrm{MSE}(\hat{\theta}_n) = \mathrm{Bias}(\hat{\theta}_n)^2 + \mathrm{Var}(\hat{\theta}_n). \tag{36.65}$$

証明。 $m = \mathbb{E}[\hat{\theta}_n]$ を推定量の中心とし、二乗の内部で m を足して引く。

$$(\hat{\theta}_n - \theta)^2 = (\hat{\theta}_n - m)^2 + 2(\hat{\theta}_n - m)(m - \theta) + (m - \theta)^2.$$

ここで項ごとに期待値をとる。第 1 項は $\mathrm{Var}(\hat{\theta}_n)$ となり、第 3 項は定数 $\mathrm{Bias}(\hat{\theta}_n)^2$ である。交差項は、$m - \theta$ が定数であり、かつ $\mathbb{E}[\hat{\theta}_n - m] = 0$ が m の定義から成り立つため 0 になる。■

この恒等式には見事な幾何学的解釈がある。実はこれ自体がピタゴラスの定理にほかならない。確率変数を内積（inner product）$\langle U, V \rangle = \mathbb{E}[UV]$ を備えたベクトルとして扱うと、平均 0 の変動 $\hat{\theta}_n - m$ は任意の定数 a と直交する（$\langle \hat{\theta}_n - m,\, a \rangle = a\,\mathbb{E}[\hat{\theta}_n - m] = 0$ となるためである）。したがって、全誤差 $\hat{\theta}_n - \theta$ をこの変動と定数バイアス $m - \theta$ の和として書くことは、長さが se と $|\mathrm{Bias}|$ である互いに垂直な 2 辺へと分解することを意味し、それらの二乗和が斜辺の二乗 MSE に等しくなる。この視点に立てば、$\mathbb{E}[X]$ は X の定数からなる部分空間（subspace）への直交射影（orthogonal projection）であり、33.1 節で登場した射影において期待値がドット積の役割を果たしたものになっている。Figure 36.23 にこの直角三角形を示す。

この分解を使えば、推定量を定量的に順位付けできる。30 個の観測値からなる先ほどの問題では、公式から予測される MSE は標本平均で 0.533（すべて分散）、縮小平均では $0.123 + 0.225 = 0.348$（$0.65\,\bar{x}$ の場合）となる。標本中央値はどうだろうか。ガウス分布の対称性から不偏ではあるものの、その分散の公式はまだ導いていない。そこで、10,000 個のサイズ 30 のデータセットをシミュレーションし、3 つの推定量をすべて経験的に比較してみよう。

```
theta, sigma, n = 1.0, 4.0, 30
data = rng.normal(theta, sigma, (10000, n))   # 10,000 data sets of size 30
estimates = {'mean': data.mean(axis=1),
             '0.65 * mean': 0.65 * data.mean(axis=1),
             'median': np.median(data, axis=1)}
```

次に、`estimates` に格納した各推定量について、10,000 個のシミュレーションデータセットにわたる経験的な二乗バイアス、分散、および全 MSE を計算し、和 $\mathrm{Bias}^2 + \mathrm{Var}$ を解析的な予測値と比較する。

```
predicted = {'mean': sigma**2 / n,            # bias^2 + variance, by formula
             '0.65 * mean': (0.35 * theta)**2 + 0.65**2 * sigma**2 / n,
             'median': None}                  # no formula yet
print(f'{"estimator":>12} {"bias^2":>7} {"var":>6} {"MSE":>6} {"predicted":>9}')
for name, est in estimates.items():
    bias2 = (est.mean() - theta) ** 2
    var = est.var()                           # a plain average: divides by the count
    mse = ((est - theta) ** 2).mean()
    pred = '-' if predicted[name] is None else f'{predicted[name]:.3f}'
    print(f'{name:>12} {bias2:7.3f} {var:6.3f} {mse:6.3f} {pred:>9}')
```

```
   estimator  bias^2    var    MSE predicted
        mean   0.000  0.532  0.532     0.533
```

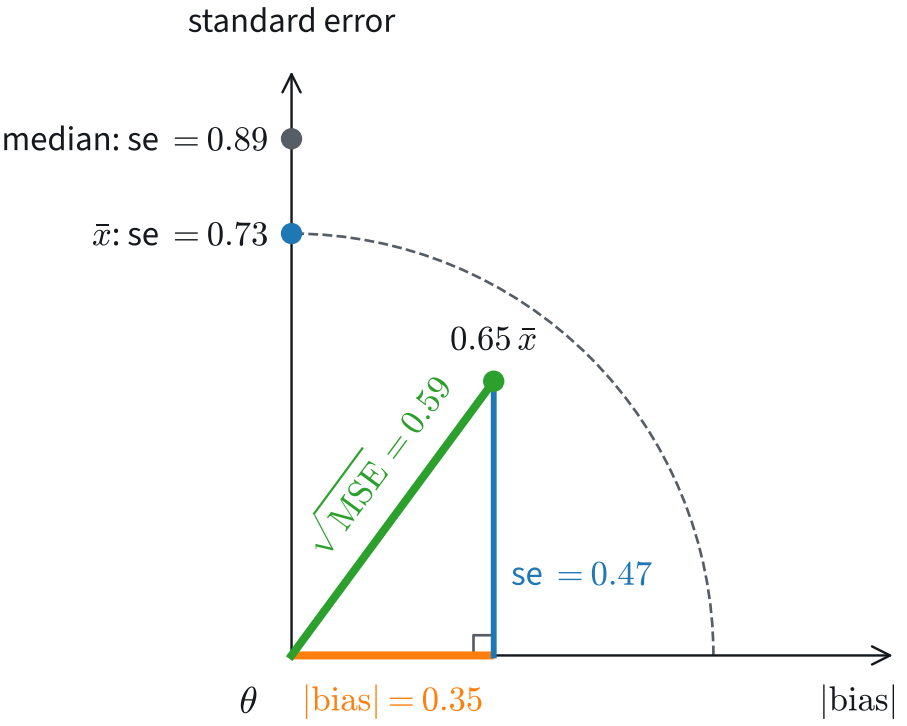


Figure 36.23: $\theta = 1$ の推定量（30 個の観測値を $\mathcal{N}(1, 16)$ から抽出）におけるピタゴラスの定理としてのバイアス・分散分解。各推定量は横座標 $|\text{bias}|$、縦座標 se に位置し、原点からの距離が $\sqrt{\text{MSE}}$ となる。$\bar{x}$ を $0.65\,\bar{x}$ に縮小すると、長さ 0.35 の水平辺が加わる代わりに垂直辺が 0.73 から 0.47 へと短縮され、斜辺は 0.59 に減少し、$\bar{x}$ と同じ精度を表す破線の円の内側に入る。標本中央値は不偏だが、円の外側に位置する。

```
0.65 * mean   0.125  0.225  0.350     0.348
     median   0.000  0.805  0.805         -
```

シミュレーション結果は公式を裏付けている。標本平均の MSE は 0.53 付近であるのに対し、縮小平均は $0.12 + 0.23 = 0.35$ に収まる。標本中央値は約 0.80 と後れをとっており、不偏であるにもかかわらず、その分散は $\bar{x}$ のおよそ 1.5 倍に達している。ここから得られる教訓は明確である。バイアスを持つ推定量が不偏推定量をはっきり上回ることがあり、不偏であることは精度の保証にはならない。

36.5.1.5 縮小平均とトレードオフ

この比較から、ただちに実用上の疑問が生じる。分散を下げる引き換えに、どれだけのバイアスを受け入れるべきなのだろうか。縮小平均 $c\,\bar{x}$ については、このトレードオフを閉じた形で解くことができる。そのバイアスと分散を Equation 36.65 に代入すると、次式が得られる。

$$\mathrm{MSE}(c) = (1-c)^2\theta^2 + c^2\frac{\sigma^2}{n}.$$

c を 0 に向けて縮小していくと、二乗バイアスが増大する一方で分散は減少する。c に関して微分し、$-2(1-c)\theta^2 + 2c\,\sigma^2/n = 0$ とおくことで、最適な縮小係数とその最小誤差が次のように求まる。

$$c^\star = \frac{\theta^2}{\theta^2 + \sigma^2/n}, \qquad \mathrm{MSE}(c^\star) = \frac{\theta^2\,\sigma^2/n}{\theta^2 + \sigma^2/n} < \frac{\sigma^2}{n}.$$

最適な縮小（shrinkage）は縮小しない標本平均を常に上回り、とりわけノイズ σ^2/n が信号 θ^2 に比べて大きいときに効果が最も大きくなることに注目しよう。これまでの数値例（$\theta = 1$、$\sigma = 4$、$n = 30$）では、この公式から先ほど選んだ係数である $c^\star = 1/1.533 \approx 0.65$ が得られ、MSE は 0.533 から 0.348 へと削減される。Figure 36.24 は、両方の誤差成分とその和を c に対してプロットしたものである。ここでは減少する放物線と増加する放物線の単純な和になっているため、典型的な U 字型曲線が厳密に現れる。

もちろん、ここには落とし穴がある。$c^\star$ は未知のパラメータ θ に依存するため、データから直接計算することはできない。さらに悪いことに、θ を知らずに固定した縮小係数を選ぶと大きな裏目に出ることがある。もし真の平均が $\theta = 10$ であったなら、同じ $c = 0.65$ を使うと -3.5 のバイアスが生じ、MSE は $12.25 + 0.23 \approx 12.5$ となって、0.533 である $\bar{x}$ より 20 倍以上悪化してしまう。0 への縮小が有効なのは、真の値が実際に 0 の近くにあるときだけである。演習問題 1 では θ の推定値を $c^\star$ に代入するとどうなるかを調べ、演習問題 9 では有名な驚くべき事実を明らかにする。3 次元以上では、同様のデータ駆動型の縮小が**あらゆる** θ の値に対して標本平均を上回るのである。

これと同じバイアス・分散のトレードオフ（bias–variance trade-off）が、教師あり予測をも支配している。任意の固定された入力 x において、無作為なデータセットで訓練されたモデルの予測 $\hat{f}(x)$ は、真の回帰関数 $f^*(x)$ の推定量である。新しいラベルに対する期待二乗誤差（squared error）は、その推定量の MSE にラベルノイズの削減不能な分散を加えたものに等しい（2.6 節）。多くのモデル族において、柔軟性を高めるとバイアスが減る一方で分散が増大し、正則化（regularization）はちょうど縮小係数 c のように働く (Geman et al. 1992)。ただし、現代の過剰パラメータ化されたモデルにおけるテスト誤差（test error）は、必ずしも単純な U 字型を描くとは限らない点に留意しよう。モデルが訓練データに完全に適合する補間閾値（interpolation threshold）を超えると、テスト誤差が再び下降することがある（36.10.3.2 節）。

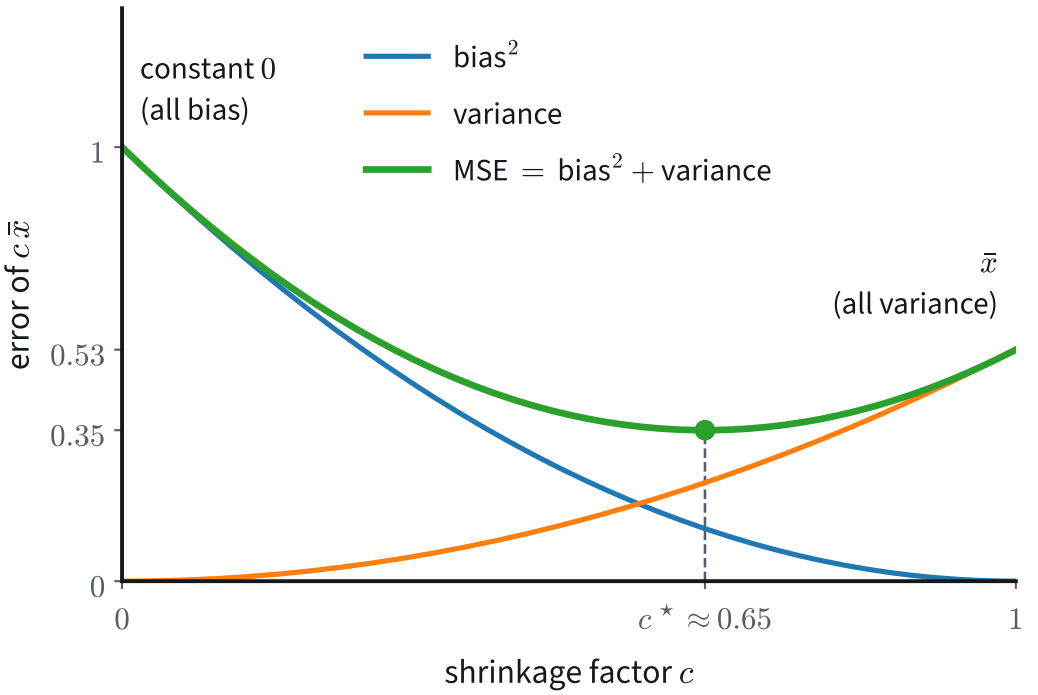


Figure 36.24: 縮小平均 $c\,\bar{x}$ の二乗バイアスと分散を c の関数として描いた図（$\theta = 1$、$\sigma = 4$、$n = 30$）。両者の和である MSE は $c^\star \approx 0.65$ で最小値に達し、0.35 まで低下する（縮小しない標本平均の 0.53、すなわち $c = 1$ との比較）。

36.5.1.6 一致性と有効性

バイアスと分散は、固定された標本サイズ n において推定量を評価するものである。それとは別の問いとして、データを際限なく集めていったときに何が起きるか、すなわち推定値が最終的に真の値へと収束するかどうかが問題になる。推定量 $\hat{\theta}_n$ が**一致推定量**（consistent estimator）であるとは、それが θ に**確率収束**（convergence in probability）することをいい、次式を意味する。

$$P(|\hat{\theta}_n - \theta| > \varepsilon) \to 0 \quad \text{as } n \to \infty, \text{ for every } \varepsilon > 0,$$

これを $\hat{\theta}_n \xrightarrow{P} \theta$ と書く。直感的には、任意の許容誤差 $\varepsilon > 0$ に対して、θ から ε を超えて外れる確率が標本の増大とともに 0 へと縮んでいくということである。

バイアス・分散分解を使えば、一致性を手早く確認できる。マルコフの不等式（36.1 節）を非負の確率変数 $(\hat{\theta}_n - \theta)^2$ と閾値 ε^2 に適用すると、大きな誤差が生じる確率を MSE によって直接上から抑えられる。

$$P(|\hat{\theta}_n - \theta| > \varepsilon) \le \frac{\operatorname{MSE}(\hat{\theta}_n)}{\varepsilon^2} = \frac{\operatorname{Bias}(\hat{\theta}_n)^2 + \operatorname{Var}(\hat{\theta}_n)}{\varepsilon^2}.$$

したがって、$n \to \infty$ のときバイアスと分散の両方が 0 に向かうならば、その推定量は一致推定量であることが保証される。どちらの条件も単独では十分でない。最初の観測値だけを使う $\hat{\theta}_n = x_1$ はすべての n で不偏だが、分散が σ^2 のまま動かないため決して収束しない。逆に $\bar{x} + 1$ は分散が $\sigma^2/n \to 0$ となるものの、誤った値へと収束してしまう。プラグイン分散 s_0^2 は好ましい折衷例を示している。あらゆる有限の n ではバイアスを持つものの、バイアス $-\sigma^2/n$ と分散（4 次モーメントが有限と仮定した場合）はいずれも n が大きくなるにつれて消えるため、一致推定量となる。

この上界を標本平均に適用すると、平均をとる操作が機能する根拠となる基本定理が得られる。

命題（大数の弱法則）。 *$x_1, \dots, x_n$ が平均 μ、有限の分散 σ^2 を持つ独立同分布の確率変数であるとき、標本平均 $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$ は次を満たす。*

$$P(|\bar{x}_n - \mu| > \varepsilon) \le \frac{\sigma^2}{n\,\varepsilon^2} \quad \text{for every } \varepsilon > 0, \qquad \text{so} \qquad \bar{x}_n \xrightarrow{P} \mu. \tag{36.66}$$

証明。 期待値の線形性より $\mathbb{E}[\bar{x}_n] = \mu$ であるから、バイアスは 0 である。各変数は独立であるため和の分散は分散の和に等しく、$\operatorname{Var}(\bar{x}_n) = n\sigma^2/n^2 = \sigma^2/n$ となる。$\operatorname{MSE} = \sigma^2/n$ を上のマルコフの上界に代入すれば目的の不等式が得られ、その右辺は $n \to \infty$ で 0 に収束する。$\blacksquare$

実際には、有限の平均 $\mathbb{E}|x_1| < \infty$ というより弱い仮定のもとでも大数の法則（law of large numbers）は成り立つが、その証明はより長くなる (Wasserman 2013)。この単純な定理は機械学習の多くの基盤を支えている。積分のモンテカルロ推定がなぜ収束するのか、大規模なテストセット上の正解率がなぜ真の正解率を反映するのか、そして平均訓練損失がなぜ期待損失を追跡するのかは、すべてこの定理から説明される。もっとも、この上界自体はかなり緩い。91.2% の正解率を持つ分類器を 1,000 問で評価する場合、テスト正解率が θ から 3 パーセントポイントを超えて外れる確率が高々 $0.0803/(1000 \cdot 0.03^2) \approx 0.09$ であることしか保証されないのに対し、ガウス近似ではその確率は 0.001 付近になる。この不等式の真の強みは一般性にあり、分散が有限であること以外には何も仮定しない。36.9 節では、裾に関してより強い仮定が成り立つときにはるかにタイトな上界を導出する。

一致性は最低限の基準にすぎず、2 つの一致推定量の優劣を区別するものではない。不偏推定量のうち、より分散の小さいものはより**有効**（efficient）であるといい、より少ない標本で同じ精度を達成できることを意味する。ガウスデータでは標本平均も標本中央値も μ に対して不偏だが、n が大きいとき中央値の分散はおよそ $\frac{\pi}{2}\,\sigma^2/n$ となる（演習問題 6）。したがって中央値が平均と同じ標準誤差を達成するには約 $\pi/2 \approx 1.57$ 倍のデータが必要となり、その効率は $2/\pi \approx 64\%$ となる。先ほどの $n = 30$ のシミュレーションでも、すでに 0.80 対 0.53 としてこの比率が示唆されていた。対照的に、標本比率はベルヌーイ分布のパラメータに対するあらゆる不偏推定量の中で最小の分散を達成しており、これはクラメール・ラオの下界（Cramér–Rao bound）として知られる基本的な下限に一致する（36.6.3.4 節）。

36.5.1.7 モンテカルロ推定

ここまでの標本は、テスト問題やノイズを含む測定値、訓練の実行結果といった外部データから得られていた。しかし大数の法則は標本がどこから来たかを問わないため、標本平均は強力な計算手法へと姿を変える。深層学習で登場する期待値のほとんど（期待損失、正規化定数（normalizing constant）、あるいはそのいずれかの勾配）は閉じた形を持たない積分であり、数値積分のグリッドが絶望的な計算量になる数千次元に及ぶことも珍しくない。グリッドを構築できない場合でも、たいていサンプリングは可能である。この発想は最初期の電子計算機にまで遡る (Metropolis and Ulam 1949)。すなわち、$\mathbb{E}[g(X)]$（ただし確率変数 X は密度 p を持つ）を近似するには、独立なサンプル $x_1, \dots, x_n$ を p から抽出し、関数値の平均をとればよい。

$$\mathbb{E}[g(X)] \approx \frac{1}{n}\sum_{i=1}^{n} g(x_i). \tag{36.67}$$

モンテカルロ推定は、評価点を等しい重み $1/n$ でランダムに選んだリーマン和（Riemann sum）とみなすことができ、密度 p が高い領域に自然と多くの点が配置される。大数の法則 Equation 36.66 により、この平均は不偏であり、$\mathbb{E}[g(X)]$ へと $n \to \infty$ で収束する。その収束率（convergence rate）は分散によって決まる。すなわち、$X_1, \dots, X_n$ が密度 p に独立同分布に従い、$\sigma^2 = \mathrm{Var}(g(X))$ が有限であれば、次式が成り立つ。

$$\mathrm{Var}\Big(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\Big) = \frac{\sigma^2}{n}. \tag{36.68}$$

数学的にはここに新しいことは何もない。大数の法則で見た標本平均の分散 σ^2/n において、$g(X_i)$ が x_i の役割を果たしているだけである（36.2 節）。注目すべきはその帰結である。二乗平均平方根（root mean square）誤差は $\sigma/\sqrt{n}$ に従って縮小する。一見するとこれは遅く、精度を 10 進法で 1 桁上げるには 100 倍のサンプルが必要になる。しかし、x の次元が指数に一切現れないこと、すなわちこの次元への非依存性こそが、高次元でサンプリングが機能する理由のすべてである。不可欠な要件は分散が有限であることただ 1 つであり、36.1 節で見たように裾が重い分布では平均が破綻し得る（演習問題 5 を参照）。

これを決定論的なグリッドと比較してみよう。グリッド間隔 ϵ を d 次元で達成するには、$N = \epsilon^{-d}$ 個の評価点が必要になる。34.6 節の左端点則の誤差は $\epsilon = N^{-1/d}$ に比例し、各セルの中心で評価する**中点則**（midpoint rule）のような 2 次精度の公式を使えば $\epsilon^2 = N^{-2/d}$ まで改善される。一方、モンテカルロ法はどの次元でも $N^{-1/2}$ のレートを保つ。1 次元では中点グリッドの圧勝であり、$d = 4$ で指数が並び、$d = 10$ になるとグリッドが $N^{-1/5}$ まで鈍化するのに対し、モンテカルロ法は依然として $N^{-1/2}$ で収束する。分子の σ は d とともに増大し得るものの、指数が悪化することは決してない（演習問題 7）。

この $1/\sqrt{n}$ のレートを具体的な積分で確かめてみよう。$\int_0^1 e^{-x^2}dx = \mathbb{E}[e^{-U^2}]$（ただし U は $[0,1]$ 上の一様分布）を考える。その厳密値は $\frac{\sqrt{\pi}}{2}\,\mathrm{erf}(1) = 0.746824$ である。以下では 10 のべき乗ごとに累積平均を追跡し、実際の誤差を予測スケール $\sigma/\sqrt{n}$ と比較する（ここで $\sigma = \mathrm{sd}(e^{-U^2}) \approx 0.2$ はサンプルから推定する）。

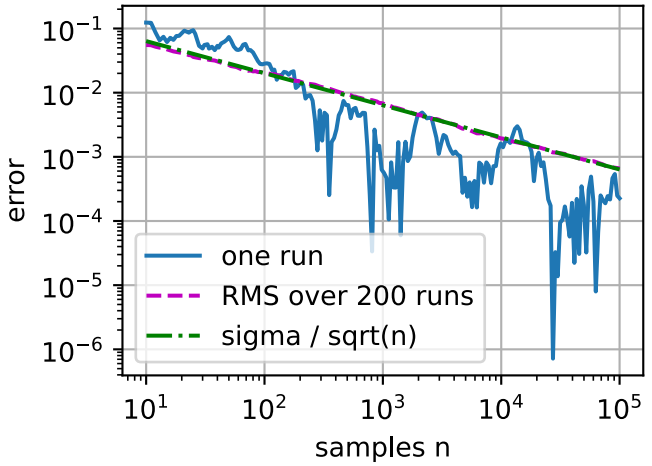
```
exact = math.sqrt(math.pi) / 2 * math.erf(1)
g = np.exp(-rng.uniform(size=10**5)**2)             # g(U), U uniform on [0, 1]
running = np.cumsum(g) / np.arange(1, 10**5 + 1)    # average of the first n
sigma = g.std()
for n in [10, 100, 1000, 10000, 100000]:
    print(f'n={n:<6}  estimate={running[n - 1]:.5f}  '
          f'error={abs(running[n - 1] - exact):.1e}  sigma/sqrt(n)={sigma / n**0.5:.1e}')
print(f'exact (sqrt(pi)/2) erf(1) = {exact:.6f}')
```

```
n=10      estimate=0.87045  error=1.2e-01  sigma/sqrt(n)=6.4e-02
n=100     estimate=0.77742  error=3.1e-02  sigma/sqrt(n)=2.0e-02
n=1000    estimate=0.74604  error=7.8e-04  sigma/sqrt(n)=6.4e-03
n=10000   estimate=0.74536  error=1.5e-03  sigma/sqrt(n)=2.0e-03
n=100000  estimate=0.74705  error=2.3e-04  sigma/sqrt(n)=6.4e-04
exact (sqrt(pi)/2) erf(1) = 0.746824
```

観測された誤差は予測スケール $\sigma/\sqrt{n}$ をよく追跡しているが、各行はあくまで 1 回のランダムな実現値にすぎない。1 回の誤差が $\sigma/\sqrt{n}$ の 2〜3 倍を超えることはまれであり、ときにはそれを大きく下回ることもある。標本サイズにわたって誤差が連続的にどう振る舞うかを見るため、対数等間隔にとった 200 個の n の値における 1 回の累積平均を、200 回の独立な実行にわたる二乗平均平方根誤差（Equation 36.68 が厳密に予測する量）と並べてプロットしてみよう。

```
n = np.unique(np.logspace(1, 5, 200).astype(int))
sq = np.zeros(len(n))
for _ in range(200):                                # 200 independent runs
    g_run = np.exp(-rng.uniform(size=10**5)**2)
    sq += (np.cumsum(g_run)[n - 1] / n - exact)**2
rms = np.sqrt(sq / 200)                             # root-mean-square error
ratio = rms * np.sqrt(n) / sigma
print(f'RMS error / (sigma/sqrt(n)) over all n: {ratio.min():.2f} to {ratio.max():.2f}')
d2l.plot(n, [np.abs(running[n - 1] - exact), rms, sigma / np.sqrt(n)],
         'samples n', 'error', xscale='log', yscale='log',
         legend=['one run', 'RMS over 200 runs', 'sigma / sqrt(n)'])
```

```
RMS error / (sigma/sqrt(n)) over all n: 0.87 to 1.08
```



2 本の曲線の違いに注目しよう。単一の累積平均は、近い n における値がほぼ同じサンプルを共有しているため理論線の上下をさまよい、誤差が 0 を横切るたびに鋭く落ち込む。対照的に、200 回の独立な実行にわたって平均した二乗平均平方根誤差は、$\sigma/\sqrt{n}$ から約 20% 以内の範囲にどの n でも収まり、4 桁の範囲すべてにわたって傾き $-\frac{1}{2}$ で着実に減少していく。

36.5.2 仮説検定

エラーバーは単一の推定値がどれほど不確かか教えてくれるが、機械学習における意思決定の大半は 2 つの推定値の比較である。新しいアーキテクチャやオプティマイザ、正則化の工夫は、本当にベースラインを上回ったのだろうか。本項では、まずシミュレーションによって比較検定をゼロから構築し、続いてその閉じた形の対応物を導出し、最後に真の改善を確実に検出するために実験がどれだけのデータを必要とするかを問う。

36.5.2.1 2 つのモデルの比較

ニューラルネットワークの訓練は本質的にランダムである。重みの初期化（weight initialization）、データのシャッフル、ドロップアウトのマスクはすべて乱数シード（random seed）によって変化するため、まったく同じ訓練スクリプトを 2 回実行してもテスト正解率はわずかに異なる。モデル A とモデル B をそれぞれ 20 個のシードで訓練し、テスト正解率を記録したとしよう。シード間のノイズが 1 回の実行結果を約 1 パーセントポイント変動させる一方で、モデル B は真に 0.8 パーセントポイント優れていると想像してみてほしい。この 40 個の数値だけから、B の方が優れていると判定できるだろうか。自然な要約量は 2 つの平均正解率の差であり、これが**検定統計量**（test statistic）の役割を果たす。

すべての仮説検定は、覆そうとする懐疑的な主張を定式化することから始まる。ここでの**帰無仮説** H_0 は、両方の正解率リストが同じ分布から生じており、2 つのモデルの平均的な性能は等しいと主張する。**対立仮説**（alternative hypothesis）H_A は、両者の分布が異なると主張する。そのうえで次のように問う。もし H_0 が真であったなら、今回観測された大きさ以上の差が生じることはどれほど驚くべきことだろうか。偶然だけではその差がめったに起こらないのであれば、H_0 を**棄却する**（reject）。そうでなければ、それを**棄却しない**（fail to reject）にとどめる。この非対称な言い回しは意図的なものである。法廷での「無罪（not guilty）」判決が潔白を証明するわけではないのと同様に、H_0 を棄却できなかったとしても 2 つのモデルが同等であることが証明されるわけではない。

36.5.2.2 並べ替え検定

観測された差が H_0 のもとで驚くべきものかどうかを判断するには、H_0 が成り立つときのその分布、すなわち**帰無分布**（null distribution）を知る必要がある。驚くべきことに、**並べ替え検定**（permutation test）を使えば、ガウス性やその他のパラメトリックな関数形を一切仮定せずに、データから直接この分布を構築できる (Fisher 1935)。理屈は単純である。もし両方のリストが本当に同じ分布から得られたのであれば、「A」と「B」というラベルには何の情報も含まれない。40 個の数値を 20 個ずつの 2 グループにどのように割り当て直しても、実際に観測された分割とまったく同じ確率で生じたはずである。ラベルをシャッフルして平均の差を再計算する操作を R 回繰り返せば、純粋な偶然が生み出す差の範囲を描き出せる。以下の補助関数は R 回のシャッフルすべてを一度にベクトル化して計算し、シャッフル後の差の絶対値が観測された差の絶対値以上になる回数を数える。

```
def perm_p(acc_a, acc_b, rng, R=2000):
    """Two-sided permutation p-value and the R shuffled gaps."""
    k = len(acc_a)
    observed = acc_b.mean() - acc_a.mean()
    pooled = np.tile(np.concatenate([acc_a, acc_b]), (R, 1))
    shuffled = rng.permuted(pooled, axis=1)    # R relabelings
    gaps = shuffled[:, k:].mean(1) - shuffled[:, :k].mean(1)
    p = (1 + (np.abs(gaps) >= abs(observed)).sum()) / (R + 1)
    return p, gaps
```

観測されたラベル割り当て自体も $R+1$ 個のシャッフルの 1 つに含めると、p **値**が次式で得られる。

$$p = \frac{1 + \#\{r : |\text{gap}_r| \geq |\text{gap}_{\text{obs}}|\}}{R+1},$$

これは H_0 のもとで、実際に観測されたものと同等以上に極端な差が観測される確率を推定している。p 値がごく小さいということは、もしラベルに意味がなければこれほど大きな差はめったに生じないことを意味する。この p 値が**両側**（two-sided）である点に注目しよう。どちらの向きであれ大きな差は H_0 に反する証拠として数えられる（もし B が A を上回るかどうかだけを問うのであれば、片側検定として観測された正の差以上のシャッフル差のみを数えることになる）。

データを見る前に**有意水準**（significance level）α（慣例的には $\alpha = 0.05$）を定め、H_0 を $p \leq \alpha$ のときに棄却すると決めておく。上の設定値を使って 1 回のシミュレーション実験を行い、この検定の動きを見てみよう。

```
k = 20                                  # training seeds per model
acc_a = rng.normal(0.850, 0.010, k)     # per-seed accuracy, model A
acc_b = rng.normal(0.858, 0.010, k)     # model B: a true gap of 0.008
p, gaps = perm_p(acc_a, acc_b, rng, R=10000)
observed = acc_b.mean() - acc_a.mean()
print(f'observed gap {observed:.4f}, permutation p-value {p:.3f}')
```

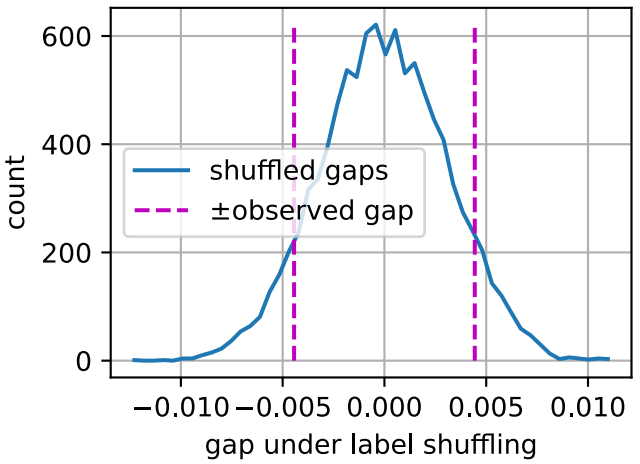
```
observed gap 0.0044, permutation p-value 0.158
```

実験のたびに新しいシードが抽出されるため、観測される差は実行ごとに真の値 0.008 の周りで変動する。大半の実行では p が 0.05 を下回って検定は H_0 を棄却するが、ここでも起こり得るように、無視できない割合の実行では棄却に至らない。1 パーセントポイント丸ごとのシードノイ

ズがある状況では、モデルあたり 20 シードでは 0.8 ポイントの改善を捉えきれないことがある。これがどれくらいの頻度で起きるかは後ほど正確に測定し、それが検定の欠陥ではなく実験の標本サイズを反映したものであることを確認する。

シャッフルされた差のヒストグラムを、観測された差（および符号を反転させた鏡像）と並べてプロットすると、帰無分布を視覚的に確認できる。

```
counts, edges = np.histogram(gaps, bins=50)
centers = (edges[:-1] + edges[1:]) / 2
top = counts.max()
d2l.plot([centers, [observed] * 2, [-observed] * 2],
         [counts, [0, top], [0, top]], 'gap under label shuffling', 'count',
         legend=['shuffled gaps', '±observed gap'],
         fmts=('-', 'm--', 'm--'))
```



並べ替え検定はデータの形状について何も仮定していないにもかかわらず、シャッフルされた差は 0 を中心とする整った釣鐘型の曲線に落ち着き、観測された差は通常その片方の裾野に位置する。何より優れているのは、これとまったく同じ手順が中央値の差、誤り率の差、検証損失の差など、**あらゆる** 2 群間の要約統計量に対してそのまま機能することである（平均を閉じた形で比較する場合、古典的な対応物はウェルチの t 検定である）。あらゆる仮説検定は同じ規律に従う。まず H_0 と H_A を明示し、α を固定して意味のある効果を検出するのに十分な標本サイズを選び、そのうえで初めてデータを収集して p 値を計算し、H_0 を $p \le \alpha$ のときに棄却するのである。

36.5.2.3 z 検定

シャッフルは帰無分布を経験的に構築するが、分散が既知のガウスデータを仮定すれば、p 値と検出率の両方について閉じた形の公式が得られる。$x_1, \ldots, x_n$ が $\mathcal{N}(\mu, \sigma^2)$（$\sigma$ は既知）に独立同分布に従うとし、$H_0 : \mu = 0$ を検定したいとする。標本平均をその標準誤差で標準化し、次の検定統計量を作る。

$$T = \frac{\sqrt{n}\,\bar{x}}{\sigma}.$$

H_0 のもとでは $\bar{x} \sim \mathcal{N}(0, \sigma^2/n)$ であるため、T は標準正規分布に従う。標準正規分布の累積分布

関数を Φ と書き、その分位点 z_q（q 分位点）を $\Phi(z_q) = q$ で定義する（たとえば $z_{0.975} \approx 1.96$）。両側 p 値は $2(1 - \Phi(|T|))$ であるから、条件 $p \le \alpha$ は $|T| \ge z_{1-\alpha/2}$ と同値である。$\alpha = 0.05$ では、H_0 が棄却されるのは標本平均が標準誤差の 1.96 倍を超えて 0 から離れたときである。Figure 36.25 はこれら両側の棄却域を示している。

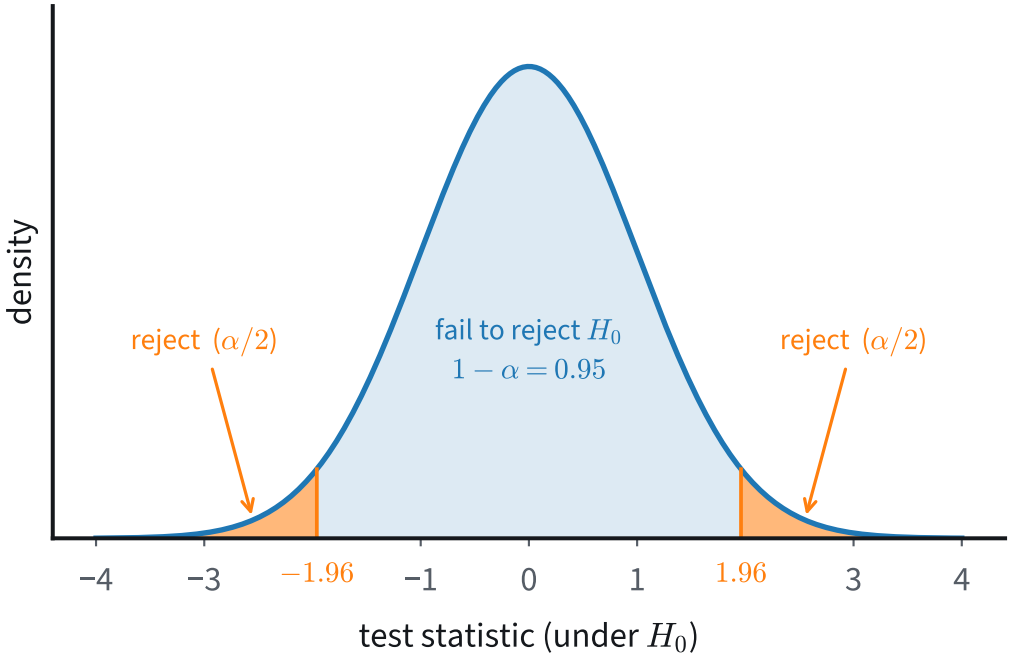


Figure 36.25: 両側 z 検定（$\alpha = 0.05$）の棄却域。H_0 のもとで標準化された統計量は標準正規分布に従い、中央の領域が確率 $1 - \alpha = 0.95$ を占め、± 1.96 を超える両裾が合わせて確率 α を占める。どちらかの裾に入る統計量が観測されることは H_0 のもとではまれであるため、H_0 を棄却する。

先ほどの 2 つのモデルの比較には、この検定の直接的な 2 標本版が対応する。モデルあたり k 個の独立なシードがあり、シードごとの標準偏差が σ であるとき、平均の差 $\bar{x}_B - \bar{x}_A$ の分散は $2\sigma^2/k$ となるため、$T = (\bar{x}_B - \bar{x}_A)/(\sigma\sqrt{2/k})$ は H_0 のもとで標準正規分布に従う。$\sigma = 0.01$ と $k = 20$ を代入すると、差の標準誤差は 0.0032 となる。したがって 0.008 の真の改善は 0 より標準誤差の 2.5 倍だけ上に位置し、1.96 の閾値は超えているものの、余裕のあるマージンとはいえない。

36.5.2.4 2 種類の過誤と検出力

あらゆる仮説検定には 2 通りの誤り方があり、優れた実験計画はその両方を制御下に置く。**第一種の過誤**（type I error、偽陽性）は、真である H_0 を棄却し、存在しない改善を主張してしまうときに生じる。**第二種の過誤**（type II error、偽陰性）は、偽である H_0 を棄却できず、本物の改善を見逃してしまうときに生じる。それぞれの確率は次式で表される。

$$\alpha = P(\text{reject } H_0 \mid H_0 \text{ true}), \qquad \beta = P(\text{fail to reject } H_0 \mid H_0 \text{ false}),$$

ここで β は、0.8 パーセントポイントの真の差といった特定の対立仮説のもとで評価される。余

事象の確率 $1-\beta$ は検定の**検出力**（power、統計的検出力）と呼ばれ、実験がその真の効果を実際に検出できる確率を表す。Figure 36.26 に起こり得る 4 つの結果をまとめる。

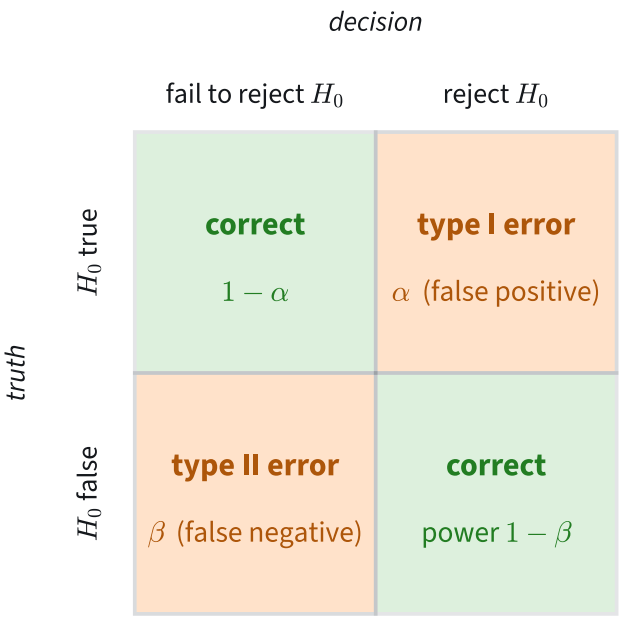


Figure 36.26: 仮説検定の 2×2 決定行列。対角成分が正しい判定、非対角成分が第一種の過誤（確率 α）と第二種の過誤（確率 β）を表し、正しく棄却するセルが統計的検出力（$1-\beta$）を表す。

「$p \le \alpha$ のときに棄却する」という単純な規則によって、なぜ第一種の過誤の確率が厳密に α に保証されるのだろうか。それは H_0 のもとで、p 値が $[0,1]$ 上の一様分布に従うからである。この事実は証明しておく価値がある。有意水準の意味を説明するだけでなく、後で扱う多重検定を理解する鍵にもなるからである。

命題（帰無仮説のもとでの p 値の一様性）。検定統計量 T が連続な累積分布関数 F を H_0 のもとで持つとし、$p = 1 - F(T)$（片側）または $p = 2\min\{F(T), 1-F(T)\}$（両側）とおく。このとき、$P_{H_0}(p \le \alpha) = \alpha$ がすべての $\alpha \in [0,1]$ に対して成り立つ。

証明。 まず $U = F(T)$ が $[0,1]$ 上の一様分布に従うことを示す。任意の $u \in (0,1)$ に対し、t_u を t のうち $F(t) = u$ を満たす最大の値とする。このような値が存在するのは、F が連続であり 0 から 1 まで動くからである。F は非減少であるため、$F(T) \le u$ が成り立つのは $T \le t_u$ のとき、かつそのときに限り、したがって $P(U \le u) = F(t_u) = u$ が得られる。片側 p 値については、ただちに $P(1-U \le \alpha) = P(U \ge 1-\alpha) = \alpha$ が従う。両側 p 値については、事象 $2\min\{U, 1-U\} \le \alpha$ は $U \le \alpha/2$ または $U \ge 1-\alpha/2$ を意味する。互いに排反なこれら 2 つの事象はそれぞれ確率 $\alpha/2$ を持つため、その和事象の確率は α となる。$\blacksquare$

並べ替え検定の p 値については、さらに単純な数え上げの議論から同じ保証が得られる。H_0 のもとでは、観測されたラベル割り当てと R 回のランダムなシャッフルは完全に交換可能（exchangeable）であるため、観測された差は $R+1$ 通りの順位のどれにも等確率で入る。したがって $p \le \alpha$ が起きるのは、観測された差の絶対値が上位 $\lfloor \alpha(R+1) \rfloor$ 個以内に入るときに限られ、$P_{H_0}(p \le \alpha) \le \alpha$ が成り立つ（同順位がある場合は p が大きくなる方向にしか働かない）。分子と分母の $+1$ が重要なのはこのためであり、有意水準 α の検定が真の帰無仮説を棄却する割

合が高々 α に抑えられることを保証している。

では、p 値の分布は H_0 が偽であるときどうなるだろうか。対比を見るため、2,000 回の 2 標本 z 検定（モデルあたり 20 シード）を、まず真の差がない場合、次に真の差が 0.008 である場合についてシミュレーションし、得られる p 値を 10 個の等幅ビンでヒストグラムにしてみよう。

```
se = 0.010 * np.sqrt(2 / k)                 # se of the gap, k seeds per model
for gap in (0.0, 0.008):
    acc_a = rng.normal(0.850, 0.010, (2000, k))
    acc_b = rng.normal(0.850 + gap, 0.010, (2000, k))
    T = (acc_b.mean(axis=1) - acc_a.mean(axis=1)) / se
    p = 2 * norm.sf(np.abs(T))               # two-sided z-test p-values
    hist = np.histogram(p, bins=10, range=(0, 1))[0] / 2000
    print(f'gap {gap:.3f}:', ' '.join(f'{h:.2f}' for h in hist),
          f'  p <= 0.05: {np.mean(p <= 0.05):.2f}')
```

```
gap 0.000: 0.10 0.10 0.11 0.10 0.09 0.10 0.10 0.10 0.10 0.10   p <= 0.05: 0.05
gap 0.008: 0.82 0.08 0.04 0.02 0.01 0.01 0.01 0.01 0.01 0.01   p <= 0.05: 0.72
```

H_0 が成り立つとき（差 0.000）、ヒストグラムは平坦になる。p 値の約 10 分の 1 ずつが各ビンに入り、約 5% が 0.05 以下に収まる。一方、0.008 の真の差があるときには、p 値は 0 の近くに鋭く集中し、約 80% が 0.1 未満に入り、70%〜74% が 0.05 以下に入る。この棄却割合が検定の検出力であり、以下で導く閉じた形の公式からは 0.72 と予測される。

その公式を導出する前に、p 値に関する 2 つの典型的な誤解に注意しておこう (Wasserstein and Lazar 2016)。第 1 に、p 値は**仮に H_0 が真であるとしたときに**これほど極端なデータが観測される確率であり、H_0 が真である確率では**ない**（後者を求めるには事前分布（prior）とベイズの定理が必要になる）。第 2 に、大きな p 値は H_0 が成り立つことを証明するものではなく、単に今回の実験では効果を検出できなかった（おそらく検出力が不足していた）ことを示すにすぎない。

1 標本 z 検定について、この検出力を陽に計算してみよう。真の平均が $\mu \neq 0$ のとき、統計量 T は平均 $\sqrt{n}\,\mu/\sigma$、分散 1 の正規分布に従う。**標準化効果量**（standardized effect）$\delta = |\mu|/\sigma$（ノイズを単位として測った効果量）を定義すると、検定は $|T| \geq z_{1-\alpha/2}$ のときに棄却し、その確率は次式で近似される。

$$1 - \beta \approx \Phi\big(\delta\sqrt{n} - z_{1-\alpha/2}\big),$$

ここでは無視できるほど小さい反対側の裾 $\Phi(-\delta\sqrt{n} - z_{1-\alpha/2})$ を落としている。このように、検出力は効果量 δ および $\sqrt{n}$ とともに上昇する。大きな効果 $\delta = 1$ に対しては、わずか $n = 8$ 個の観測値でも検出力は $\Phi(2.83 - 1.96) \approx 0.81$ に達する。

2 標本のモデル比較では、統計量の平均は $\delta\sqrt{k/2}$（ただし $\delta = 0.008/0.010 = 0.8$）となるため、$z$ 検定の検出力はモデルあたり k シードのとき $\Phi(0.8\sqrt{k/2} - 1.96)$ となり、約 0.72（20 シードの場合）、あるいはわずか 0.24（5 シードの場合）となる。この公式は σ が既知であることを前提としているため、事前に何も推定しない並べ替え検定が各シード数において 400 回の新たなシミュレーション実験でどの程度の検出力を示すか確かめてみよう。

```
def power(k, gap, rng, num_experiments=400):
    """Fraction of fresh experiments in which the permutation test rejects."""
    rejections = 0
    for _ in range(num_experiments):
```

```
            acc_a = rng.normal(0.850, 0.010, k)
            acc_b = rng.normal(0.850 + gap, 0.010, k)
            rejections += perm_p(acc_a, acc_b, rng)[0] <= 0.05
        return rejections / num_experiments

    for k in (20, 5):
        print(f'{k:2d} seeds: rejects in {power(k, 0.008, rng):.2f} of runs, '
              f'formula {norm.cdf(0.8 * np.sqrt(k / 2) - 1.96):.2f}')
```

```
20 seeds: rejects in 0.70 of runs, formula 0.72
 5 seeds: rejects in 0.16 of runs, formula 0.24
```

400 回の実行にわたって、並べ替え検定が棄却するのは約 70%（20 シードの場合）であり、わずか約 20%（5 シードの場合）にとどまる（それぞれの割合は実行ごとに約 2 パーセントポイント変動する）。どちらの数値も z 検定の公式をわずかに下回っているが、これは並べ替え検定が実質的にノイズをデータから推定しており、その分だけ検出力がわずかに下がるためである。つまり 20 シードを使ったとしても、本物の 0.8 ポイントの改善を 10 回中約 3 回は見逃してしまい、わずか 5 シードでは 5 回中 4 回も見逃すことになる。検出力が 0.2 しかない実験で棄却できなかったとしても、そこからはほとんど何もわからない。実務では通常、少なくとも 0.8 の検出力を目指す。

36.5.2.5 標本サイズ

事後になって実験の検出力不足に気づくのではなく、あらかじめ標本サイズ（sample size）を設計しておくことができる。検出力の公式を $1-\beta$ に等しいとおくと $\delta\sqrt{n} - z_{1-\alpha/2} = z_{1-\beta}$ となり、これを n について解けば、標準化効果量 δ を有意水準 α、検出力 $1-\beta$ で検出するために必要な標本サイズが求まる。

$$n = \frac{\left(z_{1-\alpha/2} + z_{1-\beta}\right)^2}{\delta^2}. \tag{36.69}$$

$1/\delta^2$ という急激なスケーリングに注目しよう。検出したい効果量を半分にすると、4 倍のデータが必要になる。標準的な基準である $\alpha = 0.05$、検出力 0.8 では $z_{0.975} \approx 1.960$ および $z_{0.8} \approx 0.842$ であるため、使いやすい経験則として $n \approx 7.85/\delta^2$ に簡略化される。Figure 36.27 に効果量ごとの検出力曲線を示す。

各群 k 個の観測値からなり共通の標準偏差 σ を持つ独立な 2 群を比較する場合、差の分散は 2 倍の $\mathrm{Var}(\bar{x}_B - \bar{x}_A) = 2\sigma^2/k$ になる。それに伴い、1 群あたりに必要な標本サイズも 2 倍になる。

$$k = \frac{2\left(z_{1-\alpha/2} + z_{1-\beta}\right)^2}{\delta^2}, \qquad \delta = \frac{|\mu_B - \mu_A|}{\sigma}. \tag{36.70}$$

$\delta = 0.8$ であった先ほどのモデル比較では、80% の検出力に到達するにはモデルあたり $k = 2 \cdot 7.85/0.64 \approx 25$ シードが必要であり、20 シードではわずかに届かなかった理由がここから説明される。

36.5.2.6 対応のある比較

ここで冒頭の問いに戻ろう。912 問と 921 問（1,000 問中）に正解した 2 つの分類器の比較である。先ほど見たように、両者のスコアを独立なものとして扱うと差はノイズに埋もれてしまった。しかし決定的に重要なのは、両方のモデルが**同じ** 1,000 問に解答していることである。問題の中

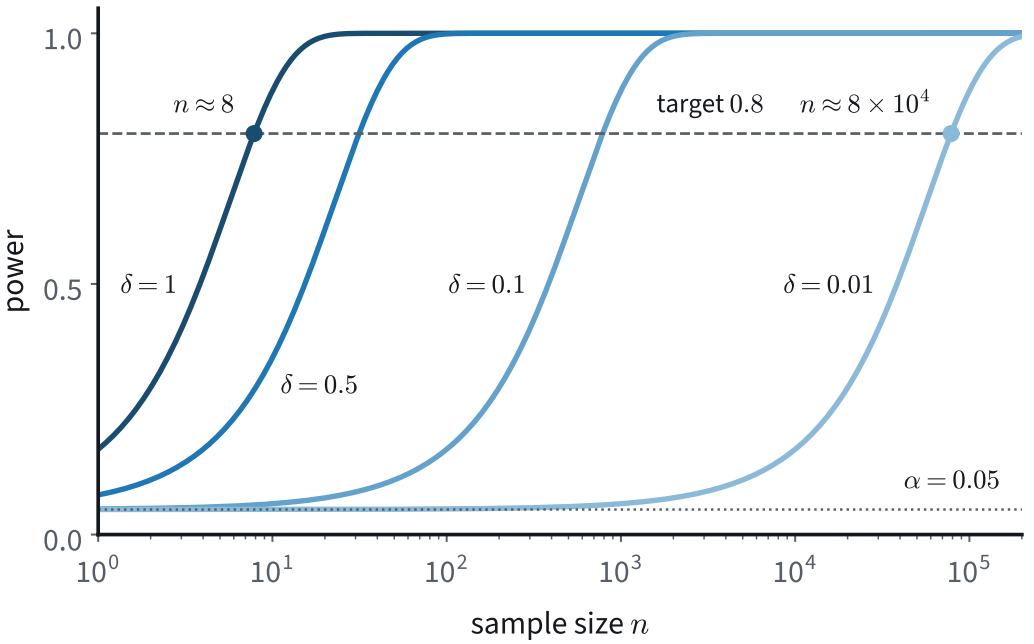


Figure 36.27: 両側 z 検定（$\alpha = 0.05$）の検出力と標本サイズ n の関係（曲線は標準化効果量 δ ごとに描かれている）。各曲線は偶然だけで棄却が起きる α（点線）付近から始まり、1 に向かって上昇する。破線は目標検出力 0.8 を示しており、到達点は $n \propto 1/\delta^2$ に従う。約 8 個の観測値で到達する $\delta = 1$ から、約 78,500 個を要する $\delta = 0.01$ まで大きく変化する。

には両モデルにとって簡単なものもあれば難しいものもあり、対応のない（unpaired）分析ではそうした共有される問題難易度のばらつきまですべてノイズに混ぜ込んでしまう。

対応のある（paired）比較は、問題ごとの差 $d_i \in \{-1, 0, 1\}$（問題 i におけるモデル B の正誤からモデル A の正誤を引いた値）を直接扱うことで、その共有変動を取り除く。両モデルの判定が一致する問題では常に $d_i = 0$ となる。いま 41 問で判定が食い違い、モデル B のみが正解した問題が 25 問、モデル A のみが正解した問題が 16 問で、差し引き $+9$ 問（0.9 パーセントポイント）のリードだったとしよう。2 つのモデルが交換可能であるという帰無仮説のもとでは、各不一致は公平なコイン投げのようなものであり、A と B のどちらに有利になる確率も等しい。したがって適切な帰無シミュレーションは、非ゼロの各 d_i の符号をランダムに反転させる手続きとなり、これを**符号反転並べ替え検定**（sign-flip permutation test）と呼ぶ。この 41 個の不一致に対して実行してみよう。

```
d = np.r_[np.ones(25), -np.ones(16), np.zeros(959)]   # B only, A only, agree
nonzero = d[d != 0]                                    # agreements never change
signs = rng.choice([-1, 1], size=(100000, len(nonzero)))
flipped = (signs * nonzero).sum(axis=1) / len(d)       # gaps under H_0
p = (1 + (np.abs(flipped) >= abs(d.mean())).sum()) / (len(flipped) + 1)
print(f'gap {d.mean():.3f}, sign-flip p-value {p:.2f}')
```

```
gap 0.009, sign-flip p-value 0.21
```

符号反転検定は $p \approx 0.21$ を返す（ランダムな符号反転を厳密な二項確率に置き換えればマクネマー検定になる）。まったく同じ集計値に対する対応のない分析では、差が標準誤差のわずか 0.7 倍にとどまり $p \approx 0.47$ となったのと比べてほしい。対応づけによって差の標準誤差は 0.012 から $\sqrt{41}/1000 \approx 0.006$ へと半減する。両モデルが一致した 959 問がもはやノイズを持ち込まないからである。それでもなお、25 対 16 という 41 問の不一致の内訳は確証を得るには小さすぎる。1,000 問のテストにおける 9 問差のマージンだけでは、モデル B の方が優れているとは結論づけられない。

対応づけが最も威力を発揮するのは、評価をゼロから設計するときである。2 つのモデルの真の正解率が 90% と 91% であり、問題の 4% で判定が食い違うとしよう（B のみ正解が 2.5%、A のみ正解が 1.5%）。このとき d_i は平均 0.01、分散 $\mathbb{E}[d_i^2] - 0.01^2 = 0.04 - 0.0001 = 0.0399$ を持ち、標準化効果量は $\delta = 0.01/\sqrt{0.0399} \approx 0.050$ となる。Equation 36.69 より、対応のある検定は 80% の検出力におよそ $7.85 \cdot 0.0399/0.01^2 \approx 3{,}100$ 問で到達する。対照的に、2 つのモデルを別々の独立なテストセットで評価すると、問題ペアあたりの対応のない分散は $0.9 \cdot 0.1 + 0.91 \cdot 0.09 = 0.1719$ のまま残り、およそ $7.85 \cdot 0.1719/0.01^2 \approx 13{,}500$ 問が**モデルごとに**必要になる。

この例では、対応づけによって各テストセットは 4 分の 1 以下に縮小し、ラベル付けの総労力は 9 分の 1 近くにまで削減される。しかも 2 つのモデルの一致率が高くなるほど、この節約効果はさらに大きくなる。実践上の原則は明快である。競合するモデルは常に同じテストセット上で評価し、問題ごとの差を分析すべきである。同じ原理は乱数シードにも当てはまる。2 つのモデルが同じ訓練シードを共有している場合は、シードごとの差の符号を反転させて検定すればよい（演習問題 5）。

36.5.2.7 多重検定

ここまでの有意性の保証 $P(\text{reject} \mid H_0) = \alpha$ はすべて、事前に選んだ**単一の**検定を実行することを前提としている。しかし機械学習では、ハイパーパラメータのスイープによって一度に何十もの比較を行うのが日常茶飯事である。実際にはベースラインより優れたものが 1 つもない状況で、20 個の新しい設定をベースラインと比較し、それぞれを $\alpha = 0.05$ で検定したとしよう。各検定

が独立であれば、少なくとも 1 つの設定が誤って有意に見えてしまう確率は $1 - 0.95^{20} \approx 0.64$ に達する。勝った設定だけをつまみ食いしたり、$p \le \alpha$ になるまで分析に繰り返し手を加えたりする行為（p **ハッキング**）は、純粋なノイズを論文に載せられそうな見かけ上の改善へと確実に変えてしまう。

最も単純な防御策は**ボンフェローニ補正**（Bonferroni correction）である (Bonferroni 1936)。m 回の検定を行うとき、個々の閾値を α/m に引き下げる。A_j を検定 j が真の帰無仮説を誤って棄却する事象とすると、ユニオンバウンド（union bound）から次式が得られる。

$$P\Big(\bigcup_{j=1}^{m} A_j\Big) \le \sum_{j=1}^{m} P(A_j) = m \cdot \frac{\alpha}{m} = \alpha.$$

この上界がいかに一般的であるかに注目しよう。**1 つでも**偽発見が生じる確率を α 以下に抑えるにあたり、m 個の検定がどのように相関しているかについて一切の仮定を置いていない。20 個の独立な検定を $0.05/20$ で実行すれば、少なくとも 1 つの偽陽性が生じる確率は 0.64 から $1 - (1 - 0.0025)^{20} \approx 0.049$ へと低下する。

残念ながら、m が数百や数千に達すると、m で割ったボンフェローニの閾値は厳しくなりすぎて、本物の発見まで埋もれてしまう。大規模なスクリーニングでは、統計学者は代わりに**偽発見率**（false discovery rate）、すなわち棄却した仮説のうち偽陽性が占める割合の期待値を制御する。ベンジャミニ・ホッホバーグ法 (Benjamini and Hochberg 1995) は、この率を α 以下に保つ（p 値が互いに独立であるか正の従属性を持つ場合）。演習問題 2 でこの規則を導入し、シミュレーションで検証する。

36.5.3 信頼区間

p 値は、単一の候補パラメータ値に関する二択の問いに答えるものである。しかし実務では通常、それ以上の情報、すなわちデータと整合するパラメータ値の全範囲を、推定値の周りのエラーバーとして直接示したい。

36.5.3.1 定義と解釈

信頼区間は、反復実験のうち指定された割合で、固定された未知のパラメータを確実に捕捉する規則によって生成される (Neyman 1937)。形式的には、$1 - \alpha$ 信頼区間とはデータから計算されるランダムな区間 C_n であって、次式を満たすものである。

$$P_\theta(C_n \ni \theta) \ge 1 - \alpha \quad \text{for every } \theta. \tag{36.71}$$

ランダム性がどこにあるかを強調するため、ここではあえて $C_n \ni \theta$ と書き、$\theta \in C_n$ とは書かない。θ は固定された数値であるのに対し、区間 C_n はデータセットごとに移動する。確率 $1 - \alpha$ は**手続き**の性質である。多数の独立な実験にわたって、得られる 95% 信頼区間のうち少なくとも 95% が θ を被覆する。1 回の計算で得られた個々の区間は θ を含んでいるか外しているかのどちらかであり、その区間自体にはどちらが起きたかを示す手がかりは一切ない。Figure 36.28 は、100 個の独立標本から構成した 100 個の区間によってこの様子を示している。実務で使われる多くの区間は、大標本極限においてのみこの保証を満たす。すなわち、**漸近**（asymptotic）$1 - \alpha$ 区間とは、$\liminf_{n\to\infty} P_\theta(C_n \ni \theta) \ge 1 - \alpha$ をすべての θ に対して満たすものである。

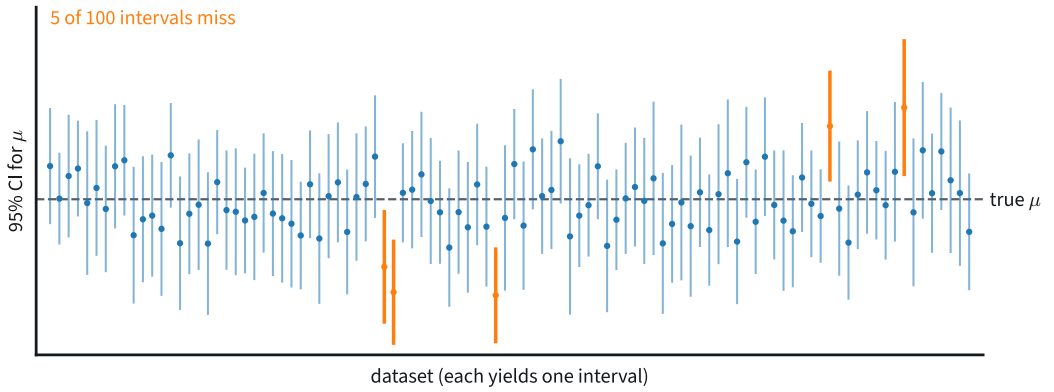


Figure 36.28: それぞれ独立標本から構成された 100 個の 95% 信頼区間を、固定された真の平均（破線）に対してプロットした図。大半の区間は真の値を被覆しているが、オレンジ色の区間は外れている。95% という保証はこの手続きの長期的なヒット率を表すものであり、個々の区間からは成功したかどうかを知ることはできない。

36.5.3.2 ガウス信頼区間

すべての段階が厳密に成り立つ最も明快な場合から始めよう。$x_1, \dots, x_n$ が $\mathcal{N}(\mu, \sigma^2)$（$\sigma$ は既知）に独立同分布に従うとする。$\bar{x} \sim \mathcal{N}(\mu, \sigma^2/n)$ であるから、半幅を $h = z_{1-\alpha/2}\,\sigma/\sqrt{n}$ とおけば、$\bar{x}$ が幅 h 以内で μ の周りに収まる確率は $1-\alpha$ となる。ここでこの不等式を裏返し、μ に関する命題として読み替えてみよう。

$$|\bar{x} - \mu| \le h \quad \Longleftrightarrow \quad \mu \in [\bar{x} - h,\, \bar{x} + h].$$

ゆえに、$P_\mu(\mu \in [\bar{x}-h, \bar{x}+h]) = 1-\alpha$ がすべての μ に対して成り立つ。この単純な代数的反転こそが、信頼区間の背後にある考え方のすべてである。$\alpha = 0.05$ では、この区間は $\bar{x} \pm 1.96\,\sigma/\sqrt{n}$ となる。

現実の問題では σ が既知であることはまれなため、標本標準偏差 s を Equation 36.63 から代入する。s をデータから推定すると、スチューデント化された統計量（studentized statistic）$T = (\bar{x}-\mu)/(s/\sqrt{n})$ はもはや正規分布には従わない。ガウスデータの場合、これはスチューデントの t 分布（自由度 $n-1$）に従う (Student (Gosset 1908))。この分布は、正規分布よりも重い裾を持つことで s に含まれる追加のランダム性を織り込んでいる（ここではこの分布を事実として認めるが、演習問題 4 ではランダムな精度を持つガウス分布として導出する）。$z_{1-\alpha/2}$ を t 分位点 $t_{n-1,1-\alpha/2}$ に置き換えると、次の区間が得られる。

$$\left[\bar{x} - t_{n-1,1-\alpha/2}\,\frac{s}{\sqrt{n}},\; \bar{x} + t_{n-1,1-\alpha/2}\,\frac{s}{\sqrt{n}}\right], \tag{36.72}$$

これはガウスデータに対して、被覆率（coverage）$1-\alpha$ をすべての $n \ge 2$ で達成する。標本サイズが小さいとき、この違いは極めて大きい。$t_{n-1,0.975}$ は 12.71（$n=2$）、2.776（$n=5$）、1.984（$n=100$）となり、1.96 に向かって n の増大とともに収束していく。応用分野全体で定番として使われている区間は、単純に大標本極限を代入した次式である。

$$\left[\bar{x} - 1.96\,\frac{s}{\sqrt{n}},\ \bar{x} + 1.96\,\frac{s}{\sqrt{n}}\right], \tag{36.73}$$

これは漸近 95% 区間であり、あらゆる有限の n においてわずかに狭すぎる。ガウスデータに対する実際の被覆率は $P(|t_{n-1}| \le 1.96)$ であり、$n = 100$ では公称値に近い（0.947）ものの、0.878 まで低下する（$n = 5$ の場合）。

データがガウス分布に従わない場合はどうなるだろうか。中心極限定理（36.3.2.3 節）により、分散が有限である限り $\sqrt{n}(\bar{x} - \mu)/\sigma$ は標準正規分布へ収束し、s も大数の法則により σ へ収束する。したがって、Equation 36.73 は有限の分散を持つ**あらゆる**分布の平均に対する漸近 95% 区間であり続ける。n がどれほど大きければ十分かはデータの形状に依存する。歪んだ分布ではより多くのサンプルが必要になり（演習問題 3）、100% に近い正解率では多くのテストセットで満点が出て標本分散が 0 になるため、はるかに多くのサンプルを要する（演習問題 4）。いずれの場合も半幅は $1/\sqrt{n}$ に比例するため、エラーバーを半分に縮めるには 4 倍のデータが必要になる。

テスト正解率は 0/1 の指示変数の平均であり $s^2 \approx \hat{\theta}(1 - \hat{\theta})$ が成り立つため、同じ公式から $\hat{\theta} \pm 1.96\sqrt{\hat{\theta}(1 - \hat{\theta})/n}$ が得られる。これは**ワルド区間**（Wald interval）として知られている。これを 1,000 問の分類器と、繰り返し登場しているコインの例（9 回の表が 13 回のコイン投げで出た例）について計算してみよう。

```
for n_correct, n in ((912, 1000), (9, 13)):
    theta_hat = n_correct / n
    half = 1.96 * np.sqrt(theta_hat * (1 - theta_hat) / n)
    print(f'{n_correct:3d} of {n:4d}: '
          f'({theta_hat - half:.3f}, {theta_hat + half:.3f})')
```

```
912 of 1000: (0.894, 0.930)
  9 of   13: (0.441, 0.943)
```

分類器については $91.2\% \pm 1.8$ パーセントポイントが得られる。コインについては区間 $(0.44, 0.94)$ が余裕をもって $\frac{1}{2}$ を含んでおり、13 回のコイン投げではコインに偏り（バイアス）があると結論づけるにはまったく足りないことが確認できる。その偏りを ± 0.05 の幅まで絞り込むには、およそ $1.96^2 \cdot 0.69 \cdot 0.31 / 0.05^2 \approx 330$ 回のコイン投げが必要になる。

これを同じコインに対するベイズ的な**信用区間**（credible interval）と対比すると示唆に富む。ベイズ分析では θ 自体を確率変数として扱い、事前分布を置いたうえで、観測されたコイン投げの結果が与えられたときの事後確率（posterior probability）$1 - \alpha$ を含む区間を報告する。一様事前分布のもとでは事後分布（posterior）は $\mathrm{Beta}(10, 5)$ となり（36.4 節）、中央の 95% 信用区間（事後分布の 2.5% 分位点と 97.5% 分位点の間）は $(0.42, 0.87)$ となる。ここでは数値がよく似ているものの、2 つの区間が主張している内容は根本的に異なる。信用区間は、選んだ事前分布のもとで θ が $(0.42, 0.87)$ に入る確率が 0.95 であると述べる。一方、信頼区間は、その構成手続きが固定された真の θ を反復実験の少なくとも 95% で捕捉すると述べており、これは θ が何であるかによらない。また、ワルド区間がその保証を満たすのは漸近的にのみである点にも留意しよう。13 回のコイン投げでは、実際の被覆率は約 91%（$\theta = \frac{1}{2}$ のとき）であり、50% 未満にまで落ち込むこともある（$\theta = 0.05$ のとき）。

被覆率は反復実験にわたる頻度であるため、シミュレーションによって直接検証できる。1,000 個の独立なデータセット（それぞれ $n = 100$ 個の標準正規分布からの観測値）を抽出し、各データセット上で漸近区間 Equation 36.73 を計算して、真の平均 0 を被覆する個数を数えてみよう。

```
mu, n, trials = 0.0, 100, 1000
data = rng.normal(mu, 1.0, (trials, n))          # one row = one data set
mu_hat = data.mean(axis=1)
se = data.std(axis=1, ddof=1) / np.sqrt(n)
hits = np.abs(mu_hat - mu) <= 1.96 * se          # does the interval contain mu?
print(f'first interval ({mu_hat[0] - 1.96 * se[0]:.3f}, '
      f'{mu_hat[0] + 1.96 * se[0]:.3f}) contains 0: {hits[0]}')
print(f'{hits.sum()} of {trials} intervals contain the true mean '
      f'(nominal 950, exact 947)')
```

```
first interval (-0.299, 0.098) contains 0: True
951 of 1000 intervals contain the true mean (nominal 950, exact 947)
```

最初の区間は実行の約 95% で 0 を被覆し、被覆した合計回数は 947（通常 ± 14 以内）を中心として分布し、950 にはならない。この 3 回の不足はランダムなノイズではない。$n = 100$ における厳密な t 分位点は $t_{99,0.975} = 1.984$ であり 1.96 ではないため、1.96 を使った区間の真の被覆率は $P(|t_{99}| \le 1.96) = 0.947$ となるのである。± 14 という 947 の周りのばらつきは、単に二項分布の標準偏差の 2 倍、すなわち $2\sqrt{1000 \cdot 0.947 \cdot 0.053} \approx 14$ にすぎない。これと同じ手順によって、k 個の訓練シードにわたる平均正解率のエラーバーも求められる。シード間の標準偏差を $\sqrt{k}$ で割り、適切な分位点を掛ければよい。

36.5.3.3 検定と信頼区間

1.96 という数値がちょうど 2 回登場したことに気づいただろうか。1 回目は z 検定の棄却閾値として、2 回目はガウス信頼区間の乗数としてである。これは偶然ではなく、検定と信頼区間はまさに同じコインの裏表なのである。

σ が既知のガウスデータを考えよう。両側 z 検定（帰無仮説 $H_0 : \mu = \mu_0$）が H_0 を棄却しないのは、まさに $|T| \le z_{1-\alpha/2}$ のときである（ただし $T = \sqrt{n}(\bar{x} - \mu_0)/\sigma$）。この不等式を変形すると $|\bar{x} - \mu_0| \le h$（ただし $h = z_{1-\alpha/2}\,\sigma/\sqrt{n}$）となり、これは単に $\mu_0 \in [\bar{x} - h,\, \bar{x} + h]$ を意味する。要するに、**95% 信頼区間とは、5% 水準の仮説検定が棄却せずに残す帰無仮説値の集合である**。その被覆保証も、検定の第一種の過誤の確率からただちに従う。すなわち、すべての μ に対して $P_\mu(\mu \in [\bar{x} - h, \bar{x} + h]) = 1 - P_\mu(\text{the test of } H_0 : \mu \text{ rejects}) = 1 - \alpha$ が成り立つ。

この双対性（duality）がこれほど有用なのは、それが**あらゆる**仮説検定の族に対して成り立つからである。すべての候補値 θ_0 に対して有意水準 α の検定（帰無仮説 $H_0 : \theta = \theta_0$）が与えられていれば、検定が棄却しないすべての値 θ_0 を集めることで、同じ 1 行の議論により自動的に $1 - \alpha$ 信頼集合（confidence set）が得られる。陽な公式が使えなかったり不自然な振る舞いをしたりするときには、代わりに検定を反転させればよい。たとえば、あるモデルが 29 問（30 問中）に正解した場合、ワルド区間は不合理にも正解率 1 を超えてはみ出してしまうが、厳密な二項検定を反転させれば $[0, 1]$ の内部に収まる妥当な区間が得られる（演習問題 4）。

またこの双対性は、2 つのモデルを比較する際のより良い習慣も示唆している。$\bar{x}_B - \bar{x}_A \pm 1.96\,\sigma\sqrt{2/k}$（$k$ シードにわたる場合）のように、差に対する信頼区間を報告するのである。この区間が 0 を含まないことは仮説検定が棄却することと同値であると同時に、実際の改善幅がどの程度であり得るかも一目でわかる。

36.5.3.4 デルタ法

実務では $\hat{\theta}$ そのものではなく、その滑らかな関数 $g(\hat{\theta})$（正解率の対数オッズ（log odds）、2 つの損失の比、あるいは相対誤差削減率など）に関心があることも多い。エラーバーを g を通してど

のように伝播させればよいだろうか。g が θ で微分可能であるとき、1 次のテイラー展開（Taylor expansion）から次式が得られる。

$$g(\hat{\theta}) \approx g(\theta) + g'(\theta)\,(\hat{\theta} - \theta).$$

θ の近傍では、関数 g は小さなずれを単にその傾き $|g'(\theta)|$ 倍に拡大・縮小する。標準偏差をとり、$\hat{\theta}$ を未知の θ の代わりに g' へ代入すると、**デルタ法**が得られる (Wasserman 2013)。

$$\mathrm{se}(g(\hat{\theta})) \approx \left|g'(\hat{\theta})\right| \mathrm{se}(\hat{\theta}). \tag{36.74}$$

具体的な数値で試してみよう。正解率が $\hat{\theta} = 0.90$（$n = 1{,}000$ 問）であるとき、標準誤差は $\mathrm{se}(\hat{\theta}) = \sqrt{0.9 \cdot 0.1 / 1000} \approx 0.0095$ である。その対数オッズ $g(\hat{\theta}) = \log \frac{\hat{\theta}}{1-\hat{\theta}} \approx 2.20$ の導関数（derivative）は $g'(\hat{\theta}) = 1/(\hat{\theta}(1-\hat{\theta})) \approx 11.1$ となるため、対数オッズの標準誤差はおよそ $11.1 \cdot 0.0095 \approx 0.105$ となる。

同じ線形化は、推定値のベクトル $\hat{\theta}$（共分散行列（covariance matrix）Σ を持つ）へもそのまま拡張でき、$\mathrm{Var}(g(\hat{\theta})) \approx \nabla g^{\top} \Sigma \, \nabla g$ となる。Σ の非対角成分である共分散（covariance）が重要である点に注意しよう。同じテストセット上で測定した 2 つの損失の比をとる場合、分子と分母は連動して変動するからである。

他のあらゆる線形近似と同様に、デルタ法にも明確な限界がある。第 1 に $g'(\theta) \neq 0$ が必要である。$g(\theta) = (\theta - \tfrac{1}{2})^2$ を $\theta = \tfrac{1}{2}$ で評価したような停留点（stationary point）では 1 次の項が消え、$g(\hat{\theta})$ の変動レートは $1/n$ と速くなって通常の $1/\sqrt{n}$ ではなくなる。第 2 に、$\hat{\theta}$ は g がほぼ線形とみなせる領域内に留まっている必要があり、これは 100% に極めて近い正解率では破綻する（演習問題 4）。そして統計量に扱いやすい導関数がそもそも存在しないときには、まったく別のアプローチ、すなわちリサンプリングが必要になる。

36.5.4 ブートストラップ法

ここまで計算してきた標準誤差はすべて解析的な公式から得られていた。しかし実務で関心を持つ量の多く（標本中央値、AUC、BLEU スコアなど）は、正解率と同様にテストセットごとに変動するにもかかわらず、その分散を表す簡潔な公式を持たない。ブートストラップ法は、標本分布をデータから直接シミュレーションすることでこの問題を解決する。以下では、まず答えがわかっている例でこの手法を検証し、次に単純な公式が存在しない場合へ適用し、最後に示唆に富む 2 つの失敗例を調べる。

36.5.4.1 プラグイン原理

もしデータを生成している真の分布 F がわかっていれば、**いかなる**統計量の標本分布を求めることも造作ない。新しいデータセットを F から何千個も抽出し、それぞれについて統計量を再計算すればよいからである。もちろん、実際には F は未知である。そこで**ブートストラップ法** (Efron 1979) は大胆な置き換えを 1 つ行う。未知の F を**経験分布**（empirical distribution）$\hat{F}_n$、すなわち確率 $1/n$ を n 個の各観測データ点に割り当てる分布で置き換えるのである。サイズ n の新しい標本を $\hat{F}_n$ から抽出することは、手元のデータから n 個の点を**復元抽出**でサンプリングすることにほかならない。アルゴリズム全体は 3 つのステップで完結する。

1. 観測標本から n 個の点を復元抽出して**ブートストラップリサンプル**を作る（元の点のうち複数回現れるものもあれば、一度も現れないものもある）。
2. そのリサンプル上で統計量 $\hat{\theta}^{*}$ を再計算する。

3. これを R 回繰り返し、反復推定値（レプリケート）$\hat{\theta}^*_1, \ldots, \hat{\theta}^*_R$ を集める。

これら R 個のレプリケートの標準偏差が $\mathrm{se}(\hat{\theta})$ の推定値となり、その $\alpha/2$ 分位点と $1-\alpha/2$ 分位点が**パーセンタイル信頼区間**（percentile confidence interval）をなす。Figure 36.29 にこの一連の流れを示す。その単純さの背後には 2 つの重要な仮定がある。観測値が独立同分布であること、そしてデータの小さな摂動に対して統計量が滑らかに変化することである。

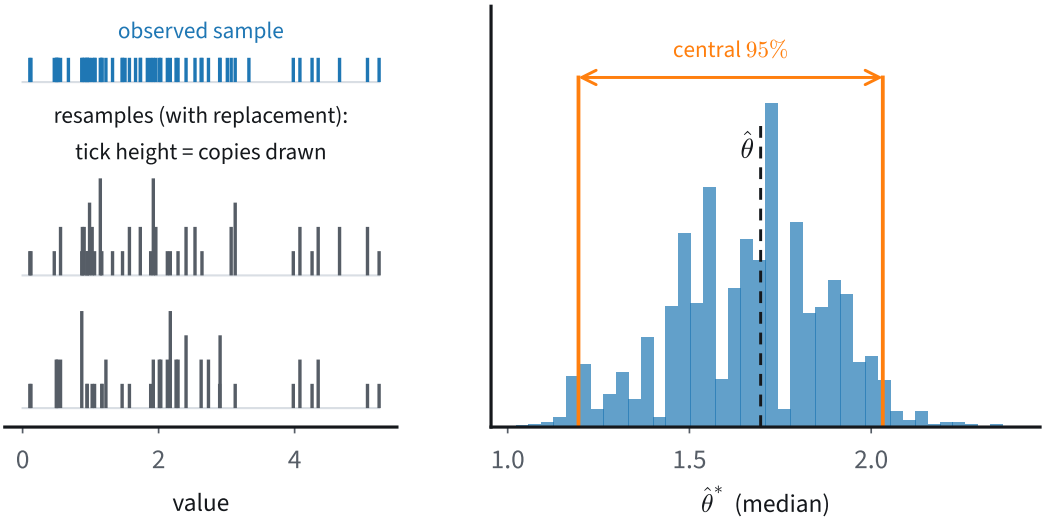


Figure 36.29: 通常のブートストラップ法。1 つの観測標本（左上）から出発し、同じサイズのリサンプルを復元抽出によって多数生成して（左下、目盛りの高さは各観測値が選ばれた回数を表す）、統計量（ここでは中央値）を再計算する。レプリケートのヒストグラム（右）は $\hat{\theta}$（破線）の周りの標本分布を近似しており、その標準偏差が標準誤差の推定値となり、中央の分位点がパーセンタイル区間をなす。ヒストグラムがぎざぎざに見えるのは、どのリサンプルの中央値も少数の観測値のいずれかに一致せざるを得ないためである。この軽微な偏在は、最大値においては完全な破綻へと変わる。

36.5.4.2 ブートストラップ標準誤差

難しい統計量でブートストラップ法を信頼する前に、既知の公式と照らし合わせて確認しておこう。冒頭のテストセットは正解 912 問、不正解 88 問で、理論的な標準誤差は 0.0090 であった。この 1,000 個の二値指示変数を 5,000 回リサンプリングし、ブートストラップ法が同じ数値を再現するか確かめてみよう。

```
correct = np.r_[np.ones(912), np.zeros(88)]          # the opening's test set
idx = rng.integers(0, len(correct), (5000, len(correct)))  # 5,000 resamples
boot = correct[idx].mean(axis=1)                     # accuracy of each resample
print(f'bootstrap se {boot.std(ddof=1):.4f}, '
      f'formula {np.sqrt(0.912 * 0.088 / 1000):.4f}')
```

```
bootstrap se 0.0090, formula 0.0090
```

2 つの推定値は数パーセント以内で一致する。では次に、実用的な閉じた形の標準誤差を持たない統計量に目を向けよう。平均 1 の指数分布から抽出した 200 個の観測値の中央値である。この分布は右に歪んでおり、真の中央値 $\log 2 \approx 0.69$ は平均よりも小さい位置にある。中央値の標準

誤差にも漸近公式 $1/(2f(m)\sqrt{n})$（ただし $f(m)$ は中央値 m における密度）は存在するものの（演習問題 6）、現実の問題では $f(m)$ がわからない。今回の合成実験ではこの密度が**既知**であり、$f(m) = e^{-\log 2} = \tfrac{1}{2}$ から理論的な基準値 $1/\sqrt{200} \approx 0.071$ が得られるため、これを使ってブートストラップ法の精度を評価できる。

```
data = rng.exponential(1.0, 200)    # its median's se formula needs the unknown density
theta_hat = np.median(data)
idx = rng.integers(0, len(data), (10000, len(data)))  # resample with replacement
boot = np.median(data[idx], axis=1)                   # one median per resample
lo, hi = np.percentile(boot, [2.5, 97.5])
print(f'median {theta_hat:.3f}, bootstrap se {boot.std(ddof=1):.3f}, '
      f'95% percentile interval ({lo:.3f}, {hi:.3f})')
```

```
median 0.741, bootstrap se 0.090, 95% percentile interval (0.636, 0.993)
```

密度を一切知らされていないにもかかわらず、ブートストラップ法は通常 0.07 付近の値を返す。異なるデータセットにわたって標本中央値は $\log 2$ の周りで変動し（通常 0.15 以内）、ブートストラップ標準誤差自体もおよそ 0.05 から 0.11 の間でばらつく。なぜこれほど大きく変動するのだろうか。それは、どのリサンプルの中央値も標本の中央付近にあるごく少数の元データ点のいずれかに一致せざるを得ず、ブートストラップ分布が離散的に偏り、それらの点同士のランダムな間隔に敏感になるためである。その結果、パーセンタイル区間はやや粗くなり、ときには $\log 2$ を辛うじて含む程度になることもある。それでもなお、中央値における背後の密度が真に正であることから、中央値はブートストラップ法が妥当であり続けるだけの滑らかさ（smoothness）を備えている。

36.5.4.3 ブートストラップ法の失敗例

ブートストラップ法がいつ破綻するかを知っておくことも同じくらい重要である。その中核にある 2 つの仮定、すなわち滑らかさと独立性（independence）は、いずれも実務において崩れ得る。

第 1 に、統計量が極端なデータ点に鋭敏に依存しすぎるとき、ブートストラップ法は破綻する。古典的な反例は標本最大値である。$x_1, \dots, x_n$ が $[0, \theta]$ 上の一様分布に独立同分布に従うとし、推定量を $\hat{\theta} = \max_i x_i$ とする。どのブートストラップリサンプルも、最大値がまったく同じ $\hat{\theta}$ になる——ただし n 回の抽出すべてがたまたま最大の観測値 1 点を取り逃がした場合を除いてであり、したがって次式が成り立つ。

$$P(\hat{\theta}^* = \hat{\theta}) = 1 - \left(1 - \frac{1}{n}\right)^n \;\to\; 1 - \frac{1}{e} \approx 0.63.$$

つまり、ブートストラップ分布は確率質量の 63% をたった 1 点に積み上げてしまうが、実際の $\hat{\theta}$ の真の標本分布は連続である。さらに悪いことに、いかなるリサンプルも $\hat{\theta}$ を超えることはできず、しかもその値は真に θ 未満であるため、パーセンタイル区間が θ を含む確率は 0 になってしまう。

第 2 に、互いに独立でない単位をリサンプリングするとブートストラップ法は破綻する。これは現代の自然言語処理やコンピュータビジョンのベンチマークで頻繁に陥る罠である。G 個の文章からなり、1 つの文章あたり m 問の設問がある読解ベンチマークを考えよう。モデルがある文章を誤解すると、その文章に属する複数の設問をまとめて間違えやすくなる。同じ文章の 2 問に対する正誤指示変数が相関（correlation）ρ を持ち、異なる文章間は独立であると仮定すると、全体の正解率の分散は次式となる。

$$\mathrm{Var}(\hat{\theta}) = \frac{\theta(1-\theta)}{Gm}\bigl(1 + (m-1)\rho\bigr),$$

この分散は、$1 + (m-1)\rho$ 倍に膨らんでいる（独立な Gm 問の場合と比較して）。個々の設問をリサンプリングする素朴なブートストラップ法はこの文章内相関を無視してしまうため、標準誤差を $\sqrt{1 + (m-1)\rho}$ 分の 1 に過小評価する。$m = 20$、控えめな相関 $\rho = 0.1$ でさえ、エラーバーを $\sqrt{2.9} \approx 1.7$ 分の 1 に過小評価してしまう (Miller 2024)。対処法は明快であり、個々の設問ではなく文章単位で丸ごとリサンプリングすればよい（演習問題 8）。

最後に、重要な未解決の問いを 2 つ残して本節を締めくくろう。第 1 に、本節では与えられた推定量をどのように評価しエラーバーを付けるかを学んだが、そもそも優れた推定量をどのように**構成**するか、あるいはいかなる推定量でもパラメータをどこまで鋭く特定できるかという限界についてはまだ問うていない。36.6 節では、尤度（likelihood）関数とその曲率（curvature）を通じて両方の問いに答える。第 2 に、本節のあらゆる保証はデータを**見る前に**固定された単一の推定量または仮説検定を前提としていた。検証セット（validation set）で最高スコアを出したという理由で選ばれたモデルはこの前提を満たさない。多数の候補から勝者を選ぶことは、姿を変えた多重検定にほかならないからである。36.10 節では、モデル族全体にわたって保証が同時に成り立つためにどれほど大きな標本が必要かを問う。

36.5.5 まとめ

ランダムなデータから計算されるあらゆる数値はそれ自体が確率変数であり、仮想的な別のデータセット間にわたる標本分布によって、その数値をどこまで信頼できるかが決まる。平均二乗誤差は統計的バイアスの二乗と分散に分解されるため、縮小平均のようにバイアスを持つ推定量が不偏な標本平均を上回ることがある。積分を期待値とみなせば標本平均はモンテカルロ推定となり、その誤差レート $\sigma/\sqrt{n}$ は決定論的なグリッドとは異なり次元が上がっても悪化しない。仮説検定は、観測された差が偶然ではめったに起こらないときに帰無仮説を棄却する。帰無仮説のもとで p 値は一様分布に従うため、$p \le \alpha$ のときに棄却すれば偽陽性率は α に制御される一方、統計的検出力は標準化効果量と $\sqrt{n}$ の積とともに増大し、競合モデルを同じ問題上で対応づけることで劇的に向上する。信頼区間とは、仮説検定が棄却しないパラメータ値の集合にほかならない。最後に、解析的な標準誤差が存在しない場合には、ブートストラップ法がデータのリサンプリングによって標本分布をシミュレーションする。これは独立な観測値の滑らかな統計量に対してはよく機能する一方、最大値のような極値やクラスタ構造を持つテストセットに対しては破綻する。

36.5.6 演習問題

1. [code] **標本平均の縮小**。本文では、縮小平均 $c\,\bar{x}$（ただし $x_1, \dots, x_n$ は $\mathcal{N}(\theta, \sigma^2)$ に独立同分布に従う）が最小の MSE を達成するのは $c^\star = \theta^2/(\theta^2 + \sigma^2/n)$ のときであることを示した。

   1. 本文における標本平均・縮小平均・中央値のシミュレーションを、係数 $c \in [0, 1]$ のグリッドへと拡張せよ（$\theta = 1$、$\sigma = 4$、$n = 30$ とする）。統計的バイアスの二乗、分散、MSE を c に対して `d2l.plot` でプロットし、経験的な最小点を $c^\star$ と比較せよ。
   2. 点 a に向かって（0 ではなく）縮小する推定量 $a + c(\bar{x} - a)$ を考える。その MSE と最適な係数を計算せよ。どのような a を選ぶべきか。また、その答えは本文における 0 という選択について何を意味するか。
   3. $c^\star$ は θ を含んでいるため推定量ではない。$\bar{x}$ を θ に、s^2 を σ^2 に代入して $\hat{c} = \bar{x}^2/(\bar{x}^2 + s^2/n)$ とし、$\hat{c}\,\bar{x}$ の MSE を $\theta \in \{0, 1, 3, 10\}$ においてシミュレーションで測定せよ。どのような場合に $\bar{x}$ を上回り、どのような場合に下回るか。演習問題 9 では、3 次元以上において同様の代入があらゆる θ で標本平均を上回ることを示す。

(c) のヒント：$\theta = 0$ ではプラグイン係数自体にノイズが含まれる。その分布を調べてみるとよい。

2. [code] **多重比較。**ハイパーパラメータのスイープでは多数の設定を 1 つのベースラインと比較するため、一度に多くの検定を実行することになる。

    1. いずれもベースラインより優れていない 20 個のモデルを、それぞれ独立な検定（$\alpha = 0.05$）によってベースラインと比較する。少なくとも 1 つの比較が有意になる確率を計算し、ボンフェローニ閾値 $\alpha/20$ における同じ確率も求めよ。また、検定間に従属性がある場合でも、ボンフェローニ補正によって少なくとも 1 つの偽陽性が生じる確率が α 以下に抑えられることを示せ。
    2. ベンジャミニ・ホッホバーグ法 (Benjamini and Hochberg 1995) は、m 個の p 値を $p_{(1)} \le \cdots \le p_{(m)}$ と並べ替え、最大の k（$p_{(k)} \le k\alpha/m$ を満たすもの）を見つけて、小さい方から k 個の p 値に対応する仮説を棄却する。$m = 200$ 個の設定からなるスイープ（そのうち $m_0 = 180$ 個は効果なし）をシミュレーションせよ。z 統計量を、効果のない設定については $\mathcal{N}(0,1)$ から、残りの $\mathcal{N}(3,1)$ から他の 20 個について抽出し、両側 p 値 `2 * norm.sf(np.abs(z))` を計算せよ。1,000 回の反復にわたり、補正なしの閾値（棄却が 1 つもない場合は偽発見割合を 0 とし、閾値は 0.05 とする）、ボンフェローニ補正、およびベンジャミニ・ホッホバーグ法のそれぞれについて、真の発見数と偽発見数の平均および偽発見割合（棄却された仮説のうち偽である割合）の平均を報告せよ。独立な検定に対して、ベンジャミニ・ホッホバーグ法は偽発見割合の期待値が高々 $\alpha m_0/m$ であることを保証する。シミュレーション結果はこの理論と一致するか。

3. [code] **小さな n における被覆率。**信頼区間 Equation 36.73 はガウス分位点 1.96 を t 分位点の代わりに用いており、標本平均が近似的にガウス分布に従うことを仮定している。

    1. 本文の被覆率シミュレーション（標準正規データセットから 1,000 個の区間を構成する実験）を、$n \in \{2, 5, 10, 30, 100\}$ および試行回数 100,000 回で実行し、区間の被覆率を記録せよ。次に、1.96 を 97.5% 分位点（t 分布、自由度 $n-1$）である 12.71、2.776、2.262、2.045、1.984（各 n に対応）に置き換え、再び被覆率を記録せよ。
    2. $n = 2$ とし、1.96 を 1.0（75% 分位点、自由度 1 の t 分布）に置き換えて、区間の被覆率が 50% になるようにせよ。幅の中央値より短い区間と長い区間に分けて、それぞれの被覆率を計算せよ。この差は Equation 36.71 と矛盾するか。また、短い区間が得られたとき、それを推定精度が高い徴候と解釈すべきか。
    3. データを平均 1 の指数分布から $n = 10$ で抽出し、t 分位点 2.262 はそのまま維持せよ。被覆率を測定し、真の平均より完全に下側に外れた区間と完全に上側に外れた区間を別々に数えよ。この非対称性を分布の形状から説明せよ。

4. [code] **境界付近におけるデルタ法。**@eq-mdl-delta-method より、正解率 $\hat{\theta} = 0.90$（$n = 1{,}000$ 個のテスト例で測定）の対数オッズ $\log \frac{\hat{\theta}}{1-\hat{\theta}}$ の標準誤差は約 0.105 である。

    1. 正しい予測の個数を二項分布（パラメータは 100,000 個のテストセットに対して $n = 1{,}000$、$\theta = 0.9$）から抽出してテストセットをシミュレーションせよ。対数オッズの標準偏差をデルタ法による値と比較せよ。
    2. $\theta = 0.999$ で同様の実験を行え。$\hat{\theta} = 1$ となるテストセットの割合を推定して厳密値と比較し、そのようなテストセットにおける対数オッズがどうなるか述べよ。
    3. 1 次展開の背後にあるどの仮定が $\theta = 0.999$ では破綻するか。誤答数の期待値 $n(1-\theta)$ を両方の θ の値について計算し、n 単独ではなくこの個数がデルタ法やワルド区間を信頼できるかどうかを決める理由を説明せよ。
    4. あるベンチマークに 30 問あり、モデルがそのうち 29 問に正解したとする。ワルド区間 $\hat{\theta} \pm 1.96\sqrt{\hat{\theta}(1-\hat{\theta})/30}$ を計算し、その端点がどこに来るか確認せよ。次に検定を

反転させよ。すなわち、各 θ_0（$[0.5, 1]$ 上の細かいグリッド）について、厳密な両側 p 値（帰無仮説 $H_0 : \theta = \theta_0$）である $\min\{1,\, 2\min\{P_{\theta_0}(K \le 29),\, P_{\theta_0}(K \ge 29)\}\}$（ただし $K \sim \text{Binomial}(30, \theta_0)$）を計算し、棄却されずに残る θ_0（$\alpha = 0.05$）を集めよ。2 つの区間を比較し、さらに $\theta = 0.95$ における 100,000 回のベンチマーク・シミュレーションでの被覆率を比較せよ。

5. [code] **対応のある比較。**●同じテスト例または同じシード上で評価された 2 つのモデルはノイズの一部を共有しており、対応のある検定はその共有ノイズを取り除く。

   1. 本文では 2 つのモデルが 41 問で食い違い、内訳は 25 対 16 であった。H_0 のもとで B のみが正解する問題数は $\text{Binomial}(41, \tfrac{1}{2})$ に従う。厳密な両側 p 値を計算し、36.5.2.6 節の符号反転コードセルによるモンテカルロ値と比較せよ。次に正味の差を 9 に固定したまま不一致の総数を変え、p を 9 対 0、17 対 8、25 対 16、55 対 46 のそれぞれについて計算せよ。判定を左右するのは正味の差と不一致数のどちらか。
   2. 両モデルが共有するシード効果を含めてシードごとの正解率をモデル化する。シード s における正解率を $0.850 + u_s + e_{A,s}$ および $0.858 + u_s + e_{B,s}$ とする（ただし $u_s \sim \mathcal{N}(0, 0.01^2)$ と $e_{A,s}, e_{B,s} \sim \mathcal{N}(0, 0.003^2)$ は互いに独立）。シードごとの差に対する符号反転検定を厳密検定として実装せよ。すなわち、k 個のシードに対してランダム抽出の代わりに全 2^k 通りの符号パターンを列挙せよ。1,000 回の実験（10 シード）にわたり、その統計的検出力を対応のない並べ替え検定 `perm_p` と比較せよ。
   3. 5 シードで同様の実験を行え。厳密な符号反転検定が返し得る最小の両側 p 値はいくらであり、それは $\alpha = 0.05$ において何を意味するか。ランダムな符号パターンを 999 個抽出するモンテカルロ版（`perm_p` がランダムなシャッフルを抽出するのと同様の方法）は、これらの実験の数パーセントで棄却してしまう。それはなぜか。

6. **中央値の標準誤差。**●$x_1, \ldots, x_n$ を累積分布関数 F、中央値 m、中央値における密度 $f(m) > 0$ を持つ独立同分布の確率変数とする。標本中央値 $\hat{m}$ が高々 $m + t$ になるのは、観測値の少なくとも半分が高々 $m + t$ であるとき、かつそのときに限る。

   1. 経験累積分布関数 $\hat{F}_n(m+t)$ は平均 $F(m+t)$、分散 $F(m+t)(1-F(m+t))/n$ を持つ。中心極限定理を用いて、$P(\hat{m} \le m + t) \approx \Phi\big(2\sqrt{n}\, f(m)\, t\big)$ が小さな t に対して成り立つことを示し、$\text{se}(\hat{m}) \approx 1/\big(2 f(m) \sqrt{n}\big)$ を結論せよ。また、因子 $1/f(m)$ を傾き g'（Equation 36.74）と関連づけよ。
   2. 平均 1 の指数分布において $n = 200$ としてこの近似値を評価し、本文における中央値のブートストラップを実行した際に出力されるブートストラップ標準誤差と比較せよ。コードセルを何度か再実行すると、出力値はおよそ 0.05 から 0.11 の間で変動する。平均の標準誤差に対するブートストラップ推定値はそれほど変動しないにもかかわらず、中央値の標準誤差に対するブートストラップ推定値がデータセットごとにこれほど大きく変動するのはなぜか。
   3. 比 $\text{Var}(\hat{m})/\text{Var}(\bar{x})$ を、大きな n のもとでガウスデータとラプラスデータのそれぞれについて計算せよ。裾の重いデータに対しては、位置パラメータの推定量としてどちらを選ぶべきか。

   (b) のヒント：リサンプルの中央値は $\hat{m}$ 付近にある少数の観測値のいずれかになり、それらの値の間隔はランダムである。

7. [code] **グリッドとモンテカルロ法の比較。**●36.5.1.7 節では指数を比較した。すなわち、N 回の評価を d 次元で行うとき、2 次精度グリッドの誤差は $N^{-2/d}$ のオーダー、モンテカルロ法の誤差は $N^{-1/2}$ のオーダーとなる。この比較を $f(\mathbf{x}) = \exp(-\|\mathbf{x}\|^2)$（領域は $[0,1]^d$）上で検証せよ。その積分値はフビニの積の法則から $\big(\tfrac{\sqrt{\pi}}{2}\,\text{erf}(1)\big)^d$ である（34.6 節）。

   1. m^d 個の超立方体からなる全グリッド上の中点則と、モンテカルロ平均 Equation 36.67

（$N = m^d$ 個の一様サンプル上）を実装せよ。$d \in \{1, 2, 4, 5, 6\}$ および N を最大約 10^6 まで変化させ、相対誤差を N に対して両対数軸でプロットせよ（モンテカルロ法には 20 回の反復にわたる二乗平均平方根誤差を用いること）。傾きは予測された指数と一致するか。

2. $d = 4$ では両者の指数が並ぶ。そこではどちらの手法がより高精度か。各レートの前に付く定数は何によって決まり、それぞれの定数は d とともにどのように変化するか。
3. 両方の誤差は理論的に予測できる。f の中点和は 1 次元の中点和の積になる。また、モンテカルロ法の二乗平均平方根誤差は $\mathrm{sd}\big(f(\mathbf{U})\big)/\sqrt{N}$（Equation 36.68 より、ただし $\mathbf{U}$ は $[0,1]^d$ 上の一様分布）であり、積の法則によって $\mathbb{E}\big[f(\mathbf{U})^2\big]$ は 1 次元積分へと因数分解される。予測される両方の相対誤差を $d \in \{4, 5, 6\}$、$N \approx 10^5$ において計算し、実測値と比較せよ。どの d で 2 つの手法の精度が逆転するか。また、20 回の反復によってその d における優劣をどれほど確実に判定できるか。

8. [code] **クラスタ構造を持つテストセット。**●評価セットは 1 つの文章に関する複数の設問のように互いに関連する項目のクラスタから構成されることが多く、その場合、独立同分布を仮定したブートストラップ法は標準誤差を過小評価する (Miller 2024)。あるテストセットが G 個のクラスタ（各クラスタに m 項目）からなり、期待正解率は θ、同一クラスタ内の 2 項目の正誤指示変数は相関 ρ を持ち、異なるクラスタに属する項目は互いに独立であるとする。

   1. 共通の分散と対ごとの相関を持つ m 個の変数（共通分散 v、対ごとの相関 ρ）の平均が分散 $v\big(1 + (m-1)\rho\big)/m$ を持つことを示し、テストセット上の正解率 $\hat{\theta}$ の分散が次式で与えられることを結論せよ。
   $$\mathrm{Var}(\hat{\theta}) = \frac{\theta(1-\theta)}{Gm}\big(1 + (m-1)\rho\big). \tag{36.75}$$
   2. $G = 100$、$m = 20$ としてシミュレーションを行え。各クラスタの正解率を $\theta_g \sim \mathrm{Beta}(6.3, 2.7)$ から抽出し、θ_g が与えられたもとで各項目を条件付き独立に抽出せよ。この設定では $\theta = 0.7$ および $\rho = 1/(6.3 + 2.7 + 1) = 0.1$ となる。300 個のテストセットにわたり、$\hat{\theta}$ の標準偏差を、項目単位でリサンプリングする（本文の正解率ブートストラップと同様の）独立同分布ブートストラップ標準誤差の平均、およびクラスタ全体をリサンプリングするクラスタ・ブートストラップ標準誤差の平均と比較せよ（それぞれ $R = 1{,}000$ 回のリサンプルを用いること）。独立同分布ブートストラップ法は標準誤差を何分の 1 に過小評価するか。その倍率を (a) の理論予測と比較せよ。

9. [code] **スタインのパラドックス。**●●$\mathbf{x} \sim \mathcal{N}(\theta, \mathbf{I}_d)$ を観測し、推定量をリスク $\mathbb{E}\|\hat{\theta} - \theta\|^2$（二乗を二乗ノルムに置き換えた平均二乗誤差）で評価する。推定量 $\mathbf{x}$ のリスクは d であり、これはすべての θ について一定である。James and Stein (1961) は、$d \geq 3$ のとき推定量
$$\hat{\theta}_{\mathrm{JS}} = \left(1 - \frac{d-2}{\|\mathbf{x}\|^2}\right)\mathbf{x} \tag{36.76}$$
がすべての θ に対してより小さいリスクを持つことを示した。

   1. スタインの補題 (Stein 1981) を証明せよ。すなわち、$z \sim \mathcal{N}(\theta, 1)$ であり、h が微分可能で $\mathbb{E}|h'(z)| < \infty$ を満たすならば、$\mathbb{E}[(z-\theta)h(z)] = \mathbb{E}[h'(z)]$ が成り立つことを部分積分により示せ。
   2. 推定量 $\mathbf{x} + \mathbf{g}(\mathbf{x})$ において $\mathbf{g}$ が弱微分可能（ここでは原点以外で微分可能かつ $\mathbb{E}\|\mathbf{g}(\mathbf{x})\|^2 < \infty$）であるとき、そのリスクが
   $$d + \mathbb{E}\|\mathbf{g}(\mathbf{x})\|^2 + 2\sum_{i=1}^{d} \mathbb{E}\left[\frac{\partial g_i}{\partial x_i}(\mathbf{x})\right]. \tag{36.77}$$

となることを示せ。この恒等式を用いて、$\widehat{\theta}_{\mathrm{JS}}$ のリスクが $d-(d-2)^2\,\mathbb{E}\big[1/\|\mathbf{x}\|^2\big]$ であることを示せ。この議論のどこで $d \geq 3$ が必要になるか。

3. $\mathbf{x}$、$\widehat{\theta}_{\mathrm{JS}}$、および正値部分推定量 $\max(0, 1-(d-2)/\|\mathbf{x}\|^2)\mathbf{x}$ のリスクを、$d=10$ および $\|\theta\| \in \{0,1,2,5,10,20\}$ においてシミュレーションし、$\|\theta\|$ に対してプロットせよ。
4. 最適な係数 $c^\star$ を $c\,\mathbf{x}$ について d 次元で書き下せ（本文の $c^\star$ の多次元版）。$\widehat{\theta}_{\mathrm{JS}}$ では未知の量がデータのどの関数で置き換えられており、なぜこの代用量は d が大きくなるにつれて正確になるのか。演習問題 1 (c) と比較せよ。

10. [extended] **実行中の検定の覗き見**。●●A/B テストや訓練中の評価では、データが到着する最中に結果を監視し、有意に見えた時点で打ち切ることがよく行われる (Johari et al. 2017)。$x_1, x_2, \ldots$ を $\mathcal{N}(\mu, 1)$ に独立同分布に従う確率変数、$H_0 : \mu = 0$、$S_n = \sum_{i \leq n} x_i$ とする。z 検定（時点 n）は $|S_n|/\sqrt{n} \geq 1.96$ のときに棄却する。

    1. 4,000 回の実行（長さ $N = 10{,}000$）を H_0 のもとでシミュレーションせよ。等間隔にとった K 回の確認時点のうち少なくとも 1 回で棄却が起きる確率を、$K \in \{1, 2, 5, 10, 100, 1000, 10000\}$ について推定せよ。重複対数の法則によれば、監視を無制限に続けるとこの確率は 1 に収束する。作成した表ではどれほどの速さで増加するか。
    2. 事前分布 $\mathcal{N}(0, \tau^2)$ を μ に置き、混合尤度比を

$$\Lambda_n = \int \prod_{i=1}^{n} \frac{p(x_i \mid \mu)}{p(x_i \mid 0)}\, \mathcal{N}(\mu; 0, \tau^2)\, d\mu. \tag{36.78}$$

    と定義する。$\Lambda_n = (1+n\tau^2)^{-1/2} \exp\!\big(\tau^2 S_n^2 / (2(1+n\tau^2))\big)$ が成り立ち、かつ H_0 のもとでこれがマルチンゲールであること、すなわち $\mathbb{E}[\Lambda_{n+1} \mid x_1, \ldots, x_n] = \Lambda_n$ を満たすことを示せ。
    3. ヴィルの不等式（ここでは事実として認める）によれば、$\Lambda_0 = 1$ を満たす非負マルチンゲールに対して $P(\sup_n \Lambda_n \geq 1/\alpha) \leq \alpha$ が成り立つ。これを用いて、最初の n（$\Lambda_n \geq 1/\alpha$ を満たす時点）で棄却する検定は、データをどれほど頻繁に覗き見ても第一種の過誤が高々 α に抑えられることを結論せよ。また、$|S_n|$ に対する棄却境界を導出し、n に対するその増大の仕方を固定境界 $1.96\sqrt{n}$ と比較せよ。
    4. この逐次検定の第一種の過誤を $\tau^2 \in \{0.01, 0.1, 1\}$ についてシミュレーションで確認せよ。次に $\mu = 0.2$ とし、固定標本の z 検定（標本サイズ n は Equation 36.69 から統計的検出力 0.8 に対して定める）を基準とする。3 つの τ^2 の値のそれぞれについて、時点 n までの逐次検定の検出力、停止時刻の中央値、および時点 N までの検出力を報告せよ。τ^2 の選択はこれらの数値にどう影響するか。また、どの確認時点でも妥当な検定は何を手放し、代わりに何を得ているか。

36.6 最尤法

コインを 13 回投げて、表が 9 回、裏が 4 回出たとしよう。誰であっても、表が出る確率を $9/13 \approx 0.69$ と見積もるだろう。しかし、直ちに 2 つの自然な疑問が生じる。なぜ他ならぬその値なのか、そしてその値をどれほど信用してよいのか、という疑問である。このコインが実は公平なコインである可能性は残っていないだろうか。

1 つの原理がこの両方の疑問に答えてくれる。観測されたデータが最も起こりやすくなるようなパラメータを選ぶ、という原理である。これが**最尤法**（maximum likelihood）の原理であり、本書におけるほぼすべての訓練損失の根底にある。コインの場合、この原理は $9/13$ を返す。ニューラルネットワークの場合、ターゲットに含まれるノイズについての仮定を、最小化すべき損失へ

と変換する。これまでの章で経験的にうまく機能するという理由から採用してきた二乗誤差や交差エントロピー（cross-entropy）は、いずれもこのようにして導かれる。さらに注目すべきことに、同じ損失から、データがパラメータをどれほど強く絞り込んでいるかも読み取れる。最小点の周りで損失が鋭く湾曲しているほどエラーバーは小さくなり、先ほどのコインについても、13 回の試行で公平なコインの可能性を棄却できるかどうかがその曲率からわかる。

先の 36.5 節では、標本平均や標本比率のように閉形式で与えられる推定量の誤差を評価した。本節では、推定量が最適化問題によって定義され、そのエラーバーは損失そのものから直接得られる。まず、これまでの章で用いてきた標準的な損失が最尤法からどのように導かれるかを示す。続いて本節の核心である、最尤推定量はどれほど正確かという問いに移る。その主結果、すなわち標本が大きくなるにつれて推定誤差はガウス分布に近づき、その分散は損失の曲率によって決まるという事実は、ぜひ押さえておきたい。最後に、3 回連続で表が出た場合を例に、データが少ないときに何が破綻するか、そしてデータを見る前に抱いていた事前の信念がいかにして正則化項へと変わるかを見る。各観測の一部が隠れているモデルについては、続く 36.7 節で扱う。

36.6.1 最尤法の原理

確率モデルは、それが生成しうるあらゆるデータセットに対して確率を割り当てる。ひとたびデータが手に入ったら、この視点を逆転させる。すなわち、観測されたデータを固定し、パラメータの各候補のもとでそのデータがどれほど起こりやすいかを問うのである。

モデルのパラメータを θ、観測されたデータセットを X とする。**尤度**（likelihood）とは、観測データの確率（連続データの場合はその密度）をパラメータの関数とみなした $P(X \mid \theta)$ のことである。**最尤推定値**（maximum likelihood estimate、MLE；推定規則としては**最尤推定量**）とは、この関数を最大化するパラメータの値であり、次式で定義される。

$$\hat{\theta} = \operatorname*{argmax}_{\theta} P(X \mid \theta). \tag{36.79}$$

あらゆる候補パラメータの中で、MLE は実際に観測された事象が最も起こりやすかったようなパラメータである。なお、この規則は θ を確率変数ではなく未知の定数として扱う。つまり確率はあくまでデータセットに割り当てられ、パラメータの値に割り当てられることは決してない。この区別は、36.6.4 節で事前分布（prior）を導入する際に重要となる。

36.6.1.1 コインの例

偏りのあるコインは、この原理を最後まで手計算で追うことができる最も単純なモデルである。唯一のパラメータは、表が出る確率 θ である。36.3 節ではこの数値を p と呼んだが、ここでは推定すべき未知数であるため θ と書く。各試行は独立なので確率は積の形になり、表が n_H 回、裏が n_T 回出る特定の系列の確率は次式となる。

$$P(X \mid \theta) = \theta^{n_H}(1-\theta)^{n_T}. \tag{36.80}$$

θ の関数とみなしたとき、この式が尤度である。13 回の試行結果の系列 HHHTHTTHHHHHT（表が $n_H = 9$ 回、裏が $n_T = 4$ 回）に対しては、尤度は $\theta^9(1-\theta)^4$ となる。公平なコインがこの系列に割り当てる確率は $2^{-13} \approx 1.2 \times 10^{-4}$ であるのに対し、$\theta = 9/13$ のコインは 3.3×10^{-4} と、約 2.7 倍の確率を割り当てる。

最大点を見つけるには、まず対数をとる。対数関数は狭義単調増加であるため、最大点の位置を変えずに積を微分しやすい和へと変換してくれる。導関数を 0 とおくと、次が得られる。

$$\frac{d}{d\theta}\Big[9\log\theta + 4\log(1-\theta)\Big] = \frac{9}{\theta} - \frac{4}{1-\theta} = 0 \quad\Longrightarrow\quad 9(1-\theta) = 4\theta \quad\Longrightarrow\quad \hat{\theta} = \frac{9}{13}.$$

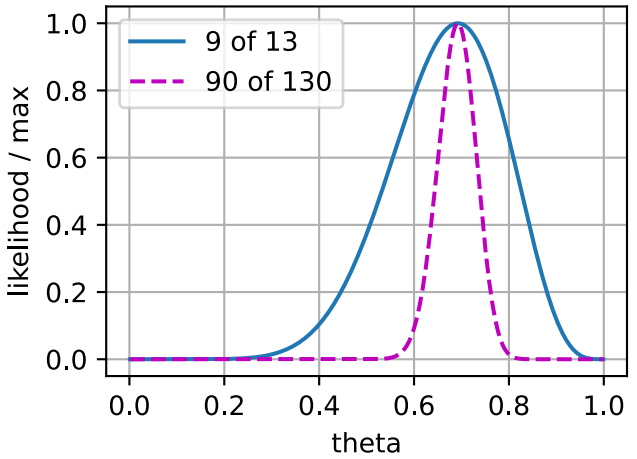
端点 $\theta = 0$ と $\theta = 1$ は、表と裏の両方を含むあらゆる系列に対して確率 0 を割り当てるため、最大点は開区間 $(0, 1)$ の内部になければならず、この停留点が実際に大域的な最大点となる。一般の尤度 Equation 36.80 に対して同じ計算を繰り返すと、次式が得られる。

$$\frac{n_H}{\theta} - \frac{n_T}{1-\theta} = 0 \quad\Longrightarrow\quad \hat{\theta} = \frac{n_H}{n_H + n_T}.$$

コインの偏りの MLE は、直感どおり単に観測された表の相対頻度にほかならない。ただし、ここで導関数を詳しく見ておこう。各試行を $x \in \{0, 1\}$（表を $x = 1$、裏を $x = 0$）と符号化すると、導関数は試行ごとの項 $x/\theta - (1-x)/(1-\theta)$ の和になっている。各項は 1 つの観測の対数確率の勾配であり、後に $\hat{\theta}$ の精度を測る際に主役となる。

ここで、同じ相対頻度が 10 倍の試行回数、すなわち 130 回中 90 回の表から得られたとしよう。点推定値は比率のみに依存するため変わらない。変わるのは、データがその値をどれほど鋭く選び出すかである。以下では、2 つの曲線が同じ縦軸を共有できるよう、それぞれの尤度を最大値で割ってプロットする。

```
theta = np.linspace(0.001, 0.999, 999)

def likelihood(theta, n_H, n_T):  # divided by its maximum
    lik = theta**n_H * (1 - theta)**n_T
    return lik / lik.max()

d2l.plot(theta, [likelihood(theta, 9, 4), likelihood(theta, 90, 40)], 'theta',
         'likelihood / max', legend=['9 of 13', '90 of 130'])
```



どちらの曲線も $9/13 \approx 0.69$ でピークに達するが、2 つ目の曲線は幅が約 3 分の 1 に狭まっている。本節の後半では、このピークの幅を正確なエラーバーへと変換する。

36.6.1.2 負の対数尤度

コインでは閉形式解が得られた。しかし数百万個のパラメータをもつモデルには閉形式解がないため、数値的に最適化する必要があり、浮動小数点演算では生の尤度は使いものにならない。n 個の独立なデータ点があるとき、尤度は n 個の確率の積であり、次式で表される。

$$P(X \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta),$$

各因子はいずれも $[0, 1]$ の範囲にある。$1/2$ 付近の因子を 10 億個掛け合わせるとおよそ $(1/2)^{10^9}$ となり、いかなる浮動小数点形式でも勾配を計算するはるか手前でゼロへとアンダーフローしてしまう。

対数をとれば、その積は表現可能な範囲に余裕をもって収まる和へと変わる。

$$\log\big((1/2)^{10^9}\big) = 10^9 \cdot \log(1/2) \approx -6.93 \times 10^8.$$

したがって、尤度の最大化は**対数尤度**（log-likelihood）$\log P(X \mid \theta)$ の最大化と同値である。最適化ルーチンは目的関数の最大化ではなく損失の最小化として実装されるため、符号を反転させて**負の対数尤度**（negative log-likelihood、NLL）を次のように定義する。

$$\ell(\theta) = -\log P(X \mid \theta) = -\sum_{i=1}^{n} \log p(x_i \mid \theta). \tag{36.81}$$

NLL は観測ごとの項の和であり、各データ点に対するモデルの「驚き（サプライズ）」を測る量となっている。コインの場合は $\ell(\theta) = -\big(n_H \log\theta + n_T \log(1-\theta)\big)$ となり、閉形式解が存在しないかのように勾配降下法で最小化することもまったく問題なくできる。以下では、約 900 万回のコイン投げに対して勾配降下法を実行し、結果を $n_H/(n_H + n_T)$ と比較する。

```
n_H = 8675309
n_T = 256245
theta = torch.tensor(0.5, requires_grad=True)
lr = 1e-9
for _ in range(100):
    loss = -(n_H * torch.log(theta) + n_T * torch.log(1 - theta))
    loss.backward()
    with torch.no_grad():
        theta -= lr * theta.grad
    theta.grad.zero_()
theta, n_H / (n_H + n_T)
```

```
(tensor(0.9713, requires_grad=True), 0.9713101437890875)
```

勾配降下法は 0.9713 に収束し、表示されたすべての桁で観測頻度と一致している。学習率 10^{-9} は極端に小さく見えるが、損失の和をとっているためにこれが必要になる。約 900 万回の試行にわたって和をとった損失の勾配には n の因子が掛かるため、安定な最大ステップサイズもそれに反比例して小さくなるからである（35.1.2 節）。本書全体を通してミニバッチ損失を例の和ではなく平均として定義しているのは、このためである。

また、対数をとることで勾配もすっきりした加法的な形になる。

$$\nabla_\theta \, \ell(\theta) = -\sum_{i=1}^{n} \frac{\nabla_\theta \, p(x_i \mid \theta)}{p(x_i \mid \theta)},$$

このように例ごとに独立な 1 つの項が得られるのに対し、尤度に直接積の微分法則を適用すれば n 個の積（各々が $n-1$ 個の因子からなる）が生じることになる。その結果、ランダムに抽出したミニバッチから $\nabla \ell$ の不偏推定値が得られ、これによって確率的勾配降下法が可能になる。

36.6.2 負の対数尤度としての損失

コインの例では、確率モデルから直接 NLL を導出した。これに対しこれまでの章では、二乗誤差や交差エントロピーといった訓練損失を直接書き下し、うまく機能するからという理由で使ってきた。実はそうした標準的な損失の大半は姿を変えた負の対数尤度であり、したがって損失関数を選ぶことはデータの確率モデルを選ぶことにほかならない。

36.6.2.1 交差エントロピーとエントロピーの下限

分類器の訓練損失は平均 NLL であり、その訓練曲線はゼロに達するのではなく正の値で頭打ちになることが多い。その値が何を表し、どこまで下がりうるのかを理解するために、平均 NLL を個々の例にわたる和ではなく、異なる結果にわたる和として書き直してみよう。標本の**経験分布**（empirical distribution）を $\widehat{p}_{\text{data}}$ とする。すなわち、結果 x が n_x 回（全 n 回の観測中）現れるとき、その観測頻度である確率 $\widehat{p}_{\text{data}}(x) = n_x / n$ を割り当てる。先ほどのコインの系列では、$\widehat{p}_{\text{data}}(\text{H}) = 9/13$ および $\widehat{p}_{\text{data}}(\text{T}) = 4/13$ となる。

ここで、データ点にわたる和を、$x_i = x$ であるすべての観測をまとめ上げることで**異なる結果**にわたる和へと組み替える。コインの例では、9 回の表が 9 個の $\log p_\theta(\text{H})$ にまとまるため、平均には表が重み $9/13$ で、裏が重み $4/13$ で入る。一般に、次式が成り立つ。

$$\begin{aligned} \frac{1}{n}\,\ell(\theta) &= -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i) = -\sum_{x} \frac{n_x}{n}\, \log p_\theta(x) \\ &= -\sum_{x} \widehat{p}_{\text{data}}(x)\, \log p_\theta(x) = \text{CE}(\widehat{p}_{\text{data}},\, p_\theta)\,. \end{aligned}$$

右辺は経験分布からモデルへの**交差エントロピー** $\text{CE}(p, q) = -\sum_x p(x) \log q(x)$ である。正の定数 $1/n$ を掛けても最小点は動かないため、離散的な結果に対する最尤法は交差エントロピーの最小化と同値になる。わずか 1 行の項のまとめ直しによって、分類損失がこの名前で呼ばれる理由が説明できる。

この交差エントロピーはどこまで小さくなりうるだろうか。実は、データによって固定される部分とモデルが制御できる部分とにきれいに分解できる。36.4.2 節で見たように、有限集合上の分布 p のエントロピーは $H(p) = -\sum_x p(x)\log p(x)$ すなわち平均の驚きであった。カルバック・ライブラー（KL）ダイバージェンス $D_{\text{KL}}(p \,\|\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ は、真の分布が p であるときに q を用いることで生じる余分な驚きの平均を測る（両者の詳細は 37.1 節で展開する）。本節を通して自然対数を用いるため、これら 3 つの量の単位はいずれも **nat** である。$\sum_x p(x)\log p(x)$ を足して引くと、次式が得られる。

$$\text{CE}(p, q) = -\sum_x p(x)\log q(x) = -\sum_x p(x)\log p(x) + \sum_x p(x) \log \frac{p(x)}{q(x)} = H(p) + D_{\text{KL}}(p \,\|\, q).$$

エントロピーは p のみに依存し、モデル q に依存するのは KL ダイバージェンスだけである点に注目しよう。ギブスの不等式（36.4.2 節）により、KL ダイバージェンスは決して負にならず、ゼロになるのは $q = p$ のときに限られる。

$p = \hat{p}_{\text{data}}$、$q = p_\theta$ とおけば、訓練損失は観測頻度のエントロピーを決して下回ることができず、その下限に達するのはモデル族がその観測頻度を厳密に再現できる場合に限られることがわかる。訓練によって減らせるのは、経験頻度からモデルへの KL ダイバージェンスだけである。先ほどのコインでは、下限は $H(9/13) = -\frac{9}{13}\log\frac{9}{13} - \frac{4}{13}\log\frac{4}{13} = 0.617$ nat となる。公平なコインのモデルは交差エントロピーが $\log 2 = 0.693$ nat であり、0.076 nat の KL ギャップを残す。また $\theta = 0.9$ のモデルは 0.164 nat のギャップを残す。Figure 36.30 に曲線全体を示す。

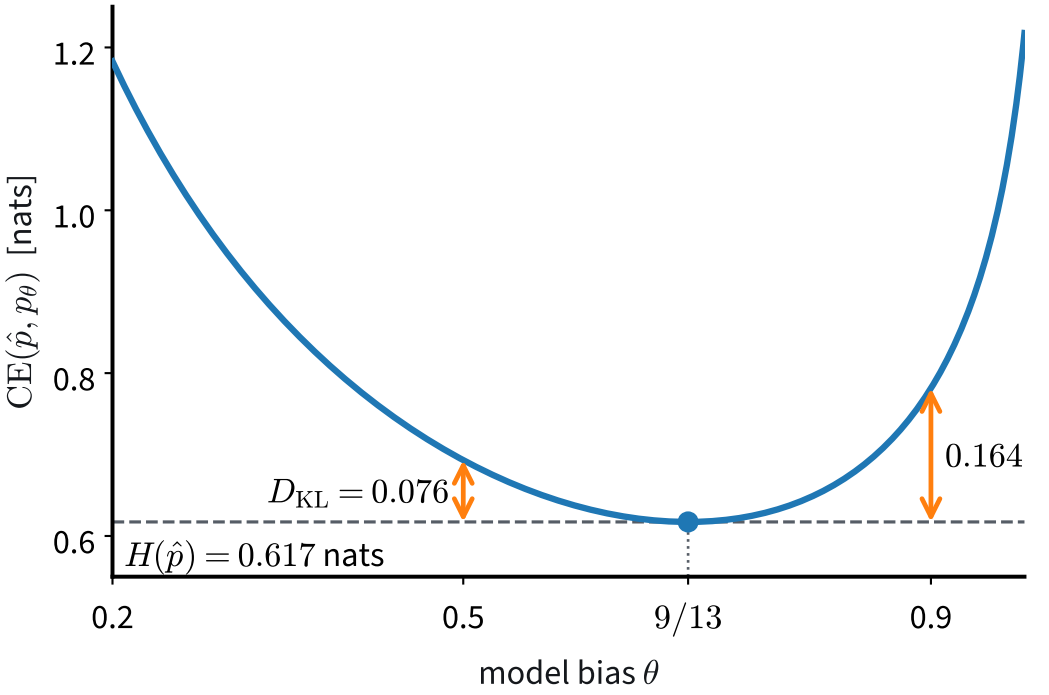


Figure 36.30: コインデータ（表 9 回、裏 4 回）に対する試行あたりの交差エントロピーをモデルの偏り θ の関数として描いた図。損失は観測頻度のエントロピーである 0.617 nat（破線）を決して下回らず、MLE である $9/13$ でその下限に接し、それ以外の任意の θ ではデータの頻度からモデルへの KL ダイバージェンス（オレンジ色）の分だけ下限を上回る。具体的には $\theta = 1/2$ で 0.076 nat、$\theta = 0.9$ で 0.164 nat となる。

同じ恒等式は分類器にも当てはまる。分類器は $p_\theta(y \mid \mathbf{x})$ をモデル化し、その平均 NLL は訓練セット上で $-\log p_\theta(y_i \mid \mathbf{x}_i)$ の平均をとる。入力ごとに例をグループ化すれば、異なる各入力において交差エントロピーの分解が復元され、その入力における経験ラベル頻度が $\hat{p}_{\text{data}}$ の役割を果たす（演習問題 1）。ワンホットラベルとソフトマックス出力の場合、この平均 NLL はまさしく 3.1 節のソフトマックス交差エントロピー損失に一致する。典型的なデータセットでは各訓練入力は 1 度しか現れないため、各入力における経験ラベル分布はワンホットとなり、ラベルにどれほどノイズが含まれていても訓練損失の下限はゼロになる。実務家が直面する削減不能な下限はホールドアウトデータ上で現れ、そこでは期待損失が入力を与えたもとでのラベルの条件付きエントロピーを下回ることはできない（37.1.3.1 節）。

言語モデルは、ラベル集合が非常に大きい同じ分類器にほかならない。入力はそこまでのテキス

ト、ラベルは次のトークンであり、訓練ではトークンあたりの平均 NLL を最小化する。この値は通常、その指数をとった**パープレキシティ**（8.3.2 節）として報告される。これは同じ平均の驚きをもたらす等確率なトークンの個数に相当する。言語モデルでは、ホールドアウトでの下限はゼロからほど遠い。たいていの文には自然な続き方が何通りもあるため、次のトークンの条件付きエントロピーは大きく、いかに巨大なモデルであってもホールドアウト損失をその下限より小さくすることはできない。

36.6.2.2 確率密度

連続データの場合、尤度の定義にはもう 1 段階の手続きが必要になる。特定の厳密な実数値が 1 つ観測される確率はゼロであるため、文字どおりにとればどの θ に対しても尤度は 0 となり、パラメータの候補どうしを区別できなくなってしまう。そこで代わりに、各観測が小さな許容幅 ϵ の範囲内で一致することだけを求め、そのうえで ϵ が最適化から消去されることを確かめる。

密度 $p(x \mid \theta)$ に従う独立同分布（i.i.d.）な観測 $x_1, \ldots, x_n$ に対し、すべての X_i が幅 ϵ の区間（x_i の周囲）に入る確率は、ϵ の一次までで次のように近似できる。

$$P\big(X_i \in [x_i, x_i + \epsilon] \; \forall i \mid \theta\big) \approx \epsilon^n \prod_{i=1}^{n} p(x_i \mid \theta).$$

負の対数をとると、次式が得られる。

$$-\log P\big(X_i \in [x_i, x_i + \epsilon] \; \forall i \mid \theta\big) \approx -n \log \epsilon - \sum_{i} \log p(x_i \mid \theta).$$

許容幅 ϵ は加法定数 $-n \log \epsilon$ を通じてのみ現れ、この項は θ に依存しない。求める精度が 4 桁であっても 400 桁であってもその定数がずれるだけで最小点は動かないため、これを落として離散の場合とまったく同様に $-\sum_i \log p(x_i \mid \theta)$ を最小化する。ただし、密度の値そのものは確率ではない点に留意したい。確率を近似するのは積 $\epsilon\, p(x_i \mid \theta)$ だけであるため、p は x の逆数の単位をもち、1 を超えることもありうる。

母集団における交差エントロピー恒等式は、連続データにもそのまま引き継がれる。標本上の負の対数密度の平均が引き続き目的関数であり、真のデータ生成密度 $p_\star$ のもとでのその期待値は次式となる。

$$\mathbb{E}_{x \sim p_\star}\big[-\log p(x \mid \theta)\big] = h(p_\star) + D_{\mathrm{KL}}(p_\star \,\|\, p_\theta),$$
$$h(p_\star) = -\int p_\star(x) \log p_\star(x)\, dx,$$

これは和を積分に置き換え、同じ項を足して引く議論から導かれる。**微分エントロピー**（differential entropy）$h(p_\star)$ は θ に依存しないため、母集団レベルでの最尤法は連続の場合もやはり KL ダイバージェンスを最小化する。なお、n 個の異なる実数値の経験分布は密度をもたないため、離散の場合の有限標本における恒等式に対応するものは連続データには存在しない。

36.6.2.3 ガウスノイズと二乗誤差

ターゲットに含まれるノイズのモデルを指定すれば、標準的な回帰損失も同じ原理から直ちに現れる。2.1.3 節では線形回帰についてこの関係を導いたが、その導出はモデルが線形であることに一切依存していない。各ターゲットがモデルの予測に**固定された**分散をもつガウスノイズを加え

たものに等しい、すなわち $y_i \sim \mathcal{N}(\hat{y}_i, \sigma^2)$（ただし $\hat{y}_i = f_\theta(\mathbf{x}_i)$）であるとしよう。ガウス密度 $(2\pi\sigma^2)^{-1/2}\exp\!\big(-(y-\hat{y})^2/(2\sigma^2)\big)$ の負の対数を n 個の独立な例にわたって足し合わせると、次式が得られる。

$$-\log \prod_{i=1}^{n} \mathcal{N}(y_i;\, \hat{y}_i,\, \sigma^2) = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \frac{n}{2}\log(2\pi\sigma^2). \tag{36.82}$$

分散 σ^2 は固定されているため、第 2 項は θ に関して定数であり、先頭の係数 $1/(2\sigma^2)$ も正の定数である。したがって、この NLL の最小化は $\sum_i (y_i - \hat{y}_i)^2$ の最小化と同値である。すなわち二乗誤差による回帰は、残差がガウス分布に従うという仮定のもとでの最尤法にほかならない。同じ手順により、以下に示す 4 つをはじめとする多くのなじみ深い損失が、条件付き分布 $p(y \mid \mathbf{x})$ の NLL と結びつく。各行は分布の負の対数をとり、θ に依存しない定数を落とすことで導かれる。

ノイズモデル $p(y \mid \mathbf{x})$	負の対数尤度 $-\log p(y \mid \mathbf{x})$	深層学習の損失
ガウス分布、固定分散 σ^2	$\frac{1}{2\sigma^2}(y-\hat{y})^2 + \text{const}$	平均二乗誤差（MSE）
ベルヌーイ分布	$-y\log\hat{p} - (1-y)\log(1-\hat{p})$	二値交差エントロピー（BCE）
カテゴリカル分布（ワンホット y）	$-\sum_k y_k \log \hat{p}_k$	ソフトマックス交差エントロピー
ラプラス分布、固定スケール s	$\frac{1}{s}\lvert y-\hat{y}\rvert + \text{const}$	平均絶対誤差（MAE）

これらの損失のいずれにおいても、損失関数を選ぶことはノイズモデルを選ぶことである。ただし、有用な損失のすべてがこの形で導かれるわけではない。サポートベクターマシンのヒンジ損失にはモデルの出力に依存しない正規化定数が存在しないため、加法定数の違いを除いてもいかなる条件付き分布の NLL にもならない（演習問題 3）。

ノイズモデルの選択は、データに外れ値が 1 つ混入した瞬間に決定的な違いを生む。5 つのセンサ測定値 $\{1, 2, 3, 4, 100\}$ に単一の定数 c を当てはめる場面を考えよう。ここで最後の測定値は故障したセンサから得られたものである。ガウスノイズのもとでは $\frac{1}{2}\sum_i (y_i - c)^2$ を最小化し、その導関数 $-\sum_i (y_i - c)$ は標本平均 $c = 22$ でゼロになる。一方、ラプラスノイズのもとでは $\sum_i |y_i - c|$ を最小化する。これは V 字型の曲線の和であり、c における傾きは c より小さい測定値の個数から大きい測定値の個数を引いた値に等しい。この傾きは中央値の左側では負、右側では正となるため、最小点は中央値 $c = 3$ に位置する。もし 5 番目の測定値が 5 であったなら、どちらのノイズモデルも 3 を返していたはずである。Figure 36.31 が示すとおり、異常な測定値はガウスモデルによる当てはめを 22 まで大きく引きずる一方、ラプラスモデルによる当てはめにはまったく影響を与えない。

36.6.3 最尤推定量の精度

ここで冒頭の問いに戻ろう。$9/13$ という値にどれほど確信をもってよいのだろうか。標本比率に対しては、36.5 節で標準誤差 $\sqrt{\hat{\theta}(1-\hat{\theta})/n}$ を与えた。これは標本抽出を繰り返したときの $\hat{\theta}$ の標準偏差を推定する量である。しかし一般のモデルでは MLE は単純な標本平均ではないため、損失そのものからエラーバーを直接読み取る一般的な方法が必要になる。その役割を担うのが、コインの例で登場した例ごとの勾配と、NLL の曲率という 2 つの要素である。

36.6.3.1 スコア関数とフィッシャー情報量

1 つの観測に対する**スコア関数**（score、値としては単に**スコア**とも呼ぶ）とは、対数確率のパラメータに関する勾配である。

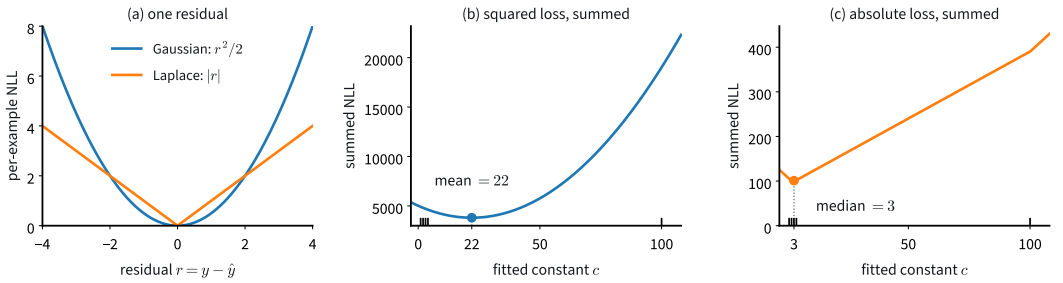


Figure 36.31: 測定値 $1, 2, 3, 4, 100$ に対する単一の定数 c の当てはめ。(a) 残差の関数としての例ごとの負の対数尤度。ガウスノイズでは二次関数、ラプラスノイズでは V 字型となる。(b) 二乗損失の和は放物線をなし、その最小点である平均 22 は外れ値の方向へ大きく引き寄せられる。(c) 絶対誤差損失の和は各測定値（黒の目盛り）で折れ曲がりをもち、中央値 3 で最小値に達する。各点はどれだけ遠く離れていても、傾き 1 単位分の力でしか c を引っ張らない。

$$s(x;\theta) = \nabla_\theta \log p(x \mid \theta).$$

これはコインにおける試行ごとの勾配を任意の確率モデルへと一般化したものであり、方程式 $\sum_i s(x_i;\hat{\theta}) = \mathbf{0}$ が MLE を定める一階最適性条件となる。

スコアの際立った性質として、真のパラメータにおいてその期待値がゼロになることが挙げられる。理由を確かめるため、正規化条件 $\int p(x \mid \theta)\,dx = 1$ を積分記号の下で微分しよう。

$$\begin{aligned}\mathbf{0} = \nabla_\theta \int p(x \mid \theta)\,dx &= \int \nabla_\theta\, p(x \mid \theta)\,dx \\ &= \int s(x;\theta)\, p(x \mid \theta)\,dx = \mathbb{E}_\theta\big[\, s(x;\theta)\,\big],\end{aligned}$$

ここで 3 つ目の等号には対数微分の恒等式 $\nabla_\theta\, p = p\,\nabla_\theta \log p = p\, s$ を用いた。言葉で言えば、モデルから抽出されたデータにわたって平均をとると、対数尤度の勾配はそのデータを生成したパラメータにおいてゼロになる。スコアの平均がゼロであるため、その共分散行列は二次モーメントに等しい。この共分散こそが**フィッシャー情報量**（Fisher information）である (Fisher 1925)。

$$\begin{aligned}I(\theta) &= \mathrm{Var}_\theta\big[\, s(x;\theta)\,\big] = \mathbb{E}_\theta\big[\, s(x;\theta)\, s(x;\theta)^\top\,\big] \\ &= -\,\mathbb{E}_\theta\big[\nabla_\theta^2 \log p(x \mid \theta)\big].\end{aligned} \tag{36.83}$$

中央の等号は平均がゼロである性質を反映している。最後の等号は**情報量恒等式**（information identity）であり、スコアの分散が NLL の期待曲率に等しいことを表す。これは $\mathbb{E}_\theta[s] = \mathbf{0}$ を再び積分記号の内部で微分することで導かれる。積の微分法則により、微分は 2 つの項に分かれる。

$$\begin{aligned}\mathbf{0} &= \nabla_\theta \int s(x;\theta)\, p(x \mid \theta)\,dx \\ &= \int \nabla_\theta^2 \log p(x \mid \theta)\; p(x \mid \theta)\,dx + \int s(x;\theta)\,\big(\nabla_\theta\, p(x \mid \theta)\big)^\top dx.\end{aligned}$$

第 2 の積分に $\nabla_\theta p = p\, s$ を代入すると $\mathbb{E}_\theta[\, s\, s^\top]$ となり、次式が得られる。

$$\mathbb{E}_\theta\big[\nabla_\theta^2 \log p(x \mid \theta)\big] + \mathbb{E}_\theta\big[\, s\, s^\top \,\big] = \mathbf{0},$$

これを移項すれば、直ちに Equation 36.83 の最後の等号が得られる。直感的には、鋭く湾曲した NLL はパラメータを強く絞り込むのに対し、平坦な NLL はパラメータを曖昧なままにする。フィッシャー情報量は、モデルから抽出されたデータにわたって平均した、観測 1 個あたりの NLL の曲率を測っている。

上の 2 つの導出では、断りなく微分を積分の内側へ入れた。以降の結果はいくつかの**正則条件**（regularity conditions）に依拠しており、ここで一度にまとめておく。

- 密度が θ に関して 2 回連続微分可能である。
- 密度の台（support）、すなわち $p(x \mid \theta) > 0$ となる x の集合が θ に依存しない。
- 微分と積分の順序交換が可能である（34.7 節の手法）。これが上の各導出における 2 つ目の等号を正当化しており、台が動く場合にはこの段階が破綻する。
- 真のパラメータがパラメータ空間の内部に存在する。
- フィッシャー情報量が有限かつ非特異（正則）である。

フィッシャー情報量が特異になる場合や、台がパラメータとともに動く場合に具体的に何が破綻するかは、演習問題 6 および 7 で確認する。

複雑さの異なる 3 つのモデルについて順に $I(\theta)$ を計算してみよう。第 1 にコインでは、1 回の試行 $x \in \{0, 1\}$ の対数尤度は $\log p(x \mid \theta) = x \log\theta + (1 - x)\log(1 - \theta)$ であるため、そのスコアと二階導関数は次式となる。

$$s(x;\theta) = \frac{x}{\theta} - \frac{1 - x}{1 - \theta}, \qquad \frac{d}{d\theta}\, s(x;\theta) = -\frac{x}{\theta^2} - \frac{1 - x}{(1 - \theta)^2}.$$

$-\mathbb{E}[\,\cdot\,]$ をとり $\mathbb{E}[x] = \theta$ を用いれば、1 試行あたりのフィッシャー情報量と、その逆数を n で割った値が得られる。

$$I(\theta) = \frac{1}{\theta} + \frac{1}{1 - \theta} = \frac{1}{\theta(1 - \theta)}, \qquad \frac{I(\theta)^{-1}}{n} = \frac{\theta(1 - \theta)}{n}. \tag{36.84}$$

1 試行あたりの情報量が $\theta = 1/2$ で最小になることに注目しよう。各試行がコインの偏りについて運ぶ情報量が最も少ないため、公平なコインが最も特定しにくいのである。

次に、2 パラメータのモデルで行列形式を見てみよう。平均 μ、標準偏差 σ のガウス分布では、1 つの観測の NLL は定数項を除いて $\log\sigma + (x - \mu)^2/(2\sigma^2)$ である。μ に関する二階導関数は $1/\sigma^2$ である一方、σ に関する二階導関数は $-1/\sigma^2 + 3(x - \mu)^2/\sigma^4$ であり、その期待値は $2/\sigma^2$ となる。交差偏導関数 $2(x - \mu)/\sigma^3$ の期待値はゼロであるため非対角成分は消え、$I(\mu, \sigma) = \mathrm{diag}(1/\sigma^2,\, 2/\sigma^2)$ となる。NLL は σ 方向には μ 方向の 2 倍鋭く湾曲しており、これはデータが平均よりもスケールの方を強く絞り込むことを意味する。Figure 36.32 に 30 個のサンプルに対するこのお椀型の曲面を示す。

コインでもガウス分布でも MLE は標本平均の形になるため、フィッシャー情報量を使わなくてもそのばらつきを計算できた。理論の真価が表れるのは、MLE が平均にならないモデルである。**コーシー位置モデル**（Cauchy location model）は次の密度をもつ。

$$p(x \mid \theta) = \frac{1}{\pi\big(1 + (x - \theta)^2\big)},$$

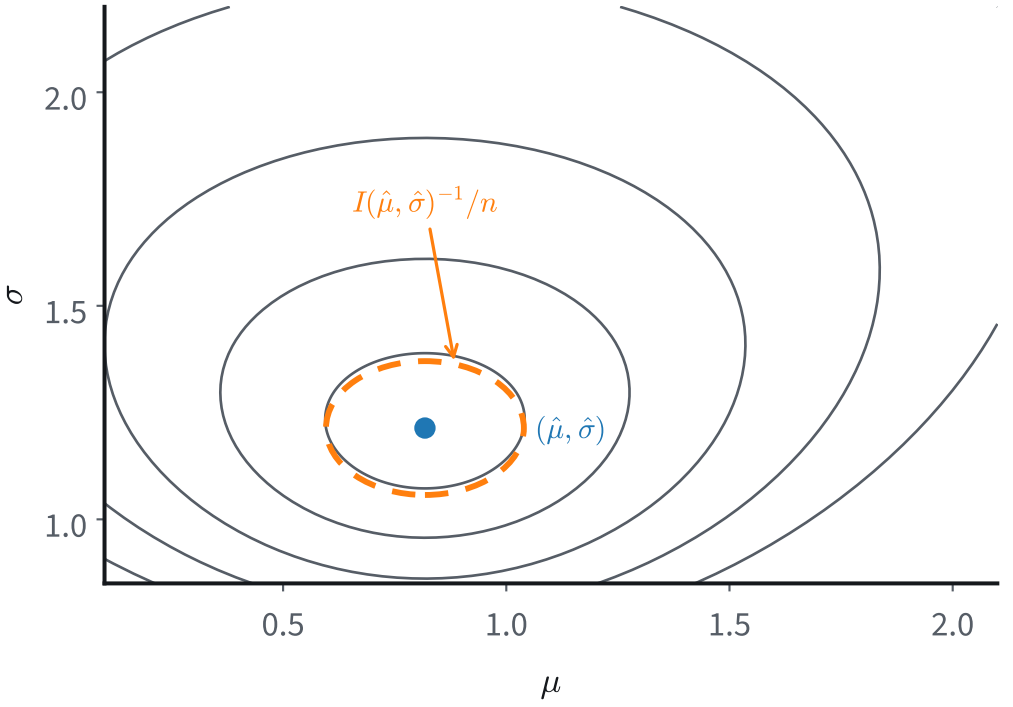


Figure 36.32: ガウス分布のパラメータ (μ, σ) に対する負の対数尤度（$n = 30$ 個のサンプルを $\mathcal{N}(1, 1.5^2)$ から抽出、点は MLE を示す）。破線の楕円は $I(\hat{\mu}, \hat{\sigma})^{-1}/n$ を表しており、NLL が最小値より半分だけ高い最も内側の等高線をよく捉え、μ 方向には σ 方向よりも広くなっている。

これは θ を中心とする対称な釣鐘型の分布であるが、36.1 節で示したとおり裾が極めて重いため平均が存在しない。$u = x - \theta$ とおくと、スコアは $s = 2u/(1+u^2)$、その導関数は $\partial_\theta s = -2(1-u^2)/(1+u^2)^2$ となる。三角関数による置換 $u = \tan\varphi$ を使えば、フィッシャー情報量は一目で求まる。$d\varphi = du/(1+u^2)$ より、標準コーシー変数の角度 φ は $(-\pi/2, \pi/2)$ 上の一様分布に従い、次式が成り立つ。

$$s = \frac{2\tan\varphi}{1+\tan^2\varphi} = \sin 2\varphi, \qquad \text{so} \qquad I = \mathbb{E}\big[\sin^2 2\varphi\big] = \frac{1}{2}.$$

したがって、コーシー分布の各観測はその中心について半単位分の情報量をもっており、これは $\sigma^2 = 2$ のガウス分布の観測と同じである。また、スコアの絶対値は決して 1 を超えず、遠く離れた点に対してはゼロへと減衰することにも注目しよう。極端な外れ値が MLE をほとんど動かさないのはこのためである。

閉形式で解く工夫が使えない場合でも、自動微分によって I の両方の表現を直接計算できる。下のセルでは、$\theta^\star = 0$ でコーシー分布から 10,000 個の観測を抽出し、スコアの平均、スコアの分散、および負の平均曲率を表示する。一様な角度表現を用いて計算すると、これら 3 つの推定値のモンテカルロ標準誤差はそれぞれ約 0.007、0.004、0.008 となる。

```
def log_p(theta, x):  # log density of the Cauchy location model
    return -torch.log1p((x - theta)**2) - np.log(np.pi)

x = torch.tensor(np.random.default_rng().standard_cauchy(10_000),
                 dtype=torch.float32)  # theta* = 0
score = torch.func.vmap(torch.func.grad(log_p), in_dims=(None, 0))
curv = torch.func.vmap(torch.func.grad(torch.func.grad(log_p)),
                       in_dims=(None, 0))
s, h = score(torch.tensor(0.0), x), curv(torch.tensor(0.0), x)
print(f'mean score {s.mean():.3f}, variance of score {s.var():.3f}, '
      f'-mean curvature {-h.mean():.3f}')
```

```
mean score 0.004, variance of score 0.498, -mean curvature 0.502
```

`torch.func.grad` が 1 つの観測の対数密度を微分し、`vmap` がそれを全 10,000 個の観測にわたって適用する。`grad` を 2 回適用すれば曲率が得られる。

3 つの標本推定値はいずれも、理論値である 0、0.5、0.5 から標準誤差の 2〜3 倍以内に収まっている。

36.6.3.2 コインのエラーバー

フィッシャー情報量を使えば、本節の冒頭で求めたかった $9/13$ のエラーバーが手に入る。次項では、大きな n に対して MLE の分散が近似的に $1/\big(nI(\theta)\big)$ となることを示す。ひとまずその結果を認めることにすれば、直ちに適用できる。コインでは Equation 36.84 より $1/(nI) = \theta(1-\theta)/n$ であり、$\hat\theta$ を未知の θ に代入すると推定標準誤差は次式となる。

$$\widehat{\mathrm{se}} = \sqrt{\frac{\hat\theta(1-\hat\theta)}{n}} = \sqrt{\frac{(9/13)(4/13)}{13}} = 0.128.$$

したがって $\hat{\theta} = 0.69 \pm 0.13$ となる。近似的な 95% 信頼区間 $\hat{\theta} \pm 1.96\,\widehat{\mathrm{se}} = [0.44,\, 0.94]$ は余裕をもって $1/2$ を含んでおり、13 回の試行ではこのコインを公平なコインと区別することはできない。95% の信頼度で θ を ± 0.05 の幅まで絞り込むには、およそ $1.96^2\,\hat{\theta}(1-\hat{\theta})/0.05^2 = 1.96^2 \cdot 0.213/0.05^2 \approx 330$ 回の試行が必要になる。

重要なのは、損失そのものからまったく同じ数値が直接出てくることである。コインの NLL の二階導関数を $\hat{\theta}$ で評価すると、次式となる。

$$\ell''(\hat{\theta}) = \frac{n_H}{\hat{\theta}^2} + \frac{n_T}{(1-\hat{\theta})^2} = \frac{9}{(9/13)^2} + \frac{4}{(4/13)^2} = 61.0,$$

そして $1/\sqrt{61.0} = 0.128$ である。コインの場合、この一致は厳密である。なぜなら $n_H = n\hat{\theta}$ かつ $n_T = n(1-\hat{\theta})$ より、観測された曲率は $n/\big(\hat{\theta}(1-\hat{\theta})\big) = nI(\hat{\theta})$ にちょうど等しいからである。正則条件を満たす任意のモデルにおいて、大きな n では同じ関係が近似的に成り立つ。モデルを当てはめた後、最小点における NLL の曲率（ヘッセ行列）を計算し、その平方根の逆数をとればエラーバーになる。パラメータがベクトルの場合、個々のパラメータのエラーバーは逆ヘッセ行列の対角成分の平方根である。深層ニューラルネットワークはこれらの正則条件に違反しており、この手順のうち何が有効なまま残るかについては 36.6.3.5 節で説明する。また、本節の冒頭で見た尤度のプロットも曲率から説明できる。データ量が 10 倍になると曲率も 10 倍になり、ピークの幅は $\sqrt{10} \approx 3.2$ 倍細く絞られるのである。

36.6.3.3 一致性と漸近正規性

なぜ MLE の分散は近似的に $1/(nI)$ になるのだろうか。これに答えるには、まずそもそも MLE が正しい値へ向かうことの保証が必要であり、次にその目標値の周りでどのように揺らぐかを記述する必要がある。これまでの節で学んだ 3 つの道具がその役割を果たす。36.5 節からは一致性（consistency、$\hat{\theta} \xrightarrow{P} \theta^\star$）と弱大数の法則を用いる。また 36.3 節からは中心極限定理を用いる。これは、平均ゼロ・分散 v の独立同分布な n 個の項の和を $\sqrt{n}$ で割ったものが、$\mathcal{N}(0, v)$ に分布収束（$\xrightarrow{d}$）するという定理である。

MLE が真のパラメータに近づく理由は、36.6.2.1 節の交差エントロピー恒等式から説明できる。データが真のパラメータ $\theta^\star$ をもつモデルから抽出されているとしよう。平均 NLL は独立同分布な項の平均であるため、大数の法則により母集団 NLL へと収束する。

$$\mathbb{E}_{\theta^\star}\big[-\log p(x \mid \theta)\big] = H(p_{\theta^\star}) + D_{\mathrm{KL}}(p_{\theta^\star} \,\|\, p_\theta),$$

ここで連続密度の場合は、微分エントロピー h を H の代わりに用いる（36.6.2.2 節）。ギブスの不等式（36.4.2 節）により、この目的関数は KL ダイバージェンスがゼロになるところ、すなわち $\theta = \theta^\star$ で最小値を達成する。モデルが**識別可能**（identifiable）であるとき、つまり異なるパラメータベクトルが常に異なる分布を定めるとき、この最小点は一意である。ニューラルネットワークは識別可能ではない。ある層の隠れユニットを、そこに入出力される重みごと並べ替えれば、計算される関数を一切変えずにパラメータベクトルだけが変わるからである。そのようなモデルにおいて一致性とは、単一のパラメータベクトルについての主張ではなく、当てはめられた分布、あるいはその分布を生み出すパラメータの同値類についての主張となる。

一致性の保証を証明なしで述べておこう (Wasserman 2013)。次の 4 つの条件が成り立つとき、MLE は一致推定量となる。すなわち、データが $p_{\theta^\star}$ からの独立同分布であること、モデルが識別可能であること、母集団の最小点が十分に分離されており θ が $\theta^\star$ から離れている限り最適値にいくらでも近づくことはないこと（コンパクトなパラメータ集合上で目的関数が連続であれば一意

性で十分である)、そして平均 NLL が θ に関して母集団 NLL に一様収束することである。一様性は見落とされがちな微妙な要件である。大数の法則が保証するのは固定された各 θ での収束であり、各点収束だけでは最小点の収束までは導けない。もちろんコインについては定理に頼るまでもなく、大数の法則から直接 $\hat{\theta} = n_H/n \to \theta^\star$ が従う。

一致性から $\hat{\theta}$ の行き先はわかる。エラーバーを得るには、$\theta^\star$ の周りでの揺らぎの大きさと形状が必要である。スカラーパラメータ θ の場合、導出は 3 つの短いステップで済み、唯一の巧妙な着想は NLL そのものではなく一階最適性条件をテイラー展開することにある。合計スコアを $S_n(\theta) = \sum_i s(x_i;\theta)$ と書こう。

I. 定義より、MLE は $S_n(\hat{\theta}) = 0$ を満たす。$S_n(\hat{\theta})$ を $\theta^\star$ の周りで一次まで展開すると、次式が得られる。

$$0 = S_n(\hat{\theta}) \approx S_n(\theta^\star) + S_n'(\theta^\star)\,(\hat{\theta} - \theta^\star).$$

II. 推定誤差について解き $\sqrt{n}$ を掛けると、2 つの平均の比として表せる。

$$\sqrt{n}\,(\hat{\theta} - \theta^\star) \approx \frac{S_n(\theta^\star)/\sqrt{n}}{-S_n'(\theta^\star)/n}.$$

III. 分子は、平均ゼロ・分散 $I(\theta^\star)$ をもつ独立同分布な n 個のスコアの和を $\sqrt{n}$ で割ったものであり、中心極限定理により $\mathcal{N}\big(0, I(\theta^\star)\big)$ に分布収束する。分母は独立同分布な n 個の項 $-\partial_\theta s(x_i;\theta^\star)$ の平均であり、情報量恒等式からその期待値は $I(\theta^\star)$ に等しいため、大数の法則により $I(\theta^\star)$ に収束する。分散 I のガウス変数を定数 I で割れば、分散 $I/I^2 = 1/I$ のガウス分布が得られる。

フィッシャー情報量の 2 つの表現がここで見事に出会うことに注目しよう。分子ではスコアの分散が使われ、分母では期待曲率が使われている。情報量恒等式によって両者が同じ値 I に一致するからこそ、最終的な公式はこれほど簡潔になるのである。ベクトルの場合も行列を用いてまったく同じ代数計算をたどることで、共分散 $I^{-1} I I^{-1} = I^{-1}$ が得られる。厳密な証明ではテイラー展開の剰余項も上から評価する必要があり、下の命題における可積分な曲率の上界がその役割を果たす。

命題（漸近正規性）正則条件、上記の一致性の条件、および上界 $\|\nabla_\theta^2 \log p(x \mid \theta)\| \le M(x)$（ただし $\mathbb{E}_{\theta^\star}[M(x)] < \infty$ がすべての θ（$\theta^\star$ の近傍）で成り立つ）のもとで、次が成立する。

$$\sqrt{n}\,(\hat{\theta} - \theta^\star) \;\xrightarrow{d}\; \mathcal{N}\!\big(\mathbf{0},\; I(\theta^\star)^{-1}\big) \tag{36.85}$$

出典は (Wasserman 2013) である。上の 3 ステップ以上の証明には立ち入らないが、これが本節の中心結果である。エラーバー、MLE の有効性、そして理論の限界はすべてこの結果の上に成り立っている。平たく言えば、MLE の推定誤差は近似的にガウス分布に従い、$1/\sqrt{n}$ の速さで縮小し、その共分散は逆フィッシャー情報量で与えられる。有限の n では近似 $\mathrm{Cov}(\hat{\theta}) \approx I(\theta^\star)^{-1}/n$ を用い、これがコインに適用したエラーバーであった。とりわけ、エラーバーを半分にするには 4 倍のデータが必要になる。モデルが誤特定されている場合には、I の分散表現と曲率表現がもはや一致せず、共分散は 36.6.3.5 節の「サンドイッチ」公式へと置き換わる。

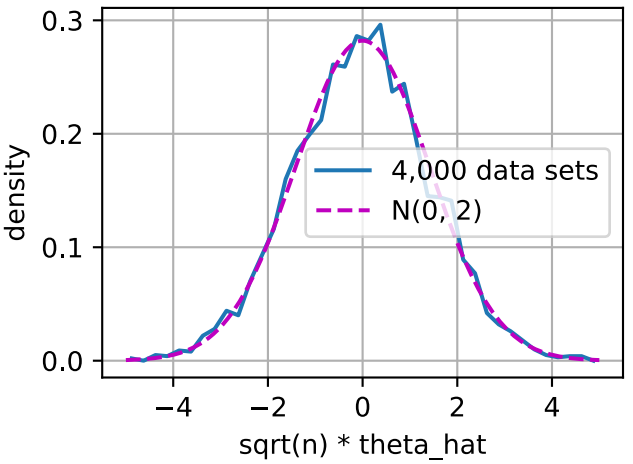
非自明な設定で漸近正規性を検証するには、MLE が標本平均にならないモデルが必要であり、コーシー位置モデルがまさにうってつけである。その尤度方程式 $\sum_i 2u_i/(1+u_i^2) = 0$ には閉形式解がないため、観測された曲率を期待曲率 $nI = n/2$ に置き換えたニュートン法である**フィッシャースコアリング**（Fisher scoring）を用いて数値的に解く。

$$\theta \leftarrow \theta + \frac{S_n(\theta)}{nI} = \theta + \frac{2}{n}\, S_n(\theta).$$

観測曲率を nI に置き換えるのは単なる計算上の便宜にとどまらない。期待曲率は常に正であるのに対し、コーシー分布の NLL の観測曲率は最小点から離れると負になりうるため、素のニュートンステップでは山を登る方向へ進んでしまうからである。ここではすでに解に近い標本中央値を各実行の初期値とし、$\theta^\star = 0$ で生成した $n = 100$ 個の観測からなる 4,000 個のデータセットに対し、それぞれ 10 回のスコアリングステップで当てはめを行う。$n = 100$ の時点ですでに漸近近似が正確であれば、$\sqrt{n}\,\hat{\theta}$ のヒストグラムは $\mathcal{N}(0, 1/I) = \mathcal{N}(0, 2)$ に一致するはずである。

```
reps, n = 4000, 100
x = np.random.default_rng().standard_cauchy((reps, n))  # theta* = 0
theta = np.median(x, axis=1)
for _ in range(10):  # Fisher scoring: theta += S_n / (n I) with I = 1/2
    u = x - theta[:, None]
    theta += (2 / n) * (2 * u / (1 + u**2)).sum(axis=1)
z = np.sqrt(n) * theta  # sqrt(n) (theta_hat - theta*)
hist, edges = np.histogram(z, bins=40, range=(-5, 5), density=True)
grid = np.linspace(-5, 5, 201)
d2l.plot([(edges[:-1] + edges[1:]) / 2, grid],
         [hist, np.exp(-grid**2 / 4) / np.sqrt(4 * np.pi)],
         'sqrt(n) * theta_hat', 'density', legend=['4,000 data sets', 'N(0, 2)'])
print(f'standard deviation of sqrt(n) * theta_hat: {z.std():.2f}')
```

```
standard deviation of sqrt(n) * theta_hat: 1.43
```



ヒストグラムは中心、幅、高さのいずれにおいても理論曲線によく一致している。実行ごとの $\sqrt{n}\,\hat{\theta}$ の標準偏差はおよそ 1.40〜1.47 の間に入り、漸近極限 $\sqrt{2} \approx 1.41$ に近い。なお $n = 100$ では、分散はまだ極限値より数パーセント高いところにある。

36.6.3.4 クラメール・ラオの下界

$1/(nI)$ という分散は優れた値なのだろうか、それともももっと巧妙な推定量ならこれを上回れるのだろうか。実は、フィッシャー情報量は「平均的に正しい」いかなる推定量も突破できない厳格

な下界をも定めている。36.5 節で見たとおり、すべての θ に対して $\mathbb{E}_\theta[\tilde{\theta}] = \theta$ が成り立つとき、推定量 $\tilde{\theta}$ は**不偏**（unbiased）であった。

命題（クラメール・ラオの下界） 正則条件のもとで、独立同分布な n 個の観測からスカラーパラメータを推定する任意の不偏推定量 $\tilde{\theta}$ は次を満たす。

$$\operatorname{Var}(\tilde{\theta}) \;\geq\; \frac{1}{n\,I(\theta)} \tag{36.86}$$

出典は (Cramér 1946; Radhakrishna Rao 1945) である。言い換えれば、いかなる不偏推定量も、尤度の曲率が許す以上に高い精度を達成することはできない。証明はわずか 3 行で済み、その唯一の着想は推定量とスコアの相関をとることである。

証明 データセット全体のスコアを $S_n = \partial_\theta \log P(X \mid \theta)$ とおく。これは独立な例ごとのスコア n 個の和であり、平均ゼロ、分散 $nI(\theta)$ をもつ。不偏性より $\int \tilde{\theta}(X)\, P(X \mid \theta)\, dX = \theta$ が成り立つ。両辺を積分記号の下で微分し、$\partial_\theta P = P\, S_n$ を代入すると次式を得る。

$$1 = \int \tilde{\theta}(X)\, S_n\, P(X \mid \theta)\, dX = \mathbb{E}\big[\tilde{\theta}\, S_n\big] = \operatorname{Cov}(\tilde{\theta},\, S_n),$$

ここで最後の等号には $\mathbb{E}[S_n] = 0$ を用いた。共分散に対するコーシー・シュワルツの不等式を適用すれば、$1 \leq \operatorname{Var}(\tilde{\theta})\operatorname{Var}(S_n) = \operatorname{Var}(\tilde{\theta})\, nI(\theta)$ が得られる。$\blacksquare$

不偏性は左辺の「1」を通じて効いており、固定された台は積分記号の下で微分する段階で使われている。演習問題 7 では、台が θ とともに動くときにこの下界を上回る不偏推定量の具体例を示す。ベクトルパラメータに対しては、半正定値順序のもとで下界は $\operatorname{Cov}(\tilde{\theta}) \succeq I(\theta)^{-1}/n$ へと一般化される。

MLE の漸近分散 Equation 36.85 はこの下界に厳密に一致しており、この意味で MLE は**漸近有効**（asymptotically efficient）である。すなわち大標本極限において、これより小さい分散を達成する不偏推定量は存在しない。コインの場合、n_H/n は分散 $\theta(1-\theta)/n = 1/(nI)$ をもつ不偏な標本平均であるため、極限だけでなくあらゆる有限の n でこの下界が達成される。しかし一般には、MLE が下界に到達するのは $n \to \infty$ においてのみである。有限の n では統計的バイアスをもちうるため、その場合は下界の対象外となる。次にガウス分布の分散で見るとおりである。

漸近有効性は実務においてどれほど重要だろうか。コーシー位置モデルでは、標本中央値が自然な対抗馬であり、標本平均は素朴な推定量である。対称性により、MLE と中央値はいずれも不偏である。分散の n 倍に対するクラメール・ラオの下界は $1/I = 2$ であるのに対し、中央値の漸近理論は $\pi^2/4 \approx 2.47$ を予測する。下の出力を見る前に、標本平均がどうなるか少し予想してみてほしい。このセルでは先ほどの $4{,}000$ 個のデータセットを再利用する。

```
print(f'n Var(MLE) = {n * theta.var():.2f}, '
      f'n Var(median) = {n * np.median(x, axis=1).var():.2f}, '
      f'n Var(mean) = {n * x.mean(axis=1).var():.3g}')
```

```
n Var(MLE) = 2.06, n Var(median) = 2.50, n Var(mean) = 7.79e+05
```

MLE はおよそ 1.9〜2.2 の間に入り、下界である 2 のすぐ近くに収まる一方、中央値は 2.5 付近にある。つまり MLE は約 20% 少ない観測数で同じエラーバーを達成する。対照的に、標本平均は実行ごとに激しく暴れる途方もない分散を示す。独立な n 個のコーシー変数の平均は依然として同じコーシー分布に従うため、平均をとっても一切改善しないのである。

36.6.3.5 理論の限界

ここまでの保証は、前述の正則条件のもと、かつ $n \to \infty$ の極限において成り立つものである。以下の最初の 4 つのケースはいずれもそれらの条件の 1 つを破っており、演習問題で詳しく扱う。5 つ目のケースである誤特定されたモデルは、深層学習のすべての実務家が身を置く状況であるため、ここで詳しく検討する。

- **有限標本での統計的バイアス**。ガウス分布の分散の MLE は残差二乗和を n で割るため、平均的に σ^2 を $(n-1)/n$ 倍だけ過小評価する（36.5 節）。MLE は極限において最適なのであり、あらゆる n で不偏なわけではない。
- **境界**。偏り 0.95 のコインのように真のパラメータが定義域の端近くにある場合、ガウス近似が正確になるまでにはるかに多くのデータを要する。精度を支配するのは n 単体ではなく、まれな結果の期待出現回数だからである。境界上そのものでは、MLE が存在しないことすらある（演習問題 4）。
- **パラメータとともに動く台**。$[0, \theta]$ 上の一様分布では、MLE は $\max_i x_i$ である。その誤差は $1/\sqrt{n}$ ではなく $1/n$ の速さで縮小し、不偏になるよう定数倍した推定量はクラメール・ラオの下界を上回る（演習問題 7）。
- **特異な情報行列**。ReLU ネットワークのリスケーリング対称性によりフィッシャー情報行列 I は零ベクトルをもち、したがって I^{-1} は存在しない（演習問題 6）。

モデルの誤特定（misspecification）。モデル族のいかなる要素 p_θ も真のデータ生成分布と一致しないとき、MLE は真の分布からの KL ダイバージェンスを最小化するパラメータへと収束する (White 1982)。その極限でもスコアの期待値は依然としてゼロになるが、スコアの分散 B と NLL の期待曲率 A はもはや一致しない。その結果、先ほどの導出のステップ III から得られる分散は B/A^2（$1/I$ ではなく）となり、行列形式では**サンドイッチ**共分散（sandwich covariance）$\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}$ となる（演習問題 8）。たとえば、平均 m、分散 $m + m^2/\alpha$ をもつ過分散なカウントデータに、レート λ のポアソンモデルを当てはめる場合を考えよう。ポアソン分布の MLE は標本平均であるため、依然として m を復元し、その真の分散は $(m + m^2/\alpha)/n$ である。ところがポアソンモデルのフィッシャー情報量 $I = 1/\lambda$ が予測する分散は m/n であり、これは $1 + m/\alpha$ 分の 1、すなわち 3 分の 1（$m = 4$、$\alpha = 2$ のとき）だけ過小になっている。

深層ニューラルネットワークは誤特定されており、かつ識別不能でもあるため、個々の重みに古典的なエラーバーを付けることはできない。それでも理論が裏づけるのは、当てはめられた予測分布に関する主張である。パラメータの対称性は予測分布を変えないからである。なお、フィッシャー情報行列そのものは本書の後半で 2 度再登場する。1 度目はパラメトリックな分布族における近傍の分布間の KL ダイバージェンスの局所曲率（ヘッセ行列）として（37.3 節）、2 度目は自然勾配最適化で用いられる前処理行列としてである（35.5 節）。

36.6.4 MAP 推定と正則化

13 回のコイン投げからは妥当な推定値が得られたが、3 回のコイン投げでは最尤法が完全に破綻しうる。3 回投げて HHH が出たとしよう。MLE は $\hat{\theta} = 3/3 = 1$ となり、裏に確率ゼロを割り当てるため、その後に裏が 1 度でも観測された瞬間に無限大の NLL を被ってしまう。訓練中に一度も現れなかった特徴量とクラスの組み合わせに遭遇したカウントベースの分類器も、これとまったく同じ病的な現象に直面する。コインを手にしたことのある人なら誰一人として、3 回表が出たからといって裏が不可能だとは考えない。データを見る前の段階で「コインが完全に片面しか出ないことはめったにない」という知識を原理に基づいて表現する方法が必要である。

パラメータ上の確率分布を導入すれば、そのような事前知識を捉えられる。最尤法はデータセットにしか確率を割り当てなかったため、事前知識を取り入れる余地がなかった。**事前分布** $p(\theta)$ を導入すると、ベイズの定理は事前分布と尤度を組み合わせ、観測データが与えられたもとでのパラメータ上の分布を与える。

$$p(\theta \mid X) = \frac{P(X \mid \theta)\, p(\theta)}{P(X)}, \qquad P(X) = \int P(X \mid \theta)\, p(\theta)\, d\theta. \tag{36.87}$$

左辺は**事後分布**（posterior）であり、データを見た後に更新された信念を表す。分母は**エビデンス**、すなわちすべてのパラメータ値にわたって平均をとったデータの周辺確率であり、決定的に重要な点として θ に依存しない。**最大事後確率推定**（maximum a posteriori、MAP）による推定値（MAP 推定値）とは、事後分布の最頻値（ピーク）である。エビデンスは θ に関して定数であるため、事後分布を最大化して負の対数をとると次式が得られる。

$$\begin{aligned} \hat{\theta}_{\text{MAP}} &= \operatorname*{argmax}_{\theta}\; P(X \mid \theta)\, p(\theta) \\ &= \operatorname*{argmin}_{\theta} \Big[\underbrace{-\textstyle\sum_i \log p(x_i \mid \theta)}_{\text{data fit (NLL)}} \; \underbrace{-\log p(\theta)}_{\text{penalty}} \Big]. \end{aligned} \tag{36.88}$$

したがって MAP 推定は、おなじみの NLL に加法的ペナルティである負の対数事前分布を加えたものを最小化する。極端なパラメータ値に低い確率を割り当てる事前分布はそうした値にペナルティを科し、まさしく正則化項の役割を果たす。HHH に対して、仮想的な表 1 回と裏 1 回を加えることに相当する事前分布（後述する $\text{Beta}(2,2)$ 事前分布）を用いれば、$\hat{\theta}_{\text{MAP}} = (3+1)/(3+2) = 4/5$ が得られる。このとき、続いて裏が出たときのコストは $-\log(1/5) = 1.6$ nat となり、無限大ではなく有限の驚きで済む。

36.6.4.1 ガウス事前分布と重み減衰

深層学習における標準的な正則化手法である重み減衰（weight decay）は、ガウス事前分布がもたらす MAP のペナルティ項である。これが成り立つのは L_2 ペナルティ形式の重み減衰に対してであり、Adam とともに用いられる分離型の変種（9.5.3 節）はもはや厳密な MAP ペナルティではない。2.7 節では線形回帰についてこの対応関係を導いたが、ここではターゲットにガウスノイズを含む任意のモデルに対して本節の記法で記述しよう。各パラメータが事前には小さく互いに独立に $\theta_j \sim \mathcal{N}(0, \tau^2)$ に従うと仮定する。ここで τ は想定される典型的な大きさを表す。事前密度は $\exp\!\big(-\|\theta\|_2^2/(2\tau^2)\big)$ に比例するため、その負の対数は定数項を除いて $\|\theta\|_2^2/(2\tau^2)$ に等しい。これをガウス分布の NLL Equation 36.82 と組み合わせると、MAP の目的関数 Equation 36.88 は次式となる。

$$\hat{\theta}_{\text{MAP}} = \operatorname*{argmin}_{\theta}\; \frac{1}{2\sigma^2} \sum_i (y_i - \hat{y}_i)^2 + \frac{1}{2\tau^2}\|\theta\|_2^2.$$

全体に正の定数 σ^2/n を掛けても最小点は変わらず、2.7 節の重み減衰の目的関数が復元される。

$$\frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\big(y_i - \hat{y}_i\big)^2 + \frac{\lambda}{2}\|\theta\|_2^2, \qquad \lambda = \frac{\sigma^2}{n\tau^2}.$$

本書で通常そうしているようにバイアス b にペナルティを科さないことは、b 上に平坦な事前分布を置くことに対応する。平均損失におけるペナルティ重み λ が $1/n$ に比例して縮小することにも注目しよう。事前分布はデータが少ないときに最も強く効き、標本が大きくなるにつれて消えていく。なお 35.7.3.2 節では、同じペナルティを重みのノルムに対する制約として同値に解釈する。

ガウス事前分布をスケール s の**ラプラス**事前分布 $p(\theta) \propto e^{-\|\theta\|_1/s}$ に置き換えると、L_1 ペナルティ $\|\theta\|_1/s$ が得られる。このペナルティはゼロに鋭い折れ曲がりをもつため、厳密なゼロを取りやすくし、スパースなパラメータベクトルを促す。Figure 36.33 はこの綱引きを 1 次元で描いたものである。NLL は MLE へ向かって引っ張り、負の対数事前分布は事前平均へ向かって引っ張り、MAP 推定値はその中間に落ち着く。

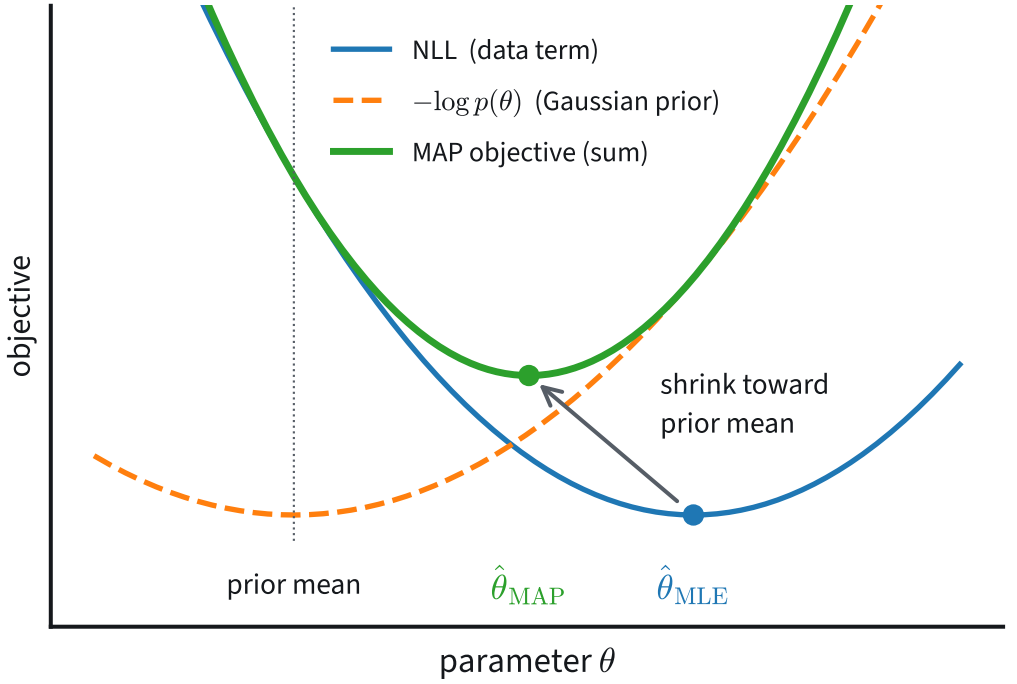


Figure 36.33: 最大事後確率推定はデータと事前分布のバランスをとる。負の対数尤度（青）は MLE で最小化されるが、事前平均を中心とするガウス事前分布の負の対数（オレンジ、破線）を加えると MAP の目的関数（緑）が得られ、最小点が MLE から事前平均の方向へ引き寄せられる。事前分布が狭い（τ が小さい）ほど強く引き寄せられ、逆に $\tau \to \infty$ では事前分布が平坦になって MAP は MLE に帰着する。

同じ図から、2 つの極限的な場合も直ちに理解できる。事前分布が平坦になる（$\tau \to \infty$ または $s \to \infty$）とペナルティは消滅する。また標本サイズ n が大きくなると、事前分布が固定されたまま NLL が n に比例して深くなる。いずれの極限においても、MAP は再び MLE へと収束する。

36.6.4.2 ベータ事前分布と擬似カウント

コインではパラメータ θ が $[0,1]$ 上にあるため、ベータ分布族が自然な事前分布となる。$a, b > 0$ をもつ $\mathrm{Beta}(a,b)$ 事前分布の密度は $\theta^{a-1}(1-\theta)^{b-1}$ に比例する。これにコインの尤度 Equation 36.80 を掛けると指数に観測回数がそのまま加算されるため、事後分布もベータ分布族にとどまる。これは 36.4 節で見た共役更新にほかならない。

$$p(\theta \mid X) \;\propto\; \theta^{\,n_H + a - 1}\,(1-\theta)^{\,n_T + b - 1}.$$

この事後分布は、表 $n_H + a - 1$ 回・裏 $n_T + b - 1$ 回のコインの尤度とまったく同じ関数形を

している。したがって両方の指数が正である限り、その最大化はコインの MLE を求めることと同一であり、「表の指数を合計の指数で割った値」として次式で与えられる。

$$\hat{\theta}_{\text{MAP}} = \frac{n_H + a - 1}{n + a + b - 2}, \qquad n = n_H + n_T. \tag{36.89}$$

つまり事後最頻値（posterior mode）にとっては、経験頻度をとる前に仮想的な表 $a-1$ 回と仮想的な裏 $b-1$ 回を標本に付け加えたかのように事前分布が働く。これは 36.4 節の慣例とも整合している。すなわち a と b は事後平均（posterior mean）の**擬似カウント**（pseudo-counts）であるのに対し、$a-1$ と $b-1$ は事後最頻値の擬似カウントである。$\text{Beta}(2,2)$ 事前分布は最頻値に対して表裏それぞれ 1 回ずつの仮想試行を加え、先ほどの $9/13 \approx 0.692$ を $(9+1)/(13+2) = 2/3$ に、HHH を $4/5$ にシフトさせる。強い事前分布 $\text{Beta}(30,30)$ は、観測された試行回数がその 58 回分の仮想試行を上回るまで、MAP 推定値を $1/2$ 付近に繋ぎ止める。

これら 2 つの要約量は、ある古典的な公式において一致する。平坦な事前分布 $\text{Beta}(1,1)$ のもとでの事後平均はラプラスの継起則 $(n_H+1)/(n+2)$ であり (Laplace 1814)、カウントベースの分類器では 1 加算平滑化（add-one smoothing）として広く知られている。これとまったく同じ比が $\text{Beta}(2,2)$ のもとでの MAP 推定値となる一方、平坦な $\text{Beta}(1,1)$ 事前分布のもとでの MAP は単なる生の MLE を返すにすぎない。したがって、1 加算平滑化は平坦な事前分布のもとでの事後平均であって、平坦な事前分布による MAP 推定値ではない。

36.6.4.3 事後最頻値と事後分布

事後最頻値には微妙な弱点がある。モデルをどのようにパラメータ化するかに依存してしまうのである。非線形な変数変換のもとでは、あらゆる領域の確率が同一に保たれるにもかかわらず、確率密度にはヤコビ行列式（ヤコビアン）の因子が掛かるためピークの位置がずれる。$\text{Beta}(1,1)$ 事前分布は θ に関しては平坦だが、まったく同じ信念を対数オッズ（log-odds）$\eta = \log \frac{\theta}{1-\theta}$ で表すと平坦ではなくなり、η に関して MAP 推定値をとると $10/15$（$9/13$ ではなく）が得られる（演習問題 5）。一方、尤度の値はパラメータ空間の各点に直接結びつけられたデータの確率であるため、ヤコビ行列式の因子は生じない。その結果、MLE は**同変**（equivariant）である。すなわち、任意の 1 対 1 変換 g に対して、$\hat{\theta}$ が θ の MLE であれば、$g(\hat{\theta})$ は $g(\theta)$ の MLE となる (Wasserman 2013)。たとえばコインの対数オッズの MLE は、単に $\log \frac{9/13}{4/13} = \log \frac{9}{4}$ である。最尤法が「平坦な事前分布をもつ MAP」としてではなく事前分布なしで定義されるのはこのためである。もし前者として定義すれば、パラメータを書き換えるたびに答えが動いてしまう。なお、同変性にも固有のトレードオフがある。同じ演習問題で示すとおり、あらゆる再パラメータ化のもとで同変かつ不偏であり続ける推定量は存在しない。

さらに根深い限界は、MAP も MLE と同様に、得られた知識を単一のパラメータベクトルへと潰してしまう点にある。完全な事後分布 $p(\theta \mid X)$ は、データを見た後にパラメータがどれほど不確実なまま残っているかを捉えている。その不確実性にわたって平均をとる予測 $\int p(x \mid \theta)\, p(\theta \mid X)\, d\theta$ はパラメータの不確実性を考慮に入れるのに対し、最頻値を代入するだけの予測はそれを捨て去ってしまう。このような積分が閉形式をもつことはまれであるため、それらをいかに近似するかが 36.8 節の主題となる。

しかしその前に、より直接的な障害が立ちはだかる。本節の議論はすべて、各観測の確率をパラメータの関数として直接書き下せることを前提としていた。各データ点が複数の隠れた生成源のいずれかから生じるものの、どれから生じたかは観測できない場合、その確率は隠れた可能性にわたる和となり、和の対数はもはや単純な項へと綺麗に分解できなくなる。36.7 節では、その代わりに何を最適化すべきかを示す。

36.6.5 まとめ

最尤法は観測されたデータが最も起こりやすくなるパラメータを選ぶ原理であり、実際には例ごとの驚きの和である負の対数尤度を最小化することを意味する。標準的な訓練損失の大半はこのようにノイズモデルの選択から導かれる。すなわちガウスノイズからは二乗誤差、ラプラスノイズからは絶対誤差、離散的な結果からは交差エントロピーが生じ、離散の場合の平均損失は経験頻度のエントロピーを決して下回ることができない。同じ損失の曲率は推定の精度を定量化する。正則条件のもとで MLE の誤差は近似的に分散 $1/(nI)$ のガウス分布に従い、ここでフィッシャー情報量 I はスコアの分散と NLL の期待曲率の両方に等しく、不偏推定量に対するクラメール・ラオの下界に一致する。これらの古典的な保証はモデルが誤特定されている場合や識別不能である場合には破綻し、深層ニューラルネットワークはその両方に該当する。最後に、事前分布を導入すると MAP 推定は NLL と正則化項の和へと変わり、ガウス事前分布からは重み減衰が、ベータ事前分布からは擬似カウントが導かれるが、それでも MAP 推定値は事後分布を単一の点へと縮約してしまうことに変わりはない。

36.6.6 演習問題

1. **交差エントロピーの下限。**離散的な結果に対しては、36.6.2.1 節より平均 NLL は $H(\hat{p}_{\text{data}}) + D_{\text{KL}}(\hat{p}_{\text{data}} \,\|\, p_\theta)$ に等しいため、経験エントロピーを下回ることはできない。
   1. 表 9 回・裏 4 回のコインの系列に対してこの下限を nat 単位で求め、さらに公平なコインのモデル $\theta = 1/2$ における KL ギャップを評価せよ。
   2. 分類器が $p(y \mid \mathbf{x})$ をモデル化し、異なる各入力に対して個別にラベル上の任意の分布を表現できるとする。訓練セット上で達成可能な最小の平均 NLL が経験条件付きエントロピー $\sum_{\mathbf{x}} \hat{p}_{\text{data}}(\mathbf{x})\, H\big(\hat{p}_{\text{data}}(\cdot \mid \mathbf{x})\big)$ であることを示せ。また、100 個の異なる入力がそれぞれ 10 回ずつ現れ、そのうちちょうど 1 回だけ誤ったラベルが付いている二値分類タスクについて、この値を評価せよ。
   3. 次に、典型的な画像データセットのように各訓練入力が 1 度しか現れないとする。(b) の下限はいくらになるか。このとき訓練損失がゼロに近いことからラベルのノイズの多寡について何も言えないのはなぜか。また、実際に測定可能な量のうち、母集団の条件付きエントロピー $H(Y \mid X)$ を下限としてもつものはどれか。
2. **学習されるノイズ分散。**ガウス分布の NLL Equation 36.82 では分散 σ^2 を固定していたが、ここでは分散も併せて学習する。
   1. 1 つの共通な σ^2 を θ と並ぶパラメータとする。NLL を σ^2 について（θ を固定して）最小化し、残る目的関数が $\frac{n}{2}\log\big(\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2\big)$ に定数を加えたものになることを示せ。θ に関する最小点は依然として最小二乗法による当てはめと一致するか。
   2. ネットワークが各入力に対して平均 $\hat{y}(\mathbf{x})$ と分散 $\hat{\sigma}^2(\mathbf{x}) > 0$ を出力するとする。加法定数を除くと、訓練セット上での NLL は次式となる。

$$\sum_{i=1}^{n} \left[\frac{(y_i - \hat{y}(\mathbf{x}_i))^2}{2\hat{\sigma}^2(\mathbf{x}_i)} + \frac{1}{2}\log \hat{\sigma}^2(\mathbf{x}_i) \right].$$

   ネットワークが 1 つの訓練点に厳密に一致しつつその分散を限りなく小さくできるとき、この目的関数が下に有界でなくなることを示せ。36.7 節における混合ガウスモデルの尤度のどの性質が同じ原因から生じるか。
   3. 例 i の項の $\hat{y}(\mathbf{x}_i)$ に関する勾配を計算せよ。予測分散が残差の二乗に追従し、$\hat{\sigma}^2(\mathbf{x}_i) \approx r_i^2$（ただし $r_i = y_i - \hat{y}(\mathbf{x}_i)$）となるとする。このとき勾配の大きさは $|r_i|$ にどのように依存し、二乗誤差の場合とどう異なるか。また、この重み付けは現在当てはめが悪い領域における平均の学習にどのような影響を与えるか (Seitzer et al. 2022)。
3. **尤度ではない損失。**$\phi(y, o)$ をラベル y とモデル出力 o の損失とし、$Z(o) = \sum_y e^{-\phi(y,o)}$ とおく。もし $Z(o)$ が o に依存しなければ、$e^{-\phi(y,o)}/Z$ は y の（o を与えたときの）条件付

き分布となり、その NLL は ϕ に定数を加えたものになるため、ϕ の最小化は最尤法となる。

1. ラベル $y \in \{-1, +1\}$ に対し、ロジスティック損失 $\phi(y, o) = \log(1 + e^{-yo})$ が $Z(o) = 1$ を満たすことを示し、このモデルが $y = +1$ に割り当てる確率を特定せよ。
2. サポートベクターマシンのヒンジ損失 $\phi(y, o) = \max(0, 1 - yo)$ について、$|o| \le 1$ の場合と $|o| > 1$ の場合に分けて $Z(o)$ を計算せよ。そこから、ヒンジ損失が定数差を除いても y の（o を与えたときの）いかなる条件付き分布の NLL にもならないことを結論づけよ。ヒンジ損失にどの項を加えれば $e^{-\phi(y,o)}/Z(o)$ の NLL が得られるか。また、その項は $|o|$ が大きくなるにつれてどのように振る舞うか。
3. 実数ターゲット y と予測 $\hat{y}$ に対し、フーバー損失のように残差のみに依存する任意の損失 $\phi(y, \hat{y}) = \rho(y - \hat{y})$（ただし $\int e^{-\rho(r)}\, dr < \infty$）をとる。$Z(\hat{y}) = \int e^{-\rho(y-\hat{y})}\, dy$ が $\hat{y}$ に依存しないことを示せ。このような回帰損失では本問の条件が自動的に成り立つのに対し、分類損失では強い制約となるのはなぜか。

4. [code] **境界：分離可能なデータとまれな結果**。最適なパラメータが定義域の端にあるとき、最尤法は異常な振る舞いを示す。
   1. ラベル付きの 2 点（$x = -1$ で $y = 0$、および $x = +1$ で $y = 1$）にバイアスなしのロジスティック回帰 $P(y = 1 \mid x) = \text{sigmoid}(wx)$ を当てはめる。NLL が $2\log(1 + e^{-w})$ であり、すべての w に対して単調に減少して 0 に近づくものの、いかなる有限の w もその下限を達成しないことを示せ。本文中のどのコイン系列がこれと同じ問題を抱えているか。また $w \to \infty$ は $x = 0.1$ にある新しい点に対して何を予測するか。
   2. ガウス事前分布 $w \sim \mathcal{N}(0, \tau^2)$ を加える。MAP 推定値が $2\,\text{sigmoid}(-w) = w/\tau^2$ の解となることを示し、$\tau = 1$ の場合に数値的に解け。さらに一般化し、任意の線形分離可能なデータセットに対してロジスティック回帰の NLL は下限 0 をもち最小点をもたないが、ガウス事前分布を導入すれば有限の MAP 推定値が回復されることを示せ。
   3. コインにおいて、$\hat{\theta} = n_H/n$ の歪度は $(1-2\theta)/\sqrt{n\theta(1-\theta)}$ である。これを $\theta = 0.7$ と $\theta = 0.95$ について $n = 400$ で評価し、どの n で $\theta = 0.95$ の歪度が $\theta = 0.7$（$n = 400$ のとき）と同じ水準に達するかを求めよ。また θ が 1 に近いとき、裏の期待出現回数 $n(1-\theta)$ を用いて歪度を表せ。ガウス極限が正確かどうかを決めるのが n 単体ではなくその期待回数であるのはなぜか。
   4. $n = 400$ 回のコイン投げからなる 20,000 個のデータセットをシミュレーションし（`rng.binomial(n, theta, 20000) / n`）、$\sqrt{n}(\hat{\theta} - \theta)$ のヒストグラムを $\theta = 0.7$ と $\theta = 0.95$ について $\mathcal{N}(0, \theta(1-\theta))$ と重ねて描け。$\theta = 0.95$ のヒストグラムが理論曲線からどのように乖離するかを記述せよ。

5. **対数オッズにおける MAP**。コインをその対数オッズ $\eta = \log \frac{\theta}{1-\theta}$ によってパラメータ化する。このとき $\theta = \text{sigmoid}(\eta)$ かつ $\frac{d\theta}{d\eta} = \theta(1-\theta)$ である。一様事前分布 $\text{Beta}(1, 1)$（θ 上）と、表 9 回・裏 4 回の系列はそのまま用いる。
   1. 密度の変数変換の公式 $p_\eta(\eta) = p_\theta(\theta)\,|d\theta/d\eta|$（36.2 節）を用いて、この信念が導く η の事前密度を求めよ。それがロジスティック密度 $e^{-\eta}/(1+e^{-\eta})^2$、すなわち $\log \frac{U}{1-U}$（U は一様確率変数）の分布であることを示し、計算を行わずに逆変換サンプリングを用いてその結果を説明せよ。
   2. η の MAP 推定値を求め、それを θ に写し戻せ。得られた値を、θ パラメータ化での MAP 推定値、36.6.4.3 節の MLE $\log \frac{9}{4}$、およびラプラスの継起則と比較せよ。
   3. 一般の事前分布 $\text{Beta}(a, b)$ と出現回数 n_H, n_T に対して (b) を繰り返せ。対数オッズでの MAP 推定値は、θ 上の事後分布のどの要約量を再現するか。
   4. MLE は同変であるため、θ^2 の MLE は $\hat{\theta}^2$ である。その統計的バイアスが $\mathbb{E}[\hat{\theta}^2] - \theta^2 = \text{Var}(\hat{\theta}) = \theta(1-\theta)/n$ であることを示し、$\theta = 9/13$、$n = 13$ で評価せよ。そこから、すべての滑らかな 1 対 1 の再パラメータ化のもとで同変であり、かつそのすべてに対して不偏であるような推定量は存在しないことを結論づけよ。

6. **ReLU ネットワークの対称性。**●スカラー入力 x とパラメータ $\theta = (a_1, w_1, \dots, a_h, w_h)$ をもつネットワーク $f_\theta(x) = \sum_{j=1}^{h} a_j \,\mathrm{ReLU}(w_j x)$、および固定された σ^2 をもつ回帰モデル $y \sim \mathcal{N}(f_\theta(x), \sigma^2)$ を考える。識別可能性は 36.6.3 節で定義されている。
   1. (a_j, w_j) を $(c\,a_j,\, w_j/c)$（任意の $c > 0$）に置き換えても f_θ が変化しないことを示せ。どの 2 つのユニットも一致しないとき、隠れユニットの置換によって同じ関数を与える異なるパラメータベクトルは何通り生じるか。
   2. Equation 36.83 から、このモデルのフィッシャー情報量が $I(\theta) = \mathbb{E}_x\big[\nabla_\theta f_\theta(x)\, \nabla_\theta f_\theta(x)^\top\big]/\sigma^2$ となることを示せ。(a) のリスケーリングを c に関して $c = 1$ で微分し、非ゼロベクトル $\mathbf{v}$ で $I(\theta)\mathbf{v} = \mathbf{0}$ を満たすものを求めよ。Equation 36.85 の背後にあるどの仮説（条件）が破綻するか。
   3. 平均二乗誤差 $\frac{1}{n}\sum_i (y_i - f_\theta(x_i))^2$ の a_j に関するヘッセ行列の対角成分は $\frac{2}{n}\sum_i \mathrm{ReLU}(w_j x_i)^2$ である。この成分は (a) のリスケーリングのもとでどのように変化するか。w_j に対する対角成分を計算し、ReLU ユニットでは（折れ曲がり点 $w_j x_i = 0$ を除いて）残差項が含まれない理由を説明せよ。また、その成分はリスケーリングのもとでどう変化するか。このことは、ヘッセ行列の固有値に基づく最小点の平坦性の指標に対して何を意味するか (Dinh, Pascanu, et al. 2017)。
7. **動く台。**●$x_1, \dots, x_n$ が未知のパラメータ $\theta > 0$ をもつ $[0, \theta]$ 上の一様分布に独立に従うとする。このモデルは 36.6.3 節の正則条件の 1 つに違反している。
   1. 尤度を θ の関数として書き下し、MLE が $\hat\theta = \max_i x_i$ であることを示せ。導関数をゼロとおく方法ではこの解が見つからないのはなぜか。
   2. $P(n(\theta - \hat\theta) > t) = \big(1 - \frac{t}{n\theta}\big)^n$ $(0 \le t \le n\theta)$ を示し、$n \to \infty$ としたときの極限を求めよ。誤差は n に関してどの速さで縮小するか。また極限分布はガウス分布になるか。
   3. 台の上でスコアは $s(x;\theta) = -1/\theta$ である。$\mathbb{E}_\theta[s]$ を計算せよ。スコアの期待値がゼロになる導出のどの段階が破綻しているか（演習問題 1（34.7 節）と比較せよ）。
   4. 不偏推定量 $\frac{n+1}{n}\hat\theta$ の分散は $\theta^2/(n(n+2))$ である。この値を、Equation 36.86 が下界として与えるはずの値 $1/(n\,\mathbb{E}_\theta[s^2])$ と比較せよ。ここでは下界の証明のどの段階が破綻しているか。
8. **[extended] モデルの誤特定とサンドイッチ共分散。**●●本問では、フィッシャー情報量の 2 つの表現が一致しない場合について漸近正規性の発見的な導出を行う。モデル族のどの分布もデータ生成法則と一致しないとき、MLE は母集団の KL ダイバージェンスを最小化する擬真のパラメータ $\theta^\dagger$ へ収束し（36.6.3.5 節）、そのばらつきはもはや I^{-1}/n ではなくなる (White 1982)。データ生成法則のもとでの期待値を用いて、$\mathbf{A} = -\mathbb{E}\big[\nabla_\theta^2 \log p(x \mid \theta^\dagger)\big]$ および $\mathbf{B} = \mathbb{E}\big[s(x;\theta^\dagger)\, s(x;\theta^\dagger)^\top\big]$ とおく。
   1. 一階条件 $\sum_i s(x_i;\hat\theta) = \mathbf{0}$ を $\theta^\dagger$ の周りで一次まで展開し、$\frac{1}{\sqrt{n}}\sum_i s(x_i;\theta^\dagger)$ に中心極限定理を適用することで、次式が成り立つことを論じよ。

$$\sqrt{n}\,(\hat\theta - \theta^\dagger) \xrightarrow{d} \mathcal{N}\big(\mathbf{0},\ \mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}\big). \tag{36.90}$$

   モデルが誤っているにもかかわらず $\mathbb{E}\big[s(x;\theta^\dagger)\big] = \mathbf{0}$ が成り立つのはなぜか。また、モデルが正しいときにはこの共分散が Equation 36.85 の共分散に帰着することを示せ。
   2. 平均 m、形状パラメータ α（分散は $m + m^2/\alpha$）の負の二項分布からのカウントデータに、レート λ のポアソンモデルを当てはめる。$\hat\lambda$、$\lambda^\dagger$、$\mathbf{A}$、$\mathbf{B}$ を求めよ。逆フィッシャー情報量は $\hat\lambda$ の分散を何分の 1 に過小評価するか。
   3. 1 つの特徴量をもつポアソン回帰 $\log \lambda(x) = w_0 + w_1 x$ をニュートン法によって当てはめよ。このモデルに対する更新式は $\mathbf{w} \leftarrow \mathbf{w} + (\mathbf{Z}^\top \mathrm{diag}(\lambda_i)\mathbf{Z})^{-1}\mathbf{Z}^\top(\mathbf{y} - \lambda)$ であ

り、ここで各行 $(1, x_i)$ をもつ計画行列を $\mathbf{Z}$ とし、$\lambda_i = \lambda(x_i)$ とする。ここでは 1 つの観測がペア (x_i, y_i) であり、$\log p(y_i \mid x_i, \mathbf{w})$ が $\log p(x \mid \theta)$ の役割を $\mathbf{A}$ と $\mathbf{B}$ において担う。当てはめた $\hat{\mathbf{w}}$ における標本平均によって両者を推定せよ。次に、2,000 個のデータセット（各 $n = 500$ 組のペア）を、$x \sim \mathcal{N}(0, 1)$、$\mathbf{w} = (1, 0.5)$、および $\alpha = 2$ の負の二項分布に従うカウントから生成せよ。すなわち、形状 α、平均 $\lambda(x)$ のガンマ分布からレートを引き、そのレートでポアソン分布からカウントを引く。そして $\hat{w}_1$ の 3 種類の標準誤差を比較せよ。逆フィッシャー情報量から求めたもの、サンドイッチ公式から求めたもの、およびデータセット間の $\hat{w}_1$ のばらつきから求めたものの 3 つである。最初の 2 つから構成した公称 95% 信頼区間のうち、それぞれどの割合が $w_1 = 0.5$ を被覆するか。さらに $\alpha = 10^6$ として同じ実験を繰り返せ。

4. 演習問題 6c（36.4 節）は、カウントが過分散であってもポアソン回帰の当てはめが依然として $\mathbf{w}$ を復元することを示す。モデルのどの部分が誤っていれば $\mathbf{w}^{\dagger}$ は $\mathbf{w}$ と異なることになるか。また、その場合 (c) のサンドイッチ信頼区間はどのパラメータを被覆するか。

36.7 潜在変数、EM、ELBO

36.6 節で扱った尤度はどれも、観測ごとに 1 つの因子を持つきれいな積の形をしていたため、対数をとれば項ごとに微分できる和へと変換できた。この扱いやすさは、モデルが生成するすべての量を観測できるという暗黙の仮定の上に成り立っていたのである。

実際には、データの重要な一部が欠けていることが少なくない。2 つの異なる母集団から得られた 500 個の測定値（一方から 300 個、もう一方から 200 個）を集めたものの、どの測定値がどちらの母集団から生じたのか誰も記録していなかったとしよう。もしそのラベルが残っていれば、モデルの当てはめは母集団ごとに混合比・平均・分散を計算するだけの 1 行で済むはずである。しかしラベルがなければ、各測定値の確率は、それが生じ得た 2 つの母集団にわたる和となる。厄介なことに、和の対数は分解できないため、すべてのパラメータがあらゆる項の中で絡み合い、勾配を 0 と置いても閉形式解（closed-form solution）を持たない方程式しか残らない。顧客のクラスタ、文書のトピック、あるいは生成器が画像へと変換するランダムなコードなど、隠れた構造を通じて観測を説明するモデルでは常に同じ障壁が現れる。このような観測されない量は**潜在変数**（latent variable）と呼ばれる。

本節の中心的なアイデアは単純でありながら驚くほど有効である。すなわち、見えないものを確定的なハードラベルではなく確率を使って推測し、そのソフトな推測が実際のデータであるかのようにモデルを当てはめるのである。このアイデアがうまく機能する理由は、推測が正しい必要すらない点にある。**いかなる**確率的な推測に対しても、そこから得られる扱いやすい目的関数は真の対数尤度（log-likelihood）の下側に位置し、その推測が隠れたラベルに関する事後分布（posterior distribution）に一致するときに限り、対数尤度にぴったり接する。この下界こそ、覚えておくべき重要な結果である。期待値最大化法（expectation–maximization）はこの下界を直接登り、厳密な事後分布が計算できない場合には、変分推論（variational inference）や変分オートエンコーダ（variational autoencoder）がこれを近似的に登る。

以下では、この 1 つのアイデアを最初から最後まで展開していく。まず混合モデル（mixture model）を定式化し、次に下界を導出してそのギャップを測り、最後に得られたアルゴリズムを先ほどの 500 点に対して実行する。その際、両方の母集団を復元できる初期値から始める場合と、対称な初期値から始めて途中で立ち往生する場合の両方を確かめる。

36.7.1 混合モデル

このラベル欠損のパズルを数式で表してみよう。潜在変数 z を持つ確率モデル（probabilistic model）は、隠れた量と観測される量の同時分布（joint distribution）を次のように定める。

$$p(x, z \mid \theta) = p(z \mid \theta)\, p(x \mid z, \theta),$$

このうち観測されるのは x のみである。実際に観測されたデータの尤度を計算するには、隠れ変数がとり得るすべての値について和をとる（z が連続ならば積分して消去する）必要があり、これにより**周辺**尤度（marginal likelihood）が得られる。

$$p(x \mid \theta) = \sum_{z} p(z \mid \theta)\, p(x \mid z, \theta). \tag{36.91}$$

言葉で言えば、モデルが x に割り当てる確率は、モデルが x を生成し得たあらゆる経路を、隠れ変数の値ごとに 1 項ずつ足し合わせたものになっている。

代表的な例が**混合ガウスモデル**（Gaussian mixture model、GMM）である。このモデルでは、潜在変数 $z \in \{1, \dots, K\}$ が成分 k を混合確率 π_k で選び、続いて観測がその成分のガウス分布 $x \mid z{=}k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ から抽出される。n 個の観測をまとめると、負の対数尤度（negative log-likelihood、NLL）は次のようになる。

$$\ell(\theta) = -\sum_{i=1}^{n} \log\Big(\sum_{k=1}^{K} \pi_k\, \mathcal{N}(x_i;\, \mu_k, \sigma_k^2) \Big), \qquad \theta = (\pi_k, \mu_k, \sigma_k)_{k=1}^{K}.$$

厄介さのすべては、和が対数の**内側**に入り込んでいることに起因する。対数は混合分布のところで止まってしまい、個々のガウス密度まで届かない。もし届けば、指数関数と打ち消し合って単純な二次式が残るはずである。これを**完全データ**（complete-data）の問題と比べてみよう。仮にオラクルがすべてのラベル z_i を教えてくれたとすれば、尤度は成分ごとにきれいに分離し、最尤法（maximum likelihood）は既知の推定量へと帰着する。各混合重み π_k は単に成分 k に属するデータ点の割合となり、コインの偏りの推定量 $\hat{\theta} = n_H/n$（36.6 節）とまったく同じ形になる。一方、μ_k と σ_k^2 はそれらのデータ点の通常の標本平均と分散になる。ラベルが欠けているときに限って問題が難しくなるのだから、自然な戦略が浮かび上がる。欠けているラベルを確率として推定し、そのうえで簡単な完全データの問題を解けばよいのである。

1 つのデータ点に対するそのような確率的推測は、どのような形になるだろうか。$x = 0.5$ における 1 つの観測を、等しい重みを持つ 2 つの成分 $\mathcal{N}(-1, 1)$ と $\mathcal{N}(1, 1)$ のもとで考えよう。$x = 0.5$ における両成分の密度はそれぞれ $e^{-(1.5)^2/2} = e^{-1.125}$ と $e^{-(0.5)^2/2} = e^{-0.125}$ に比例するため、ベイズの定理により左側の成分には次の事後確率（posterior probability）が割り当てられる。

$$\frac{e^{-1.125}}{e^{-1.125} + e^{-0.125}} = \frac{1}{1 + e} = 0.27.$$

ベイズの定理はハードな選択を強いるのではなく、このデータ点を 2 つの成分の間で 27% 対 73% に分割する。観測をずっと右側の、たとえば $x = 3$ に動かせば、事後分布は 99.8% の確信度で右側の成分に属すると判定し、中点の $x = 0$ に置けば、分割はちょうど半々の 50% 対 50% になる。ソフトな推測は、データが明確な領域では確信を持ち、成分が重なり合う領域では適切に判断を留保する（Figure 36.34）。

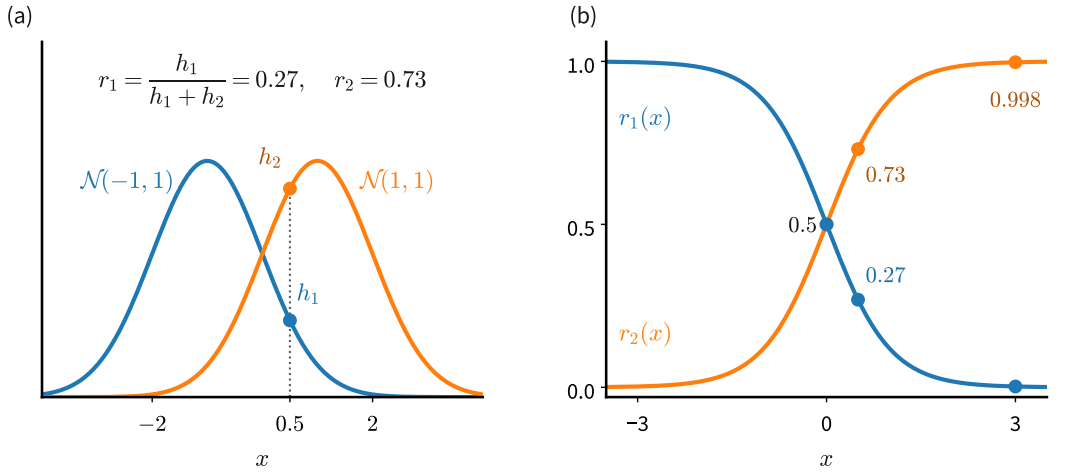


Figure 36.34: 等しい重みを持つ 2 つの成分 $\mathcal{N}(-1, 1)$ と $\mathcal{N}(1, 1)$ における、密度の高さの比としてのソフト割り当て。(a) $x = 0.5$ において各密度は高さ h_1 と h_2 を持ち、左側の成分は割合 $r_1 = h_1/(h_1 + h_2) = 0.27$ を受け取る。(b) 2 つの事後確率 $r_1(x)$ と $r_2(x)$。$x = 0$ で均等分割、0.27 対 0.73（$x = 0.5$）、そして右側の成分に対する 0.998（$x = 3$）となる。

36.7.2 エビデンス下界

この直感的な戦略を数学的保証のあるアルゴリズムへと昇華させるため、対数が和の内側へと戻るような対数尤度の下界を構成しよう。$q(z)$ を潜在変数上の**任意の**確率分布とし、ソフトな推測を表すものとする。周辺尤度の内側で $q(z)$ を掛けて割り、イェンセンの不等式（Jensen's inequality）を適用する。対数関数は凹関数であるため、$\log \mathbb{E}[\cdot] \geq \mathbb{E}[\log(\cdot)]$（35.3.3.1 節）が成り立つ。これにより次式を得る。

$$\log p(x \mid \theta) = \log \sum_z q(z) \frac{p(x, z \mid \theta)}{q(z)} \;\geq\; \sum_z q(z) \log \frac{p(x, z \mid \theta)}{q(z)} \;=:\; \mathcal{L}(q, \theta). \quad (36.92)$$

ここで和は、すべての z（ただし $q(z) > 0$）にわたってとる。仮に q が、同時分布が正となるある z に確率 0 を割り当てていたとしても、最初の等号が単に $\geq$ に変わるだけなので、この下界は依然として成り立つ。なお、周辺尤度 $p(x \mid \theta)$ は潜在変数 z に対して、エビデンス $P(X)$（36.6.4 節におけるベイズの定理の分母）がモデルパラメータに対して果たすのとまったく同じ役割を果たしている。すなわち、観測されない量をすべて和で消去した後の、観測データの確率である。そのため、この量も**エビデンス**と呼ばれ、右辺の $\mathcal{L}(q, \theta)$ は**エビデンス下界**（evidence lower bound、**ELBO**）と呼ばれる。比の対数を分解すると、ELBO は見通しのよい 2 つの項に分かれる。

$$\mathcal{L}(q, \theta) = \mathbb{E}_q\big[\log p(x, z \mid \theta)\big] + H(q).$$

第 1 項は期待**完全データ**対数尤度であり、扱いやすい目的関数をソフトな推測 q で平均したものである。第 2 項はその推測のエントロピーであり、θ には一切依存しない。さらに都合のよいことに、ELBO と真の対数尤度の間の厳密なギャップを測ることもできる。その正体は KL ダイバージェンスであり、EM と変分推論はどちらもこのただ 1 つの恒等式を土台としている。

命題（ELBO のギャップは KL ダイバージェンスである）。任意の θ と、いかなる分布 q（ただし事後分布 $p(z \mid x, \theta)$ の台の外側に質量を持たないもの）に対しても、次式が成り立つ。

$$\log p(x \mid \theta) = \mathcal{L}(q, \theta) + D_{\mathrm{KL}}\big(q(z) \,\|\, p(z \mid x, \theta)\big). \tag{36.93}$$

したがって $\mathcal{L}(q,\theta) \le \log p(x \mid \theta)$ であり、等号は q が事後分布 $p(z \mid x, \theta)$ に一致するとき、かつそのときに限り成り立つ。

証明。 ベイズの定理 $p(z \mid x, \theta) = p(x, z \mid \theta)/p(x \mid \theta)$ を KL ダイバージェンスの定義に直接代入すると、次のようになる。

$$D_{\mathrm{KL}}(q \,\|\, p(z \mid x, \theta)) = \mathbb{E}_{q}\left[\log \frac{q(z)\, p(x \mid \theta)}{p(x, z \mid \theta)}\right] = \log p(x \mid \theta) - \mathcal{L}(q, \theta),$$

ここで $\log p(x \mid \theta)$ は z に依存しないため、そのまま期待値の外に出せる。式を整理すれば Equation 36.93 が得られる。ギブスの不等式（36.4.2 節）より、KL ダイバージェンスは常に非負であり、2 つの引数が一致するとき、かつそのときに限り 0 になる。■

前項の具体例 $x = 0.5$ でこの恒等式を確かめてみよう。この例では、真の対数エビデンスは $\log p(x) = -1.42$ である。判断を留保した推測 $q = (\tfrac{1}{2}, \tfrac{1}{2})$ を用いると、ELBO の値は -1.54 となり、不足分の 0.12 nat は $(\tfrac{1}{2}, \tfrac{1}{2})$ から事後分布 $(0.27, 0.73)$ への厳密な KL ダイバージェンスに一致する。言い換えれば、下界が対数エビデンスに届かない分は、推測と真の事後分布とのずれそのものであり、それ以外の何ものでもない。Figure 36.35 に、θ を変化させたときのこの下界の振る舞いを示す。

36.7.3 期待値最大化法

ELBO の恒等式により、難しかった最適化問題は $\mathcal{L}(q, \theta)$ 上の**座標上昇法**（coordinate ascent）へと変換される。すなわち、一方の引数を固定したままもう一方に関して最大化し、役割を入れ替えて繰り返せばよい。独立な観測からなるデータセット $X = \{x_1, \dots, x_n\}$ が与えられたとき、この下界は各観測に個別に適用でき、データ点ごとに 1 つの推測 q_i を置いてデータセット全体にわたる和をとる形になる。**期待値最大化**（EM）アルゴリズムは、これら 2 つの座標を交互に更新する (Dempster et al. 1977)。

- **E ステップ：** $\theta^{(t)}$ を固定し、推測 q_i に関して最大化する。Equation 36.93 の等号成立条件より、各データ点に対する最適な選択はその厳密な事後分布 $q_i^{(t)}(z) = p(z \mid x_i, \theta^{(t)})$ である。これによりすべての KL ギャップが 0 になり、現在のパラメータにおいて下界は**タイト**になる。すなわち $\sum_i \mathcal{L}(q_i^{(t)}, \theta^{(t)}) = \log P(X \mid \theta^{(t)})$ が成り立つ。
- **M ステップ：** 推測 $q_i^{(t)}$ を固定し、パラメータ θ に関して最大化する。

$$\theta^{(t+1)} = \operatorname*{argmax}_{\theta} \sum_{i} \mathbb{E}_{q_i^{(t)}}\big[\log p(x_i, z \mid \theta)\big],$$

  ここでエントロピー項 $H(q_i^{(t)})$ は θ に依存しないため消える。こうして、欠けているオラクルのラベルをソフトな確率で代用した、扱いやすい完全データの目的関数へと戻ってくる。

なぜこの交互更新の手順を信頼できるのだろうか。それは Figure 36.35 の幾何学的構造により、1 ステップごとに真の対数尤度が必ず改善する（少なくとも悪化しない）ことが保証されているからである。

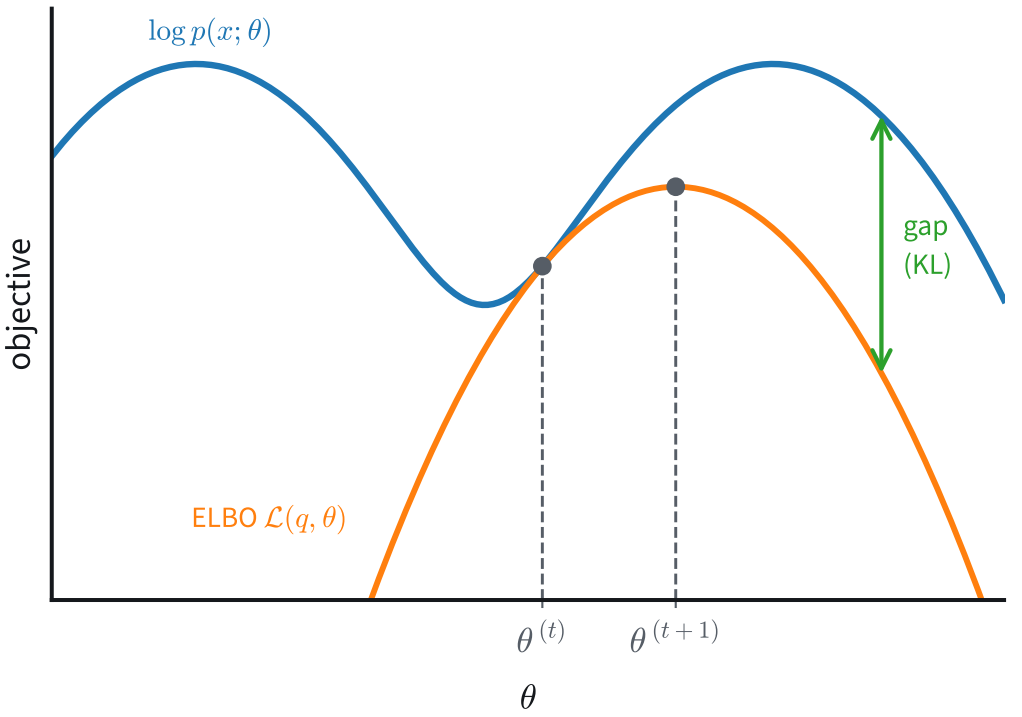


Figure 36.35: エビデンスとその下界。固定された推測 q に対し、ELBO $\mathcal{L}(q,\theta)$ はエビデンス $\log p(x \mid \theta)$ の下側にすべての θ で位置し、両曲線の縦方向のギャップは q から事後分布 $p(z \mid x,\theta)$ への KL ダイバージェンスとなる。EM では、E ステップで q を現在の反復点 $\theta^{(t)}$ における事後分布に設定し、ギャップを閉じてその点で下界をエビデンスに接しさせる。続く M ステップでは、下界をその頂点 $\theta^{(t+1)}$ まで登る。下界は $\theta^{(t)}$ でエビデンスに接し、かつ決してその上に出ないため、$\theta^{(t+1)}$ におけるエビデンスは少なくとも $\theta^{(t)}$ と同じ高さ以上になることが保証される。

命題（EM は尤度を決して減少させない）。EM のすべての反復は $\log P(X \mid \theta^{(t+1)}) \geq \log P(X \mid \theta^{(t)})$ **を満たす。**

証明。 1 行で表せる 3 つの事実をつなぎ合わせる。

$$\log P(X \mid \theta^{(t+1)}) \;\geq\; \sum_i \mathcal{L}(q_i^{(t)}, \theta^{(t+1)}) \;\geq\; \sum_i \mathcal{L}(q_i^{(t)}, \theta^{(t)}) \;=\; \log P(X \mid \theta^{(t)}).$$

最初の不等号は新しいパラメータで評価した ELBO の下界 Equation 36.92 であり、2 番目の不等号は M ステップが θ に関して下界を最大化することから成り立ち、最後の等号は E ステップによって確保されたタイトさそのものである。■

混合ガウスモデルでは、両方のステップが閉形式で求まる。E ステップでは**負担率**（responsibility）、すなわちデータ点 i が成分 k から生成された事後確率を、先ほど $x = 0.5$ に対して手計算したのと同様に計算する。

$$r_{ik} = \frac{\pi_k \, \mathcal{N}(x_i;\, \mu_k, \sigma_k^2)}{\sum_j \pi_j \, \mathcal{N}(x_i;\, \mu_j, \sigma_j^2)}.$$

続く M ステップでは、負担率で重み付けしたカウントと平均化によって各成分を当てはめ直す。$n_k = \sum_i r_{ik}$ を成分 k に割り当てられた実効的なデータ点数とおけば、次式が得られる。

$$\pi_k = \frac{n_k}{n}, \qquad \mu_k = \frac{1}{n_k} \sum_i r_{ik} \, x_i, \qquad \sigma_k^2 = \frac{1}{n_k} \sum_i r_{ik} \, (x_i - \mu_k)^2. \tag{36.94}$$

これらの式がいかに見慣れた形をしているかに注目してほしい。通常の完全データに対する推定量（混合比、標本平均、標本分散）を、単にソフト化しただけなのである。各データ点は 1 つの成分にハードな 1 票を投じる代わりに、負担率に応じて複数の成分へと票を分割する。実際、すべての成分が単一の固定された分散 σ^2 を共有するよう制約して $\sigma^2 \to 0$ の極限をとれば、負担率は最も近い平均への $0/1$ 割り当てへと硬化し、EM はそのまま K-means アルゴリズムに帰着する。

アルゴリズム全体は十数行のコードで書ける。300 点を $\mathcal{N}(-2, 0.8^2)$ から、200 点を $\mathcal{N}(1.5, 0.6^2)$ から生成し、1 次元の混合ガウスモデルに対する短い関数 `em`（K 個のガウス成分）を書いて、最終的に当てはめられたパラメータとともに各ステップの対数尤度を記録してみよう。

```
rng = np.random.default_rng()
x = np.concatenate([rng.normal(-2.0, 0.8, 300), rng.normal(1.5, 0.6, 200)])

def gauss(x, mu, sigma):
    return np.exp(-0.5 * ((x - mu) / sigma)**2) / (sigma * np.sqrt(2 * np.pi))

def em(x, pi, mu, sigma, steps):
    """EM for a 1-D Gaussian mixture: log-likelihood trace and final fit."""
    pi, mu, sigma = (np.array(v, dtype=float) for v in (pi, mu, sigma))
    lls = []
    for _ in range(steps):
        joint = pi * gauss(x[:, None], mu, sigma)  # (n, K)
        lls.append(np.log(joint.sum(axis=1)).sum())
```

```
        r = joint / joint.sum(axis=1, keepdims=True)  # E-step
        n_k = r.sum(axis=0)  # M-step
        pi, mu = n_k / len(x), (r * x[:, None]).sum(axis=0) / n_k
        sigma = np.sqrt((r * (x[:, None] - mu)**2).sum(axis=0) / n_k)
    return np.array(lls), pi, mu, sigma
```

先の命題は対数尤度が決して減少しないことを保証するが、EM がどれほど速く登るかについては何も語らない。そこで、かなり悪い初期推測（中心が ± 0.5 にあり大きく重なり合う単位分散の 2 成分）から出発して 30 回の反復を実行し、最初の 6 ステップにおける対数尤度の値と最終ステップの値を比べてみよう。

```
lls, pi, mu, sigma = em(x, [0.5, 0.5], [-0.5, 0.5], [1, 1], 30)
print('steps 0-5:', lls[:6].round(1), ' step 29:', lls[-1].round(1))
print(f'pi = {pi.round(2)}, mu = {mu.round(2)}, sigma = {sigma.round(2)}')
```

```
steps 0-5: [-1298.1  -989.9  -942.   -871.   -848.6  -847. ]  step 29: -846.8
pi = [0.59 0.41], mu = [-2.04  1.54], sigma = [0.74 0.6 ]
```

対数尤度は約 400 nat 上昇し、およそ $-1{,}280$ からおよそ -860 へと達する（どちらの数値も無作為標本によって数十 nat ほど変動する）。驚くべきことに、改善のほぼすべてが直後に起きている。わずか 6 ステップ後には対数尤度は最終値から約 1 nat 以内に入っており、復元された重み・平均・スケールもすべて誤差約 0.2 以内で真の生成パラメータ $(0.6, 0.4)$、$(-2, 1.5)$、$(0.8, 0.6)$ の近くに落ち着く。

このようにきれいに収束する一方で、EM には知っておくべき 4 つの限界がある。1 つ目は無害なものである。K 個の成分のラベルを入れ替えると、あらゆる最大値に対して $K!$ 個の同一なコピーが生じる。この**ラベルスイッチング**は、ニューラルネットワークの隠れユニットを入れ替えたときに生じる識別不能性（non-identifiability）と同じものであり、これによってある解が別の解より劣るということはない。

2 つ目の限界はより重大である。EM は最初に到達した停留点（stationary point）で停止してしまい、そこが大域的最大値であるかどうかは問わない。対称な初期化はこの現象を鮮明に映し出す。次のセルを実行する前に少し手を止め、両方の成分がまったく同じ平均とスケールから出発したとき EM がどう振る舞うか予想してみてほしい。

```
lls2, pi2, mu2, sigma2 = em(x, [0.5, 0.5], [0, 0], [1, 1], 30)
print(f'log-likelihood {lls2[0]:.1f} -> {lls2[-1]:.1f}, '
      f'{lls[-1] - lls2[-1]:.1f} below the previous fit')
print(f'pi = {pi2.round(2)}, mu = {mu2.round(2)}, sigma = {sigma2.round(2)}')
print(f'mean of x = {x.mean():.2f}, standard deviation of x = {x.std():.2f}')
```

```
log-likelihood -1438.7 -> -1027.5, 180.7 below the previous fit
pi = [0.5 0.5], mu = [-0.59 -0.59], sigma = [1.89 1.89]
mean of x = -0.59, standard deviation of x = 1.89
```

2 つの成分は決して分離しない。まったく同じ状態で始まるため、あらゆるデータ点に対して同一の負担率を受け取り、M ステップでも互いに貼り付いたまま更新されるからである。その結果、EM はデータセット全体の平均と標準偏差を当てはめただけの単一ガウス分布の解に落ち着き、2 クラスタの解より 100 nat 以上も低い値にとどまる。EM のすべての不動点（fixed point）は尤度の停留点ではあるが、必ずしも最大点とは限らない。実務において複数のランダム初期化から EM を再実行するのはこのためである。ニューラルネットワークの層における隠れユニット間の

まったく同じ対称性を破るのもランダム初期化である（4.4 節）。等しい初期値から出発したユニットは等しい勾配を受け取り、永久に等しいままにとどまる。

3 つ目の限界は連続な混合モデル特有の性質であり、尤度そのものが上に有界でないことに起因する。第 2 成分を最初のデータ点の真上（$\mu_2 = x_1$）に置き、重みを $\pi_2 = 1/n$、標準偏差を ϵ としよう。すると x_1 におけるその密度は $\frac{1}{n}(2\pi)^{-1/2}\epsilon^{-1}$ となる一方、第 1 成分は残りの $n-1$ 個の点をそのまま覆い続けるため、次式が得られる。

$$\ell(\theta) \le \log\left(n\sqrt{2\pi}\right) + \log\epsilon + \text{const},$$

ここで定数項は他の $n-1$ 個の点からの寄与をまとめたものである。$\epsilon \to 0$ とすると負の対数尤度は $-\infty$ へと発散するため、最尤法による大域的な「解」は、単一のデータ点の上に立つ無限に鋭いスパイクにすぎないことになる（Figure 36.36）。分散に下限を設けるか、分散に事前分布（prior）を置けば、このような退化したスパイクは取り除ける。初期点と同等以上に良いパラメータの集合がコンパクトになりさえすれば、EM のすべての集積点は正則化された問題の停留点になることが保証される (Wu 1983)。

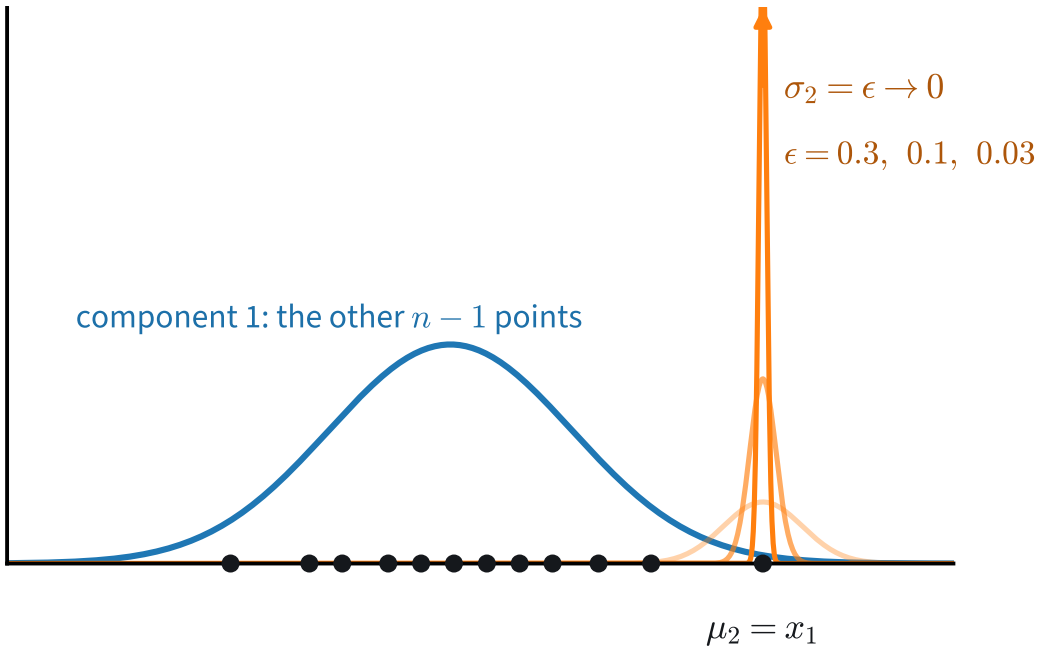


Figure 36.36: 混合ガウスモデルの尤度における退化した最大値。第 1 成分（青）がデータ点（黒点）のうち $n-1$ 個を覆い、第 2 成分（オレンジ）が残りの 1 点 $\mu_2 = x_1$ の上に標準偏差 $\sigma_2 = \epsilon$ で乗っている。ϵ が 0.3、0.1、0.03 へと縮むにつれ、x_1 における密度は $1/\epsilon$ に比例して増大し、それに伴って尤度も際限なく大きくなる。

4 つ目の限界は、EM の枠組みそのものを超えた先へとつながるものである。単調性の証明は、E ステップに**厳密な**事後分布を代入することに依存していた。そのため、近似的な E ステップで妥協せざるを得ない場合にはその保証が失われる。たとえば**変分オートエンコーダ** (Kingma and Welling 2014) は、深層非線形生成器が閉形式の事後分布を持たないため、ELBO という同じ目的関数を保ちつつも、データ点ごとの厳密な事後分布を、デコーダと同時に訓練されるエンコーダ

のニューラルネットワーク $q_\phi(z \mid x)$ に置き換える。そして厳密な EM ステップの代わりに、確率的勾配を用いて ELBO を登るのである。

ここに至って、下界はそれを導入したアルゴリズム自身の枠を超えて広がりを見せる。ELBO において隠れ変数 z がクラスタのラベルである必要はどこにもない。z の位置にモデルパラメータそのものを置けば、E ステップが求める事後分布はパラメータに関する事後分布となり、それこそ 36.6 節の末尾で求めたかった対象にほかならない。この事後分布も閉形式で得られることは稀であるため、続く 36.8 節では、それを近似しサンプリングするための実践的な手法を扱う。

36.7.4 まとめ

モデルが隠れ変数を含むとき、その周辺尤度は対数の内側に和を閉じ込めてしまい、本来なら単純なカウントと平均化で済むはずの問題を閉形式解のない最適化問題へと変えてしまう。隠れ変数に対する確率的推測 q を導入すると、扱いやすい完全データの目的関数が下界として復活する。すなわち、ELBO は q から真の潜在事後分布への KL ダイバージェンスの分だけ、対数エビデンスの下側に位置する。EM はそのギャップを閉じる操作（E ステップ）と、得られた下界をパラメータに関して最大化する操作（M ステップ）を交互に繰り返す。各反復で尤度が決して減少しないことは保証されるものの、EM は最適でない停留点で停止することがあり、連続な混合モデルでは退化したスパイクを防ぐために分散の下限や事前分布を必要とし、さらに単調性の保証を保つには厳密な事後分布を必要とする。

36.7.5 演習問題

1. [code] **EM の速度と欠損情報量**。●混合分布 $p(x \mid \pi) = \pi f_0(x) + (1-\pi) f_1(x)$ を考える。ただし成分 $f_0 = \mathcal{N}(\cdot\,; 0, 1)$ と $f_1 = \mathcal{N}(\cdot\,; \Delta, 1)$ は既知であり、混合重み π のみが未知であるとする。
   1. EM が重みを $\pi^{(t+1)} = M(\pi^{(t)})$ によって更新することを示せ。ただし $M(\pi) = \frac{1}{n}\sum_i r_i(\pi)$ であり、負担率は $r_i(\pi) = \pi f_0(x_i) / p(x_i \mid \pi)$ である。
   2. 不動点 $\hat\pi = M(\hat\pi)$ の近傍では、各反復で誤差が近似的に $M'(\hat\pi)$ 倍される。観測情報量を $I_{\text{obs}} = -\frac{d^2}{d\pi^2}\sum_i \log p(x_i \mid \pi)$（$\hat\pi$ で評価したもの）、ラベルが観測されたとした場合にラベルが持つ完全データ情報量を $I_{\text{com}} = n / (\hat\pi(1-\hat\pi))$ と定義する（36.6 節と比較せよ）。このとき次式を示せ。

$$M'(\hat\pi) = 1 - \frac{I_{\text{obs}}}{I_{\text{com}}}. \tag{36.95}$$

      両辺を負担率で表し、$\sum_i r_i(\hat\pi) = n\hat\pi$ を用いよ。Dempster et al. (1977) はこの結果の一般形を証明している。すなわち、EM の収束率は潜在変数によって隠された情報量の割合に支配される。
   3. $n = 10{,}000$ 個の点を $\pi = 0.3$ として各 $\Delta \in \{0.5, 1, 2, 4\}$ について抽出し、$\pi^{(0)} = 0.5$ から EM を実行して、連続する誤差の比の実測値を (b) からの予測と比較せよ。十分に長い実行から得た $\hat\pi$ を基準としたとき、各 Δ で誤差を 10^{-6} 未満にするには何回の反復が必要か。
   4. ラベルが観測された同じサイズのデータセットと比較したとき、比 $I_{\text{obs}}/I_{\text{com}}$ は $\hat\pi$ の標準誤差について何を意味するか。
2. **手計算による下界の確認**。本文で扱った単一の観測 $x = 0.5$ と、等しい重みを持つ 2 つの成分 $\mathcal{N}(-1, 1)$ および $\mathcal{N}(1, 1)$ を考える。
   1. $\log p(x)$ を計算し、あわせて ELBO Equation 36.92 を一様な推測 $q = (\tfrac12, \tfrac12)$ と事後分布 $q = (0.27, 0.73)$ のそれぞれについて計算して、本文に挙げた値と一致することを確かめよ。

2. KL ダイバージェンスを直接計算することで、一様な推測に対して Equation 36.93 が成り立つことを確かめよ。
3. ハードな推測では q のすべてを右側の成分に置く。このときの ELBO とそのギャップを計算せよ。ハードな推測と M ステップを交互に繰り返すアルゴリズムでも ELBO は決して減少しないが、対数尤度が増加する保証がなくなるのはなぜか。

3. **極限としての K-means**。混合ガウスモデルの K 個すべての成分が等しい重みと単一の固定分散 σ^2 を共有し、平均のみを当てはめるものとする。
1. 負担率 r_{ik} を、$-(x_i - \mu_k)^2/(2\sigma^2)$ の k にわたるソフトマックスとして表せ。温度の役割を果たすのはどの量か。また、最近傍の平均が一意である限り、$\sigma^2 \to 0$ のときその最近傍の平均に対する負担率が 1 に収束することを示せ。
2. このとき M ステップ Equation 36.94 が各平均をそこに割り当てられたデータ点の平均値に設定すること、および両ステップを合わせると $\sum_i (x_i - \mu_{z_i})^2$（ただし z_i はデータ点 i に割り当てられた成分）を決して増加させないことを示せ。
3. 本文の対称な初期状態では、両方の平均が同じ点から始まる。そのような初期状態から K-means はどのように動作するか。また、このことは実装において同値を解消する方法について何を示唆するか。

36.8 ベイズ計算

本書の「基礎」パート全体を通じて、最尤法（maximum likelihood）、あるいは正則化項を伴う事後分布（posterior distribution）のピークにおける最大事後確率推定（maximum a posteriori、MAP）によって、各モデルを単一のパラメータベクトルへと訓練してきた。36.6 節ではさらに一歩進め、その推定値にエラーバーを付した。それでもなお、訓練データをほぼ同様にうまく説明できるパラメータベクトルが他に数多く存在するにもかかわらず、予測は依然としてその単一のベクトルから得ていた。こうしたもっともらしいモデル同士はどの程度食い違うのか、そして食い違うときには何を予測すべきだろうか。

具体例として、入力が 0 から 4 の間にある 20 個のラベル付きデータ点に、切片と傾きを持つロジスティック回帰を当てはめる場合を考えよう。訓練データよりはるかに小さいテスト入力 -3 において、当てはめた MAP の直線はラベル 1 に約 1,000 分の 4 の確率を割り当てる。ところが、その 20 個の点にほぼ同じくらいよく当てはまる他のパラメータベクトルは、その入力において何桁も異なる予測を出す。データが与えられた下での各ベクトルのもっともらしさで重み付けして予測を平均すると、得られる確率はほぼ 5 倍大きくなる。一方、訓練データのちょうど内側にある入力 2 では、単一ベクトルによる予測と平均された予測は 0.01 以内で一致する。不正検知のアラートや医療診断のような稀な事象では、予測確率における 5 倍の差が意思決定を左右しうる。

この重み付き平均こそがベイズ予測であり、その計算はベイズの定理そのものと同じくらい古い問題である。事後分布はエビデンス、すなわちすべてのパラメータベクトルにわたる積分で割ることで得られるが、この積分が閉じた形で求まることは滅多にない。1774 年、Laplace は比率の事後分布をそのピークに一致するガウス分布で置き換えることで積分を回避した。1953 年には、Metropolis らが正規化定数（normalizing constant）の未知な分布に比例してランダムウォークに各状態を訪問させる方法を示した (Metropolis et al. 1953)。1990 年代までには、研究者たちはこの積分問題を最適化問題へと変換する手法も確立した。これらすべての手法を貫く統一的な仕掛けは 1 つである。いずれの手法も密度の比のみに依拠しており、そこでは未知の定数が約分されて消えるため、正規化定数を除いてしかわからない分布の上で平均を計算できるのである。

本節では、4 つの問いを順に立てることで 4 つのアプローチを展開する。第 1 に、事後分布をそのピークに一致するガウス分布で近似できるか。第 2 に、より単純な分布からサンプルを引き、各サンプルを補正できるか。第 3 に、事後確率質量に比例した時間を各領域で過ごすランダムウォークを作れるか。第 4 に、単純な分布を事後分布にできるだけ近づけるよう最適化できるか。最初の 3 つの手法はそれぞれ直前の手法の限界を克服するものであり、4 つ目の手法はサンプリン

グを最適化に置き換える。定数倍を除いてのみ既知の分布を扱ううえで、低次元の事後分布は最も見通しのよい舞台であるが、同じ発想は本書の随所で再び現れる。それぞれがどこで再登場するかは 36.8.5 節で整理する。これら 4 つの手法はいずれも、新しい入力 $\mathbf{x}_\star$ におけるラベル $y_\star$ の**事後予測**（posterior predictive）分布を次式により計算することを目的とする。

$$p(y_\star \mid \mathbf{x}_\star, \mathcal{D}) = \int p(y_\star \mid \mathbf{x}_\star, \theta)\, p(\theta \mid \mathcal{D})\, d\theta, \tag{36.96}$$

この式は、データ $\mathcal{D}$ が与えられた下での事後密度を重みとして、あらゆるパラメータベクトルの予測を平均したものである。

36.8.1 ベイズロジスティック回帰の事後分布

近似手法を公平に比較するには、閉じた形を持たないにもかかわらず厳密解と突き合わせて検証できる事後分布が必要である。コインに対するベータ事前分布が何回コイントスを繰り返してもベータ分布のままであるように、事後分布が事前分布と同じ分布族にとどまるとき、事前分布は尤度に対して**共役**（conjugate）であった（36.4 節）。ガウス事前分布はロジスティック尤度に対して共役ではないため、任意のパラメータベクトルにおいて未正規化事後分布の値は評価できても、それを解析的に積分することはできない。しかし、ここで通して用いる例にはパラメータが 2 つしかないため、細かい 2 次元グリッドを使えば、あらゆる近似の良し悪しを判定するための厳密な基準値が得られる。

36.8.1.1 モデルと未正規化事後分布

1 次元のロジスティック回帰では、各二値ラベル $y_i \in \{0, 1\}$ の確率は入力 x_i の線形関数によって次のように定まる。

$$P(y_i = 1 \mid x_i, \theta) = \mathrm{sigmoid}(\theta_0 + \theta_1 x_i) = \mathrm{sigmoid}(\mathbf{z}_i^\top \theta), \qquad \mathrm{sigmoid}(z) = \frac{1}{1 + e^{-z}},$$

ここで、$\mathbf{z}_i = (1, x_i)^\top$ である。予測確率が $\tfrac{1}{2}$ を横切るのは線形関数がゼロを横切る $x = -\theta_0/\theta_1$ においてであり、傾き θ_1 はその点での曲線の立ち上がりの急峻さを制御する。切片と傾きには独立なガウス事前分布（Gaussian prior）$\theta_j \sim \mathcal{N}(0, \tau^2)$（事前標準偏差は τ）を置く。ベイズの定理により、事後分布は n 個のラベルの尤度と事前分布の積に比例する。

$$p(\theta \mid \mathcal{D}) \propto \left[\prod_{i=1}^{n} \mathrm{sigmoid}(\mathbf{z}_i^\top \theta)^{y_i} \big(1 - \mathrm{sigmoid}(\mathbf{z}_i^\top \theta)\big)^{1 - y_i}\right] \exp\!\left(-\frac{\|\theta\|^2}{2\tau^2}\right). \tag{36.97}$$

角括弧内の積が尤度であり、指数関数の因子が事前分布である。ここに欠けている正規化定数は**エビデンス** $p(\mathcal{D}) = \int p(\mathcal{D}, \theta)\, d\theta$ であり、すべてのパラメータベクトルにわたる閉じた形を持たない積分である。右辺を**未正規化事後分布**（unnormalized posterior）と呼んで $\tilde{p}(\theta) = p(\mathcal{D}, \theta)$ と表せば、正規化された事後分布は単に $\tilde{p}/p(\mathcal{D})$ となる。

本節を通して用いるデータセットとして、$n = 20$ 個の入力を $[0, 4]$ から一様に抽出し、真のパラメータベクトル $\theta^\star = (-3, 1.5)$ からラベルを生成する。この確率曲線は $\tfrac{1}{2}$ を $x = 2$ で横切り、$\mathrm{sigmoid}(-3) \approx 0.05$（$x = 0$）から $\mathrm{sigmoid}(3) \approx 0.95$（$x = 4$）まで上昇する。事前分散は $\tau^2 = 4$ に設定する。データ点が 20 個であれば、本節で扱うあらゆる近似の効果がはっきりと浮かび上がるだけの広がりが事後分布に残る。実行のたびに新しいデータセットを生成するのでは

なく、具体的な基準値を引用できるよう、代表的な 20 個の点のサンプルを 1 つ固定する（以下で測定する事後分布の要約統計量に関して、1,500 個のシミュレーションデータセットの中から最も典型的なものを選んだ）。実行する各サンプラは毎回新しい乱数を使うため、サンプラの結果は典型的な範囲として報告する。コード中では、`log_joint` が $\log \tilde{p}$ を評価する際、大きなロジットでのオーバーフローを防ぐために `logaddexp(0, z)` を用いて $\log(1 + e^{z})$ を計算している。

```
rng = np.random.default_rng()
n, prior_var = 20, 4.0                       # prior_var is tau^2
theta_star = np.array([-3.0, 1.5])

def sigmoid(z):
    return np.exp(-np.logaddexp(0.0, -z))

x = np.array([0.1, 0.3, 0.3, 0.4, 0.8, 1.4, 1.5, 1.6, 1.8, 2.1,
              2.2, 2.3, 2.6, 2.7, 3.0, 3.1, 3.4, 3.5, 3.6, 3.7])
y = np.array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1])
Z = np.column_stack([np.ones(n), x])         # rows z_i = (1, x_i)

def log_joint(theta, Z=Z, y=y, prior_var=prior_var):
    """log p(D, theta) up to a constant; theta has final dimension 2."""
    logits = np.einsum('...d,nd->...n', theta, Z)
    log_lik = (y * logits - np.logaddexp(0.0, logits)).sum(axis=-1)
    return log_lik - 0.5 * (theta**2).sum(axis=-1) / prior_var

print(f'{n} points, {y.sum()} of them labeled 1')
```

```
20 points, 10 of them labeled 1
```

36.8.1.2 勾配とヘッセ行列

以下のいくつかの手法は $\log \tilde{p}$ の傾きと曲率に依拠するため、最初に一度だけ導出しておくと見通しがよい。各対数尤度項 $y_i \, \mathbf{z}_i^{\top}\theta - \log(1 + e^{\mathbf{z}_i^{\top}\theta})$ を微分すると $(y_i - p_i)\,\mathbf{z}_i$（ただし $p_i = \mathrm{sigmoid}(\mathbf{z}_i^{\top}\theta)$）が得られ、ガウス事前分布からは $-\theta/\tau^2$ が加わる。入力行 $\mathbf{z}_i^{\top}$ を縦に並べて $n \times 2$ 行列 $\mathbf{Z}$ とすれば、勾配 $\nabla \log \tilde{p}(\theta) = \mathbf{Z}^{\top}(\mathbf{y} - \mathbf{p}) - \theta/\tau^2$ が得られる。その幾何学的な意味は明快である。各訓練点は θ を $\mathbf{z}_i$ の方向に予測残差 $y_i - p_i$ に比例して押し動かし、事前分布はパラメータを原点へと穏やかに引き戻す。もう一度微分すると、次の負のヘッセ行列が得られる。

$$\mathbf{H}(\theta) = -\nabla^2 \log \tilde{p}(\theta) = \mathbf{Z}^{\top} \operatorname{diag}\big(p_i(1 - p_i)\big)\mathbf{Z} + \tau^{-2}\mathbf{I}. \tag{36.98}$$

ここでデータの項は、入力の重み付きグラム行列（Gram matrix）である。各重み $p_i(1-p_i) \le \tfrac{1}{4}$ は $p_i = \tfrac{1}{2}$ で最大となるため、モデルの不確実性が最も高い点ほど曲率に大きく寄与する。事前項 $\tau^{-2}\mathbf{I}$ を加えることで、すべての固有値が少なくとも τ^{-2} だけ押し上げられるため、$\mathbf{H}(\theta)$ は至るところで狭義正定値となる。したがって対数事後密度は狭義凹関数であり、唯一の最大値を持つ。たとえ訓練データが線形分離可能であっても、対数尤度が正になることはない一方で二次形式の事前ペナルティはあらゆる方向へ際限なく減少するため、その最大値は有限にとどまる。

```
def grad_log_joint(theta, Z=Z, y=y, prior_var=prior_var):
    """Gradient of log p(D, theta); one row per row of theta."""
```

```
    theta = np.atleast_2d(theta)
    return (y - sigmoid(theta @ Z.T)) @ Z - theta / prior_var

def neg_hessian(theta, Z=Z, y=y, prior_var=prior_var):
    """H(theta) = Z^T diag(p (1 - p)) Z + I / tau^2."""
    p = sigmoid(Z @ theta)
    return Z.T @ (Z * (p * (1 - p))[:, None]) + np.eye(2) / prior_var
```

パラメータが 2 つしかなければ、これらの導関数を手で書き下すのは造作もない。より大規模なモデルでは、$\log \tilde{p}$ のみを実装し、勾配とヘッセ行列は自動微分に任せることになる。

なお、本節のすべての手法は、値・勾配・ヘッセ行列のいずれを用いるにせよ、もっぱら $\tilde{p}$ のみを通じて事後分布にアクセスする。

36.8.1.3 事後最頻値とラプラス近似

最初の最も計算コストの低い近似は、すでに実行方法を知っている最適化の上に直接構築される。事後分布のピークは MAP 推定値、すなわち 36.6 節で扱ったペナルティ付きの当てはめであり、ニュートン法 $\theta \leftarrow \theta + \mathbf{H}(\theta)^{-1}\nabla \log \tilde{p}(\theta)$ によってわずか数回の反復でそこに到達できる。あとはヘッセ行列をもう 1 回評価するだけで、その点推定（point estimate）はガウス分布へと変わる。

具体的には、事後最頻値（posterior mode）θ_{MAP} を中心とするガウス分布を当てはめる。$\log \tilde{p}$ を θ_{MAP} の周りで 2 次までテイラー展開（Taylor expansion）すると、$\mathbf{H} = \mathbf{H}(\theta_{\mathrm{MAP}})$ として次式が得られる。

$$\log \tilde{p}(\theta) \approx \log \tilde{p}(\theta_{\mathrm{MAP}}) - \tfrac{1}{2}(\theta - \theta_{\mathrm{MAP}})^{\top}\mathbf{H}(\theta - \theta_{\mathrm{MAP}}). \tag{36.99}$$

最頻値では勾配が消えるため、1 次の項は現れない。加法的な定数を除けば、右辺は平均 θ_{MAP}、共分散 $\mathbf{H}^{-1}$ を持つガウス分布の対数密度にほかならない。したがって、指数をとることで次の**ラプラス近似**（Laplace approximation）が得られる。

$$p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\theta_{\mathrm{MAP}}, \mathbf{H}^{-1}). \tag{36.100}$$

この近似が適切に定義されるためには、$\log \tilde{p}$ が内点にある最頻値で 2 回微分可能であり、負のヘッセ行列 $\mathbf{H}$ が正定値でなければならないが、ここではその両方が成り立っている。直観的には、鋭いピークは曲率が大きいため、分散の小さい幅の狭いガウス分布を生み出す。このガウス分布が一致させるのは事後分布のピークの位置と局所的な曲率だけであり、それ以外の性質は一切反映しない。

1 次元の例を見れば、この振る舞いは一目瞭然である。歪んだ事後分布 $\mathrm{Gamma}(4, 1)$ を考えよう。その未正規化密度は $\theta^3 e^{-\theta}$（$\theta > 0$）である。対数密度 $3\log\theta - \theta$ の導関数は $3/\theta - 1$ であり、最頻値 $\theta = 3$ でゼロとなり、その点での負の二階導関数は $3/\theta^2 = \tfrac{1}{3}$ となる。したがってラプラス近似は $\mathcal{N}(3, 3)$ となり、標準偏差は $\sqrt{3} \approx 1.73$ である。厳密な事後分布は平均 4、分散 4 を持つため、このガウス分布は左に寄りすぎているうえに幅も狭すぎ、さらには質量の $\Phi(-\sqrt{3}) \approx 4.2\%$ を、真の密度がゼロである $\theta = 0$ 未満に割り当ててしまう（Figure 36.37）。

一様事前分布の下で表が 9 回、裏が 4 回出たコインに同じ計算を適用すると、36.6.3.2 節で求めた最尤推定量（maximum likelihood estimate、MLE）$9/13$ とその標準誤差 0.128 が復元される。実際、平坦な事前分布の下では、ラプラス近似は単に MLE とその漸近的なエラーバーを組み合わせたものにほかならない（演習問題 3）。

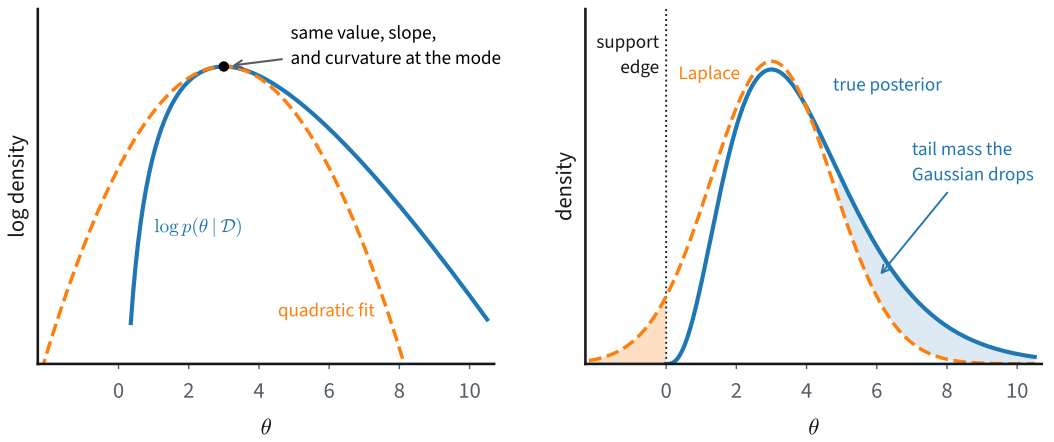


Figure 36.37: 歪んだ 1 次元事後分布である最頻値 3 の Gamma(4,1) 密度に対するラプラス近似。左：対数密度への二次近似は、最頻値における値・傾き・曲率のみに一致し、それ以外には一致しない。右：指数をとると、ラプラスガウス分布 $\mathcal{N}(3, 3)$ はピークには一致するものの歪みを捉え損ね、質量の 4.2%（オレンジ色の網掛け）を事後分布がゼロであるサポート境界 $\theta = 0$ 未満に割り当てる一方で右裾を過小評価している。真の平均は 4 である。

未知のエビデンスがこの計算に一切現れないことに注目しよう。$\log p(\mathcal{D})$ は $\log p(\theta \mid \mathcal{D})$ における加法的な定数であるため、微分すると消えるからである。さらに都合のよいことに、Equation 36.99 の右辺の指数を積分すれば、エビデンス自体の推定値 $p(\mathcal{D}) \approx \tilde{p}(\theta_{\text{MAP}})(2\pi)^{d/2} \det(\mathbf{H})^{-1/2}$（$d$ 次元の場合）まで得られる（演習問題 6）。

ラプラス近似は単一のピークにおける局所的な導関数しか用いないため、歪度（skewness）、重い裾、パラメータの境界、他の最頻値（モード）を捉えることができず、分布の中心も平均ではなく最頻値に置かれる。スケーリングも課題となる。d 個のパラメータを持つニューラルネットワークでは $\mathbf{H}$ の要素数が d^2 個に上るため、実応用では対角成分のみを残すか、近似を最終層のみに制限するのが一般的である。

コード上では、原点から出発する 10 回のニュートンステップで最頻値が求まり、1 回の逆行列計算でラプラス共分散が得られる。

```
theta_map = np.zeros(2)
for _ in range(10):                                    # Newton's method
    theta_map += np.linalg.solve(neg_hessian(theta_map),
                                 grad_log_joint(theta_map)[0])
laplace_cov = np.linalg.inv(neg_hessian(theta_map))
laplace_sd = np.sqrt(np.diag(laplace_cov))
laplace_rho = laplace_cov[0, 1] / (laplace_sd[0] * laplace_sd[1])
print('MAP               :', theta_map.round(3))
print('Laplace sd        :', laplace_sd.round(3))
print('Laplace correlation:', round(laplace_rho, 3))
```

```
MAP               : [-2.138  1.117]
Laplace sd        : [1.004 0.457]
Laplace correlation: -0.861
```

事後最頻値は $(-2.14, 1.12)$ にあり、データ生成パラメータ $(-3, 1.5)$ よりも緩やかな傾きの直

線となっている。ラプラス標準偏差は切片で 1.00、傾きで 0.46 であり、真の値は各座標において標準偏差の 1 倍弱だけ離れた位置にある。また、-0.86 という強い負の相関にも注目しよう。傾きが急な直線ほど、同じ点群を通過するためにはより低い位置から出発しなければならないからである。

36.8.1.4 グリッドによる基準値

ラプラス近似が実際どれほど正確かを確かめるには、ベンチマークとして厳密な事後分布が必要である。パラメータが 2 つしかなければ、細かい $m \times m$ のグリッド上で $\log \tilde{p}$ を評価し、数値安定性のために最大値を引いてから指数をとり、重みの和が 1 になるよう正規化できる。d 次元ではこのような格子（lattice）に m^d 回の評価が必要になるため、グリッド積分はあくまで低次元専用の基準手法である。2 次元であっても領域をどこに置くかは決める必要がある。ここでは θ_{MAP} を中心として各方向にラプラス標準偏差の 7 倍の範囲をとることで、領域外に無視できるほどの裾しか残らないようにする。

```
k, m = 7, 241                    # half-width in Laplace sds, points per axis
grid_axes = [np.linspace(c - k * s, c + k * s, m)
             for c, s in zip(theta_map, laplace_sd)]
B0, B1 = np.meshgrid(*grid_axes, indexing='ij')
grid_theta = np.stack([B0.ravel(), B1.ravel()], axis=1)
grid_logp = log_joint(grid_theta)
grid_weight = np.exp(grid_logp - grid_logp.max())
grid_weight /= grid_weight.sum()
post_mean = grid_weight @ grid_theta
centered = grid_theta - post_mean
post_cov = (centered * grid_weight[:, None]).T @ centered
post_sd = np.sqrt(np.diag(post_cov))
post_rho = post_cov[0, 1] / (post_sd[0] * post_sd[1])
print('grid mean       :', post_mean.round(3), '  MAP:', theta_map.round(3))
print('grid sd         :', post_sd.round(3), '  Laplace:', laplace_sd.round(3))
print('grid correlation:', round(post_rho, 3))
print('(mean - MAP)/sd :', ((post_mean - theta_map) / post_sd).round(2))
```

```
grid mean       : [-2.347  1.237]   MAP: [-2.138  1.117]
grid sd         : [1.042 0.48 ]   Laplace: [1.004 0.457]
grid correlation: -0.858
(mean - MAP)/sd : [-0.2   0.25]
```

グリッド計算によって、ラプラス近似が見落としていた歪度が明らかになる。傾きの事後平均（posterior mean）1.24 は MAP 推定値 1.12 より事後標準偏差の 4 分の 1 だけ上にあり、切片の平均 -2.35 は MAP 推定値より標準偏差の 5 分の 1 だけ下にある。ラプラス標準偏差は 4〜5% 狭すぎ、相関は 2 桁まで一致している。こうした幅の誤差は控えめなものであるが、より重大な食い違いは、事後分布が急な傾きの直線側へと歪んでいるために平均と最頻値が一致しないことである。

36.8.1.5 プラグイン予測と事後予測

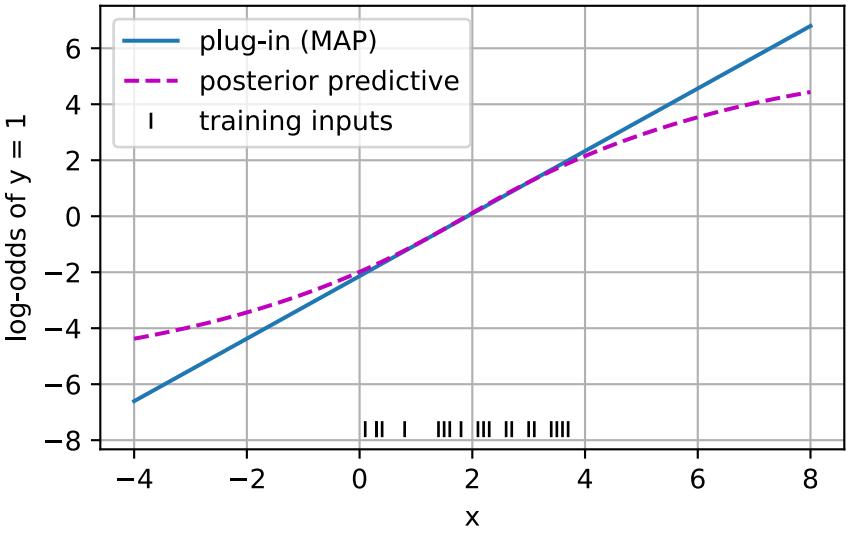
これで冒頭の問いに直接答える準備が整った。**プラグイン**予測 $\text{sigmoid}(\mathbf{z}_\star^\top \theta_{\text{MAP}})$ は最頻値のみに依拠するのに対し、事後予測 Equation 36.96 は事後分布の重みを使ってグリッド全体で $\text{sigmoid}(\mathbf{z}_\star^\top \theta)$ を平均する。訓練データの下側・内側・上側に位置する 3 つのテスト入力 $x_\star \in \{-3, 2, 7\}$ において両者の予測を、もっともらしいパラメータベクトルにわたる予測値の中央 90% を含む区間とともに比較してみよう。

```
x_new = np.array([-3.0, 2.0, 7.0])
Z_new = np.column_stack([np.ones(len(x_new)), x_new])
grid_probs = sigmoid(grid_theta @ Z_new.T)     # each grid vector's answer
plug_in = sigmoid(Z_new @ theta_map)
predictive = grid_weight @ grid_probs
for j, x0 in enumerate(x_new):
    order = np.argsort(grid_probs[:, j])
    cdf = np.cumsum(grid_weight[order])
    lo, hi = grid_probs[order[np.searchsorted(cdf, [0.05, 0.95])], j]
    print(f'x={x0:+.0f}: plug-in {plug_in[j]:.4f}, predictive '
          f'{predictive[j]:.4f}, middle 90%: [{lo:.2g}, {hi:.4f}]')
a = grid_theta @ Z_new[0]                      # logits at x = -3
mu = grid_weight @ a
s2 = grid_weight @ (a - mu)**2
print(f'x=-3, log scale: plug-in {np.log(plug_in[0]):.2f}, '
      f'mean logit {mu:.2f}, plus s^2/2 {mu + s2 / 2:.2f},')
print(f'                 log E[e^a] {np.log(grid_weight @ np.exp(a)):.2f}, '
      f'predictive {np.log(predictive[0]):.2f}')
```

```
x=-3: plug-in 0.0041, predictive 0.0187, middle 90%: [3.8e-05, 0.0887]
x=+2: plug-in 0.5238, predictive 0.5292, middle 90%: [0.32, 0.7364]
x=+7: plug-in 0.9966, predictive 0.9826, middle 90%: [0.92, 1.0000]
x=-3, log scale: plug-in -5.49, mean logit -6.06, plus s^2/2 -3.19,
                 log E[e^a] -3.66, predictive -3.98
```

訓練データの範囲内にある $x_\star = 2$ では、2 つの予測（0.524 と 0.529）は 0.01 以内で一致し、もっともらしいパラメータベクトルによる予測も 0.32 から 0.74 という穏やかな範囲に収まる。データから大きく外れた $x_\star = -3$ では、もっともらしい予測の中央 90% は 4×10^{-5} から 0.09 まで 3 桁以上にわたって広がり、事後予測（0.019）はプラグイン値（0.0041）の 4.6 倍になる。$x_\star = 7$ でも、ラベル 0 の確率に対して同じ増幅が生じる（0.017 対 0.003）。これほど 0 や 1 に近い確率は対数オッズ（log odds）スケール $\log \frac{P}{1-P}$ で比較するのが最も見やすいため、下端に訓練入力の位置を記したうえで、より広い範囲にわたって両方の曲線をプロットしよう。

```
def log_odds(p):
    return np.log(p) - np.log1p(-p)

xs = np.linspace(-4, 8, 121)
Z_xs = np.column_stack([np.ones_like(xs), xs])
plug_xs = Z_xs @ theta_map                        # log-odds of the plug-in
pred_xs = log_odds(grid_weight @ sigmoid(grid_theta @ Z_xs.T))
rug = np.full(n, min(plug_xs.min(), pred_xs.min()) - 1)
d2l.plot([xs, xs, x], [plug_xs, pred_xs, rug],
         xlabel='x', ylabel='log-odds of y = 1', fmts=('-', 'm--', 'k|'),
         legend=['plug-in (MAP)', 'posterior predictive', 'training inputs'],
         figsize=(5, 3))
```



対数オッズスケールでは、プラグイン予測は単なる直線 $\theta_0 + \theta_1 x$ になる。$[0, 4]$ の内側では事後予測はこの直線にぴたりと沿うが、訓練データの外側ではゼロに向かって折れ曲がる。この乖離は 2 つの競合する効果によって生じる。第 1 に、事後平均は MAP 推定値よりも急な傾きを持つため、極端な予測を $\frac{1}{2}$ から**遠ざける**方向に押しやる。第 2 に、広がりを持つロジットの上でシグモイドを平均することは、予測を $\frac{1}{2}$ に**近づける**方向に引き寄せる。データから離れた領域では、傾きの緩やかなもっともらしいパラメータベクトルが典型的なベクトルよりもはるかに大きな確率を生み出し、その大きな確率が平均を支配するからである。

どちらの効果がどれだけ勝るのだろうか。簡単な概算によって、平均化の効果の大きさを見積もることができる。確率が小さい領域では $\mathrm{sigmoid}(a) \approx e^a$ であるから、予測確率はおおよそ e^a の事後平均（ただしロジットは $a = \mathbf{z}_\star^\top \theta$）となる。もし a が平均 μ、分散 s^2 のガウス分布に従うなら、積率母関数（モーメント母関数）より $\mathbb{E}[e^a] = e^{\mu + s^2/2}$ となり、対数確率（log probability）は平均ロジットにおける値より $s^2/2$ だけ押し上げられる。$x_\star = -3$ ではグリッド計算より $s^2/2 = 2.87$ となるため、$e^{2.87} \approx 18$ 倍の増加が予測されるはずである。ところが実際の倍率は 4.6 倍、対数スケールではわずか 1.51 の上昇にとどまる。消えた 1.36 nat はどこへ行ったのだろうか。上のコードの最後の 2 行は、プラグイン予測から事後予測までを 1 段階ずつたどっている。

$x_\star = -3$ における段階	対数確率	変化量
プラグイン：最頻値におけるロジット	-5.49	
ロジットの事後平均	-6.06	-0.57
ガウスロジットと同様に $s^2/2$ を加算	-3.19	$+2.87$
ロジットの実際の分布上で e^a を平均	-3.66	-0.47
$\mathrm{sigmoid}(a)$ を平均（e^a の代わり）：事後予測	-3.98	-0.32

3 つの下向き補正によって、2.87 と 1.51 の差はすべて説明がつく。第 1 に、$s^2/2$ の押し上げはプラグインのロジットではなく平均ロジットを起点としており、傾きがより急な事後平均は 0.57

だけ低い位置にある。第 2 に、ロジットはガウス分布ではなく歪んでいる。負の大きな値の方向へ長い裾を持ち、e^{a} の平均が質量のほぼすべてを得るゼロ方向への裾は短いため、さらに 0.47 削られる。第 3 に、平均を支配するパラメータベクトルは $x_{\star} = -3$ においてすでに微小とはいえない確率を持つため、シグモイドが e^{a} より下に曲がり、最後の 0.32 を差し引く。以上を合わせると、$2.87 - 0.57 - 0.47 - 0.32 = 1.51$ となる。

したがってガウス分布による計算は、厳密な法則というよりは有用な指針であり、変化の方向、大まかなスケール、および $x_{\star}$ に対する漸近的な依存性を正しく捉えている。具体的には、ロジットの分散 $\mathrm{Var}(\theta_0) + 2x_{\star}\mathrm{Cov}(\theta_0, \theta_1) + x_{\star}^2\mathrm{Var}(\theta_1)$ はデータからの距離に対して二次関数的に増大するのに対し、平均のシフトは線形にしか増大しない。訓練入力から遠く離れると必然的に分散の項が勝り、データの内側では両方の効果とも無視できるほど小さいままにとどまる（演習問題 4）。

なお、このより大きな予測確率が、データを生成した真の値に必ずしも近いわけではないことに注意しよう。真の直線は $\mathrm{sigmoid}(-3 - 1.5 \cdot 3) = \mathrm{sigmoid}(-7.5) \approx 0.0006$ を $x_{\star} = -3$ において与えるが、20 個の点では傾きを確定できず事前分布がそれをゼロに向かって縮小（shrinkage）させるため、この値はプラグイン予測と事後予測のどちらよりも小さい。それでもこの真の確率は、典型的なデータセットで期待されるとおり、90% 事後区間の内側に余裕をもって収まっている。事後予測は真のパラメータを知っていると主張するのではなく、事前分布と 20 個の観測に整合するすべてのパラメータベクトルにわたって平均した $y_{\star} = 1$ の厳密な確率を与えるのである。本節の残りの部分では、パラメータ空間が大きすぎてグリッドが使えないときに、このような平均をどう計算するかを問う。

36.8.2 棄却サンプリングと重要度サンプリング

ラプラス近似は高速である反面、最頻値を中心としており歪度を無視してしまう。もし厳密な事後分布から直接サンプルを抽出できれば、その経験平均によって任意の事後期待値をすっきり推定でき、歪度も裾も自動的に捉えられるはずである。本節で扱う 2 つの手法は、扱いやすい**提案**分布（proposal distribution）q からのサンプルを、比 $\tilde{p}/q$ のみに依拠して事後分布からの厳密なサンプルまたは重み付きサンプルへと変換する。

36.8.2.1 棄却サンプリング

最も単純な厳密サンプラである棄却サンプリング（rejection sampling）は、q から候補を引き、一部を残して残りを捨てる。サンプリングと評価が容易な提案分布 q に加え、定数 M を $M q(\theta) \geq \tilde{p}(\theta)$ がすべての θ で成り立つように選び、未正規化事後分布の上を覆う包絡線（エンベロープ）を作る。各サンプル $\theta \sim q$ に対し、次の確率でこれを受理して残す。

$$a(\theta) = \frac{\tilde{p}(\theta)}{M\, q(\theta)}. \tag{36.101}$$

幾何学的には、この受理確率は提案された点における包絡線の高さに対する目標分布の高さの比である。受理されたサンプルはすべて $p(\theta \mid \mathcal{D})$ からの厳密なサンプルとなり、全体の受理率は $p(\mathcal{D})/M$ となる。この確認は 2 行で済む。提案された点が微小領域 $d\theta$（θ の周囲）に入る確率は $q(\theta)\, d\theta$ であり、それが受理される確率は $a(\theta)$ であるから、その領域に入って残される同時確率（joint probability）は $\tilde{p}(\theta)\, d\theta / M$ となる。すべての θ にわたって積分すると周辺受理確率 $\int \tilde{p}/M = p(\mathcal{D})/M$ が得られ、同時密度をこの周辺確率で割れば、残されたサンプルの条件付き密度 $\tilde{p}(\theta)/p(\mathcal{D}) = p(\theta \mid \mathcal{D})$ が得られる。

重要なことに、未知のエビデンスは受理率にのみ現れ、受理・棄却の判定規則そのものには一切現れない。提案分布 q として事前分布を選べば、$\tilde{p}/q$ は単なる尤度となり、M はその最大値とな

る。すなわち、事前分布からパラメータベクトルを引き、最尤法における最大尤度でその尤度を割った確率で各サンプルを残せばよい。

データの蓄積に伴って棄却サンプリングがどれほど急速に劣化するかを見るため、一様事前分布の下で表が 9 回、裏が 4 回出たコインを考えよう。尤度 $\theta^9(1-\theta)^4$ は $\theta = 9/13$ でピークに達して高さ $M = (9/13)^9(4/13)^4 \approx 3.3 \times 10^{-4}$ をとり、一方エビデンスは $p(\mathcal{D}) = \int_0^1 \theta^9(1-\theta)^4\,d\theta = 9!\,4!/14! \approx 1.0 \times 10^{-4}$ であるから、受理率は 0.305 となる。データを 10 倍（表 90 回、裏 40 回）にすると受理率は 0.10 に落ち、100 倍にすると 0.03 まで下がる。事前分布が固定されたまま事後分布の幅が $1/\sqrt{n}$ のオーダーで縮むため、受理率は 1 次元では $1/\sqrt{n}$、また $n^{-d/2}$（d 次元の場合）のオーダーで減衰するのである。一様事前分布からの 100,000 個のサンプルによる簡単なシミュレーションでこの計算を確認できる。

```
N = 100_000
theta_coin = rng.random(N)                    # draws from the uniform prior
lik = theta_coin**9 * (1 - theta_coin)**4     # likelihood of 9 heads, 4 tails
M = (9 / 13)**9 * (4 / 13)**4                 # its maximum, at theta = 9/13
kept = theta_coin[rng.random(N) < lik / M]
print(f'acceptance {len(kept) / N:.3f}, mean {kept.mean():.3f}, '
      f'sd {kept.std():.3f}')
```

```
acceptance 0.304, mean 0.666, sd 0.118
```

実用的には 2 つの深刻な障壁が棄却サンプリングを制約する。第 1 に、妥当な上界 M を知っている必要があるが、単純なモデル以外でそれが得られることは稀である。第 2 に、パラメータあたりの観測数を一定に保ったとしても、受理率は次元 d に対して少なくとも指数関数的に減衰する。

36.8.2.2 重要度重み

棄却サンプリングは、せっかく生成したサンプルの大半を捨ててしまう。比 $w = \tilde{p}/q$ を使ってコイントスをして候補を捨てる代わりに、すべてのサンプルを残して w で重み付けすることができる。なぜこれがうまくいくのかを見るため、任意の事後期待値を 2 つの積分の比として表し、両方の積分の中で q を掛けて割ってみよう。

$$\begin{aligned}\mathbb{E}_{p(\cdot\mid\mathcal{D})}[h(\theta)] &= \frac{\int h\,\tilde{p}\,d\theta}{\int \tilde{p}\,d\theta} = \frac{\int h\,\frac{\tilde{p}}{q}\,q\,d\theta}{\int \frac{\tilde{p}}{q}\,q\,d\theta} \\ &= \frac{\mathbb{E}_q[w(\theta)h(\theta)]}{\mathbb{E}_q[w(\theta)]}, \qquad w(\theta) = \frac{\tilde{p}(\theta)}{q(\theta)}.\end{aligned} \tag{36.102}$$

分母 $\mathbb{E}_q[w]$ は未知のエビデンス $p(\mathcal{D}) = \int \tilde{p}$ に等しいが、同じ正規化定数が分子と分母の両方を定数倍するため、これを明示的に計算する必要はまったくない。2 つの期待値をサンプル $\theta_1, \ldots, \theta_N \sim q$ にわたる標本平均で置き換えると、**自己正規化重要度サンプリング**（self-normalized importance sampling）$\sum_s w_s h(\theta_s) / \sum_s w_s$ が得られる。大数の法則（law of large numbers）により、この比は $N \to \infty$ で厳密な事後期待値に収束するが、有限の N では 2 つの確率的な平均の比であるためにわずかな統計的バイアスを伴う。多数の尤度項の積は浮動小数点演算ですぐにアンダーフローを起こすため、log-sum-exp トリックと同様、常に対数重みを計算し、最大の対数重みを引いてから指数をとる。q が事前分布のとき、重要度重み（importance weight）は尤度そのものに帰着し、事前分布からの各サンプルは観測データをどれだけうまく説明できるかによって重み付けされる。

36.8.2.3 有効標本サイズ

すべてのサンプルを残すからといって、すべてのサンプルが等しく役に立つわけではない。たった 1 つのサンプルが重みのほぼすべてを占めるような重み付き標本は、N がどれほど大きくても実質的に 1 サンプル分の価値しかない。これを定量化するため、正規化された重み $\bar{w}_s = w_s / \sum_r w_r$ から**有効標本サイズ**（effective sample size、ESS）を次のように定義する。

$$N_{\text{eff}} = \frac{1}{\sum_{s=1}^{N} \bar{w}_s^2}. \tag{36.103}$$

すべての重みが等しいとき（$\bar{w}_s = 1/N$）は $N_{\text{eff}} = N$ となり、1 つのサンプルがすべての重みを独占するときは $N_{\text{eff}} = 1$ となる。大まかに言えば、ESS はその重み付き標本が厳密な事後分布からの独立サンプル何個分に相当するかを推定するものである。しかし、まもなく見るとおり、ESS は 2 つの重要な点で誤解を招くことがある。

命題。$N \to \infty$ のとき、次が成り立つ。

$$\frac{N_{\text{eff}}}{N} \to \frac{(\mathbb{E}_q[w])^2}{\mathbb{E}_q[w^2]} = \frac{1}{1 + \chi^2(p \,\|\, q)}, \qquad \chi^2(p \,\|\, q) = \int \frac{p^2}{q}\, d\theta - 1, \tag{36.104}$$

ここで、p は正規化された事後分布である。同じ提案分布 q に対して、この極限効率は常に棄却サンプリングの受理率 $p(\mathcal{D})/M$ 以上である。

証明。@eq-mdl-bayes-is-ess の分子と分母を N^2 で割ると $N_{\text{eff}}/N = (\frac{1}{N}\sum_s w_s)^2 / (\frac{1}{N}\sum_s w_s^2)$ となり、両方の平均に大数の法則が適用できる。$w = p(\mathcal{D})\, p/q$ を代入すると $\mathbb{E}_q[w^2] = p(\mathcal{D})^2 \int p^2/q$ および $\mathbb{E}_q[w] = p(\mathcal{D})$ が得られ、χ^2 ダイバージェンスによる表現が導かれる。最後に、$w \le M$ より $\mathbb{E}_q[w^2] \le M\, \mathbb{E}_q[w]$ であるから、この極限は少なくとも $\mathbb{E}_q[w]/M = p(\mathcal{D})/M$ 以上である。■

したがって ESS は、q が p にどれだけ近いかを表す経験的な指標である。大きな N の極限では、重要度サンプリングは包絡線の上界 M を一切必要とすることなく、棄却サンプリングが受理する数以上の有効サンプルを確保する。一様事前分布を提案分布とする先ほどのコインの例に戻ると、$\mathbb{E}_q[w^2] = \int_0^1 \theta^{18}(1-\theta)^8\, d\theta$ であり、極限の ESS 比率は 0.42 となって棄却サンプリングの受理率 0.305 を上回る。棄却サンプリングのコードセルで生成したまったく同じサンプルを再利用し、3 分の 2 を捨てる代わりに重み付けし直すだけでよいのである。

```
w = lik / lik.sum()                          # normalized weights
post_coin = w @ theta_coin
print(f'ESS fraction {1 / (w @ w) / N:.3f}, mean {post_coin:.3f}, '
      f'sd {np.sqrt(w @ (theta_coin - post_coin)**2):.3f}')
```

```
ESS fraction 0.420, mean 0.666, sd 0.118
```

この極限公式はまた、スケールの不適切な提案分布がいかに急激に破綻するかも暴き出す。標準偏差 s_p の 1 次元ガウス目標分布と、同じ中心を持ち標準偏差 $s_q = r s_p$ のガウス提案分布を考えると、極限の ESS 比率は $\sqrt{2r^2 - 1}/r^2$ となる（演習問題 5）。この曲線は 0.83（$r = 0.8$）をとり、最大値 1（$r = 1$）に達したのち、0.66（$r = 2$）、0.46（$r = 3$）へと緩やかに減少する。ところが $r \le 1/\sqrt{2}$ では、$\mathbb{E}_q[w^2] = \infty$ となるために突如としてゼロへと崩壊する。狭すぎる提案分布は、広すぎる提案分布よりもはるかに壊滅的な失敗を招くのである。

重要度サンプリングの恒等式 Equation 36.102 が成り立つことは、2 つの条件によって保証される。すなわち、$q(\theta) > 0$ が $\tilde{p}(\theta) > 0$ なるすべての点で成り立つこと、および $\mathbb{E}_q|wh| < \infty$ であることである。中心極限定理、ひいては $O(1/\sqrt{N})$ のエラーバーを得るには、さらに $\mathbb{E}_q[w^2]$ と $\mathbb{E}_q[w^2h^2]$ が有限である必要がある。しかもこれらの条件が満たされていても、Figure 36.38 が示すように、有限標本における ESS は危険なほど楽観的な値を出すことがある。

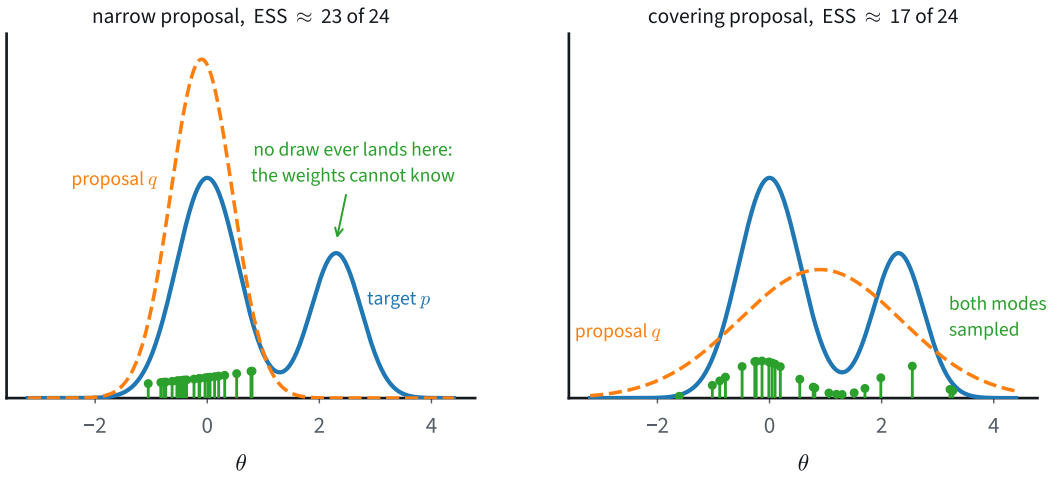


Figure 36.38: 2 つの異なる提案分布を用いた二峰性目標分布の重要度サンプリング。縦棒は 24 個のサンプルの正規化された重みを表す。左：幅の狭い提案分布は右側のモードを一度も訪れないため、推定値が大きく誤っているにもかかわらず重みはほぼ一様に見える（ESS は 23 / 24）。右：より幅の広い提案分布は両方のモードを覆っており、重みの均一性は下がる（ESS は 17 / 24）ものの、推定値ははるかに正確である。ESS が測るのは重みの不均等さであり、提案分布が事後確率質量のすべてを見つけられたかどうかではない。

36.8.2.4 ロジスティック事後分布に対する提案分布

ロジスティック回帰の事後分布に対しては、提案分布をどう選べばよいだろうか。20,000 サンプルのガウス事前分布と 5,000 サンプルのラプラス近似という 2 つのガウス分布を、サンプル数に対する ESS の割合、最大の正規化重み、およびグリッド基準値に対する推定事後平均の誤差（事後標準偏差を単位として測定）を記録することで比較してみよう。

```
def gaussian_logpdf(samples, mean, cov):
    diff = samples - mean
    _, logdet = np.linalg.slogdet(cov)
    maha = np.einsum('ni,ij,nj->n', diff, np.linalg.inv(cov), diff)
    return -0.5 * (len(mean) * np.log(2 * np.pi) + logdet + maha)

def importance(log_target, mean, cov, N):
    """Self-normalized importance sampling from a Gaussian proposal."""
    samples = rng.multivariate_normal(mean, cov, N)
    log_w = log_target(samples) - gaussian_logpdf(samples, mean, cov)
    w = np.exp(log_w - log_w.max())
    w /= w.sum()
    return w @ samples, 1 / (w @ w), w.max()
```

```
for name, mean, cov, N in [('prior', np.zeros(2), prior_var * np.eye(2), 20000),
                           ('Laplace', theta_map, laplace_cov, 5000)]:
    estimate, ess, w_max = importance(log_joint, mean, cov, N)
    error = np.abs(estimate - post_mean) / post_sd
    print(f'{name:8s}: ESS {ess / N:.1%} of {N}, largest weight '
          f'{w_max:.4f}, error in sds {error.round(3)}')
```

```
prior   : ESS 4.2% of 20000, largest weight 0.0025, error in sds [0.013 0.001]
Laplace : ESS 85.5% of 5000, largest weight 0.0030, error in sds [0.006 0.003]
```

事前分布からのサンプルの大半は尤度がほぼゼロの領域に落ちるため、事前分布を提案分布に使うと ESS は全サンプルの約 4% にしかならない。ラプラスガウス分布によって提案分布の照準を事後分布に合わせると、ESS は劇的に向上し、まれにずっと低い値に落ちる実行があるものの、通常はサンプルの 80% から 90% が有効となる。どちらの提案分布でも、事後平均の推定誤差は通常グリッド基準値から事後標準偏差の数パーセント以内に収まり、0.15 を超える誤差が出ることは滅多にない。

ところがラプラス提案分布には、通常の ESS の出力結果からは決して見えてこない巧妙な欠陥が潜んでいる。原点から遠く離れた領域では、事後分布の裾は事前分布に支配される。任意の半直線 $\theta = t\mathbf{v}$ に沿って、対数尤度は t に関して高々線形にしか減衰しないのに対し（演習問題 8）、対数事前密度は $-t^2/(2\tau^2)$ として二次関数的に減衰するからである。その結果、ガウス提案分布 $\mathcal{N}(\mathbf{m}, \mathbf{S})$ に対して、積分 $\mathbb{E}_q[w^2] = \int \tilde{p}^2/q$ は $\mathbf{S}$ の固有値が 1 つでも $\tau^2/2 = 2$ を下回ると発散する。今の例ではラプラス共分散の 2 つの固有値（およそ 1.17 と 0.05）がいずれも 2 を大きく下回っているため、重要度重みは 1 本の細い軸方向だけでなく**あらゆる**方向で無限大の分散を持つ。

では、なぜ 5,000 サンプルの実行結果はあれほど健全に見えたのだろうか。重みが爆発する領域はラプラス標準偏差の何倍も離れた場所にあるため、5,000 サンプルの実行の大半はたまたまそこに足を踏み入れないからである。しかし 100 回中数回の実行では、遠くに落ちた 1 つのサンプルが総重みの巨大な割合を独占し、ESS が急落する。1 回実行しただけではこの不安定性は見抜けず、繰り返し実行するか裾の解析を行って初めて明らかになる。重みの分散を有限に戻す簡単な対策は 2 つある。ガウス分布をより裾の重いスチューデント t 提案分布に置き換えること（演習問題 8）、あるいは $q = 0.9\,\mathcal{N}(\theta_{\text{MAP}}, \mathbf{H}^{-1}) + 0.1\,\text{prior}$ のような**防御的混合分布**（defensive mixture）を使うことである。この混合密度は常に事前密度の少なくとも 10 分の 1 以上あるため、すべての重みは尤度の 10 倍で上から抑えられ、しかも尤度は決して 1 を超えない。

重要度サンプリングは未知のエビデンスを自動的に相殺してくれるが、それでも事後分布全体を覆う単一の大域的な提案分布 q を必要とする。高次元空間では、そのような役割を果たせる単純な大域分布は存在しない。

36.8.3 マルコフ連鎖モンテカルロ法

事後分布全体を覆う単一の大域的な提案分布を探し求める代わりに、局所的な移動を通じてパラメータ空間を探索し、各領域の事後確率質量に比例した時間をそこで過ごすランダムウォークを設計することができる。このようなランダムウォークは**マルコフ連鎖**（Markov chain）であり、次の状態は**遷移規則**（transition rule）$T(\theta' \mid \theta)$（θ から θ' へ移動する確率）を通じて現在の状態のみに依存する。現在の状態を分布 π から引いたときに次の状態も同じく π に従うなら、π はその連鎖に関して**定常**（stationary）であるといい、次式で表される。

$$\sum_{\theta} \pi(\theta)\, T(\theta' \mid \theta) = \pi(\theta') \quad \text{for every } \theta', \tag{36.105}$$

連続パラメータの場合、この和は積分に置き換わる。ここでの目標は、事後分布を定常分布（stationary distribution）として持つ単純な遷移規則を構成することである。

36.8.3.1 メトロポリス規則

メトロポリス法（Metropolis algorithm）は、任意の対称な提案分布 $g(\theta' \mid \theta) = g(\theta \mid \theta')$（たとえば局所的なガウスステップ $\theta' = \theta + \mathbf{L}\epsilon$）からそのような連鎖を構成する。現在の状態 θ から候補 $\theta' \sim g(\cdot \mid \theta)$ を提案する。候補の事後密度が現在以上、すなわち $\tilde{p}(\theta') \geq \tilde{p}(\theta)$ であれば、必ず θ' へ移動する。下り方向への移動となる場合は、確率 $\tilde{p}(\theta')/\tilde{p}(\theta)$ でこれを受理し、そうでなければ θ にとどまる。決定的に重要なのは、提案が棄却されたときには現在の状態を繰り返し、その繰り返しも連鎖における次の状態として数えることである（Figure 36.39）。

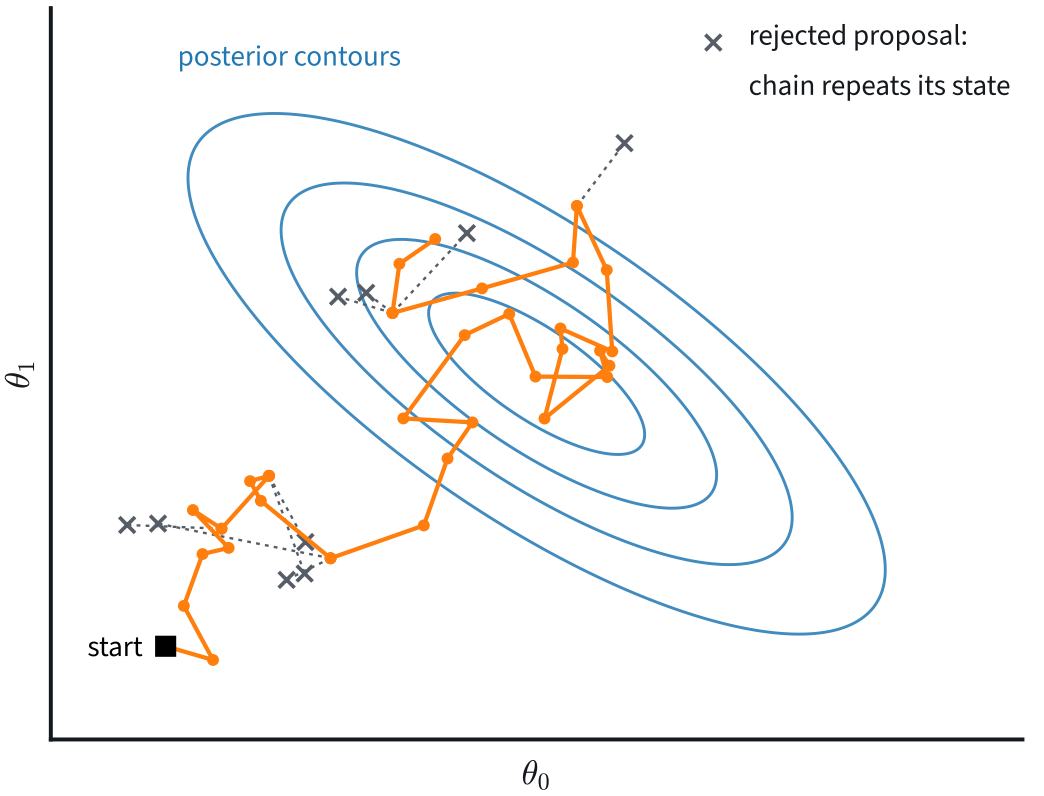


Figure 36.39: 事後分布の等高線上におけるメトロポリスランダムウォーク。開始点の四角形から、各ステップで局所的な移動を提案する。上り方向の移動は常に受理され（折れ線）、下り方向の移動は確率的に受理され、棄却された提案（罰印）では連鎖がもう 1 ステップその場にとどまる。長期的には、任意の領域で過ごす時間の割合がその事後確率質量に一致する。

Hastings は、提案分布の方向的な偏り（バイアス）を補正する因子 $g(\theta \mid \theta')/g(\theta' \mid \theta)$ を密度比（density ratio）に掛けることで、この規則を非対称な提案分布へと一般化した (Hastings 1970)。これにより、候補は次の確率で受理される。

$$\alpha(\theta, \theta') = \min\left(1, \frac{\tilde{p}(\theta')\, g(\theta \mid \theta')}{\tilde{p}(\theta)\, g(\theta' \mid \theta)}\right). \tag{36.106}$$

提案分布が滅多に訪れない状態への移動ほど受理されやすくなり、g が対称なときには補正因子は 1 に等しくなる。ここでもやはり、未知のエビデンスは $\tilde{p}$ の値の比の中で相殺される。

なぜ棄却を数に入れることが不可欠なのかは、最小のマルコフ連鎖を見ればよくわかる。目標分布 $\pi = (\frac{1}{4}, \frac{3}{4})$ を持つ 2 つの状態と、常に**もう一方**の状態を提案する対称な提案分布を考えよう。状態 1 からは目標密度の比が 3 なので、連鎖は必ず状態 2 へジャンプする。状態 2 からは比が $\frac{1}{3}$ なので、状態 1 へ確率 $\frac{1}{3}$ でジャンプし、状態 2 に確率 $\frac{2}{3}$ でとどまる。行を現在の状態として遷移行列を書き下せば、定常性を直接確かめられる。

$$\mathbf{T} = \begin{pmatrix} 0 & 1 \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}, \qquad \left(\tfrac{1}{4},\, \tfrac{3}{4}\right)\mathbf{T} = \left(\tfrac{3}{4}\cdot\tfrac{1}{3},\; \tfrac{1}{4} + \tfrac{3}{4}\cdot\tfrac{2}{3}\right) = \left(\tfrac{1}{4},\, \tfrac{3}{4}\right).$$

もし棄却を捨てて受理された移動だけを記録したなら、系列は単に $2, 1, 2, 1$ を交互に繰り返すだけとなり、$(\frac{1}{2}, \frac{1}{2})$ となって本来の $(\frac{1}{4}, \frac{3}{4})$ は得られない。棄却後に状態 2 を繰り返し訪問することこそが、状態 2 に本来の重みを与えているのである（Figure 36.40）。

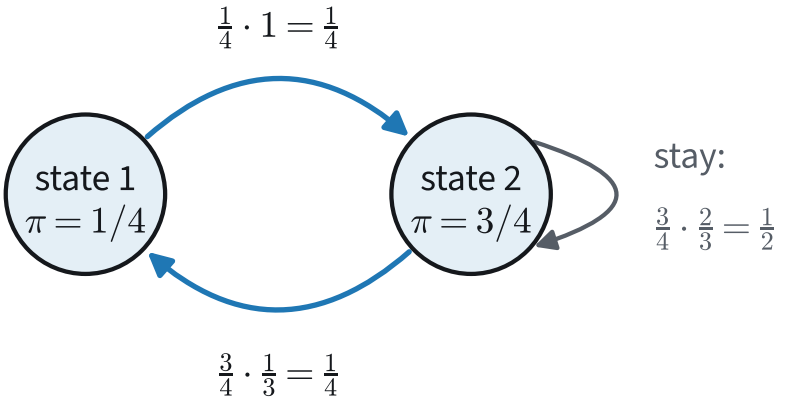


Figure 36.40: 目標分布 $\pi = (1/4, 3/4)$ を持つ 2 状態メトロポリス連鎖。状態 1 から状態 2 への確率フローは $\frac{1}{4}\cdot 1$ であり、逆方向のフローは $\frac{3}{4}\cdot\frac{1}{3}$ である。両者はともに $1/4$ に等しいため、滞在比率 $(1/4, 3/4)$ が毎ステップ保たれる。自己ループは状態 2 からの棄却された移動を表しており、受理された移動のみを記録すると $(1/2, 1/2)$ になってしまう。

36.8.3.2 詳細釣り合いと定常性

Figure 36.40 は、メトロポリス・ヘイスティングス法の背後にある中核的な仕組みを捉えている。1 ステップのうちに状態 1 から状態 2 へ流れる確率質量と逆向きに流れる確率質量が等しいため、どちらの状態も確率を得たり失ったりしない。このペアごとの釣り合いがあらゆる状態の組に対して成り立つとき、目標分布が定常分布であることが保証される。本節から理論的な結果を 1 つだけ覚えて帰るとすれば、この結果にせよ。

命題。遷移規則が次の**詳細釣り合い**（detailed balance）を満たすとする。

$$\pi(\theta)\,T(\theta' \mid \theta) = \pi(\theta')\,T(\theta \mid \theta') \quad \text{for all } \theta, \theta', \tag{36.107}$$

このとき、π はその連鎖の定常分布である。さらに、メトロポリス・ヘイスティングス連鎖は $\pi \propto \tilde{p}$ に対して詳細釣り合いを満たす。

証明。@eq-mdl-bayes-detailed-balance の両辺を θ にわたって足し合わせると Equation 36.105 の左辺が得られ、一方右辺は $\pi(\theta')\sum_{\theta} T(\theta \mid \theta') = \pi(\theta')$ へと簡約される。なぜなら、θ' からの遷移確率の和は 1 だからである。後半の主張について、$\theta' = \theta$ のときは詳細釣り合いが自動的に成り立つ。$\theta' \neq \theta$ のときは、移動が起きるのは θ' が提案されかつ受理されるとき、かつそのときに限り、次式が成り立つ。

$$\begin{aligned} \pi(\theta)\,T(\theta' \mid \theta) &= \pi(\theta)\,g(\theta' \mid \theta)\,\alpha(\theta, \theta') \\ &= \min\bigl\{\pi(\theta)\,g(\theta' \mid \theta),\ \pi(\theta')\,g(\theta \mid \theta')\bigr\}, \end{aligned}$$

この式は明らかに θ と θ' に関して対称である。π の正規化定数は α の中で約分されて消えるため、$\tilde{p}$ を π の代わりに代入してよい。$\blacksquare$

鍵となる代数的なステップは πg を最小値関数の内側へ引き入れることであり、これによって対称性が一目でわかるようになる。連続な状態空間では、棄却された移動が作る点質量に対してもう少し丁寧な扱いが必要になる（演習問題 1）。

定常性だけでは、任意の初期点から出発した連鎖が π に収束することまでは保証されない。連鎖はさらに、正の事後確率質量を持つあらゆる領域に到達可能でなければならず、領域間を決定論的に巡回してもならない。これら 2 つの緩やかな条件の下で、連鎖が任意の領域で過ごす時間の長期的な割合はその領域の事後確率（posterior probability）に収束する。

メトロポリス法の実装はわずか数行で済む。以下では、ラプラス共分散のコレスキー因子（Cholesky factor）$\mathbf{L}\mathbf{L}^{\top} = \mathbf{H}^{-1}$ を用いてガウス提案分布の形状を整える。

```
def metropolis(log_target, start, proposal_chol, n_steps):
    current = np.array(start, dtype=float)
    current_logp = log_target(current)
    states, accepted = np.empty((n_steps, len(current))), 0
    for t in range(n_steps):
        proposal = current + proposal_chol @ rng.standard_normal(len(current))
        proposal_logp = log_target(proposal)
        if np.log(rng.random()) < proposal_logp - current_logp:   # accept
            current, current_logp = proposal, proposal_logp
            accepted += 1
        states[t] = current            # a rejection repeats the state
    return states, accepted / n_steps
```

ラプラスガウス分布の 2 倍の広がりで散らした過分散（over-dispersed）な初期点 $\theta_{\mathrm{MAP}} + 2\mathbf{L}\epsilon$ から、それぞれ 12,000 ステップからなる 4 本の独立な連鎖を走らせ、大きさ $1.1\,\mathbf{L}\epsilon$ のステップを提案する。初期の状態にはまだ各連鎖がどこから出発したかの影響が残っているため、平均を計算する前に各連鎖の最初の 2,000 回の反復、すなわち**ウォームアップ**期間を捨てる。

```
def run_chains(scale, n_chains=4, n_steps=12000, warmup=2000):
    L = np.linalg.cholesky(laplace_cov)
    chains, rates = [], []
    for _ in range(n_chains):
        start = theta_map + 2 * L @ rng.standard_normal(2)  # over-dispersed
        states, rate = metropolis(log_joint, start, scale * L, n_steps)
        chains.append(states[warmup:])
        rates.append(rate)
    return np.array(chains), np.array(rates)

chains, rates = run_chains(1.1)
print('acceptance rates:', rates.round(3))
```

```
acceptance rates: [0.523 0.529 0.533 0.526]
```

どの連鎖でも全提案のおよそ半分が受理される。定常分布を決めるのはガウス提案分布ではなくメトロポリス受理規則であることを思い出そう。したがって、提案分布の形状が悪くても連鎖の進行が遅くなるだけで、統計的バイアスが持ち込まれることはない。

36.8.3.3 マルコフ連鎖の混合に関する診断

独立標本（independent samples）とは異なり、マルコフ連鎖における連続する状態には相関があり、どの連鎖も初期化した場所から出発する。連鎖から計算した平均を信頼する前に、3 つの診断上の問いを立て、それぞれに具体的な指標で答えることにしよう。

各連鎖は初期点を忘れ、同じ分布に収束したか。視覚的には各連鎖のトレースプロットを描けば停滞やドリフトが露わになり、$\widehat{R}$ は連鎖間の不一致を定量化する。それぞれ N 個の状態からなる複数の連鎖が与えられたとき、W を連鎖内分散の平均とし、B を連鎖平均の分散の N 倍とする。すると次式が成り立つ。

$$\widehat{R}^2 = \frac{\frac{N-1}{N} W + \frac{1}{N} B}{W}. \tag{36.108}$$

分子は全体の事後分散（posterior variance）を推定するものであり、連鎖同士が離れたままでいる間は過大評価する傾向がある。一方、各連鎖が事後分布の一部しか探索していない間は、W は事後分散を過小評価する。したがって $\widehat{R}$ は 2 つの分散推定量を比較しており、連鎖同士が混ざり合うにつれて 1 に近づく。実務では、まず各連鎖を前半と後半の 2 つに分割してから**分割** $\widehat{R}$ を計算する。こうすれば、単一の連鎖がドリフトしている場合でも、互いに食い違う 2 本の連鎖として検出される。出力を信頼するには 0.01 以内の差で 1 の近くに収まっていることが必要条件であるが、それだけで十分条件になることは決してない。すべての連鎖が同じ遠くのモードを見逃していれば、連鎖同士は何事もないかのように一致してしまうからである。

相関を持つこれらの状態は、実際には独立サンプル何個分に相当するのか。ρ_k を k ステップ離れた状態間の自己相関（autocorrelation）とする。積分自己相関時間（integrated autocorrelation time）$\kappa = 1 + 2\sum_{k \geq 1} \rho_k$ は、連鎖の何ステップ分が 1 個の独立サンプルと同じ情報量を持つかを数える量である。したがって、c 本の連鎖（各 N 状態）からは有効標本サイズ $N_{\text{eff}} = cN/\kappa$ が得られ、これは重要度サンプリングにおけるものと同じ意味と記号を持つ。

系列相関の最も単純なモデルは 1 次自己回帰過程（first-order autoregressive process）AR(1) であり、$x_t = \phi x_{t-1} + \varepsilon_t$（$\varepsilon_t$ は独立なノイズで $|\phi| < 1$）で与えられる。各ステップで連鎖は直前の値の割合 ϕ を保持する。ラグ k の自己相関は $\rho_k = \phi^k$ であり、これから次式が得られる。

$$\kappa = 1 + 2\sum_{k \geq 1} \phi^k = \frac{1+\phi}{1-\phi}.$$

$\phi = 0.9$ のとき $\kappa = 19$ となり、10,000 個の状態は約 530 個の独立サンプルにしかならない。$\phi = 0.99$ では $\kappa = 199$ となり、わずか 50 個程度の独立サンプルしか残らない。

非常に大きなラグにおける推定自己相関はノイズに支配されるため、ここでの実装では連続するペア $\rho_{2k} + \rho_{2k+1}$ ごとにまとめ、最初に現れる負のペアで和を打ち切る。実用レベルの推論パッケージでは両方の診断指標がさらに改良されており、ランク正規化された $\widehat{R}$ や、分布の中心部（バルク）と裾で別々に計算した有効標本サイズが用いられている。

推定された事後平均はどれほど正確か。マルコフ連鎖の中心極限定理により、推定された事後平均の**モンテカルロ標準誤差**（Monte Carlo standard error、MCSE）は、事後標準偏差を $\sqrt{N_{\text{eff}}}$ で割ったものになる。

```
def split_rhat(chains):
    """Split R-hat per coordinate; chains has shape (c, N, d)."""
    half = chains.shape[1] // 2
    halves = np.concatenate([chains[:, :half], chains[:, half:2 * half]])
    W = halves.var(axis=1, ddof=1).mean(axis=0)
    B = half * halves.mean(axis=1).var(axis=0, ddof=1)
    return np.sqrt(((half - 1) / half * W + B / half) / W)

def autocorrelation(x):
    """rho_k of one sequence for k = 0, ..., N - 1, by FFT."""
    x = x - x.mean()
    f = np.fft.rfft(x, n=2 * len(x))
    acov = np.fft.irfft(f * np.conj(f))[:len(x)]
    return acov / acov[0]

def chain_ess(chains):
    """c N / kappa per coordinate, summing rho in pairs (Geyer)."""
    c, N, d = chains.shape
    ess = np.empty(d)
    for j in range(d):
        rho = np.mean([autocorrelation(chain[:, j]) for chain in chains],
                      axis=0)
        pairs = rho[:N - 1:2] + rho[1:N:2]        # rho_2k + rho_2k+1
        negative = np.flatnonzero(pairs < 0)
        n_pairs = negative[0] if len(negative) else len(pairs)
        ess[j] = c * N / (-1 + 2 * pairs[:n_pairs].sum())
    return ess
```

これらの診断指標を実際に試してみよう。もしエラーバーが適切に較正されていれば、4 本の連鎖から得られた MCMC 平均は、厳密なグリッド平均からおよそ 1 MCSE 以内に収まるはずである。あわせて、測定された傾きの自己相関時間 κ が、ラグ 1 の自己相関 ρ_1 で評価した AR(1) の公式とどれほどよく一致するかも確認する。

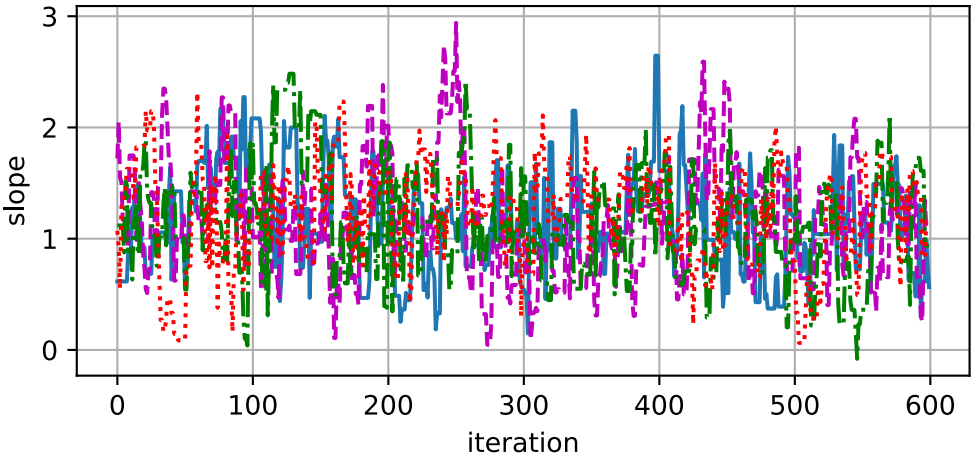
```
flat = chains.reshape(-1, 2)
ess = chain_ess(chains)
mcmc_mean, mcmc_sd = flat.mean(axis=0), flat.std(axis=0, ddof=1)
```

```
mcse = mcmc_sd / np.sqrt(ess)
rho1 = np.mean([autocorrelation(chain[:, 1])[1] for chain in chains])
print('split R-hat       :', split_rhat(chains).round(3))
print('ESS               :', ess.round().astype(int), 'of', len(flat))
print('(MCMC - grid)/MCSE:', ((mcmc_mean - post_mean) / mcse).round(1))
print(f'slope: rho_1 {rho1:.2f}, AR(1) kappa {(1 + rho1) / (1 - rho1):.1f},'
      f' measured kappa {len(flat) / ess[1]:.1f}')
```

```
split R-hat       : [1.001 1.001]
ESS               : [3798 3697] of 40000
(MCMC - grid)/MCSE: [ 0.6 -0.4]
slope: rho_1 0.81, AR(1) kappa 9.4, measured kappa 10.8
```

最後に、ウォームアップ後の最初の 600 反復における傾きのトレースプロットを描いて診断チェックを締めくくる。

```
d2l.plot(np.arange(600), [chain[:600, 1] for chain in chains],
         xlabel='iteration', ylabel='slope', figsize=(6, 2.5))
```



診断結果は良好である。分割 $\widehat{R}$ は 0.01 以内の差で 1 の近くに収まり、4 本のトレースプロットは密に絡み合っている。ESS は保持された 40,000 個の状態のおよそ 10 分の 1 であり、$\kappa \approx 10$ となってラグ 1 の自己相関 $\rho_1 \approx 0.8$ から得られる AR(1) の予測とぴったり整合する。最も心強いことに、MCMC 平均と厳密なグリッド平均の差はおよそ 1 MCSE であり、不確実性の見積もりが実際の推定誤差と確かに一致していることが確認できる。

36.8.3.4 提案分布のスケールと混合の失敗

メトロポリス法の実行から出力するのが最も簡単な数値は受理率であるが、それ単体からわかることはほとんどない。提案スケールを 0.05 および 10 に設定して（1.1 の代わりに）4 本の連鎖を再実行してみよう。数値を見る前に少し立ち止まり、この両極端のうちどちらの方が有効サンプル数が少なくなるか予想してみてほしい。

```
for scale in [0.05, 10]:
    bad_chains, bad_rates = run_chains(scale)
    print(f'scale {scale:5}: acceptance {bad_rates.mean():.3f}, max R-hat '
          f'{split_rhat(bad_chains).max():.2f}, '
          f'min ESS {chain_ess(bad_chains).min():.0f}')
```

```
scale  0.05: acceptance 0.975, max R-hat 1.13, min ESS 43
scale    10: acceptance 0.020, max R-hat 1.00, min ESS 689
```

微小なステップ（スケール 0.05）では提案の約 97% が受理されるにもかかわらず、1 ステップあたりほとんど動かないため、$\widehat{R}$ は 1.01 を大きく超え、40,000 回の反復からわずか数十個の有効サンプルしか得られない。巨大なステップ（スケール 10）では提案の約 2% しか受理されず、連鎖は長い区間にわたって同じ状態を繰り返して停滞するものの、それでも数百個の有効サンプルは確保できている。最適なステップサイズはこれら両極端の間の程よい位置にある（演習問題 2）。より本質的には、闇雲なランダムウォークが上り方向へ進むのは偶然頼みであるのに対し、$\log \tilde{p}$ の勾配に導かれる提案分布は設計上はじめから上り方向へドリフトする。このような勾配情報を活用した移動はランジュバン動力学（Langevin dynamics）（38.6.2 節）から生じ、そのメトロポリス調整版は演習問題 9 で展開する。

36.8.4 変分推論

サンプリング手法は事後期待値をその都度新たに推定し、常にモンテカルロノイズを伴う。**変分推論**（variational inference）は根本的に異なる道筋をとり、積分を最適化へと変換する。扱いやすい分布族 q_ψ（変分パラメータ ψ で添字付けられる）を選び、ψ を最適化して q_ψ が事後分布をできるだけ忠実に近似するようにしたうえで、q_ψ の下での期待値を閉じた形または高速な独立サンプリングによって計算するのである。

36.8.4.1 エビデンス下界

q_ψ を最適化するには、未知の正規化定数 $p(\mathcal{D})$ を必要とせずに事後分布への近さを測る目的関数が要る。幸い、そのような目的関数はすでに 36.7 節で導出している。そこでは潜在変数上の近似分布 q が対数エビデンスの下界を与え、その差（ギャップ）が KL ダイバージェンスとなっていた。パラメータベクトル θ を潜在変数、データセット $\mathcal{D}$ を観測とみなせば、まったく同じ恒等式が得られる。

$$\log p(\mathcal{D}) = \underbrace{\mathbb{E}_{q_\psi}\big[\log p(\mathcal{D},\theta) - \log q_\psi(\theta)\big]}_{\mathcal{L}(\psi)} + D_{\mathrm{KL}}\big(q_\psi \,\|\, p(\cdot \mid \mathcal{D})\big). \tag{36.109}$$

KL ダイバージェンスは常に非負であるため、$\mathcal{L}(\psi)$ は q_ψ の**エビデンス下界**（evidence lower bound、ELBO）となる。

$$\mathcal{L}(\psi) = \mathbb{E}_{q_\psi}[\log p(\mathcal{D},\theta) - \log q_\psi(\theta)] \le \log p(\mathcal{D}). \tag{36.110}$$

Equation 36.109 の右辺にある 2 つの項の和は定数 $\log p(\mathcal{D})$ に固定されているため、$\mathcal{L}(\psi)$ が 1 単位増えるごとに $D_{\mathrm{KL}}(q_\psi \,\|\, p(\cdot \mid \mathcal{D}))$ はちょうど 1 単位減少する。したがって、ψ に関して ELBO を最大化することは KL ダイバージェンスを最小化することと同値であり、未知のエビデンスは加法的な定数として消える。期待値の中のすべての項は点ごとに評価でき、期待値自体も q_ψ からのサンプルを使って容易に推定できる。

36.8.4.2 KL ダイバージェンスの向き

Equation 36.109 における引数の順序に注目しよう。変分推論が最小化するのは $D_{\mathrm{KL}}(q\|p) = \int q \log(q/p)$、すなわち第 1 引数に q を置いた**逆方向** KL ダイバージェンス（reverse KL divergence）である。その被積分関数は $q > 0$ かつ $p \approx 0$ となる場所ではどこでも発散するため、逆方向 KL は事後分布が質量を持たない領域に q が質量を置くことに対して重いペナルティを課す。その結果、q は低密度領域から縮小して単一の最頻値（モード）に張り付く傾向がある。つまり、逆方向 KL は**モード探索型**（mode-seeking）である。対照的に、**順方向**ダイバージェンス（forward divergence）$D_{\mathrm{KL}}(p\|q) = \int p \log(p/q)$ は $p > 0$ かつ $q \approx 0$ となる領域にペナルティを課すため、q は事後分布の質量全体を覆うように引き伸ばされる。つまり、順方向 KL は**質量被覆型**（mass-covering）である。ガウス分布族の場合、順方向 KL の最小化は事後分布の厳密な平均と共分散に一致させることに帰着するが、その計算には扱いにくい事後分布 p の下での期待値が必要になるのに対し、逆方向 KL は q の下での期待値しか必要としない。Figure 36.41 は、二峰性の目標分布におけるこれら 2 つの振る舞いを対比している。

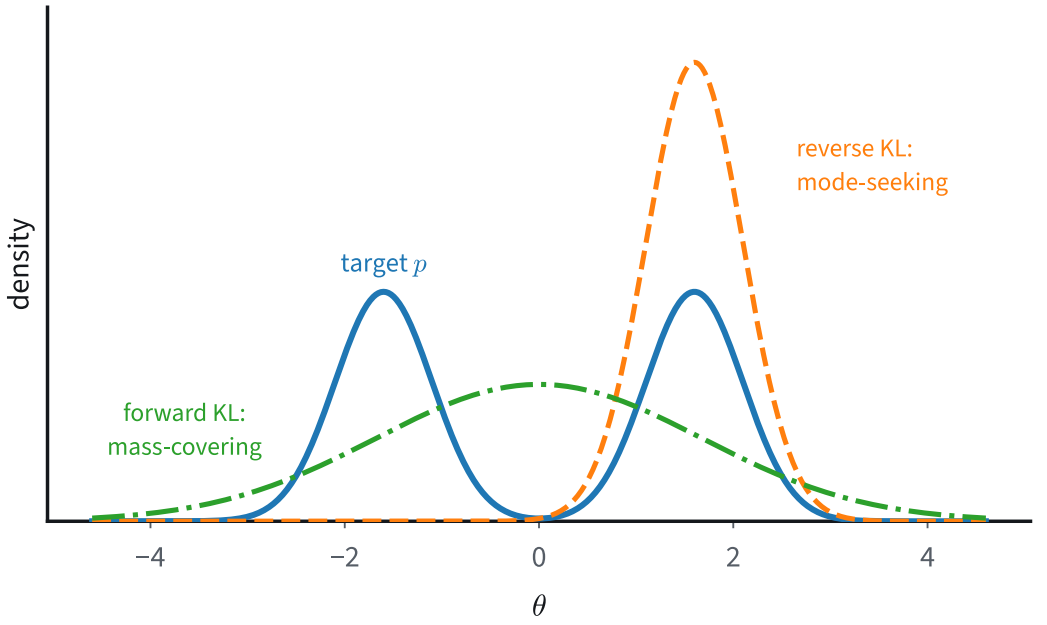


Figure 36.41: KL ダイバージェンスの向きによって、単純すぎる近似がどのように破綻するかが決まる。二峰性の目標分布に対し、変分推論が求める逆方向 KL のガウス分布は単一のモードに張り付いて不確実性を過小評価するのに対し、順方向 KL のガウス分布は両方のモードを覆い、その間にある低密度の谷間にピークを置く。

36.8.4.3 平均場ガウス分布の当てはめ

最も単純な変分分布族は、パラメータを互いに独立としてモデル化する**平均場**（mean-field）ガウス分布 $q = \mathcal{N}(\mathbf{m}, \mathrm{diag}(\mathbf{s}^2))$ である。$\mathbf{m}$ と $\log \mathbf{s}$ に関する ELBO の勾配を計算するため、サンプルを $\theta = \mathbf{m} + \mathbf{s} \odot \epsilon$（ただし $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$）と再パラメータ化する。ノイズの分布は変分パラメータに依存しないため、勾配を期待値の内側へ移動できる（34.7.3 節）。エントロピー $-\mathbb{E}_q[\log q]$ は加法的な定数を除いて $\sum_j \log s_j$ に等しいため、次の不偏な勾配公式が得られる。

$$\nabla_{\mathbf{m}}\mathcal{L} = \mathbb{E}[\nabla_\theta \log p(\mathcal{D},\theta)], \qquad \nabla_{\log \mathbf{s}}\mathcal{L} = \mathbb{E}[\nabla_\theta \log p(\mathcal{D},\theta)\odot \mathbf{s}\odot\epsilon] + \mathbf{1}, \tag{36.111}$$

ここで、$\mathbf{1}$ はエントロピー $\sum_j \log s_j$ を $\log \mathbf{s}$ で微分した項に由来する。第 1 の勾配は平均的に上り方向へ平均を動かし、第 2 の勾配はエントロピーの外向きに広げる力と事後分布の下向きの曲率が釣り合うまで q を膨張させる。ロジスティック事後分布に対して最適化を実行する前に、目標分布がガウス分布である場合を厳密に解くことで、何が起きるかを予測しておこう。

命題。 精度行列 (precision matrix) Λ を持つガウス目標分布 $p = \mathcal{N}(\mu, \Lambda^{-1})$ に対し、$D_{\mathrm{KL}}(q\|p)$ を最小化する平均場ガウス分布は、平均 μ と、目標分布の**条件付き**分散である分散 $s_j^2 = 1/\Lambda_{jj}$ を持つ。相関 ρ を持つ 2 次元の場合、各 s_j は真の周辺標準偏差の $\sqrt{1-\rho^2}$ 倍となる。

証明。 q に依存しない定数を除くと、$D_{\mathrm{KL}}(q\|p) = -\sum_j \log s_j + \frac{1}{2}\mathbb{E}_q[(\theta-\mu)^\top \Lambda(\theta-\mu)]$ となる。q の下では各座標が平均 m_j、分散 s_j^2 で独立であるため、ゆらぎの交差項は期待値をとるとすべて消え、$\sum_j \Lambda_{jj} s_j^2 + (\mathbf{m}-\mu)^\top\Lambda(\mathbf{m}-\mu)$ が残る。Λ は正定値であるから、$\mathbf{m}$ に関する二次形式は $\mathbf{m} = \mu$ で最小となる。s_j に関する導関数をゼロとおく式 $-1/s_j + \Lambda_{jj} s_j = 0$ より、$s_j^2 = 1/\Lambda_{jj}$ が得られる。$d = 2$ の場合、2×2 共分散行列の逆行列を求めると $1/\Lambda_{jj} = \sigma_j^2(1-\rho^2)$ となる (ここで σ_j は周辺標準偏差である)。■

対角ガウス分布は相関のある事後分布に合わせて主軸を回転させることができず、逆方向 KL の下では低密度の隅にはみ出すことに対して厳しいペナルティを支払う。そのため、事後分布の周辺全体の幅ではなく、事後分布を切断した条件付きスライスの幅まで収縮する (Figure 36.42)。この収縮は、分解された分布族と逆方向 KL を組み合わせたことによる固有の帰結である点に注意しよう。完全共分散ガウス分布をまったく同じ ELBO 目的関数でガウス目標分布に当てはめれば、誤差なく元の分布が復元される。ここでのロジスティック回帰の例では $\rho = -0.86$ であるから、上の命題より平均場の標準偏差は真の周辺標準偏差のおよそ $\sqrt{1-0.86^2} = 0.51$ 倍、すなわち約半分の幅になると予測される。

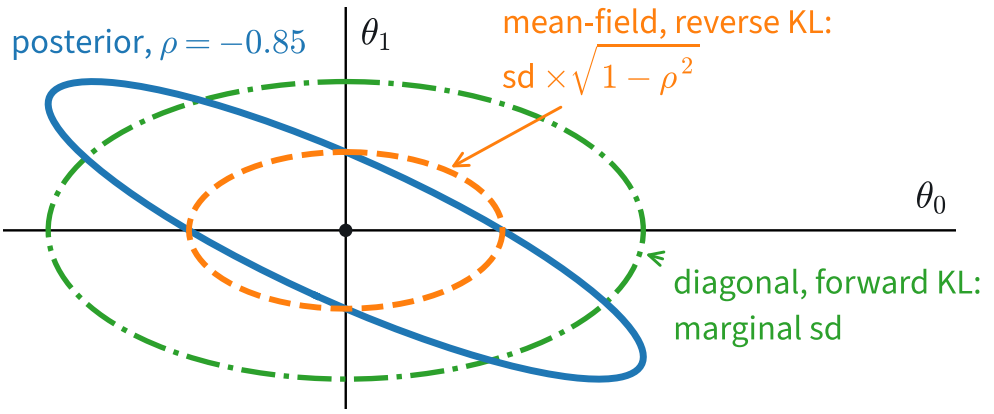


Figure 36.42: 標準偏差 $(1, 0.5)$、相関 $\rho = -0.85$ を持つ相関付きガウス事後分布に当てはめた対角ガウス分布 (2 標準偏差の楕円として表示)。分解された分布族は傾くことができないため、逆方向 KL (オレンジ) は周辺標準偏差の $\sqrt{1-\rho^2} \approx 0.53$ 倍である条件付きの幅まで収縮するのに対し、順方向 KL (緑) は低密度領域へはみ出しながらも周辺の幅を保つ。

この予測を数値的に確かめてみよう。ラプラス近似を初期値とし、1 ステップあたり 256 個の再パラメータ化されたサンプルと $1/\sqrt{t}$ で減衰する学習率を用いて、Equation 36.111 による確率

的勾配上昇法を 3,000 反復実行する。最後の 1,000 反復を平均して最適化ノイズを平滑化することで、当てはめられた標準偏差を、グリッドから求めた相関 ρ を使った命題の予測値と直接比較できる。

```
m_vi, log_s_vi = theta_map.copy(), np.log(laplace_sd)
m_avg, log_s_avg = np.zeros(2), np.zeros(2)
n_steps, n_avg = 3000, 1000
for t in range(1, n_steps + 1):
    eps = rng.standard_normal((256, 2))
    s_vi = np.exp(log_s_vi)
    g = grad_log_joint(m_vi + s_vi * eps)        # at theta = m + s * eps
    lr = 0.05 / np.sqrt(t)
    m_vi += lr * g.mean(axis=0)
    log_s_vi += lr * ((g * eps * s_vi).mean(axis=0) + 1)
    if t > n_steps - n_avg:                      # average the last iterates
        m_avg += m_vi / n_avg
        log_s_avg += log_s_vi / n_avg
s_avg = np.exp(log_s_avg)
print('             mean             sd')
print('grid      ', post_mean.round(3), post_sd.round(3))
print('Laplace   ', theta_map.round(3), laplace_sd.round(3))
print('mean-field', m_avg.round(3), s_avg.round(3))
print('predicted sd, grid sd * sqrt(1 - rho^2):',
      (post_sd * np.sqrt(1 - post_rho**2)).round(3))
```

```
             mean             sd
grid       [-2.347  1.237] [1.042 0.48 ]
Laplace    [-2.138  1.117] [1.004 0.457]
mean-field [-2.276  1.218] [0.533 0.244]
predicted sd, grid sd * sqrt(1 - rho^2): [0.536 0.247]
```

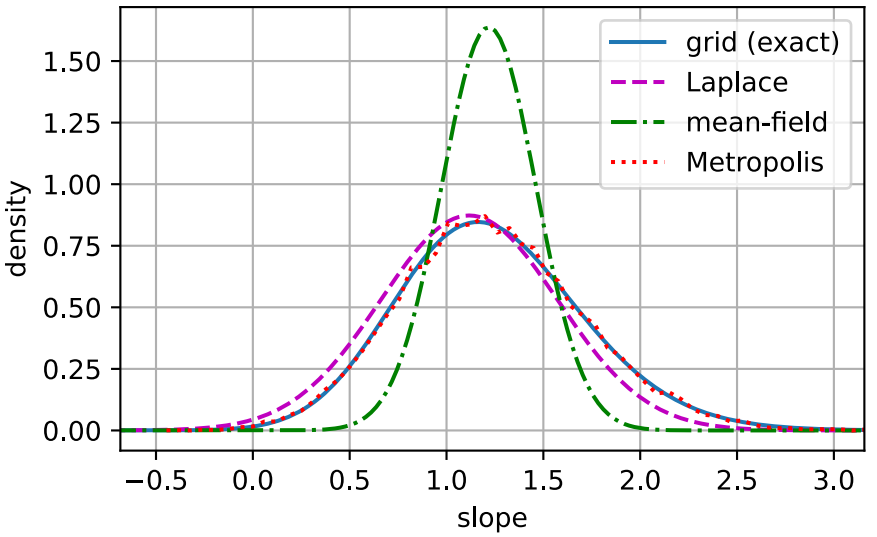
平均場近似の平均はグリッド平均から事後標準偏差のおよそ 10 分の 1 以内に収まる一方、標準偏差は真の幅の約半分まで縮み、解析的な予測値（0.54 と 0.25）と 2% 以内で一致する。最適化はきれいに収束しているため、残る食い違いはすべて制約の強い変分分布族に起因するものである。完全共分散ガウス分布、混合分布、正規化フロー（normalizing flow）といったより表現力の高い分布族を用いれば、より高い計算コストと引き換えにこのギャップを埋められる。ELBO の収束が確認してくれるのはあくまで**選んだ分布族の内部で**最適化が完了したことだけであり、それ自体はその分布族が実際に事後分布に適合していることを決して保証しない点を心に留めておこう。

36.8.5 各手法の比較

これで、まったく同じ事後分布に対する 4 つの異なる近似が出揃った。厳密なグリッド曲線、ラプラス近似および平均場近似のガウス分布、そしてメトロポリス法によるサンプルのヒストグラムを、傾き θ_1 の周辺密度の上で 1 つの図に並べて比較しよう。

```
def gauss_pdf(t, mean, sd):
    return np.exp(-0.5 * ((t - mean) / sd)**2) / (sd * np.sqrt(2 * np.pi))

b1 = grid_axes[1]
exact = grid_weight.reshape(m, m).sum(axis=0) / (b1[1] - b1[0])
hist, edges = np.histogram(flat[:, 1], bins=60, density=True)
```

```
d2l.plot([b1, b1, b1, (edges[:-1] + edges[1:]) / 2],
         [exact, gauss_pdf(b1, theta_map[1], laplace_sd[1]),
          gauss_pdf(b1, m_avg[1], s_avg[1]), hist],
         xlabel='slope', ylabel='density', figsize=(5, 3),
         xlim=(post_mean[1] - 4 * post_sd[1], post_mean[1] + 4 * post_sd[1]),
         legend=['grid (exact)', 'Laplace', 'mean-field', 'Metropolis'])
```



メトロポリス法のヒストグラムは、歪度も含めて厳密な事後分布を忠実に捉えている。ラプラス近似の曲線は真の平均より左にある最頻値を中心としており、わずかに幅が狭すぎる。平均場ガウス分布は厳密な周辺分布のおよそ半分の幅しかなく、その構成上、真の相関 $\rho = -0.86$ に対してゼロの相関を報告する。

手法	表現	主な診断指標	典型的な破綻様式
グリッド	格子上の重み付き点	グリッド端における質量	計算コスト m^d（次元 d）
ラプラス近似	最頻値における単一のガウス分布	サンプルとの比較	歪み、裾、境界、他のモード
棄却サンプリング	独立な厳密サンプル	受理率	受理率が $n^{-d/2}$ で低下、上界 M が必要
重要度サンプリング	独立な重み付きサンプル	重みの ESS、最大重み、再実行	裾の軽い提案分布や位置のずれた提案分布
MCMC	相関のある事後サンプル	トレース、分割 $\widehat{R}$、ESS、MCSE	混合の遅さ、未訪問のモード
変分推論	分布族内の最適化された分布	サンプルとの比較	狭すぎる分布族、分解された q の過信

実務ではどの手法を手に取るべきだろうか。パラメータがほんの数個であれば、グリッドを使えばよい。滑らかで単峰な事後分布に対しては、ラプラス近似が高速な第 1 の答えと強力な初期提案分布を与えてくれる。期待値に信頼できるエラーバーが必要なときは、3 つの診断指標と組み合わせた MCMC に頼るのがよい。事後分布の近似をより大きな最適化ループの内部に組み込む必要がある場面では変分推論が威力を発揮するが、分解された分布族は不確実性を過小評価する

ことを念頭に置いておく必要がある。2 つのガウス近似はいずれもそれ単体で完結した精度検証手段を持たないため、可能な限りサンプラと突き合わせて検証すべきである。

スケールについては率直に述べておく価値がある。これら 4 つの手法はいずれも、大規模なニューラルネットワークの全パラメータの事後分布へはそのまま きれいにスケールしない。次元が増えるにつれて重要度重みは崩壊し、ランダムウォーク MCMC は独立サンプル 1 個あたりにますます多くのステップを要するようになり、完全なヘッセ行列は d^2 個の要素を持ち、平均場分布 q は先ほど導出した相関因子の分だけ不確実性を過小評価する。さらに悪いことに、ニューラルネットワークのアーキテクチャ上の対称性によって、隠れユニットのあらゆる置換と符号反転ごとに事後分布のすべてのモードが複製される（演習問題 7）。

大規模なネットワークにおいて、本節の冒頭で立てた問いに対して最も広く使われている答えは、4 つの手法のどれよりも単純である。異なるランダム初期化から同じネットワークを複数回訓練し、予測確率を平均するのである。このような**深層アンサンブル**（deep ensemble）(Lakshminarayanan et al. 2017) は、独立に訓練された少数のネットワークを事後分布からのサンプルの代用とする、事後予測 Equation 36.96 の大雑把なモンテカルロ推定（Monte Carlo estimate）である。各実行は異なるモードに落ち着くため、その平均は複数のモードにまたがることになり、これは単一のラプラス近似や平均場近似にはできない芸当である。典型的に訓練データから遠く離れた領域で各メンバーの予測が食い違うと、平均された予測の確信度は低くなる。ただし、各メンバーは事後分布からのサンプルではなく、等しい重みを付けるべきだという保証もどこにもない。また各メンバーの誤差には相関があるため、メンバーを増やす効果にも限界がある。ペアごとの相関が ρ のとき、平均の分散は単一メンバーの分散の ρ 倍より小さくなることは決してない（36.2 節の演習問題 2）。

それでもなお、本節の中核的なアイデアは、正規化定数を除いてしか分布がわからないあらゆる場面で深層学習の随所に再び現れる。エビデンス下界は、エンコーダネットワークが q を出力する変分オートエンコーダ（variational autoencoder、VAE）(Kingma and Welling 2014)（17.7 節）や離散拡散モデル（17.9 節）を訓練するのに使われる。重要度重みと有効標本サイズによって、方策勾配法は過去の軌跡を再利用できる（15.2 節）。そして勾配誘導型の提案から受理・棄却ステップを取り除いたものは、スコアベース生成モデルを支えるランジュバンサンプラとなる（17.1 節）。最後に、あらゆるサンプリング手法は有限個のサンプルにわたる経験平均を返す。次節では、サンプルが独立であるときに、そのような有限標本平均がその期待値の周りにどれほど密に集中するかを問う。

36.8.6 まとめ

モデルを単一のパラメータベクトルへと訓練することは、その点推定があたかも確実に既知であるかのように扱うことに等しい。訓練データから遠く離れた領域では、もっともらしいパラメータベクトル同士が何桁も食い違うことがあり、事後予測においてそれらを平均すると、稀な事象の確率が数倍変化しうる。その平均を計算するには、正規化定数を除いてしかわからない事後分布に関する積分が必要になる。本節で扱った 4 つの手法はすべて、未知の定数が相殺される比や差のみを扱うことでこの定数を回避している。すなわち、最頻値における局所的な曲率（ラプラス近似）、密度比 $\tilde{p}/q$（棄却サンプリングと重要度サンプリング）、受理比 $\tilde{p}(\theta')/\tilde{p}(\theta)$（メトロポリス・ヘイスティングス法）、そして逆方向 KL ダイバージェンスと定数しか違わない最適化目的関数（変分推論）である。サンプリング手法には組み込みの診断指標（重要度サンプリングにおける重みの ESS、MCMC における分割 $\widehat{R}$、自己相関時間 κ、および MCSE）が備わっている一方、ガウス近似はサンプルと突き合わせて検証しなければならない。大規模なニューラルネットワークに対する完全なベイズ推論（Bayesian inference）は依然として計算困難であるが、エビデンス下界、重要度重み、ランジュバン動力学は、未正規化分布を扱うたびに本書の各所で再び登場する。

36.8.7 演習問題

1. **メトロポリス・ヘイスティングス法の受理確率。**受理確率 Equation 36.106 は、任意の提案分布を事後分布を目標とするマルコフ連鎖へと変える。
   1. 連続な状態空間では、遷移カーネルは密度部分 $g(\theta' \mid \theta)\alpha(\theta, \theta')$ と、現在の状態 θ における質量 $r(\theta) = 1 - \int g(\theta' \mid \theta)\alpha(\theta, \theta')\, d\theta'$（棄却確率）の点質量（アトム）からなる。このアトムを考慮に入れたうえで、事後分布が定常分布であることを示せ。**ヒント：**1 ステップ後、アトムは $\pi(\theta')r(\theta')$ だけ点 θ' における密度に寄与する。
   2. 独立サンプラは、現在の状態によらず $\theta' \sim q$ を提案する。このサンプラの受理確率が、Equation 36.102 の重要度重み w を用いて $\min(1, w(\theta')/w(\theta))$ となることを示せ。典型的な提案の重みよりもはるかに大きな重みを持つ状態に到達したあと、この連鎖はどのように振る舞うか。
   3. Barker の規則は、対称な提案を確率 $\tilde{p}(\theta')/(\tilde{p}(\theta) + \tilde{p}(\theta'))$ で受理する。これが詳細釣り合いを満たすことを示せ。本文の 2 状態連鎖に対する遷移行列を書き下し、メトロポリス連鎖の方が両方の状態からより頻繁に移動することを示せ。2 状態連鎖では、状態の任意の関数の自己相関は $\rho_k = \lambda^k$（λ は遷移行列の第 2 固有値）となる。両方の規則について λ と $\kappa = (1 + \lambda)/(1 - \lambda)$ を計算せよ。同じステップ数に対してより小さいモンテカルロ誤差を与えるのはどちらか。**ヒント：**非対角成分は、Barker の規則では $3/4$ と $1/4$ となるのに対し、メトロポリス法では 1 と $1/3$ となる。
2. [code] **提案分布のスケール。**関数 `run_chains` は `laplace_cov` のコレスキー因子にスケール（本文中では 1.1）を掛けている。
   1. 各連鎖 12,000 ステップとウォームアップを保ったまま、スケールを $\{0.05, 0.2, 0.5, 1.1, 1.7, 2.4, 3, 10\}$ にわたって変化させよ。各値について、受理率、分割 $\widehat{R}$、および `chain_ess` の最小 ESS を 4 本の連鎖の対数同時密度評価回数 $48{,}000$ で割った値を記録せよ。評価 1 回あたりの ESS を受理率に対してプロットせよ。評価 1 回あたりの ESS がその最大値の 2 分の 1 以上にとどまるのは、どの受理率の範囲か。
   2. d 次元の標準ガウス目標分布と提案分布 $\mathcal{N}(\theta, (\ell^2/d)\mathbf{I})$ に対し、Roberts et al. (1997) は $d \to \infty$ において $\ell \approx 2.38$（このとき受理率は約 0.234）で効率が最大になることを示している。ロジスティック事後分布はラプラス近似に近いため、コレスキー因子によっておおよそ標準ガウス分布に変換される。この結果は $d = 2$ に対してどのスケールを示唆するか。スイープで得られた最良のスケールおよびその受理率とどのように比較されるか。
   3. 本文中の診断指標のうち、スケール 0.05 の実行に異常のフラグを立てるのはどれか。受理率はその中に含まれるか。その実行における事後平均は、それ自身の MCSE を単位としてグリッド平均からどれだけ離れているか。
3. **ラプラス近似とパラメータ化。**コインを投げて表が 9 回、裏が 4 回出たとする。一様事前分布の下で、表の確率 θ の事後分布は $\text{Beta}(10, 5)$ となり、平均は $2/3$、標準偏差は 0.118、中央 95% 区間は $[0.419, 0.872]$ である。
   1. パラメータ θ におけるラプラス近似 Equation 36.100 は、本文で計算したとおり平均 $9/13$、標準偏差 0.128 を持つ。その中央 95% 区間と、$[0, 1]$ の外側に割り当てられる確率を求めよ。
   2. 対数オッズ $\eta = \log(\theta/(1-\theta))$ では、事後密度は $\text{sigmoid}(\eta)^{10}(1 - \text{sigmoid}(\eta))^{5}$ に比例する（ヤコビ行列、1 次元ではヤコビアン $d\theta/d\eta = \theta(1-\theta)$ が表裏それぞれに 1 回分のカウントを加えるためである）。その最頻値が $\eta = \log 2$ であることを示し、そこでの対数密度の曲率を計算し、ラプラス近似の中央 95% 区間を θ に写し戻せ。
   3. 2 つの近似のうち、厳密な事後分布により近いのはどちらか。$\theta > 1/2$ のような事象の厳密な事後確率がパラメータ化に依存しないのに対し、ラプラス近似がパラメータ化に依存するのはなぜか。

4. [code] **穏健化された予測。**●ラプラス近似 Equation 36.100 の下では、テスト入力におけるロジット $a = \mathbf{z}_\star^\top \theta$（ただし $\mathbf{z}_\star = (1, x_\star)^\top$）は、平均 $\mu_a = \mathbf{z}_\star^\top \theta_{\text{MAP}}$、分散 $s_a^2 = \mathbf{z}_\star^\top \mathbf{H}^{-1} \mathbf{z}_\star$ のガウス分布に従う。プロビット近似 $\text{sigmoid}(a) \approx \Phi(\lambda a)$（ただし $\lambda^2 = \pi/8$ であり、Φ は標準正規分布の累積分布関数）を用いると、予測積分 Equation 36.96 は閉じた形に変換される (MacKay 1992)。
   1. $\mathbb{E}[\Phi(\lambda a)] = \Phi(\lambda \mu_a / \sqrt{1 + \lambda^2 s_a^2})$ が $a \sim \mathcal{N}(\mu_a, s_a^2)$ に対して成り立つことを示せ。**ヒント：**$\Phi(\lambda a)$ を $P(U \le \lambda a)$（U は独立な標準正規確率変数）と書け。これから、穏健化された予測 $\text{sigmoid}(\mu_a / \sqrt{1 + \pi s_a^2/8})$ が $\text{sigmoid}(\mu_a)$ と $1/2$ の間にあることを結論づけよ。
   2. `theta_map` と `laplace_cov` から $x_\star \in \{-3, 2, 7\}$ における穏健化された予測を計算し、実行結果として出力されるプラグイン予測および事後予測の値と比較せよ。
   3. 同じ入力において、プラグイン予測 $\text{sigmoid}(\mathbf{z}_\star^\top \theta_{\text{MAP}})$、事後平均における予測 $\text{sigmoid}(\mathbf{z}_\star^\top \mathbb{E}[\theta \mid \mathcal{D}])$、および事後予測 $\mathbb{E}[\text{sigmoid}(\mathbf{z}_\star^\top \theta) \mid \mathcal{D}]$ の 3 つの量を計算せよ。最頻値から平均への変化と、平均における予測から予測の平均への変化という 2 つのステップのうち、予測を $1/2$ から遠ざけるのはどちらで、逆にそこへ近づけるのはどちらか。$x_\star = 2$ では両ステップとも小さいのに、$x_\star = -3$ や $x_\star = 7$ ではそうならないのはなぜか。このことは、過信した予測を穏健化するためにラプラス近似を用いることに対してどのような示唆を与えるか。
5. [code] **重みの ESS の予測。**●本文の命題により、N_{eff}/N は $(\mathbb{E}_q[w])^2 / \mathbb{E}_q[w^2]$ へと、サンプル数 N が増えるにつれて近づく。
   1. 1 次元において正規化された目標分布を $\mathcal{N}(\mu, s_p^2)$、提案分布を $\mathcal{N}(\nu, s_q^2)$ とし、$r = s_q / s_p$ とおく。$s_q^2 > s_p^2/2$ のとき
   $$\frac{\mathbb{E}_q[w^2]}{(\mathbb{E}_q[w])^2} = \frac{r^2}{\sqrt{2r^2 - 1}} \exp\!\left(\frac{(\mu - \nu)^2}{2 s_q^2 - s_p^2} \right),$$
   が成り立ち、$s_q^2 \le s_p^2/2$ のときはこの比が無限大になることを示せ。どの r が $\mu = \nu$ のときにこの比を最小にするか。
   2. 相関を無視し、グリッドから求めた平均と標準偏差を持つ独立なガウス分布で事後分布を近似して、上の比が各座標の積に分解されるようにせよ。事前分布を提案分布とする 20,000 個のサンプルの ESS を予測し、出力された値と比較せよ。$\tau = 0.5$ と $\tau = 10$ の事前提案分布についても同様に行え。すなわち、モデルのコードセルで `prior_var` $= \tau^2$ と設定し、重要度サンプリングのセルまで再実行して予測値と実測値を比較せよ。公式の中のどの因子が狭い事前分布を不適切な提案分布にし、どの因子が広い事前分布を不適切な提案分布にするか。
   3. 同じ近似の下で、提案分布 $\mathcal{N}(\theta_{\text{MAP}}, 0.3\,\mathbf{I})$ の重みの分散が無限大になるのは実行結果のどの座標か。5,000 個のサンプルでこれを 10 回実行し、ESS、最大の正規化重み、および推定事後平均を記録せよ。推定値がもっともらしい値に見える 1 回の実行結果だけでは、提案分布が適切であることの証拠にならないのはなぜか。
   4. 事前分布からの棄却サンプリングでは、確率 $L(\theta) / \max L$ でサンプルを残す（ここで L は尤度、$\max L$ は最尤推定量における尤度の値であり、線形分離可能なデータでは上限である 1 とする）。元のデータと 10 倍の点数を持つデータで受理率を測定せよ。事後分布の体積から n に伴う減衰を説明し、同じ事前分布からの重要度サンプリングの ESS 比率とその受理率を比較せよ。
6. [code] **エビデンスの推定。**●正規化された提案分布を q とし、$w = p(\mathcal{D}, \theta)/q(\theta)$ とおき、独立なサンプル $\theta_s \sim q$ に対して $\hat{Z}_N = \frac{1}{N} \sum_{s=1}^{N} w(\theta_s)$ とする。
   1. $\hat{Z}_N$ が $p(\mathcal{D})$ の不偏推定量であること、および $\mathbb{E}[\log \hat{Z}_N] \le \log p(\mathcal{D})$ が成り立つことを示せ。また、$\mathbb{E}[\log \hat{Z}_1]$ が q の ELBO Equation 36.110 に等しく、$\mathbb{E}[\log \hat{Z}_N]$ が N に関して減少しないことを示せ (Burda et al. 2016)。

2. 関数 `log_joint` は 2 次元事前分布の正規化定数 $-\log(2\pi\tau^2)$ を省略している。これを加えたうえで、グリッド上の密度を足し合わせてグリッドの 1 マスの面積を掛ける方法、ラプラス提案分布からの 5,000 サンプルを用いた $\widehat{Z}_N$ による方法、およびラプラスエビデンス $\log \tilde{p}(\theta_{\mathrm{MAP}}) + \log 2\pi - \frac{1}{2}\log\det \mathbf{H}$ による方法の 3 通りで $\log p(\mathcal{D})$ を計算せよ。$p(\mathcal{D})$ のラプラス推定値は数パーセント低く、$\log p(\mathcal{D})$ で約 0.04 小さくなる。グリッド計算のセルで見られた事後分布のどの性質がこの符号を説明するか。
3. ラプラス近似と平均場近似のそれぞれから 10^5 個のサンプルを用いて ELBO を推定し、各推定値の MCSE を報告せよ。Equation 36.109 より、$\log p(\mathcal{D})$ と q の ELBO の差は $D_{\mathrm{KL}}(q \| p(\cdot \mid \mathcal{D}))$ である。このダイバージェンスによって 2 つのガウス分布を比較せよ。

7. [code] **2 つのモードを持つ事後分布。**●$y_i \sim \mathcal{N}(\theta^2, 1)$（真の値は $\theta = 1$）に従う 50 個の観測を生成し、θ には事前分布 $\mathcal{N}(0, 4)$ を置く。この尤度は θ に θ^2 を通じてのみ依存する（ちょうど tanh 隠れユニットの入力重み・バイアス・出力重みの符号を同時に反転させてもニューラルネットワークの損失が変わらないのと同様である）。したがって、事後分布は $\theta \mapsto -\theta$ の下で対称となる。
   1. 1 次元グリッド上で事後分布を計算せよ。$\mathbb{E}[\theta \mid \mathcal{D}]$、$\mathbb{E}[\theta^2 \mid \mathcal{D}]$、およびモードにおける事後密度に対する $\theta = 0$ における事後密度の比を報告せよ。
   2. 順方向ダイバージェンス $D_{\mathrm{KL}}(p\|q)$ を最小化するガウス分布が p の平均と分散に一致することを示せ。今回の事後分布に対してこのガウス分布はどこに最頻値を置き、その点は p の下でどれほどもっともらしいか。
   3. 正のモードにおけるラプラス近似、$m = 0.5$ から出発して変分推論のセルの ELBO 上昇法で当てはめた 1 次元ガウス分布、および提案標準偏差 0.2 で $\theta \in \{0.5, 1, 1.5, 2\}$ から出発する 4 本のメトロポリス連鎖という 3 つの局所近似を求めよ。続いて、初期値を $\theta \in \{-2, -0.5, 0.5, 2\}$ に変えて連鎖を再実行せよ。単一のモードしか報告しないのはどの手法か。各実行において、診断指標（連鎖については分割 $\widehat{R}$ と ESS、変分近似については ELBO）のうち問題を明らかにするのはどれか。
   4. 単一モードの近似でも依然として正しく推定できるのはどのような事後期待値か。h 個の tanh 隠れユニットを持つネットワークには、各ユニットに対するこの問題の符号反転に加え、隠れユニットの $h!$ 通りの置換が存在する（演習問題 6）。等方的なガウス事前分布の下で、その事後分布には 1 つのモードの等価なコピーが何個存在するか。このことは、ニューラルネットワークの事後分布を本節のいずれかの手法で近似することについて何を示唆するか。
8. [code] **ロジスティック事後分布の裾。**●●重要度サンプリングにおいて中心極限定理が成り立つには、重みの二次モーメントが有限である必要がある。重要度サンプリングのセルにおけるラプラス提案分布は、通常は高い ESS に達する。以下を通じて、$\tau^2 = $ `prior_var` $= 4$ は事後分布 Equation 36.97 における事前分散である。
   1. 単位ベクトル $\mathbf{v}$ を固定し、$\theta = t\mathbf{v}$ とおく。Equation 36.97 の対数尤度が $-c(\mathbf{v})t + O(1)$（$t \to \infty$）に等しくなることを示せ。ここで $c(\mathbf{v})$ は $|\mathbf{z}_i^\top \mathbf{v}|$ を、$y_i = 1$ かつ $\mathbf{z}_i^\top \mathbf{v} < 0$、または $y_i = 0$ かつ $\mathbf{z}_i^\top \mathbf{v} > 0$ を満たす観測にわたって足し合わせた和である。どのようなデータセットに対して、$c(\mathbf{v}) > 0$ がすべての $\mathbf{v}$ で成り立つか。
   2. これから、ガウス提案分布 $\mathcal{N}(\mathbf{m}, \mathbf{S})$ は $\mathbb{E}_q[w^2] = \infty$（$\mathbf{S}$ のある固有値が $\tau^2/2$ より小さいとき）および $\mathbb{E}_q[w^2] < \infty$（$\mathbf{S}$ のすべての固有値が $\tau^2/2$ を超えるとき）となることを結論づけよ。重要度サンプリングのセルにおける 2 つの提案分布のうち、有限分散の重みを持つのはどちらか。この答えを、事後分布をガウス分布で近似したときに演習問題 5a の公式がラプラス提案分布に対して与える答えと比較せよ。
   3. `laplace_cov` の各固有ベクトルに沿って両方向に調べ、θ_{MAP} からのマハラノビス距離がいくつを超えると $\tilde{p}^2/q$ が θ_{MAP} における値を超えるかを求めよ。ラプラス提案分布からの 1 個のサンプルがそれほど遠くに落ちる確率はどの程度か。その答えと実行時に出力される ESS との整合性を説明せよ。

4. Geweke (1989) は事後最頻値における多変量スチューデント t 近似から重要度密度を構成している。自由度 3、位置 θ_{MAP}、スケール行列 `laplace_cov` を持つスチューデント t 提案分布を用い、その重みが有限の分散を持つことを示せ。20 回の実行にわたり、ESS と裾確率 $P(\theta_1 > \bar{\theta}_1 + 3s_1 \mid \mathcal{D})$（$\bar{\theta}_1$ と s_1 はグリッドから求めた傾きの平均と標準偏差）の推定値についてラプラス提案分布と比較せよ。基準値には 2 倍の幅を持つグリッド上で事後分布を評価した値を用いよ。また本文の防御的混合分布も試し、なぜその重みが有界になるのかを説明せよ。

9. [code] **ランジュバン提案。●●** メトロポリス調整付きランジュバンアルゴリズムは $\theta' = \theta + \frac{\epsilon^2}{2}\mathbf{C}\nabla \log \tilde{p}(\theta) + \epsilon\,\mathbf{C}^{1/2}\xi$（ただし $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$、$\mathbf{C}$ は前処理行列）を提案する。ここでは $\mathbf{C}$ をラプラス共分散 $\mathbf{H}^{-1}$、$\mathbf{C}^{1/2}$ をそのコレスキー因子とせよ。
   1. 提案密度 $g(\theta' \mid \theta)$ と受理規則 Equation 36.106 におけるヘイスティングス因子を書き下せ。この提案分布が対称でないのはなぜか。
   2. `grad_log_joint` を用いてこのサンプラを実装し、$\epsilon \in \{0.5, 1, 1.5\}$ に対して 4 本の連鎖を実行せよ。受理率、分割 $\widehat{R}$、および勾配評価 1 回あたりの ESS を報告し、ランダムウォークにおける対数同時密度評価 1 回あたりの ESS と比較せよ。
   3. 受理ステップを除いた連鎖は未調整ランジュバンアルゴリズムとなる。$\epsilon = 1$ でこれを実行し、平均と標準偏差をグリッドの値と比較せよ。離散化による統計的バイアスとモンテカルロ誤差の 2 つの誤差のうち、メトロポリスステップが取り除くのはどちらか。

36.9 集中不等式

モデルのテスト正解率を報告するとき、それは有限標本に基づく測定値を報告しているにほかならない。36.5 節では、中心極限定理（central limit theorem）を用いてその測定値にエラーバー $\pm 1.96\,\hat{\sigma}/\sqrt{n}$ を付与した。このガウス近似は手軽で精度も高いことが多いが、厳密に成り立つのは $n \to \infty$ の極限においてのみであり、推定された標準偏差 $\hat{\sigma}$ のプラグインに依存している。それより前の 36.1 節では、任意の有限な n で近似なしに成り立つ上界としてチェビシェフの不等式を見たが、あまりにも緩すぎることがわかった。ここから自然な問いが生じる。手元にある実際の標本サイズで成り立ち、しかもめったに失敗しないことを要求してもなお有用であり続ける保証が欲しいとき、テストセット誤差は真の値からどれだけ離れうるのだろうか。

この隔たりを具体的に見るため、830 個の誤りを 10,000 枚のテスト画像に対して出す分類器、すなわち観測された誤り率が 8.3% の場合を考えよう。チェビシェフの不等式は、n 個の独立な抽出の平均がその期待値（真の平均）から t 以上外れる確率を $\sigma^2/(nt^2)$ で上から抑える。0/1 の結果の分散は高々 $\sigma^2 \le \tfrac{1}{4}$ であるため、チェビシェフの不等式は真の誤り率が $\pm 2.2\%$ 以内で 8.3% の周りに収まることを 95% の信頼度で保証する。これはすでにガウス区間より広く、さらに悪いことに高い信頼性を求めると破綻してしまう。100 万回に 1 回しか失敗しない保証を求めると、チェビシェフの不等式が返す答えは $\pm 500\%$ となり、何の情報も与えない。しかし、0/1 の結果の平均が真の平均から大きく外れる真の確率は、n に関して**指数関数的に**減衰し、$1/n$ のような遅い減衰にはならない。この指数減衰を捉えた上界は、$\pm 1.4\%$（信頼度 95%）および $\pm 2.7\%$（100 万回に 1 回の失敗確率）という答えを与える。

こうした鋭い上界の背後にある核心的な工夫は驚くほど単純である。マルコフの不等式を平均そのものではなくその指数関数に適用することで、独立な項の和を積へと変換するのである。Sergei Bernstein は 1920 年代にこの種の上界を発展させ、Herman Chernoff は 1952 年に統計的検定がどれほど速く信頼できるようになるかを測るためにこの手法を用い、Wassily Hoeffding は 1963 年に機械学習の随所で用いられる簡潔な定式化を与えた (Hoeffding 1963)。

実務家はなぜ、100 万回に 1 回しか失敗しないような保証を求めることがあるのだろうか。実際には、ただ 1 つのモデルだけを評価することはまれであり、学習アルゴリズムは多数の候補を比

較して最良のものを残す。そのうちの**どれ 1 つとして**偶然よく見えているのではないと保証するには、個々の候補に対する失敗確率を極めて小さく抑える必要があり、現実的な標本サイズでそれを可能にするのは指数関数的に縮小する裾だけである。このテーマは 36.10 節で本格的に展開する。本節ではまず、指数型の裾上界を用いて目標の精度を達成するためのテストセットのサイズを決定する。続いて、まったく同じ道具立てを $\mathbb{R}^d$ 内の確率ベクトルの各座標に適用し、2 つの際立った幾何学的事実を明らかにする。すなわち、$\mathbb{R}^{784}$ の標準ガウスベクトルはほぼ確実に 28 の数パーセント以内の長さを持ち、独立な 2 つのランダムな方向はほぼ直交するのである。これら 2 つの事実は、ニューラルネットワークの初期化と高次元の埋め込みの双方を支えている。

本節の結果のうち、次の 3 つは記憶にとどめておく価値がある。チェルノフ法 Equation 36.112、ヘフディングの信頼区間 Equation 36.116、そしてユークリッドノルムの集中 Equation 36.121 である。

```
rng = np.random.default_rng()
```

36.9.1 指数型の裾上界

ガウス信頼区間が狭いのは、ガウス分布の裾が $e^{-z^2/2}$ のように減衰するからであるが、この減衰レートは中心極限定理に由来するため、大きな n の極限でしか保証されない。ここで欲しいのは、それと同じ速い減衰と、任意の有限標本サイズで成り立つ保証との両立である。そのような上界を構築する前に、チェビシェフの不等式が真の値からどれほどかけ離れているかを確認しておこう。

36.9.1.1 チェビシェフの減衰レート

チェビシェフの上界とは、単にマルコフの不等式（36.1 節）を二乗偏差に適用したものにすぎない。標本平均 $\bar{X}$ を考えよう。これは n 個の独立な抽出の平均であり、各抽出は平均 μ と有限の分散 σ^2 を持つとする。二乗偏差の期待値は $\mathbb{E}[(\bar{X}-\mu)^2] = \mathrm{Var}(\bar{X}) = \sigma^2/n$ であるから、マルコフの不等式より次式が得られる。

$$P(|\bar{X}-\mu| \ge t) \le \frac{\sigma^2}{n t^2}.$$

ここでは、外れる確率は $1/n$ の速さでしか減少しない。失敗確率を 1 から 10^{-6} まで（偏差 t を固定したまま）引き下げるには、標本サイズを 100 万倍に増やさなければならない。

実際に外れる確率はどれほど小さいのだろうか。厳密に計算できる最も単純な設定は公平なコインである。$\hat{p}$ を n 回の独立なコイン投げで表が出た割合とし、$\hat{p}$ が $\frac{1}{2}$ から $t = 0.1$ 以上外れる確率を求めよう。1 回の試行の分散は $\frac{1}{4}$ であるから、チェビシェフの上界は $0.25/(0.01\,n) = 25/n$ となる。厳密な確率は二項確率の和であり、$n = 5{,}000$ でも 0 にアンダーフローしないよう対数空間で評価する。

```
def exact_tail(n, t):
    """P(|p_hat - 1/2| >= t) for n fair flips, a binomial sum in log space."""
    k = np.arange(n + 1)
    log_pmf = n * np.log(0.5) + np.concatenate(
        ([0.0], np.cumsum(np.log((n - k[1:] + 1) / k[1:]))))
    in_tail = log_pmf[np.abs(k / n - 0.5) >= t - 1e-12]  # fp-safe boundary
    m = in_tail.max()
```

```
    return np.exp(m) * np.exp(in_tail - m).sum()  # stable log-sum-exp
```

$n = 1{,}000$ のとき、チェビシェフの上界は 40 回の実験のうち 1 回の割合で外れる可能性を許容する。出力の表を読む前に、そのようなずれが実際にはどのくらいの頻度で起きるか予想してみてほしい。

```
t = 0.1
print(f'{"n":>6} {"exact":>10} {"Chebyshev":>10}')
for n in (10, 100, 1000, 5000):
    print(f'{n:6d} {exact_tail(n, t):10.2e} {0.25 / (n * t**2):10.2e}')
```

```
     n      exact  Chebyshev
    10   7.54e-01   2.50e+00
   100   5.69e-02   2.50e-01
  1000   2.73e-10   2.50e-02
  5000   1.30e-45   5.00e-03
```

$n = 10$ では厳密な裾確率は 0.75 である一方、上界は 2.5 と自明な値にとどまる。$n = 100$ では上界は 0.25 まで下がり、厳密値 0.057 の約 4 倍となる。しかしその先では、2 つの曲線は劇的に乖離していく。$n = 1{,}000$ では厳密な裾確率が 2.7×10^{-10} であるのに対し、チェビシェフの上界は依然として 0.025 を許容しており、8 桁もの開きがある。$n = 5{,}000$ になると、その差は 42 桁にまで広がる。n を 10 倍にするごとにチェビシェフの上界は単に 10 分の 1 になるだけだが、厳密な裾確率はコインを 100 回投げるごとに一定の桁数ずつ小さくなり、対数スケールでは直線に沿って下降していく。

残念ながら、分布についてさらなる仮定を置かない限り、チェビシェフの上界をこれ以上引き締めることはできない。36.1 節で見たとおり、3 点上に台を持つある分布では等号が成立するため、平均と分散のみに依拠するいかなる上界も一般にはこれを改善できないからである。しかし、コイン投げには 2 つの有限モーメントよりはるかに豊かな構造がある。とる値が有界であるため、高次のモーメントもすべて小さく抑えられるのである。すべてのモーメントを一度に活用するために、モーメント母関数に目を向けよう。

36.9.1.2 チェルノフ法

高次のモーメントをどう活用するかを見るため、マルコフの不等式とチェビシェフの不等式に共通する仕組みに注目しよう。関数 g がいたるところで指示関数 $\mathbf{1}\{x \geq t\}$ の上側にある、すなわち指示関数の**優関数**（majorant）であるとする。期待値をとれば直ちに $P(X \geq t) = \mathbb{E}[\mathbf{1}\{X \geq t\}] \leq \mathbb{E}[g(X)]$ が得られる。マルコフの不等式は線形な優関数 $g(x) = x/t$ を非負の X に対して選び、チェビシェフの不等式は二次の優関数 $g(x) = (x/t)^2$ を（中心化された偏差 $X - \mu$ に適用して）選ぶ。ここで任意の $\lambda > 0$ に対し、指数関数 $e^{\lambda(x-t)}$ も実数直線全体で指示関数の上側にあることに注目しよう。Figure 36.43 の (a) は、これら 3 つの優関数を非負の X について比較したものである。上界は単に $g(X)$ の平均（期待値）であるから、その鋭さを決めるのは、関数 g の値が確率変数 X の確率質量の集中している領域でどれだけ小さく保たれるかである。

この指数関数の期待値には標準的な名前がついている。確率変数 X の**モーメント母関数**（moment generating function、MGF）とは、期待値が有限となる範囲で $\mathbb{E}[e^{\lambda X}]$ を λ の関数とみなしたものである。0 を中心とする公平なコイン $Z \in \{-\frac{1}{2}, \frac{1}{2}\}$ の場合、それは単に $\frac{1}{2}(e^{\lambda/2} + e^{-\lambda/2}) = \cosh(\lambda/2)$ となる。この MGF は任意の λ に対して有限になる（X が有界である限り成り立つ）が、任意の $\lambda > 0$ に対して無限大に発散してしまうこともある（スチューデントの t 変数のように右裾が多項式的にしか減衰しない場合）。そのような裾の重い確率変数には、以下の指数関数による手法は適用できない。

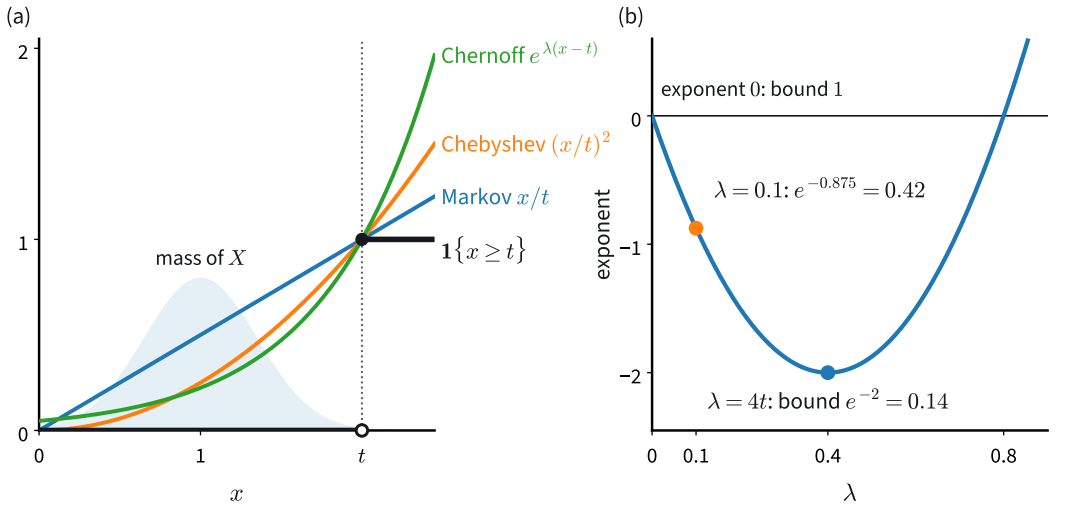


Figure 36.43: 裾上界は指示関数の上側にある関数の期待値として与えられる。(a) 閾値 $t = 2$ に対する指示関数 $\mathbf{1}\{x \geq t\}$（黒）と、その上側にある 3 つの関数：マルコフの x/t、チェビシェフの $(x/t)^2$、およびチェルノフの $e^{\lambda(x-t)}$（$\lambda = 1.5$）。網掛けの山は X の確率質量が存在する領域を表す。各上界はその質量のもとでの関数の平均であり、λ を大きくするとその領域で指数関数は小さくなる。(b) チェルノフ指数 $-\lambda nt + n\lambda^2/8$（公平なコイン $n = 100$ 枚の平均、$t = 0.1$）。任意の $\lambda > 0$ が有効な上界を与え、放物線の最小点 $\lambda = 4t$ において最良の上界 e^{-2} が得られる。

非負の確率変数 $e^{\lambda X}$ にマルコフの不等式を適用すれば、1 行で上界が得られる。$x \mapsto e^{\lambda x}$ は $\lambda > 0$ のとき狭義単調増加であるため、事象 $\{X \geq t\}$ は $\{e^{\lambda X} \geq e^{\lambda t}\}$ と一致する。したがって、次式が成り立つ。

$$P(X \geq t) = P\big(e^{\lambda X} \geq e^{\lambda t}\big) \leq e^{-\lambda t}\,\mathbb{E}\big[e^{\lambda X}\big], \qquad \text{hence} \qquad P(X \geq t) \leq \inf_{\lambda > 0} e^{-\lambda t}\,\mathbb{E}\big[e^{\lambda X}\big]. \tag{36.112}$$

$\lambda > 0$ をどう選んでも有効な上界が得られ、下限（infimum）をとる操作はその中で最も鋭いものを選び出しているにすぎない。この 2 段階の処方箋が**チェルノフ法**（Chernoff method）である。

指数関数が適切な優関数であるのには、同じくらい決定的な第 2 の理由がある。$S = X_1 + \cdots + X_n$ が独立な確率変数の和であるとき、$e^{\lambda S} = \prod_{i=1}^{n} e^{\lambda X_i}$ は独立な因子の積になり、独立な確率変数の積の期待値はそれぞれの期待値の積になる。

$$\mathbb{E}\big[e^{\lambda S}\big] = \prod_{i=1}^{n} \mathbb{E}\big[e^{\lambda X_i}\big].$$

このように、和のモーメント母関数は各項の MGF の積へと因数分解されるため、チェルノフ上界の対数は n 個の項の和に分解され、n に対して線形にスケールする。多項式の優関数はこの性質を持たない。実際、$\mathbb{E}[S^4]$ は 4 次モーメントの積には分解されない。残る課題は、和を構成する 1 つの項の MGF を上から抑えることだけである。

36.9.1.3 公平なコイン

公平なコインの場合を最後まで具体的に計算してみよう。この最も単純なケースに、一般の証明のすべてのステップがすでに含まれているからである。中心化されたコイン投げ $Z = X - \frac{1}{2}$ は値 $\pm\frac{1}{2}$ を等確率でとるため、その MGF は $\mathbb{E}[e^{\lambda Z}] = \cosh(\lambda/2)$ である。$\cosh u$ と $e^{u^2/2}$ のテイラー級数を項別に比較すると、次式が得られる。

$$\cosh u = \sum_{k\geq 0} \frac{u^{2k}}{(2k)!} \leq \sum_{k\geq 0} \frac{(u^2/2)^k}{k!} = e^{u^2/2},$$

これは $(2k)! \geq 2 \cdot 4 \cdots (2k) = 2^k k!$ が成り立つからである。実際、$(2k)!$ の偶数因子だけでもすでに $2^k k!$ になり、残る奇数因子はすべて少なくとも 1 以上である。$u = \lambda/2$ とおけば、1 回の中心化されたコイン投げは $\mathbb{E}[e^{\lambda Z}] \leq e^{\lambda^2/8}$ をすべての λ に対して満たすことがわかる。比較すると、ガウス分布 $\mathcal{N}(0, \sigma^2)$ は（積分内で平方完成することにより）$\mathbb{E}[e^{\lambda X}] = e^{\lambda^2\sigma^2/2}$ を満たすため、中心化されたコインの MGF は標準偏差 $\frac{1}{2}$ のガウス分布の MGF を決して超えない。

ここで n 回の独立なコイン投げを考える。事象 $\hat{p} - \frac{1}{2} \geq t$ は $\sum_{i=1}^{n} Z_i \geq nt$ と同値であるから、チェルノフ法と積の法則より次式が得られる。

$$P(\hat{p} - \tfrac{1}{2} \geq t) \leq e^{-\lambda nt} \prod_{i=1}^{n} \mathbb{E}[e^{\lambda Z_i}] \leq \exp\!\Big(-\lambda nt + \frac{n\lambda^2}{8}\Big).$$

指数部は λ に関する単純な放物線である。Figure 36.43 の (b) は $n = 100$、$t = 0.1$ の場合をプロットしたもので、このとき指数部は $-10\lambda + 12.5\lambda^2$ に等しい。任意の $\lambda > 0$ が有効な上界を与え、たとえば $e^{-0.875} \approx 0.42$ が $\lambda = 0.1$ で得られる。最も鋭い上界は放物線の頂点 $\lambda = 4t$ で達成され、そこで指数部は $-4nt^2 + 2nt^2 = -2nt^2$ へと簡約される。対称性により下側の裾も同じ上界に従うため、両側の裾を足し合わせると次式が得られる。

$$P(|\hat{p} - \tfrac{1}{2}| \geq t) \leq 2e^{-2nt^2}.$$

これにより、t 以上外れる確率は nt^2 に関して指数関数的に減衰する。$n = 100$、$t = 0.1$ では $2e^{-2} \approx 0.27$ となり、$n = 1{,}000$ では $2e^{-20} \approx 4.1 \times 10^{-9}$ まで低下する。チェビシェフの上界 0.025 と比べれば劇的な改善である。

なぜ指数型の優関数はこれほど効果的に働くのだろうか。閾値 $S = nt$ において関数 $e^{\lambda(S-nt)}$ は 1 に等しい一方、確率質量の大部分が集まっている平均 $S = 0$ においては $e^{-\lambda nt}$ まで縮小し、n に関して指数関数的に小さくなる。λ を無限大へと飛ばせない理由は、この関数が $S > 0$ で急激に増大するため、上方向の揺らぎが過大な重みを持ってしまうことにある。$e^{n\lambda^2/8}$ で上から抑えられる MGF の積がそのペナルティを測っており、最適な $\lambda = 4t$ が両者の厳密なバランスをとっている。

以上の導出全体が、コインのただ 1 つの性質だけに依拠していた点に注目しよう。すなわち、中心化された 1 回の試行の MGF が $e^{\lambda^2/8}$、つまりコインの値域の半分である標準偏差 $\frac{1}{2}$ のガウス分布の MGF で抑えられるという性質である。ヘフディングの補題はこの 1 ステップを一般化し、区間 $[a, b]$ に値をとる**すべての**確率変数が、$(b-a)/2$ で $\frac{1}{2}$ を置き換えた同じ上界に従うことを示す。

36.9.1.4 ヘフディングの補題とヘフディングの不等式

テストセットの損失が公平なコインと同じ振る舞いをすることはめったにない。8.3% という誤り率は大きく偏ったコインのように振る舞い、クリップされた損失は 0 と 1 だけでなく連続的な値をとる。そこで、公平なコインで得た指数型の上界が**任意の**有界な確率変数に対して成り立つようにしたい。これを実現する補題が本節における主要な技術的ステップであり、その証明は分散に関する 1 つの初等的な事実に基づいている。

命題（有界な確率変数の分散） Y **が** $[a,b]$ **に値をとるとき、** $\mathbf{Var}(Y) \le (b-a)^2/4$ **が成り立つ。等号は、** Y **が値** a **と** b **をそれぞれ確率** $\tfrac{1}{2}$ **でとるときに成立する。**

証明： 平均は二乗偏差の期待値を最小化する。実際、$\mathbb{E}[(Y-c)^2] = \mathrm{Var}(Y) + (\mathbb{E}[Y]-c)^2$ が中心 c の選び方によらず成り立つからである。ここで $c = (a+b)/2$ ととれば、$[a,b]$ のすべての点は中点から $(b-a)/2$ 以内の距離にあるため、次式が得られる。

$$\mathrm{Var}(Y) \le \mathbb{E}\Big[\Big(Y - \frac{a+b}{2}\Big)^2\Big] \le \Big(\frac{b-a}{2}\Big)^2.$$

Y が両端点 a と b に等しい確率質量を置くとき、2 つの不等号はいずれも等号になる。■

$[0,1]$ 上の公平なコインはまさにこの極端なケースにあたり、分散 $\tfrac{1}{4}$ を持つ。この分散の上界を手にすれば、ヘフディングの補題を証明できる。

命題（ヘフディングの補題） X **を** $\mathbb{E}[X]=0$ **を満たし** $[a,b]$ **に値をとる確率変数とする。このとき、すべての** $\lambda \in \mathbb{R}$ **に対して次式が成り立つ。**

$$\mathbb{E}\big[e^{\lambda X}\big] \le \exp\!\left(\frac{\lambda^2 (b-a)^2}{8}\right). \tag{36.113}$$

証明： 対数 MGF である $\psi(\lambda) = \log \mathbb{E}[e^{\lambda X}]$ の増大が、放物線 $\lambda^2(b-a)^2/8$ を超えないことを示したい。X は有界であるため、ψ はいたるところ有限かつ滑らかであり、期待値記号の下で微分できる。$\lambda = 0$ において $\psi(0) = \log 1 = 0$ であり、さらに次が成り立つ。

$$\psi'(\lambda) = \frac{\mathbb{E}[X e^{\lambda X}]}{\mathbb{E}[e^{\lambda X}]}, \qquad \text{so} \qquad \psi'(0) = \mathbb{E}[X] = 0.$$

指数分布族の対数分配関数（36.4 節）とまったく同様に、ψ の二階導関数は分散になる。**傾斜**（tilted）分布 q_λ を、X の分布に $e^{\lambda x}$ の重みを掛けて正規化したものとして定義しよう。q_λ のもとでは $\psi'(\lambda)$ は X の平均であり、もう一度微分すると次式が得られる。

$$\psi''(\lambda) = \frac{\mathbb{E}[X^2 e^{\lambda X}]}{\mathbb{E}[e^{\lambda X}]} - \left(\frac{\mathbb{E}[X e^{\lambda X}]}{\mathbb{E}[e^{\lambda X}]}\right)^2 = \mathrm{Var}_{q_\lambda}(X).$$

重要なのは、指数傾斜は各点の重みを付け替えるだけであり、確率質量を $[a,b]$ の外へ移すことは決してないという点である。したがって、先ほどの分散の上界はすべての q_λ（すなわちすべての λ）に対して成り立ち、$\psi''(\lambda) \le (b-a)^2/4$ を与える。あとはラグランジュの剰余項を用いたテイラーの定理によって証明が完結する。ある ξ が 0 と λ の間に存在して、次式が成り立つ。

$$\psi(\lambda) = \psi(0) + \lambda\,\psi'(0) + \frac{\lambda^2}{2}\,\psi''(\xi) \le \frac{\lambda^2 (b-a)^2}{8},$$

両辺の指数をとれば Equation 36.113 が得られる。$\blacksquare$

言葉で言えば、有界で中心化された任意の確率変数のモーメント母関数は、標準偏差 $(b-a)/2$ のガウス分布のモーメント母関数以下となる。中心化されたコインでは $b-a=1$ であるから、この補題はテイラー級数によって証明した上界 $\cosh(\lambda/2) \le e^{\lambda^2/8}$ を直ちに再現する。このたった 1 つの部品が揃えば、コインに対する証明は独立な有界確率変数の任意の平均へとそのまま引き継がれる。

命題（ヘフディングの不等式） **$X_1, \dots, X_n$ を $X_i \in [a,b]$ を満たす独立な確率変数とし、$\bar{X} = \frac{1}{n}\sum_i X_i$ とする。このとき、すべての $t > 0$ に対して次式が成り立つ。**

$$P\big(|\bar{X} - \mathbb{E}[\bar{X}]| \ge t\big) \le 2\exp\!\left(-\frac{2nt^2}{(b-a)^2}\right). \tag{36.114}$$

証明： $Z_i = X_i - \mathbb{E}[X_i]$ とおく。各 Z_i は平均 0 で長さ $b-a$ の区間に値をとるため、すべての項にヘフディングの補題が適用できる。任意の $\lambda > 0$ に対し、チェルノフ法 Equation 36.112 と積の法則より次式が得られる。

$$\begin{aligned} P(\bar{X} - \mathbb{E}[\bar{X}] \ge t) &= P\Big(\sum_{i} Z_i \ge nt\Big) \le e^{-\lambda n t} \prod_{i=1}^{n} \mathbb{E}\big[e^{\lambda Z_i}\big] \\ &\le \exp\!\left(-\lambda n t + \frac{n\lambda^2 (b-a)^2}{8}\right). \end{aligned}$$

これは $(b-a)^2$ で 1 を置き換えただけで、コインの場合に最小化したものと同じ放物線である。最小値は $\lambda = 4t/(b-a)^2$ で達成され、そのとき指数部は $-2nt^2/(b-a)^2$ に等しい。$-Z_i$ に同じ議論を適用すれば下側の裾も同じ量で抑えられ、両側の裾を足し合わせることで係数 2 が生じる。$\blacksquare$

このように、n 個の独立な有界変数の標本平均が期待値から t 以上外れる確率は、nt^2 に関して指数関数的に減衰する（ここで t は値域の幅 $b-a$ を単位として測られる）。この上界が何を必要と**しない**かに注目しよう。ガウス性も、分散の推定値も、漸近極限も不要である。求められるのは有界性と独立性だけであり、任意の有限な n に対して成り立つ。ただし独立性は本質的な要件であり、マルコフ連鎖の相関を持つサンプルにはこの上界は適用できない。

証明を振り返ると、実はすべての区間の長さが等しい必要はどこにもなかったことに気づくだろう。独立な中心化確率変数が $Y_i \in [a_i, b_i]$ を満たすなら、ヘフディングの補題による上界の積をとることで $n(b-a)^2$ が $\sum_i (b_i - a_i)^2$ に置き換わり、同じ二次関数の最小化から次式が得られる。

$$P\Big(\Big|\sum_{i} Y_i\Big| \ge s\Big) \le 2\exp\!\Big(-\frac{2s^2}{\sum_i (b_i - a_i)^2}\Big). \tag{36.115}$$

この異なる値域に対する不等式は、のちほど高次元におけるランダムな方向を解析する際に活用する。

ヘフディングの不等式を実用的なテストセットのサイズへと換算してみよう。$[0,1]$ に値をとる損失では $b-a=1$ であるから、失敗確率が δ 以下となる条件は $n \ge \log(2/\delta)/(2t^2)$ である。誤り率を誤差 $t = 0.01$ 以内、失敗確率 $\delta = 0.05$ で推定するには、$n \ge \log 40 / (2 \cdot 10^{-4}) = 18{,}444.4$、すなわち 18,445 個のテスト例が必要になる。これは第 3.5 章で挙げた数値であり、漸近的なガウス近似が示唆する 9,604 個のおよそ 2 倍にあたる。

同じことだが、Equation 36.114 の右辺を δ とおいて t について解けば、実務で使われる信頼区間の形式が得られる。すなわち、少なくとも確率 $1-\delta$ で次式が成り立つ。

$$\left|\bar{X} - \mathbb{E}[\bar{X}]\right| \le (b-a)\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{36.116}$$

ガウス区間 $\hat{\mu} \pm 1.96\,\hat{\sigma}/\sqrt{n}$ と同様、区間の半幅は $1/\sqrt{n}$ のレートで縮小するが、こちらには厳密な有限標本保証が伴っている。

830 個の誤りを 10,000 枚の画像に対して出した分類器に $\delta = 0.05$ で立ち返ると、ヘフディングの不等式は $\pm 1.36\%$ を与えるのに対し、ガウス区間は $\pm 1.96\sqrt{0.083 \cdot 0.917 / 10^4} = \pm 0.54\%$ となる。両者の違いは定数のみにある。ヘフディングの不等式は経験標準偏差 $\hat{\sigma} = 0.28$ を最悪ケースの値 $(b-a)/2 = 0.5$ に置き換え、ガウス分布の分位点 1.96 を $\sqrt{2\log(2/\delta)} = 2.72$ に置き換えるため、区間が 2.5 倍広くなるのである。データの実際のばらつきがその値域よりはるかに小さいときには、ヘフディングの不等式はどうしても保守的になるが、この隔たりはすぐ後でベルンシュタインの不等式が埋めることになる。

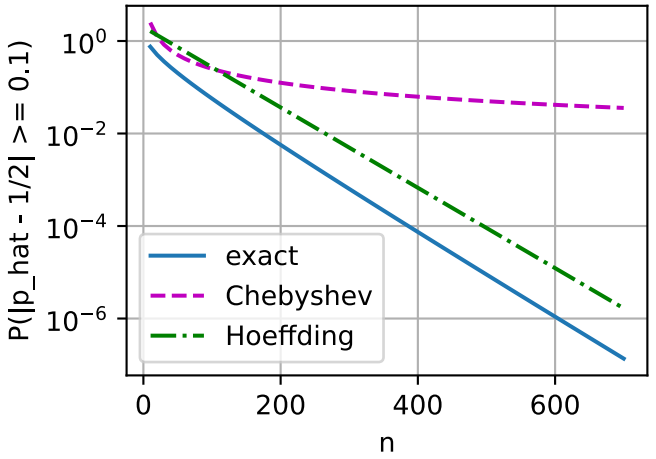
ヘフディングの補題における定数 $\frac{1}{8}$ をさらに小さく改善できるのではないかと疑問に思うかもしれない。しかしそれは不可能である。確率 $\frac{1}{2}$ ずつを $[a, b]$ の両端点に置く確率変数が、実際にこの定数 $\frac{1}{8}$ を λ の 2 次の項において達成するからである（演習問題 1）。

36.9.1.5 数値で見る公平なコイン

ヘフディングの上界 $2e^{-2nt^2}$ が、公平なコインの $t = 0.1$ における厳密な裾確率にどれほど迫るかを確かめてみよう。両者の値とその比を表示し、さらにチェビシェフの上界と合わせて n に対する推移を対数スケールでプロットする。

```
print(f'{"n":>6} {"exact":>10} {"Hoeffding":>10} {"ratio":>8}')
for n in (10, 100, 1000, 5000):
    hoeffding = 2 * np.exp(-2 * n * t**2)
    print(f'{n:6d} {exact_tail(n, t):10.2e} {hoeffding:10.2e} '
          f'{hoeffding / exact_tail(n, t):8.1f}')

ns = np.arange(10, 701, 10)
d2l.plot(ns, [np.array([exact_tail(n, t) for n in ns]),
              0.25 / (ns * t**2), 2 * np.exp(-2 * ns * t**2)],
         'n', 'P(|p_hat - 1/2| >= 0.1)',
         legend=['exact', 'Chebyshev', 'Hoeffding'], yscale='log')
```

```
     n      exact  Hoeffding     ratio
    10   7.54e-01   1.64e+00       2.2
   100   5.69e-02   2.71e-01       4.8
  1000   2.73e-10   4.12e-09      15.1
  5000   1.30e-45   7.44e-44      57.2
```



$n = 10$ ではどちらの上界も 1 を超え、情報を持たない。$n = 100$ では、実はチェビシェフの 0.25 の方がヘフディングの 0.27 よりわずかに鋭い。標本サイズが小さい領域では先頭の定数が効いてくるため、厳密な分散 $\frac{1}{4}$ を知っているチェビシェフの不等式が有利になるからである。しかし n が大きくなると、指数減衰レートが完全に支配的になる。ヘフディングの上界は厳密な裾確率に対し、15 倍以内に $n = 1{,}000$ で収まり、57 倍以内に $n = 5{,}000$ でもとどまっている。

対数プロット上では、厳密な裾確率とヘフディングの上界はほぼ平行な直線を描く。コインを 1 回投げるごとに、ヘフディングの上界の対数は $2t^2 = 0.02$ ずつ減少するのに対し、厳密な裾確率は約 0.0201 ずつ減少する。このわずかに急な傾きは、カルバック・ライブラー・ダイバージェンス $D_{\mathrm{KL}}(0.6 \| 0.5) = 0.6 \log 1.2 + 0.4 \log 0.8$、すなわち表の確率が 0.6 の偏ったコインと公平なコインとの間のダイバージェンスにほかならない（37.3 節）。ベルヌーイ変数の厳密な MGF を用いてチェルノフ法を適用すれば、この厳密な減衰レートが導かれる（演習問題 3）。

もちろん、n がごく小さいときには、いかなる不等式も存在しない証拠を作り出すことはできない。本章の導入部で登場した、表が 9 回（13 回中）出たコインを思い出そう。これは $t = 2.5/13$ だけ $\frac{1}{2}$ からずれた事象である。このようなずれに対するヘフディングの上界は $2e^{-2 \cdot 13 \cdot (2.5/13)^2} = 2e^{-0.96} \approx 0.77$ であり、何も棄却できない。厳密な両側二項確率でさえ 0.27 である。13 回のコイン投げでは、公平なコインである可能性を到底排除できないのである。

36.9.1.6 準ガウス確率変数と準指数確率変数

誤り率 8.3% の分類器に対するヘフディングの区間がガウス区間より 2.5 倍広くなったのは、0/1 損失の値域だけを使い、その小さな分散を無視しているからであった。**ベルンシュタインの不等式**は、その両方を取り入れる。

命題（ベルンシュタインの不等式） $X_1, \dots, X_n$ **を独立な確率変数とし、**$\mathbf{Var}(X_i) \le \sigma^2$ **および** $|X_i - \mathbb{E}[X_i]| \le M$ **を満たすとする。このとき、すべての** $t > 0$ **に対して次式が成り立つ。**

$$P\big(|\bar{X} - \mathbb{E}[\bar{X}]| \ge t\big) \le 2 \exp\left(- \frac{n t^2}{2(\sigma^2 + M t/3)} \right). \tag{36.117}$$

ここでも証明はチェルノフ法 (Boucheron et al. 2013) であり、分散と値域の双方を反映した MGF の上界と組み合わせることで得られる（詳細は演習問題 7 で導出する）。分母に現れる 2 つの領域に注目しよう。t が小さいときは σ^2 が $Mt/3$ を支配し、上界は**真の**分散 σ^2 を持つガウス

分布の裾のように振る舞う。一方、t が大きいときは 1 次の項 $Mt/3$ が支配的になり、裾は t に関して指数関数的に減衰するにとどまり、ガウス分布のように t^2 では減衰しない。

分散を考慮することは、実務上どれほど役に立つのだろうか。誤り率 2% の分類器を $n = 10{,}000$ 個のテスト例で評価する場合を考えよう。各 0/1 損失は分散 $\sigma^2 = 0.02 \cdot 0.98$ を持ち、中心化された偏差は $M = 1$ を満たす。Equation 36.117 の右辺を δ とおくと、t に関する二次方程式 $nt^2/(2(\sigma^2 + t/3)) = \log(2/\delta)$ が得られる。以下では $\delta = 0.05$ において、誤り率 8.3% と 2% の両方についてこの二次方程式を解き、ヘフディングの区間およびガウス区間と比較する。ベルンシュタインの区間については、（観測された誤り率を分散に代入した）プラグイン推定値と、出力の直後で説明する完全保証版の 2 通りを計算する。

```
def half_widths(p, n, delta=0.05, z=1.96):
    """Hoeffding, Bernstein (plug-in, guaranteed), Gaussian half-widths."""
    L = np.log(2 / delta)
    hoeffding = np.sqrt(L / (2 * n))
    c = 2 * L / 3  # Bernstein with M = 1: n t^2 - c t - 2 L var = 0
    bernstein = lambda var: (c + np.sqrt(c**2 + 8 * n * L * var)) / (2 * n)
    plug_in = t = bernstein(p * (1 - p))
    for _ in range(20):  # true rate at most p + t: a fixed point in t
        t = bernstein((p + t) * (1 - p - t))
    return hoeffding, plug_in, t, z * np.sqrt(p * (1 - p) / n)
```

`half_widths(p, 10000)` を 8.3% と 2.0% の両方の誤り率で評価すると、4 つの手法すべてにおける 95% 信頼区間の半幅を次のように比較できる。

```
for p in (0.083, 0.02):
    hoeffding, plug_in, bernstein, gauss = half_widths(p, 10000)
    print(f'error {p:.1%}: Hoeffding ±{hoeffding:.2%}, Bernstein '
          f'±{plug_in:.2%} (plug-in) ±{bernstein:.2%} (guaranteed), '
          f'Gaussian ±{gauss:.2%}')
```

```
error 8.3%: Hoeffding ±1.36%, Bernstein ±0.76% (plug-in) ±0.79% (guaranteed), Gaussian ±0.54%
error 2.0%: Hoeffding ±1.36%, Bernstein ±0.39% (plug-in) ±0.43% (guaranteed), Gaussian ±0.27%
```

誤り率 2% では、ベルンシュタインのプラグイン半幅は $\pm 0.39\%$ まで縮まり、ヘフディングの $\pm 1.36\%$ と比べて約 3.5 倍鋭く、さらに 1.4 倍以内に漸近ガウス区間の $\pm 0.27\%$ へと迫っている。誤り率 8.3% では改善幅はやや小さくなるものの依然として大きく、$\pm 1.36\%$ が $\pm 0.76\%$ まで狭まる。

厳密に言えば、これら 2 つのプラグイン値（$\pm 0.39\%$ と $\pm 0.76\%$）はまだ真の有限標本保証にはなっていない。上のコードでは観測された誤り率 $\hat{p}$ を分散 $p(1-p)$ に代入したが、ベルンシュタインの不等式が必要とするのは**真の**分散の上界であり、これはガウス区間に対して向けたプラグインの問題点とまったく同じである。幸い、この修正にコストはほとんどかからない。上界が成り立っている限り真の誤り率は高々 $\hat{p} + t$ であるから、最悪ケースの分散 $(\hat{p} + t)(1 - \hat{p} - t)$ を用いて同じ方程式を解けばよい。（コードセルで行っているように）この不動点方程式を反復すると、半幅はわずかに広がるだけで、完全に保証された $\pm 0.43\%$ と $\pm 0.79\%$ が得られる。これにより冒頭の問いに対する鋭い答えが定まる。830 個の誤りを 10,000 枚の画像に対して出した分類器では、真の誤り率は約 0.8% 以内で 8.3% の周りに 95% の信頼度で収まるのである。実用上のモデルは通常数パーセントの誤り率で動作するため、分散を考慮した上界は大きな違いをもたらす。

一歩引いて見直すと、ヘフディングの証明が依拠していたのは和の各項のただ 1 つの性質、すなわちモーメント母関数がガウス分布のモーメント母関数で上から抑えられることだけであった。

この性質に名前をつけておけば、同じ証明を非有界な確率変数へと拡張できるとともに、どこから先でその議論が成り立たなくなるかも明確になる。確率変数 X が**準ガウス**（sub-Gaussian）で**分散プロキシ**（variance proxy）σ^2 を持つとは、その中心化された MGF が $\mathcal{N}(0,\sigma^2)$ の MGF を決して超えないことをいう。

$$\mathbb{E}\big[e^{\lambda(X-\mathbb{E}[X])}\big] \le \exp\left(\frac{\lambda^2\sigma^2}{2}\right) \qquad \text{for all } \lambda \in \mathbb{R}. \tag{36.118}$$

任意の準ガウス確率変数は、少なくともガウス分布と同程度に軽い裾を持つ。実際、$\lambda = t/\sigma^2$ としてチェルノフ法を適用すれば $P(|X-\mathbb{E}[X]| \ge t) \le 2e^{-t^2/(2\sigma^2)}$ が得られる。ただし、ここでの σ^2 はあくまで**プロキシ**であり、必ずしも真の分散ではないことに留意しよう。Equation 36.118 の両辺を λ の 2 次まで比較すると、このプロキシは決して $\mathrm{Var}(X)$ より小さくならないことがわかり、むしろはるかに大きくなる場合がある。たとえば 0/1 の結果で誤り率が 2% のとき真の分散は 0.0196 であるが、ヘフディングの補題が与えるプロキシは $\frac{1}{4}$ であり、まさにそのためにヘフディングの信頼区間はあれほど広くなったのである。

代表的な 3 つの分布族が準ガウスである。ガウス分布 $\mathcal{N}(\mu,\sigma^2)$ は等号成立のもとでプロキシ σ^2 を持ち、$[a,b]$ 上の任意の確率変数はヘフディングの補題によりプロキシ $(b-a)^2/4$ を持ち、一様なランダム符号 $\xi \in \{-1,+1\}$ はプロキシ 1 を持つ（先ほどの級数比較より $\mathbb{E}[e^{\lambda\xi}] = \cosh\lambda \le e^{\lambda^2/2}$ が成り立つためである）。何より好都合なことに、準ガウス性は平均操作のもとで保たれる。$X_1,\dots,X_n$ が独立でプロキシ σ^2 を持つなら、積の法則により $\bar{X}$ はプロキシ σ^2/n を持ち、したがって $P(|\bar{X}-\mathbb{E}\bar{X}| \ge t) \le 2e^{-nt^2/(2\sigma^2)}$ が成り立つ。

しかし、性質のよい確率変数がすべて準ガウスであるわけではなく、すぐ次に必要となる量がまさにその反例となっている。ガウスベクトルの長さの二乗は標準ガウス変数の二乗和であるが、1 つの標準ガウス変数 x について平方完成を行うと次式が得られる。

$$\mathbb{E}\big[e^{\lambda(x^2-1)}\big] = \frac{e^{-\lambda}}{\sqrt{1-2\lambda}} \qquad \text{for } \lambda < \tfrac{1}{2},$$

一方、すべての $\lambda \ge \frac{1}{2}$ に対してこの期待値は無限大に発散する。MGF が $\lambda = \frac{1}{2}$ で発散してしまうため、どのような分散プロキシをとっても Equation 36.118 が x^2 に対して成り立つことはない。その右裾は $e^{-t/2}$ のように減衰し、ガウス型の上界で抑えるには遅すぎるからである。

それでもなお、制限された $\lambda = 0$ の近傍の区間上ではガウス型の MGF 上界が**実際に**成り立つ。確率変数 X が**準指数**（sub-exponential）でパラメータ (ν,α) を持つとは、次式を満たすことをいう。

$$\mathbb{E}\big[e^{\lambda(X-\mathbb{E}[X])}\big] \le \exp\left(\frac{\lambda^2\nu^2}{2}\right) \qquad \text{for } |\lambda| \le \frac{1}{\alpha}. \tag{36.119}$$

ここで ν^2 は $\lambda = 0$ の近傍における局所的な分散プロキシとして働き、$1/\alpha$ は許容区間の半幅を定め、チェルノフ法で選べる λ の大きさに上限を課す。36.9.2.1 節では、x^2 が $\nu = 2$、$\alpha = 4$ の準指数となることを見る。準指数確率変数の平均にチェルノフ法を適用すると、最適化はこの許容区間の境界を境に自然と 2 つに分かれる。初読時には以下の証明を読み飛ばしてもよい。次項で用いるのは命題の主張のみである。

命題（準指数分布の裾上界） $X_1,\dots,X_n$ **を独立な** (ν,α)**-準指数確率変数とする。このとき、すべての** $t > 0$ **に対して次式が成り立つ。**

$$P\big(|\bar{X} - \mathbb{E}[\bar{X}]| \ge t\big) \le 2\exp\!\left(-\frac{n}{2}\min\!\Big(\frac{t^2}{\nu^2},\, \frac{t}{\alpha}\Big)\right). \tag{36.120}$$

証明： 任意の $0 < \lambda \le 1/\alpha$ に対し、チェルノフ法と積の法則により $P(\bar{X} - \mathbb{E}[\bar{X}] \ge t)$ は $\exp(-\lambda n t + n\lambda^2\nu^2/2)$ で上から抑えられる。$t \le \nu^2/\alpha$ のときは、制約なし最小点 $\lambda = t/\nu^2$ が許容区間の内側に収まり、$e^{-nt^2/(2\nu^2)}$ を与える。$t > \nu^2/\alpha$ のときは、その最小点が許容区間の外に出るため、λ を許容される最大値 $\lambda = 1/\alpha$ に固定する。すると指数部は $-nt/\alpha + n\nu^2/(2\alpha^2)$ となり、これは高々 $-nt/\alpha + nt/(2\alpha) = -nt/(2\alpha)$ である（$\nu^2/\alpha < t$ による）。両方の領域を合わせると、指数部は高々 $-\frac{n}{2}\min(t^2/\nu^2, t/\alpha)$ となる。下側の裾も $\lambda \in [-1/\alpha, 0)$ を用いてまったく同様に従う。$\blacksquare$

これで準ガウス変数の平均との対比は極めて明快になった。有界または準ガウスな確率変数の平均は、あらゆるスケールでガウス型の裾 e^{-ct^2} を持つ。これに対し、準指数確率変数の平均は ν^2/α 以下の適度な偏差に対しては同じガウス型の裾を持つが、さらに外側ではより遅い指数型の裾 $e^{-nt/(2\alpha)}$ へと遷移する。

36.9.2 高次元における確率

ここまでの上界は、単一の標本平均を保証するものであった。今度は $\mathbb{R}^d$ 内の d 個の独立な座標を持つ確率ベクトルを考えよう。その長さの二乗も、他のベクトルとの内積も、それら d 個の座標にわたる和である。したがって、高次元空間の幾何学は集中不等式によって支配される。ここから読み始める場合でも、本節の前半から必要な結果は 2 つだけである。長さの二乗に対する準指数分布の裾上界 Equation 36.120 と、内積に対する異なる値域のヘフディング上界 Equation 36.115 である。

36.9.2.1 薄い球殻

標準ガウスベクトル $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{784})$（$28 \times 28$ の画像と同じ数の座標を持つベクトル）をサンプリングしてみよう。その確率密度は原点で最大になる。しかし驚くべきことに、$\mathbf{x}$ が原点から距離 14（典型的な長さの半分）以内に落ちる確率は約 10^{-110}、すなわち 784 個の標準ガウス変数の二乗和が高々 196 以下になる確率にすぎない。コンピュータ上でどれほど多くのサンプルを抽出しても、最頻値の近くに落ちる点は 1 つとして現れないだろう。その代わり、典型的なサンプルはノルム 28.0 ± 0.7 を持つ。

その理由は、$\|\mathbf{x}\|^2 = \sum_{i=1}^{d} x_i^2$ が d 個の独立な項（各平均は 1）の和であり、したがって $\|\mathbf{x}\|^2/d$ が単なる標本平均になっていることにある。x_i^2 は非有界であるためヘフディングの不等式は適用できないが、先ほど導いた準指数上界を用いれば見事に扱える。

命題（ノルムの集中） $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ **とする。すべての** $0 < \varepsilon \le 1$ **に対して次式が成り立つ。**

$$P\!\left(\Big|\frac{\|\mathbf{x}\|}{\sqrt{d}} - 1\Big| \ge \varepsilon\right) \le 2\exp\!\left(-\frac{d\,\varepsilon^2}{8}\right). \tag{36.121}$$

証明： 議論は 3 つの短いステップで進む。第 1 に、べき級数の比較（演習問題 4）から、中心化された二乗の MGF である $e^{-\lambda}/\sqrt{1-2\lambda}$ は次式を満たす。

$$\mathbb{E}\big[e^{\lambda(x_i^2 - 1)}\big] \le e^{2\lambda^2} \qquad \text{for } |\lambda| \le \tfrac{1}{4}. \tag{36.122}$$

Equation 36.119 の用語で言えば、x_i^2 は $\nu = 2$、$\alpha = 4$ の準指数である。第 2 に、準指数分布の裾上界 Equation 36.120 を $n = d$、$t = \varepsilon$ として適用すると次式が得られる。

$$P\Big(\Big|\frac{\|\mathbf{x}\|^2}{d} - 1\Big| \ge \varepsilon\Big) \le 2\exp\Big(-\frac{d}{2}\min\Big(\frac{\varepsilon^2}{4}, \frac{\varepsilon}{4}\Big)\Big) = 2e^{-d\varepsilon^2/8} \qquad \text{for } \varepsilon \le 1.$$

第 3 に、二乗ノルムからノルムそのものへと移る。$z = \|\mathbf{x}\|/\sqrt{d} \ge 0$ とおく。$|z-1| \ge \varepsilon$ が成り立つときは常に、$|z^2 - 1| = |z-1|\,(z+1) \ge \varepsilon$ も成り立つ（$z + 1 \ge 1$ による）。したがって、ノルムに関する事象は二乗ノルムに関する事象に含まれ、その上界をそのまま引き継ぐ。■

要するに、標準ガウスベクトルの長さが $\sqrt{d}$ から一定の割合以上外れる確率は、次元 d に関して指数関数的に減衰する。なお、$\varepsilon \le 1$ という制限は二**乗**ノルムを経由して評価したことに伴う見かけ上の制約にすぎない。二乗ノルムの上側の裾は $\varepsilon = 1$ を超えると $e^{-d\varepsilon/8}$ の指数減衰にとどまるが（演習問題 4）、ノルム自体はあらゆるスケールでガウス型の裾を保つ。実際、ノルムが $(1+\varepsilon)\sqrt{d}$ を超えると二乗ノルムは $(1 + 2\varepsilon + \varepsilon^2)d$ を超え、その確率は高々 $e^{-d\min(s, s^2)/8}$（ただし $s = 2\varepsilon + \varepsilon^2$）であり、$s \ge \varepsilon^2$ かつ $s^2 \ge 4\varepsilon^2$ であるから、これは高々 $e^{-d\varepsilon^2/8}$ で抑えられる。

この典型的なノルムが形作る球殻の厚さはどの程度だろうか。$\mathbb{E}[x^4] = 3$ であるから、各座標の二乗は $\mathrm{Var}(x^2) = 3 - 1 = 2$ を満たし、$\mathrm{Var}(\|\mathbf{x}\|^2) = 2d$ となる。平方根を d の周りで線形化すると $\|\mathbf{x}\| \approx \sqrt{d} + (\|\mathbf{x}\|^2 - d)/(2\sqrt{d})$ となるため、$\|\mathbf{x}\|$ の標準偏差は近似的に $\sqrt{2d}/(2\sqrt{d}) = 1/\sqrt{2} \approx 0.71$ となり、次元によらない。驚くべきことに、$\mathbb{R}^d$ における標準ガウス分布は半径 $\sqrt{d}$ の薄い球殻（thin shell）上に集中し、その厚さは d がどれほど大きくなっても定数オーダーのままである。

なぜ確率質量のすべてが最頻値からこれほど離れた場所に集まるのだろうか。同時確率密度 $(2\pi)^{-d/2}e^{-\|\mathbf{x}\|^2/2}$ は $\mathbf{x}$ にその半径 $r = \|\mathbf{x}\|$ を通じてのみ依存する一方、半径 r の球の表面積は r^{d-1} に比例して増大する。密度に表面積を掛けると、半径 $\|\mathbf{x}\|$ の分布は $r^{d-1}e^{-r^2/2}$ に比例することがわかる。r が大きくなるにつれて、体積因子 r^{d-1} は爆発的に増大する一方で密度 $e^{-r^2/2}$ は急落し、両者の積は $(d-1)\log r - r^2/2$ の導関数が 0 になる点、すなわち $r = \sqrt{d-1}$ でピークを迎える。Figure 36.44 の (a) は、この体積と密度のせめぎ合いを $d = 20$ において示している。

ノルム $\sqrt{d}$ の典型的な点における点ごとの密度は、原点での値の $e^{-d/2}$ 倍（極小の 10^{-170}、$d = 784$ のとき）にすぎないが、半径 $\sqrt{d}$ にはそれを補って余りある表面積があるため、ほぼすべての確率質量がそこに存在する。(b) は $\|\mathbf{x}\|$ の厳密な密度を $d = 2$ から $d = 512$ まで示したものであり、ピークは $\sqrt{d}$ に従って外側へ移動していく一方、その幅は $1/\sqrt{2}$ の近くに留まり続けている。

なお、ここで導いた上界は厳密な数値までは再現しないものの、d に関する指数スケーリングを正しく捉えている。ノルムが 14 未満となる事象は $d = 784$ において $\varepsilon = \frac{1}{2}$ に対応し、このとき Equation 36.121 が与える値は $2e^{-784/32} = 2e^{-24.5} \approx 5 \times 10^{-11}$ であり、厳密値 10^{-110} とは差がある。分母の定数 8 は緩いが、d に関する指数減衰という定性的な結論は的を射ている。

36.9.2.2 ランダムな方向の近似直交性

33.1 節では、固定された単位ベクトルと $\mathbb{R}^d$ 内のランダムな方向のなす角のコサインが分散 $1/d$ を持つことを見た。これをチェビシェフの不等式と組み合わせても、$1/(d\varepsilon^2)$ のオーダーの弱い裾評価しか得られなかった。集中不等式はこれを指数型の裾へと鋭くし、それによって同時にほぼ直交しうるランダムな方向が何本あるかを数えられるようになる。

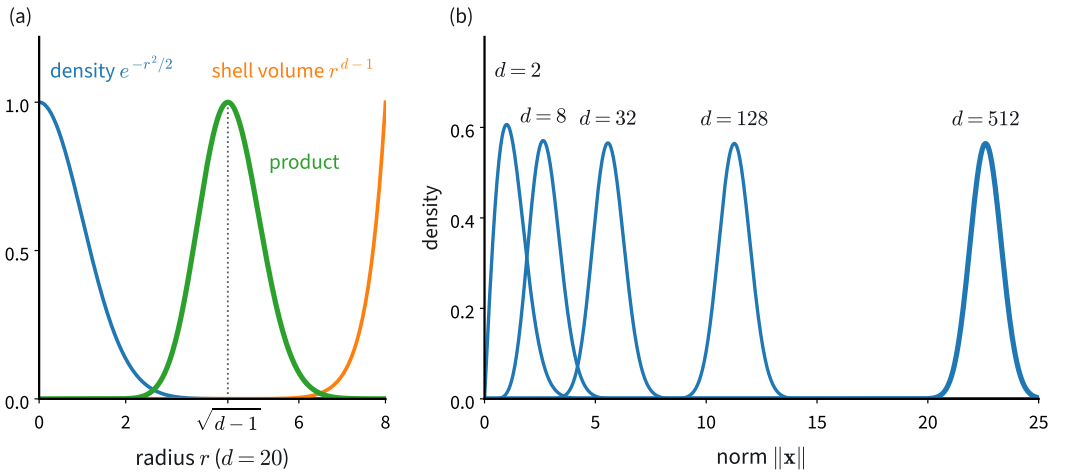


Figure 36.44: 薄い球殻。(a) $d = 20$ において、ガウス密度 $e^{-r^2/2}$ は半径 r とともに減少し、球殻の体積因子 r^{d-1} は半径 r とともに増大する。両者の積である $\|\mathbf{x}\|$ の密度は、密度そのものが最大となる原点から遠く離れた $\sqrt{d-1}$ でピークに達する（各曲線は最大値が 1 になるようスケールしてある）。(b) $\|\mathbf{x}\|$ の厳密な密度（$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$、$d = 2, 8, 32, 128, 512$）。中心は $\sqrt{d}$ に従って外側へ移動し、幅は $1/\sqrt{2}$ の近くに保たれる。

ランダム符号ベクトルを用いると最も見通しよく計算できる。任意の単位ベクトル $\mathbf{u}$ を固定し、独立な一様ランダム符号 $x_i \in \{-1, +1\}$ を抽出する。各積 $u_i x_i$（内積 $\langle \mathbf{x}, \mathbf{u} \rangle$ の各項）は平均 0 であり、区間 $[-|u_i|, |u_i|]$（長さ $2|u_i|$）に収まる。これらの値域を異なる値域のヘフディング上界 Equation 36.115 に代入すると、分母は $4 \sum_i u_i^2 = 4$ となり、次式が得られる。

$$P\big(|\langle \mathbf{x}, \mathbf{u} \rangle| \geq s\big) \leq 2e^{-s^2/2}. \tag{36.123}$$

ランダム符号ベクトルと任意の固定された単位ベクトルとの内積は、$\pm|u_i|$ の寄与の大部分が打ち消し合うため、ガウス型の裾を伴って定数オーダーにとどまる。一方、$\mathbf{x}$ のノルムは $\sqrt{d}$ である。したがって $s = \varepsilon\sqrt{d}$ とおけば、直ちに $P(|\cos(\mathbf{x}, \mathbf{u})| \geq \varepsilon) \leq 2e^{-d\varepsilon^2/2}$ が得られる。標準ガウスベクトル $\mathbf{x}$ に対しても同じ振る舞いが成り立つ。内積 $\langle \mathbf{x}, \mathbf{u} \rangle$ は任意の単位ベクトル $\mathbf{u}$ に対してそれ自体が標準ガウス変数となるため、まったく同じ裾上界 $2e^{-s^2/2}$ に従い、さらに薄い球殻の性質により $\|\mathbf{x}\|$ は $\sqrt{d}$ の数パーセント以内に保たれるからである。

指数型の裾が手に入れば、もう 1 つの道具である**ユニオンバウンド**を使うことで**多数の**ペアを同時に制御できる。ユニオンバウンドとは、複数の事象が独立であるかどうかにかかわらず、そのうち少なくとも 1 つが起こる確率は個々の事象の確率の和以下になるという性質である。m 本の独立なランダム符号ベクトルをとろう。これらは $m(m-1)/2 < m^2/2$ 個の異なるペアを作り、任意の 1 ペアが $|\cos| \geq \varepsilon$ となる確率は高々 $2e^{-d\varepsilon^2/2}$ である。ユニオンバウンドにより、**いずれかの**ペアが ε を超える確率は高々 $m^2 e^{-d\varepsilon^2/2}$ となる。したがって、確率 $1 - \delta$ 以上ですべてのペアのコサインが ε 未満に収まるための条件は $m \leq \sqrt{\delta}\, e^{d\varepsilon^2/4}$ となる。高次元では、**指数関数的に**多くのほぼ直交するベクトルを $\mathbb{R}^d$ 内に詰め込むことができるのである。

具体的に $d = 784$、$\delta = 0.05$ を代入してみよう。この緩い上界でさえ、およそ 570 本の方向について全ペアで $|\cos| < 0.2$ となることを保証し、およそ 10^7 本の方向について全ペアで $|\cos| < 0.3$ となることを保証する。しかもこれらは全ペアにわたる最悪ケースの保証であり、

典型的な 1 ペアの $|\cos|$ はそれよりはるかに小さく、$\sqrt{2/(\pi d)} \approx 0.028$ 程度、すなわち標準偏差 $1/\sqrt{d}$ のガウス変数の絶対値の平均となる。

もちろん、ニューラルネットワークが学習した埋め込みは独立な確率ベクトルではないが、まさにそのためにこの計算が役に立つ。$d = 784$ において、0.3 というコサイン類似度は、独立なランダム方向が生み出す値より標準偏差（$1/\sqrt{784} = 0.036$）の 8 倍以上も上にある。したがって、ランダムな方向は、訓練済みの埋め込みに現れる真の幾何学的構造を測るための帰無モデル（null model）として機能する。

36.9.2.3 数値で見る薄い球殻

理論的な上界に含まれる定数は緩いため、実際の数値を直接測定してみよう。先ほどの接線近似の議論は $\|\mathbf{x}\|$ の標準偏差が $1/\sqrt{2} \approx 0.71$ 付近にあらゆる次元 d でとどまると予測し、ガウス近似は $|\cos|$ の平均を $\sqrt{2/(\pi d)}$ と予測する。以下では 10,000 組の標準ガウスベクトルのペアを $d = 2, 20, 200, 2000$ でサンプリングし、両方の予測を検証する。なお、1 つの行だけは予測と一致しないことに注目してほしい。

```
for d in (2, 20, 200, 2000):
    x, y = rng.standard_normal((2, 10000, d))
    nx = np.linalg.norm(x, axis=1)
    cos = (x * y).sum(1) / (nx * np.linalg.norm(y, axis=1))
    print(f'd = {d:4d}: mean ||x||/sqrt(d) = {(nx / np.sqrt(d)).mean():.3f}, '
          f'sd ||x|| = {nx.std():.2f}, mean |cos| = {np.abs(cos).mean():.4f}'
          f' vs {np.sqrt(2 / (np.pi * d)):.4f}')
```

```
d =    2: mean ||x||/sqrt(d) = 0.886, sd ||x|| = 0.65, mean |cos| = 0.6377 vs 0.5642
d =   20: mean ||x||/sqrt(d) = 0.986, sd ||x|| = 0.71, mean |cos| = 0.1827 vs 0.1784
d =  200: mean ||x||/sqrt(d) = 0.999, sd ||x|| = 0.70, mean |cos| = 0.0560 vs 0.0564
d = 2000: mean ||x||/sqrt(d) = 1.000, sd ||x|| = 0.71, mean |cos| = 0.0179 vs 0.0178
```

予測どおり、ノルムの標準偏差は 0.65 付近（$d = 2$）から始まり、それ以降はノルム自体が約 1.3 から 45 へと増大しても 0.7 のすぐ近くに落ち着く。漸近公式と一致しない唯一の行は $d = 2$ であり、そこでの $|\cos|$ の平均は約 0.64 である。平面内では 2 つの等方的なベクトルのなす角は一様分布に従うため $\mathbb{E}|\cos\theta| = 2/\pi$ となるのに対し、$\sqrt{2/(\pi d)}$ は高次元近似だからである。$d = 20$ 以降では、実測値（約 0.18、0.057、0.018）は公式と数パーセント以内で一致する。

さらに 2 つの問いを立てると、この幾何学がいっそう鮮明になる。第 1 に、典型的なノルムの 10% 以内、すなわち球殻 $(1 \pm 0.1)\sqrt{d}$ の内側に確率質量のどれだけの割合が収まるだろうか。第 2 に、ユークリッド距離は依然として近い点と遠い点を区別できるだろうか。この分離度を測るため、クエリ点の**コントラスト** $(d_{\max} - d_{\min})/d_{\min}$ を 200 個のランダムな点への距離にわたって計算し、100 個のクエリにわたるコントラストの中央値を報告する。

```
eps = 0.1
for d in (2, 20, 200, 2000):
    x = rng.standard_normal((10000, d))
    ratio = np.linalg.norm(x, axis=1) / np.sqrt(d)
    shell = np.mean(np.abs(ratio - 1) <= eps)          # mass in the 10% shell
    query, points = x[:100], x[100:300]                # 100 queries, 200 points
    dist = np.linalg.norm(query[:, None] - points[None], axis=2)  # (100, 200)
    contrast = (dist.max(1) - dist.min(1)) / dist.min(1)
    print(f'd = {d:4d}: mass in (1 ± 0.1) sqrt(d) = {shell:6.1%}, '
```

```
            f'median contrast = {np.median(contrast):.3f}')
```

```
d =    2: mass in (1 ± 0.1) sqrt(d) =  14.5%, median contrast = 30.417
d =   20: mass in (1 ± 0.1) sqrt(d) =  46.7%, median contrast = 1.179
d =  200: mass in (1 ± 0.1) sqrt(d) =  95.6%, median contrast = 0.270
d = 2000: mass in (1 ± 0.1) sqrt(d) = 100.0%, median contrast = 0.079
```

$d = 2$ では、10% の球殻は確率質量の約 15% しか捉えないが、$d = 20$ では約 47%、$d = 200$ では約 95% を捉える。この 95% という値は先ほどの接線近似による見積もりと一致する。すなわち、$d = 200$ では比 $\|\mathbf{x}\|/\sqrt{d}$ の標準偏差が $1/\sqrt{2d} = 0.05$ となるため、± 0.1 は平均の両側に標準偏差 2 つ分の幅を張るからである。$d = 2000$ になると、10,000 個のサンプルすべてがこの球殻の内側に落ちる。その一方で、距離コントラストの中央値は約 30（$d = 2$、実行ごとに約 25 から 40 の間で変動）から、約 1.2、0.27、そして最終的には 0.08 へと急激に縮小する。$d = 2000$ では、典型的なクエリに対し、200 個のランダムな点のうち最も遠い点であっても、最近傍の点より 10% 未満しか離れていない。

36.9.2.4 深層学習への帰結

深層学習における 3 つの標準的な実践は、この高次元幾何学に直接依拠している。

初期化：LeCun 初期化 (LeCun, Leon Bottou, et al. 1998) は $d \times d$ 重み行列の各成分を分散 $1/d$ で独立にサンプリングし、He 初期化 (He et al. 2015) は分散 $2/d$ を ReLU ネットワークに対して用いる（4.4 節）。これらのスケールの古典的な導出は、重みのランダムな抽出全体にわたる期待値のもとで行われる。たとえば LeCun のスケールでは、各活性化前の値 $\langle \mathbf{w}, \mathbf{x} \rangle$ は分散 $\|\mathbf{x}\|^2/d$ を持ち、各座標が単位分散を持つ入力に対してはノルムの集中によってこれが 1 の近くに保たれる。しかし実際には、無限個のネットワークの平均ではなく**ただ 1 つの**ネットワークを初期化して訓練する。その期待値が手元の 1 回の抽出を実際にいつよく記述するのかを教えてくれるのが、集中不等式である。

He 初期化された幅 d の単一の ReLU 層では、活性化値の二乗ノルムの比 $\rho = \|\mathbf{h}_k\|^2/\|\mathbf{h}_{k-1}\|^2$ は平均 1、分散 $5/d$ を持つ（演習問題 5）。したがって、1 つの層は二乗ノルムをおよそ $\sqrt{5/d}$、すなわち約 10%（$d = 512$ のとき）摂動させる。K 層にわたると、これらの乗法的な揺らぎは対数スケールで加算され、$\log(\|\mathbf{h}_K\|^2/\|\mathbf{h}_0\|^2)$ の分散はおよそ $5K/d$ になる。ゆえに、期待分散に基づくヒューリスティックが典型的なネットワークをよく記述するのは $K \ll d$ の範囲に限られる。$K = 50$、$d = 512$ では対数の分散は約 0.5 に達し、初期化された典型的なネットワークの二乗ノルムはその時点で上下いずれかの方向に 2 倍ほどずれてしまう。

距離と最近傍：独立な 2 つの標準ガウスサンプルに対し、その差は $\mathbf{x} - \mathbf{y} \sim \mathcal{N}(\mathbf{0}, 2\mathbf{I})$ を満たすため、ノルムの集中が $(\mathbf{x}-\mathbf{y})/\sqrt{2}$ にそのまま適用できる。すなわち、各ペア間の距離は $(1 \pm \varepsilon)\sqrt{2d}$ の範囲に、高々 $2e^{-d\varepsilon^2/8}$ の例外確率を除いて収まる。方向に対して用いたのと同じユニオンバウンドを適用すれば、n 個の点の間のペアワイズ距離は**すべて**同時にその狭い帯域内に収まることになり、そのための条件 $d \geq (8/\varepsilon^2)\log(n^2/\delta)$ の閾値は n に対して対数的にしか増大しない。あらゆる距離がほぼ等しくなると、先ほどのコントラスト実験が示したとおり、最近傍と最遠傍の差は相対的にごくわずかなマージンしか持たなくなる。実際に学習された埋め込みが意味のある距離コントラストを保っているのは、実データがはるかに低次元の構造の近くに分布しているからにほかならない。裏を返せば、この同じ計算から強力な道具が得られる。k 次元へのランダム射影は、元の空間の次元によらず、n 個の点すべてのペアワイズ距離を k が $\log n$ のオーダーに達した時点で保存するのである（演習問題 6）。

潜在コード：潜在コード $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ 上で訓練された生成デコーダは、ノルムが $\sqrt{d}$ に近い入力ばかりを目にする。事前分布の平均（ノルム 0）も、独立な 2 つの潜在コードを結ぶ直線の弦の中点（ノルムが $\sqrt{d/2}$ 付近）も、デコーダが一度も訓練されていない空っぽの内部領域の奥深

くに位置してしまう。このため実務では、Figure 36.45 に示すように、潜在コード間の補間を直線ではなく球面上の弧に沿って行うのが一般的である（演習問題 2）。

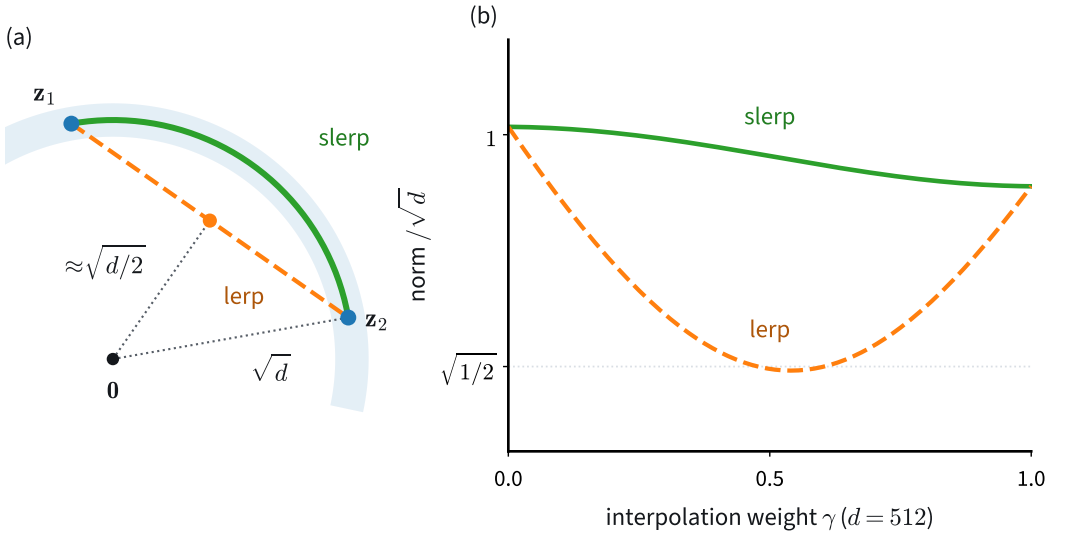


Figure 36.45: 潜在コード $\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ 間を補間する 2 つの方法。(a) 両コードとも半径 $\sqrt{d}$ の薄い球殻（網掛け）内にあり、ほぼ直交している。直線の弦（lerp）は内部を横切り、その中点のノルムは約 $\sqrt{d/2}$ まで落ち込むが、球面上の弧（slerp）は球殻内にとどまる。(b) 各経路に沿ったノルムを $\sqrt{d}$ で割った値（$d = 512$ におけるランダムな 1 ペア）。

最後に、1 つの重要な制約が残っている。本節で導いたすべての上界が保証するのは、データが抽出される**前**に固定された単一の量である。これに対し、学習アルゴリズムはデータを観察した**後**で多数の候補の中から最良のモデルを選び出すが、多数の公平なコインの中から最も成績のよかったものを選べば、それは必ず偏っているように見える。36.10 節では、このようなモデル選択を織り込むために集中不等式をどのように拡張するかを示す。

36.9.3 まとめ

n 個の独立な項の平均は、有限標本サイズ n においてその期待値からどれだけ離れうるのだろうか。チェビシェフの不等式が与える裾確率は $1/n$ の速さでしか減衰せず、めったに失敗しない保証が必要なときには遅すぎる。チェルノフ法はマルコフの不等式を $e^{\lambda X}$ に適用することで、独立和のモーメント母関数を積へと変換したうえで λ について最適化する。これをヘフディングの補題と組み合わせると、指数型の裾 $2e^{-2nt^2}$ が $[0,1]$ 内の損失に対して得られ、半幅 $\sqrt{\log(2/\delta)/(2n)}$ の有限標本信頼区間が導かれる。さらに、値域に比べて分散が小さいときには、ベルンシュタインの不等式がこの区間を大幅に引き締める。同じ考え方を確率ベクトルの d 個の座標に適用すると、標準ガウス分布が半径 $\sqrt{d}$、厚さ約 $1/\sqrt{2}$ の薄い球殻上に集中すること、そして指数関数的に多くのランダムな方向が互いにほぼ直交することが示される。以上の結果から、テストセットのサイズをどう設計すべきか、初期化のヒューリスティックが訓練される単一のネットワークに対していつ成り立つか、そして高次元の埋め込みにおけるコサイン類似度や距離をどう解釈すべきかが明らかになる。

36.9.4 演習問題

1. **準ガウスなランダム符号：**ランダム符号 ξ は値 -1 と $+1$ をそれぞれ確率 $\tfrac{1}{2}$ でとるため、$\mathbb{E}[e^{\lambda \xi}] = \cosh \lambda \le e^{\lambda^2/2}$ を満たす。
   1. チェルノフ法 Equation 36.112 を用い、平均 $\bar{\xi}$（n 個の独立なランダム符号の標本平均）が $P(|\bar{\xi}| \ge t) \le 2e^{-nt^2/2}$ を満たすことを導け。
   2. ヘフディングの補題 Equation 36.113 を $[a,b] = [-1,1]$ で適用しても ξ の分散プロキシが得られる。これを上界 $\cosh\lambda \le e^{\lambda^2/2}$ と比較せよ。さらに $\log\cosh\lambda = \lambda^2/2 - \lambda^4/12 + \cdots$ を $\lambda = 0$ の周りで展開し、すべての有界な確率変数に対して成り立つ限り、$\tfrac{1}{8}$ より小さいいかなる定数でも Equation 36.113 の $\tfrac{1}{8}$ を置き換えられない理由を説明せよ。
2. **潜在空間の補間：**生成モデルが潜在コード $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$（$d = 512$）をデコードするとする。独立な 2 つの潜在コードを $\mathbf{z}_1, \mathbf{z}_2$ とし、それらの線形補間を $\mathbf{z}(\gamma) = (1-\gamma)\mathbf{z}_1 + \gamma \mathbf{z}_2$（$\gamma \in [0,1]$）とする。
   1. $\mathbf{z}(\gamma)$ の分布と $\|\mathbf{z}(\gamma)\|$ の近似値を求めよ。ノルムが最小になるのはどの γ においてか。
   2. Equation 36.121 を用いて、中点 $\mathbf{z}(\tfrac{1}{2})$ が少なくとも $0.9\sqrt{d}$ 以上のノルムを持ち、典型的なコードが存在する球殻 $(1 \pm 0.1)\sqrt{d}$ に到達する確率の上界を求めよ。次に、厳密な確率 $P(\chi^2_{512} \ge 2 \cdot 0.81 \cdot 512)$（自由度 512 のカイ二乗変数に対する上側確率、`scipy.stats.chi2.sf`）を計算せよ。上界は何桁ほど緩いか。
   3. 球面補間 (White 2016) は、弦を次の弧に置き換える。
   $$\tilde{\mathbf{z}}(\gamma) = \frac{\sin((1-\gamma)\theta)}{\sin\theta}\,\mathbf{z}_1 + \frac{\sin(\gamma\theta)}{\sin\theta}\,\mathbf{z}_2,$$
   ここで、θ は $\mathbf{z}_1$ と $\mathbf{z}_2$ のなす角である。$\|\tilde{\mathbf{z}}(\gamma)\| = r$ がすべての γ に対して成り立つことを、$\|\mathbf{z}_1\| = \|\mathbf{z}_2\| = r$ のもとで示せ。また、θ は $d = 512$ のときおよそどのような値をとるか。それはなぜか。
3. [code] **コインに対するチェルノフ上界：**●$\hat{p}$ を、n 回の独立なコイン投げ（表が出る確率がバイアス p のコイン）における表の割合とし、$a > p$ とする。
   1. チェルノフ法 Equation 36.112 を $n\hat{p}$（そのモーメント母関数は $(1 - p + pe^{\lambda})^n$ である）に適用せよ。λ について最適化し、$P(\hat{p} \ge a) \le e^{-n\,D_{\mathrm{KL}}(a\|p)}$ を示せ。ただし $D_{\mathrm{KL}}(a\|p) = a\log\frac{a}{p} + (1-a)\log\frac{1-a}{1-p}$ とする。
   2. $D_{\mathrm{KL}}(a\|p) \ge 2(a-p)^2$ を、a に関する二階導関数を調べることで示し、(a) の上界がヘフディングの上界の片側より決して弱くならないことを確かめよ。さらに $p = \tfrac{1}{2}$、$a = 0.6$ に対して両方の指数を評価せよ。
   3. 両側上界 $2e^{-n\,D_{\mathrm{KL}}(0.6\|0.5)}$ を 36.9.1.5 節の表に新たな列として $n \in \{10, 100, 1000, 5000\}$ について追加せよ。厳密な裾確率のこの上界に対する比は、依然として n とともに減少する。この比は n に対してどのようにスケールするか。また、そのスケーリングはチェルノフ法が失う因子について何を物語っているか。
4. **カイ二乗分布の裾：**●Equation 36.121 の証明は、Equation 36.122 による $x^2 - 1$（$x \sim \mathcal{N}(0,1)$）の上界に基づいているが、本文の証明では導出なしにこれを用いた。
   1. $\mathbb{E}[e^{\lambda(x^2-1)}] = e^{-\lambda}/\sqrt{1-2\lambda}$ が $\lambda < \tfrac{1}{2}$ で成り立つことを、ガウス積分において平方完成することで示せ。
   2. $\log \mathbb{E}[e^{\lambda(x^2-1)}]$ を 2λ のべき級数に展開し、Equation 36.122 を証明せよ。この級数は 2 次の項から始まる。
   3. Equation 36.121 の証明では $\lambda = \varepsilon/4$ とおいたが、これが許容されるのは $\varepsilon \le 1$ のときに限られる。$\varepsilon > 1$ に対しては $\lambda = \tfrac{1}{4}$ とおき、$P(\|\mathbf{x}\|^2 \ge (1+\varepsilon)d)$ を上から評価せよ。得られる結果は $e^{-c\varepsilon^2}$ と $e^{-c\varepsilon}$ のどちらの形で ε に関して減衰するか。また、それは Equation 36.120 のどちらの領域に属するか。

5. [code] **深さにわたる活性化値のノルム：●**バイアスを持たない ReLU ネットワーク（幅 d、深さ K）$\mathbf{h}_k = \mathrm{ReLU}(\mathbf{W}_k \mathbf{h}_{k-1})$ を考え、その重みは $\mathcal{N}(0, 2/d)$ からの独立なサンプルとする（He 初期化、4.4 節）。$\rho_k = \|\mathbf{h}_k\|^2 / \|\mathbf{h}_{k-1}\|^2$ とおく。このようなノルムが深さと幅によってどう変動するかは、Hanin and Nica (2020) によって解析された。
   1. $\mathbf{h}_{k-1} \neq \mathbf{0}$ が与えられたとき、比 ρ_k は $\frac{2}{d}\sum_{j=1}^{d} \mathrm{ReLU}(g_j)^2$（ただし $g_j \sim \mathcal{N}(0,1)$ は独立）と同じ分布に従うことを、$\mathbf{h}_{k-1}$ の方向によらず示せ。さらに $\mathbb{E}[\rho_k] = 1$ および $\mathrm{Var}(\rho_k) = 5/d$ を、$\mathbb{E}[g^2] = 1$ と $\mathbb{E}[g^4] = 3$ を用いて導け。
   2. $\log(\|\mathbf{h}_K\|^2 / \|\mathbf{h}_0\|^2)$ を K 個の独立な項の和として表せ（各層で $\rho_k = 0$ となる確率は 2^{-d} であるが、これは無視せよ）。$\log \rho$ を 1 の周りで 2 次まで展開し、この和の平均と分散を K/d を用いて予測せよ。
   3. 1,000 個のネットワークを各幅 $d \in \{32, 128\}$ と各深さ $K \in \{1, 10, 50\}$ について、それぞれ新しい標準ガウス入力を与えてシミュレーションせよ。$\log(\|\mathbf{h}_K\|^2/\|\mathbf{h}_0\|^2)$ の平均と分散を (b) と比較し、さらに比 $\|\mathbf{h}_K\|^2/\|\mathbf{h}_0\|^2$ 自体の平均と中央値を報告せよ。4.4 節の分散計算によれば、あらゆる深さで $\mathbb{E}[\|\mathbf{h}_K\|^2/\|\mathbf{h}_0\|^2] = 1$ となる。$K = 50$、$d = 32$ での実行を数回繰り返せ。1,000 個のネットワークにわたる標本平均が 1 の周りでこれほど大きくばらつくのはなぜか。また、そこではなぜ期待値が典型的なネットワークを記述しないのか。
6. [code] **距離を保存するランダム射影：●**$\mathbf{A} \in \mathbb{R}^{k \times d}$ を、独立な $\mathcal{N}(0, 1/k)$ の成分を持つ行列とする。
   1. 固定された $\mathbf{v} \neq \mathbf{0}$ に対し、比 $\|\mathbf{A}\mathbf{v}\|^2/\|\mathbf{v}\|^2$ が $\|\mathbf{g}\|^2/k$（$\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$）と同じ分布に従い、したがって Equation 36.121 より $P(|\|\mathbf{A}\mathbf{v}\|/\|\mathbf{v}\| - 1| \geq \varepsilon) \leq 2e^{-k\varepsilon^2/8}$ が成り立つことを示せ。ヒント：$\mathbf{v}$ が第 1 座標軸に沿うように回転させるとよい。
   2. n 個の点に対し、(a) を $n(n-1)/2$ 個の差ベクトルに適用してユニオンバウンドをとれ。すべてのペアワイズ距離が係数 $1 \pm \varepsilon$ の範囲内に少なくとも確率 $1 - \delta$ で保存されるための条件が $k \geq (8/\varepsilon^2)\log(n^2/\delta)$（$d$ に依存しない上界）となることを示せ。これがジョンソン・リンデンシュトラウスの補題である。$n = 1{,}000$、$\varepsilon = 0.3$、$\delta = 0.05$ に対してこの値を評価せよ。
   3. 1,000 個の標準ガウス点（$d = 2{,}000$）上でこれを検証せよ。$k \in \{50, 200, 800\}$ 次元へ射影し、全ペアにわたる最大の歪み $|\|\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)\|/\|\mathbf{x}_i - \mathbf{x}_j\| - 1|$ を、典型的な揺らぎ $1/\sqrt{2k}$ と併せて報告せよ。ペアワイズ距離はグラム行列から $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\,\mathbf{x}_i^\top \mathbf{x}_j$ により計算し、$1{,}000 \times 1{,}000 \times 2{,}000$ 配列のブロードキャストは使わないこと。得られた 2 つの数値の比をペアの個数から説明せよ。すでに $k = 200$ の時点で最大の歪みは $\varepsilon = 0.3$ を下回る一方、(b) ではるかに大きな k が要求された。上界のどの部分が緩いのか、定数 8 か、それとも $\log n$ か。
7. **ベルンシュタインの不等式：●**X を中心化された確率変数とし、$|X| \leq M$ および $\mathrm{Var}(X) = \sigma^2$ を満たすとする。
   1. $\mathbb{E}[X^k] \leq \sigma^2 M^{k-2}$ が $k \geq 2$ で成り立つことを示し、そこから $\mathbb{E}[e^{\lambda X}] \leq 1 + \sigma^2(e^{\lambda M} - 1 - \lambda M)/M^2$ が $\lambda > 0$ で成り立つことを導け。
   2. $e^u - 1 - u \leq u^2/(2(1 - u/3))$ が $0 \leq u < 3$ で成り立つことを、$u^k/k!$ と $(u^2/2)(u/3)^{k-2}$ の比較、すなわち $k! \geq 2 \cdot 3^{k-2}$ を用いることで示せ。さらに $1 + y \leq e^y$ と合わせて、$\mathbb{E}[e^{\lambda X}] \leq \exp(\lambda^2\sigma^2/(2(1 - \lambda M/3)))$ が $0 < \lambda < 3/M$ で成り立つことを導け。
   3. このような確率変数 n 個の平均に、$\lambda = t/c$（ただし $c = \sigma^2 + Mt/3$）としてチェルノフ法を適用せよ。$\lambda < 3/M$ および $1 - \lambda M/3 = \sigma^2/c$ を確認し、指数部がちょうど $-nt^2/(2c)$ になることを導いて Equation 36.117 を証明せよ。
   4. (b) により X が Equation 36.119 の意味で $\nu^2 = 2\sigma^2$、$\alpha = 2M/3$ の準指数になることを示せ。
   5. $n = 10{,}000$、$\delta = 0.05$ のとき、最悪ケースの分散 $\sigma^2 = \frac{1}{4}$（0/1 の結果における最

大分散）ではベルンシュタインの上界がヘフディングの上界より弱くなることを示せ。分散がどの値を下回るとベルンシュタインの上界が勝るか。また、それはどの範囲の誤り率に対応するか。

36.10 一様収束と汎化

前節では、データから計算した平均にエラーバーを付ける方法を示した。ホールドアウトされた10,000 枚の画像で評価した分類器は、95% の信頼度で、テスト誤差から約 1.4 パーセントポイント以内に真の誤差を持つ。ただし、この保証は見落としやすい条件に依拠している。分類器はテスト画像を抽出する**前**に固定されていなければならないのである。実際には、そのような手順をとることはめったにない。多数の設定を訓練し、検証スコアを調べ、最良のものを残す。報告されるスコアは競争の勝者に属するものであり、その勝者は部分的には標本上で最も運が良かった設定にすぎない。

第 3.5 章では、この問題とともに Vapnik と Chervonenkis による解決策を導入した (Vapnik and Chervonenkis 1971)。すなわち、すべての候補関数に対して同時に成り立つ保証を要求することで、学習者がどの関数を選んでも自動的にカバーされるようにするのである。候補のリストが有限であれば、上界の幅は候補数の対数に応じて広がるだけで済む。ここではその有限クラスの結果を簡単に振り返り、選び取れる運がどれほど大きいかを示す具体的な実験と組み合わせよう。1,000 枚の公平なコインをそれぞれ 100 回投げ、表が最も多く出たコインを残すのである。そのコインは何回の表を示すだろうか。

しかし本節の核心は、実際に訓練する無限の関数クラスにある。こうしたクラスに対しては、純粋なコイン投げで決めたラベルにどれほどよく適合できるかによってクラスの豊かさを測る、後の時代のアイデアである**ラデマッハ複雑度**（Rademacher complexity）を用いる。この指標は、線形モデルの容量がパラメータ数ではなく重みノルムに依存する理由を説明するとともに、一様収束（uniform convergence）が限界に達する場所も明らかにする。それは本編で扱った深層ネットワークのように、ランダムなラベルに適合できるほど柔軟なモデルにおいてである。最後のパートでは、メカニズムのあらゆる要素を閉じた形で計算できるほど単純なモデルを用いて、4.5.2 節の二重降下（double descent）の実験を再訪する。

```
rng = np.random.default_rng()
```

36.10.1 一様収束

固定されたモデルをテストセット上で評価することと、学習者を訓練することの違いは、関数の選択と標本の抽出という 2 つの事象の順序だけにある。ここでは、その順序を逆にすることでスコアがどれほど歪むか、そして選択が標本に依存するようになった途端にどれほど多くのデータが必要になるかを定量化する。

36.10.1.1 多数のコインから選ぶ最良のコイン

まず、本節を通して用いる記法で単一関数に対する保証を述べよう。標本 $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ は未知の分布からの n 個の独立な抽出からなり、損失 ℓ は $[0,1]$ に値をとって各予測を評価する。たとえば 0-1 損失は、誤りに対して 1、それ以外に対して 0 をとる。

関数 f の**経験リスク**（empirical risk）とは標本上の平均損失 $\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{x}_i), y_i)$ であり、**リスク**（risk）とは新たな抽出に対する期待損失 $R(f) = \mathbb{E}[\ell(f(\mathbf{x}), y)]$ である（0-1 損失の場合、これらは本編における経験誤差 $\epsilon_S(f)$ と母集団誤差 $\epsilon(f)$ にあたる）。f が S の抽出前に固定されていれば、$\hat{R}(f)$ は n 個の $[0,1]$ 内の独立な項（平均 $R(f)$）の平均となる。したがっ

て、ヘフディングの不等式（Hoeffding's inequality、36.9 節）により $P(|\hat{R}(f) - R(f)| \geq t)$ は $2e^{-2nt^2}$ で上から抑えられる。この失敗確率を $\delta = 0.05$ とおいて t について解くと、本編でヘフディング半径（Hoeffding radius）と呼んだ半幅 $\sqrt{\log(2/\delta)/(2n)}$ が得られる。$n = 10{,}000$ のとき、この値は 0.0136、すなわち冒頭で述べた 1.4 パーセントポイントとなる。

学習者は事象の順序を逆にする。関数のクラス $\mathcal{F}$ を探索し、予測器 $\hat{f}$ として観測標本上で経験リスク $\hat{R}$ が最小になるものを返す。各 $\hat{R}(f)$ は真のリスク $R(f)$ の周りで変動するため、学習者はたまたま都合のよいランダムな変動を示した関数を自然に選び取ってしまう。$\hat{R}(\hat{f})$ は多数の変動する量の最小値であるため、固定された各 f に対して個別に成り立つ不等式ではもはやカバーできない。

コイン投げの実験を行えば、この選択効果がどれほど大きくなり得るかがよくわかる。$k = 1{,}000$ 枚の公平なコインをそれぞれ $m = 100$ 回投げるとしよう。事前に指定した 1 枚のコインについては、$\delta = 0.05$ でのヘフディングの不等式により表の割合のずれとして $\sqrt{\log 40/200} = 0.136$ が許容されるため、そのコインは少なくとも 95% の確率で表が 37〜63 回の範囲に収まる（厳密な確率は 99.3% である）。しかし、1,000 枚のコインのうち最良のものにはそのような保証はない。この実験を 2,000 回繰り返し、勝者となったコインがどれほど大きく外れ、1 枚のコインに対する帯域をどれほどの頻度で逸脱するかを測定してみよう。

```
k, m, repeats = 1000, 100, 2000          # coins, flips per coin, experiments
best = rng.binomial(m, 0.5, size=(repeats, k)).max(axis=1) / m
one = np.sqrt(np.log(2 / 0.05) / (2 * m))       # Hoeffding, one fixed coin
many = np.sqrt(np.log(2 * k / 0.05) / (2 * m))  # union bound over k coins
print(f'best of {k} coins: median {np.median(best):.2f}, 90% of experiments '
      f'in [{np.percentile(best, 5):.2f}, {np.percentile(best, 95):.2f}]')
print(f'95% half-width: one coin {one:.3f}, all {k} coins {many:.3f}')
print(f'experiments in which the best coin leaves the one-coin band: '
      f'{np.mean(best > 0.5 + one):.0%}')
```

```
best of 1000 coins: median 0.66, 90% of experiments in [0.64, 0.69]
95% half-width: one coin 0.136, all 1000 coins 0.230
experiments in which the best coin leaves the one-coin band: 97%
```

典型的な実験では、最良のコインは 66 回の表を示し、10 回中 9 回の実験で 64〜69 回の範囲に入る。実行の 96%〜97% で単一コインの帯域を逸脱するため、事前に指定したどのコインに対しても成り立つ保証が、事後に選んだコインに対してはほぼ毎回破綻することになる。どのコインにも偏りはない。勝者が 16 パーセントポイントも外れるのは、ひとえに 1,000 個のランダムな変動を調べて最大のものを選んだからである。

Figure 36.46 を見れば、その理由は一目瞭然である。個々のコインが帯域の上端を超える確率は 0.0033 であり、単独ではまれな事象にすぎない。しかし 1,000 枚のコインの中では、約 3.3 枚が上端を超えると期待される。1 枚のコインではほとんど到達しない裾の領域も、1,000 枚のコインを投げれば十分に埋まるのである。

検証セット上でのモデル選択は、ラベルを貼り替えただけのまったく同じ実験である。各設定が 1 枚のコインであり、各検証例が 1 回のコイン投げにあたる。独立な公平コインとは異なり、検証される各設定は同じデータ例を共有し、真の誤差もそれぞれ異なるが、次に導出する上界は候補間の独立性も平均が等しいことも必要としない。

では、選択された関数をもカバーする保証はどうすれば得られるだろうか。いま、クラス全体にわたる最大のずれを考える。

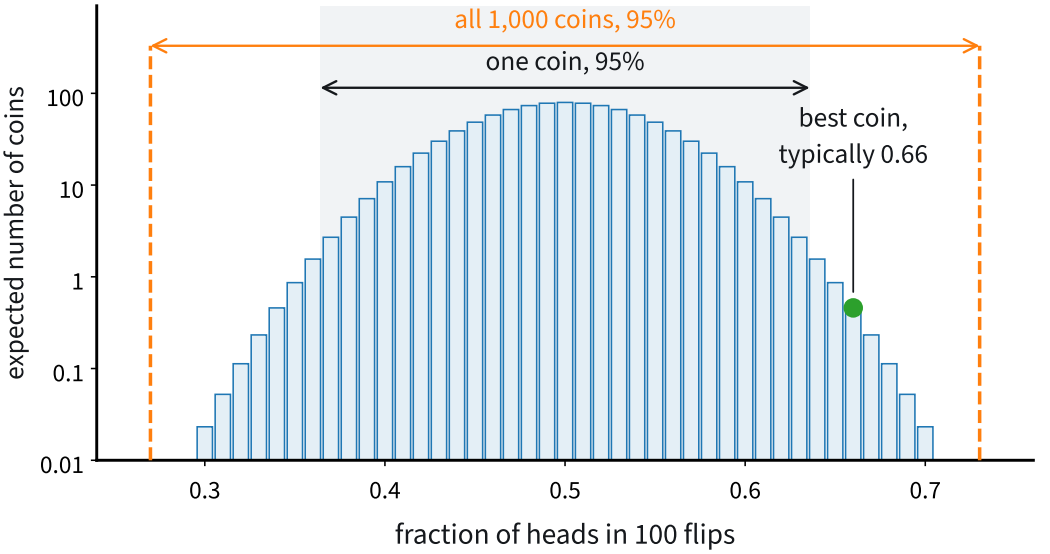


Figure 36.46: それぞれ 100 回投げた 1,000 枚の公平なコインにおける表の割合の期待ヒストグラム（対数スケール表示）。単一コインに対する 95% ヘフディング帯域はコインの 99% をカバーするが、その上端より上に約 3.3 枚のコインが入ると期待され、最良のコインは典型的に 0.66 付近に着地する。1,000 枚すべてのコインに対するより広いユニオンバウンド帯域は、最良のコインもカバーする。

$$\sup_{f \in \mathcal{F}} \left| \hat{R}(f) - R(f) \right|,$$

この上限が高々 t であると仮定する。すると、すべての関数の経験リスクはその真のリスクから t 以内に収まり、とりわけ学習者が選んだ関数についても同じことが成り立つ。この上限に対する上界は**一様収束**上界と呼ばれる (Vapnik and Chervonenkis 1971)。データが到着する前にすべての候補に対して保証が成り立っていれば、どの候補を選んでもすでにカバーされているため、データを見た後の選択が安全になるのである（Figure 36.47）。

36.10.1.2 有限クラスとユニオンバウンド

最も単純な設定は、事前に指定されたハイパーパラメータ設定のグリッドのように、あらかじめ固定された有限個の候補リストである。この場合の結果は本編の第 3.5 章ですでに述べており、その証明はわずか 2 行で終わる。ヘフディングの不等式により、固定された単一の f が t 以上ずれる確率は $2e^{-2nt^2}$ で抑えられる。すべての $|\mathcal{F}|$ 個の候補にわたってユニオンバウンド（36.9 節）を適用すると、それらの**いずれか**が t 以上ずれる確率は $2|\mathcal{F}|e^{-2nt^2}$ で抑えられる。この全体の失敗確率を δ とおいて t について解けば、最初の一様収束定理が得られる。すなわち、$[0,1]$ に値をとる損失とサイズ n の独立同分布（i.i.d.）標本に対し、少なくとも $1-\delta$ の確率で次が成り立つ。

$$\left| \hat{R}(f) - R(f) \right| \le \sqrt{\frac{\log(2|\mathcal{F}|/\delta)}{2n}} \qquad \text{simultaneously for every } f \in \mathcal{F}. \tag{36.124}$$

固定された単一の関数を評価する場合と比べると、$|\mathcal{F}|$ 個の候補から選択することは平方根の中

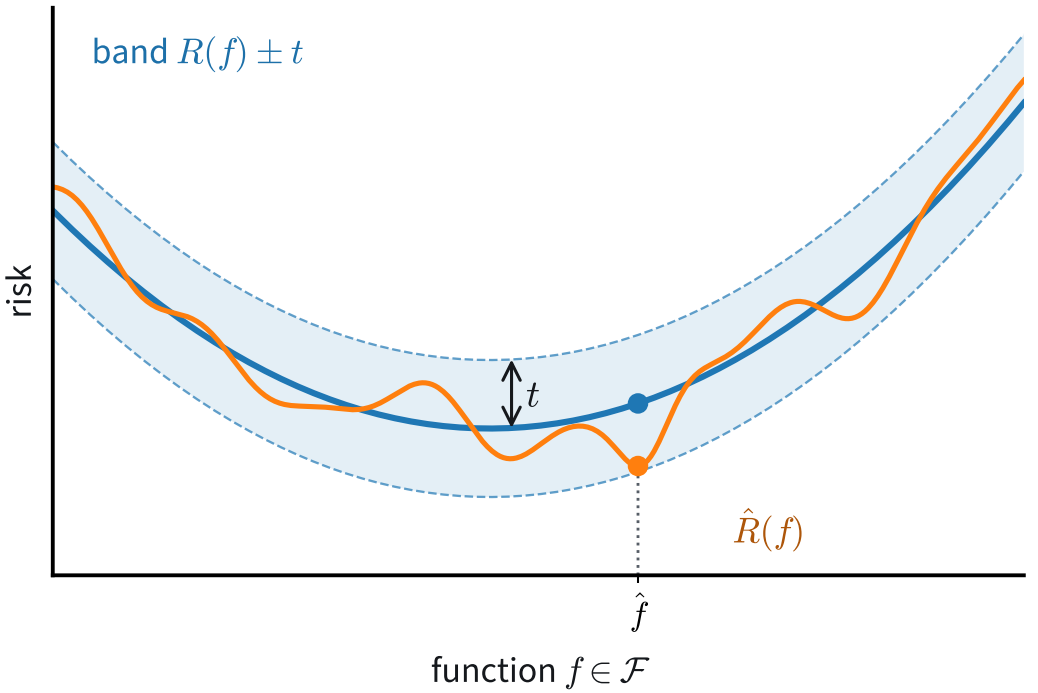


Figure 36.47: 一様収束。経験リスク $\hat{R}(f)$（オレンジ）は、クラス内のすべての関数に対して同時に、ずれ t 以内の範囲で真のリスク $R(f)$（青）の周囲に収まる。学習者は経験リスクの最小点 $\hat{f}$ を選ぶが、これは真のリスクより下側に落ち込んだ谷に位置している。帯域はその関数もカバーするため、$R(\hat{f}) \leq \hat{R}(\hat{f}) + t$ が成り立つ。

に $\log|\mathcal{F}|$ を加えるだけであり、それ以外のペナルティはない。

先ほどのコインの実験で $|\mathcal{F}| = 1{,}000$、$n = 100$ とおくと半幅は $\sqrt{\log(2{,}000/0.05)/200} = 0.230$ となり、少なくとも 95% の確率で、勝者のコインを含むすべてのコインが表 27〜73 回の範囲に収まる。確かにこの上界は保守的である。いずれかのコインが表 27〜73 回の外に出る真の確率は約 0.17% であり、許容予算の 5% を大きく下回る。この 30 倍近い緩さのほぼすべては単一コインに対するヘフディングの不等式に由来している。1 枚のコインがこの範囲を外れる真の確率が 1.7×10^{-6} であるのに対し、ヘフディングの不等式は 5×10^{-5} まで許容するためであり、まれな独立事象に対するユニオンバウンド自体はほぼタイトである。保守的であれ何であれ、単一コインの帯域が妥当でなかったのに対し、この帯域は妥当な保証を与えている。

10,000 例からなる検証セットについては、本編で半幅が単一関数で 0.014、100 万個の関数で 0.030 となることを記録した。Equation 36.124 を標本サイズ n について解くと、探索のコストがいっそう明確になる。

$$n = \frac{\log(2/\delta) + \log|\mathcal{F}|}{2t^2}.$$

$\log|\mathcal{F}|$ が 1 増えるごとに、さらに $1/(2t^2)$ 個のデータ例が必要になる。許容誤差 $t = 0.01$ のもとでは、候補モデルの数を 2 倍にするコストは追加の検証例 $\log 2/(2 \cdot 10^{-4}) \approx 3{,}500$ 個に相当する。

この対数依存性は、ヘフディングの指数的な裾がもたらす直接の恩恵である。仮にチェビシェフの不等式（Chebyshev's inequality）を代わりに用いると（$[0,1]$ に値をとる損失の分散は高々 $\tfrac{1}{4}$ であることに注意する）、同じユニオンバウンドから得られるのは $t = \sqrt{|\mathcal{F}|/(4n\delta)}$ となる。2,000 個の設定を 10,000 例上で $\delta = 0.05$ として評価する場合、チェビシェフの不等式は $t = 1$ となって損失の値域全体に広がり何も教えてくれないのに対し、ヘフディングの不等式は $t = 0.024$ を与える。

証明は最初のステップで 1 つの決定的な仮定に依拠している。候補クラス $\mathcal{F}$ が標本の抽出前に固定されていなければならないという点である。公開ベンチマークのリーダーボードでは、過去の提出のテストスコアを調べた後に新しい提出が選ばれるたび、この仮定が破られる (Dwork et al. 2015)。ひとたび候補がホールドアウトセットに適応するようになると、誤差保証はオーダー $\sqrt{\log k/n}$（固定された k 個の候補の場合）から、オーダー $\sqrt{k/n}$（ランダムな推測器 k 個から構成した適応的多数決の場合）へと劣化する（演習問題 5）。

36.10.2 ラデマッハ複雑度

実際に訓練する関数クラスは無限である。すべての重みベクトル $\mathbf{w} \in \mathbb{R}^d$ がそれぞれ異なる線形分類器を定めるため、無限個の事象にわたるユニオンバウンドは空虚なものになる。古典的な学習理論では、クラスを有限個の代表元で近似するか（被覆数、covering number）、あるいは n 個の点上で実現できる相異なるラベル付けの個数を数えるか（VC 次元、第 3.5 章で概説）によってこの問題を回避する。ここではより直接的なアプローチをとり (Bartlett and Mendelson 2002)、実務家なら誰でも検証できる操作的な問いを立てる。すなわち、そのクラスは純粋なノイズにどれほどよく適合できるかという問いである。

本節の中心となる 2 つの結果はこのパートに収められている。一様収束定理 Equation 36.126 と、ノルム有界な線形モデルに対する複雑度上界 $Br/\sqrt{n}$（Equation 36.128）である。唯一技巧的なステップは、前者の証明における対称化（symmetrization）の議論である。

36.10.2.1 ランダム符号との相関

ある関数クラスが信号をまったく含まないラベルに適合できるのであれば、実際のラベルに含まれるノイズにも同じくらい容易に適合できてしまい、その経験リスクから真のリスクについてわかることはほとんどなくなる。クラスがノイズを暗記する容量を調べるため、架空のランダムなラベルを与えてみよう。$\varepsilon_1, \dots, \varepsilon_n$ を独立な公平ランダム符号とし、それぞれが値 $+1$ または -1 を確率 $\tfrac{1}{2}$ でとるものとする。これらは**ラデマッハ変数**（Rademacher variables）と呼ばれる。与えられた符号ベクトル ε に対し、平均 $\frac{1}{n}\sum_i \varepsilon_i f(\mathbf{x}_i)$ は f が標本上でそれらの符号とどれほど相関するかを測り、$f \in \mathcal{F}$ にわたる上限はそのクラスが達成できる最良の相関を測る。

定義。実数値関数のクラス $\mathcal{F}$ の標本点 $S = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ 上における**経験ラデマッハ複雑度**（empirical Rademacher complexity）とは、そのクラスがランダム符号との間で達成する最良の相関の期待値であり、次式で与えられる。

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_{\varepsilon}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i \, f(\mathbf{x}_i) \right], \tag{36.125}$$

また、その**ラデマッハ複雑度** $\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_S[\widehat{\mathfrak{R}}_S(\mathcal{F})]$ とは、サイズ n の無作為標本にわたるこの量の平均である。

任意の符号パターンに一致できるクラスは高いスコアを示し、自由度のほとんどないクラスはゼロに近いスコアとなる。厳密な上限は関数の値域に依存する。すべての $f \in \mathcal{F}$ が $[-1,1]$ に値をとるなら、どの相関も高々 1 であり、$[-1,1]$ への**すべて**の関数からなるクラスは、相異なる点上で 1 を達成する（任意の符号パターンに対して $f(\mathbf{x}_i) = \varepsilon_i$ とおけばよい）。$[0,1]$ に値をとるクラスの場合、とり得る最大の複雑度は $\tfrac{1}{2}$ である。最良の選択は $f(\mathbf{x}_i) = 1$ を $\varepsilon_i = +1$ の箇所でとり、それ以外では 0 とおくことであり、プラス符号の割合がスコアとなってその平均は $\tfrac{1}{2}$ になる。後ほど調べる線形関数のような非有界なクラスにはこのような天井はない。標本上のあらゆる符号パターンを補間できるなら、その複雑度は少なくとも 1 になり、さらに大きくなることもある。

数値の感覚をつかむため、最も単純な非自明クラスとして 2 つの定数関数 $\{+1, -1\}$ を考えよう。符号ベクトルとの最良の相関は $\max(\bar{\varepsilon}, -\bar{\varepsilon}) = |\bar{\varepsilon}|$ となる。ただし $\bar{\varepsilon}$ は符号の標本平均である。$n = 2$ のとき、等確率で生じる 4 つのパターン $(+,+), (+,-), (-,+), (-,-)$ に対して $|\bar{\varepsilon}| = 1, 0, 0, 1$ となるため、$\widehat{\mathfrak{R}}_S = \tfrac{1}{2}$ である。$n = 4$ では、符号の和は ± 4 が確率 $\tfrac{2}{16}$、± 2 が確率 $\tfrac{8}{16}$、それ以外では 0 となるので、$\widehat{\mathfrak{R}}_S = \tfrac{3}{8}$ が得られる。大きな n に対しては、$\bar{\varepsilon}$ は平均 0、分散 $1/n$ の正規分布に近似的に従うため、$\widehat{\mathfrak{R}}_S \approx \sqrt{2/(\pi n)}$ となる。$n = 100$ では厳密値が 0.0796 であるのに対し、正規近似からは 0.0798 が得られる。わずか 2 つの関数しかもたないクラスであっても、ノイズとレート $1/\sqrt{n}$ で相関する。n 個のランダム符号の平均がそのスケールで変動するからにほかならない。対照的に、単一要素クラス（singleton class）$\{f\}$ の複雑度は 0 である。選ぶ余地がないため、$\mathbb{E}[\frac{1}{n}\sum_i \varepsilon_i f(\mathbf{x}_i)] = 0$ となるからである。

線形クラスの場合、最良の相関には明快な幾何学的解釈がある。Figure 36.48 に先取りして示すように、それは入力ベクトルの符号付き和の長さを n で割ったものにほかならない。このことは後ほど 36.10.2.3 節で証明する。

$[-1,1]$ に値をとる有限クラスは**マサールの補題**（Massart's lemma）$\widehat{\mathfrak{R}}_S(\mathcal{F}) \le \sqrt{2\log|\mathcal{F}|/n}$ を満たすため（演習問題 2）、その複雑度は関数の個数に対して対数的にしか増大しない。より一般に、ラデマッハ複雑度はパラメータ数や要素数にかかわらず、実際の標本点上でクラスがどれほど自由に変化できるかを捉える。

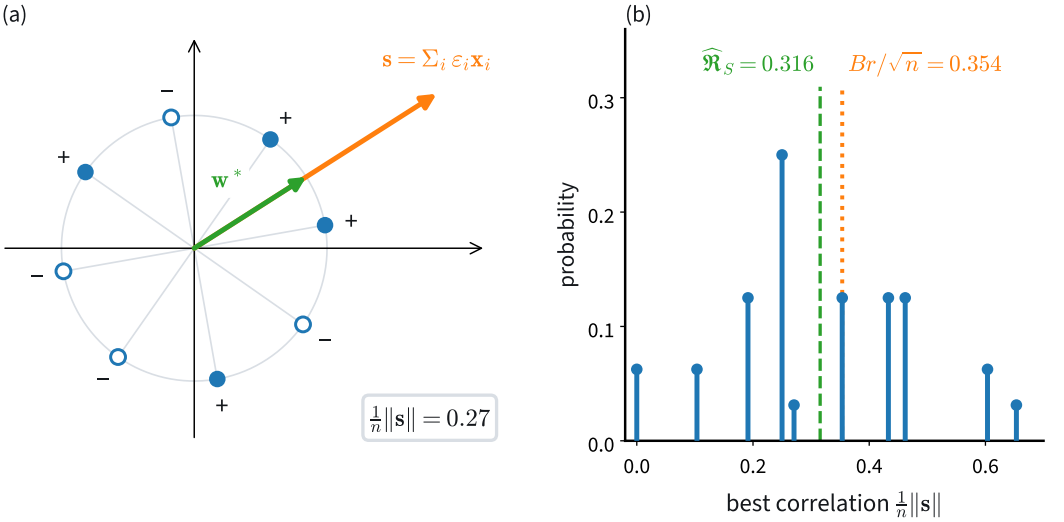


Figure 36.48: 平面上の 8 個の単位ベクトルと単位ノルム線形関数のクラスに対する線形クラスのラデマッハ複雑度。(a) 単一の符号パターンにおいて、塗りつぶした点は $+1$、白抜きの点は -1 を表す。最良の相関は、単位重みベクトル $\mathbf{w}^*$ を符号付き和 $\mathbf{s} = \sum_i \varepsilon_i \mathbf{x}_i$ と同じ向きに揃えたときに達成され、その値は $\|\mathbf{s}\|/n = 0.27$ に等しい。(b) 全 256 通りの符号パターンにわたる最良の相関の厳密な分布。その平均である経験ラデマッハ複雑度は 0.316 であり、本文で証明する上界 $Br/\sqrt{n} = 0.354$ と比較できる。

36.10.2.2 一様収束定理

これで、ラデマッハ複雑度がクラス全体にわたる経験リスクと真のリスクの最大ギャップを制御することを示す準備が整った。予測と損失は異なる関数クラスに属するため、まず記法上の小さな区別をしておく。当てはめるモデルを含むのは**予測器クラス**（predictor class）$\mathcal{F}$ である一方、平均を比較する対象となる量は損失である。そこで、次のように**損失クラス**（loss class）を定義する。

$$\mathcal{G} = \{(\mathbf{x}, y) \mapsto \ell(f(\mathbf{x}), y) : f \in \mathcal{F}\},$$

このクラスに属する関数は $[0,1]$ に値をとる。ラベル付きデータ例を $z = (\mathbf{x}, y)$ と書き、$R(g) = \mathbb{E}[g(z)]$ および $\hat{R}_S(g) = \frac{1}{n}\sum_i g(z_i)$ を $g \in \mathcal{G}$ に対して定める。予測器のリスクとは、対応する損失関数のリスクにほかならない。

命題（ラデマッハ複雑度による一様収束）（Bartlett and Mendelson 2002）。$S = (z_1, \dots, z_n)$ **を独立同分布標本とし、** $\mathcal{G}$ **を** $[0,1]$ **に値をとる関数のクラスとする。このとき、少なくとも** $1 - \delta$ **の確率で次が成り立つ。**

$$R(g) \le \hat{R}_S(g) + 2\,\mathfrak{R}_n(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2n}} \qquad \text{simultaneously for every } g \in \mathcal{G}. \quad (36.126)$$

この上界が有限クラスの上界とまったく同じ 3 項構造をもっていることに注目しよう。すなわち、訓練標本上の経験リスク、クラスの豊かさを測る複雑度項、そして固定された単一の関数に対し

ても現れる信頼項である。ここでは、豊かさの項がランダム符号に適合するクラスの能力の 2 倍になっている。

証明。まず最悪ケースのギャップの期待値の上界 $\mathbb{E}_S[\sup_g (R(g) - \hat{R}_S(g))] \le 2\mathfrak{R}_n(\mathcal{G})$ を 3 つのステップで示し、続いて集中不等式から高確率での主張を導く。

I. ゴースト標本（ghost sample）。証明内部の純粋な解析上の道具として、独立なコピー $S' = (z'_1, \dots, z'_n)$ を標本 S に対して導入する。$R(g) = \mathbb{E}_{S'}[\hat{R}_{S'}(g)]$ であり、また期待値の上限はつねに上限の期待値以下であるため、次を得る。

$$\mathbb{E}_S\Big[\sup_{g\in\mathcal{G}} (R(g) - \hat{R}_S(g))\Big] \le \mathbb{E}_{S,S'}\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \big(g(z'_i) - g(z_i)\big)\Big]. \tag{36.127}$$

これにより、未知の母集団リスクが新たな標本上の経験リスクに置き換えられた。

II. 符号の入れ替え。両方の点は同じ分布からの独立な抽出であるため、z_i と z'_i を入れ替えても (S, S') の同時分布はまったく変わらない。しかしその入れ替えは、i 番目の差 $g(z'_i) - g(z_i)$ の符号をすべての $g \in \mathcal{G}$ に対して同時に反転させる。したがって、符号 $\varepsilon_1, \dots, \varepsilon_n \in \{-1, +1\}$ を任意に固定したとき、i 番目の差に ε_i を掛けても Equation 36.127 の右辺は変化しない。ランダム符号にわたって平均をとっても同様に不変である。

$$\mathbb{E}_{S,S'}\Big[\sup_g \frac{1}{n}\sum_i \big(g(z'_i) - g(z_i)\big)\Big] = \mathbb{E}_{S,S',\varepsilon}\Big[\sup_g \frac{1}{n}\sum_i \varepsilon_i \big(g(z'_i) - g(z_i)\big)\Big].$$

III. 分割。和の上限は上限の和以下であるため、右辺は次式で上から抑えられる。

$$\mathbb{E}_{S',\varepsilon}\Big[\sup_g \frac{1}{n}\sum_i \varepsilon_i\, g(z'_i)\Big] + \mathbb{E}_{S,\varepsilon}\Big[\sup_g \frac{1}{n}\sum_i (-\varepsilon_i)\, g(z_i)\Big] = 2\,\mathfrak{R}_n(\mathcal{G}),$$

ここで最後の等号は、$-\varepsilon$ が ε と同じ分布に従い、両方の期待値がともに $\mathfrak{R}_n(\mathcal{G})$ に等しくなることによる。係数 2 はこの分割から生じる。

期待値から高確率の上界へ移るため、$\Phi(S) = \sup_g (R(g) - \hat{R}_S(g))$ と定義する。単一の標本点 z_i を置き換えても、各経験リスク $\hat{R}_S(g)$ の変化は高々 $1/n$ であり（g は $[0,1]$ に値をとるため）、したがって上限 $\Phi(S)$ の変化も高々 $1/n$ である。ここで、ヘフディングの不等式を和から「単一の入力では大きく変化しない任意の関数」へと一般化する**マクダイアミッドの不等式**（McDiarmid's inequality）(McDiarmid 1989) を適用すると、$P(\Phi - \mathbb{E}\Phi \ge t) \le e^{-2nt^2}$ が得られる（演習問題 1）。右辺を δ とおき、$\mathbb{E}\Phi \le 2\mathfrak{R}_n(\mathcal{G})$ と組み合わせれば Equation 36.126 が得られる。$\blacksquare$

この**対称化**の議論の背後にある直観はきわめて明快である。ゴースト標本は母集団の身代わりを務める。実際の標本点とゴースト標本点は交換可能であるため、どのペアを入れ替えても分布は一切変わらず、その入れ替えをランダム化してしまえばコイン投げだけが残る。つまり、訓練リスクと母集団リスクの最悪ケースのギャップは、クラスがランダム符号をどれほどよく追いかけられるかによって決まるのである。

実務上の細部が 1 つ残っている。定理は期待複雑度 $\mathfrak{R}_n(\mathcal{G})$ を用いて上界を述べているが、実際のデータセット上では手元の標本に対する経験複雑度 $\widehat{\mathfrak{R}}_S(\mathcal{G})$ しか評価できない。標本点を 1 つ変更したときの $\widehat{\mathfrak{R}}_S(\mathcal{G})$ の変化も高々 $1/n$ であるため、マクダイアミッドの不等式をもう一度適

用すれば、信頼項を広げた代償のもとで $\widehat{\mathfrak{R}}_S(\mathcal{G})$ を $\mathfrak{R}_n(\mathcal{G})$ の代わりに Equation 36.126 へ代入でき、信頼項は $3\sqrt{\log(2/\delta)/(2n)}$ となる。以下の数値検証ではこの経験形式を用いる。

極端な場合で定理を確かめておくことはつねに有益である。単一要素クラスでは $\mathfrak{R}_n = 0$ となり、Equation 36.126 はそのまま片側のヘフディングの不等式に帰着する。有限クラスでは、マサールの補題により定数は緩くなるものの Equation 36.124 と同じ形が復元される。たとえば 2,000 個の設定と 10,000 例の場合、複雑度項だけで $2\sqrt{2\log 2000/n} = 0.078$ となるのに対し、ユニオンバウンドでは半幅全体でも 0.024 である。ラデマッハ複雑度は無限クラスを苦もなく扱える引き換えとして、有限クラスでは定数倍のペナルティを払うことになる。二値分類器の場合、n 個の点上で実現可能なラベル付けの総数を数えることで、$\mathfrak{R}_n$ は $\sqrt{d_{\mathrm{VC}}\log n / n}$ の定数倍で上から抑えられる (Boucheron et al. 2005)。ここで d_{VC} は本編で導入した VC 次元、すなわちそのクラスがあらゆる可能なラベル付けを実現できる最大の点数である。

反対の極端なケースとして、$[0,1]$ に値をとる任意の損失クラスは $\mathfrak{R}_n(\mathcal{G}) \le \tfrac{1}{2}$ を満たす。等号が成り立つとき、複雑度項 $2\mathfrak{R}_n(\mathcal{G})$ は 1 に等しくなり、上界は最大 100% までのいかなるリスクも許容する空虚なものになる。$\{-1,+1\}$ に値をとる二値分類器と 0-1 損失 $g(\mathbf{x},y) = (1 - y f(\mathbf{x}))/2$ のもとでは、損失の複雑度と予測器の複雑度は厳密に 2 倍の比になる。実際、相関を展開すると $\sum_i \varepsilon_i g(z_i) = \tfrac{1}{2}\sum_i \varepsilon_i + \tfrac{1}{2}\sum_i (-\varepsilon_i y_i) f(\mathbf{x}_i)$ となる。第 1 の和は平均ゼロで f に依存せず、一方で固定されたラベル $y_i \in \{-1,+1\}$ に対して積 $-\varepsilon_i y_i$ は再び独立な公平ランダム符号となる。上限と期待値をとれば次式が得られる。

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \tfrac{1}{2}\,\widehat{\mathfrak{R}}_S(\mathcal{F}).$$

したがって、二値予測器のクラスが標本上のあらゆる符号パターンに適合できるなら、$\widehat{\mathfrak{R}}_S(\mathcal{F}) = 1$ となり、損失の複雑度の 2 倍が 1 に達するため、一様収束定理は誤り率が 100% 以下であること以上の保証を何も与えない。

36.10.2.3 線形クラス

汎化定理がどれほど役に立つかは、実際に関心のあるクラスに対してその複雑度項を評価できるかどうかにかかっている。重みノルムが有界な線形予測器の場合、その計算はわずか 4 行で済み、しかもその答えは驚くべきことに入力次元に依存しない。

命題（ノルム有界な線形クラスのラデマッハ複雑度）。$\mathcal{F} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x}\rangle : \|\mathbf{w}\|_2 \le B\}$ とし、すべての標本入力が $\|\mathbf{x}_i\|_2 \le r$ を満たすとする。このとき、次が成り立つ。

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{B}{n}\,\mathbb{E}_{\varepsilon}\Big\|\sum_{i=1}^{n}\varepsilon_i \mathbf{x}_i\Big\| \le \frac{B\,r}{\sqrt{n}}. \tag{36.128}$$

証明。 内側の上限は閉じた形で書ける。内積の線形性とコーシー・シュワルツの不等式により、最大値は重みベクトル $\mathbf{w}$（長さ B）が符号付き和の方向を向くときに達成される。

$$\sup_{\|\mathbf{w}\|\le B} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i \langle \mathbf{w}, \mathbf{x}_i\rangle = \sup_{\|\mathbf{w}\|\le B}\frac{1}{n}\Big\langle \mathbf{w}, \sum_i \varepsilon_i \mathbf{x}_i\Big\rangle = \frac{B}{n}\Big\|\sum_{i=1}^{n}\varepsilon_i \mathbf{x}_i\Big\|.$$

ε に関して期待値をとれば命題の等号が得られる。上界を得るには、平方根の凹性（$\mathbb{E}\|\cdot\| \le \sqrt{\mathbb{E}\|\cdot\|^2}$）を用いて二乗ノルムを展開する。

$$\mathbb{E}_{\varepsilon}\Big\|\sum_{i}\varepsilon_i \mathbf{x}_i\Big\| \le \sqrt{\sum_{i,j}\mathbb{E}[\varepsilon_i\varepsilon_j]\,\langle \mathbf{x}_i,\mathbf{x}_j\rangle} = \sqrt{\sum_{i}\|\mathbf{x}_i\|^2} \le r\sqrt{n}.$$

独立な公平符号は $\mathbb{E}[\varepsilon_i\varepsilon_j]=0$（$i\neq j$）を満たすため、交差項はすべて消える。これに B/n を掛ければ $Br/\sqrt{n}$ が得られる。■

Figure 36.48 のパネル (a) はこの証明の最初のステップを図示している。任意の符号パターンに対し、最適な重みベクトルは単に入力の符号付き和と同じ向きに揃う。

注目すべきことに、この証明で緩みが生じるのは 1 箇所だけである。恒等式 $\mathbb{E}\|\sum_i \varepsilon_i\mathbf{x}_i\|^2 = \sum_i\|\mathbf{x}_i\|^2$ は厳密に成り立つため、ギャップはすべて符号付き和の平均の長さをその二乗平均平方根の長さで置き換えることから生じる。この不等式は、符号パターン間で長さがあまり変動しないときにはつねにタイトになる。図の 8 点では、緩みは 11% である（0.316 対 0.354）。2 次元では、よく広がった多数の単位ベクトルの符号付き和は 2 次元ガウスベクトルのように振る舞い、その平均の長さは二乗平均平方根の長さの $\sqrt{\pi}/2\approx 0.89$ 倍になるため、緩みは n にかかわらず 11% 付近にとどまる。しかし 20 次元では、36.9.2.1 節の薄殻現象（thin-shell phenomenon）により符号付き和の長さは二乗平均平方根の周りに鋭く集中し、緩みは約 1% まで縮小する。

Equation 36.128 に何が含まれていないかに注目しよう。入力次元 d が一切現れないのである。次元は B と r を通じて間接的に関与するにすぎない。この上界において、線形モデルの容量は扱う特徴量が 10 個であろうと 100 万個であろうと、重みノルムと入力半径の積によって支配される。これは重み減衰（2.7 節）に対する明快な理論的動機を与える。$\|\mathbf{w}\|$ を小さく保てば、訓練されたモデルを含む関数クラスが縮小するのである。また、この次元に依存しない性質は線形モデルに限られない。出力重みの ℓ_1 ノルムが高々 B_1 であり、隠れ層の重みベクトルの ℓ_2 ノルムが高々 B_2 である 2 層 ReLU ネットワークは、幅にかかわらずラデマッハ複雑度が高々 $2B_1B_2r/\sqrt{n}$ となる（演習問題 8）。

注意深い読者は自然な疑問を抱くかもしれない。命題では半径 B をデータを見る**前**に固定する必要があるのに対し、訓練された重みベクトルのノルムは訓練の**後**になって初めてわかるという点である。この指摘はもっともだが、半径の列にわたるユニオンバウンドによって解決できる。各半径 $B_j=2^j$ に対して失敗確率 $\delta_j=\delta/(j(j+1))$（総和は δ になる）で定理を適用し、当てはめた重みベクトルを含む最小の B_j で上界を評価すればよい。$\|\mathbf{w}\|>1$ のとき、これによるコストは複雑度項で高々 2 倍にとどまり、穏やかな $\sqrt{\log\log\|\mathbf{w}\|/n}$ のペナルティが加わるだけで済む（演習問題 3）。

Equation 36.128 を線形分類器の汎化上界へと変換するには、あと 1 つ橋渡しが残っている。一様収束定理が必要とするのは**損失**クラス $\mathcal{G}$ の複雑度であるのに対し、先ほどの命題が上から抑えたのは**予測器**クラス $\mathcal{F}$ であった。Lipschitz 損失関数がこの 2 つを結び付ける。

補題（縮小補題）（Boucheron et al. 2013）。**$\phi_1,\dots,\phi_n:\mathbb{R}\to\mathbb{R}$ がそれぞれ L-Lipschitz であるとする。このとき、任意の関数クラス $\mathcal{F}$ と点 $\mathbf{x}_1,\dots,\mathbf{x}_n$ に対して次が成り立つ。**

$$\mathbb{E}_{\varepsilon}\Big[\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\varepsilon_i\,\phi_i(f(\mathbf{x}_i))\Big] \le L\,\mathbb{E}_{\varepsilon}\Big[\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\varepsilon_i\,f(\mathbf{x}_i)\Big]. \tag{36.129}$$

平たく言えば、距離を高々 L 倍にしか引き伸ばさない変換を関数クラスに合成しても、そのラデマッハ複雑度は高々 L 倍にしかならない。各 ϕ_i が異なってよいという点は、まさに教師あり学習の損失が求める性質である。データ例 i において、ϕ_i は写像 $u\mapsto \ell(u,y_i)$（ラベル y_i を固定したもの）にあたる。定数 c を ϕ_i に加えても和は期待値ゼロの $c\,\varepsilon_i$ だけずれるにすぎないため、中心化の条件も不要である。ここでは縮小補題（contraction lemma）を証明なしで認める。

また一様収束定理では、損失が $[0,1]$ に有界であることも必要になる。ヒンジ損失やロジスティック損失といった一般的な訓練損失は非有界であり、ノルム有界な線形クラス上ではおよそ $1+Br$ に達し得るため、定理を直接適用するにはその最大値で割って再スケーリングした損失を抑える必要がある。解析上より見通しがよいのは、最初から有界な損失を用いることである。すなわち、分類マージン $u = yf(\mathbf{x})$ に適用する**ランプ損失**（ramp loss）であり、次式で定義される。

$$\phi_\gamma(u) = \min\{1, \max\{0,\, 1 - u/\gamma\}\}.$$

ランプ損失は誤分類で 1 をとり（$u \le 0$）、0 に下がるのは予測が少なくともマージン γ 以上の余裕をもって正解したときであり（$u \ge \gamma$）、その間を線形に補間する。Figure 36.49 に示すとおり、この損失はいたるところで 0-1 損失を上から抑え、$(1/\gamma)$-Lipschitz である。

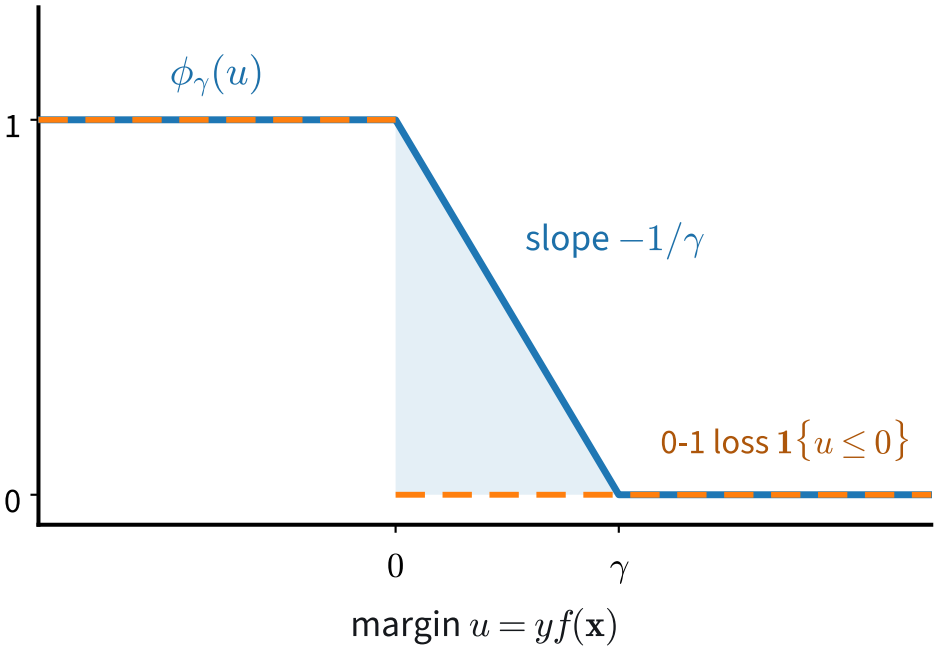


Figure 36.49: ランプ損失 ϕ_γ（青）をマージン $u = yf(\mathbf{x})$ の関数としてプロットしたもの、および 0-1 損失 $\mathbf{1}\{u \le 0\}$（破線）。両者は $0 < u < \gamma$（網掛け部分）を除いて一致し、この領域では予測は正しいもののマージンに届かず、ランプ損失が傾き $-1/\gamma$ で減少する。

標本上の経験ランプ損失を $\hat{R}_\gamma(\mathbf{w}) = \frac{1}{n}\sum_i \phi_\gamma(y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$ と書く。すべての入力が $\|\mathbf{x}\| \le r$ を満たすとき、縮小補題と線形クラスの命題を組み合わせることで、損失クラスの複雑度は $Br/(\gamma\sqrt{n})$ で上から抑えられる。これを Equation 36.126 に代入すると**マージン上界**（margin bound）が得られる。すなわち、少なくとも $1-\delta$ の確率で、すべての $\mathbf{w}$（$\|\mathbf{w}\| \le B$）に対して同時に次が成り立つ。

$$P(y\langle \mathbf{w}, \mathbf{x} \rangle \le 0) \le \hat{R}_\gamma(\mathbf{w}) + \frac{2Br}{\gamma\sqrt{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}. \tag{36.130}$$

左辺は真の分類誤り率であり、右辺は訓練ランプ損失に計算可能な 2 つの項を加えたものである。これらの数値のスケールを見るため、特定のデータセット上で測定するのではなく比を選ん

だ例として、$Br/\gamma = 10$、$n = 50{,}000$、$\delta = 0.05$ という設定を考えよう。このとき複雑度項は 0.089、信頼項は 0.005 で、合計は 0.095 となる。したがって $B = 10\gamma/r$ を事前に固定しておけば、訓練セットをマージン γ で分離するクラス内の任意の線形分類器は、95% の信頼度で真の誤り率が 9.5% 未満であると保証される。対照的に、線形モデルをランダムラベルにマージン γ で無理やり適合させると Br/γ が押し上げられる。$Br/\gamma \approx 10.6$ を超えると 2 つのペナルティ項の和が 0.1 を上回り、$Br/\gamma \approx 110$ 付近では複雑度項だけで 1 に達して上界は空虚になる。

36.10.2.4 数値で見るラデマッハ複雑度

Equation 36.128 は線形クラスの経験ラデマッハ複雑度をユークリッドノルムの期待値として表しているため、ランダムな符号ベクトルにわたって平均をとることで直接推定できる。以下では、50 個の単位ノルム点に対して $d = 2$、20、および $1{,}000$ 次元でこれを計算する。先ほどの緩みの解析によれば、2 次元では推定値が上界 $Br/\sqrt{n} = 1/\sqrt{50}$ を約 11% 下回り、20 次元に達すると上界から約 1% 以内に収まると予測される。

```
n, B, num_signs = 50, 1.0, 2000
signs = rng.choice([-1.0, 1.0], size=(num_signs, n))  # rows: sign vectors
for d in (2, 20, 1000):
    X = rng.standard_normal((n, d))
    X /= np.linalg.norm(X, axis=1, keepdims=True)       # ||x_i|| = r = 1
    sup_corr = B * np.linalg.norm(signs @ X, axis=1) / n  # closed-form sup
    print(f'd = {d:4d}: Monte-Carlo R_S = {sup_corr.mean():.3f}, '
          f'bound B r / sqrt(n) = {B / np.sqrt(n):.3f}')
```

```
d =    2: Monte-Carlo R_S = 0.123, bound B r / sqrt(n) = 0.141
d =   20: Monte-Carlo R_S = 0.141, bound B r / sqrt(n) = 0.141
d = 1000: Monte-Carlo R_S = 0.141, bound B r / sqrt(n) = 0.141
```

モンテカルロ推定値は平面上で約 0.125 となり、20 次元および 1,000 次元では 0.139〜0.141 の範囲に入って上界の 0.141 にぴたりと張り付き、予測どおりの結果となる。次元を 20 から 1,000 へと 50 倍に増やしても、複雑度は変化しない。

スペクトルの反対側の端には、訓練標本上のいかなるラベル割り当ても厳密に再現できるほど豊かな、**補間**（interpolate）可能なクラスがある。$p = 2n$ 個の標準ガウス特徴量上の最小二乗法を考えよう。ここで特徴量行列 $\Phi \in \mathbb{R}^{n \times p}$ は n 個の点の特徴ベクトルを行として並べたものである。33.5.3.1 節の擬似逆行列を用いると、解 $\mathbf{w} = \Phi^{+}\varepsilon$ は任意の符号ベクトル ε に対して最小ノルム補間解を返し、構成上すべてのパターンで相関 1 を達成する。これらの当てはめすべてを収めるのにどれほど大きな重み球が必要かは、先ほどの命題からわかる。あらゆる符号パターンを補間するクラスは $\widehat{\mathfrak{R}}_S \geq 1$ を満たさねばならず、一方で Equation 36.128 はそれを $Br/\sqrt{n}$ で上から抑えるため、すべての補間解を含む球は $Br/\sqrt{n} \geq 1$ を満たさなければならない。実際の当てはめはこの下限にどれほど近づくだろうか。次に、2,000 個のランダム符号ベクトルに当てはめを行い、その重みノルムを理論予測値 $\sqrt{n/(p-n-1)}$（演習問題 4 で導出）と比較し、それらすべてを包含する最小の球について $Br/\sqrt{n}$ を評価する。

```
p = 2 * n                                   # more features than points
Phi = rng.standard_normal((n, p))
W = np.linalg.pinv(Phi) @ signs.T           # min-norm fit of each sign vector
w_norm = np.linalg.norm(W, axis=0)
r = np.linalg.norm(Phi, axis=1).max()       # input radius: largest row norm
row_norm = np.linalg.norm(Phi, axis=1).mean()
print(f'mean ||w|| = {w_norm.mean():.2f} (predicted '
```

```
        f'{np.sqrt(n / (p - n - 1)):.2f}), largest {w_norm.max():.2f}')
    print(f'typical fit: mean ||w|| * mean row norm / sqrt(n) = '
        f'{w_norm.mean() * row_norm / np.sqrt(n):.2f}')
    print(f'r = {r:.1f}, B r / sqrt(n) with B = largest ||w||: '
        f'{w_norm.max() * r / np.sqrt(n):.1f}')
```

```
mean ||w|| = 0.99 (predicted 1.01), largest 1.31
typical fit: mean ||w|| * mean row norm / sqrt(n) = 1.40
r = 12.4, B r / sqrt(n) with B = largest ||w||: 2.3
```

各当てはめの重みノルムは 1 付近であり、Φ の各行の平均の長さは $\sqrt{p} = 10$ 付近であるため、典型的な当てはめでは $\|\mathbf{w}\| r/\sqrt{n} \approx 1.4$ となる。2,000 通りの符号パターンにわたると、最大の重みノルムは 1.3〜1.8、最長の行ノルムは 11〜12.5 の範囲に入り、2,000 個すべての当てはめを包む球では $Br/\sqrt{n}$ が 2〜3 となる。したがって、純粋なノイズを補間するのに必要なノルムは理論上の下限のわずか 1.5〜3 倍にすぎない。線形クラスの上界 $Br/\sqrt{n}$ は、情報量をもたなくなる領域においてさえほぼタイトなままである。

同じ算術はニューラルネットワークにもそのまま当てはまる。与えられたアーキテクチャが表現できるすべての分類器のクラスを考えよう。ランダム符号 ε 上で訓練したときにほぼすべての点で一致する関数が見つかれば、Equation 36.125 の上限は 1 に近い値をその ε に対してとる。これが典型的な符号ベクトルで起きるなら、ラデマッハ複雑度自体が 1 に近くなる。4.5.1.1 節では、MLP が訓練画像上のランダムラベルに誤差ゼロで適合し、Zhang et al. (2021) の画像分類実験を再現した（これらの実験は 10 クラスで少数のラベル付けを試したものであるため、形式的な証明ではなく経験的証拠である）。このとき損失クラスの複雑度の 2 倍は 1 に近くなり、Equation 36.126 はいかなるテスト誤差も許容してしまう。それにもかかわらず、まったく同じネットワークが実際のラベルで訓練すると良好に汎化する。したがって、テスト誤差を支配している要因は関数クラス全体にわたるいかなる最悪ケースの上界からも見えず、データ、オプティマイザ、そして訓練が到達する特定の解に依存している。その仕組みを見るため、学習された解を閉じた形で解析できるモデルに目を向けよう。

36.10.3 補間と二重降下

4.5.2 節では、隠れ層をランダム初期化のまま凍結したネットワークの幅 h をスイープした。テスト誤差は、補間が初めて可能になり隠れ表現の最小特異値が潰れるちょうどその閾値 $h = n$ で急騰し、さらに h を増やすと再び減少に転じた。ここでは、メカニズムのあらゆる要素を陽に計算できる合成データ上でその構成を再現し、本編の説明に 2 つの要素をつけ加える。特異値による当てはめ重みの閉じた公式と、テスト誤差を真の目標関数に起因する部分とラベルノイズに起因する部分へ分解する式である。特徴量は $\phi(\mathbf{x}) = \mathrm{ReLU}(\mathbf{V}\mathbf{x})$（$\mathbf{V}$ は凍結されたランダム行列）であり、これは**ランダム特徴量**（random-features）モデルとして知られる (Rahimi and Recht 2007)。当てはめるのは線形の出力ヘッド $\mathbf{w}$ のみである。初読の読者は、本章の流れを損なうことなくこのパートを読み飛ばしてよい。

36.10.3.1 最小ノルム補間解

補間閾値 $p = n$ がなぜこれほど劇的な振る舞いを示すのかを理解するため、解を閉じた形で書き下そう。p 個の特徴量と n 個の訓練点があるとき、訓練特徴ベクトルを $\Phi \in \mathbb{R}^{n \times p}$ の各行として積み重ね、$\mathbf{w} = \Phi^{+}\mathbf{y}$ を当てはめる。$p = n$ 未満ではこれは通常の最小二乗法であり、$p = n$ より上ではユークリッドノルム最小の補間解となる。

最小ノルム補間解は、数ある補間解の中から恣意的に選んだものではなく、訓練が見つけ出す解そのものである。二乗損失の勾配 $\Phi^{\top}(\Phi\mathbf{w} - \mathbf{y})$ は訓練特徴ベクトルの線形結合であるため、

$\mathbf{w} = \mathbf{0}$ から出発した勾配降下法は Φ の行空間から決して外に出ない。その行空間内に存在する唯一の補間解が $\Phi^{+}\mathbf{y}$ であり、勾配降下法はノルムに対するペナルティを一切課さなくともそこへ収束する。データに同等に適合する多数の解の中から特定の 1 つを選び出すこのようなオプティマイザの選好は、**暗黙的バイアス**（implicit bias）と呼ばれる。

特異値分解を用いると、閾値未満・閾値上・閾値超の 3 つの領域すべてが明快に理解できる。Φ がフルランク $q = \min(n, p)$ をもち、特異値分解 $\Phi = \sum_{i=1}^{q} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\top}$（$\sigma_1 \geq \cdots \geq \sigma_q > 0$）をもつとしよう。このとき擬似逆行列による解は次式となる。

$$\mathbf{w} = \Phi^{+}\mathbf{y} = \sum_{i=1}^{q} \frac{\langle \mathbf{u}_i, \mathbf{y} \rangle}{\sigma_i} \mathbf{v}_i. \tag{36.131}$$

$\mathbf{y}$ の各左特異ベクトル $\mathbf{u}_i$ 方向の成分はその特異値 σ_i で割られるため、最小特異値 $\sigma_{\min} = \sigma_q$ が最も強力な増幅器として働く。$\mathbf{y}$ のうち $\mathbf{u}_q$ 方向に落ちる成分は何であれ $\mathbf{w}$ に $1/\sigma_{\min}$ 倍に拡大されて入り込み、この増幅はラベルノイズと、特徴量ではうまく近似できない真の目標関数の成分の両方を直撃する。

$p < n$ のとき、当てはめ予測 $\Phi\mathbf{w}$ は単に $\mathbf{y}$ を p 次元の $\mathbf{u}_i$ が張る空間へ射影し、残りを訓練残差として残す。しかし $p = n$ では、それら n 個の方向が $\mathbb{R}^n$ 全体を張るため、当てはめは厳密になり、表現しにくい成分を吸収する残差が一切残らない。正方のランダム ReLU 特徴量行列は概収束の意味でほぼ確実に正則であるが、条件が**良い**ことはめったにない。その最小特異値は典型的に極小であり、33.5.3.2 節の条件数解析により、解は $\mathbf{v}_q$ 方向に大きく飛び出す。$p = n$ を超えて特徴量を追加すると、行列が $\mathbb{R}^n$ を張るための余裕が増え、$\sigma_{\min}$ は再び上昇する。

閾値を超えると、特徴量集合が入れ子になっている限り（$(p+1)$ 特徴量モデルがそれまでの p 個の特徴量をすべて保持して新たに 1 つ追加する構成）、特徴量を追加するにつれて解のノルムは減少するほかない。これは 4.5.2 節のパディングの議論から従う。新しい特徴量に重みゼロを割り当てれば、p 特徴量の最小ノルム補間解はそのまま同じノルムをもつ妥当な $(p+1)$ 特徴量の補間解になるため、新たな最小ノルムはそれより大きくなり得ないからである。なお、この単調性は厳密な補間に依拠している。$p = n$ 未満では最小二乗解は補間を行わず、そのノルムは実際には p とともに**増大**する。以下のスイープでは、約 0.5（$p = 2$）から 2 近く（$p = 15$）まで上昇する。

線形クラスの上界は、スイープに沿って監視すべき自然な量を示唆している。すなわち**容量積**（capacity product）$\|\mathbf{w}\| \cdot \max_i \|\phi(\mathbf{x}_i)\| / \sqrt{n}$ であり、これは当てはめた重みを含む最小の球に対する $Br/\sqrt{n}$ に対応する。このスイープでは有界な分類損失ではなく非有界な回帰目標に対する二乗損失を用いるため、この量は形式的な上界としてではなく純粋な診断指標として借用する。閾値を超えると、特徴量半径が $\sqrt{p}$ のように増大する一方で $\|\mathbf{w}\|$ は縮小するため、積全体の正味の挙動は事前には自明でない。

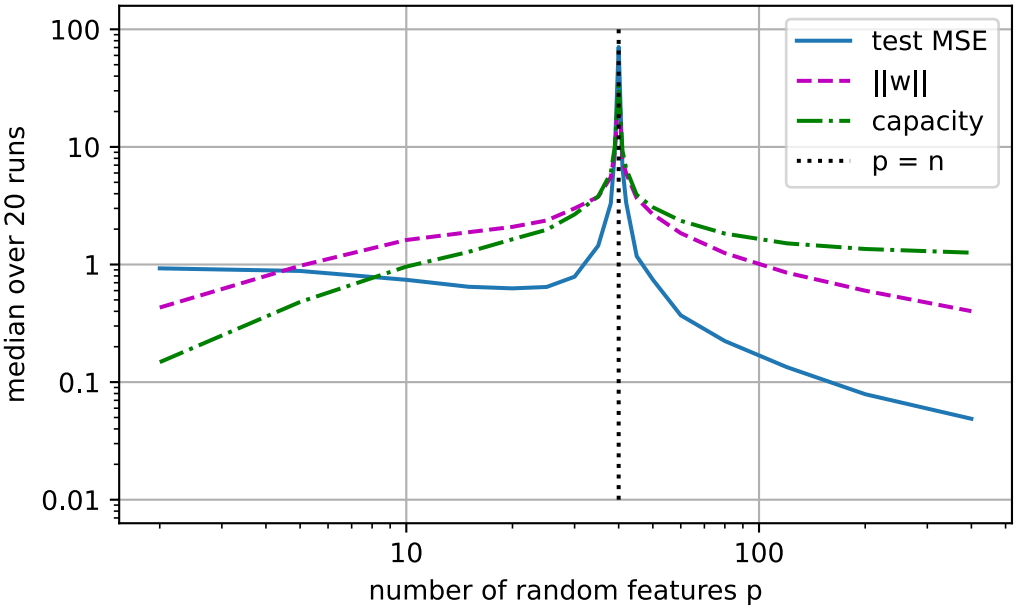
36.10.3.2 二重降下のスイープ

具体的な問題上で、これらの量が連動して変化する様子を観察しよう。$n = 40$ 個の訓練点を線形教師 $y = \langle \beta, \mathbf{x} \rangle$（$d = 15$ 次元、$\|\beta\| = 1$）から抽出し、標準偏差 0.1 のガウスラベルノイズを加える。入れ子になった ReLU ランダム特徴量を構成し、`pinv` によって線形ヘッドを当てはめることで、$p = 40$ 未満では最小二乗解、それより上では最小ノルム補間解を得る。閾値のピークが鋭く解像されるよう、p を 2 から 400 まで（$p = 39$、40、41 を含めて）スイープする。各 p において、テスト MSE、訓練 MSE、$\|\mathbf{w}\|$、$\sigma_{\min}$、および容量積に加え、36.10.3.3 節で解析するテスト誤差の 2 つの成分、すなわち**クリーン成分**（clean part、ノイズのないラベルに当てはめたときのテスト誤差）と**ノイズ成分**（noise part、ラベルノイズによって注入される追加のテスト誤差）を記録する。20 回の独立な試行にわたって中央値をとることで、曲線を読みやすく保つ。

```
n, n_test, d_in, p_max, trials, noise = 40, 500, 15, 400, 20, 0.1
p_grid = np.array([2, 5, 10, 15, 20, 25, 30, 35, 38, 39, 40, 41, 42, 45, 50,
                   60, 80, 120, 200, 400])
keys = ('test', 'train', 'norm', 's_min', 'capacity', 'clean', 'noise',
        'ratio_q', 'ratio')
stats = np.zeros((len(keys), trials, len(p_grid)))
ols = np.zeros(trials)                              # least squares on x itself
for tr in range(trials):
    beta = rng.standard_normal(d_in)
    beta /= np.linalg.norm(beta)                    # teacher: y = <beta, x>
    X, X_te = (rng.standard_normal((size, d_in)) for size in (n, n_test))
    y_clean, y_te = X @ beta, X_te @ beta
    y = y_clean + noise * rng.standard_normal(n)    # noisy training labels
    ols[tr] = np.mean((X_te @ np.linalg.lstsq(X, y, rcond=None)[0] - y_te) ** 2)
    V = rng.standard_normal((p_max, d_in)) / np.sqrt(d_in)
    Phi, Phi_te = np.maximum(X @ V.T, 0), np.maximum(X_te @ V.T, 0)
    for j, p in enumerate(p_grid):                  # nested feature prefixes
        F, F_te = Phi[:, :p], Phi_te[:, :p]
        P = np.linalg.pinv(F)                       # least squares / min norm
        w = P @ y
        r = np.linalg.norm(F, axis=1).max()         # feature radius
        _, S, Vt = np.linalg.svd(F, full_matrices=False)
        ratio = ((np.linalg.norm(F_te @ Vt.T, axis=0) ** 2 / n_test)
                 / (S ** 2 / n))                    # test/train feature power
        stats[:, tr, j] = (
            np.mean((F_te @ w - y_te) ** 2),                # test MSE
            np.mean((F @ w - y) ** 2),                      # training MSE
            np.linalg.norm(w),                              # ||w||
            S.min(),                                        # sigma_min
            np.linalg.norm(w) * r / np.sqrt(n),             # capacity
            np.mean((F_te @ (P @ y_clean) - y_te) ** 2),    # clean part
            noise ** 2 * np.linalg.norm(F_te @ P) ** 2 / n_test,  # noise part
            ratio[-1], ratio.mean())                # along v_q, average
med = dict(zip(keys, np.median(stats, axis=1)))
```

下の図は、テスト MSE、解のノルム、および容量積の中央値を両対数軸上にプロットしたものである。

```
d2l.plot([p_grid] * 3 + [[n, n]],
         [med['test'], med['norm'], med['capacity'], [1e-2, 1e2]],
         'number of random features p', 'median over 20 runs',
         legend=['test MSE', '||w||', 'capacity', 'p = n'], xscale='log',
         yscale='log', fmts=('-', 'm--', 'g-.', 'k:'), figsize=(6, 3.5))
```



下の表は、代表的な 7 つの特徴量数における詳細な内訳を出力したものである。最後の 2 列は 36.10.3.3 節に対応する量であり、最終行は生の入力に直接最小二乗法を当てはめた古典的ベースラインを報告している。

```
print('   p ' + ' '.join(f'{key:>8}' for key in keys))
for p in (15, 38, 40, 42, 50, 80, 400):
    row = [med[key][p_grid == p][0] for key in keys]
    print(f'{p:4d} ' + ' '.join(f'{v:8.2g}' for v in row))
print(f'least squares on the {d_in} raw inputs: test MSE {np.median(ols):.4f}')
```

```
   p     test    train     norm    s_min capacity    clean    noise  ratio_q    ratio
  15     0.65     0.28      1.9        1      1.3     0.67   0.0066      3.7      1.8
  38      3.3   0.0081      5.5     0.08        6      2.9     0.31  6.8e+02       32
  40       71  2.3e-28       29    0.016       34       58      4.2  1.5e+04  4.2e+02
  42      3.4  1.3e-29        6    0.076      6.5      2.9     0.26  5.8e+02       26
  50     0.74  6.6e-30      2.7     0.25      3.1     0.64    0.063       62      6.3
  80     0.22    7e-30      1.3     0.71      1.8     0.19    0.021       10      2.1
 400    0.049  5.3e-30      0.4      2.6      1.3     0.04   0.0095      2.3     0.95
least squares on the 15 raw inputs: test MSE 0.0066
```

この表は、閾値未満・閾値上・閾値超の 3 つの領域に分けて読むと最もわかりやすい。

閾値未満では、モデルは訓練点数より少ないパラメータをもち、古典的に振る舞う。テスト誤差はまず約 1（$p=2$）から約 0.65〜0.85（$p=15$ 付近）へと下がり、その後 p が 40 に近づくにつれて再び上昇して、おなじみの古典的な U 字曲線を描く。この領域と何かを比較する前に、2 つの注意点が欠かせない。第 1 に、最良の地点でさえこの古典的な当てはめは貧弱である。目標値の分散は 1（つねにゼロと予測したときの誤差）であるため、U 字曲線の最小点でも分散の約 3 分の 1 しか取り除けていない。第 2 に、はるかに優れた古典的モデルが存在する。真の教師は 15 個の生の入力に関して線形であるため、生の入力に対する通常の最小二乗法は期待テスト誤差 $0.01 \cdot 15/(40-15-1) \approx 0.006$（ノイズ分散に $d/(n-d-1)$ を $d=15$ として掛けた値）を達成し、これは出力の最終行からも確認できる。ランダム特徴量モデルは設計上ハンディキャ

ップを負っているのである。ReLU の尾根から線形関数を再構成しなければならず、40 点では到底足りない。

閾値上では、3 つの列が連動して動く。訓練誤差は 10^{-3} の数倍程度（$p = 38$）から約 10^{-28} へと急落し、浮動小数点精度の範囲で厳密な補間となる。同時に、最小特異値は 0.01〜0.04 付近まで落ち込み、$p = 38$ や $p = 42$ のときより数分の 1 に小さくなる。Equation 36.131 は $\sigma_{\min}$ で割るため、解は爆発する。$\|\mathbf{w}\|$ は 10〜30 に跳ね上がり、テスト MSE は 10〜70 へと急騰して U 字曲線の最小値より 1〜2 桁高くなる。これらの範囲が広いのは、正方ガウス行列で $\mathbb{E}[1/\sigma_{\min}^2] = \infty$ となるのとちょうど同じくピークが裾の重い分布をもつためであり、20 試行の中央値でさえ実行ごとに目立って変動する。

閾値超では、急騰したすべての量が再び下がる。最小特異値は約 2.8（$p = 400$）まで回復する。パディングの議論が保証するとおり、$\|\mathbf{w}\|$ はすべての試行で（中央値だけでなく）単調に減少し、約 0.4 に達する。テスト MSE もそれに追随して低下し、約 0.2（$p = 80$）、さらに 0.04〜0.07（$p = 400$）まで下がる。その一方で、1 つの列は完全に沈黙する。訓練誤差は浮動小数点の範囲でゼロのまま $p = 40$ 以降の全域に張り付くため、訓練損失に基づくいかなる基準も $p = 42$ と $p = 400$ を見分けられない（両者のテスト誤差が 1 桁以上異なるにもかかわらずである）。解のノルム、$\sigma_{\min}$、および容量積はいずれも両者を明確に識別する。

では、このスイープは実際のところ何を証明しているのだろうか。劇的な曲線が一見して示唆するほど多くはない。入れ子になったランダム特徴量の族の内部では、最も幅の広いモデルがすべてのラベルノイズを暗記する地点をはるかに超えてなお、最良の過少パラメータモデルを含むあらゆる狭いモデルを約 10 倍の差で打ち負かす。これは紛れもなく驚くべき現象である。しかしそれでも、生の入力に対する通常の最小二乗法にはさらに 10 倍の差で敗れる。補間モデルが正しく特定された小さなモデルを上回れるかどうかは別の問いであり、この設定での答えは否である。

容量積は 10〜35 付近でピークに達し（$p = 40$）、その後は約 1.2〜1.3（$p = 400$）まで低下する。テスト誤差曲線の形状を映し出してはいるが、これを汎化保証書と取り違えてはならない。本節のどの定理も非有界な二乗損失をカバーしておらず、この積は目標値の再スケーリングに対して不変でもなく、さらに B はデータを見る前に固定されたのではなく当てはめ後の解から読み取ったものだからである。

最後に、ピークを引き起こしている原因はラベルノイズでは**ない**ことに注意しよう。`noise = 0` としてスイープを再実行してもほぼ同じ高さのピークが現れ、その理由は表からわかる。$p = 40$ の時点で、テスト誤差のクリーン成分はすでにノイズ成分より大きく、典型的には数倍に達しているからである。線形教師は有限個の ReLU 特徴量の線形結合としては書けないため、$\mathbf{u}_q$ 方向の近似誤差こそが $\sigma_{\min}$ によって割られる正体なのである。教師を最初の 10 個の特徴量の厳密な線形結合に置き換えてノイズを取り除けば、ピークは完全に消滅する（演習問題 6）。

36.10.3.3 良性過学習

謎が 1 つ残っている。$p = 400$ において、モデルはノイズを含むすべての訓練ラベルを残差ゼロで補間しているにもかかわらず、スイープ中で最小のテスト誤差を達成している。なぜその領域では、ノイズへの適合がこれほどわずかな悪影響しかもたらさないのだろうか。

テスト誤差を計算可能な 2 つの部分に分解することで、この問いに答えることができる。ノイズを含む訓練ラベルを $\mathbf{y} = \mathbf{y}_0 + \xi$ と書く。ここで $\mathbf{y}_0$ はノイズのないラベル、ξ は各例あたり分散 $\sigma_\xi^2 = 0.01$ をもつ独立なノイズである。擬似逆行列の線形性により $\mathbf{w} = \Phi^{+}\mathbf{y}_0 + \Phi^{+}\xi$ となるため、テスト予測は $\Phi_{\text{te}}\Phi^{+}\mathbf{y}_0 + \Phi_{\text{te}}\Phi^{+}\xi$ の 2 項に分かれる。ξ は平均ゼロであるため交差項は期待値において消え、n_{te} 個のテスト点にわたる期待テスト MSE の厳密な分解が得られる。

$$\mathbb{E}_{\xi}\big[\text{test error}\big] = \underbrace{\frac{1}{n_{\text{te}}}\big\|\Phi_{\text{te}}\Phi^{+}\mathbf{y}_0 - \mathbf{y}_{\text{te}}\big\|^2}_{\text{clean part}} + \underbrace{\frac{\sigma_\xi^2}{n_{\text{te}}}\big\|\Phi_{\text{te}}\Phi^{+}\big\|_F^2}_{\text{noise part}}.$$

表の `clean` 列と `noise` 列はこれら 2 つの項を出力している。$p = 40$ ではノイズ成分は約 1〜4（ラベルノイズ分散 0.01 の 100 倍以上）となり、クリーン成分はさらに大きい。$p = 42$ になるとノイズ成分は 0.1 の数倍程度まで下がり、$p = 80$ で約 0.02、$p = 400$ では 0.01 をわずかに下回る。同じ区間において、クリーン成分はピーク時の約 8〜60 から約 0.05（$p = 400$）まで低下するため、閾値以降の改善の大部分は目標関数の近似精度の向上に由来している。

$p = 40$ で急騰を引き起こしたまさにその仕組みである「小さな特異値による除算」が、なぜ $p = 400$ ではほぼ無害になるのだろうか。Equation 36.131 より、$\mathbf{u}_i$ 方向のノイズ成分は $1/\sigma_i$ 倍にスケーリングされて $\mathbf{v}_i$ 方向の重みに入り込み、テスト特徴量の $\mathbf{v}_i$ 方向成分に比例してテスト予測を動かす。$\|\Phi_{\text{te}}\Phi^{+}\|_F^2 = \sum_i \|\Phi_{\text{te}}\mathbf{v}_i\|^2/\sigma_i^2$ と $\sigma_i^2 = \|\Phi\mathbf{v}_i\|^2$ を併せて用いると、$p \geq n$ におけるノイズ成分は次の示唆に富む形をとる。

$$\sigma_\xi^2 \cdot \frac{1}{n}\sum_{i=1}^{n} \frac{\|\Phi_{\text{te}}\mathbf{v}_i\|^2/n_{\text{te}}}{\|\Phi\mathbf{v}_i\|^2/n},$$

これはノイズ分散に、各右特異方向 $\mathbf{v}_i$ に沿ったテスト特徴量の二乗平均エネルギーと訓練特徴量の二乗平均エネルギーの比の平均を掛けたものである。極小の特異値 σ_i が問題を引き起こすのは、テスト入力が同じ方向に大きなエネルギーをもつ場合に限られる。$p = n$ における極小の $\sigma_{\min}$ は 40 個の訓練点にたまたま生じた偶然の産物であり、$\mathbf{v}_q$ 方向ではテスト特徴量の二乗平均が訓練特徴量より典型的に数千倍も大きくなる。対照的に $p = 400$ では、小さな σ_i は訓練点でもテスト点でも同様に特徴量が小さい方向を表している。表の最後の 2 列がこのことを裏付けており、`ratio_q`（$\mathbf{v}_q$ 方向）は約 2 にすぎず、`ratio`（全方向の平均）は 1 をわずかに下回るため、小さな分母が小さな分子によって打ち消される。過学習が**良性**（benign）と呼ばれるのは、この意味においてである。なお、良性とはゼロではなく**有界**という意味であることに注意しよう。$p = 400$ においても、予測は依然としてノイズを含む 1 個の訓練ラベルとほぼ同程度の分散を抱えている。

Bartlett et al. (2020) は線形回帰に対して、入力の共分散行列の 2 つの有効ランクを通じてこの現象を厳密に定式化しており、演習問題 7 でこれを直接計算する。深層ニューラルネットワークでは特徴量自体が学習され、そのスペクトルも訓練を通じて変化するため、訓練済みネットワークのテスト誤差を第一原理から予測する理論はいまだ手の届かないところにある。本節で扱ったすべての形式的な上界はデータを見る前に選んだクラス内の最悪の関数をカバーするものであるのに対し、実際に運用する予測器はオプティマイザが見つけ出す特定の解である。関数クラスだけでなくアルゴリズムとデータに依存する保証によってその隔たりを埋めることこそ、本節が残す中心的な未解決問題である。

36.10.4 まとめ

データを見る前に固定した単一の関数に対する集中上界は、標本上で最良のスコアを出したという理由で選ばれたモデルをカバーしない。一様収束は、クラス内のすべての関数の推定誤差を同時に上から抑えることでこの隔たりを埋める。有限クラス $\mathcal{F}$ に対しては、ヘフディングの不等式

とユニオンバウンドにより、追加データ例 $\log|\mathcal{F}|/(2t^2)$ 個（許容誤差 t のもと）のコストでこれが達成される。無限クラスに対しては、ゴースト標本を用いた対称化によって最悪ケースのギャップがランダム符号との相関に帰着し、損失クラスのラデマッハ複雑度の 2 倍が経験リスクに加わる。ノルム有界な線形モデル（$\|\mathbf{w}\|_2 \le B$、$\|\mathbf{x}_i\|_2 \le r$）では、この複雑度は次元に依存せず高々 $Br/\sqrt{n}$ となる。一様収束が限界に達するのは、ランダムラベルに適合する深層ネットワークのように任意の符号パターンを補間できるほど柔軟なクラスにおいてであり、そこでは上界が空虚になる。ランダム特徴量のスイープは、そのような領域でもなお汎化が成功し得る理由を鮮やかに示している。テスト誤差は $\sigma_{\min}$ が潰れる場所で急騰し、最小ノルム補間解が縮小するにつれて低下するが、これらは関数クラス全体ではなく学習された解の性質なのである。

36.10.5 演習問題

1. **マクダイアミッドの不等式。**●McDiarmid (1989) の不等式は、ヘフディングの不等式を和から**有界差分**をもつ関数へと拡張する。いま、i 番目の引数のみを変化させたときの $g(x_1,\dots,x_n)$ の値の変化が高々 c_i であり、かつ $X_1,\dots,X_n$ が独立であるとする。このとき、

$$P\big(g(X_1,\dots,X_n) - \mathbb{E}[g(X_1,\dots,X_n)] \ge t\big) \le \exp\left(-\frac{2t^2}{\sum_{i=1}^{n} c_i^2}\right), \tag{36.132}$$

が成り立ち、下側の裾に対しても同じ上界が成り立つ。この主張の証明にはマルチンゲールを用いるため、ここでは証明なしで認めることとする。

   1. この不等式から、独立な項 $X_i \in [a_i, b_i]$ の平均に対するヘフディングの不等式を導出せよ。
   2. $\Phi(S) = \sup_{g\in\mathcal{G}} \big(R(g) - \widehat{R}_S(g)\big)$ を、損失クラス $\mathcal{G}$（値域は $[0,1]$）に対して定める。Φ が有界差分 $c_i = 1/n$ をもつことを示せ。さらにマクダイアミッドの不等式を本文の対称化上界 $\mathbb{E}_S[\Phi(S)] \le 2\mathfrak{R}_n(\mathcal{G})$ と組み合わせ、Equation 36.126 を導出せよ。
   3. $[0,1]$ 内のデータに対し、標本平均は有界差分 $1/n$ をもつ。奇数サイズ n の $[0,1]$ 内の標本であって、1 点を変更すると標本中央値が 1 だけ動くものを構成せよ。このことは、36.5 節における中央値のブートストラップ推定値にマクダイアミッドの不等式でエラーバーを付けることに対して何を意味するか。

2. **マサールの補題と ℓ_1 球。**●有限クラスに対するマサールの補題と、ℓ_1 有界な線形クラスのラデマッハ複雑度は、いずれもサブガウス変数の最大値の期待値に対する 1 つの上界に基づいている。

   1. $Z_1,\dots,Z_N$ を中心化された分散プロキシ σ^2 のサブガウス変数、すなわち $\mathbb{E}[e^{\lambda Z_j}] \le e^{\lambda^2\sigma^2/2}$ がすべての λ で成り立つ変数とする（互いに独立である必要はない）。$\mathbb{E}[\max_j Z_j] \le \sigma\sqrt{2\log N}$ が成り立つことを、$e^{\lambda \mathbb{E}[\max_j Z_j]}$ を $\sum_j \mathbb{E}[e^{\lambda Z_j}]$ で上から抑えて $\lambda > 0$ について最適化することにより示せ。さらにこれから、マサールの補題 $\widehat{\mathfrak{R}}_S(\mathcal{F}) \le \sqrt{2\log|\mathcal{F}|/n}$ を $[-1,1]$ に値をとる有限クラスに対して導け。
   2. $\mathcal{F} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x}\rangle : \|\mathbf{w}\|_1 \le B\}$ とし、$\|\mathbf{x}_i\|_\infty \le r$ がすべての i で成り立つとする。Equation 36.128 の証明にならい、コーシー・シュワルツの不等式の代わりに双対性 $\sup_{\|\mathbf{w}\|_1\le B} \langle \mathbf{w}, \mathbf{v}\rangle = B\|\mathbf{v}\|_\infty$ を用いて、$\widehat{\mathfrak{R}}_S(\mathcal{F}) \le Br\sqrt{2\log(2d)/n}$ を示せ。なぜここでは次元 d が現れ、しかも対数的にしか効かないのか。
   3. 入力が $\{-1,+1\}^d$ に属し、目標の重みベクトルが k 個の非ゼロ成分（それぞれ ± 1 に等しい）をもつとする。目標ベクトルを含む最小の半径 B をそれぞれ用いたとき、(b) の上界を Equation 36.128 と比較せよ。どの範囲の k に対して ℓ_1 上界の方が小さくなるか（$d = 10^4$ の場合）。

3. **マージン上界。**●$\mathcal{F}=\{\mathbf{x}\mapsto\langle\mathbf{w},\mathbf{x}\rangle:\|\mathbf{w}\|_2\le B\}$、$\|\mathbf{x}\|_2\le r$ とし、ラベルを $y\in\{-1,+1\}$ とする。またランプ損失 $\phi_\gamma(u)=\min\{1,\max\{0,1-u/\gamma\}\}$（固定した $\gamma>0$）をマージン $u=yf(\mathbf{x})$ に作用させる。

   1. $\phi_\gamma(yf(\mathbf{x}))$ が少なくとも 0-1 損失 $\mathbf{1}\{yf(\mathbf{x})\le 0\}$ 以上であり、ϕ_γ が $(1/\gamma)$-Lipschitz であることを示せ。さらに、各予測 $f(\mathbf{x}_i)$ に固定されたラベル y_i を掛けても $\widehat{\mathfrak{R}}_S(\mathcal{F})$ が変化しないことを示せ。
   2. (a)、縮小補題 Equation 36.129、Equation 36.126、および Equation 36.128 を組み合わせて Equation 36.130 を導け。$n=50{,}000$、$\delta=0.05$ のとき、複雑度項と信頼項の和を小さく保てる最大の比 Br/γ（和が 0.1 未満となる上限）を求めよ。
   3. この上界はデータの抽出前に固定した半径 B に対して成り立つが、訓練された重みベクトルのノルムは事後にしかわからない。半径 $B_j=2^j$、失敗確率 $\delta_j=\delta/(j(j+1))$ としてすべての $j\ge 1$ に上界を適用し、j にわたってユニオンバウンドをとれ。データを見た後に選ばれた $\|\mathbf{w}\|>1$ を満たす重みベクトルに対して成り立つ上界を書き下せ。さらに、半径を選べる自由度の代償が複雑度項において高々 2 倍と、オーダー $\sqrt{\log\log\|\mathbf{w}\|/n}$ の追加項にとどまることを示せ。

4. [code] **ランダム符号への当てはめ。**●36.10.2.4 節の 2 番目のコードセルでは、$2{,}000$ 個の符号ベクトル $\varepsilon\in\{-1,+1\}^n$ に対し $\mathbf{w}=\Phi^{+}\varepsilon$ を $p=2n$ 個の標準ガウス特徴量上で当てはめ、得られた重みのノルムを測定している。

   1. $p<n$ とし、Φ が列フルランクをもつとする。相関 $\frac{1}{n}\sum_i \varepsilon_i(\Phi\mathbf{w})_i$ の符号に関する期待値が p/n になることを示せ。行列 $\Phi\Phi^{+}$ は射影である。
   2. $p>n$ とし、r を Φ の最大の行ノルムとし、B をクラス $\{\|\mathbf{w}\|\le B\}$ がすべての符号ベクトルの補間解を含むほど十分大きくとる。このとき $Br/\sqrt{n}$ が 1 より小さくなり得ない理由を説明せよ。標準ガウス行列 Φ で $p>n+1$ のとき、$\mathbb{E}[(\Phi\Phi^{\top})^{-1}]=\mathbf{I}_n/(p-n-1)$ が成り立つ。この恒等式を用いて、典型的な $\|\mathbf{w}\|$、および $\|\mathbf{w}\|$ に Φ の典型的な行ノルムを掛けて $\sqrt{n}$ で割った典型値を、p と n の関数として予測せよ。
   3. $n=50$ とし、$p\in\{10,25,45,55,100,400,1000\}$ に対してセルを再実行して（$p<n$ ではノルムの代わりに相関を出力すること）、(a) および (b) と比較せよ。(b) の容量値は $p\to\infty$ のときどうなるか。また、最大の行ノルムを用いた版が 1 より上にとどまるのはなぜか。

5. [code] **リーダーボード攻撃。**●●あるリーダーボードが、提出された各ラベルベクトルの 0-1 損失を $n=4{,}000$ 例（ラベル $\mathbf{y}\in\{0,1\}^n$）のホールドアウトセット上で報告するものとする。Blum and Hardt (2015) の**ブースティング攻撃**はタスクに関する知識を一切使わない。k 個の一様ランダムなラベルベクトルを提出し、報告された損失が高々 $\frac{1}{2}$ であったものを残して、それらの座標ごとの多数決を提出するのである。新たなデータ上では、この多数決の損失は $\frac{1}{2}$ である。

   1. 一様ランダムな $\mathbf{y}$ と $k\in\{10,30,100,300,1000,3000\}$ に対してこの攻撃を実装し、多数決に対して報告される損失を記録せよ。その値と $\frac{1}{2}$ との距離を、事前登録された Equation 36.124 による $k+1$ モデルのプール（$\delta=0.05$）に許容されるずれと比較せよ。どの k から攻撃がこの許容ずれを超え、また命題のどの仮定に違反しているか。
   2. Blum and Hardt (2015) は報告される損失が $\frac{1}{2}-\Omega(\sqrt{k/n})$ になることを証明しているが、定数は与えていない。ガウス近似を用いて発見的にこの定数を導出せよ。まず残されたベクトルが $\mathbf{y}$ に対して示す過剰一致率の期待値を見積もり、次にそのような約 $k/2$ 個のベクトルの多数決が 1 つの座標で正解する確率を見積もること。得られた定数を (a) の測定値と比較せよ。

3. Blum and Hardt (2015) の **Ladder** メカニズムは、提出がそれまでに公開された最良スコアをステップ幅 η より大きく上回ったときにのみ新しいスコアを η の倍数に丸めて公開し、それ以外では前回のスコアを繰り返す。Ladder に対しては、攻撃者は公開スコアを更新させたベクトルしか残せない。両者を $\eta \in \{0.005, 0.01\}$ および $k \in \{100, 1000, 3000\}$ で実装せよ。このとき多数決は $\frac{1}{2}$ をどれだけ下回り、(a) の許容ずれと比べてどうなるか。

6. [code] **正方行列のケース**。●36.10.3.2 節の二重降下実験のスイープを用いよ。

   1. 保存されたノルムから、閾値未満では入れ子の単調性が成り立たないことを示し、その領域でパディングの議論が適用できない理由を説明せよ。
   2. $p = n$ において、$\mathbf{w}$ を Equation 36.131 に沿って分解せよ。最小特異方向の係数 $|\langle \mathbf{u}_n, \mathbf{y} \rangle| / \sigma_{\min}$ と $\|\mathbf{w}\|$ を試行にわたる中央値として並べて報告せよ。この 1 つの方向がノルムのどれだけの割合を占めているか。
   3. `noise = 0` としてスイープを再実行せよ。次に教師を、最初の 10 個の特徴量の厳密な線形結合である目標関数 $y = \langle \mathbf{c}, \phi_{1:10}(\mathbf{x}) \rangle$（$\|\mathbf{c}\| = 1$、ノイズなし）に置き換えよ。2 つの曲線の様子を述べよ。それぞれの場合において、$\sigma_{\min}$ は何を割っているか。また、目標関数が 10 個の特徴量しか使っていないとき、特徴量を追加するにつれて最小ノルム補間解がかえって**悪化**するのはなぜか。
   4. 標準偏差 0.1 のラベルノイズを (c) の目標関数に加えよ。このときテスト誤差が最も高くなるのはどこか、そのピークはノイズ分散と比べてどうなるか、また誤差が最も低くなるのはどこか。

7. [extended] **良性スペクトル**。●●Bartlett et al. (2020) は、線形回帰における最小ノルム補間解がいつ小さな過剰リスクをもつかを特徴づけており、これは 36.10.3.3 節で定性的に説明したメカニズムに対応する。$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$、$\Sigma = \mathrm{diag}(\lambda_1, \dots, \lambda_p)$（$\lambda_1 \geq \lambda_2 \geq \cdots$）とし、$y = \langle \theta^*, \mathbf{x} \rangle + \xi$（$\xi \sim \mathcal{N}(0, \sigma^2)$）とし、$n < p$ 個の訓練入力を $\mathbf{X} \in \mathbb{R}^{n \times p}$ に積み重ねて $\hat{\theta} = \mathbf{X}^{+} \mathbf{y}$ とおく。過剰リスクは $(\hat{\theta} - \theta^*)^{\top} \Sigma (\hat{\theta} - \theta^*)$ である。

   1. ノイズに関する過剰リスクの期待値が、統計的バイアス項 $\theta^{*\top} (\mathbf{I} - \mathbf{P}) \Sigma (\mathbf{I} - \mathbf{P}) \theta^*$（ただし $\mathbf{P} = \mathbf{X}^{+}\mathbf{X}$）と分散項 $\sigma^2 \, \mathrm{tr}\big((\mathbf{X}^{+})^{\top} \Sigma \mathbf{X}^{+} \big)$ の和になることを示せ。両方の項を実装せよ（テストセットは不要である）。
   2. 同論文の有効ランクは $\rho_k = \sum_{i>k} \lambda_i / \lambda_{k+1}$ および $\varrho_k = \big(\sum_{i>k} \lambda_i\big)^2 / \sum_{i>k} \lambda_i^2$ で定義される。k^* を、条件を満たす最小の k（$\rho_k \geq b n$ となるもの）とおくと、分散項は定数因子を除いてオーダー $\sigma^2 (k^*/n + n/\varrho_{k^*})$ となる。ここでは $b = 1$ とせよ。$n = 100$、$\sigma = 1$ とし、$\lambda_i = 1$ をもち単位ノルムの θ^* を含む 5 つの強い方向と、q 個の弱い方向（それぞれ $\lambda_i = 1/q$）を $q \in \{100, 300, 10^3, 3 \cdot 10^3, 10^4\}$ に対して用いよ。測定された分散を $\sigma^2 (k^*/n + n/\varrho_{k^*})$ と比較せよ。$q = 100$ では何が起き、それはなぜか。
   3. べき乗則スペクトル $\lambda_i = i^{-\alpha}$（$p = 40n$ で打ち切り）に対して、$\alpha \in \{0.5, 1, 2\}$、$\theta^* = \mathbf{e}_1$、$n \in \{50, 100, 200, 400\}$ として同様の実験を行え。どのスペクトルで過剰リスクが n とともに減少するか。それ以外のスペクトルではバイアスと分散のどちらが破綻し、ρ_0/n、k^*/n、および n/ϱ_{k^*} はそれをどのように表しているか。

8. **幅ではなくノルム：2 層ネットワーク**。●●$\mathcal{F}_1 = \{\mathbf{x} \mapsto \mathrm{ReLU}(\langle \mathbf{v}, \mathbf{x} \rangle) : \|\mathbf{v}\|_2 \leq B_2\}$ および $\mathcal{F}_2 = \{\mathbf{x} \mapsto \sum_{j=1}^{h} w_j \, \mathrm{ReLU}(\langle \mathbf{v}_j, \mathbf{x} \rangle) : \|\mathbf{w}\|_1 \leq B_1, \, \|\mathbf{v}_j\|_2 \leq B_2\}$ を、$\|\mathbf{x}_i\| \leq r$ を満たす点上で考える。

   1. $\widehat{\mathfrak{R}}_S(\mathcal{F}_1) \leq B_2 r / \sqrt{n}$ を示せ。
   2. $\mathcal{F}_2$ が、凸包を B_1 倍した集合（$\mathcal{F}_1 \cup (-\mathcal{F}_1)$ の凸包の定数倍）に含まれること、およびクラスの凸包をとっても経験ラデマッハ複雑度は変わらないことを示せ。ヒント：

$\sum_j w_j f_j = \|\mathbf{w}\|_1 \sum_j \frac{|w_j|}{\|\mathbf{w}\|_1} (\pm f_j)$。

3. $0 \in \mathcal{F}_1$ を用いて $\mathbb{E}_\varepsilon \sup_{f \in \mathcal{F}_1} |\sum_i \varepsilon_i f(\mathbf{x}_i)| \le 2n \widehat{\mathfrak{R}}_S(\mathcal{F}_1)$ を示し、$\widehat{\mathfrak{R}}_S(\mathcal{F}_2) \le 2B_1 B_2 r/\sqrt{n}$ を導け。
4. 幅 h はこの上界に現れない。この上界は両方の層に対する重み減衰について何を示唆するか。また、なぜこれでは 4.5.1.1 節のランダムラベル実験を説明できないのか。

36.11 ナイーブベイズ

3.3.5 節では、画像が与えられたときのラベルの確率に対して直接勾配降下法を実行し、Fashion-MNIST 上でソフトマックス回帰を訓練した。この当てはめでは 7,850 個のパラメータ（各ピクセルと各クラスの組に対する重み 1 つずつと、10 個のオフセット）が得られ、約 83% の正解率を達成し、3.4 節の較正（calibration）チェックも通過した。しかし、分類器を構築する方法は勾配降下法だけではない。代わりに、典型的なコートやサンダルがどのような見た目をしているかを捉えることで各クラスがどのように画像を生成するかを記述し、ベイズの定理によってその記述を逆転させて予測を得ることもできる。この別のアプローチはどの程度うまく機能し、それを選ぶことで何を失うのだろうか。

直ちに立ちはだかる障害はスケールの問題である。コートの見た目を記述するとは、あり得るすべての画像に確率を割り当てることを意味する。たとえ各ピクセルを 1 ビット（塗られているか空白か）に単純化したとしても、あり得る 28×28 画像の総数は観測可能な宇宙に存在する原子の数をはるかに上回るのに対し、手元にある訓練画像は 60,000 枚にすぎない。**ナイーブベイズ**分類器は、たった 1 つの大胆な仮定によってこの障害を回避する。すなわち、クラスがひとたび判明すれば、各特徴量は互いについて何の情報ももたらさないと仮定するのである。

この着想は、含まれる単語によって文書を分類した Maron の初期の研究にさかのぼり (Maron 1961)、1990 年代のスパムフィルタを支えた (Sahami et al. 1998)。独立性の仮定のもとでは、スパムフィルタは各単語がスパムメールと通常メールでそれぞれどの程度の頻度で現れるかさえ知っていればよく、衣類分類器は各クラスで各ピクセルがどの程度の頻度で塗られているかさえ知っていればよい。訓練は数え上げに帰着し、予測は乗算に帰着する。

ナイーブベイズは、各クラスがどのように入力を生成するかというモデルから構築される最も単純な**生成**（generative）分類器である。オプティマイザを使わずにデータを 1 回走査するだけで当てはめが完了し、欠損した特徴量も対応する因子を取り除くだけで簡潔に扱える。その独立性の仮定は、現実世界のほとんどのデータと同様に画像に対しても明らかに成り立たないが、誤った独立性の仮定が分類器の判定や確率にどのような影響を与えるかをこれほど明快に示してくれるモデルはほかにない。

本節は本章のケーススタディという位置づけにある。36.10 節の汎化限界（generalization bound）の後ではここでの数学は軽めであるが、本章で学んだ道具立てを 1 つの具体的なモデルの上に結集させることに本節の意義がある。最尤法（maximum likelihood）でパラメータを当てはめ、ベータ事後分布でそれらを平滑化し、ブートストラップ法で正解率にエラーバーを付け、信頼性チェックでその確率を検証する。

以下の実装では通常の NumPy のみを用い、深層学習フレームワークからはデータローダだけを利用する。

```
d2l.use_svg_display()
```

36.11.1 分類のためのベイズの定理

入力が与えられたときのラベルの確率に至るには 2 つのルートがある。本書のこれまでの章では、その確率を直接学習した。本節では、入力のモデルを経由する間接的なルートをたどり、一般のままではなぜ計算困難になるのかを確認したうえで、この分類器の名前の由来である仮定によってそれを実用可能にする。

36.11.1.1 生成ルート

分類器はラベル $y \in \{1, \dots, K\}$ を入力 $\mathbf{x}$ に割り当てる。もし真の事後分布 $p(y \mid \mathbf{x})$（入力が与えられたときの各ラベルの確率）が既知であれば、最も高い頻度で正解する決定規則は、単に最も確率の高いラベルを選ぶものとなる。

$$\hat{y} = \operatorname*{argmax}_{y} \, p(y \mid \mathbf{x}). \tag{36.133}$$

Fashion-MNIST では $K = 10$ である。784 個の各ピクセルを塗られているか空白かで記録すると、$\mathbf{x}$ は 2^{784} 通りの二値パターンのいずれかになる。$784 \log_{10} 2 = 236.0$ であるから、これは約 10^{236} 通りの異なる画像に相当する。$p(y \mid \mathbf{x})$ のルックアップテーブルを作ろうとすればすべてのパターンに対して 1 行が必要になるが、60,000 枚の訓練画像ではそのほとんどすべてが空欄のまま残ってしまう。

ベイズの定理（36.2 節）により、事後分布は $p(y \mid \mathbf{x}) = p(\mathbf{x} \mid y)\, p(y) / p(\mathbf{x})$ と書き換えられる。分母の $p(\mathbf{x})$ は y に依存しないため、どのクラスが最大値を達成するかは変わらない。

$$\hat{y} = \operatorname*{argmax}_{y} \, p(\mathbf{x} \mid y)\, p(y). \tag{36.134}$$

各クラスについて、分類器はそのクラスのもとで画像がどれほど生じやすいかに、そのクラス自体が全体でどれほど一般的かを掛け合わせる。この積 $p(\mathbf{x} \mid y)\, p(y)$ を**ベイズ分子**（Bayes numerator）と呼ぶ。K 個の分子をそれらの総和で割れば $p(\mathbf{x})$ が復元され、完全な事後分布が得られる。

これにより、分類は大きく 2 つの系統に分かれる。**生成分類器**は、クラス条件付き分布（class-conditional distribution）$p(\mathbf{x} \mid y)$ と事前分布（prior）$p(y)$ を組み合わせて各クラスの入力がどのように生じるかをモデル化し、ベイズの定理を適用する。一方、**識別分類器**（discriminative classifier）は $p(y \mid \mathbf{x})$ を直接モデル化し、入力そのものがどのように生成されるかは一切記述しない。ソフトマックス回帰（3.1 節）はその典型例である。

ベイズの定理を使ってもスケールの問題は何も解決していないのではないか、すなわち $p(\mathbf{x} \mid y)$ は依然として 2^{784} 通りのパターン上の分布ではないか、と疑問に思うのも無理はない。その指摘は正しく、確率の連鎖律を使えば組合せ爆発がどこに潜んでいるかを正確に特定できる。特徴量を 1 つずつ展開すると、次のようになる。

$$p(\mathbf{x} \mid y) = p(x_1 \mid y)\, p(x_2 \mid x_1, y) \cdots p(x_d \mid x_1, \dots, x_{d-1}, y). \tag{36.135}$$

各因子はそれより前の特徴量の全履歴に条件付けられているため、最後の因子だけでも 2^{d-1} 通りの履歴にわたるテーブルが必要になる。次項では、これらの履歴をすべて切り捨てる。

36.11.1.2 ナイーブな仮定

現実的な規模のデータセットから $p(\mathbf{x} \mid y)$ を推定するには、パラメータ数を大幅に減らす必要がある。ナイーブな仮定を平易な言葉で言えば、衣類のクラスがひとたびわかれば、ある 1 つのピクセルが塗られていると知っても他のどのピクセルについてもそれ以上の情報は得られない、ということである。記号で表せば、各特徴量は**ラベルが与えられたもとで条件付き独立**（conditionally independent given the label）であり、連鎖律の各因子は自身の履歴を忘れる。

$$p(\mathbf{x} \mid y) = \prod_{i=1}^{d} p(x_i \mid y). \tag{36.136}$$

二値ピクセルの場合、各因子はたった 1 つの数値、すなわちピクセル i がクラス y において塗られている確率によって決まる。制約のない $p(\mathbf{x} \mid y)$ はクラスごとに $2^{784} - 1$ 個の自由な確率を必要とするのに対し、因数分解されたモデルが必要とするのは 784 個だけである。これら 784 個の数値はそれぞれ通常の頻度であり、クラスあたり 6,000 枚の訓練画像があれば高々 $0.5/\sqrt{6000} \approx 0.0065$ の標準誤差で推定できる（36.5 節）。また、どちらのモデルもクラス事前分布のために $K - 1 = 9$ 個の数値を必要とする。

なお、この仮定は d 個すべての特徴量に対する同時条件であり、特徴量のペアだけを調べても検証するには不十分である。ある 1 つのクラス内で、x_1 と x_2 が独立な公平ビットであり、$x_3 = x_1 \oplus x_2$ はそのうちちょうど一方が 1 のときに 1 をとるとしよう。特徴量のどのペアも独立性を満たすが、3 番目の特徴量は他の 2 つによって完全に決定される。その結果、パターン $(1, 1, 1)$ の真の確率は 0 であるにもかかわらず、3 つの周辺分布の積はこれに $\frac{1}{8}$ を割り当ててしまう（演習問題 1）。

各変数からそれが直接影響を与える変数へと矢印を引いた有向グラフとして描くと、この因数分解は星型の構造をなす。すなわち、ラベル y がすべての特徴量の親となり、特徴量同士を結ぶ辺は存在しない（Figure 36.50 左）。特徴量間の条件付き依存関係を保持するより表現力の高いモデルでは、特徴量間に辺が追加される（右）。

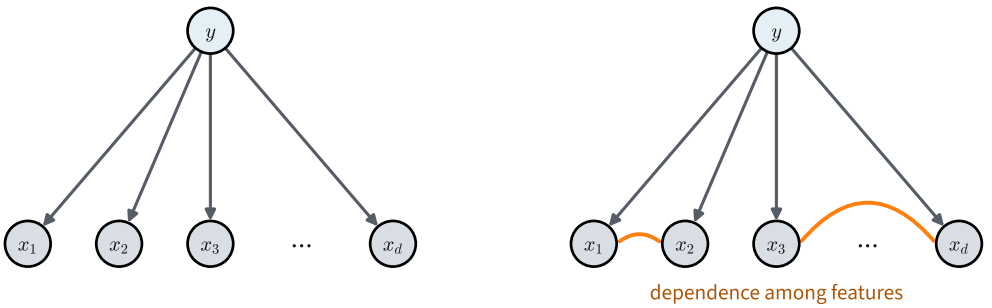


Figure 36.50: 条件付き独立の構造。左：ナイーブベイズでは、ラベル y がすべての特徴量の親であり特徴量間に辺はなく、$p(\mathbf{x} \mid y) = \prod_i p(x_i \mid y)$ を表す。右：特徴量間の条件付き依存性を保持するモデルでは、特徴量間に辺が追加される。

因数分解 Equation 36.136 をベイズの定理に代入すると、**ナイーブベイズ則**（naive Bayes rule）が得られる (Maron 1961)。

$$\widehat{y} = \operatorname*{argmax}_{y} \; p(y) \prod_{i=1}^{d} p(x_i \mid y). \tag{36.137}$$

画像に対して、この仮定が誤っていることは明らかである。衣類の隣接するピクセルは同じ袖や同じ裾に属するため、ともに塗られているか、ともに空白であるかのどちらかになりやすい。以下では、Fashion-MNIST 上でこの仮定がどれほど大きく破綻しているか、そしてその破綻が分類器に何をもたらすかを定量的に確かめる。

36.11.1.3 2 単語のスパムフィルタ

784 ピクセルへスケールアップする前に、最小の例を手計算で追ってみよう。メッセージに単語「free」が含まれるかどうかと、「meeting」が含まれるかどうかという 2 つの二値特徴量だけを調べるメールフィルタを考える。訓練セットに 10 通のメッセージ（スパム 4 通、通常メール 6 通）があり、出現数が次のとおりであったとする。

	スパム（4 通）	通常メール（6 通）
「free」を含む	3	1
「meeting」を含む	0	4

単純な数え上げだけで、モデルが必要とするすべての数値が揃う。事前分布は $\widehat{p}(\text{spam}) = 4/10 = 0.4$ である。単語「free」はスパムの $3/4$ と通常メールの $1/6$ に現れ、「meeting」はスパムの $0/4$ と通常メールの $4/6$ に現れる。ここで、新しいメッセージが「free」を含み、「meeting」を含まないとする。その 2 つのベイズ分子は次のようになる。

$$\text{spam: } 0.4 \cdot \tfrac{3}{4} \cdot \big(1 - \tfrac{0}{4}\big) = 0.30, \qquad \text{ordinary: } 0.6 \cdot \tfrac{1}{6} \cdot \big(1 - \tfrac{4}{6}\big) = \tfrac{1}{30} \approx 0.033,$$

これより、スパムの事後確率（事後分布）は $0.30/(0.30 + 0.033) = 0.90$ となる。同じ計算はオッズで表すとさらに見通しがよい。観測された各特徴量は、事前オッズにその尤度比（likelihood ratio）、すなわち観測された事象がスパムのもとで生じる確率を通常メールのもとで生じる確率で割った値を掛ける。

$$\begin{aligned} \frac{p(\text{spam} \mid \mathbf{x})}{p(\text{ordinary} \mid \mathbf{x})} &= \underbrace{\frac{0.4}{0.6}}_{\text{prior}} \cdot \underbrace{\frac{3/4}{1/6}}_{\text{"free"}} \cdot \underbrace{\frac{1}{2/6}}_{\text{no "meeting"}} \\ &= \frac{2}{3} \cdot 4.5 \cdot 3 = 9. \end{aligned}$$

9 対 1 のオッズは確率 $9/10$ に対応する。以下で構築する 784 ピクセルの分類器も、尤度比が 2 個ではなく 784 個になるだけで、これとまったく同じ計算を実行する。この小さなフィルタには、カウントが 0 になるとき、対数へ移行するとき、そして特徴量が二重に数えられるときの計 3 回、改めて立ち返る。

36.11.2 数え上げによる学習

ナイーブベイズ則には、クラス事前分布と、クラスおよび特徴量の組ごとに 1 つの確率が必要である。これらの推定値はどこから得られるのか、またコンピュータは数値的なアンダーフローで値を失うことなく 785 個の確率の積をどのように計算できるのだろうか。

36.11.2.1 最尤法と平滑化

先ほどのスパムフィルタでは、観測された頻度をそのまま代入した。最尤法はこの選択を正当化すると同時に、それがどこで破綻するかも明らかにする。$\theta_{iy} = p(x_i = 1 \mid y)$ を、特徴量 i がクラス y において活性化する確率とする。クラス y が n_y 回、全 n 個の訓練例の中に現れ、特徴量 i がそのうち n_{iy} 個の例で活性化しているとする。ナイーブベイズの因数分解のもとでは、訓練セット全体の対数尤度はクラス事前分布に関する 1 つの項と、クラスおよび特徴量の組ごとに独立な項へと分解される。

$$\sum_y n_y \log p(y) + \sum_y \sum_{i=1}^{d} \Big[n_{iy} \log \theta_{iy} + (n_y - n_{iy}) \log(1 - \theta_{iy}) \Big]. \tag{36.138}$$

角括弧内の各項は、n_{iy} 回の表を、n_y 回投げたバイアス θ_{iy} のコインから得る対数尤度にほかならないため、すべてのパラメータを個別に当てはめることができる。コインのバイアスの最尤推定量は観測頻度であるから（36.6 節）、$\hat{\theta}_{iy} = n_{iy}/n_y$ および $\hat{p}(y) = n_y/n$ が得られる。

しかし、生の頻度には深刻な落とし穴がある。特徴量 i がクラス y の訓練例に一度も現れなければ $\hat{\theta}_{iy} = 0$ となり、将来特徴量 i が活性化した例が与えられると、他のすべての特徴量が何を示していようとベイズ分子として 0 がクラス y に割り当てられる。たった 1 つの特徴量がそのクラスを棄却してしまうのである。2 単語のスパムフィルタにもすでにそのようなゼロが含まれている。「meeting」は訓練用のスパムメール 4 通に一度も現れなかったため、「free」と「meeting」の両方を含むメッセージのスパム分子は $0.4 \cdot \frac{3}{4} \cdot 0 = 0$ となり、100% の確信をもって通常メールと判定されてしまう。Fashion-MNIST のピクセルがあるクラスで一度も塗られないことがどの程度の頻度で起きるかは、後ほど確認する。

標準的な対処法は**ラプラス平滑化**（Laplace smoothing）である (Laplace 1814)。その特徴量が活性化した仮想的な例を 1 つ、非活性な仮想的な例を 1 つ加えることで、次の推定値を得る。

$$\hat{\theta}_{iy} = \frac{n_{iy} + 1}{n_y + 2}. \tag{36.139}$$

これにより、すべての推定値は $\frac{1}{2}$ の方向へわずかに引き寄せられ、推定確率が 0 や 1 になることは決してなくなる。

ラプラス平滑化には明快なベイズ的解釈もある。θ に一様な事前分布をおいたもとで、n_{iy} 回の成功が n_y 回の試行において観測されると事後分布 $\mathrm{Beta}(n_{iy} + 1,\, n_y - n_{iy} + 1)$ が得られ、その平均はちょうど $(n_{iy} + 1)/(n_y + 2)$ となる（36.4 節）。Laplace はこれを継起の法則（rule of succession）と呼んだ。

2 単語のスパムフィルタに戻ると、平滑化によって「free」と「meeting」の推定確率はスパムでは $4/6$ と $1/6$ に、通常メールでは $2/8$ と $5/8$ に変わる一方、事前分布は 0.4 のままである。両方の単語を含むメッセージのベイズ分子は $0.4 \cdot \frac{4}{6} \cdot \frac{1}{6} = 0.044$ と $0.6 \cdot \frac{2}{8} \cdot \frac{5}{8} = 0.094$ になり、スパム確率は 0.32 となって平滑化なしの 0 にはならない。その一方で、最初のメッセージ（「free」を含み「meeting」を含まない）の確率は 0.90 から 0.80 へと下がる。これは、スパムが会議（meeting）に決して言及しないという証拠としてはスパム 4 通では控えめにすぎないという事実を反映している。

36.11.2.2 対数空間におけるスコア

平滑化を施しても、ナイーブベイズ則をそのまま評価すると $d+1$ 個の数値（いずれも 0 から 1 の間）を掛け合わせることになり、画像ではその積は極端に小さくなる。仮にすべての因子が $\frac{1}{2}$ であったとすると、784 個の積は $2^{-784} \approx 10^{-236}$ となる。正の最小の単精度浮動小数点数は約 $1.4 \cdot 10^{-45}$、正の最小の倍精度浮動小数点数は約 $4.9 \cdot 10^{-324}$ であるため、単精度では積が直ちに 0 へとアンダーフローする。倍精度であれば 2^{-784} には耐えられるが、以下で計算する実際の画像のスコアには 10^{-1000} 付近まで落ち込むものもあり、倍精度でも持ちこたえられない。

対数をとれば、積は和に変わる。log は狭義単調増加であるため、最大となるクラスは変化しない（この手法一般については 35.8 節で論じている）。ベイズ分子の対数として**クラススコア**（class score）$s_y(\mathbf{x})$ を定義すると、決定規則は次のようになる。

$$\hat{y} = \operatorname*{argmax}_y \; s_y(\mathbf{x}), \qquad s_y(\mathbf{x}) = \log p(y) + \sum_{i=1}^{d} \log p(x_i \mid y). \tag{36.140}$$

785 個の確率の積が 785 個の対数の和に置き換わり、どの確率も 0 でない限り（これは平滑化によって保証される）すべての項は有限の値に留まる。平滑化なしのスパムフィルタでは、最初のメッセージの対数スコアは $\log 0.30 = -1.20$ と $\log \frac{1}{30} = -3.40$ であり、その差 $2.20 = \log 9$ は先ほど手計算で求めた事後オッズの対数にほかならない。

36.11.2.3 線形決定境界

二値特徴量の場合、クラススコアはきわめて見慣れた形に簡約され、その形からナイーブベイズがどのような分類器を表現できるかが明らかになる。二値特徴量は活性化しているときに $\log \theta_{iy}$、非活性のときに $\log(1-\theta_{iy})$ だけ寄与するため、$\log p(x_i \mid y) = x_i \log \theta_{iy} + (1-x_i)\log(1-\theta_{iy})$ と書ける。x_i が掛かる項と掛からない項をまとめると、次の式が得られる。

$$s_y(\mathbf{x}) = \mathbf{w}_y^{\top}\mathbf{x} + b_y, \qquad w_{iy} = \log \frac{\theta_{iy}}{1-\theta_{iy}}, \qquad b_y = \log p(y) + \sum_{i=1}^{d} \log(1-\theta_{iy}). \tag{36.141}$$

塗られた各ピクセルは w_{iy}、すなわちピクセル i がクラス y で塗られる対数オッズ（log-odds）を加算し、オフセット b_y は完全に空白の画像のスコアを表す。どのクラススコアも $\mathbf{x}$ のアフィン関数であるため、任意の 2 クラス間でスコアが等しくなる決定境界（decision boundary）は超平面となる (Bishop 2006)。これが本節で覚えておくべき 2 つの結果のうちの 1 つ目である。すなわち、ベルヌーイナイーブベイズはソフトマックス回帰とまったく同じモデル族に属する線形分類器であり、パラメータ配列の形状も完全に一致する（784×10 個の重みと 10 個のオフセット）。両手法の違いは、頻度を数え上げるか $p(y \mid \mathbf{x})$ に対して勾配降下法を実行するかという、それらの数値の決め方だけにある。2 クラスの場合、事後分布はスコア差のシグモイド $p(y=1 \mid \mathbf{x}) = \operatorname{sigmoid}\big((\mathbf{w}_1 - \mathbf{w}_0)^{\top}\mathbf{x} + b_1 - b_0\big)$ に帰着し、これはロジスティック回帰と同一の形式である（演習問題 2）。

境界が線形のまま保たれるかどうかは特徴量の分布に依存する。ガウスナイーブベイズは、各連続特徴量 x_i をクラス y において平均 μ_{iy}、分散 σ_{iy}^2 のガウス分布としてモデル化し、クラススコアに $-(x_i-\mu_{iy})^2/(2\sigma_{iy}^2)$ だけ寄与させる。すべての特徴量が全クラスで同じ分散を共有していれば、x_i^2 の項はどのスコアでも同一となってあらゆる 2 クラス比較で相殺され、線形な境界が残る。クラス固有の分散を持つ場合には x_i^2 の項が生き残り、境界は各 x_i について二次になるが、交差項 $x_i x_j$ は依然として現れない（演習問題 3）。

ガウス分布の場合を見ると、当てはめ手法の違いがどれほど重要かを視覚的に理解できる。Figure 36.51 では、各クラス内で 2 つの特徴量が強く相関しており、両方の点群が斜めに傾いている。ナイーブベイズは各クラスに座標軸に沿ったガウス分布を当てはめる（両クラスで分散が等しいため、境界は直線になる）。座標軸に沿った楕円はこの傾きを無視するため、得られる境界は両方の点群を横切ってしまい、19% の点を誤分類する。右のパネルは、十分なデータを用いて $p(y \mid \mathbf{x})$ を直接当てはめることで得られた線形境界を示している。この境界は傾きに沿って配置され、7% の誤り率を達成する。これはこの問題における最適値であり、いかなる種類の分類器もこれを上回ることはできない。

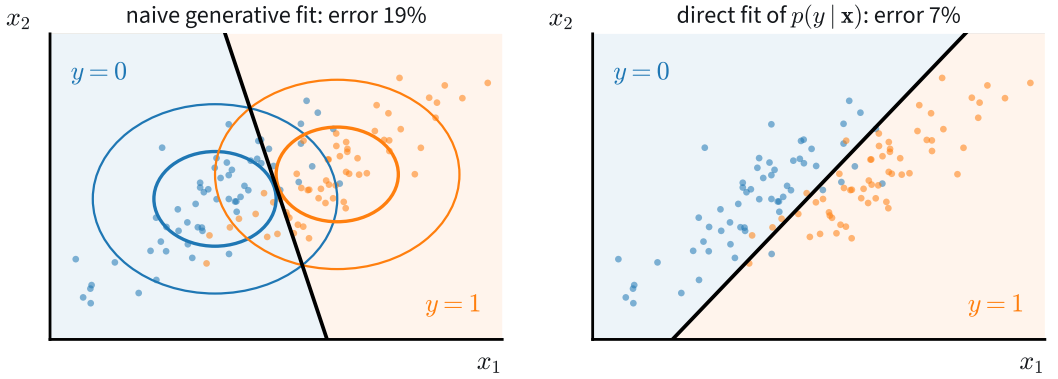


Figure 36.51: 各クラス内で 2 つの特徴量が相関している同一データ上で線形分類器に至る 2 つのルート。左：ナイーブな生成モデルの当てはめは、各クラスを独立な特徴量を持つガウス分布（標準偏差の 1 倍および 2 倍の楕円）としてモデル化し、ベイズの定理によって境界を導出する。右：$p(y \mid \mathbf{x})$ に直接当てはめた線形境界。誤り率は等しい事前分布に対する母集団での値である。

歴史的には、ナイーブベイズの代表的な応用先はピクセルではなくテキストであった。実際のスパムフィルタは、先ほどの 2 単語の例を数万語へと拡大したものにすぎない (Manning et al. 2008)。文書はどの単語が出現するかによっても、各単語が何回出現するかによっても表現でき、クラススコアはいずれの場合も線形になる。本節では引き続きピクセルを扱うが、演習問題 4 ではこれら 2 つのテキスト事象モデルを導出し、同一の文書に対して両者の判定が食い違い得ることを示す。

36.11.3 数え上げによる Fashion-MNIST 分類

2 単語のスパムフィルタであれば手計算で容易に確かめられる。しかし、目に見えて相関している 784 個のピクセルに対しても同じ単純な数え上げが通用するかどうかは実際に試してみるほかなく、Fashion-MNIST を使えばまったく同じ画像上でソフトマックス回帰と結果を直接比較できる。

36.11.3.1 二値ピクセル

Fashion-MNIST (Xiao et al. 2017) は 10 種類の衣類クラスにわたる 60,000 枚の訓練画像と 10,000 枚のテスト画像からなり、各画像は 28×28 のグレースケールグリッドである（3.2 節）。背景ピクセルは厳密に 0 であるため、$x_i = 1$ をピクセル i が正（衣類の一部）である場所に設定し、背景上では $x_i = 0$ とおくことで、各画像を二値ベクトルに変換できる。ローダから両分割の生画像を取得し、784 個の二値特徴量からなるベクトルへと平坦化する。

```
data = d2l.FashionMNIST()
def binarize(images):  # 1 where the pixel belongs to the garment
    return (images.reshape(-1, 784) > 0).astype(np.float32)

X, Y = binarize(data.train.data.numpy()), data.train.targets.numpy()
X_test, Y_test = binarize(data.val.data.numpy()), data.val.targets.numpy()
print(X.shape, X_test.shape, f'ink fraction {X.mean():.2f}')
```

```
(60000, 784) (10000, 784) ink fraction 0.50
```

全ピクセルのおよそ半分が塗られている。どのクラスも 6,000 枚の訓練画像と 1,000 枚のテスト画像を持つため、推定される事前分布は一様な 0.1 となり、すべてのクラススコアに同じ定数を加えるだけであるが、実装の一般性を保つためコードには事前項を残しておく。

36.11.3.2 テンプレートとサンプル

訓練に必要なのは、各クラスの画像枚数 n_y と、そのうち各ピクセルが塗られている画像の枚数 n_{iy} という 2 種類のカウントだけである。以下のセルでは、ラプラス平滑化を施した推定値 $\hat{\theta}_{iy}$ をクラスごとに 1 行として `P_xy` に格納する。同時に、平滑化に関して先ほど提起した実務上の疑問、すなわちあるクラスの訓練画像で一度も塗られないピクセルがいくつあり、そのうちのいずれかが塗られているテスト画像が何枚あるかにも答える。

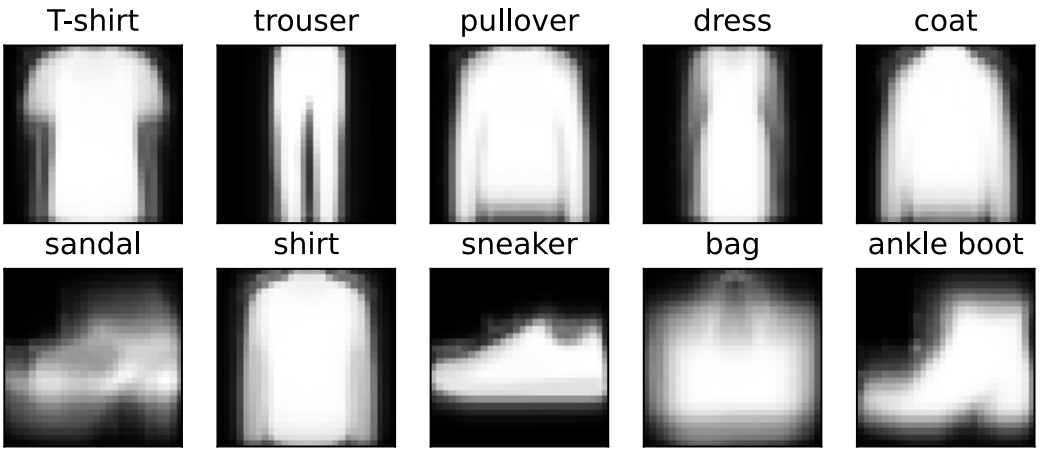
```
n_y = np.bincount(Y, minlength=10)              # images per class
n_x = np.stack([X[Y == y].sum(axis=0) for y in range(10)])
P_y = n_y / n_y.sum()                           # prior p(y)
P_xy = (n_x + 1) / (n_y + 2)[:, None]           # smoothed theta_iy
print('prior:', P_y)
print('never-inked pixels per class:', (n_x == 0).sum(axis=1))
unseen = (X_test * (n_x[Y_test] == 0)).sum(axis=1) > 0
print('test images inking one of their class:', unseen.sum())
```

```
prior: [0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]
never-inked pixels per class: [ 0 13  1  0  3  3  0 57  0  1]
test images inking one of their class: 5
```

ここではゼロカウントはかなりまれである。10 クラス中 4 クラスでは 1 つもなく、ズボンでは 13 個、スニーカーでは 57 個である（スニーカーは平たい形状をしており、画像フレームの上部が空白のままになるためである）。テストセット全体でも、正解クラスが訓練中に一度も塗らなかったピクセルが塗られている画像は 5 枚しかない。しかし平滑化がなければ、これら 5 枚の画像はいずれも正解クラスに対するスコアが $-\infty$ となり、残りの 783 ピクセルがどれほど強くそのクラスを支持していても必ず誤分類されてしまう。

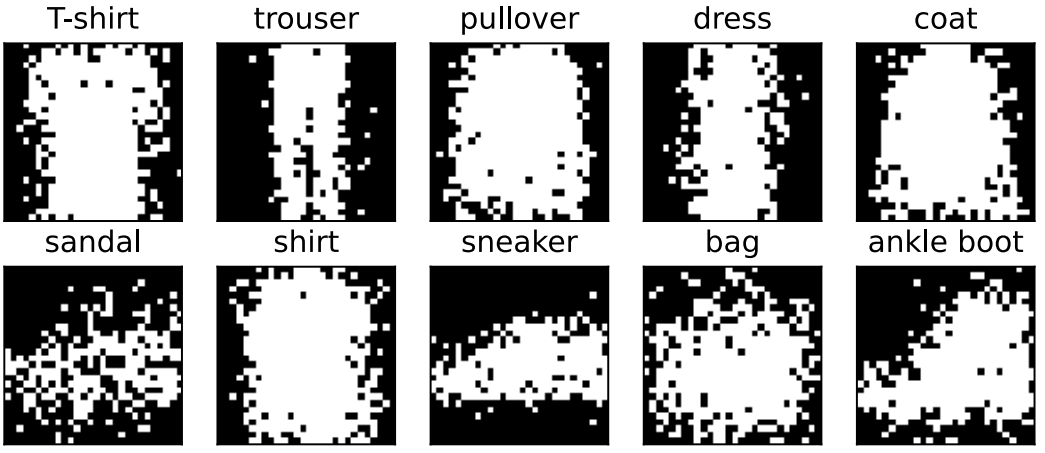
`P_xy` の各行は訓練画像のうち各ピクセルが塗られている割合を記録しているため、10 個の行を 28×28 の画像に形状変換して戻せば、モデルが各衣類クラスについて何を学習したかを視覚化できる。

```
d2l.show_images(P_xy.reshape(10, 28, 28), 2, 5,
                titles=data.text_labels(range(10)), cmap='gray');
```



得られるテンプレートは、10 種類の衣類の柔らかくぼやけた輪郭のように見える。各テンプレートは単にそのクラスのピクセルごとの平均であるため、このぼやけは衣類の形状や位置の実際のばらつきを反映している。独立性を仮定するかどうかによらず、データのピクセルごとの頻度に一致するモデルであればまったく同じテンプレートを生成するため、このぼやけは仮定ではなくデータ自体に由来するものである。ナイーブな仮定が付け加えるのは、これらの平均と事前分布さえあれば知るべきことはすべて尽きているという主張である。その主張が何を意味するかを確かめるため、すべてのピクセルについて確率 $\hat{\theta}_{iy}$ の独立なコインを投げることで、生成モデルから合成画像をサンプリングしてみよう。

```
rng = np.random.default_rng()
samples = (rng.random((1000, 10, 784)) < P_xy).astype(np.float32)
d2l.show_images(samples[0].reshape(10, 28, 28), 2, 5,
                titles=data.text_labels(range(10)), cmap='gray');
```



各サンプルは全体の大まかなシルエットこそ正しく捉えているものの、境界上のすべてのピクセルがそれぞれ独立なコイン投げで決まるため、縁がギザギザで斑点状に散らばっている。この視覚的な違いは直接定量化できる。実際の画像の 1 行を横に走査すると、塗られた部分（インク）と背景は数回しか切り替わらないが、モデルのサンプルでははるかに頻繁に切り替わるはずである。以下では、訓練セット全体とクラスあたり 1,000 枚の生成サンプルとで、全体のインク率と 1 行

あたりのインク・背景間の平均切り替わり回数の両方を比較する。

```
def switches(images):  # ink/background changes along a row, mean per row
    return np.abs(np.diff(images.reshape(-1, 28, 28), axis=2)).sum(axis=2).mean()

print(f'ink fraction: data {X.mean():.3f}, samples {samples.mean():.3f}')
print('switches per row, data:   ',
      np.round([switches(X[Y == y]) for y in range(10)], 1))
print('switches per row, samples:',
      np.round([switches(samples[:, y]) for y in range(10)], 1))
```

```
ink fraction: data 0.498, samples 0.498
switches per row, data:    [3.8 3.2 3.1 3.1 3.1 3.  3.3 1.5 3.1 2.6]
switches per row, samples: [5.1 4.1 4.8 4.7 4.8 8.2 5.3 4.  8.2 5.9]
```

すべてのピクセルが正しい周辺確率を持つ以上当然ではあるが、サンプルが持つ全体のインク量は実データと一致している。それにもかかわらず、インクと背景の間の切り替わりははるかに頻繁に起きており、プルオーバーでは 1 行あたり約 3 回ではなく約 5 回、サンダルやバッグでは約 3 回ではなく約 8 回に達する。実際の衣類が滑らかな輪郭を持つのは隣接するピクセル同士が同じ値をとりやすいためであり、この局所的な一致こそまさに独立性の仮定が切り捨ててしまうものにほかならない。

36.11.3.3 仮定の検証

生成されたサンプルは画像全体で独立性の仮定が破綻していることを示しているが、たった 1 組のピクセルペアを見るだけでも具体的な数値でそれを裏付けられる。プルオーバーの下裾に沿った 26 行目の 14 列目と 15 列目にある隣接する 2 つのピクセルを考えよう。もしこれら 2 つのピクセルがクラスが与えられたもとで条件付き独立であれば、両方が塗られているプルオーバーの割合はそれぞれのインク率の積に等しくなるはずである。訓練用のプルオーバーとサンプリングされた 1,000 枚のプルオーバーの両方について、その積と実際の同時頻度を比較してみよう。

```
i, j = 26 * 28 + 14, 26 * 28 + 15     # two neighbouring hem pixels
for name, Z in (('data', X[Y == 2]), ('model', samples[:, 2])):
    p_i, p_j, both = Z[:, i].mean(), Z[:, j].mean(), (Z[:, i] * Z[:, j]).mean()
    print(f'{name:5s}: P(i)={p_i:.2f}  P(j)={p_j:.2f}  '
          f'P(i)P(j)={p_i * p_j:.2f}  P(i,j)={both:.2f}')
```

```
data : P(i)=0.55  P(j)=0.55  P(i)P(j)=0.30  P(i,j)=0.54
model: P(i)=0.54  P(j)=0.55  P(i)P(j)=0.30  P(i,j)=0.31
```

どちらのピクセルも約 55% のプルオーバーで塗られている。独立性のもとでは、両方が同時に塗られる割合は $0.55 \times 0.55 \approx 30\%$ になるはずである。ところが実データでは、54% のプルオーバーで両方が塗られており、片方のピクセル単独とほぼ変わらない頻度になっている。プルオーバーの裾は 26 行目に届くかその手前で終わるかのどちらかであるため、これら 2 つの隣接ピクセルはほぼ 1 つの特徴量が 2 回記録されたかのように振る舞う。一方、モデルのサンプルは構成上その積に一致する。

36.11.3.4 テスト正解率

テストセットを分類するため、アフィン形式 Equation 36.141 を用いる。`P_xy` と `P_y` から重み w_{iy} とオフセット b_y を計算し、1 回の行列積ですべてのテスト画像に対して全 10 クラスのスコアを求め、続く分析のために予測結果を保存する。ソフトマックス回帰はこれとまったく同じ形

式のスコアで約 83% の正解率に達したが、代わりに数え上げを使うことでどれほど正解率を譲ることになるのだろうか。

```
W = np.log(P_xy) - np.log1p(-P_xy)              # w_iy, log-odds per pixel
b = np.log1p(-P_xy).sum(axis=1) + np.log(P_y)   # b_y, score of a blank image
s = X_test @ W.T + b                            # s_y(x), shape (10000, 10)
pred = s.argmax(axis=1)
print(f'test accuracy {(pred == Y_test).mean():.4f}, training accuracy '
      f'{((X @ W.T + b).argmax(axis=1) == Y).mean():.4f}')
```

```
test accuracy 0.7059, training accuracy 0.7171
```

ナイーブベイズは約 71% のテスト画像を正しく分類しており、ランダムな当て推量のベースラインである 10% を大きく上回る。数え上げには乱数性が一切入らないため、この数値は実行ごとに完全に決定論的である。サンプリングに伴う不確実性のエラーバーについてはすぐ後で付与する。

訓練正解率も同様に示唆に富んでいる。当てはめに用いた 60,000 枚の画像に対する正解率は 72% であり、未知のテスト画像に対する値よりわずか 1 パーセントポイント高いにすぎない。訓練セットに過学習したモデルであれば両者の間に大きな乖離が生じるが、ここではそれがほとんど見られない。したがって、誤りの原因は訓練データの不足ではない。モデルの仮定が制約的すぎるために、新しい画像に対するのと同じ誤りを自身の訓練セット上でも犯しているのである。36.5 節の言葉で言えば、誤差は分散ではなく統計的バイアス（bias）に支配されている。画像をさらに集めても状況はほとんど改善しない。必要なのはより良いパラメータの組であり、以下で行うソフトマックス回帰との比較によって、まったく同じモデル族の中にはるかに優れた線形分類器が存在することが確認できる。

また、これらの対数スコアを使えば、仮に生の積 Equation 36.137 をそのまま評価していたら何が起きていたかも確認できる。各スコアはベイズ分子の自然対数であるため、$\log 10$ で割れば底 10 の指数が得られる。各分子を、正の最小の単精度数、および倍精度が精度を失い始める境界である正の最小の倍精度**正規化数**と比較してみよう。

```
log10 = s / np.log(10)                    # decimal exponent of each numerator
print(f'log10 of the numerators: largest {log10.max():.0f}, smallest '
      f'{log10.min():.0f}, median best {np.median(log10.max(axis=1)):.0f}')
f32 = -149 * np.log(2)     # log of 2**-149, the smallest positive float32
f64 = -1022 * np.log(2)    # log of 2**-1022, the smallest normal float64
print(f'images whose best numerator underflows float32: '
      f'{(s.max(axis=1) < f32).mean():.2f}')
print(f'numerators below the smallest normal float64: {(s < f64).mean():.2f}, '
      f'images whose best one is: {(s.max(axis=1) < f64).mean():.4f}')
print(f'accuracy of the raw float64 product: '
      f'{(np.exp(s).argmax(axis=1) == Y_test).mean():.4f}')
```

```
log10 of the numerators: largest -32, smallest -1002, median best -81
images whose best numerator underflows float32: 0.96
numerators below the smallest normal float64: 0.44, images whose best one is: 0.0003
accuracy of the raw float64 product: 0.7058
```

ベイズ分子はおよそ 10^{-32} から 10^{-1000} までの範囲に広がっており、典型的な画像における最大クラスの分子は 10^{-81} 付近にある。テスト画像の 96% では 10 個の分子のうち最大のものでさえ正の最小の単精度浮動小数点数を下回るため、単精度のまま積を計算すると 10 個すべてがゼロになってまったく分類できなくなる。倍精度では全分子の 44% が最小の正規化数 float64 を下回るが、そのほとんどは敗北クラスに属するものである。最大の分子は 10,000 枚中 3 枚を除くすべ

ての画像で表現可能な範囲に留まるため、生の float64 の積による正解率は 10,000 枚中 1 枚以内の差で対数空間での正解率と一致する。とはいえ、倍精度が 784 ピクセルに耐えられる余裕はわずかにすぎない。典型的な最大スコアが 10^{-81} であることを踏まえると、入力長が 4 倍になれば float64 でもアンダーフローしてしまう。対数空間で計算すれば、あらゆるスコアが通常の浮動小数点の範囲内に無理なく収まる。

36.11.4 分類器の評価

71% という見出しの正解率を前にすると、直ちに 3 つの疑問が生じる。この推定値はどれほど正確なのか。残りの 29% はどのような取り違えによるものなのか。そして、分類器が報告する確率は信頼できるのだろうか。

36.11.4.1 エラーバー

ここでのテスト正解率は 10,000 枚のランダムなテスト画像にわたる平均であるため、別のテスト標本をとればわずかに異なる数値が得られるはずである。各テスト画像は正解か不正解かという 1 回のベルヌーイ試行の結果を与える。これは本章の冒頭で登場したコイン投げをテスト画像ごとに 1 回ずつ行うのと同じであるから、観測される正解率 $\hat{p}$ は n 回の試行の標本平均であり、標準誤差 $\sqrt{\hat{p}(1-\hat{p})/n}$ を持つ（36.5 節）。$\hat{p} = 0.706$、$n = 10{,}000$ のとき、次のようになる。

$$\sqrt{\frac{0.706 \cdot 0.294}{10^4}} \approx 0.0046, \qquad 1.96 \times 0.0046 \approx 0.009,$$

これより、上下いずれの方向にも約 0.9 パーセントポイント広がる 95% 信頼区間（confidence interval）が得られる。ブートストラップ法（36.5 節）を使えば、公式に頼らない第 2 の確認ができる。テスト画像を復元サンプリングし、各再標本で正解率を再計算して、得られる分布の中央 95% をとればよい。

```
rng = np.random.default_rng()
correct = pred == Y_test                          # per-image 0/1 outcomes
n, p = len(correct), correct.mean()
boot = correct[rng.integers(0, n, (1000, n))].mean(axis=1)  # 1,000 resamples
print(f'accuracy {p:.4f}, bootstrap 95% interval '
      f'({np.percentile(boot, 2.5):.3f}, {np.percentile(boot, 97.5):.3f})')
print(f'binomial half-width 1.96 se = {1.96 * np.sqrt(p * (1 - p) / n):.4f}')
```

```
accuracy 0.7059, bootstrap 95% interval (0.698, 0.715)
binomial half-width 1.96 se = 0.0089
```

ブートストラップ法による区間は約 0.70 から 0.715 までとなり、二項分布に基づく半幅 0.009 と一致する。したがって正解率は $71\% \pm 0.9$ ポイントと報告すべきであり、「70.59%」の小数点以下の桁はサンプリングノイズにすぎない。なお、まったく同じテストセット上で 2 つの分類器を比較する場合には両者の誤りに相関が生じるため、2 つの別々の信頼区間を比較するよりも、画像ごとの差に対する対応のある分析を行う方がはるかに検出力が高い（演習問題 5）。

36.11.4.2 混同と重み

全体の正解率は分類器がどの程度の頻度で誤るかを示すのに対し、3.4 節の混同行列（confusion matrix）は**どのように**誤るかを示す。3.3 節の慣例に従い、行を予測クラス、列を正解クラスに対応させると、各列はある衣類のテスト画像 1,000 枚がどのクラスに振り分けられたかを表す。

```
    conf = np.zeros((10, 10), dtype=int)
    np.add.at(conf, (pred, Y_test), 1)       # N[i, j]: class j predicted as i
    print(conf)
```

```
[[748   9   8  50   2   0 218   0   0   0]
 [  0 907   0  53   3   0   4   0   3   0]
 [ 14  18 665   5 372   0 232   0  14   0]
 [122  48  12 808 104   2  93   0  68   7]
 [  8  16 242  45 483   0 222   0   6   0]
 [  0   0   0   0   0 680   0   5   1   0]
 [ 83   0  50  35  25   2 167   0  20   0]
 [  1   0   0   0   0 232   1 893  55 113]
 [ 24   2  23   4  11   9  63   0 832   4]
 [  0   0   0   0   0  75   0 102   1 876]]
```

10×10 のカウント表は、最も多い誤りを抜き出すと格段に読み取りやすくなる。以下では、非対角成分のうち上位 6 個を「正解クラス→予測クラス」の形式で出力するとともに、各クラス個別の正解率（対角成分を列和で割った値）を表示する。

```
    names = data.text_labels(range(10))
    off = conf - np.diag(np.diag(conf))      # errors only
    order = np.argsort(off, axis=None, kind='stable')[::-1][:6]  # largest first
    pairs = [f'{names[t]}->{names[q]} {off[q, t]}'  # true -> predicted
             for q, t in zip(*np.unravel_index(order, off.shape))]
    print(', '.join(pairs[:3]))
    print(', '.join(pairs[3:]))
    print('per-class accuracy:', np.round(np.diag(conf) / conf.sum(axis=0), 2))
```

```
coat->pullover 372, pullover->coat 242, sandal->sneaker 232
shirt->pullover 232, shirt->coat 222, shirt->T-shirt 218
per-class accuracy: [0.75 0.91 0.66 0.81 0.48 0.68 0.17 0.89 0.83 0.88]
```

誤りは特定の組に強く集中している。コートがプルオーバーと誤分類されるのが 372 回、プルオーバーがコートとされるのが 242 回あり、シャツはプルオーバー、コート、T シャツのそれぞれと 200 回以上取り違えられ、サンダルはスニーカーと 232 回取り違えられている。ズボン（91%）とスニーカー（89%）は容易である一方、コート（48%）とシャツ（17%）ははるかに難しい。これらは 3.3 節でソフトマックス回帰が苦戦した 2 つの混同グループ、すなわち上半身の衣類 4 種と履物 3 種とまったく同じであるが、数え上げによるモデルは上半身グループで大幅に成績を落としている。ソフトマックス回帰がシャツを 40〜60% の割合で認識したのに対し、ナイーブベイズは約 6 分の 1 しか正しく認識できていない。

71% というのは、二値ピクセル上の線形分類器にとって単なる上限なのだろうか。ソフトマックス回帰はこれとまったく同じ形式の超平面を定めるが、$p(y \mid \mathbf{x})$ の交差エントロピーを最小化することで重みを決定する。当てはめ手順だけが異なる比較を行うため、同じ二値ピクセル上でミニバッチ確率的勾配降下法によりソフトマックス回帰を訓練してみよう。パラメータ更新には、ロジットに関する交差エントロピーの勾配が予測確率ベクトルからワンホットラベルを引いたものに等しいという 3.1 節の結果を用いる。

```
    rng = np.random.default_rng()
    W_sr, b_sr = np.zeros((10, 784)), np.zeros(10)
    for epoch in range(20):                  # minibatches of 256 images
```

```
    for idx in np.array_split(rng.permutation(len(X)), len(X) // 256):
        O = X[idx] @ W_sr.T + b_sr                    # logits
        E = np.exp(O - O.max(axis=1, keepdims=True))
        G = E / E.sum(axis=1, keepdims=True) - np.eye(10)[Y[idx]]
        W_sr -= 0.05 * G.T @ X[idx] / len(idx)        # mean-loss gradient
        b_sr -= 0.05 * G.mean(axis=0)
pred_sr = (X_test @ W_sr.T + b_sr).argmax(axis=1)
conf_sr = np.zeros((10, 10), dtype=int)
np.add.at(conf_sr, (pred_sr, Y_test), 1)
print(f'accuracy: softmax regression {(pred_sr == Y_test).mean():.3f}, '
      f'naive Bayes {(pred == Y_test).mean():.3f}')
print('pullover<->coat errors: softmax regression',
      conf_sr[2, 4] + conf_sr[4, 2], 'naive Bayes', conf[2, 4] + conf[4, 2])
```

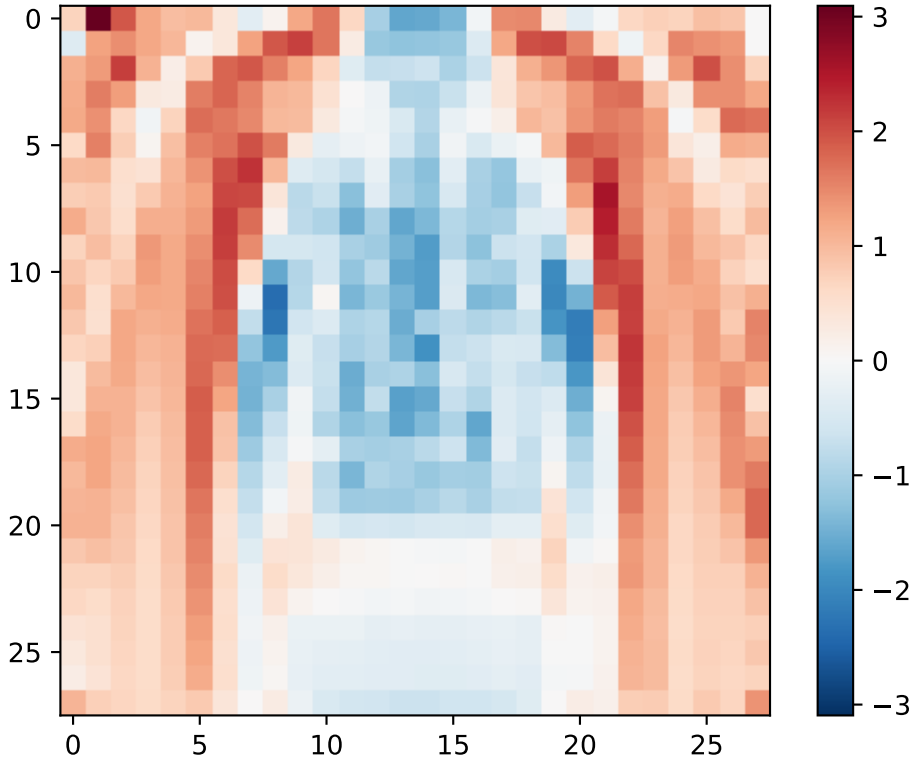
```
accuracy: softmax regression 0.830, naive Bayes 0.706
pullover<->coat errors: softmax regression 312 naive Bayes 614
```

ソフトマックス回帰はナイーブベイズを 12 パーセントポイント上回る約 83% の正解率に達し、プルオーバーとコートの間の誤りもおよそ半減させる。したがって、線形モデル族そのものがボトルネックなのではなく、この差はすべて数え上げによる重みの決め方に起因している。なぜ数え上げが重みを歪めるのかについては、次項で較正を調べるときに突き止める。

その前に、ナイーブベイズの重みを直接観察して、分類器が 2 つのクラスを分離するためにどの特徴量を頼りにしているかを見てみよう。プルオーバーとコートのスコア差は重みベクトル $\mathbf{w}_{\text{pullover}} - \mathbf{w}_{\text{coat}}$ を持つアフィン関数であり、ピクセル i が空白ではなく塗られているときには、このベクトルの第 i 成分だけスコアがプルオーバー側へシフトする。プルオーバー寄りを赤、コート寄りを青としてこの差ベクトルを 28×28 画像としてプロットし、絶対値が最大の成分とその背後にあるカウント数を出力しよう。

```
w = (W[2] - W[4]).reshape(28, 28)        # pullover minus coat, in nats
m = np.abs(w).max()
d2l.plt.imshow(w, cmap='RdBu_r', vmin=-m, vmax=m)
d2l.plt.colorbar()
k = np.abs(w).argmax()
print(f'largest |weight| {m:.1f} nats at row {k // 28}, column {k % 28}: '
      f'inked in {n_x[2, k]:.0f} pullovers and {n_x[4, k]:.0f} coats')
```

```
largest |weight| 3.1 nats at row 0, column 1: inked in 21 pullovers and 0 coats
```



プルオーバーの方がコートよりも頻繁に塗られる画像の両脇や上端の行に沿って、赤いピクセルが並んでいる。胴体部分は -1 付近の重みを持つ青いピクセルで埋め尽くされているが、これらの青いピクセルが塗られているときにコートの強い証拠になると解釈するのは誤りである。各重みは、空白のピクセルに**対する**塗られたピクセルの相対的なスコア差を測っていることを思い出そう。胴体のピクセルはプルオーバーの約 99%、コートの 99.5% 以上で塗られているため、塗られているのを観測してもスコアは約 -0.01 nat しか動かない。-1 という重みは実際には、胴体のピクセルがまれに空白のままだったときにモデルがプルオーバー側へ加算する寄与を表している。実質的な判定は両脇の部分で行われているのである。一方、単一で最大の重み（約 3 nat）はプルオーバー約 20 枚とコート 0 枚で塗られている隅のピクセルに位置しており、その大きさは豊富なデータではなく主にラプラスの擬似カウントによって決まっている。

生成モデルの当てはめをさらに批判する前に、識別モデルに対して持つ 1 つの本質的な利点に注目しておこう。遮蔽物の陰に隠れるなどしてテスト時に一部のピクセルが欠損している場合、ナイーブベイズは単に積からそれらの因子を取り除くだけでよい。$p(x_i \mid y)$ を未観測の x_i がとり得る両方の値にわたって周辺化（marginalization）すると和が 1 になるため、生成モデルのもとではこれは数学的に厳密である。ソフトマックス回帰にはこれに対応する周辺化の規則がなく、欠損ピクセルに何らかの値を補完（imputation）しなければならない（演習問題 9）。

36.11.4.3 較正

クラススコアの役割は 10 個のクラスを順位付けするだけではない。指数関数をとればベイズ分子が復元され、正規化すれば先ほど省いた分母 $p(\mathbf{x})$ が復元されるため、モデルの事後確率（事後分布）はスコアのソフトマックスそのものになる。

$$p(y \mid \mathbf{x}) = \frac{\exp s_y(\mathbf{x})}{\sum_{y'} \exp s_{y'}(\mathbf{x})}. \tag{36.142}$$

したがって、すべての予測には主張される確信度（confidence）、すなわちこれら事後確率の最大値が伴う。3.4 節と同様、分類器が主張する確信度が経験的な頻度と一致するとき、すなわち確信度 c でなされたすべての予測のうち実際に割合 c が正解であるとき、その分類器は**較正されている**という。較正を検証するため、テスト画像を確信度のビンにグループ分けし、各ビンにおける平均確信度と実際の正解率を比較する。

期待較正誤差（expected calibration error、ECE）は、この表を 1 つの数値に集約したものであり、主張される確信度と実際の正解率との差を、各ビンに入るテスト画像の割合で重み付けして全ビンにわたり平均した値である。以下ではまず最大スコアを引く数値的に安定な形式（35.8.2 節）でソフトマックスを計算し、あわせて第 1 位と第 2 位のクラススコアの差の中央値も出力する。

```
post = np.exp(s - s.max(axis=1, keepdims=True))      # softmax, stably
post /= post.sum(axis=1, keepdims=True)
c = post.max(axis=1)                                 # claimed confidence
print(f'mean confidence {c.mean():.3f}, accuracy {correct.mean():.3f}')
edges, ece = [0, 0.9, 0.99, 0.999, 1], 0
for lo, hi in zip(edges[:-1], edges[1:]):
    m = (c > lo) & (c <= hi)
    ece += m.mean() * abs(c[m].mean() - correct[m].mean())
    print(f'({lo}, {hi}]: {m.sum():5d} images, claimed {c[m].mean():.3f}, '
          f'achieved {correct[m].mean():.3f}')
top2 = np.sort(s, axis=1)[:, -2:]
print(f'ECE {ece:.2f}, median top-two score gap '
      f'{np.median(top2[:, 1] - top2[:, 0]):.0f} nats')
```

```
mean confidence 0.992, accuracy 0.706
(0, 0.9]:   257 images, claimed 0.726, achieved 0.350
(0.9, 0.99]:   297 images, claimed 0.961, achieved 0.384
(0.99, 0.999]:   265 images, claimed 0.996, achieved 0.419
(0.999, 1]:  9181 images, claimed 1.000, achieved 0.735
ECE 0.29, median top-two score gap 61 nats
```

結果は鮮烈である。モデルは平均して 99% の確信度を主張しながら、実際には予測の 71% しか当てられていない。全テスト画像の約 92% が最上位のビンに入り、そこではモデルが小数第 3 位まで確実だと主張しているにもかかわらず、実際の正解率は約 73% にとどまる。下位 3 つのビンでも、73% から 99.6% の確信度を主張しながら実際の正解率は 35〜42% にすぎない。すべてのビンが過信状態にあるため、期待較正誤差は平均確信度と正解率の差そのものに等しく、約 0.29 となる（3.4 節のソフトマックス回帰では 0.03 未満であった）。ビンが上がるにつれて正解率も上昇しているため確信度のランキング自体は依然として有用な情報を含んでいるが、報告される確率を額面どおりに受け取ることはできない。

原因はプルオーバーの裾で見た現象と同じであり、相関するピクセルが独立な証拠として数えられていることにある。この破綻の最も単純な形は単一の特徴量が複数回複製された場合であり、その計算は 3 行で済む。事前分布が π_0 と π_1 の 2 つのクラスと、特徴量 x（尤度比は $r(x) = p(x \mid y=1)/p(x \mid y=0)$）を考えよう。データセットに k 個の同一コピーが x について含まれているとする。k 個のコピーは常に一致するため 1 個のコピー以上の情報は持たず、真の対数事後オッズは $\log(\pi_1/\pi_0) + \log r(x)$ である。ところがナイーブベイズは、これらのコピーを条件付き独立として扱い、k 個の同一因子 $p(x \mid y)$ を掛け合わせて次の値を報告する。

$$\log \frac{\pi_1}{\pi_0} + k \log r(x). \tag{36.143}$$

重複した証拠は対数オッズを k 倍にする。事前分布が等しいとき、対数オッズに正の定数を掛けても符号は決して反転しないため、ナイーブベイズはどの k に対してもまったく同じクラスを予測するが、$\bar{r}^k$ という事後オッズを主張する。実際には $\bar{r}$（ただし $\bar{r} = \max(r, 1/r)$）こそが正しい値である。これが本節で覚えておくべき 2 つ目の結果である。すなわち、ランキングは生き残るが、確信度は崩壊する。

具体的な数値で見ると、この水増しがいかに激しいかがはっきりわかる。$p(x = 1 \mid y = 1) = 0.8$、$p(x = 1 \mid y = 0) = 0.2$ であり、活性化しているときに $r = 4$ となる、80% の割合で当たる手がかりを考えよう。コピーが 1 個なら正しい確信度 0.80 が得られる。しかしコピーを 2 個、3 個、4 個、5 個と増やすと、予測が実際に当たる割合は依然として 80% のままであるにもかかわらず、ナイーブベイズが主張する確信度は 0.94、0.985、0.996、0.999 へと跳ね上がる。2 単語のスパムフィルタで単語「free」を複製しても同じことが起き、最初のメッセージのオッズは 9 から $\frac{2}{3} \cdot 4.5^2 \cdot 3 = 40.5$ へ跳ね上がって、報告されるスパム確率は 0.976 まで上昇する。

ただし、ランキングの安全性を過大評価してはならない。上の議論は事前分布が等しいことに依存しており、この但し書きは重要である。事前オッズが $\pi_1/\pi_0 = 1/5$ で $r = 4$ のとき、コピーが 1 個なら事後オッズは $4/5$ となって正しくクラス 0 を予測するが、コピーが 2 個になると $16/5$ となって予測がクラス 1 へ反転してしまう。繰り返された証拠は、1 回の観測だけでは覆すべきでない事前分布を圧倒し得るのである。ランキングが保たれるのは、証拠がもともと事前分布と同じ向きにある場合か、1 個のコピーだけでも事前分布を上回る場合に限られる。

Fashion-MNIST の 784 個のピクセルは文字どおりのコピーではないが、広い領域にわたって強く相関している。先ほどの 2 つの裾ピクセルはほぼ 1 つの特徴量が 2 回数えられたかのように振る舞っており、そのような相関領域が何十個も積み重なることで、第 1 位と第 2 位のクラススコアの間に中央値で約 60 nat もの差が生じる。$e^{-60} \approx 10^{-26}$ であるから、ソフトマックスは 60 nat の差を報告確信度 1.000 へと変換する。極端な過信にもかかわらずナイーブベイズがクラスを有用に順位付けできることは、このモデルの古典的な性質として知られている (Domingos and Pazzani 1997)。それでもランキングが完全に無傷というわけではない。相関したパッチの大きさは一様ではないため、冗長なピクセルからなる広い領域が、独立な情報を持つ小さな領域を容易に多数決で押し切ってしまう。隣接ピクセルと同じ情報を繰り返すだけのピクセルの重みを下げられるソフトマックス回帰に対して 12 パーセントポイント遅れをとった理由は、この不均衡で無理なく説明できる（ただし、ここでの実験だけではそのメカニズムを切り分けられない）。

過信（overconfidence）には驚くほど単純な事後的修正法があり、ここではその名前だけ紹介しておく。温度スケーリング（temperature scaling、3.4 節）は、ホールドアウト検証データ上で当てはめた単一のスカラー（ソフトマックス温度）T ですべてのクラススコアを割る手法であり、クラスのランキングを一切変えない。同一コピーが k 個あり事前分布が等しい場合には、$T = k$ とおけば厳密な事後分布が復元される。演習問題 8 では、数え上げモデルに対して T を当てはめる。また、過信はナイーブベイズ特有の問題ではない。深層ニューラルネットワークも、通常はこれより軽度でありアーキテクチャや訓練レシピに依存するものの、同様の統計的バイアスに陥り得る (Guo et al. 2017)。そのため、出力確率に基づいて意思決定を行うあらゆる分類器において、信頼性チェックを行うことは良い実践である。

自然な疑問が 1 つ残っている。クラスあたり 6,000 枚の画像がある設定では、識別モデルの当てはめが圧勝した。しかし訓練例がきわめて少ない場合には、優劣のランキングがしばしば逆転する。Ng and Jordan (2002) は、ナイーブベイズのパラメータが次元 d に対して対数オーダーの訓練例数で収束するのに対し、ロジスティック回帰は d に対して線形オーダーの訓練例数を要し得ることを証明した。演習問題 10 では、プルオーバーとコートを用いてこの逆転現象を探る。

36.11.5 まとめ

ベイズの定理は各クラスがどのように入力を生成するかを表す生成モデルを分類器へと変換するが、制約のない $p(\mathbf{x} \mid y)$ のテーブルを 10^{236} 通りの二値画像にわたって推定することは不可能である。クラスが与えられたもとで特徴量間の条件付き独立を仮定すると、モデルはラプラスの擬似カウント 1 個で平滑化された、クラスおよび特徴量の組ごとの単一の頻度へと縮約されるため、訓練は数え上げに帰着する。ナイーブベイズ分子の対数は二値特徴量に関してアフィンであるため、ベルヌーイナイーブベイズはソフトマックス回帰とまったく同じモデル族に属する線形分類器となる。

Fashion-MNIST 上では、この数え上げによる分類器の正解率は約 $71\% \pm 0.9$ パーセントポイントに達するが、同じ二値ピクセルに直接当てはめたソフトマックス回帰を 12 ポイント下回る。また、相関するピクセルが独立な証拠として扱われ、重複した証拠が対数オッズを何倍にも増幅するため、平均確信度として 99% を報告する。この極端な過信は、実務で分類器の確率を利用する際には必ず較正を検証すべきであることを示している。

36.11.6 演習問題

1. **XOR とナイーブな仮定。** 4 つの点 $(0,0)$、$(0,1)$、$(1,0)$、$(1,1)$ にラベル $y = x_1 \oplus x_2$、すなわちそれぞれ 0、1、1、0 が付いているとする。訓練セットには各点が 1 回ずつ現れる。
   1. $p(y)$ と $\theta_{iy} = p(x_i = 1 \mid y)$ の最尤推定量（最尤推定値）、および 4 点すべてのクラススコアを計算せよ。ナイーブベイズ則 Equation 36.137 はどのような予測を下すか。
   2. スコアのアフィン形式 Equation 36.141 を用いて、ベルヌーイナイーブベイズのパラメータをどのように選んでも 4 点すべてを正しく分類できないことを示せ。退化したパラメータ $\theta_{iy} \in \{0, 1\}$ を許しても解決しないのはなぜか。（各クラスは自身に属する 2 点に正の確率を与えなければならない。）
   3. 2 つの因子 $p(x_1 \mid y)\, p(x_2 \mid y)$ を同時確率表 $p(x_1, x_2 \mid y)$ に置き換えよ。ベイジアンネットワーク (Koller and Friedman 2009) の言葉では、これは Figure 36.50 の右パネルに示されている x_1 と x_2 の間の辺を追加することにあたる。4 つの点からこのモデルを推定し、4 点を正しく分類できることを示せ。そのクラススコアがもはや $\mathbf{x}$ に関してアフィンでないのはなぜか。
   4. ある 1 つのクラス内で、x_1 と x_2 を独立な公平ビットとし、$x_3 = x_1 \oplus x_2$ とおく。3 つの特徴量のどのペアも独立であることを示せ。8 通りの各パターンの確率を 3 つの周辺分布の積と比較せよ。クラス内のすべてのペアごとの相関を調べても仮定 Equation 36.136（条件付き独立）を確認できないのはなぜか。
2. **ロジスティック回帰としてのナイーブベイズ。** 2 クラス $y \in \{0, 1\}$ を考え、事前分布を $\pi_1 = p(y = 1)$、ベルヌーイパラメータを $\theta_{iy} = p(x_i = 1 \mid y)$ とおく。
   1. ナイーブベイズの事後分布が $p(y = 1 \mid \mathbf{x}) = \mathrm{sigmoid}(\mathbf{w}^\top \mathbf{x} + b)$（ただし $\mathrm{sigmoid}(z) = 1/(1 + e^{-z})$、$w_i = \log \frac{\theta_{i1}(1-\theta_{i0})}{\theta_{i0}(1-\theta_{i1})}$）となることを示せ。また、$b$ を θ_{iy} と π_1 を用いて表せ。w_i は、クラス 1 でクラス 0 よりも頻繁に塗られるピクセルに対してどちらの符号を持つか。
   2. 逆に、任意の組 $(\mathbf{w}, b)$ がある $\theta_{iy} \in (0, 1)$ と $\pi_1 \in (0, 1)$ の選択から生じることを示せ。したがって、ナイーブベイズとロジスティック回帰は同じ事後分布の集合を定めており、その当てはめ方だけが異なる。
   3. 訓練セットの 2 クラスが線形分離可能であるとする。2 クラスを分離する $(\mathbf{w}, b)$ に係数 $t \to \infty$ を掛けたとき、ロジスティック回帰の条件付き尤度はどのように変化するか。このことは $(\mathbf{w}, b)$ の最尤推定量に対して何を意味するか。同じデータに対して、ラプラス平滑化を伴うナイーブベイズはどのような $(\mathbf{w}, b)$ を返すか。
3. **ガウスのクラス条件付き分布。** 単一の特徴量が $x \mid y = 0 \sim \mathcal{N}(0, 1)$ および $x \mid y = 1 \sim \mathcal{N}(0, 4)$ に従い、両クラスの事前分布はともに $1/2$ であるとする。

1. x のうち、ベイズ分類器がクラス 1 に割り当てるものの集合を求めよ。x に対する単一のしきい値ではこの集合を作れないのはなぜか。また、どのような変換特徴量を用いれば境界がしきい値になるか。
2. d 次元において、クラス固有の分散を持つガウスナイーブベイズのクラススコアは次の式で与えられる。

$$\log p(y) - \sum_{i=1}^{d} \left[\frac{(x_i - \mu_{iy})^2}{2\sigma_{iy}^2} + \frac{1}{2} \log\left(2\pi\sigma_{iy}^2\right) \right]. \tag{36.144}$$

   2 クラス間の境界が交差項 $x_i x_j$ を持たない二次曲面になることを示せ。
3. 2 次元上の 2 クラスが等しい事前分布を持ち、クラス条件付き分布 $\mathcal{N}(\mathbf{0}, \Sigma_+)$ および $\mathcal{N}(\mathbf{0}, \Sigma_-)$（ただし $\Sigma_\pm = \left(\begin{smallmatrix} 1 & \pm\rho \\ \pm\rho & 1 \end{smallmatrix}\right)$、$0 < \rho < 1$）に従うとする。2 つのクラススコアの差が $2\rho x_1 x_2/(1-\rho^2)$ となり、したがってベイズ分類器がクラス $+$ を割り当てるのが $x_1 x_2 > 0$ のとき、かつそのときに限られること、さらに各クラスの単独のスコアは $x_1^2 + x_2^2$ にも依存することを示せ。両クラスは同じ周辺分布を持つため、ガウスナイーブベイズは無制限のデータがあってもランダム予測の水準にとどまる。$\mathcal{N}(\mathbf{0}, \Sigma_+)$ のもとでは $P(x_1 x_2 > 0) = \frac{1}{2} + \frac{1}{\pi}\arcsin\rho$（Sheppard の公式）が成り立つ。ベイズ誤差を ρ の関数として求め、$\rho = 0.1$ と $\rho = 0.9$ に対してその値を計算せよ。

4. **テキストに対する 2 つの事象モデル**。語彙 V が **goal**、**match**、**vote**、**poll** の 4 語からなるとする。訓練セットには、スポーツ文書が「goal goal match」「goal match match」「match vote」の 3 件、政治文書が「vote poll poll」「vote vote match」「poll vote」の 3 件含まれている。多項モデルは、文書をクラス固有の V 上の分布からの独立な単語抽出の系列として扱う。ベルヌーイモデルは、V の各単語についてその文書内に出現するかどうかだけを記録する。
   1. ラプラス平滑化を用いて両方のモデルを推定せよ。多項モデルでは、クラスの単語ごとの出現回数に 1 を加え、クラスの総単語数に $|V|$ を加える。ベルヌーイモデルでは、単語が出現する確率を $(n_{iy} + 1)/(n_y + 2)$ により推定する（ここで n_{iy} はクラス y の文書のうち単語 i を含む文書の件数である）。
   2. 文書「goal match poll poll」を両方のモデルで分類せよ。それぞれの判定を決定づけているのはどの単語か。
   3. ある文書をそれ自身と k 回連結せよ。2 つのモデルの対数事後オッズはどのように変化するか。実際のテキストでは、文書内に一度出現した単語は再び出現しやすい傾向があり (Church and Gale 1995)、この知見は演習問題 3（36.3 節）でも再現されている。これは多項モデルのどの仮定に違反するか。
5. [code] **対応のある比較**。本文では、`pred` と `pred_sr` の正解率によってナイーブベイズとソフトマックス回帰を比較した。
   1. 両モデルに対して同じ再標本インデックスを用いて画像ごとの正誤の差にブートストラップ法を適用し、正解率の差に対する 95% 区間を求めよ。
   2. この区間の幅を、2 つの周辺区間の幅および対応のない標準誤差 $\sqrt{p_1(1-p_1)/n + p_2(1-p_2)/n}$ と比較せよ。2 つの正誤ベクトルの相関を計算し、それを用いて対応付けが有効である理由を説明せよ。
   3. 両モデルとも誤るテスト画像、ナイーブベイズだけが正解するテスト画像、ソフトマックス回帰だけが正解するテスト画像の枚数をそれぞれ数えよ。正解率の差を決定するのはこれらのうちどの枚数か。また、ナイーブベイズの誤りの一部だけを修正したモデルであれば、この表はどのようになるか。
6. **重複および相関した証拠**。2 つのクラスが等しい事前分布を持ち、ある二値特徴量が $p(x = 1 \mid y = 1) = 0.8$ および $p(x = 1 \mid y = 0) = 0.2$ を満たすとする。
   1. パイプラインがこの特徴量を k 回保存するとする。実際には 80% の割合でしか当たらない予測に対してナイーブベイズが少なくとも 99.9% の確信度を主張する最小の k

を求めよ。

2. 次に、部分的にのみ一致する 2 つのコピーを保存する。y が与えられたもとで、2 番目のコピーは確率 q で 1 番目と一致し、それ以外の場合は同じクラス条件付き分布から新たに独立にサンプルされるとする。両コピーとも x と同じ周辺分布を持ち、したがってナイーブベイズが依然として $16/17 \approx 0.94$ という確信度を両コピーが 1 のときに主張することを示せ。また、正しい事後分布が $r/(1+r)$（ただし $r = 4(0.8+0.2q)/(0.2+0.8q)$）で与えられることを示し、$q \in \{1, 0.75, 0.5, 0.25, 0\}$ に対してその値を計算せよ。過剰計上の度合いはコピー間の相関にどのように依存するか。
3. ソフトマックスの前にすべてのクラススコアを $T = k$ で割ると、(a) において正しい事後分布が復元されることを示せ。これに等しい事前分布が必要となるのはなぜか。（事前項も k で割られるためである。）

7. [code] **ラベルシフト**。ラベルシフト（3.6 節）のもとでは、ターゲットの事前分布 $p_T(y)$ がソースの事前分布 $p_S(y)$ と異なる一方、$p(\mathbf{x} \mid y)$ は同じままである。
   1. ターゲットの事前分布のもとでの事後分布が $p_T(y \mid \mathbf{x}) \propto p_S(y \mid \mathbf{x})\, p_T(y)/p_S(y)$ となることを示せ。ナイーブベイズがこれを利用するのに再訓練を必要としないのはなぜか。
   2. プルオーバーが画像の 90% を占めるテストセット（テスト用プルオーバー全 1,000 枚と、他のクラスからランダムに抽出した 111 枚）を構築せよ。その上で、`pred` の正解率を、$p_T(\text{pullover}) = 0.9$ および他のすべてのクラスで $p_T(y) = 1/90$ とした (a) の補正規則の正解率と比較せよ。
   3. この補正はプルオーバーのスコアに $\log 9 \approx 2.2$ nat を加え、他のクラスから 2.2 を引く。これを `#mdl-naive-bayes-calibration-3` セルのスコア差と比較し、(b) で測定した変化の大きさを説明せよ。
8. [code] **温度スケーリング**。●演習問題 6 では、クラススコアを（ソフトマックス温度）$T = k$ で割ることで特徴量の k 重の複製が打ち消されることを見た。Fashion-MNIST では実効的な重複度は未知であるため、温度スケーリング (Guo et al. 2017) は代わりにホールドアウトデータ上で T を当てはめ、事後分布を $s_y(\mathbf{x})/T$ のソフトマックスとする。
   1. 最初の 50,000 枚の訓練画像でモデルを再当てはめし、T を残りの 10,000 枚における事後分布の平均負の対数尤度を最小化するように選べ。T を変えてもテスト正解率が変化しないのはなぜか。
   2. 当てはめた T を、同じ 50,000 枚モデルのテストセット上のスコアに適用し、ビンの境界を 0、0.5、0.7、0.9、0.99、1 として `#mdl-naive-bayes-calibration-3` セルの信頼性表を再び作成せよ。主張される平均確信度を正解率と比較せよ。
   3. 演習問題 6 の設定において、実データ上のスコア差の標準偏差がモデルが予測する標準偏差の $\sqrt{k}$ 倍になることを示せ。プルオーバーとコート、T シャツとシャツ、プルオーバーとシャツの組についてこの比の二乗を計算し、当てはめた T と比較せよ。
   4. 演習問題 7 で構築したテストセット上で、温度スケーリング後の事後分布に同演習の事前分布補正を適用せよ。温度スケーリングの後では、補正された事前分布によってより多くの予測が変化するのはなぜか。
9. [code] **欠損ピクセル**。すべてのテスト画像のランダムな半分のピクセルを、ブールマスク `M`（`X_test` と同じ形状で、観測可能なピクセルに対して `True` をとる）を用いて隠せ。
   1. この生成モデルのもとで、隠されたピクセルを $p(\mathbf{x} \mid y)$ から周辺化して消去すると対応する因子が取り除かれ、観測可能なピクセルのスコアが $\log p(y) + \sum_{i \text{ visible}} \log p(x_i \mid y)$ になることを示せ。これを 2 つの行列積 `(X_test * M) @ np.log(P_xy).T + ((1 - X_test) * M) @ np.log(1 - P_xy).T` に対数事前分布を加える形で実装し、全ピクセルを用いた場合の正解率 71% と比較せよ。
   2. 代わりに、隠されたピクセルを 0、1、または訓練画像の平均で埋めて、アフィン形式でスコアを計算せよ。それぞれの正解率を Equation 36.141 から説明せよ。また、0 で埋めることはすべてのクラスのスコアに何を加算するか。

3. 隠されたピクセルを 0 に設定した画像に対して、本文で訓練したソフトマックス回帰のスコアを計算せよ。(a) で生成モデルが用いた操作のうち、識別モデルに欠けている操作は何か。

10. [extended] **生成学習と識別学習の比較。●●**Ng and Jordan (2002) は、ナイーブベイズと、演習問題 2 により同じ事後分布族を条件付き尤度の最大化で当てはめるロジスティック回帰とを比較している。同論文では、ナイーブベイズのパラメータは次元 d に対して対数オーダーの訓練例数で母集団の値に近づくことが証明されている。一方、ロジスティック回帰が漸近誤差に近づくために必要な訓練例数は高々 d の線形オーダーであり、最悪ケースでは少なくとも d の線形オーダーになる。同論文の 15 個のデータセットの多くで、訓練セットが小さいときにはナイーブベイズの方がテスト誤差が低く、大きいときにはロジスティック回帰の方が低くなる。
    1. 各クラスが少なくとも m 個の訓練例を持つとする。Hoeffding の不等式（36.9 節）とユニオンバウンドを用いて、2 クラスモデルの全 $2d$ 個の平滑化なしの推定値 $\hat{\theta}_{iy}$ がすべて母集団の値の ε 以内に入る確率が、次式のもとで少なくとも $1-\delta$ となることを示せ。
    $$m \geq \frac{\log(4d/\delta)}{2\varepsilon^2}. \tag{36.145}$$
    また、$d = 784$、$\varepsilon = 0.05$、$\delta = 0.05$ に対してこの下界を計算せよ。
    2. $|\hat{\theta}_{iy} - \theta_{iy}|$ に対する一様上界だけでは、クラススコアの変化を上界で抑えることはできない。本文でプルオーバー対コートの最大の重みを持っていた隅のピクセルのように、θ_{iy} が 0 または 1 に近いピクセルではなぜ抑えられないのか。スコアがパラメータと同じレートで収束するためには、θ_{iy} について何を仮定しなければならないか。
    3. データをプルオーバーとコートに限定せよ。訓練セットの総サイズ $n \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000\}$ および全 12,000 枚の画像について、両クラスを含むランダムな訓練部分集合を、20 個ずつ（$n = 200$ 以下の各サイズ）および 5 個ずつ（それより大きい各サイズ）抽出せよ。ラプラス平滑化を伴うナイーブベイズと、重み減衰なしのロジスティック回帰を当てはめよ。小さい n では訓練データが線形分離可能となり、演習問題 2c によりロジスティック損失は最小点を持たないため、ゼロ初期化から固定回数のフルバッチ勾配ステップを実行せよ。`d2l.plot` を用いて、両分類器の平均テスト誤差を対数軸上の n に対してプロットせよ。ナイーブベイズの方が低い誤差を示す n の範囲は存在するか、存在するならどの範囲か。
    4. ナイーブな仮定（条件付き独立）が成り立つ合成データを用いて (c) を繰り返せ。すなわち、均衡したラベルを抽出し、各ピクセルを Bernoulli(θ_{iy}) から独立にサンプルせよ（ただし θ_{iy} は 2 クラスの全訓練画像から推定した値とする）。2 本の学習曲線とそれぞれの漸近誤差はどのように変化するか。
    5. Ng and Jordan (2002) の定理 4 は、ナイーブベイズの過剰誤差を、母集団モデルのスコア差がゼロ付近に落ちる確率によって上から抑えている。実際のピクセルと (d) の合成データについてこの確率を比較し、その比較結果と (c) および (d) の漸近誤差とをあわせることで、2 組の学習曲線の振る舞いがどのように説明できるかを述べよ。

36.12 関連文献

本節の末尾にあるリストには、本章で学んだ内容に対応する書籍や講義をまとめている。それに先立ち、まずは本章で割愛した話題に目を向けよう。本章の構成順に沿って、各小項目ではこれまで扱った手法の限界から出発し、研究文献がそれをどのように乗り越えているかを解説する。なかでもコンフォーマル予測（conformal prediction）と信頼列（confidence sequence）の 2 つは実務でただちに役立つ手法であり、実際の実験における雑多な条件のもとで 36.5 節の区間推定や検定を補強する。また、VC 次元のように主として理論的な関心事であるものもあれば、勾配

ノイズが本当に重い裾（heavy tail）を持つのかどうかのように、現在も活発に議論されている話題もある。本書の以降の箇所は本節の内容に依存しないため、さらなる探究のための地図として活用されたい。

36.12.1 確率モデル

まずは分布の扱いに関する 3 つの拡張から始めよう。すなわち、離散と連続の狭間に位置する微妙な分布のクラス、分散が無限大となる平均に対する極限法則、そして選択肢どうしが似通っているときにソフトマックスが見せる古典的な癖である。

36.12.1.1 アトムも密度も持たない分布

36.1 節では、密度を持つ確率変数は点質量（アトム）を持たないことを述べ、あらゆる分布を離散分布、密度を持つ連続分布、またはその 2 つの混合として扱った。驚くべきことに、その逆は成り立たない。すなわち、アトムを持たないだけでは密度の存在は保証されないのである。理由を見るために、区間 $[0, 1]$ 内の乱数を 3 進展開で構成することを考えよう。各桁を公正なコインで独立に選び、数字は 0 と 2 のみを用いるものとする。この累積分布関数はいたるところで連続であるため、正の確率を持つ単独の点は存在しない。それにもかかわらず、確率質量のすべては全長が 0 であるカントール集合（Cantor set）の上に集中している。長さ 0 の集合上で積分して 1 になる密度関数は存在しない。

この分類を完結させるのが**ルベーグ分解**（Lebesgue decomposition）である。実直線上のあらゆる確率分布は、離散部分、密度を持つ部分、そしてカントール分布のような**特異連続**（singular continuous）部分の混合として表される。密度を持つことは、長さに関して**絶対連続**（absolutely continuous）であることと厳密に対応しており、その場合にはラドン・ニコディムの定理（Radon–Nikodym theorem）が密度を与える。Billingsley (1995) はこのような測度論的基礎の上に確率論を展開しており、そこでは期待値が分布に対する単一の積分として定式化され、本章で扱った和と積分は単なる 2 つの特別な場合として現れる。

36.12.1.2 重い裾と安定法則

36.1 節で述べたように、独立なコーシー分布からのサンプルの平均はまったく同じコーシー分布に従い、また演習問題 5 で見たように、べき乗則の裾を持つ確率変数の平均は、おなじみの $1/\sqrt{n}$ のレートよりはるかに遅くしか収束しない。ガウス分布に近づかないとすれば、いったい何に収束するのだろうか。裾 $P(|X| > x)$ が $x^{-\alpha}$（ただし $0 < \alpha < 2$）のように減衰するとき、分散は無限大となり、適切に中心化した和を $n^{1/\alpha}$ で（$\sqrt{n}$ ではなく）スケーリングすると、α **安定**法則（stable law）に収束する。コーシー分布は $\alpha = 1$ の対称安定法則であり、ガウス分布はその境界である $\alpha = 2$ に位置する。Gnedenko and Kolmogorov (1954) は、これらの極限法則の古典理論を展開している。

深層学習では、Simsekli et al. (2019) が α 安定分布としてモデル化したミニバッチ勾配ノイズと、訓練された重み行列の固有値スペクトル (Martin and Mahoney 2021) の双方において、重い裾が注目を集めている。もっとも、こうした主張は割り引いて受け止める必要がある。この種の経験的な主張はいずれも裾指数（tail index）α の推定に依拠しているが、その推定値は裾のどこまで外側を見るかによって大きく変動し得るからである。

36.12.1.3 選択モデルと無関係な選択肢

36.4.4 節では、各選択肢のスコアに独立なガンベルノイズを加えることでソフトマックスを導出した。この独立性の帰結として、任意の 2 つの選択肢間のオッズ比（odds ratio）$p_i/p_j = e^{z_i - z_j}$ は、選択肢リストにある他のすべての選択肢に依存しない。この性質は、2 つの選択肢がほぼ同

一である途端に破綻する。自家用車の運転と赤いバスの利用を半々で使い分けている通勤者を想像してみよう。市がまったく同じ青いバスを新たに導入したとしても、この通勤者は引き続き半分の割合で車を運転し、残りの半分を 2 種類のバスで分け合うはずである。ところがソフトマックスのもとでは、3 つの選択肢すべてに確率 $\frac{1}{3}$ が割り当てられてしまう。

入れ子ロジットモデル（nested logit model）や相関のあるガウスノイズを用いたプロビットモデルのように、類似した選択肢のノイズ間に相関を許せば、閉じた形式の選択確率は得られなくなるものの、車の選択確率を $\frac{1}{2}$ 付近に保つことができる。McFadden (1974) はガンベルノイズからロジットを導出するとともに、Luce の**無関係な選択肢からの独立性**（independence of irrelevant alternatives）として知られるこの制約を、モデル化における中心的な限界として指摘した。

36.12.2 推定と検定

36.5 節の信頼区間と仮説検定は、ガウス近似、誤差に対する既知のモデル、またはデータを見る前に固定された標本サイズという 3 つの好都合な前提に依拠していた。以下の 3 つの話題では、こうした前提を 1 つずつ手放したときに何が起こるかを見ていく。

36.12.2.1 改良されたブートストラップ区間

36.5 節で扱った単純なパーセンタイルブートストラップ法は、再サンプリングされた推定値の統計的バイアスや歪度（skewness）をそのまま引き継ぐため、小標本や歪んだ統計量に対しては実際の被覆率（coverage）が公称水準を大きく下回ることがある。2 つの改良手法がこのずれの大部分を解消する。**スチューデント化**（studentized）ブートストラップ法は、自己正規化されたピボット量 $(\hat{\theta}^{*} - \hat{\theta})/\widehat{\mathrm{se}}^{*}$ を（$\hat{\theta}^{*}$ そのものに代えて）再サンプリングする。一方、**BCa**（バイアス補正・加速、bias-corrected and accelerated）区間は、バイアスと θ に応じた標準誤差の変化の双方を考慮してパーセンタイルのカットオフを移動させる。片側区間では、いずれの改良も被覆誤差をオーダー $1/\sqrt{n}$ からオーダー $1/n$ へと縮小させる (Hall 1992)。数千件のテスト例に対する正解率を報告する際にはほとんど必要ないが、小規模な評価セットや誤り率の比を扱う場面ではただちに威力を発揮する。これらの手法については DiCiccio and Efron (1996) が明快な解説を与えている。

36.12.2.2 コンフォーマル予測

訓練済みのニューラルネットワークは点予測を出力するが、パラメトリックな誤差モデルを仮定しない限り、被覆率が保証された予測区間（prediction interval）は与えてくれない。**分割コンフォーマル予測**（split conformal prediction）は、モデル化の仮定を一切置かずにこれを実現する。ホールドアウトした n 個の較正点を用い、たとえば残差 $|y - f(\mathbf{x})|$ によってモデルが各点をどれだけ外したかをスコア化する。そして新しい入力に対しては、スコアが小さい方から数えて $\lceil (n+1)(1-\alpha) \rceil$ 番目の較正スコア以下となるすべてのラベルを返す。

較正点とテスト点が交換可能（exchangeable）である限り、得られる予測集合は**任意の**モデルおよび**任意の**データ分布に対して、少なくとも $1-\alpha$ の確率で真のラベルを含む。証明には並べ替え検定（permutation test）と同じ対称性を用いる。すなわち、交換可能性のもとでは、全 $n+1$ 個のスコアの中でのテストスコアの順位は一様に分布する。実務上は 1 つ注意点がある。この保証は入力全体にわたる平均として成り立つものであり、個々の入力で条件付けたときに成り立つわけではない。理論の全体像は Vovk et al. (2022) が展開しており、Angelopoulos and Bates (2023) は分類と回帰の両方についてコード付きのわかりやすいチュートリアルを提供している。

36.12.2.3 信頼列

36.5 節の z 検定は、結果を 1 つでも見る前に標本サイズ n を決めていることを前提としている。それとは逆に、進行中の実験をバッチごとに覗き見して $p < 0.05$ となった瞬間に打ち切るなら、十分に長く観察を続ける限り誤警報はほぼ確実に発生する。演習問題 10 では、どれほど頻繁に確認しても妥当性を保つ混合ガウスマルチンゲール検定を構成した。**信頼列**はこの考え方を区間推定全体へと推し進めたものであり、単一のパラメトリックモデルに限らず任意の有界データや準ガウス（sub-Gaussian）データに対して、すべての標本サイズ n で同時に成り立つ区間を与える。この自由度と引き換えに区間幅は広がるものの、最も鋭い境界を用いれば、余分な幅の増大は $\sqrt{\log \log n}$ 程度と驚くほど緩やかである。Howard et al. (2021) は非負優マルチンゲール（supermartingale）からこれらの列を構成しており、Johari et al. (2017) は常時妥当（always-valid）な p 値が商用の A/B テストプラットフォームにどのように導入されたかを解説している。

36.12.3 事後分布を用いた計算

36.8 節では、2 パラメータの事後分布に対して、単一のマルコフ連鎖であるランダムウォーク・メトロポリス法を適用した。しかし実際のベイズモデルは数千から数百万個のパラメータを持つため、より効率的なサンプラとスケーラブルな近似手法が求められる。

36.12.3.1 ギブスサンプリングとハミルトニアンモンテカルロ

ランダムウォーク・メトロポリス法（36.8.3 節）では、あらゆる方向について提案分布のスケールを調整する必要があり、小さな拡散的ステップで少しずつしか進めない。事後分布が高次元であったり強い相関を持っていたりすると、連鎖は大半の時間を行き来に費やしてしまう。**ギブスサンプリング**は、厳密な条件付き分布から 1 座標ずつ順番に更新することで提案分布の調整を不要にし、すべての移動が採択されるようにする。ただし難点として、それらの条件付き分布から容易にサンプリングできなければならず、また座標間の相関が強いと依然として収束は遅くなる (Geman and Geman 1984)。

ハミルトニアンモンテカルロはより物理学的なアプローチをとる。すなわち、$-\log \tilde{p}$ をポテンシャルエネルギーとみなし、ランダムな運動量（モメンタム）を注入したうえで、勾配を用いて生じる軌跡を多数ステップにわたってシミュレートする。このダイナミクスはエネルギーをほぼ保存するため、1 回の提案で事後分布上を遠くまで滑らかに移動しつつ、高い確率で採択される。手法とその調整法については Neal (2011) が明快な入門を与えており、また Hoffman and Gelman (2014) は No-U-Turn Sampler によって軌跡長を自動化し、これが Stan の標準エンジンとなった。

36.12.3.2 ニューラルネットワークの事後分布

36.8 節のすべてのアルゴリズムは、各ステップでデータセット全体にわたる完全な対数同時確率を評価し、またラプラス近似では $d \times d$ のヘッセ行列を保持して逆行列を求める必要がある。数百万個のパラメータと数百万件の訓練例を持つニューラルネットワークでは、そのどちらも実行不可能である。**確率的勾配ランジュバン動力学**（stochastic-gradient Langevin dynamics）は、演習問題 9 の未調整ランジュバン更新にミニバッチ勾配と減衰するステップサイズを組み込み、メトロポリス採択判定を手放す代わりに 1 ステップあたりのコストを通常の SGD と同一にする (Welling and Teh 2011)。一方、ラプラス近似は別の方法でスケールアップを図る。層ごとのクロネッカー分解ブロックや最終層のみのヘッセ行列など、曲率のうち最も情報量の多い部分だけを保持するのである。Daxberger et al. (2021) は、こうした軽量な近似がごくわずかな計算量でありながら、はるかに高コストなベイズ手法に匹敵する性能をしばしば達成することを示している。

36.12.4 集中と学習理論

以下の最初の 2 つの話題では、36.9 節と 36.10 節の道具立てを、単純な平均やラデマッハ複雑度（Rademacher complexity）の先へと一般化する。後半の 2 つの話題では、本章の理論パートの締めくくりで残された謎に取り組む。すなわち、仮説クラスに潜む最悪の関数に対してではなく、アルゴリズムが実際に見つけ出す特定の解に対してどのように保証を証明するかという問題である。

36.12.4.1 有界差分とリプシッツ関数

36.9 節のチェルノフ法は、独立な項の和であることを前提としている。しかし、モデルクラス全体にわたる真のリスクと経験リスクの最大乖離など、関心のある多くの量は独立な入力に依存するものの単純な和の形をしていない。**マクディアミッドの不等式**（McDiarmid's inequality）は、ヘフディングの上界をそのような関数へと一般化する。「各項が長さ c_i の区間に収まる」という条件を「入力 i のみを変えたとき関数の変化量は高々 c_i である」に置き換えても、積率母関数（moment generating function）の積をマルチンゲールに置き換える証明によって、まったく同じ裾の上界 $e^{-2t^2/\sum_i c_i^2}$ が成り立つ。

ガウスベクトルはさらに明快な性質を持つ。任意の次元 d の標準ガウスベクトルに対して、すべての L リプシッツ関数 f は $P(|f - \mathbb{E}f| \geq t) \leq 2e^{-t^2/(2L^2)}$ を満たす。ユークリッドノルムは 1 リプシッツであるため、薄い球殻（thin shell）の周りでの揺らぎはオーダー 1 にとどまり、それは d がどれほど大きくなっても変わらない。Boucheron et al. (2013) は、これら両方の結果とその背後にある理論的枠組みを包括的に扱っている。

36.12.4.2 VC 次元と被覆数

36.10 節では、ラデマッハ複雑度を通じて無限の仮説クラスを制御した。これに代わる古典的なアプローチでは、有限個の点の集合に対してクラスが何通りの異なるラベル付けを生成できるかを数える。二値分類器の族について、全 2^{d+1} 通りのラベル付けをいかなる $d+1$ 個の点に対しても実現できないとする（たとえば $\mathbb{R}^m$ 内の線形半空間では $d = m + 1$ である）。このときザウアーの補題（Sauer's lemma）により、このクラスが生成するラベル付けの総数は高々 $(en/d)^d$ 通りであることが任意の $n \geq d$ 個の点に対して保証され、それらのラベル付けにわたるユニオンバウンドからオーダー $\sqrt{d \log n / n}$ の汎化ギャップが得られる。これが Vapnik and Chervonenkis (1971) による著名な議論である。実数値関数クラスに対しては、代わりに各解像度 ϵ でクラスを有限個の代表元によって被覆する。スケールにわたって和をとる**チェイニング**と呼ばれる技法を用いると、余分な $\log n$ 因子を取り除くことができる (Vershynin 2018)。

36.12.4.3 PAC-Bayes 限界

36.10 節のすべての限界は、データを見る前に固定された仮説クラス内の最悪の関数に対して備えるものであった。**PAC-Bayes** 限界は視点を転換し、データを観測した**後**に選ばれる関数上の事後分布 Q を、あらかじめ固定した事前分布 P を基準として評価する。このとき、クラスの複雑度項は平方根の中の $D_{\mathrm{KL}}(Q\|P)/n$ に置き換わる (McAllester 1999)。仮説クラス全体がどれほど巨大であっても、事後分布が事前分布の近くにとどまる限り、支払う複雑度のペナルティは小さくて済む。この考え方を土台として、Dziugaite and Roy (2017) は SGD が見つけた重みの周囲でガウス事後分布を最適化し、訓練例よりはるかに多くのパラメータを持つ確率的ネットワークに対して、二値 MNIST タスクにおける非自明な（nonvacuous）汎化限界を導出した。

36.12.4.4 アルゴリズム的安定性

ランダムラベル実験（4.5.1.1 節）から明らかなように、ニューラルネットワークが表現**し得る**関数のクラスは、実データ上で SGD が実際に訪れる部分集合よりはるかに大きい。**アルゴリズム的安定性**（algorithmic stability）は、仮説クラスではなく学習アルゴリズムそのものを抑える。任意の 1 個の訓練例を入れ替えたとき、学習されたモデルの任意のテスト点における損失の変化が高々 β であれば、汎化ギャップの期待値もオーダー β となり、マクディアミッドの不等式によってその期待値を高確率の限界へと引き上げることができる (Bousquet and Elisseeff 2002)。滑らかでリプシッツ連続な凸損失に対して、SGD は β がステップサイズの総和を n で割った値に比例する形で安定となるため、訓練のステップ数が少ないかステップサイズが小さいほど、よりタイトな保証が得られる (Hardt et al. 2016)。

36.12.5 カウントに基づく分類器

最後に、36.11 節から続く 2 つの発展的話題で締めくくろう。1 つは単一の温度だけでは不十分なときに較正のずれた確率を補正する手法であり、もう 1 つは純粋に識別的な当てはめでは直接扱えないラベルなしデータからの学習を生成分類器によって実現する手法である。

36.12.5.1 単一温度を超える較正

温度スケーリング（演習問題 8）が機能するのは、分類器のスコアが単一の乗法的スケールを除いて正確である場合である。歪みがより複雑なときには、さらに表現力の高い写像が役立つ。**Platt スケーリング**は二値スコアを確率へ写すロジスティック曲線の内部で傾きとオフセットの両方を当てはめ (Platt 1999)、一方アイソトニック回帰（isotonic regression）は任意の非減少関数を当てはめる。Zadrozny and Elkan (2002) はアイソトニック回帰を適用して、ナイーブベイズをはじめとする分類器を較正した。ここでのトレードオフはおなじみのものである。柔軟な較正器ほど過学習を避けるためにより多くのホールドアウトデータを必要とし、現代の深層ネットワークに対しては単一の温度だけでも同等にうまく機能することが多い (Guo et al. 2017)。

36.12.5.2 半教師ありナイーブベイズ

多くの実問題では、ラベル付き例の取得にはコストがかかる一方で、ラベルなしの入力は豊富に手に入る。ナイーブベイズは各クラスがどのように特徴量を生成するかをモデル化するため、欠測しているクラスを潜在変数として扱えば、ラベルなし例も尤度の中に自然に組み込める。すると 36.7 節の EM アルゴリズムによって、ラベルなし例に対するクラスの事後分布の計算と、その事後分布を重み付きの小数ラベルとして用いた単語カウントやピクセルカウントの更新を交互に繰り返すことができる。Nigam et al. (2000) は、ごく少数の文書しかラベル付けされていない状況でテキスト分類の正解率が大幅に向上することを実証する一方、生成モデルのデータへの当てはまりが悪い場合にはラベルなしデータがかえって性能を損ない得ることも警告している。

36.12.6 確率・統計の一般的な参考資料

書籍

- 『Introduction to Probability』（Blitzstein・Hwang）は、確率変数、期待値、条件付け、および 36.1 節と 36.3 節で扱った代表的な分布への親しみやすい入門書である。Harvard 大学の講義 Stat 110 に合わせた無料 PDF が公開されている。
- 『Mathematics for Machine Learning』（Deisenroth・Faisal・Ong）では、第 6 章において 36.4 節の指数分布族を含む確率と分布の自己完結した解説が与えられている。

- 『All of Statistics』（Wasserman）は、推定、検定、信頼区間、ブートストラップ法をコンピュータ科学者にも読みやすく明快にまとめた書籍であり、36.5 節と 36.6 節の範囲を網羅している。サポートページには正誤表、コード、データセットが用意されている。
- 『Computer Age Statistical Inference』（Efron・Hastie）は、ブートストラップ法、大規模検定、および頻度主義推論とベイズ推論の関係を掘り下げており、36.5 節を補完する一冊である。
- 『Pattern Recognition and Machine Learning』（Bishop）は、指数分布族、混合モデルと EM（36.7 節）、および 36.8 節のラプラス近似と変分近似を扱っている。Microsoft Research から無料 PDF が提供されている。
- 『Probabilistic Machine Learning: An Introduction』（Murphy）は、MAP 推定（36.6 節）や生成分類器（36.11 節）を含め、確率と決定理論の視点から機械学習を展開しており、無料の草稿と実行可能なノートブックが公開されている。
- 『Information Theory, Inference, and Learning Algorithms』（MacKay）は、推論と情報理論を有機的に結びつけた書籍であり、ラプラスの方法、モンテカルロ法、変分推論に関する各章は 36.8 節の副読本となる。
- 『Bayesian Data Analysis』（Gelman ほか）は、36.8 節に関連する事前分布、事後分布、収束診断を含むマルコフ連鎖モンテカルロ法、および実践的なベイズワークフローを扱っており、第 3 版が無料 PDF として公開されている。
- 『High-Dimensional Probability』（Vershynin）は、集中不等式、準ガウス変数、高次元幾何学を展開し、36.9 節の内容をさらに発展させている。
- 『Understanding Machine Learning: From Theory to Algorithms』（Shalev-Shwartz・Ben-David）は、36.10 節と同様に一様収束、ラデマッハ複雑度、マージン限界を導出しており、著者らによる無料 PDF が提供されている。

講義とビデオ講義

- Statistics 110: Probability（Harvard 大学、Blitzstein）では、36.1 節と 36.3 節に対応する講義動画、演習課題、および解答付きの練習問題 250 問以上が公開されている。
- Stat 110 講義動画（YouTube）には、確率、分布、条件付けに関する講義シリーズ全編が収録されている。

チュートリアルと講義ノート

- Probability Theory Review（Stanford 大学 CS229、Maleki・Do）は、機械学習向けに確率変数、期待値、分散、ガウス分布、多変量確率を復習する資料であり、36.1 節の簡潔な補助資料として役立つ。

Chapter 37

情報理論とダイバージェンス

本書のほとんどのモデルは、交差エントロピー損失（cross-entropy loss）を最小化することで訓練される。これは、観測された結果に高い確率を割り当てるという直観的な目標を、勾配降下法のための滑らかな目的関数へと変換するものである。しかし、その損失の数値は実際には何を測っているのだろうか。言語モデルが文字あたり 1.2 nat を示したとき、その値は原理的にどこまで下がり得るのか。また、交差エントロピーを変分オートエンコーダ（variational autoencoder）の KL ペナルティや敵対的生成ネットワーク（generative adversarial network）のクリティックに置き換えるとき、何を比較しているのだろうか。

Claude Shannon は 1948 年、ノイズのある回線を通してどれだけ効率よくメッセージを送れるかという問いに取り組むなかで、これらの疑問に答えた。予測不可能性の尺度である**エントロピー**は、いかなるモデルの交差エントロピーも下回ることのできない厳密な下限を定める（3.1.2.3 節）。ここから導かれる 3 つの帰結が、深層学習全体を貫いている。第 1 に、交差エントロピーはデータのエントロピーを、データからモデルへの **KL ダイバージェンス**の分だけ上回り、この差は両者が一致するときに限り 0 になる。第 2 に、ビット単位で表したモデルの総損失は、高々 2 ビットのオーバーヘッドを除けば文字どおり圧縮ファイルサイズそのものであり、これによって言語モデルのパープレキシティ（8.3.2 節）に直接的な操作的意味が与えられる。第 3 に、学習されたクリティックはダイバージェンスや共有情報量の下界しか与えない。対照学習（contrastive learning）において N 個の候補から対応する相手 1 つを識別できても保証されるのは高々 $\log N$ nat であり、弱い敵対的クリティック（16.2 節）のもとでは遠く離れた分布どうしが近く見えてしまうことがある。

本章は 3 部構成をとる。第 1 部では、エントロピーと交差エントロピーを展開し（37.1 節）、対数損失（log-loss）を可逆圧縮（lossless compression）およびパープレキシティへと直接結びつける（37.2 節）。第 2 部では、密度比ダイバージェンス（density-ratio divergence）の決定的な破綻を取り上げる（37.3 節）。モデルのサンプルがデータと重ならないときには必ず KL が発散し、有界な f ダイバージェンスは飽和して勾配が 0 になってしまう。滑らかなテスト関数の期待値や最適輸送（optimal transport）のコストを比較すれば（37.4 節）有用な勾配を回復でき、一方で対数密度（log density）の傾きを一致させれば（37.5 節）、計算困難な分配関数（partition function）を扱うことなく非正規化モデルを比較できる。第 3 部では、相互情報量（mutual information）、データ処理、ファノの不等式（Fano's inequality）、情報ボトルネック（information bottleneck）を通じて確率変数の対を調べ（37.6 節）、最終的にサンプルに基づく推定と InfoNCE へと至る（37.7 節）。なお、最後の 2 節は 37.1 節の後だけで自己完結しているため、輸送とスコアの話題を第 16 章および第 17 章まで取っておきたい場合は、そのまま先に読み進めてもよい。

本章では、イェンセンの不等式（第 35 章）、条件付き分布、および最尤法（第 36 章）を既知とす

る。誤り訂正符号（error-correcting code）、コルモゴロフ複雑性（Kolmogorov complexity）、情報幾何学（information geometry）については、37.8 節で概観する。

37.1 エントロピー、交差エントロピー、KL ダイバージェンス

分類器を訓練するとき最初に確認するのは損失曲線であり、その形状の読み方はすぐに身につく。右下がりの曲線は順調な兆候であり、横ばいの曲線は疑わしい。一方で、縦軸の数値そのものを解釈するのはずっと難しい。これまでのところ説明できる数値は 1 つだけである。初期化したばかりの 10 クラス分類器は約 2.30 の損失を示すが、これは 4.2 節で未訓練ネットワークの健全性チェックに使った値である。この数値は $\ln 10$ であり、モデルが 10 個のラベルをすべて等確率とみなしているときに、そのうち 1 つを指定するためのコストにあたる。曲線の反対側の端は、さらに深い問いを投げかける。最終的な損失 0.6 は良い値なのだろうか。曲線が横ばいになったとき、そのプラトーは訓練のバグなのか、それともデータの避けられない性質なのか。また、ラベルの一部に誤りがある場合、モデルはそもそもどこまで損失を下げられるのだろうか。

交差エントロピー損失（cross-entropy loss）は $-\log q(y)$ の平均であり、実際に生じたラベル y に対してモデルがどれだけ驚いたかを測る。この平均は自然に 2 つの部分に分解される。1 つ目は**エントロピー**であり、ラベルの真の分布をすでに知っている観測者が経験する平均的な驚きを表す。エントロピーはデータそのものによって決まるため、いかなるモデルもこれを減らすことはできない。2 つ目は**カルバック・ライブラー（KL）ダイバージェンス**であり、真の分布の代わりに不完全なモデルを使うことで生じる余分な驚きを表す。この量は決して負にならず、モデルが真の分布と一致するときに限り 0 になる。したがって、損失の値は、この削減不能な下限に照らして初めて評価できる。

前章では、これらの考え方のうち 2 つにすでに出会った。最尤法（maximum likelihood）が交差エントロピー（cross-entropy）を最小化すること（36.6.2.1 節）、そしてギブスの不等式（Gibbs' inequality）として知られる結果、すなわち KL ダイバージェンスが決して負にならないこと（36.4.2 節）である。前章ではこれらを道具として用いたが、本節ではこれら自体を主題とする。まずは 1 組のトランプと 4 記号の情報源から始め、そこではエントロピーが単に「はい・いいえ」の質問の回数を数えるものにすぎないことを見る。そこから 2 つの分布の比較へと進み、続いて分類器に戻ってその損失の下限を求め、最後にターゲットをソフト化した場合や結果が連続な実数値である場合に何が変わるかを調べる。

対数の底は単に単位の選択にすぎない。底 2 は情報量を**ビット**で測り、底 e は **nat** で測る。**本書では一貫して nat で計算を進める。**これは数値計算ライブラリや報告される訓練損失で使われる自然対数 `log` に合わせたものである。「はい・いいえ」の質問の回数を数えるときにはビットの方が便利であるため、ビット単位の量には下付き添字 2 を付ける（たとえばエントロピーなら H_2）。1 ビットは $\ln 2 \approx 0.693$ nat に等しい。Table 37.1 に、本章全体で用いる記号をまとめる。

Table 37.1: 情報理論の章で用いる記号。2 値エントロピー $H_b(p)$ は確率変数ではなく数値の関数である。相互情報量 $I(X;Y)$ および最後の 5 行は後の節で定義する。

記号	意味	単位
$I(x) = -\log p(x)$	結果 x の自己情報量（サプライザル）（慣例により相互情報量 $I(X;Y)$ と同じ文字を用いる）	nat
$H(X)$, $H(P)$	離散確率変数または離散分布のエントロピー（下付き添字 2 は H_2 や CE_2 のようにビットを表す）	nat
$H_b(p)$	表の確率が p であるコインの 2 値エントロピー	nat

記号	意味	単位
$h(X)$	連続確率変数の微分エントロピー	nat（X の単位に依存）
$D_{\mathrm{KL}}(P\|Q)$	P から Q への KL ダイバージェンス	nat
$\mathrm{CE}(P,Q)$	P から Q への交差エントロピー	nat
k	クラス数（結果の集合のサイズ）	
n	例の数、またはメッセージ中の記号数、$\mathbb{R}^n$ の次元	
$\mathbf{p}$, $\mathbf{q}$	k クラス上の確率ベクトル（真の分布、モデル）	
$\mathbf{z}$	ロジット（$\mathbf{q} = \mathrm{softmax}(\mathbf{z})$）	
ϵ	ラベル平滑化の質量（後の節ではノイズ変数）	
τ	温度	
ρ	ラベルノイズ率（記録されたラベルがクリーンなクラスでない確率）	
$H(X \mid Y)$, $I(X;Y)$	条件付きエントロピー、相互情報量	nat
$D_f(P\|Q)$	ダイバージェンス関数 f を持つ f ダイバージェンス	
T	クリティック（2 つの分布のもとでの平均がダイバージェンスの限界を与える関数）	
g	積分確率指標のテスト関数	
γ	2 つの分布のカップリング（輸送計画）	
N	比較する 2 つの標本の各点数、または対照バッチ内の候補数	

37.1.1 情報量とエントロピー

37.1.1.1 自己情報量

シャッフルしたばかりの 1 組のトランプに関する 4 つの報告を考えてみよう。「カードが見える」という報告は、事象が確実であるため何も伝えない。「ハートが見える」は結果を等確率な 4 つのスートのうち 1 つに絞り込み、「スペードの 3 である」は等確率な 52 枚のカードのうち 1 枚を特定し、デッキ全体の並び順の報告は $52!$ 通りの可能な並び順のうち 1 つを特定する。事象の確率が低いほど、それを観測したときに得られる情報は大きくなる。Shannon (Shannon 1948) はこの考え方を、結果 x（確率 $p(x)$ で生じる）の**自己情報量**（self-information）によって定量化した。

$$I(x) = -\log p(x).$$

この量は結果の**サプライザル**とも呼ばれ、3.1.2.3 節ではこちらの用語を用いた。結果を予期していなかったほど、驚きは大きくなる。重要なのは、この驚きは確率を割り当てた主体に属するという点であり、確信を持って誤っているモデルはありふれた事象に対しても強く驚くことになる。確実な事象が持つ情報量は 0 であり、結果の確率が半分になるとサプライザルは $\ln 2$ nat（1 ビット）増加する。最初の 3 つのカードの報告が持つ情報量は、それぞれ 0、$\ln 4 \approx 1.39$、$\ln 52 \approx 3.95$ nat である。公正なコインを 1 回投げて得られる情報量を 1 ビットとするビット単位では、それぞれ 0、2、5.70 ビットとなる。

なぜ対数を選ぶのだろうか。ここで欲しい性質は 3 つある。確実な事象の情報量は 0 であること、まれな事象ほど多くの情報量を持つこと、そして 2 つの独立な事象の情報量は足し合わされることである。独立な事象の確率は掛け合わされるため、3 つ目の要件は $I(p_1 p_2) = I(p_1) + I(p_2)$ と書ける。この恒等式を満たす減少関数は $-\log p$ の正の定数倍に限られる（Csiszár (2008)、証明は演習問題 1 で扱う）。定数倍の選択は、単に単位の選択に相当する。

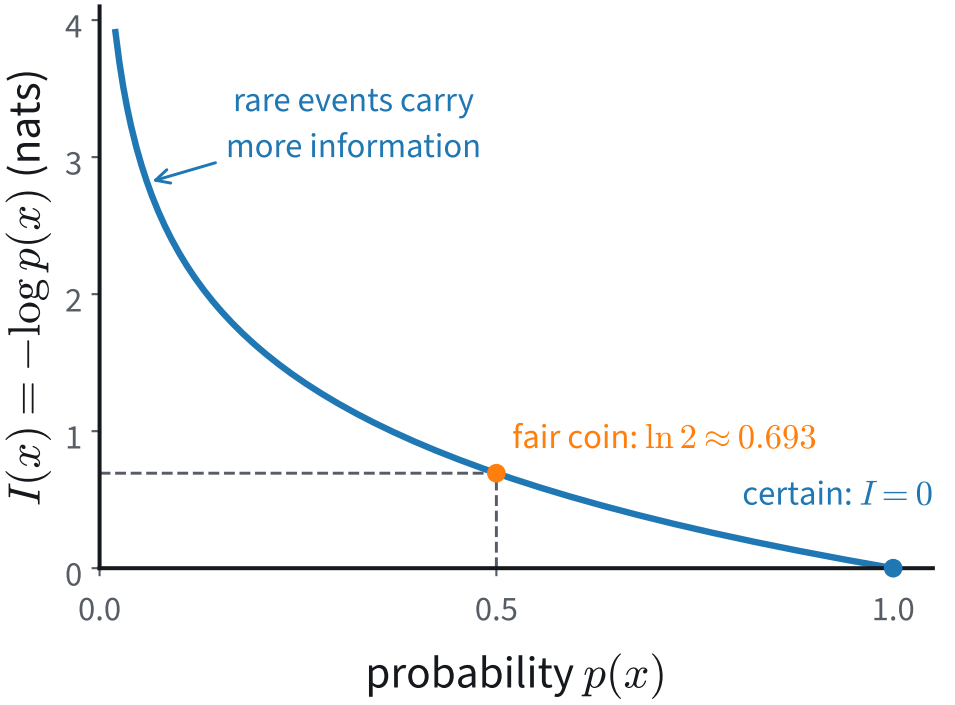


Figure 37.1: 確率 p の関数として表した nat 単位の自己情報量 $I(x) = -\log p(x)$。確実な事象（$p = 1$）の情報量は 0 であり、公正なコイン（$p = \frac{1}{2}$）は $\ln 2 \approx 0.693$ nat を持ち、$p \to 0$ で曲線は発散する。

Figure 37.1 にこの曲線を示す。単調減少かつ凸であり、$p = 1$ で 0 となり、$p \to 0$ で発散する。4 つ目のカードの報告であるデッキ全体の並び順は、この曲線のはるか左方の $\ln 52!$ に位置する。その値は対数ガンマ関数（log-gamma function）を用いて $\ln 52! = \ln \Gamma(53)$ として計算できるため、68 桁の整数 52! を直接作る必要はない。

```
deck_order = math.lgamma(53)          # ln 52! without forming 52!
print(f'the order of the deck: {deck_order:.2f} nats '
      f'= {deck_order / math.log(2):.1f} bits')
```

```
the order of the deck: 156.36 nats = 225.6 bits
```

シャッフルされたデッキの並び順を指定することは 156 nat の情報量を持ち、これは公正なコイン投げ約 226 回分に相当する。

37.1.1.2 エントロピー

サプライザルは結果が観測された後にしか計算できない。事前の段階では、平均してどれほど驚くと期待されるかを問うことができる。X を、値 x_i を確率 p_i でとる離散確率変数とする。その**エントロピー**は、結果の自己情報量の期待値として次の式で定義される。

$$H(X) = -\mathbb{E}_{x\sim P}[\log p(x)] = -\sum_i p_i \log p_i, \tag{37.1}$$

ここで規約として $0\log 0 = 0$ と定めるが、これは連続性（$t\log t \to 0$ が $t \to 0$ で成り立つこと）によって正当化される。すなわち、起こり得ない結果の寄与は 0 である。エントロピーは分布 P（X の分布）にのみ依存するため、$H(P)$ と $H(X)$ を区別せずに用いる。

有限集合上では、ビット単位のエントロピーは具体的な操作的意味を持つ。すなわち、結果を特定するために優れた質問者が行う「はい・いいえ」の質問の平均回数を表す。分布 $P = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$（4 つの記号 a, b, c, d 上の分布）を考えよう。本節を通してこの例を用いることとし、すべての確率が $\frac{1}{2}$ のべき乗であることから、これを **2 進情報源**（dyadic source）と呼ぶ。優れた質問者はまず「それは a か」と尋ね、半分の確率でその場で結果を確定させる。そうでなければ「それは b か」、さらに「それは c か」と尋ね、ここでも「いいえ」であれば d だけが残る。質問回数の期待値は次のとおりである。

$$\tfrac{1}{2}\cdot 1 + \tfrac{1}{4}\cdot 2 + \tfrac{1}{8}\cdot 3 + \tfrac{1}{8}\cdot 3 = 1.75,$$

これはビット単位のエントロピー $H_2(P) = \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + 2\cdot\frac{1}{8}\log_2 8 = 1.75$ ビット（すなわち 1.213 nat）と一致する。各結果は $\log_2(1/p_i)$ 回の質問で特定され、これはビット単位の自己情報量にほかならない。

対照的に、確率を無視する質問者は「a または b か」と尋ねてから「その 2 つのうちどちらか」と尋ねるため、常に 2 回の質問を必要とする。ここで、質問戦略とは実は姿を変えた 2 元符号（binary code）であることに注目しよう。「はい」に 1、「いいえ」に 0 を割り当てれば、結果を特定する回答の系列がそのまま符号語（codeword）となり、ここでは 1、01、001、000 がそれぞれ a、b、c、d に対応する。ここで平均質問回数がエントロピーとちょうど一致したのは、すべての確率が $\frac{1}{2}$ のべき乗だったからである。符号の言葉で言えば、この回数が大きく外れることは決してないことが 37.2 節で証明される。有限集合上のいかなる分布に対しても、最良の戦略が必要とする平均質問回数は H_2 から $H_2 + 1$ 回の範囲に収まる。

コードでは、関数 `entropy` が確率 0 の要素をマスクして除外することで $0\log 0 = 0$ を実装している。まずは 2 進情報源から始め、本節を通してこの関数を再利用する。

```python
def entropy(p):
    """Entropy of a probability vector, in nats (0 log 0 = 0 by masking)."""
    p = np.asarray(p, dtype=float)
    p = p[p > 0]
    return float(-np.dot(p, np.log(p)))

p_dyadic = np.array([1 / 2, 1 / 4, 1 / 8, 1 / 8])
h = entropy(p_dyadic)
print(f'H = {h:.3f} nats = {h / math.log(2):.2f} bits')
```

```
H = 1.213 nats = 1.75 bits
```

質問で数えたのと同じ 1.75 ビットが返される。

$0 \le p_i \le 1$ であるため、各項 $-p_i \log p_i$ は非負であり、したがって $H(X) \ge 0$ が成り立つ。等号成立は点質量（point mass、すなわち結果が確定している変数）の場合に限られる。また、エントロピーは確率の並びだけに依存し、結果の数値や名前には依存しないことにも注目したい。値 1、100、10^6、10^9 をこの 2 進情報源の確率でとる変数も、エントロピーは同じく 1.75 ビットである。

37.1.1.3 結合エントロピーと条件付きエントロピー

機械学習では、ラベルは常に入力と対になって現れ、文字はそれに先行するテキストの後に続く。したがって、2 つの変数を合わせたエントロピーと、一方の変数が既知となったときに他方に残るエントロピーの両方が必要になる。前者には新しい概念は不要である。対 (X, Y)（同時確率質量関数 $p(x, y)$ を持つ）は単に結果が対である 1 つの確率変数にすぎないため、その**結合エントロピー**（joint entropy）は $H(X, Y) = -\mathbb{E}[\log p(X, Y)]$ である。**条件付きエントロピー**（conditional entropy）は、Y に関して、X が観測された後に残る平均サプライザルであり、次の式で定義される。

$$H(Y \mid X) = -\mathbb{E}[\log p(Y \mid X)] = \sum_x p(x)\, H(Y \mid X = x),$$

ここで、$H(Y \mid X = x)$ は条件付き分布 $p(\cdot \mid x)$ のエントロピーである。$\log p(x, y) = \log p(x) + \log p(y \mid x)$ が成り立つため、両辺の期待値をとるとエントロピーの**連鎖律**（chain rule）が得られる。

$$H(X, Y) = H(X) + H(Y \mid X). \tag{37.2}$$

言葉で言えば、対のエントロピーは、第 1 の変数のエントロピーに、第 1 の変数がわかった後に第 2 の変数に残る不確実性を加えたものである。連鎖律を繰り返し適用すると、$H(X_1, \dots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_1, \dots, X_{i-1})$ が得られる。すなわち、系列のエントロピーは、各記号をそれ以前のすべての記号で条件付けたエントロピーの総和である。言語モデルは、これと同じ左から右への順序で、1 記号ずつテキストを予測する。

質問ゲームはこの恒等式を直接的に例示している。X を最初の質問「それは a か」への回答とし、Y を 2 進情報源の結果とする。この回答は公正なコインと同じであるため、$H_2(X) = 1$ ビットである。「はい」の後はもう尋ねることは残らない（0 ビット）。「いいえ」の後は、結果は b、c、d のいずれかであり、その確率は $(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$、エントロピーは 1.5 ビットである。2 つの回答にわたって平均をとると、$H_2(Y \mid X) = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1.5 = 0.75$ ビットとなる。連鎖律により、$H_2(X, Y) = 1 + 0.75 = 1.75$ ビットであり、最初の回答が 1 ビットを与え、残りの質問が残りを与える。この和は当然ながら $H_2(Y)$ に等しい。結果 Y を知れば回答 X は完全に定まるため $H(X \mid Y) = 0$ であり、逆順の連鎖律は $H(X, Y) = H(Y) + 0$ となるからである。

37.1.1.4 有限集合上の最大エントロピー

k 個の結果を持つ変数のエントロピーは、最大でどれほど大きくなり得るだろうか。最小の場合である 1 回のコイン投げから始めよう。表の出る確率が p のベルヌーイ変数は、次の **2 値エントロピー**（binary entropy）を持つ。

$$H_b(p) = -p \log p - (1 - p) \log(1 - p),$$

これは確率変数ではなくスカラー p の関数である。Figure 37.2 にこの曲線を示す。結果が確定する $p \in \{0, 1\}$ でエントロピーは 0 になり、公正なコインで最大値 $H_b(\frac{1}{2}) = \ln 2 \approx 0.693$ nat（1 ビット）に達する。$p = 0.1$ の偏りのあるコインでは $H_b(0.1) \approx 0.325$ nat となるが、これは結果がはるかに当てやすいためである。

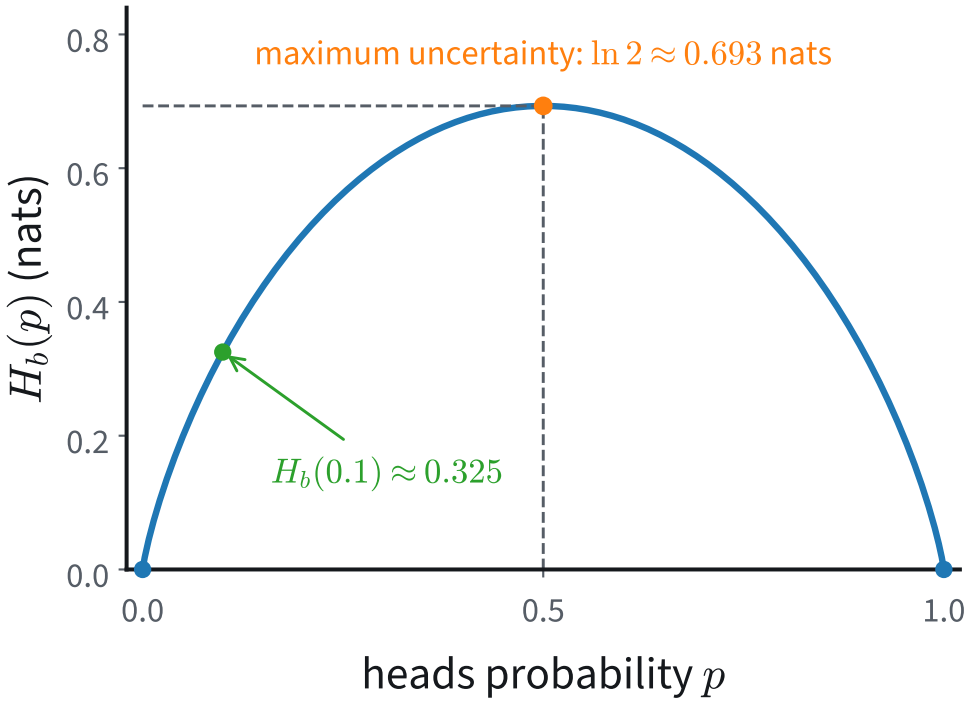


Figure 37.2: nat 単位の 2 値エントロピー $H_b(p)$。$p = \frac{1}{2}$ に関して対称であり、そこで最大値 $\ln 2 \approx 0.693$ nat をとる。決定論的な両端 $p = 0$ および $p = 1$ では結果に驚きがなく、エントロピーは 0 になる。曲線は凹であり、任意の弦はその下側にある。

この曲線が凹であることに注目しよう。曲線上の任意の 2 点を結ぶ弦は曲線の下側にある。曲線は一様なコインで最大となり、同じ原理が任意の有限個の結果に対しても成り立つ。X が高々 k 個の値をとるならば、次が成り立つ。

$$H(X) \le \log k,$$

等号が成り立つのは、X が k 個すべての値の上で一様分布に従うとき、かつそのときに限る。一様分布は確率質量を可能な限り均等に広げるため、結果の推測が最も難しくなる。4 つの結果に対する上界は $\log_2 4 = 2$ ビットであり、これは 0.25 ビットだけ 2 進情報源の 1.75 ビットを上回る。証明はギブスの不等式を導入するまで少しの間だけ先送りしよう。ギブスの不等式が揃えば、わずか 1 行で証明できる。その不等式は 2 つの分布を比較するものであるため、次は分布の比較へと進む。

37.1.2 KL ダイバージェンスと交差エントロピー

37.1.2.1 カルバック・ライブラー・ダイバージェンス

モデル Q が真の分布 P をどれほど悪く近似しているかを測りたいとしよう。情報理論はこれに**カルバック・ライブラー（KL）ダイバージェンス**（別名**相対エントロピー**（relative entropy））で答える (Kullback and Leibler 1951)。これは、Q を信じている人が、実際には P からサンプリングされる結果に対して被る余分なサプライザルの平均である。質問ゲームにおいて、ビット単位の KL ダイバージェンスは、Q 向けに設計された戦略が P からサンプリングされる結果に対して必要とする余分な質問の平均回数（整数の質問回数への丸めを除く）にあたる。P と Q が確率質量関数（または確率密度）p と q を持つとする。P から Q への KL ダイバージェンスは次の式で定義される。

$$D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_{x\sim P}\left[\log \frac{p(x)}{q(x)}\right]. \tag{37.3}$$

期待値の中の量は $\log p(x) - \log q(x) = (-\log q(x)) - (-\log p(x))$ であり、これは Q を信じる人のサプライザルから P を信じる人のサプライザルを引いた差である。この差は、Q が結果を過小評価する場所（$q(x) < p(x)$）で正となり、Q が過大評価する場所で負となる。KL ダイバージェンスは、この差を真の分布 P のもとで平均する。

P がサンプルと分子の両方を与えるため、この定義は非対称であり、一般に $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$ である。$p(x) = 0$ となる結果は、規約 $0\log(0/q) = 0$ に従い何も寄与しない。しかし、$p(x) > 0$ かつ $q(x) = 0$ となる結果が 1 つでもあれば、$D_{\mathrm{KL}}(P\|Q) = +\infty$ となる。

小さな数値例を見れば定義が具体的になるため、本節の残りではこの例を引き続き用いる。3 つの結果上の真の分布を $P = (0.6, 0.3, 0.1)$、モデルを $Q = (0.2, 0.5, 0.3)$ とする。このとき次が得られる。

$$D_{\mathrm{KL}}(P\|Q) = 0.6\ln\frac{0.6}{0.2} + 0.3\ln\frac{0.3}{0.5} + 0.1\ln\frac{0.1}{0.3} = 0.659 - 0.153 - 0.110 = 0.396 \text{ nats},$$

一方、逆方向は $D_{\mathrm{KL}}(Q\|P) = 0.365$ nat となる。ビット単位では、2 つの方向はそれぞれ 0.571 と 0.527 である。両者の値が異なるだけでなく、1 つ目の値からは、Q を信じる質問者が 1 回の結果あたり必要より 0.57 回多く質問してしまうことがわかる（P から得られる結果の長い列で平均した場合）。

以下は離散 KL ダイバージェンスのコードであり、`entropy` と同様に $0\log 0 = 0$ を P の確率 0 の要素をマスクして除外することで実装している。通し例である対に対して両方向で評価するとともに、3 番目の結果を起こり得ないとするモデル $Q' = (0.5, 0.5, 0)$ に対しても評価する。

```
def kl_divergence(p, q):
    """KL(P || Q) for two probability vectors, in nats; inf where q = 0 < p."""
    p, q = np.asarray(p, dtype=float), np.asarray(q, dtype=float)
    m = p > 0
    if np.any(q[m] == 0):
        return float('inf')
    return float(np.dot(p[m], np.log(p[m] / q[m])))

p = np.array([0.6, 0.3, 0.1])
q = np.array([0.2, 0.5, 0.3])
```

```
print(f'KL(P||Q) = {kl_divergence(p, q):.4f} nats, '
      f'KL(Q||P) = {kl_divergence(q, p):.4f} nats')
print(f"KL(P||Q') = {kl_divergence(p, [0.5, 0.5, 0.0])}")
```

```
KL(P||Q) = 0.3961 nats, KL(Q||P) = 0.3653 nats
KL(P||Q') = inf
```

特に覚えておくべきは 2 行目の結果である。モデル Q' は 10 回に 1 回起こる結果に確率 0 を割り当てており、その KL ダイバージェンスは無限大になる。ある事象を不可能だと断言するモデルは、その事象が実際に起きたとき無限の驚きを被るのである。

37.1.2.2 ギブスの不等式

平均的な余分なサプライザルが負になることはあり得るだろうか。もしあり得るなら、誤ったモデルの方が真の分布そのものよりも平均して驚かないことになる。ギブスの不等式はそれを退ける (Gibbs 1902)。平均において真の分布を上回るモデルは存在しない。以降の議論はすべてこの結果に依拠するため、凹関数の平均の帰結としてその完全な証明を示そう。任意の凹関数 f と有限の平均を持つ任意の確率変数 X に対して、**イェンセンの不等式**（Jensen's inequality）は次のことを主張する。

$$\mathbb{E}[f(X)] \le f(\mathbb{E}[X]).$$

言葉で言えば、先に平均をとってから f を適用した方が大きな値になる。X が 2 つの値をとるとき、これは単に Figure 37.2 において弦が曲線の下側にあるという事実にほかならない。f が狭義凹である場合、等号が成り立つのは X が確率 1 で定数となるときに限られる（一般の場合の証明は 35.3.3.1 節にある）。ここでイェンセンの不等式を必要とするのは、狭義凹関数である対数関数ただ 1 つに対してである。すなわち、平均の対数は対数の平均以上である。

命題（ギブスの不等式）：同じ空間上の任意の分布 P と Q に対して、次が成り立つ。

$$D_{\mathrm{KL}}(P\|Q) \ge 0,$$

等号が成り立つのは $P = Q$ のとき、かつそのときに限る。

証明：$q(x) = 0$ となる x が $p(x) > 0$ の範囲に存在すれば、ダイバージェンスは $+\infty$ となり証明すべきことはないため、そうでないと仮定する。ダイバージェンスを $\log(q(x)/p(x))$ の $x \sim P$ のもとでの平均にマイナスを付けた形に書き換え、対数関数にイェンセンの不等式を適用する。

$$\begin{aligned} D_{\mathrm{KL}}(P\|Q) &= -\mathbb{E}_{x\sim P}\left[\log \frac{q(x)}{p(x)}\right] \ge -\log \mathbb{E}_{x\sim P}\left[\frac{q(x)}{p(x)}\right] \\ &= -\log \sum_{x\,:\,p(x)>0} q(x) = -\log Q(\operatorname{supp} P) \ge 0. \end{aligned}$$

先頭のマイナス符号によって、イェンセンの不等式の「高々」が「少なくとも」に反転する。証明の残りは 2 行目で完結する。期待値は P の台（support）、すなわち $p(x) > 0$ を満たす結果の集合の上でとられる。因子 $p(x)$ が約分されて消え、その台の上での Q による確率の総和が残る。この値は高々 1 であるため、その負の対数は非負となる。連続密度に対しても、和を積分に置き換えれば同じ手順が成り立つ。等号が成り立つには、2 つの不等号がともに等号でなければなら

ない。対数関数は狭義凹であるため、1 つ目の不等号が等号になるのは、$q(x)/p(x)$ が定数 c に、P に関してほとんどすべての x で（すなわち P の確率が 0 の集合を除いて）等しい場合に限られ、2 つ目の等号には $Q(\operatorname{supp} P) = 1$ が必要となる。そこで $q = c\,p$ を P の台の上で足し合わせる（または積分する）と $c = 1$ が強制され、したがって（密度の場合はやはり確率 0 の集合を除いて）$q = p$ となる。$\blacksquare$

この証明はただ 1 つの着想に基づいている。真の分布のもとで平均すると、比 q/p は高々 1 であり、したがってその対数の平均は高々 0 となり、等号成立は q/p がまったく変動しない場合（すなわち $q = p$）に限られる。モデルが余分なサプライザルを一切生まないのは、モデルが正しい場合だけである。

ギブスの不等式により、証明を残していた一様分布の上界も直ちに示される。任意の分布 P（k 個の結果上）と同じ集合上の一様分布 U に対して $D_{\mathrm{KL}}(P\|U) = \sum_i p_i \log(k\,p_i) = \log k - H(P)$ が成り立ち、これは非負であって $P = U$ のときに限り 0 になる。

また、条件付けによって平均的なエントロピーが増えることはあり得ないことも証明できる。対 (X, Y) の同時分布と、その周辺分布（marginal distribution）の積を比較しよう。対数比は $\log \frac{p(x,y)}{p(x)\,p(y)} = \log p(y \mid x) - \log p(y)$ であり、同時分布のもとでのその期待値は $H(Y) - H(Y \mid X)$ となる。この期待値は KL ダイバージェンスであるため、次が成り立つ。

$$H(Y) - H(Y \mid X) = D_{\mathrm{KL}}(p_{X,Y} \,\|\, p_X\, p_Y) \geq 0,$$

等号が成り立つのは、X と Y が独立なとき、かつそのときに限る。平均的には、X を知ることで Y がより不確実になることは決してない。ただし、**個別の**観測では不確実性が増すこともあり得る。Y は、$X = 0$ のとき確実に裏となり、$X = 1$ のとき公正なコインになるとし、$X = 1$ は 5 回に 1 回の割合で起こるとしよう。全体として、Y が表になる確率は 0.1 であるため、$H(Y) = H_b(0.1) \approx 0.325$ nat である。$X = 1$ を観測すると Y に関する不確実性は $\ln 2 \approx 0.693$ nat へと上昇するが、条件付きエントロピーは X の両方の結果にわたって平均をとるため、$H(Y \mid X) = 0.2 \cdot 0.693 \approx 0.139$ nat と、$H(Y)$ の半分未満になる。

37.1.2.3 交差エントロピー

分類器の訓練には、KL ダイバージェンスと密接に関係する量が用いられる。**交差エントロピー**（P から Q への交差エントロピー）は、Q を信じる人が P からサンプリングされる結果に対して経験する平均サプライザルであり、次の式で定義される。

$$\mathrm{CE}(P, Q) = -\mathbb{E}_{x \sim P}[\log q(x)]. \tag{37.4}$$

期待値の中で $\log p(x)$ を足して引くと（これは $H(P)$ が有限である限り正当化される）、モデルのサプライザルは真の分布自身のエントロピーと過剰なダイバージェンスの和に分解される。

$$\mathrm{CE}(P, Q) = H(P) + D_{\mathrm{KL}}(P\|Q). \tag{37.5}$$

したがって交差エントロピーは、削減不能なエントロピーの下限と非負の過剰分の和である。ギブスの不等式により $\mathrm{CE}(P, Q) \geq H(P)$ が成り立ち、等号成立は $Q = P$ のとき、かつそのときに限る。通し例の対では、$\mathrm{CE}(P, Q) = -(0.6 \ln 0.2 + 0.3 \ln 0.5 + 0.1 \ln 0.3) = 1.294$ nat、$H(P) = 0.898$ nat であり、両者の差 0.396 nat は先ほど計算した KL ダイバージェンスに一致する。$H(P)$ は Q に依存しないため、交差エントロピーを Q に関して最小化することは KL ダイバージェンスを最小化することと同値である。離散データに対する最尤法はまさにこれを行っ

ており、その平均負の対数尤度（negative log-likelihood）は経験データ分布からモデルへの交差エントロピーに等しい（36.6.2.1 節）。

Figure 37.3 は、2 進情報源 P と一様モデル $U = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ に対する質問ゲームでこの分解を図示したものである。P に適合した質問を行えば、平均して $H_2(P) = 1.75$ 回の質問で済む。一様な戦略のもとで固定した 2 問を尋ねると $\mathrm{CE}_2(P, U) = 2$ 回かかるため、無駄は $D_{\mathrm{KL}}(P \| U) = 0.25$ ビットとなる。質問戦略は 2 元符号であるため、これらの数値はそのまま 1 記号あたりのビット単位の平均符号語長でもある（なお、この特定の対では逆方向のダイバージェンス $D_{\mathrm{KL}}(U \| P)$ もたまたま 0.25 ビットに等しくなるため、先ほど KL ダイバージェンスの非対称性を示す際には別の対を用いた）。

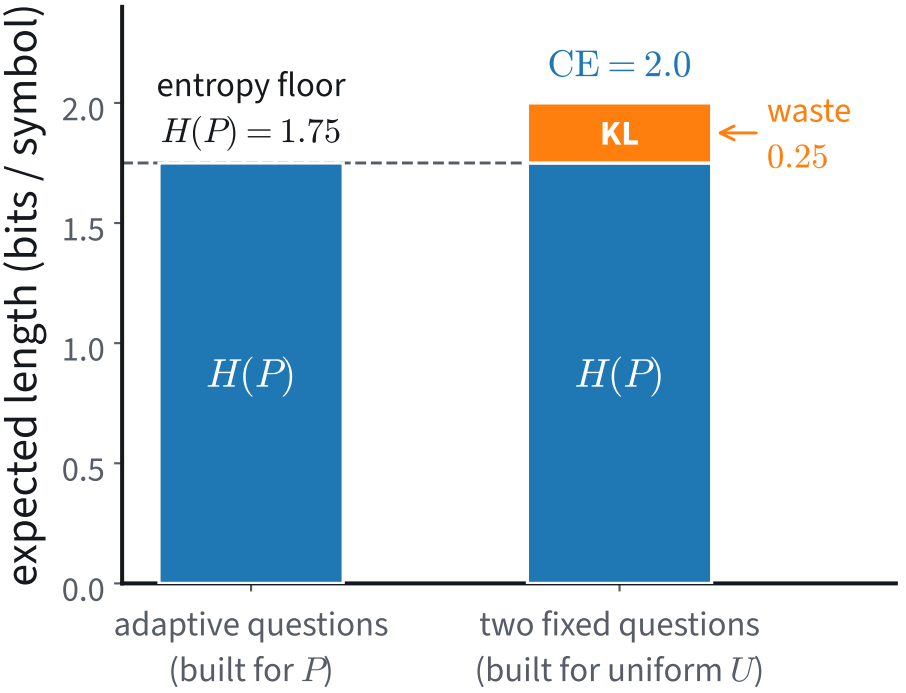


Figure 37.3: 分解 Equation 37.5 を 2 進情報源 $P = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$ と一様モデル U に対してビット単位で表した図。P に適合した質問では平均 $H_2(P) = 1.75$ 問、固定した 2 問では $\mathrm{CE}_2(P, U) = 2.0$ 問を要し、無駄は $D_{\mathrm{KL}}(P \| U) = 0.25$ ビットである。

37.1.2.4 KL ダイバージェンスの連鎖律

エントロピーは対に沿って分解される。すなわち、(X, Y) の驚きは X の驚きと Y に残る驚きの和である（37.1.1.3 節）。KL ダイバージェンスもまったく同じ理由で同様に分解される。$P_{X,Y}$ と $Q_{X,Y}$ を 2 つの同時分布とし、それぞれ $p(x, y) = p(x)\, p(y \mid x)$ および $q(x, y) = q(x)\, q(y \mid x)$ と分解する。このとき比 $p(x, y)/q(x, y)$ の対数は 2 項の和となり、$P_{X,Y}$ のもとで期待値をとると KL ダイバージェンスの**連鎖律**が得られる。

$$D_{\mathrm{KL}}(P_{X,Y} \| Q_{X,Y}) = D_{\mathrm{KL}}(P_X \| Q_X) + \mathbb{E}_{x \sim P_X}\left[D_{\mathrm{KL}}(P_{Y|x} \| Q_{Y|x})\right]. \tag{37.6}$$

対に関する誤ったモデルは、まず X についての誤差の代償を払い、次に Y について（X が既知

となったもとでの）誤差の代償を平均的に払う。なお、この平均はモデルではなく真の分布 P_X にわたってとられる。この恒等式自体は 1 行の式変形で得られるが、本書の以降の議論はその 3 つの帰結に大きく依拠している。

第 1 に、KL ダイバージェンスは独立な座標にわたって加法的に足し合わされる。X と Y が P と Q の両方で独立であれば、条件付きの項はもはや x に依存せず、対のダイバージェンスは 2 つのダイバージェンスの和になる。したがって、通し例の対から独立に 10 回サンプルを引くと、P と Q は $10 \cdot 0.396 = 3.96$ nat だけ引き離される。長いサンプルの方が短いサンプルよりも 2 つの分布をはるかに明確に識別できるのはこのためである。第 2 に、条件付きの項はギブスの不等式により非負であるため、2 つの同時分布間のダイバージェンスは少なくともそれらの周辺分布間のダイバージェンス以上になる。変数を捨てることは、2 つの分布の識別をより難しくする方向にしか働かない。潜在変数モデル（latent variable model）はこの性質を利用している。データと潜在変数にわたる同時分布のダイバージェンスは定数差を除けば計算可能な目的関数であり、それはデータ単独のダイバージェンスを条件付き項の分だけ上回る。この差がエビデンス下界（evidence lower bound）のギャップである（36.7 節）。第 3 に、この規則を系列に対して繰り返し適用すると次が得られる。

$$D_{\mathrm{KL}}(P_{X_1,\dots,X_n} \| Q_{X_1,\dots,X_n}) = \sum_{i=1}^{n} \mathbb{E}_{x_{<i} \sim P}\big[D_{\mathrm{KL}}(P_{X_i \mid x_{<i}} \| Q_{X_i \mid x_{<i}})\big],$$

したがって、系列全体に対する自己回帰モデル（autoregressive model）の過剰損失は、個々のステップにおける過剰損失の総和となる。それゆえ、次の各トークンをうまく予測することは系列全体をうまくモデル化することと同じ課題であり、37.2.4.1 節ではこの分解を用いてあらゆる言語モデルの損失を下から抑える。

37.1.3 分類損失

k クラス分類問題において、1 つの例に対するターゲットは記録されたラベル y にすべての質量を置くワンホット分布であり、モデルは予測確率分布 $\mathbf{q} = (q_1, \dots, q_k)$ を出力する。両者の間の交差エントロピー Equation 37.4 は単一の項 $-\log q_y$ へと縮約され、これは観測されたラベルに対するモデルのサプライザルそのものである。n 個の例（ラベル y_i と予測 $\mathbf{q}_i$）にわたって平均をとると、次の**交差エントロピー損失**が得られる。

$$\mathrm{CE} = -\frac{1}{n} \sum_{i=1}^{n} \log q_{i,y_i},$$

単位は nat である。深層学習フレームワークは、この損失を**ロジット z**（ソフトマックスをとると **q** になる非正規化スコア）から直接計算する。先に確率を計算すると大きなロジットでオーバーフローし、非常に小さな負のロジットで $\log 0$ へとアンダーフローするため、内部では数値的に安定な対数ソフトマックス（log-softmax）を適用している（3.3.5.2 節、安定な形式の導出は 35.8.2 節を参照）。続く各項では、この損失からデータ中のノイズ、確率の較正、およびソフト化したターゲットについて何がわかるかを解き明かす。

37.1.3.1 削減不能な損失

本節の冒頭で、一部のラベルが誤っているときに損失はどこまで下がり得るかと問いかけた。固定された訓練セット上では、その答えから有益な知見は得られない。データセットを丸暗記できるほど大きなニューラルネットワークであれば、誤ったものも含めて記録されたすべてのラベルに適合し、訓練損失を 0 に近づけられるからである。この問いが実質的な意味を持つのはモデル

がまだ見ていない未知の例に対してのみであり、そこではワンホットターゲットは真実のすべてではない。実データでは、ラベルが入力の決定論的な関数であることはまれであり、本質的な曖昧さやアノテーション誤りのために同じ画像に異なるラベルが付くことがある。入力 $\mathbf{x}$ に対する真のターゲットは、そのラベルが抽出される条件付き分布であり、記録されたラベルはそこからの 1 つのサンプルにすぎない。

入力 $\mathbf{x}$ とラベル $y \in \{1, \dots, k\}$ が同時分布に従い、$p(\cdot \mid \mathbf{x})$ がラベルの真の条件付き分布、$q(\cdot \mid \mathbf{x})$ が任意のモデルの予測であるとする。未知の例の数が増えるにつれ、その平均損失は $(\mathbf{x}, y)$ にわたる期待値へと収束する。これを**母集団損失**（population loss）と呼ぶ。いかなるモデルも、この母集団損失を入力が与えられたもとでのラベルの平均エントロピーより下に押し下げることはできない。

命題（削減不能な分類損失）：任意のモデルの母集団交差エントロピー損失は次を満たす。

$$\begin{aligned} \mathbb{E}_{(\mathbf{x},y)}\big[-\log q(y \mid \mathbf{x})\big] &= \mathbb{E}_{\mathbf{x}}\big[H(p(\cdot \mid \mathbf{x}))\big] + \mathbb{E}_{\mathbf{x}}\big[D_{\mathrm{KL}}(p(\cdot \mid \mathbf{x}) \,\|\, q(\cdot \mid \mathbf{x}))\big] \\ &\geq \mathbb{E}_{\mathbf{x}}\big[H(p(\cdot \mid \mathbf{x}))\big], \end{aligned} \tag{37.7}$$

等号が成り立つのは、$q(\cdot \mid \mathbf{x}) = p(\cdot \mid \mathbf{x})$ がほとんどすべての $\mathbf{x}$ に対して（すなわち確率 0 の集合を除くすべての入力に対して）成り立つとき、かつそのときに限る。

証明： $\mathbf{x}$ で条件付ける。y にわたる内側の期待値は交差エントロピー $\mathrm{CE}(p(\cdot \mid \mathbf{x}), q(\cdot \mid \mathbf{x}))$ であるため、分解 Equation 37.5 が各 $\mathbf{x}$ において点ごとに適用できる（すべてのエントロピーは高々 $\log k$ で有限である）。外側の期待値を $\mathbf{x}$ にわたってとれば等式が得られる。ギブスの不等式によりすべての KL 項は非負であり、非負の項の平均が 0 になるのは、そのほとんどすべてが 0 であるとき、かつそのときに限る。■

この下限は、入力が与えられたもとでのラベルの条件付きエントロピーである（37.1.1.3 節）。これはデータ固有の性質であり、いかなるアーキテクチャ、追加の訓練データ、あるいはより長い訓練スケジュールによっても下げることはできない。本節のすべての結果のなかで、損失曲線を読むたびに念頭に置くべきなのはこの事実である。以下の 3 つの事例が、その意味するところを具体的に示している。

- 一様分布を予測するモデルは、ラベルにかかわらずすべての入力で $\ln k$ の交差エントロピーを被り、おなじみの 2.30 nat（$k = 10$ のとき）を与える。
- ラベルが入力の決定論的な関数であるとき、すべての $H(p(\cdot \mid \mathbf{x}))$ は 0 となり、完璧なモデルは損失 0 を達成する。
- 対称なラベルノイズのもとでは、クリーンなクラスは入力によって定まる一方、記録されるラベルは確率 $1-\rho$ でクリーンなクラスと一致し、残りの $k-1$ 個の各クラスにはそれぞれ確率 $\rho/(k-1)$ で割り振られる。どの入力においても、条件付きエントロピーは $H = -(1-\rho)\log(1-\rho) - \rho \log \frac{\rho}{k-1} = H_b(\rho) + \rho \log(k-1)$ となる。$\rho = 0.1$ かつ $k = 10$ のとき、この下限は $0.325 + 0.220 = 0.545$ nat である。$1-\rho > \rho/(k-1)$ である限りクリーンなクラスを予測することが最適であり続けるため、達成可能な最高の正解率は $1-\rho$（ここでは 90%）となる。

次のセルを実行する前に少し立ち止まり、このノイズのある設定でクリーンなクラスに確率 0.99 を割り当てる確信度の高い予測器の損失を予想してみてほしい。この予測器は 10 回中 9 回は正解し、しかも毎回ほぼ確信している。以下のコードでは、これを一様予測器、および真の条件付き分布 $q = p$ を出力する**ベイズ予測器**（Bayes predictor）と比較する。

```
k, rho = 10, 0.1
p_noisy = np.array([1 - rho] + [rho / (k - 1)] * (k - 1))  # clean class first
```

```
    predictors = {'uniform': np.full(k, 1 / k),
                  'confident 0.99': np.array([0.99] + [0.01 / (k - 1)] * (k - 1)),
                  'Bayes q = p': p_noisy}
    print(f'floor = H(p) = {entropy(p_noisy):.3f} nats')
    for name, q_hat in predictors.items():
        ce = float(-np.dot(p_noisy, np.log(q_hat)))
        print(f'{name:15s} CE = {ce:.3f} nats')
```

```
floor = H(p) = 0.545 nats
uniform         CE = 2.303 nats
confident 0.99  CE = 0.689 nats
Bayes q = p     CE = 0.545 nats
```

過信した予測器は 0.689 の損失を被り、下限を大きく上回る。反転したラベルのたびに不意を突かれ、その高い確信度が反転した 10 分の 1 のラベルで支払うコストは、残りのラベルで得る利益をはるかに上回るからである。下限に到達するのはベイズ予測器だけである。

これで冒頭の問いに答えられる。最終的な損失 0.6 は良い値なのだろうか。下限が 0 であるクリーンなラベルに対しては、0.6 にはまだモデルを改善する余地が十分に残っている。しかし、このようなノイズのあるラベルに対しては下限から 0.06 nat 以内に位置しており、改善の余地はほとんど残っていない。下限でのプラトーは単にデータの性質であり、下限より上でのプラトーだけがモデルまたはその訓練の限界を示す。なお、以上の議論はすべて未知のデータに対するものであることを忘れないでほしい。ホールドアウト分割において下限を**下回る**損失が出た場合、それは優れたモデルによるものではあり得ない。それが代わりに何を意味するかは演習問題 2 で問う。

ここまでは、単一のペナルティ（$-\log q_y$）、そのままのワンホットラベル、および可算個の結果に対して損失の下限を導出した。本節の残りでは、これら 3 つの仮定を順に緩め、他のペナルティ、ソフト化したターゲット、そして連続な実数値の結果を調べていく。

37.1.3.2 固有スコアリングルール

損失の下限が真の条件付き確率によってのみ達成されるという性質は、分類器の出力を確率として解釈したい場合にまさに望まれるものである。決定理論ではこの性質に名前が付けられている。**スコアリングルール**は、ペナルティ $S(\mathbf{q}, y)$ を、確率ベクトル $\mathbf{q}$ を報告して実際にクラス y が生じた場合に対して課す規則である。正直な報告が最適であるとき、すなわちラベルが $\mathbf{p}$ に従うもとで期待ペナルティ $\mathbb{E}_{y\sim\mathbf{p}}[S(\mathbf{q}, y)]$ が $\mathbf{q} = \mathbf{p}$ を報告することで最小化されるとき、そのルールは**固有**（proper）であるという。さらに、この正直な報告が唯一の最小点であるとき、**狭義固有**（strictly proper）であるという。交差エントロピー損失は**対数スコア**（log score）$S(\mathbf{q}, y) = -\log q_y$ であり、ギブスの不等式から狭義固有であることがわかる。その期待ペナルティは $\mathrm{CE}(\mathbf{p}, \mathbf{q})$ であり、$\mathbf{q} = \mathbf{p}$ で一意に最小化されるからである。

その結果、真の分布を含むほど十分に表現力の高いモデルクラス上で母集団最適解に達すれば、交差エントロピーで訓練された分類器は真の条件付き確率を出力する。とりわけ、その予測は**較正**（calibration）されている。すなわち、モデルがあるクラスに対して確率 0.8 と予測したすべての入力のうち、実際にそのクラスが生じる割合は 80% になる（ただし実際には、有限の訓練セットや制限されたモデルクラスのために、訓練済みニューラルネットワークの較正がずれることは依然として起こり得る）。

狭義固有なスコアリングルールは対数スコアだけではない。**ブライアスコア**（Brier score）$S(\mathbf{q}, y) = \sum_j (q_j - \mathbf{1}(j = y))^2$ も狭義固有であり、対数スコアとは異なって、$-\log q_y$ が発散するような確信度の高い誤りに対しても有界なペナルティしか課さない (Gneiting and Raftery

2007)。演習問題 3 ではその過剰ペナルティを計算し、一見もっともらしいが実は固有でないルールについても検討する。

37.1.3.3 ラベル平滑化

ここまで、訓練ターゲットはワンホットラベルであった。これをよりソフトなターゲットに置き換えると、損失がロジットに要求するものも損失の下限の位置も変化するが、そうすることには実用上の理由がある。ワンホットターゲットは、ソフトマックス層が実際には出力できないものを要求する。真のクラスに確率 1 を割り当てるには、そのロジットが他のすべてのロジットより無限に大きくなければならないからである。その結果、損失は有限の最小点を持たず、訓練が続く限り勾配降下法はロジット間の差を広げ続ける。**ラベル平滑化**（label smoothing）(Szegedy et al. 2016) は、ワンホットターゲット $\mathbf{e}_y$ に比率 ϵ で全 k クラス上の一様分布を混合することでこれを解消する。

$$\mathbf{p}^{\epsilon} = (1-\epsilon)\,\mathbf{e}_y + \frac{\epsilon}{k}\,\mathbf{1}. \tag{37.8}$$

このターゲットのもとでは、真のクラスには確率 $1-\epsilon+\epsilon/k$ が、他のすべてのクラスには ϵ/k が割り当てられる。訓練の目的関数は、この平滑化されたターゲットに対する通常の交差エントロピー $\mathrm{CE}(\mathbf{p}^{\epsilon}, \mathbf{q})$ である（訓練レシピを扱う 7.6 節で用いたものと同じ損失である）。これは $\mathbf{q} = \mathbf{p}^{\epsilon}$ で最小化され、ソフトマックスは真のクラスのロジットと他の各クラスのロジットとの間に有限のギャップを持たせることでこれを達成する。

$$z_y - z_j = \log\frac{1-\epsilon+\epsilon/k}{\epsilon/k} = \log\!\left(1 + \frac{k(1-\epsilon)}{\epsilon}\right).$$

$k = 10$ クラスかつ $\epsilon = 0.1$ のとき、この最適なロジットギャップは $\ln 91 \approx 4.51$ であり、最適点での損失は平滑化されたターゲットのエントロピー $H(\mathbf{p}^{\epsilon}) = 0.500$ nat に等しい。

平滑化を行わないときにロジットがどれほどの速さでドリフトするかを見ておくのは有益である。ロジットが 0 から始まり、ステップサイズ 1 の勾配降下法で更新される 1 つの例を考えよう。ロジットに関する交差エントロピーの勾配は $\mathbf{q} - \mathbf{p}^{\epsilon}$ である。ワンホットターゲットでは、誤ったクラスはすべて同値のまま保たれ、各ステップで真のクラスと残りのクラスとのギャップ g は $(1-q_y) + q_j = k/(e^{g} + k - 1)$ だけ広がる。この増分は約 ke^{-g} となる（g が大きくなったとき）。g を ke^{-g} だけ増やすと、e^{g} には $e^{ke^{-g}} \approx 1 + ke^{-g}$ が掛かり、その結果、各ステップで約 k が e^{g} に加算される。したがって t ステップ後には $e^{g} \approx kt$、すなわち $g \approx \ln(kt)$ になると予測される。ギャップの増大は決して止まらないが、その成長は対数的であり、ステップ数を 10 倍にしても $\ln 10 \approx 2.30$ しか増えない。10,000 ステップ後には $g \approx \ln 10^{5} \approx 11.51$ となるはずである。次のセルでは、平滑化なしと平滑化ありの両方でこの勾配降下法を実行する。

```
k, checkpoints = 10, (10, 100, 1000, 10000)
for eps in (0.0, 0.1):
    p_eps = np.full(k, eps / k)
    p_eps[0] += 1 - eps                  # the true class first
    z = np.zeros(k)                      # the logits of one example
    for step in range(1, checkpoints[-1] + 1):
        q = np.exp(z - z.max())
        z -= q / q.sum() - p_eps         # gradient step, step size 1
        if step in checkpoints:
            log_q = z - z.max() - np.log(np.exp(z - z.max()).sum())
```

```
            print(f'eps = {eps}  step {step:5d}  gap = {z[0] - z[1]:5.2f}  '
                  f'loss = {-p_eps @ log_q:.4f}')
    print(f'optimum for eps = 0.1: gap = ln 91 = {math.log(91):.2f}, '
          f'loss = H(p_eps) = {entropy(p_eps):.4f}')
```

```
eps = 0.0  step    10  gap =  4.29  loss = 0.1158
eps = 0.0  step   100  gap =  6.87  loss = 0.0093
eps = 0.0  step  1000  gap =  9.21  loss = 0.0009
eps = 0.0  step 10000  gap = 11.51  loss = 0.0001
eps = 0.1  step    10  gap =  3.81  loss = 0.5243
eps = 0.1  step   100  gap =  4.51  loss = 0.5003
eps = 0.1  step  1000  gap =  4.51  loss = 0.5003
eps = 0.1  step 10000  gap =  4.51  loss = 0.5003
optimum for eps = 0.1: gap = ln 91 = 4.51, loss = H(p_eps) = 0.5003
```

ワンホットターゲットでは、ロジットギャップは 4.29、6.87、9.21、11.51 に達する。近似式 $\ln(kt)$ の予測値は 4.61、6.91、9.21、11.51 であり、実際のギャップは（ke^{-g} の近似がまだ粗い）初期段階では予測よりわずかに小さいものの、$1{,}000$ ステップ以降は小数第 2 位まで一致する。その間、損失は決して 0 に到達することなく少しずつ下がり続ける。訓練セットに適合できるほど大きなニューラルネットワークでは、この際限のないドリフトは、成長し続ける重みと、誤ラベルの例も含めたすべての訓練例に対する 1 付近の予測確率として現れる。対照的に $\epsilon = 0.1$ では、ギャップは 4.51 に落ち着いて（ステップ 100 までに到達）そこに留まり、損失も 0.500 で静止する。

これで訓練は有限の最適解を持つようになるが、平滑化を用いたモデルの解釈には注意が必要である。平滑化されたターゲットを誤差なく予測できたとしても、訓練損失は 0 ではなく $H(\mathbf{p}^{\epsilon})$ でプラトーに達する。さらに、その出力はもはや真の条件付き確率ではなく、どのクラスも $1 - \epsilon + \epsilon/k$（ここでは 0.91）を超える確率を受け取ることはない。実務では $\epsilon = 0.1$ が標準的な選択肢であり、7.6 節の訓練レシピでもこれを採用している。演習問題 4 では、ラベル平滑化が誤ったクラスのロジットにどのような影響を与えるかを調べる。

37.1.3.4 知識蒸留

ソフトなターゲットは、別の訓練済みモデルから得ることもできる。**知識蒸留**（knowledge distillation）(Hinton et al. 2015) は、より大きな**教師**（teacher）ネットワークを模倣するようにコンパクトな**生徒**（student）ネットワークを訓練する手法であり、26.1 節（英語版） では言語モデルに適用した。生徒はワンホットラベルで訓練する代わりに、クラス全体にわたる教師の確率分布そのものをターゲットとして訓練する。手書きの「2」が「0」よりも「7」にはるかに似ているという事実は、ワンホットラベルが捨ててしまう情報であるが、教師の誤クラスに対するロジットの中には残っている（ラベル平滑化で訓練された教師がこの情報をあまり伝えられない理由は、演習問題 4 で問う）。

任意の固定されたターゲット分布の場合と同様、損失は生徒がターゲット自体を再現したときに最小化されるため、分解 Equation 37.5 からここで新たに付け加わることはない。説明を要するのは損失のスケールであり、そこには一見すると恣意的に見える係数が掛かっている。

よく訓練された教師のソフトマックス出力はほぼワンホットであるため、誤クラスに割り当てられる確率は通常きわめて小さく、生徒を十分に引き寄せる力を持たない。そこでまず、ソフトマックスを適用する前にロジットを**温度**（temperature）$\tau > 0$ で割ることで、両方の分布を**ソフト化**する。

$$p_j^{(\tau)} = \mathrm{softmax}(\mathbf{z}^{\mathrm{tea}}/\tau)_j, \qquad q_j^{(\tau)} = \mathrm{softmax}(\mathbf{z}^{\mathrm{stu}}/\tau)_j.$$

$\tau > 1$ に設定すると、ロジット間のギャップが圧縮され、確率質量が誤クラス側へと移る。蒸留損失は、こうしてソフト化された分布間の KL ダイバージェンスに τ^2 を掛けたものである。

$$\mathcal{L}_{\text{distill}} = \tau^2 \, D_{\mathrm{KL}}\big(\mathbf{p}^{(\tau)} \,\|\, \mathbf{q}^{(\tau)}\big). \tag{37.9}$$

この損失が 0 になるのは、生徒がソフト化された教師と一致するとき、かつそのときに限る。なぜ係数 τ^2 が必要なのかを見るために、スケール前の KL 項を生徒のロジットに関して微分してみよう。生徒に依存するのは交差エントロピーの部分 $-\sum_j p_j^{(\tau)} \log q_j^{(\tau)}$ だけであり、温度 τ における対数ソフトマックスの導関数は、おなじみの「ソフトマックス引くターゲット」の式を $1/\tau$ 倍したものになる。

$$\frac{\partial}{\partial z_j^{\text{stu}}} D_{\mathrm{KL}}(\mathbf{p}^{(\tau)} \| \mathbf{q}^{(\tau)}) = \frac{1}{\tau}\big(q_j^{(\tau)} - p_j^{(\tau)}\big).$$

したがって、生徒の各ロジットは、そのクラスに対する生徒と教師のソフト化確率の差を τ で割った分だけ動かされる。

高温では、その確率の差自体も縮小する。平均ロジット $\bar{z}$ を引いてもソフトマックスは変化せず、大きな τ に対しては一次のテイラー展開により $e^{(z_j - \bar{z})/\tau} \approx 1 + (z_j - \bar{z})/\tau$ と近似できる。中心化された偏差 $z_j - \bar{z}$ の和は 0 であるため、ソフトマックスの分母は約 k となり、$\mathrm{softmax}(\mathbf{z}/\tau)_j \approx \frac{1}{k}\big(1 + (z_j - \bar{z})/\tau\big)$ が得られる。その結果、$q_j^{(\tau)} - p_j^{(\tau)}$ は $1/\tau$ のオーダーで縮小し、スケール前の勾配は $1/\tau^2$ のオーダーで縮小する。

$$\frac{1}{\tau}\big(q_j^{(\tau)} - p_j^{(\tau)}\big) \;\approx\; \frac{1}{k\,\tau^2}\Big(\big(z_j^{\text{stu}} - \bar{z}^{\text{stu}}\big) - \big(z_j^{\text{tea}} - \bar{z}^{\text{tea}}\big)\Big).$$

損失に τ^2 を掛けることでこの減衰が打ち消され、τ を変化させても勾配ノルムはほぼ一定に保たれる。蒸留損失が単独で使われることはまれであり、通常はハードラベルに対する標準的な交差エントロピー損失に固定の重みで加算されるため、この安定性は実務上不可欠である。τ^2 の係数がなければ、τ を調整した際に意図せず 2 つの損失の相対的な重みまで変わってしまう。この係数があることで、温度は教師の小さな誤クラス確率を生徒にどれだけ見せるかだけを制御できる。

教師ロジット $(5, 2, -1)$ と生徒ロジット $(3, 3, 0)$ を用いて、このスケーリングを数値的に確認しよう。どちらのベクトルも平均が 2 であるため、勾配の τ^2 倍は $\frac{1}{3}\big((1,1,-2) - (3,0,-3)\big) = (-0.667, 0.333, 0.333)$（ノルム 0.816）に収束するはずである。下のセルでは、自動微分を用いて 5 つの温度にわたってスケール前の KL 項の勾配を計算し、τ^2 を掛ける前後のノルムを比較する。

```
z_tea = torch.tensor([5.0, 2.0, -1.0])
z_stu = torch.tensor([3.0, 3.0, 0.0], requires_grad=True)
print(' tau  |grad|  |tau^2 grad|  tau^2 grad')
for tau in (1.0, 2.0, 5.0, 10.0, 20.0):
    p_tau = torch.softmax(z_tea / tau, 0)
    kl = (p_tau * (torch.log(p_tau) - torch.log_softmax(z_stu / tau, 0))).sum()
    g = torch.autograd.grad(kl, z_stu)[0].numpy()
    print(f'{tau:4.0f}  {np.linalg.norm(g):.4f}  '
          f'{np.linalg.norm(tau**2 * g):12.3f}  {np.round(tau**2 * g, 3)}')
limit = (z_stu - z_stu.mean()) - (z_tea - z_tea.mean())
limit = limit.detach().numpy() / len(limit)
```

```
    print(f'limit {"":6s}  {np.linalg.norm(limit):12.3f}  {np.round(limit, 3)}')
```

```
 tau  |grad|  |tau^2 grad|  tau^2 grad
   1  0.6391         0.639  [-0.462  0.441  0.022]
   2  0.2190         0.876  [-0.672  0.549  0.123]
   5  0.0368         0.920  [-0.741  0.478  0.263]
  10  0.0088         0.884  [-0.719  0.413  0.306]
  20  0.0021         0.854  [-0.697  0.374  0.322]
limit                0.816  [-0.667  0.333  0.333]
```

τ^2 の係数がないと勾配は崩壊する。そのノルムは 0.64（$\tau = 1$）から 0.0021（$\tau = 20$）へと約 300 分の 1 に落ち込み、最後に τ を 2 倍にしたときにはちょうど $1/\tau^2$ が予測するとおり約 4 分の 1 になっている。一方、係数を含めればノルムは 0.64 から 0.92 の間に留まる。漸近極限の 0.816 をわずかにオーバーシュートして $\tau = 5$ 付近でピークに達した後、$1/\tau$ のオーダーの速さでゆっくりと極限へ戻っていく（0.854 が $\tau = 20$ での値である）。また、$\tau = 1$ では損失が通常の KL ダイバージェンスに帰着し、さらにワンホットな教師（エントロピーが 0）に対しては通常の交差エントロピー損失になるため、標準的な教師あり訓練はその特別な場合となっていることにも注目したい。

37.1.4 確率密度に対するエントロピーと KL ダイバージェンス

ここまではすべての結果が離散かつ可算であった。対照的に、回帰目標は確率密度によって記述される実数であるため、和は積分に置き換わる。KL ダイバージェンスはこの移行後もそのままの性質を保つが、エントロピーはそうではない。連続なエントロピーは負の値になり得るうえ、変数を測定する単位に依存する。その結果、連続モデルの負の対数尤度損失は 0 を下回ることがあり、データをリスケーリングするたびにその下限もシフトする。絶対的な意味を保ち続けるのは下限からの過剰分であり、それに伴って同じデータ上で評価した任意の 2 つのモデル間の損失の差である（初読時には、続く 3 つの項を飛ばして本節の最後の段落から再開してもよい）。

37.1.4.1 微分エントロピー

連続確率変数 X（密度 p）に対し、離散エントロピーの和を積分に置き換えると**微分エントロピー**（differential entropy）が得られる。

$$h(X) = -\int p(x) \log p(x)\, dx. \tag{37.10}$$

これは負の対数密度の期待値であるが、密度の値は確率ではない。密度は $1/x$ の単位を持ち、容易に 1 を超え得る。微分エントロピーには小文字の h を用いて離散の H と区別するが、それは両者の振る舞いが大きく異なるからである。

最も単純な例からすでにその理由がわかる。区間 $[0, a]$ 上の一様密度は区間全体で $1/a$ に等しいため、$h = -\int_0^a \frac{1}{a} \log \frac{1}{a}\, dx = \log a$ となる。$a = \frac{1}{2}$ のとき、これは $-\ln 2 \approx -0.693$ nat となり、いかなる離散分布もとり得ない負のエントロピーを与える。さらに厄介なことに、まったく同じ長さを m ではなく cm で測ると、a が 100 倍になり、背後にある不確実性は一切変わっていないにもかかわらず $\ln 100 \approx 4.61$ nat が h に加算される。

これと同種のシフトは、任意の滑らかな変数変換のもとで生じる。$y = g(x)$ が可逆で導関数 g' を持つならば、Y の点 $g(x)$ における密度は $p_X(x)/|g'(x)|$ であり、したがって次が成り立つ。

$$h(Y) = h(X) + \mathbb{E}\big[\log |g'(X)|\big].$$

確率ベクトル $\mathbf{x} \in \mathbb{R}^n$ に対しても、積分領域を $\mathbb{R}^n$ とし、ヤコビ行列式 $|\det \partial g/\partial \mathbf{x}|$ で $|g'|$ を置き換えれば、定義もこの変換則もまったく同じ形を保つ。変数を引き伸ばせば微分エントロピーは増加し、圧縮すればエントロピーは際限なく下がっていく。37.1.4.3 節で導出するように、標準偏差 σ のガウス分布の微分エントロピーは $h = \tfrac{1}{2}\log(2\pi e\sigma^2)$ であり、$\sigma < (2\pi e)^{-1/2} \approx 0.24$ になった途端に負の値となる。

幸いなことに、KL ダイバージェンスはこのような単位への依存性を持たない。データ密度 p とモデル密度 q の両方に同じ変数変換を適用すると、両方とも同じヤコビ行列式の因子で割られるため、比 p/q の中でそれが相殺されるからである。したがって、KL ダイバージェンスは可逆な再パラメータ化に対して不変である。たとえば、2 つの一様密度 p_1（区間 $[0, a_1]$ 上）と p_2（区間 $[0, a_2]$ 上、$a_1 \le a_2$）に対しては $\int p_1 \log(p_1/p_2)\,dx = \log(a_2/a_1)$ となり、$a_1 = \tfrac{1}{2}$、$a_2 = 1$ であれば、m 単位で測っても cm 単位で測っても $\ln 2 \approx 0.693$ nat である。連続モデルの訓練に用いる交差エントロピー $-\mathbb{E}_P[\log q]$ は h と同じ定数だけシフトするが、その定数はモデルに依存しないため、勾配も最小点も変化しない。$h(p)$ が有限である限り、分解 Equation 37.5 は h で H を置き換えた形でそのまま成り立つため、密度に対する最尤法は母集団レベルで $D_{\mathrm{KL}}(p \| q_\theta)$ を最小化する。

また、微分エントロピーは単に細かいビニングのもとでの離散エントロピーの極限ではないことにも注意したい。実数直線を幅 Δ のセルに分割し、X_Δ を X がどのセルに含まれるかを記録する変数とする。点 x_i（セル i の内部の点）に対して $P(X_\Delta = i) \approx p(x_i)\Delta$ となり、次の極限が成り立つ。

$$H(X_\Delta) + \log \Delta \to h(X) \quad \text{as } \Delta \to 0$$

これは、リーマン和が積分に収束するほど p が十分に正則であるとき（たとえば p が連続であり、p と $p\log p$ の両方がリーマン可積分であるとき）にいつでも成り立つ。離散エントロピー $H(X_\Delta)$ 自体は $\log(1/\Delta)$ のように発散する。グリッドを細かくするほど、常により多くの「はい・いいえ」の質問が必要になるからである。そして $h(X)$ は、その増大分を差し引いた後に残る有限のオフセットにあたる。区間 $[0,1]$ 上の一様密度に対して $\Delta = \tfrac{1}{8}$ とすると、8 つのセルはすべて等確率となるため $H_2(X_\Delta) = 3$ ビット、$\log_2 \Delta = -3$ であり、その和は $0 = h$ となる。演習問題 6 ではこの極限を証明し、ガウス分布上で数値的に検証する。

37.1.4.2 ガウス分布間の KL ダイバージェンス

回帰モデルが実数値とその不確実性をあわせて予測するとき、最もよく出力されるのは平均と分散からなるガウス分布である。そのようなモデルをデータと比較したり、2 つのガウスモデルどうしを比較したりするには、2 つのガウス分布間の KL ダイバージェンスが必要になるが、これには簡潔な閉形式がある。$P = \mathcal{N}(\mu_1, \sigma_1^2)$ と $Q = \mathcal{N}(\mu_2, \sigma_2^2)$ に対して、次が成り立つ。

$$D_{\mathrm{KL}}\big(\mathcal{N}(\mu_1, \sigma_1^2) \,\big\|\, \mathcal{N}(\mu_2, \sigma_2^2)\big) = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}. \tag{37.11}$$

この公式は、標準偏差の対数比と、P の分散および平均の二乗誤差（いずれもモデルの分散 σ_2^2 との相対値として測られる）を組み合わせたものである。ギブスの不等式が保証するとおり、この値は非負であり、$\mu_1 = \mu_2$ かつ $\sigma_1 = \sigma_2$ のときに限り 0 になる。また、σ_1 と σ_2 が異なる形で現れていることから一目でわかるとおり非対称である。演習問題 7 では定義からこの式を直接導出する。

最も単純な場合として、分散が等しく平均が 1 だけシフトした $P = \mathcal{N}(0,1)$ と $Q = \mathcal{N}(1,1)$ から始めよう。このとき公式はどちらの方向でも $(\mu_1 - \mu_2)^2/(2\sigma^2) = \frac{1}{2}$ nat に帰着する。次に、代わりに分散が異なる場合として $P = \mathcal{N}(0,1)$ と $Q = \mathcal{N}(0,4)$（したがって $\sigma_2 = 2$）をとろう。式に代入すると $D_{\mathrm{KL}}(P\|Q) = \ln 2 + \frac{1}{8} - \frac{1}{2} \approx 0.318$ nat となるのに対し、逆方向は $D_{\mathrm{KL}}(Q\|P) = -\ln 2 + 2 - \frac{1}{2} \approx 0.807$ nat と、2.5 倍以上の大きさになる。

Figure 37.4 を見ると、それぞれのダイバージェンスがどこから生じているかがわかる。どちらの方向でも、被積分関数は 2 つの密度が交差する $|x| = (8\ln 2/3)^{1/2} \approx 1.36$ で符号を変える。パネル (a) では、幅の広いモデルが幅の狭いデータを説明しており、中心部（$|x| < 1.36$、寄与は $+0.424$）で過剰に驚く一方、裾野では真の分布よりわずかに驚きが小さい（寄与は -0.106）。パネル (b) では、幅の狭いモデルが幅の広いデータを説明しなければならず、中心部の寄与は -0.240 である一方、実際に生じる結果に幅の狭いモデルがごくわずかな確率しか割り当てない裾野が $+1.046$ という大きな寄与をもたらす。

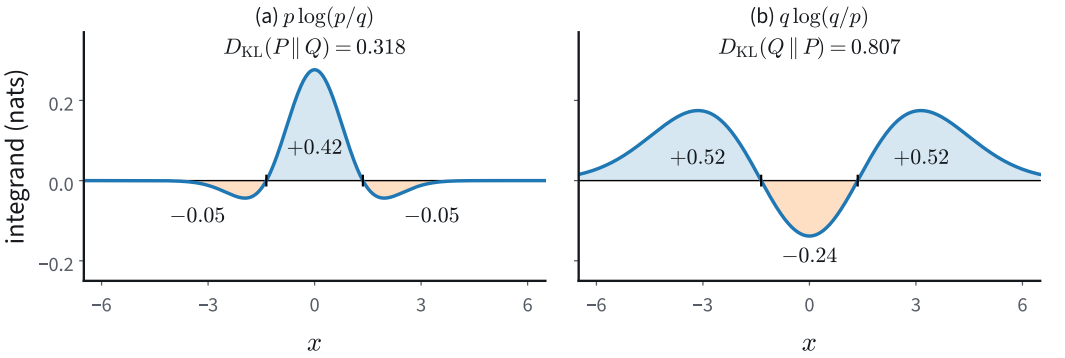


Figure 37.4: $\mathcal{N}(0,1)$ と $\mathcal{N}(0,4)$ の間の KL ダイバージェンスの内訳。(a) 幅の広いモデルで幅の狭い分布を説明する場合：正の中心部とわずかに負の裾野の和が 0.318 nat になる。(b) 幅の狭いモデルで幅の広い分布を説明する場合：幅の狭いモデルがほとんど確率を割り当てない裾野が 1.05 nat 寄与し、中心部の -0.24 と合わせて合計 0.807 になる。どちらの方向でも $|x| = 1.36$ で符号が変わる。

幅の狭いモデルが幅の広いデータから受ける驚きは、幅の広いモデルが幅の狭いデータから受ける驚きよりもはるかに大きい。一般に KL ダイバージェンスは、モデルがほぼ除外していた結果が実際に起こるたびに厳しいペナルティを課す一方、データが現れない領域にモデルが余分な確率を広げているだけの場合には穏やかな値にとどまる。この非対称性が、生成モデルを一方の方向で当てはめる場合と逆方向で当てはめる場合とで何を意味するかは、37.3.2 節の主題である。

また、この定義は閉形式が存在しない場合に KL ダイバージェンスを推定する一般的な方法も示唆している。すなわち、P からサンプルを引き、$\log p(x) - \log q(x)$ の平均をとればよい。後の節でサンプルに基づく推定値を用いるため、このガウス分布の対でどれほどの精度が出るか確認しておこう。次のセルでは、閉形式とモンテカルロ推定量およびその標準誤差を実装する。

```
def gaussian_kl(mu1, sigma1, mu2, sigma2):
    """Closed-form KL(N(mu1, sigma1^2) || N(mu2, sigma2^2)), in nats."""
    return (math.log(sigma2 / sigma1)
            + (sigma1 ** 2 + (mu1 - mu2) ** 2) / (2 * sigma2 ** 2)
            - 0.5)
```

```
def gaussian_log_pdf(x, mu, sigma):
    """Log-density of N(mu, sigma^2) at x."""
    return (-0.5 * math.log(2 * math.pi * sigma ** 2)
            - (x - mu) ** 2 / (2 * sigma ** 2))

def mc_kl(mu1, sigma1, mu2, sigma2, n=200000):
    """Monte Carlo estimate of the same KL from n samples x ~ P, with its
    standard error."""
    x = np.random.default_rng().normal(loc=mu1, scale=sigma1, size=n)
    r = gaussian_log_pdf(x, mu1, sigma1) - gaussian_log_pdf(x, mu2, sigma2)
    return float(r.mean()), float(r.std() / math.sqrt(n))
```

平均シフトの場合については、推定量の精度を事前に予測できる。対数比は $\log p(x) - \log q(x) = \frac{1}{2} - x$ であり、分散は 1（P のもとで）であるため、$n = 2 \cdot 10^5$ 個のサンプルの平均は標準誤差 $1/\sqrt{n} = 0.0022$ を持つ。

しかし分散が異なる対では、2 つの方向の推定のしやすさは同じではない。符号を除けばどちらも同じ関数 $\ln 2 - \frac{3}{8}x^2$ の平均をとるが、一方は幅の狭いガウス分布からサンプリングするのに対し、もう一方は x^2 が 4 倍大きく変動する幅の広いガウス分布からサンプリングする。3 つのダイバージェンスをすべて推定して確かめてみよう。

```
pairs = {'N(0,1) || N(1,1)': (0.0, 1.0, 1.0, 1.0),
         'N(0,1) || N(0,4)': (0.0, 1.0, 0.0, 2.0),
         'N(0,4) || N(0,1)': (0.0, 2.0, 0.0, 1.0)}
print(f'{"P || Q":18s} {"closed":>7s} {"sampled":>8s} {"std err":>8s}')
for name, args in pairs.items():
    estimate, std_err = mc_kl(*args)
    print(f'{name:18s} {gaussian_kl(*args):7.4f} {estimate:8.4f} {std_err:8.4f}')
```

```
P || Q             closed  sampled  std err
N(0,1) || N(1,1)    0.5000   0.5010   0.0022
N(0,1) || N(0,4)    0.3181   0.3169   0.0012
N(0,4) || N(0,1)    0.8069   0.8017   0.0047
```

標準誤差はそれぞれ 0.0022、0.0012、0.0047 となり、いずれの推定値も閉形式から標準誤差の数倍以内に収まっている。幅の広いガウス分布からサンプリングする方向は逆方向よりノイズが 4 倍大きく、同じ精度に達するには 16 倍のサンプル数を要することに注目しよう。Figure 37.4 のパネル (b) がその理由を説明している。このダイバージェンスの大半は、対数比が大きくサンプルがまばらな裾野から生じているためである。これと同じサンプリングの難しさには、37.7 節ではるかに大きなスケールで再び出会うことになる。

37.1.4.3 最大エントロピーとガウス分布

有限集合上では、一様分布が不確実性を最大化する。実数直線上では、どの密度が同じ役割を果たすだろうか。何の制約も置かなければ、そのような密度は存在しない。$\mathbb{R}$ 全体で一様な密度は存在せず、密度を引き伸ばせば微分エントロピーは際限なく大きくなるからである。そこで代わりに、ガウスモデルが出力する 2 つのモーメント、すなわち平均 μ と分散 σ^2 を固定してみよう。これら 2 つのモーメントを持つすべての密度のなかで、ガウス分布が最大の微分エントロピーを持つ。この厳密な意味において、ガウスノイズのモデルは最も仮定の少ないモデルである。平均と分散だけを定め、それ以外には何ひとつ仮定を置かない。

証明は、一様分布の上界を示した 1 行の議論とほぼ同じであり、ひねりが 1 つだけ加わる。離散

の場合には $-\log u(x) = \log k$ がすべての結果で一定であったため、どの分布 P も一様モデルに対して同じ交差エントロピー $\log k$ を持っていた。ここでは φ を平均 μ、分散 σ^2 のガウス密度とする。そのサプライザル $-\log \varphi(x) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{(x-\mu)^2}{2\sigma^2}$ はもはや定数ではないが、x の二次関数であり、二次関数の期待値は P に依存するものの、その依存は固定した 2 つのモーメントだけを通じたものになる。任意の密度 P（平均 μ、分散 σ^2）に対して、次が成り立つ。

$$\mathrm{CE}(P, \varphi) = \tfrac{1}{2}\log(2\pi\sigma^2) + \frac{\mathbb{E}_P\big[(x-\mu)^2\big]}{2\sigma^2} = \tfrac{1}{2}\log(2\pi\sigma^2) + \tfrac{1}{2} = \tfrac{1}{2}\log(2\pi e\sigma^2).$$

注目すべきことに、φ に対する交差エントロピーは、そのような P のどれがデータを生成したかに依存しない。$P = \varphi$ と置けば交差エントロピーはエントロピーに帰着するため、この定数はガウス分布自身の微分エントロピー、すなわち先ほど引用した閉形式にほかならない。同じモーメントを持つ他の任意の密度 P に対しては、分解 Equation 37.5（h で H を置き換えたもの）とギブスの不等式から次が得られる。

$$h(P) = \mathrm{CE}(P, \varphi) - D_{\mathrm{KL}}(P\|\varphi) \le \tfrac{1}{2}\log(2\pi e\sigma^2) = h(\mathcal{N}(\mu, \sigma^2)), \tag{37.12}$$

等号が成り立つのは $P = \mathcal{N}(\mu, \sigma^2)$ のとき、かつそのときに限る（$h(P) = -\infty$ の場合、不等式は自明に成り立つ）。この上界とのギャップ $D_{\mathrm{KL}}(P\|\varphi)$ にも直接的な解釈がある。すなわち、データがガウス分布でないときに最良のガウスモデルが被る過剰損失である（演習問題 8）。

一様分布とガウス分布は、単一の原理の 2 つの具体例である。候補となる密度のサプライザルが、制約を満たすすべての分布のもとで同じ定数に平均されるのであれば、その候補は制約を満たすすべての分布のなかでエントロピーを最大化する。36.4.2 節では任意の統計量の期待値に関する制約に対してこの結果を完全な一般性のもとで確立しており、そこでエントロピーを最大化する分布は指数分布族（exponential family）をなす。非負整数 $\{0, 1, 2, \ldots\}$ 上の幾何分布（geometric distribution）は 3 つ目の例である。そのサプライザルは結果に関してアフィンであるため、与えられた平均を持つ非負整数上のすべての分布のなかでエントロピーを最大化する。

一歩引いて、まだ未解決のまま残っている問いを確認しよう。先ほど、離散モデル Q' は実際に生じる結果に確率 0 を割り当てたために KL ダイバージェンスが無限大になった。Figure 37.4 の幅の狭いガウス分布もそれに近く、ほぼ除外していた裾野でダイバージェンスの大半を積み上げていた。少数の乱数を画像へと写す生成モデルは、ピクセル空間内の薄い曲面上にすべての確率質量を置く。もし実データがその曲面上にちょうど乗っていなければ、曲面がデータのすぐ隣を通っていようとはるか遠くに離れていようと、KL ダイバージェンスは無限大になる。37.3 節では密度比 p/q から構築されるあらゆるダイバージェンスがこの盲点を持つことを示し、37.4 節ではそのような状況でも遠近を識別できる比較尺度を展開する。2 つ目の問いはさらに身近なところにある。先ほど「はい・いいえ」の質問の回数を数えてそれを 2 元符号語として読み替えたが、それはサプライザルが整数のビット数になる 2 進情報源に対してだけであった。37.2 節では任意の情報源に対する符号を構成し、実際の圧縮器がどこまでエントロピーに迫れるかを問う。

37.1.5 まとめ

交差エントロピー損失は、データに対するモデルの平均サプライザルを測る。この損失は、いかなるモデルも下げられないデータのエントロピーと、データからモデルへの KL ダイバージェンスの和に分解される。ギブスの不等式により、後者は決して負にならず、真の分布に対してのみ 0 になる。したがって、いかなる損失の値も、その削減不能な下限である「入力が与えられたもとでのラベルの条件付きエントロピー」に照らして判断しなければならない。未知のデータ上では、真の条件付き確率だけがこの下限に到達できる。この下限は、入力がラベルを一意に定める場合には 0 となり、割合 ρ の対称なラベルノイズのもとでは $H_b(\rho) + \rho\log(k-1)$ となる。

ビット単位では、有限情報源のエントロピーは結果を特定するのに必要な「はい・いいえ」の質問の平均回数（1 問以内の誤差を除く）に等しく、高々 $\log_2 k$（k 個の結果に対して）である。ターゲットをソフト化すると最適解が移動する。ラベル平滑化は正の損失下限を伴う有限の最適ロジットギャップを生み出し、知識蒸留はソフト化された教師に一致させつつ、係数 τ^2 によって温度が変化しても勾配のスケールを一定に保つ。連続密度に対しては、微分エントロピーは負になり得るうえ測定単位によってシフトするが、KL ダイバージェンスと交差エントロピーの最小点はいずれも不変に保たれる。

37.1.6 演習問題

1. **自己情報量の一意性。**●自己情報量がスケールを除いて一意であることを示せ。すなわち、狭義減少関数 $I(p)$（区間 $(0,1]$ 上）が $I(1)=0$、および $I(p_1 p_2) = I(p_1) + I(p_2)$ をすべての $p_1, p_2 \in (0,1]$ に対して満たすならば、$I(p) = -c \log p$ がある定数 $c > 0$ に対して成り立つことを示せ。関数 $g(t) = I(e^{-t})$（$t \geq 0$）を用いて議論せよ。

2. [code] **ラベルノイズの下限。** k クラス上のラベルが、入力とは独立に確率 $1-\rho$ で正しく、そうでなければ残りの $k-1$ クラス上で一様に振られるとする。

   1. クリーンなラベルが入力の関数であると仮定する。任意の分類器の母集団交差エントロピーが少なくとも $H_b(\rho) + \rho \log(k-1)$ であること、最適な予測はノイズを含む条件付き分布そのものであること、そして最適な正解率は $1-\rho$ になる（$1-\rho > \rho/(k-1)$ のとき）ことを示せ。
   2. $k = 10$ および $\rho = 0.05, 0.1, 0.2$ に対してこの下限を計算せよ。
   3. すべての分割にこのようなノイズが 10% 含まれることがわかっている 10 クラスのデータセットで、同僚が 0.40 nat の検証損失を報告した。考えられる説明を列挙し、そのうちどれがこの下限によって棄却され、どれが棄却されないかを述べよ。
   4. 3.3 節のソフトマックス回帰を、訓練ラベルとテストラベルの 10% をこの方法で反転させた Fashion-MNIST 上で訓練せよ。テスト損失を 0.545 と、テスト正解率を 90% と比較せよ。ギャップは大きいはずである。Fashion-MNIST は (a) のどの仮定に反しているか、線形モデル固有の誤差は何を上乗せしているか、そして実際に適用される下限はどのように推定できるか。

3. **その他のスコアリングルール。** スコアリングルール $S(\mathbf{q}, y)$ は、期待ペナルティ $\mathbb{E}_{y \sim \mathbf{p}}[S(\mathbf{q}, y)]$ が $\mathbf{q} = \mathbf{p}$ で一意に最小化されるとき狭義固有である。対数スコアでは、この性質はギブスの不等式そのものである。

   1. $S(\mathbf{q}, y) = \sum_j (q_j - \mathbf{1}(j = y))^2$ で定まるブライアスコアの期待値が、正直な報告に対するブライアスコアの期待値を $\|\mathbf{q} - \mathbf{p}\|^2$ だけ上回り、したがってブライアスコアが狭義固有であることを示せ (Gneiting and Raftery 2007)。
   2. 線形スコア $S(\mathbf{q}, y) = -q_y$ は、観測されたクラスに割り当てられた確率を高く評価する。どの報告がその期待ペナルティを最小化するか。線形スコアは固有か。ラベルがクラス 1 となる確率が 0.6、クラス 2 となる確率が 0.4 である入力に対して、線形スコアで訓練された分類器は何を報告するか。

4. **曖昧な入力に対するラベル平滑化。**●ラベル平滑化はターゲット Equation 37.8 を用いて訓練する。

   1. 訓練分布において、ある入力 $\mathbf{x}$ がラベル y を確率 $p(y \mid \mathbf{x})$ で持つとする。この入力に対する平滑化損失の期待値が $\mathbf{q}^*(\cdot \mid \mathbf{x}) = (1-\epsilon)\, p(\cdot \mid \mathbf{x}) + \frac{\epsilon}{k}\mathbf{1}$ によって最小化されることを示せ。$p(\cdot \mid \mathbf{x})$ は $\mathbf{q}^*$ からどのように復元できるか。この最小点が 1 つのクラスに割り当て得る最大の確率はいくらか。

2. ロジット $\mathbf{z}$ と $\mathbf{q} = \mathrm{softmax}(\mathbf{z})$ を持つ 1 つの例に対し、$\mathrm{CE}(\mathbf{p}^{\epsilon}, \mathbf{q})$ の z_j に関する勾配は $q_j - p_j^{\epsilon}$ である。$\epsilon > 0$ のとき、すべての停留点において誤クラスのロジットがすべて等しくなることを示せ。
3. ワンホットターゲット（$\epsilon = 0$）では、誤クラスのロジットに関する勾配は q_j であり、損失は停留点を持たない。それでも 1 つの例のロジットに対する勾配降下法が 2 つの誤クラスロジット間のギャップを縮小させることを示し、この縮小がなぜ $q_y \to 1$ につれて停滞するのかを説明せよ。ニューラルネットワークでは、すべての例のロジットが共有された特徴から計算される。したがって $\epsilon = 0$ での訓練後にも生き残り得る誤クラスロジット間の差はどのようなものか。また、$\epsilon > 0$ で訓練するとなぜそれらが消えるのか。
4. Müller et al. (2019) は、ラベル平滑化で訓練した教師を用いると、蒸留損失 Equation 37.9 のもとでより性能の低い生徒が得られることを観察している。(b) と (c) の結果を用いてこの観察を説明せよ。

5. [code] **高温における蒸留。**●Hinton et al. (2015) はその 2.1 節において、高温では蒸留がゼロ平均ロジットの二乗誤差による一致に帰着すると述べている。
   1. $q_j = p_j(1 + u_j)$（したがって $\sum_j p_j u_j = 0$）と置き、$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = -\sum_j p_j \log(1 + u_j)$ を二次まで展開して次を導け。

$$D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \frac{1}{2}\sum_j \frac{(q_j - p_j)^2}{p_j} + O(\|\mathbf{q} - \mathbf{p}\|^3).$$

   この展開を、大きな τ に対する $\mathrm{softmax}(\mathbf{z}/\tau)$ の近似（37.1.3.4 節）と組み合わせ、$\tau \to \infty$ のとき次が成り立つことを示せ。

$$\tau^2 D_{\mathrm{KL}}(\mathbf{p}^{(\tau)} \,\|\, \mathbf{q}^{(\tau)}) \to \frac{1}{2k}\sum_{j=1}^{k}\Big((z_j^{\mathrm{stu}} - \bar{z}^{\mathrm{stu}}) - (z_j^{\mathrm{tea}} - \bar{z}^{\mathrm{tea}})\Big)^2.$$

   2. $\tau^2 D_{\mathrm{KL}}(\mathbf{p}^{(\tau)}\|\mathbf{q}^{(\tau)})$ を $\tau = 1, 2, 5, 10, 100$ に対して計算し（ロジットには $\texttt{z_tea} = (5, 2, -1)$ と $\texttt{z_stu} = (3, 3, 0)$ を 37.1.3.4 節と同様に用いる）、極限値と比較せよ。
   3. 低温では、教師のどのロジットが損失にほとんど影響しなくなるか。Hinton et al. (2015) は、これが利点にも欠点にもなり得ると論じている。教師自身の交差エントロピー訓練がこれらのロジットに何をもたらすかに基づき、双方の立場からそれぞれ 1 つずつ理由を挙げよ。

6. [code] **量子化エントロピーと微分エントロピーの比較。** X が密度 p を持ち、X_Δ がセル $[i\Delta, (i+1)\Delta)$ のどれに X が含まれるかを記録するとする。
   1. $H(X_\Delta) + \log\Delta \to h(X)$ が $\Delta \to 0$ で成り立つことを、p が連続であり p と $p\log p$ がリーマン可積分である場合について示せ。積分の平均値の定理により、$P(X_\Delta = i) = p(x_i)\Delta$（$x_i$ はセル内の点）と表して議論せよ。
   2. 区間 $[0, 1]$ 上の一様密度と $\Delta = \frac{1}{8}$ に対して手計算でこの極限を確認せよ。1 を割り切らないセル幅（たとえば $\Delta = 0.3$）ではどうなるか。
   3. $H(X_\Delta) + \log\Delta$ を $X \sim \mathcal{N}(0, 1)$ および $\Delta = 0.5, 0.1, 0.01$ に対して数値的に計算し（セルの確率にはガウス分布の分布関数を用いよ）、$h = \frac{1}{2}\log(2\pi e) \approx 1.4189$ nat と比較せよ（$\Delta = 0.5$ では 1.4292 が得られるはずである）。

4. なぜ $H(X_\Delta)$ 自体は $\Delta \to 0$ で発散するのか。回帰目標を幅 Δ のセルにビニングしてソフトマックス分類器で予測するとき、その損失は最良でもどの量に近づくか。また、連続モデルの負の対数密度と比較する前に何を足さなければならないか。

7. **ガウス分布の KL ダイバージェンス。** $P = \mathcal{N}(\mu_1, \sigma_1^2)$ とする。
   1. 閉形式 Equation 37.11 を、$\mathbb{E}_{x\sim P}[\log p(x) - \log q(x)]$ を書き下し、$\mathbb{E}_{x\sim P}[x] = \mu_1$ および $\mathbb{E}_{x\sim P}[(x-\mu_1)^2] = \sigma_1^2$ を用いることで導出せよ。
   2. Q が平均 $\mu_1 + \delta$、標準偏差 $\sigma_1 e^{\eta}$ を持つとする。$D_{\mathrm{KL}}(P\|Q)$ と $D_{\mathrm{KL}}(Q\|P)$ の両方を δ と η に関して二次まで展開せよ。2 つの方向が初めて異なるのは何次の項からか。
   3. $P = \mathcal{N}(0,1)$ と $Q = \mathcal{N}(0,4)$ に対してこの二次近似を計算し、2 つの方向の厳密値 0.318 nat および 0.807 nat と比較せよ。

8. **ガウスモデルの過剰損失。** 密度 P が平均 μ、分散 σ^2、および有限の微分エントロピーを持つターゲットに対して、回帰モデルがガウス分布 $\mathcal{N}(m, s^2)$ を出力するとする。
   1. $\mathrm{CE}(P, \mathcal{N}(m,s^2))$ が $m = \mu$ かつ $s^2 = \sigma^2$ で最小になること、そしてこの最良のガウスモデルが下限 $h(P)$ を $D_{\mathrm{KL}}(P\|\varphi) = h(\varphi) - h(P)$ だけ上回ることを示せ（ここで φ は $\mathcal{N}(\mu,\sigma^2)$ の密度である）。
   2. 区間 $[0, a]$ 上の一様密度（分散 $a^2/12$）、およびラプラス密度 $\frac{1}{2b}e^{-|x|/b}$（分散 $2b^2$、微分エントロピー $1 + \log(2b)$）に対してこの過剰損失を計算せよ（それぞれ約 0.176 nat と約 0.072 nat が得られるはずである）。
   3. いずれの答えもスケール a や b に依存しない。計算を行うことなく、37.1.4.1 節で述べた単位変換の性質からその理由を説明せよ。
   4. 与えられた分散に対してガウス分布は最大のエントロピーを持つため、それより高い下限を持つ密度は存在しない。では、過剰損失にも最大値は存在するか。2 点 $\mu \pm \sigma$ の近くに集中する密度を考えよ。この答えは、よく離れた 2 つのモードを持つターゲットに対するガウスモデルについて何を物語っているか。

37.2 符号、圧縮、パープレキシティ

37.1 節では、エントロピーと交差エントロピーを平均符号語長として解釈し、記号あたり x ビットの損失を達成するモデルであれば、テキストを記号あたり約 x ビットに圧縮できるはずだと示唆した。しかし、その主張を確かめたのはサプライザルがたまたま整数ビットになる都合のよい 1 つの情報源に対してだけであり、それ以外の分布に対する符号はまだ構成していなかった。そのような符号を実際に構築しない限り、対数損失（log-loss）とファイルサイズの結びつきは比喩の域を出ない。本節では、2 つの問いを立てることでこのギャップを埋める。モデルの対数損失に長さが一致する符号を構成できるのか、そして実用的な圧縮器はその限界にどこまで迫れるのか、という問いである。

日常的に使われる圧縮器にも、すでにその答えのヒントがある。8.2 節で扱った小説『The Time Machine』を、27 記号のアルファベット（英小文字 26 文字とスペース）からなる 173,428 文字に整形したものを考えよう。繰り返し現れる文字列を後方参照に置き換えてから各記号を個別に符号化する汎用圧縮器 `zlib` は、このテキストを文字あたり 2.84 ビットで保存する。これに対し、直前の 3 文字から各文字を予測し、読み進めながら出現回数を更新していく小さなカウントモデルは、同じテキストを文字あたり 2.55 ビットに圧縮する。予測が優れているほどファイルは短くなるのであり、本節ではその理由を解き明かす。符号によって対数損失がファイルサイズへと変換されれば、本章の冒頭で触れた文字あたり 1.2 nat といった損失は具体的な保存レートとなり、2 つのモデルの比較はそれらが生成するファイルの長さを比べるのと同じくらい直接的なものになる。

この結果に向けて 4 つのステップで議論を進める。第 1 に、記号間に区切り文字のない 2 進スト

リームを受信者が解釈しなければならないとき、符号語をどこまで短くできるかを問う。第 2 に、すべての符号語を整数ビットに制限することによるペナルティを測り、そのペナルティを記号ごとに支払わずに済むストリーム符号を構築する。第 3 に、各文字が過去に依存する実際のテキストへと目を向け、ファイルを理論上どこまで短くできるか、そしてモデルを学習するコストが全体の長さにどう組み込まれるかを見る。最後に、これらの符号長を、言語モデルがホールドアウトされたテキストに対して報告する評価指標であるパープレキシティと結びつける。本節を通じて、符号語が 2 進文字列であることから nat ではなくビットを単位として用いる。なお、x ビットは $x \ln 2$ nat に等しい。

本書の標準的なインポートに加え、テキストの前処理用に `re` を、ベースラインの圧縮器として `zlib` を用いる。

```
import re
import zlib
```

37.2.1 記号符号

最も単純な通信問題から始めよう。送信者は、分布 P（k 通りの結果上の分布）から独立に抽出された記号のストリームを観測し、記号間に区切り文字を入れない単一の 2 進文字列として送信しなければならない。一方、受信者は元の系列を一意に復元しなければならない。

37.2.1.1 接頭符号とクラフトの不等式

記号符号（symbol code）では、各結果 i に 2 進文字列である**符号語**（codeword）を長さ ℓ_i で割り当てる。区切り文字なしで、受信者はどうやって 1 つの符号語の終わりと次の符号語の始まりを見分ければよいのだろうか。最も単純な保証は、どの符号語も他のいかなる符号語の接頭辞にもならないようにすることである。このような符号は**プレフィックスフリー**、あるいは単に**接頭符号**（prefix code）と呼ばれる。電話番号も同じ規則に従っている。ダイヤルした数字の並びが有効な番号に一致した時点で、同じ並びから始まるより長い番号は存在しないため、交換機は番号の入力が完了したことを即座に判別できる。4 つの記号に対して符号 $0, 10, 110, 111$ はプレフィックスフリーであり、受信ストリーム `110010111` は `110|0|10|111` というただ 1 通りの方法でのみ区切ることができる。

接頭符号を二分木として可視化すると（Figure 37.5）、符号語長に対する制約が直ちに明らかになる。長さ ℓ の符号語は深さ ℓ の葉で終わり、その下にあるすべての子孫を除外するため、木全体の割合 $2^{-\ell}$ を占有する。異なる符号語どうしは重なり合えないため、それぞれが占有する割合は合計予算 1 の単一の木の中に収まらなければならず、それゆえ短い符号語は高くつく。図の長さ $(1, 2, 3, 3)$ は $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} = 1$ を費やし、木全体を使い切っている。対照的に、長さ 1 の符号語を 4 つ用意するには予算が 2 必要になるが、長さ 1 の 2 進文字列は 2 つしか存在しないため不可能である。長さ $(1, 2, 4)$ は $\frac{1}{2} + \frac{1}{4} + \frac{1}{16} = 0.8125$ を費やすだけで、たとえば $0, 10, 1100$ のように余力を残して無理なく収まる。

同じ木は、37.1 節の「はい・いいえ」質問ゲームの戦略も表している。各内部ノードが二択の質問を投げかけ、各符号語はある記号に至るまでの回答の系列を記録する。「はい」を 0 とすると、符号 $0, 10, 110, 111$ は順に「1 番目の記号か」「2 番目の記号か」「3 番目の記号か」と尋ねることに対応する。前節では「はい」に 1 を用いたため、その鏡像である $1, 01, 001, 000$ が得られていた。

命題（クラフトの不等式） 符号語長 $\ell_1, \dots, \ell_k$ を持つプレフィックスフリーな 2 進符号は、次の不等式が成り立つとき、かつそのときに限り存在する。

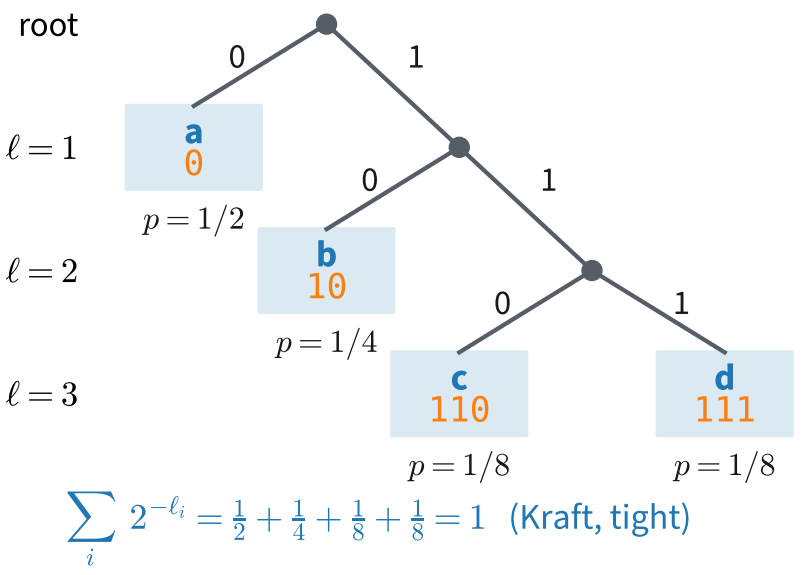


Figure 37.5: 二分木として表したプレフィックスフリー符号 $0, 10, 110, 111$。各符号語は根から葉への経路（左の辺が 0、右の辺が 1）であり、どの符号語も他の符号語を通過しない。深さ ℓ の葉は木全体の割合 $2^{-\ell}$ を占有し、ここでは 4 つの葉が $2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1$ と予算全体を使い切っている。記号 a, b, c, d とその確率は、前節で扱った 2 進情報源のものである。

$$\sum_{i=1}^{k} 2^{-\ell_i} \le 1. \tag{37.13}$$

証明 各 2 進符号語 c（長さ ℓ）に、半開区間 $[0.c,\ 0.c + 2^{-\ell})$、すなわち $[0, 1)$ 内の実数のうち 2 進展開が c から始まるもの全体からなる区間を対応させる。たとえば、符号語 10 は $[0.10_2, 0.11_2) = [\tfrac{1}{2}, \tfrac{3}{4})$ に対応し、その幅は 2^{-2} である。Figure 37.5 では、この区間は b の葉と、その下で取り除かれた部分木全体を表している。このような任意の 2 つの **2 進**（dyadic）区間は、一方が他方に含まれる（入れ子になる）か、互いに素であるかのいずれかである。もし c が c' の接頭辞であれば、c' から始まるすべての数は c からも始まるため区間は入れ子になり、そうでなければ c と c' は短い方の長さ以内のどこかの位置で異なるため、両方から始まる数は存在せず区間は互いに素になる。したがって、プレフィックスフリー符号は $[0, 1)$ 内の互いに素な区間に対応し、その合計幅 $\sum_i 2^{-\ell_i}$ は高々 1 である。

逆に、符号語長が Equation 37.13 を満たすとする。これらを $\ell_1 \le \cdots \le \ell_k$ と非減少順に並べ、幅 $2^{-\ell_i}$ の区間を 0 から順に隙間なく並べていく。i 番目の区間は $s_i = \sum_{m<i} 2^{-\ell_m}$ から始まり、$s_i + 2^{-\ell_i} \le 1$ で終わる。先行するすべての長さが $\ell_m \le \ell_i$ を満たすため、和の各項 $2^{-\ell_m}$（s_i を構成する項）は $2^{-\ell_i}$ の整数倍であり、したがって s_i 自身も $2^{-\ell_i}$ の整数倍となる。ゆえに i 番目の区間は、先頭 ℓ_i 桁の 2 進数字を s_i から取り出した符号語の 2 進区間である。これらの区間は互いに素であるため、証明の前半より、どの符号語も他の符号語の接頭辞にはならないことが保証される。■

注目すべきことに、クラフトの予算はプレフィックスフリー符号に固執したために生じる人工的な制約ではない。たとえ一部の符号語が他の符号語の接頭辞になっていても、符号語のあらゆる

連結がただ 1 通りにしか区切れないとき、その符号は**一意復号可能**（uniquely decodable）であるという。McMillan は、すべての一意復号可能符号がやはり Equation 37.13 を満たさなければならないことを証明した (McMillan 1956)（演習問題 1）。したがって、一意復号可能符号で達成可能な符号語長の組はすべて接頭符号ですでに達成可能であり、考察の対象を接頭符号に限定しても何も失われない。

37.2.1.2 シャノン長とエントロピー限界

予算がわかったので、次はどの符号語長が期待符号長 $\mathbb{E}[\ell] = \sum_i p_i \ell_i$ を最小化するかを問うことができる。頻出する記号には明らかに短い符号語を割り当てるべきだが、長さ 1 の符号語を 1 つ割り当てるだけでクラフトの予算全体の半分を消費してしまう。

理想的なバランスを見つけるため、一時的に整数制約を緩和し、長さ ℓ_i を実数として扱ってみよう。どの符号語を短くしても $\mathbb{E}[\ell]$ は狭義に減少するため、予算を使い残すことが有利になることは決してなく、最適解は $\sum_i 2^{-\ell_i} = 1$ を満たさなければならない。等式制約付き最適化に対するラグランジュの未定乗数法（34.2.4 節）により、最小点では目的関数の勾配が制約関数の勾配に比例しなければならない。$\mathbb{E}[\ell]$ の ℓ_i に関する導関数は p_i であり、一方 $\sum_i 2^{-\ell_i}$ の導関数は $-2^{-\ell_i}\ln 2$ である。したがって、$2^{-\ell_i} = c\,p_i$ が単一の定数 c を用いてすべての i について成り立つ。両辺を i にわたって足し合わせると $c = 1$ が定まり、最適な実数符号語長とそれによる期待符号長が次のように得られる。

$$\ell_i = \log_2 \frac{1}{p_i}, \qquad \mathbb{E}[\ell] = \sum_i p_i \log_2 \frac{1}{p_i} = H_2(P).$$

各記号の理想的な長さは、単にその**サプライザル** $\log_2(1/p_i)$、すなわち 37.1 節の自己情報量をビット単位で測ったものにほかならず、その結果得られる期待符号長は 2 進エントロピー $H_2(P)$ である。もちろん、実際の符号語は整数の長さを持たなければならない。2 進情報源 $P = (\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{8}, \tfrac{1}{8})$ の場合、サプライザル $1, 2, 3, 3$ はすでに整数であるため、Figure 37.5 の符号は無駄なく $\mathbb{E}[\ell] = 1.75$ ビット $= H_2(P)$ を達成する。本章で継続して用いる 2 つの分布の組のうち 1 つ目である $P = (0.6, 0.3, 0.1)$ の場合、サプライザルは 0.74、1.74、3.32 ビットとなる。それぞれを次の整数へ切り上げると**シャノン長**（Shannon lengths）$1, 2, 4$ が得られ、これは先ほど余力を残して二分木に収まった長さの組と同じである。そのクラフト和は 0.8125 であり、期待符号長は $\mathbb{E}[\ell] = 0.6 + 0.6 + 0.4 = 1.6$ ビットとなる（これに対し $H_2(P) = 1.295$ である）。

整数長をどれほど慎重に選んでも、エントロピーを下回る接頭符号は存在せず、一方でシャノン長がエントロピーを 1 ビット以上超えることも決してない。これら両方の上下界は、37.1 節で示した 1 つの恒等式、すなわち交差エントロピーの分解から導かれる。これをビット単位で改めて記すと次のようになる（添字の 2 と上付き添字の (2) は底 2 の対数を表す）。モデル Q が P から抽出されたデータに対して持つ期待対数損失は、P のエントロピーと、P から Q へのカルバック・ライブラー・ダイバージェンスとに分解される。

$$\mathrm{CE}_2(P, Q) = -\sum_i p_i \log_2 q_i = H_2(P) + D_{\mathrm{KL}}^{(2)}(P \| Q), \qquad D_{\mathrm{KL}}^{(2)}(P\|Q) \geq 0. \tag{37.14}$$

ここで KL ダイバージェンスの非負性はギブスの不等式（Gibbs' inequality）そのものである。いかなるモデルも情報源の真のエントロピーより低い期待対数損失を達成することはできず、等号は $Q = P$ の場合に限り成り立つ。継続して用いている分布の組 $P = (0.6, 0.3, 0.1)$ と $Q = (0.2, 0.5, 0.3)$ では、これら 3 つの量は（小数最終桁の丸めの範囲で）$1.867 = 1.295 + 0.571$ ビットとなる。

命題（エントロピーによる符号長の上下界） P を、$p_i > 0$ がすべての i について成り立つ分布とする（確率 0 の記号には符号語は不要である）。すべてのプレフィックスフリーな 2 進符号は $\mathbb{E}[\ell] \geq H_2(P)$ を満たし、等号は $p_i = 2^{-\ell_i}$ がすべての i について成り立つとき、かつそのときに限り成立する。逆に、シャノン符号（Shannon code）（長さ $\ell_i = \lceil \log_2(1/p_i) \rceil$ を持つ符号）はプレフィックスフリーであり、$\mathbb{E}[\ell] < H_2(P) + 1$ を満たす。

証明 下界については、符号をそれが暗に定める確率分布と比較する。$Z = \sum_i 2^{-\ell_i}$ とおくと、クラフトの不等式より $Z \leq 1$ を満たすので、正規化された分布 $q_i = 2^{-\ell_i}/Z$ を定義する。ℓ_i について解くと $\ell_i = -\log_2 q_i - \log_2 Z$ となり、P のもとで期待値をとると次式が得られる。

$$\mathbb{E}[\ell] = \sum_i p_i \ell_i = -\sum_i p_i \log_2 q_i - \log_2 Z = H_2(P) + D_{\mathrm{KL}}^{(2)}(P \| Q) - \log_2 Z \geq H_2(P). \tag{37.15}$$

2 つの過剰項はいずれも非負である。すなわち、ギブスの不等式より $D_{\mathrm{KL}}^{(2)}(P\|Q) \geq 0$ であり、また $-\log_2 Z \geq 0$ も $Z \leq 1$ から成り立つ。等号成立には両方の項が 0 になる必要があり、それは $Q = P$ かつ $Z = 1$、すなわち $p_i = 2^{-\ell_i}$ がすべての i について成り立つことを意味する。上界については、シャノン長が $2^{-\ell_i} \leq p_i$ を満たすことから $\sum_i 2^{-\ell_i} \leq \sum_i p_i = 1$ となるため、クラフトの条件を満たし有効な接頭符号を定める。天井関数による切り上げが加える量は狭義に 1 ビット未満であるから $\ell_i < \log_2(1/p_i) + 1$ であり、P のもとで平均をとれば $\mathbb{E}[\ell] < H_2(P) + 1$ が得られる。$\blacksquare$

恒等式 Equation 37.15 は、それが確立する不等式以上に多くのことを教えてくれる。すなわち、接頭符号がビットを無駄にする原因は次の 2 通りしか存在しない。符号が暗に定める分布 Q が P と異なること、またはクラフトの予算の一部を未消費のまま残すこと（$Z < 1$）である。継続して用いている分布 P に対してシャノン符号は記号あたり $1.6 - 1.295 = 0.305$ ビットを無駄にするが、そのうち分布の不一致に由来するのはわずか 0.005 ビットであり、残りの 0.300 ビットは未消費の予算に由来する。

この命題の前半と後半を合わせると、いかなる記号符号の最適期待符号長も $H_2(P)$ ビットと $H_2(P) + 1$ ビットの間に挟まれる。これはまさに 37.1 節で予告した、「はい・いいえ」質問の期待回数に対する上下界である。エントロピーの下限が達成されるのは、すべての確率がちょうど $\frac{1}{2}$ のべき乗になっている場合に限られる。

より賢い記号符号を使えば、この 1 ビットのオーバーヘッドをどこまで取り戻せるだろうか。ハフマンのアルゴリズム（Huffman's algorithm）(Huffman 1952) は、最も確率の低い 2 つの記号を 1 つの合成記号へと繰り返し統合することで最適な接頭符号を構成する手法であり（演習問題 2）、`zlib` の内部で使われている記号符号器でもある。継続して用いている P に対して、ハフマンのアルゴリズムは長さ $1, 2, 2$ を選び、平均 1.4 ビット（エントロピーをわずか 0.1 ビット上回るだけ）を達成してシャノン符号の無駄の 3 分の 2 を解消する。しかし、最適な記号符号であっても避けられない壁に直面する。いかなる符号語も 1 ビットより短くはできないため、極端に偏った情報源ではこの 1 ビットの下限が節約分のほぼすべてを飲み込んでしまうのである。

`a` を確率 0.9 で、`b` を確率 0.1 で出力する情報源を考えよう。そのエントロピーは記号あたり 0.47 ビットであり、各 `a` が持つサプライザルはわずか $\log_2(1/0.9) = 0.15$ ビットにすぎない。それでも、いかなる記号符号もすべての `a` に少なくとも丸々 1 ビット、すなわちその情報量の 6 倍以上を費やさなければならない。もし `a` が確率 0.999 で生じるなら、エントロピーは記号あたり 0.011 ビットまで下がるにもかかわらず、どの記号符号も依然として 1 ビットを費やす。この「$+1$」のペナルティは、記号ごとに 1 回丸めを行うコストである。一度に n 記号のブロックをまとめて符号化すれば、このコストを償却できる。n 記号のブロックをより大きなアルファベット上の単一の記号として扱えば、切り上げはブロックごとに 1 回で済み、記号あたりのオーバーヘ

ッドは $1/n$ 未満に抑えられる。ハフマンのアルゴリズムで a と b の記号の**ペア**を符号化するだけでも、レートは元の記号あたり 1 ビットから直ちに 0.645 ビットへと下がる。

37.2.1.3 不一致符号と交差エントロピー

実際には、真のデータ分布 P が既知であることはまれであり、代わりにモデル Q から符号を構築する。Q に対するシャノン符号は長さ $\ell_i = \lceil \log_2(1/q_i) \rceil$ を割り当てるが、これには $q_i > 0$ が（P が出力し得るすべての記号について）成り立つ必要がある。この符号を P から実際に生成されたデータに用いるとき、その期待符号長は次式を満たす。

$$\mathrm{CE}_2(P, Q) \;\le\; \mathbb{E}_P[\ell] \;<\; \mathrm{CE}_2(P, Q) + 1, \tag{37.16}$$

これは $\log_2(1/q_i) \le \ell_i < \log_2(1/q_i) + 1$ がすべての記号 i について成り立ち、P のもとで平均をとっても両方の不等号が保たれるからである。したがって交差エントロピーとは、モデル Q のために設計された符号の期待符号長（ただし最大 1 ビットの丸め誤差を除く）にほかならず、Equation 37.14 により KL ダイバージェンスはモデル Q を真の分布 P の代わりに用いることで支払う正確なペナルティとなる。

これらの上下界を数値的に確かめてみよう。以下のセルでは、モデル q に対するシャノン符号を構成し、データ分布 p のもとでの期待符号長を計算して、クラフト和と交差エントロピーを並べて出力する。本節で扱った 2 つの情報源をそれぞれ一致符号と不一致符号のもとでテストする。すなわち、2 進情報源と一様モデルの組、および継続して用いている P と $Q = (0.2, 0.5, 0.3)$ の組である。

```
def shannon_lengths(q):
    return [math.ceil(math.log2(1 / q_i)) for q_i in q]

def code_report(name, p, q):
    """The Shannon code built from q, used on data from p (all in bits)."""
    ells = shannon_lengths(q)
    kraft = sum(2 ** -ell for ell in ells)
    avg = sum(p_i * ell for p_i, ell in zip(p, ells))
    ce2 = -sum(p_i * math.log2(q_i) for p_i, q_i in zip(p, q))
    print(f'{name:<16} lengths {ells}  Kraft {kraft:.4f}  '
          f'E[ell] {avg:.3f}  CE2 {ce2:.3f}')

dyadic, uniform = [1/2, 1/4, 1/8, 1/8], [1/4] * 4
p, q = [0.6, 0.3, 0.1], [0.2, 0.5, 0.3]
code_report('dyadic, own', dyadic, dyadic)
code_report('dyadic, uniform', dyadic, uniform)
code_report('P, own', p, p)
code_report('P, code from Q', p, q)
```

```
dyadic, own      lengths [1, 2, 3, 3]  Kraft 1.0000  E[ell] 1.750  CE2 1.750
dyadic, uniform  lengths [2, 2, 2, 2]  Kraft 1.0000  E[ell] 2.000  CE2 2.000
P, own           lengths [1, 2, 4]  Kraft 0.8125  E[ell] 1.600  CE2 1.295
P, code from Q   lengths [3, 1, 2]  Kraft 0.8750  E[ell] 2.300  CE2 1.867
```

分布が一致している 2 つの行は手計算の結果を再現している。ここでは CE2 がエントロピーに一致し、期待符号長はそれぞれ 1.75 ビットと 1.6 ビットである。不一致の行からは、誤ったモデルを使うコストが浮かび上がる。Q からシャノン符号を構成すると長さは $3, 1, 2$ となり、図らずも最長の符号語を P の最頻出記号に割り当ててしまう。P からのデータに対しては記号あたり平均

2.3 ビットとなり、区間 $1.867 \le 2.3 < 2.867$（Equation 37.16 による予測）のちょうど内側に収まる。一様符号は 2 進情報源のすべての記号に 2 ビットを割り当てる。丸めは一切生じないため下界が等号で成立し、$2.0 = 1.75 + 0.25$ ビットはエントロピーと 0.25 ビットの KL ダイバージェンスの和にきれいに分解される。最後に Kraft 列を見ると、各符号がどれだけの予算を未使用のまま残しているかがわかる。2 進情報源では未使用分は 0 であり、未使用分が 0.1875 となるのが P のシャノン符号、0.125 となるのが Q から構成した符号である。

1 つ微妙な点に注意しよう。不等式 Equation 37.16 は特に Q のシャノン符号に対して成り立つものであり、Q から構成し得るあらゆる符号に当てはまるわけではない。Q に対するハフマン符号は長さ $2, 1, 2$ を持ち、クラフトの予算をすべて使い切り、平均はわずか $1.2 + 0.3 + 0.2 = 1.7 < 1.867$ ビット（P からのデータに対して）にとどまる。ここに矛盾はない。この符号が暗に定める分布は $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ であって Q ではなく、その 2 進分布がたまたま P のより近くに位置しているのである。Equation 37.15 が示したとおり、符号の過剰な長さはそれが暗に定める分布と未消費の予算のみによって決まり、Q はそれが生み出す符号語長を通じてのみ影響する。

37.2.2 ストリーム符号と記述長

記号符号は、記号ごとに丸めのペナルティを支払う。次に、1 つのメッセージにつき 1 回だけ丸めを行う符号を構築して言語モデルに直接結びつけ、さらにモデル自体を記述するコストをどのように織り込むべきかを問う。

37.2.2.1 算術符号化

記号のブロックをまとめて符号化すれば記号ごとの丸め損失はなくなるが、すぐさま組み合わせ論の壁に突き当たる。100 個の 2 進記号からなるブロックの符号帳には 2^{100} 個もの項目が必要となり、到底保存しきれないからである。本当に欲しいのは、ルックアップテーブルを持たずにブロック符号の効率を実現する仕組み、すなわちメッセージ全体に対する単一の符号語をメッセージ自体から逐次的に計算する符号である。この種の符号は**ストリーム符号**（stream code）と呼ばれ、**算術符号化**（arithmetic coding）はその古典的な構成法である (Rissanen 1976; Witten et al. 1987)。符号語を表から引く代わりに、算術符号化はメッセージを $[0, 1)$ の部分区間として表現する。Figure 37.6 は、10 記号のメッセージ aaaaaaaaba（$q(\mathtt{a}) = 0.9$ の偏った情報源から生成されたもの）を真のモデルのもとで符号化する様子を示している。

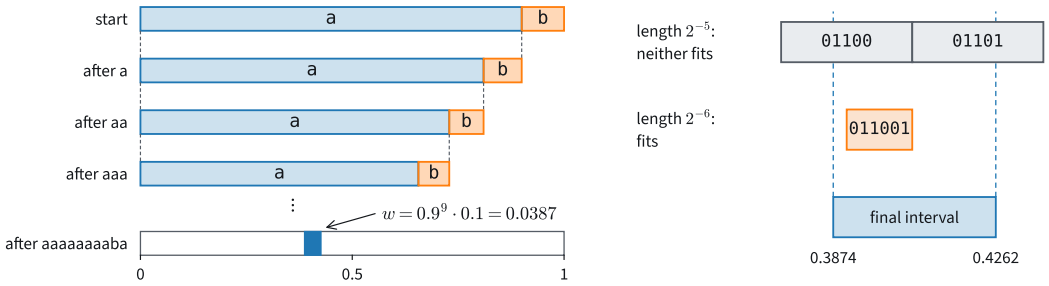


Figure 37.6: モデル $q(\mathtt{a}) = 0.9$ のもとでのメッセージ aaaaaaaaba の算術符号化。(a) 各記号は現在の区間をモデルがその記号に割り当てた部分区間へと狭めていき、10 記号後の残りの幅はメッセージの確率 $w = 0.9^9 \cdot 0.1 = 0.0387$ になる。(b) 最終区間は、その完全に内側に収まる 2 進区間によって符号化される。この区間と重なる長さ 2^{-5} の 2 進区間はいずれも完全には内側に収まらないのに対し、長さ 2^{-6} の 0.390625 から 0.40625 までの区間は完全に収まる。その 2 進アドレス 011001 が符号語であり、サプライザルが 4.69 ビットのメッセージを 6 ビットで表している。

エンコーダは現在の区間（初期値は $[0,1)$）を保持する。次の記号 x_i を符号化する際、エンコーダは現在の区間を、モデルの条件付き確率 $q(\cdot \mid x_{<i})$ に比例する幅で取り得る記号ごとのスライスに分割し、実際に生じた記号に対応するスライスだけを残す。各ステップで区間幅に観測記号の確率が掛け合わされるため、$x_1, \dots, x_n$ を処理し終えた後の最終区間の幅は次式で与えられる。

$$w = \prod_{i=1}^{n} q(x_i \mid x_{<i}),$$

これはまさに、モデルがメッセージ全体に割り当てる確率に一致する。長さ n の相異なるメッセージは互いに素な区間に入るため、最終区間の内部にある任意の 1 点がメッセージを一意に特定する。同じモデルを再現するデコーダは、送信された数値がどのスライスに含まれるかを調べ、対応する記号を出力し、自身の区間をエンコーダと歩調を合わせて狭めていく。デコーダはまた、事前に n を知っておくか専用のメッセージ終端記号を含めることで停止時期を知る必要があるが、ここではそのフレーミングコストは脇に置く。残る作業は最終区間内の 1 点を 2 進法で指定することだけであり、クラフトの不等式の証明で用いた 2 進区間がこれを明快に解決する。

幅 w の任意の区間は、長さ $2^{-\ell}$ の 2 進区間を必ず含むことが $\ell = \lceil \log_2(1/w) \rceil + 1$ と選ぶことで保証される。なぜだろうか。このように選べば $2^{-\ell} \le w/2$ が成り立つため、区間の左端以上である最初の $2^{-\ell}$ の倍数は必ず区間の左半分に入り、そこを始点とする長さ $2^{-\ell}$ の 2 進区間が右端をはみ出すことはあり得ないからである。Figure 37.6 の (b) は両方の状況を示している。長さ 2^{-5} の 2 進区間は最終区間の端からはみ出してしまうが、長さ 2^{-6} の区間は完全に内部に収まる。$\ell < \log_2(1/w) + 2$ であるから、得られる符号語長は次式を満たす。

$$\ell \;<\; \sum_{i=1}^{n} \log_2 \frac{1}{q(x_i \mid x_{<i})} + 2, \tag{37.17}$$

すなわち、（既知のメッセージ長 n に対して）メッセージはモデルのもとでの総サプライザルに高々 2 ビットを加えた長さで符号化される。いまや丸めは記号ごとに 1 回ではなく、メッセージ全体で 1 回だけ行われる。さらに、相異なる長さ n のメッセージに対する符号語は互いに素なメッセージ区間の内部にある 2 進区間を指定するため、クラフトの不等式の証明の前半により、メッセージ全体の上の接頭符号をなす。したがってエントロピー下界は、単一の記号に対してと同様にメッセージ全体に対しても成り立つ。どのようなモデルを選ぼうとも、期待符号長がメッセージのエントロピーを下回ることは決してない。

この区間縮小の手順を aaaaaaaaba に対して実行し、記号ごとのシャノン符号と長さを比較してみよう。以下のセルでは区間を追跡し、2 進符号語を取り出してその 2 進区間が最終区間の内部に収まることを検証し、両方のビット数を出力する。

```
msg = 'aaaaaaaaba'                  # nine a's, one b
q = {'a': 0.9, 'b': 0.1}            # an i.i.d. model of the source

lo, hi = 0.0, 1.0
for ch in msg:                      # keep the observed symbol's slice
    width = hi - lo
    if ch == 'a':                   # 'a' owns the left 90% of the interval
        hi = lo + width * q['a']
    else:                           # 'b' owns the right 10%
        lo = lo + width * q['a']
```

```
    ell = math.ceil(-math.log2(hi - lo)) + 1      # codeword length
    m = math.ceil(lo * 2 ** ell)                  # first multiple of 2^-ell >= lo
    assert lo <= m / 2 ** ell and (m + 1) / 2 ** ell <= hi
    surprisal = -math.log2(math.prod(q[ch] for ch in msg))
    shannon_bits = sum(math.ceil(math.log2(1 / q[ch])) for ch in msg)
    print(f'final interval [{lo:.4f}, {hi:.4f}), width {hi - lo:.4f}')
    print(f'surprisal {surprisal:.2f} bits, codeword {m:0{ell}b} ({ell} bits)')
    print(f'symbol-by-symbol Shannon code: {shannon_bits} bits')
```

```
final interval [0.3874, 0.4262), width 0.0387
surprisal 4.69 bits, codeword 011001 (6 bits)
symbol-by-symbol Shannon code: 13 bits
```

最終的な幅 0.0387 は予想どおり積 $0.9^9 \cdot 0.1$ に一致している。このメッセージのサプライザルは 4.69 ビットであり、算術符号化はこれを 6 ビット（`011001`）で保存するのに対し、記号ごとのシャノン符号は情報量の小さい 9 個の `a` のそれぞれに丸々 1 ビットを費やすため 13 ビットを要する。この文字列はたまたま情報源の平均的な構成（9 個の `a` と 1 個の `b`）を持っているため、そのサプライザルは $10\,H_2(0.9, 0.1) = 4.69$ ビットに等しい。`b` を 2 個含むメッセージであれば、より多くのビットが必要になる。

素朴な浮動小数点による区間縮小は、短いメッセージにしか機能しない。ステップごとに幅が倍率 q で縮むため、およそ 100 記号も進むと倍精度浮動小数点数の 53 ビットの仮数部では両端点が区別できなくなるからである（演習問題 3）。実用的な算術符号器は固定幅の整数演算を用いてこれを解決し、両端点の上位ビットが一致し次第そのビットを順次出力して、作業区間を常に最大の精度へと再スケーリングし続ける (Witten et al. 1987)。再スケーリングを行えばメッセージは任意の長さになり得て、2 ビットのオーバーヘッドも記号あたりでは消失する。すなわち、n 記号のテキストは、モデルのビット単位の平均対数損失の n 倍で保存される。

37.2.2.2 圧縮器としての言語モデル

算術符号化の議論では、記号が独立であることを一切仮定していなかった。エンコーダとデコーダが有限精度のもとで次の記号の確率を 1 ビットの違いもなく同一に計算できる限り、任意の条件付きモデル $q(\cdot \mid x_{<i})$ によって区間縮小を駆動できる。自己回帰言語モデルを算術符号器と組み合わせれば、それだけで可逆圧縮器になる。トークナイザ、モデル、符号器を共有していれば、いかなる文書もモデルのもとでの総サプライザルに高々 2 ビットを加えた長さへと圧縮される（実用上のファイル形式ではわずかなフレーミングのオーバーヘッドが加わり、またモデル自体を事前に固定する場合には送受信の双方があらかじめそのモデルを保持している必要がある）。

読み進めながらその場で学習する最も単純なモデルは、カウントモデルである。**適応的 m 次モデル**は、各文字をその直前の m 文字、すなわち**コンテキスト** s から次式を用いて予測する。

$$q(x \mid s) = \frac{c(s, x) + \tfrac{1}{2}}{c(s) + \tfrac{k}{2}},$$

ここで、$c(s, x)$ はこれまでに文字 x が s の直後に現れた回数、$c(s) = \sum_x c(s, x)$ はコンテキスト s の合計出現回数、k はアルファベットサイズである。すべてのカウントに $\tfrac{1}{2}$ を加えることで、どの文字も確率 0（無限ビットを要してしまう）にならないことが保証される。決定的に重要なのは、モデルが各文字を符号化した**後に**カウントを更新する点である。デコーダもすでに復元した文字に対して同じ更新を行うため、エンコーダとデコーダは完全に歩調を合わせて動作し、モデルパラメータを送信する必要が一切なくなる。

この圧縮器を『The Time Machine』で試してみよう。最初のセルではテキストをダウンロードして英小文字とスペースのみに整形し、適応的符号長を文字あたりビット数で計算する関数を定義する。

```
fname = d2l.download(d2l.DATA_URL + 'timemachine.txt',
                     sha1_hash='090b5e7e70c295757f55df93cb0a180b9691891a')
with open(fname) as f:
    text = re.sub('[^A-Za-z]+', ' ', f.read()).lower()   # 26 letters, space

def adaptive_bits(text, m, k):
    """Bits per character of the adaptive order-m model with half counts."""
    counts, totals, bits = {}, {}, 0.0
    for i, ch in enumerate(text):
        s = text[max(0, i - m):i]                # the context
        c = counts.setdefault(s, {})
        bits -= math.log2((c.get(ch, 0) + 0.5) / (totals.get(s, 0) + k / 2))
        c[ch] = c.get(ch, 0) + 1                 # learn after coding
        totals[s] = totals.get(s, 0) + 1
    return bits / len(text)
```

次に、長さ 0 から 4 までのコンテキストを評価し、最高圧縮レベルの `zlib` と比較する。長いコンテキストほどカウントが十分に集まれば優れた予測を行えるが、学習すべきカウントの数もはるかに多くなる。出力結果は、そのトレードオフがどこで逆転するかを示している。

```
k, n = len(set(text)), len(text)
freq = [text.count(ch) / n for ch in set(text)]
h_freq = -sum(f * math.log2(f) for f in freq)
print(f'{n} characters over {k} symbols, letter-frequency entropy '
      f'{h_freq:.3f} bits')
bits = [adaptive_bits(text, m, k) for m in range(5)]
for m, b in enumerate(bits):
    print(f'order {m}: {b:.2f} bits/char = {b * math.log(2):.2f} '
          f'nats/char, perplexity 2^bits = {2 ** b:.2f}')
print(f'zlib -9: {8 * len(zlib.compress(text.encode(), 9)) / n:.2f} bits/char')
print(f'order 0 minus the letter-frequency entropy: {bits[0] - h_freq:.4f} '
      f'bits/char, {(bits[0] - h_freq) * n:.0f} bits in all')
```

```
173428 characters over 27 symbols, letter-frequency entropy 4.078 bits
order 0: 4.08 bits/char = 2.83 nats/char, perplexity 2^bits = 16.91
order 1: 3.29 bits/char = 2.28 nats/char, perplexity 2^bits = 9.81
order 2: 2.75 bits/char = 1.90 nats/char, perplexity 2^bits = 6.70
order 3: 2.55 bits/char = 1.77 nats/char, perplexity 2^bits = 5.84
order 4: 2.78 bits/char = 1.92 nats/char, perplexity 2^bits = 6.85
zlib -9: 2.84 bits/char
order 0 minus the letter-frequency entropy: 0.0011 bits/char, 183 bits in all
```

コンテキストを追加すると、符号長はコンテキストなしの文字あたり 4.08 ビットから、3 文字のコンテキストでの文字あたり 2.55 ビットまで下がり、`zlib` の 2.84 を上回る。しかし 4 文字目を追加すると改善は逆転し、4 次モデルは文字あたり 2.78 ビットへと再び悪化する。取り得る $27^4 \approx 530{,}000$ 通りのコンテキストの大半は小説中にごくまれにしか現れないため、モデルはそれらの次の文字の確率を十分に学習できず、カウントがウォームアップする間に費やされた余分なビットはすべてそのままファイルサイズに跳ね返るのである。

対照的に、コンテキストをまったく使わない場合、学習すべきことはごくわずかしかない。出力の 1 行目は小説の経験的な文字頻度のエントロピーである文字あたり 4.078 ビットを報告しており、これはコンテキストを使わない符号器がそれらの頻度を事前に知っていた場合に達成できるレートである。0 次モデルはゼロから出発するが、最後の行はそれらの頻度を学習するのにかかった正確なコストを測っており、文字あたりわずか 0.0011 ビット、小説全体でも 183 ビットにすぎない。26 個の文字頻度の学習は安上がりだが、50 万個のコンテキストの学習はそうはいかない（nat とパープレキシティの列については 37.2.4.1 節で改めて扱う）。

適応的モデルはパラメータの送信を不要にする代わりに、学習途上にあるファイルの序盤では予測精度が低くなるというコストを払う。事前学習済みモデルはこれとは逆のトレードオフをとる。重みさえ一度共有されていれば、どのファイルも最初の 1 バイト目から高い圧縮率で圧縮できる。Delétang et al. (2023) は 700 億パラメータの言語モデルを用いてこのトレードオフを実証している。その次トークン確率で算術符号器を駆動すると、ImageNet の画像パッチは元のサイズの 43.4%（PNG では 58.5%）、LibriSpeech の音声は 16.4%（FLAC では 30.3%）に圧縮される。ただし、これらの数値にはモデル自体のサイズは含まれていない。1 パラメータあたり 2 バイトとすると、重みは 140 GB を占める（各データセットは 1 GB にすぎない）ため、モデルサイズを含めた途端に標準コーデックには勝てなくなる。

37.2.2.3 最小記述長

『The Time Machine』の実験では、各適応的モデルは読み進めながら自身のカウントを学習するコストを支払っていた。その負担を取り除くと、ランキングは完全に逆転する。まず小説全体からカウントを集め、そのカウントを使って同じ小説をスコア付けすると、より大きなルックアップテーブルはテキストをそのまま暗記できるため、4 次は 3 次に勝ち、5 次は 4 次に勝つといった具合になる（演習問題 4）。事後的にデータへどれだけよく適合するかだけで判断すれば、表現力の高いモデルが負けることは決してない。しかし、モデルをまだ持っていない受信者には、まずそのモデルの記述を送らなければならない。**最小記述長**（minimum description length、MDL）の原理は、モデルとデータを合わせた総符号長を最小化するモデル $\mathcal{M}$ を選ぶことでこの問題を解決する。

$$\widehat{\mathcal{M}} = \operatorname*{argmin}_{\mathcal{M}} \big[L(\mathcal{M}) + L(\mathcal{D} \mid \mathcal{M}) \big], \tag{37.18}$$

ここで、$L(\mathcal{M})$ はモデルを指定するのに必要なビット数であり、$L(\mathcal{D} \mid \mathcal{M})$ はそのモデルのもとでのデータ $\mathcal{D}$ の対数損失、すなわち前項までの議論によりビット単位の符号長である。MDL のもとでは、より精巧なモデルが選ばれるのは、モデルの記述に要するビット数よりもデータ上で節約できるビット数の方が大きい場合に限られる。この意味で、MDL はオッカムの剃刀（Occam's razor）をビット単位で定量化したものである。さらに、モデルの記述が事前分布から $L(\mathcal{M}) = -\log_2 P(\mathcal{M})$ によって与えられるとき、この和の最小化は負の対数事前確率と負の対数尤度の和の最小化にほかならないため、MDL は最大事後確率推定（maximum a posteriori estimation）と一致する（36.6.4 節）。

単純なコイン投げの例によって、両方の項が具体的になる。$n = 1{,}000$ 回のコイン投げで 600 回表が出たとする。公平なコインという仮説 $\mathcal{M}_0$ には送信すべきパラメータがなく、各試行を 1 ビットで符号化するため、$L(\mathcal{D} \mid \mathcal{M}_0) = 1{,}000$ ビットとなる。偏りのあるコインという仮説 $\mathcal{M}_1$ では、まずその偏り（表の確率）θ を送信しなければならず、そのためには有限の精度を選ぶ必要がある。n 回の試行における表の標本比率は $1/\sqrt{n}$ のスケールで揺らぐため、θ の桁を $1/\sqrt{n}$ より細かく送ってもサンプリングノイズを記述するのにビットを浪費するだけである（演習問題 5）。間隔 $1/\sqrt{n}$ で区間 $(0, 1)$ 上に設けたグリッドには $\sqrt{n}$ 個の点があるため、最も近いグリッド点を指定するコストは $L(\mathcal{M}_1) = \log_2 \sqrt{n} = \tfrac{1}{2} \log_2 1{,}000 \approx 5$ ビットとなる。θ が与えられれば、データの符号化には $L(\mathcal{D} \mid \mathcal{M}_1) \approx 1{,}000\, H_2(0.6, 0.4) = 971$ ビットを要し、合計は

$976 < 1{,}000$ ビットになる。したがって MDL は偏りのあるコインを選択する。パラメータに 5 ビットを費やすことで、データ上で 29 ビットが節約されるからである。もし表が 520 回しか出なかったとすれば、データ上の節約はわずか 1.2 ビットにとどまり、パラメータのコストである 5 ビットを下回るため、MDL は公平なコインを維持する。このように実数パラメータ 1 個あたり $\frac{1}{2}\log_2 n$ ビットを課す規準は、**ベイズ情報量規準**（Bayesian information criterion）として知られている。

実は、『The Time Machine』の実験はすでに MDL を適用していたのである。適応的符号長は、モデルを学習するコストがビットストリームの中に直接折り込まれた、データの有効かつ完全な記述になっている。0 次モデルでは、その学習コストはごくわずかである。$\frac{1}{2}\log_2 n \approx 8.7$ ビットを 26 個の自由な文字頻度のそれぞれに課すと約 226 ビットと予測され、実際に観測された 183 ビットにかなり近い値となる（この規準は漸近展開の主項のみを残したものである）。4 次モデルでは、観測されたコンテキストごとに別々の分布を学習するコストが、コンテキストの 4 文字目によって節約されるビット数を上回ったため、適応的符号長は自動的に 3 次を選好したのである。

ここで、素朴なパラメータ数による評価は大きく的を外してしまうことに注意しよう。4 次モデルは形式上 $27^4 \cdot 26 \approx 14$ 百万個の自由な頻度パラメータを持つため、1 個あたり 8.7 ビットとすると、ベイズ情報量規準は 120 百万ビットものペナルティを課すことになる（1 文字 8 ビットの非圧縮でもわずか 1.4 百万ビットしか要しない小説に対してである）。パラメータ数を数える漸近論は、パラメータよりもはるかに多くのデータがあることを前提としている。現実には、そうした 4 文字のコンテキストの大半は一度も現れず、適応的符号はそれらに対して 1 ビットも課さない。現代のニューラルネットワークを含め、モデルがデータよりも多くのパラメータを持つ状況では常に、適応的符号長こそが意味を保ち続ける記述長となる。

37.2.3 エントロピー率と典型系列

ここまでは、モデルを所与のものとしてその符号長を測ってきた。ここで問いを裏返そう。記憶を持つ情報源に対して、**考え得る最良の**モデルの符号はどこまで短くできるのだろうか。

37.2.3.1 エントロピー率

独立な記号に対しては、圧縮の基本下限は 1 記号あたりのエントロピーであった。しかし実際のテキストは依存関係に満ちている。`q` の後には `u` が続き、`th` の後には `e` や `a` が `z` よりはるかに高い頻度で現れる。記憶を持つ情報源を解析するため、系列 $X_1, X_2, \dots$ は**定常**（stationary）であるとする。すなわち、任意の記号ブロックの同時分布が時間をずらしても変化しないものとする。

37.1.1.3 節で導入した結合エントロピーと条件付きエントロピーを用いる。本節の残りの部分では、確率変数に対するビット単位のエントロピーを H で表し、明示的な確率ベクトルの 2 進エントロピーは引き続き $H_2(p_1, \dots, p_k)$ で表す。結合エントロピー $H(X_1, \dots, X_n)$ は n 記号のブロック全体を単一の確率変数として扱い、条件付きエントロピー $H(X_n \mid X_{<n}) = \mathbb{E}\big[-\log_2 p(X_n \mid X_{<n})\big]$ は考え得る最良の次記号予測器、すなわち真の情報源そのものの期待対数損失を測る。連鎖律 $H(X, Y) = H(X) + H(Y \mid X)$ を繰り返し適用すると、ブロックエントロピーは条件付きエントロピーの和 $H(X_1, \dots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_{<i})$ へと分解される。情報源の**エントロピー率**（entropy rate）とは、条件付きエントロピーの極限、あるいは同値な表現として長いブロックにおける記号あたりエントロピーの極限である。

$$H_{\text{rate}} = \lim_{n \to \infty} \frac{1}{n} H(X_1, \dots, X_n) = \lim_{n \to \infty} H(X_n \mid X_{<n}). \tag{37.19}$$

ブロックエントロピーは n 個の条件付きエントロピーの和であるため、第 1 の式は、第 2 の式にその極限が現れている各項の逐次平均になっている。任意の定常情報源に対して両方の極限は存在

して一致し、かつ $\frac{1}{n}H(X_1, \dots, X_n)$ は H_{rate} に向かって単調に減少するが (Cover and Thomas 1999)、これらの事実はここでは認めることとする。直感的な意味は明快である。条件付けによってエントロピーが増加することは決してないため（37.1 節）、コンテキストの記号が 1 つ増えるごとに次の記号の不確実性は減少するか変化しないかのいずれかであり、非増加系列の逐次平均もまた非増加となる。記号が独立同分布であるとき、両方の極限は直ちに $H(X_1)$ へと帰着する。

2 状態のマルコフ連鎖を考えると、記憶がいかに大きく下限を引き下げるかがわかる。連鎖が a または b を出力し、現在の記号に確率 0.9 で留まり、確率 0.1 で切り替わるとしよう。対称性により両方の記号は同じ頻度で現れるため、単一の記号を単独で見ればエントロピーは $H(X_1) = 1$ ビットである。しかし直前の記号が与えられると、次の記号は切り替わる確率が 0.1 の偏りのあるコインのように振る舞うため、$H(X_n \mid X_{n-1}) = H_2(0.9, 0.1) = 0.47$ ビットとなり、さらに過去を遡っても新たな情報は加わらない。記憶があるというだけで、基本下限は記号あたり 1 ビットから 0.47 ビットへと下がるのである（演習問題 6 では一般の非対称な 2 状態連鎖を解析する）。

任意の定常情報源に対し、いかなる一意復号可能符号も平均して記号あたり H_{rate} ビット未満にはできない。実際、n 記号のブロックに対する任意の符号は n 記号ブロックのアルファベット上の記号符号であるから、（マクミランの定理により一意復号可能符号へ拡張された）エントロピー限界によってブロックあたり少なくとも $H(X_1, \dots, X_n)$ ビットを費やさなければならず、単調性から $H(X_1, \dots, X_n) \geq nH_{\text{rate}}$ が保証される。

同様に重要なこととして、この下限は達成可能である。情報源の真の条件付き確率で駆動される算術符号器は、Equation 37.17 により、メッセージのサプライザルに 2 ビットを加えた値より狭義に少ないビット数しか費やさない。n 記号のブロックのそれ自身の分布のもとでの期待サプライザルは $H(X_1, \dots, X_n)$ であるから、この符号器の平均符号長は記号あたり $\frac{1}{n}H(X_1, \dots, X_n) + \frac{2}{n}$ ビット未満となり、これは H_{rate} へと（n が大きくなるにつれて）収束する。

古典的な実験において、Shannon は印刷された英語の次の文字を人間に推測させ、英語のエントロピー率が文字あたりおおよそ 0.6 から 1.3 ビットの間に位置すると結論づけた (Shannon 1951)。これは一様ランダムな文字とスペースに必要な $\log_2 27 = 4.75$ ビットをはるかに下回る。ここで、実際に動作するいかなる圧縮器もエントロピー率の**上界**を与えることに注目しよう。先ほどの 3 次カウントモデルによる文字あたり 2.55 ビットもそのような上界の 1 つである一方、いかなる圧縮器も下界を証明することは決してできない。本章の冒頭に戻ると、文字あたり 1.2 nat の損失を持つ言語モデルは文字あたり 1.73 ビットを達成しており、カウントモデルの 2.55 を大きく下回るものの、依然として Shannon の推定範囲の上端よりは上にある。

現在得られている最良の上界は、大規模言語モデルによるものである。算術符号器を駆動することで、Delétang et al. (2023) のモデルは Wikipedia のテキストをバイトあたり 1 ビット未満、すなわち Shannon の推定範囲の内側まで圧縮する。ただし、これを英語のエントロピー率として解釈する前に 3 つの注意点がある。モデルのサイズが含まれていないこと、テスト用テキストの一部が訓練中に見られていた可能性があること、そして Wikipedia のマークアップの 1 バイトは Shannon の 27 文字アルファベットの 1 記号とは異なることである。真のエントロピー率がこれらの上界より下のどこに位置するのかは、誰にもわかっていない。

37.2.3.2 典型系列

ここまでは**平均**符号長の上下界を求めてきた。では、1 本の長いメッセージはどのような姿をしており、起こりやすいメッセージは実際に何通り存在するのだろうか。この問いに答えることで、起こりやすいメッセージの集合の大きさを数える指数という第 2 の角度からエントロピーに光が当たる。本節の残りの部分はこの項に依存しないため、初読時には読み飛ばしても構わない。数値を具体的に保つため、値 1 を確率 0.3 でとる独立な 2 進記号からなる、長さ $n = 200$ のメッセージを考えよう。

単独で最も確率の高いメッセージはすべてが 0 の系列であり、その確率は $0.7^{200} = 2^{-103}$ である。期待個数である 60 個の 1 を含むメッセージの確率は $0.3^{60}\,0.7^{140} = 2^{-176.3}$ であり、これは 2^{-nH}（ただし $H = H_2(0.3, 0.7) = 0.881$ ビット）に等しく、そのようなメッセージは $\binom{200}{60} = 2^{172.2}$ 通り存在する。一方、文字列の総数は 1 が 100 個のときにピークを迎え $2^{195.9}$ 通りにもなるが、その 1 本ずつの確率は 2^{-225} にすぎないため、それらすべてを合わせても全確率質量のわずか 2^{-29} しか担っていない。ほとんどのメッセージは起こりにくく、しかも最も起こりやすいメッセージは集団を代表するものではないのである。

表記 $x_{1:n}$ で系列 $x_1, \dots, x_n$ を表すことにする。記号は互いに独立であるため、記号あたりのサプライザルはそれぞれ期待値 $H = H_2(P)$ を持つ独立同分布な確率変数の標本平均になる。大数の法則により、この標本平均はその期待値へと確率収束する。

$$-\frac{1}{n}\log_2 P(X_{1:n}) = \frac{1}{n}\sum_{i=1}^{n} -\log_2 p(X_i) \xrightarrow{P} H. \tag{37.20}$$

したがって、任意の固定された許容誤差 $\delta > 0$ に対し、長いメッセージのサプライザルは 1 に近づく確率で誤差 $n\delta$ 以内の範囲（nH ビットの周囲）に収まる。この現象を**漸近的等分配性**（asymptotic equipartition property、AEP）と呼び、それによって捉えられるメッセージ全体は許容誤差 δ における**典型集合**（typical set）をなす。

$$A_\delta = \left\{ x_{1:n} : \left| -\frac{1}{n}\log_2 P(x_{1:n}) - H \right| < \delta \right\}. \tag{37.21}$$

命題（AEP） $X_1, \dots, X_n$ **を有限アルファベット上の分布** P**（2 進エントロピー** H**）からの独立な抽出とし、** $\delta > 0$ **とする。このとき次が成り立つ。(i)** $P(A_\delta) \to 1$（$n \to \infty$ **のとき**）**。(ii) すべての** $x_{1:n} \in A_\delta$ **は** $2^{-n(H+\delta)} < P(x_{1:n}) < 2^{-n(H-\delta)}$ **を満たす。(iii)** $|A_\delta| \le 2^{n(H+\delta)}$ **であり、また** $|A_\delta| \ge (1-\eta)\,2^{n(H-\delta)}$ **が** $P(A_\delta) \ge 1-\eta$ **である限り成り立つ。**

証明 (i) は Equation 37.20 から直ちに従い、(ii) は A_δ を定義する不等式において 2 のべき乗をとったものにすぎない。(iii) については、(ii) の上下界を A_δ のすべての元にわたって足し合わせることで $1 \ge P(A_\delta) > |A_\delta|\,2^{-n(H+\delta)}$ および $1-\eta \le P(A_\delta) < |A_\delta|\,2^{-n(H-\delta)}$ が得られ、ここから要素数に関する両方の上下界が直ちに従う。■

したがって指数スケールで見れば、ほぼすべての確率質量は約 2^{nH} 本の典型系列に集中し、それぞれの系列は約 2^{-nH} の確率で生じる。ただし、「約」という言葉の裏には $2^{\pm n\delta}$ の因子が隠れており、有限の長さではこれが無視できない大きさになることに留意しよう。実際、$n = 200$、$\delta = 0.1$ のとき、この幅は 2^{20} にもなる。

AEP から、シャノンの情報源符号化定理（Shannon's source-coding theorem）が直ちに導かれる (Shannon 1948; Cover and Thomas 1999)。圧縮するには、A_δ 内の各系列に $\lceil n(H+\delta) \rceil$ ビットの固定長インデックスを割り当て、非典型的なメッセージに対しては失敗を宣言すればよい。情報源の 1 記号あたりに費やすビット数を符号の**レート**と呼ぶ。ここでのレートは（丸めを除けば）$H + \delta$ であり、失敗確率は $P(A_\delta^c) \to 0$ となる。$\delta > 0$ はいくらでも小さくできるため、H より大きいすべてのレートが達成可能である（先頭に 1 ビットのフラグを付加し、非典型的なメッセージには非圧縮のコピーへフォールバックするようにすれば、同じ漸近レートで厳密に可逆な符号にできる）。逆に、レート $R < H$ の任意の固定長符号は 2^{nR} 個の符号語しか持たないため、表現できるメッセージの集合 B の要素数は高々 2^{nR} 個にとどまる。$\delta < H - R$ と選ぶと、次式が得られる。

$$P(B) \le P(B \cap A_\delta) + P(A_\delta^c) < 2^{nR}\,2^{-n(H-\delta)} + P(A_\delta^c) \to 0.$$

したがって、H 未満のレートで動作するいかなる固定長符号も、1 に収束する確率で失敗する。同じ定理は定常エルゴード情報源に対しても H_{rate} で H を置き換えることで成り立つが、この一般化は証明なしに認めることとする (Cover and Thomas 1999)。

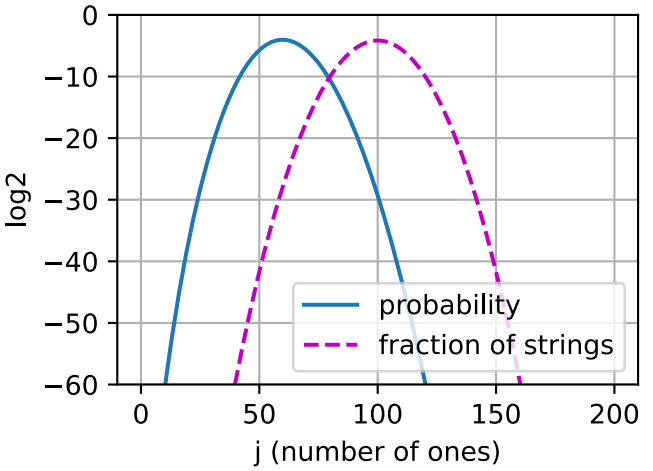
このベルヌーイ情報源に対しては、1 の個数 j が同じメッセージはすべて同じ確率を持つため、典型集合の要素数を**厳密に**数え上げることができる。以下のセルでは、全 2^{200} 通りの 2 進文字列を j ごとにグループ化し、$\delta = 0.1$ における典型性の条件を満たすグループの個数と確率を合計して、得られた集合のサイズを基準値 $2^{nH} = 2^{176.3}$ と比較する。

```
n, p, delta = 200, 0.3, 0.1
j = np.arange(n + 1)                                # number of ones
log_fact = np.concatenate([[0.0], np.cumsum(np.log(np.arange(1, n + 1)))])
log2_count = (log_fact[n] - log_fact[j] - log_fact[n - j]) / np.log(2)
bits_per_symbol = -(j * np.log2(p) + (n - j) * np.log2(1 - p)) / n
log2_prob_j = log2_count - n * bits_per_symbol       # log2 P(j ones)
h = -(p * np.log2(p) + (1 - p) * np.log2(1 - p))
typical = np.abs(bits_per_symbol - h) < delta
log2_size = np.logaddexp2.reduce(log2_count[typical])
prob = (2 ** log2_prob_j[typical]).sum()
print(f'H = {h:.3f} bits/symbol, nH = {n * h:.1f} bits')
print(f'typical: 2^{log2_size:.1f} of 2^{n} strings, '
      f'{j[typical].min()} <= j <= {j[typical].max()}')
print(f'probability they carry: {prob:.4f}')
print(f'bounds: {prob:.3f} * 2^{n * (h - delta):.1f} <= |A| '
      f'<= 2^{n * (h + delta):.1f}')
```

```
H = 0.881 bits/symbol, nH = 176.3 bits
typical: 2^188.8 of 2^200 strings, 44 <= j <= 76
probability they carry: 0.9893
bounds: 0.989 * 2^156.3 <= |A| <= 2^196.3
```

$2^{188.8}$ 通りの典型文字列（1 の個数が 44 個から 76 個までのもの）は、全 2 進文字列のうち 2^{11} 分の 1 を占めるにすぎないが、それら全体で確率 0.989 を担っている。その個数が 2^{nH} を数千倍上回っているのは、$\delta = 0.1$ としたことで 1 を最大 76 個含む文字列（60 個付近の文字列よりもはるかに数が多い）まで許容されるためである。それでもなお、このサイズは命題が与える上下界である $0.989 \cdot 2^{156.3}$ と $2^{196.3}$ の間に余裕をもって収まっている。確率と文字列の割合の両方を j の関数としてプロットし、両者がどれほど鮮やかに分離するか見てみよう。

```
d2l.plot(j, [log2_prob_j, log2_count - n], 'j (number of ones)', 'log2',
         legend=['probability', 'fraction of strings'],
         ylim=[-60, 0])
```



2 本の曲線のピークは大きく離れている。確率質量は $j = 60$ でピークに達するが、そこでの文字列は全可能性のうちわずか 2^{-28} の割合しか占めない。一方、文字列の個数は $j = 100$ でピークに達するが、それらの合計確率はわずか 2^{-29} にすぎない。単独で最も確率の高い文字列（$j = 0$、すべて 0）の確率は 2^{-103} であり、グラフの範囲から完全に外れてしまう。典型系列とは、最も確率の高い文字列でもなければ最も数の多い文字列でもなく、個数と個々の確率の積が最大になる狭い帯域に存在する系列なのである。

典型集合は証明のための強力な道具ではあるが、実用的な符号器としては不器用である。これに直接インデックスを振ると、すべての典型文字列に 189 ビット（13 ビットだけ $nH = 176.3$ より多い）を要するうえ、おおよそ 100 本に 1 本のメッセージで失敗する。これに対し、真のモデルを用いた算術符号化は厳密に可逆であり、Equation 37.17 により平均 $nH + 2 = 178.3$ ビット未満で済む。典型集合が与えてくれるのは幾何学的な直感である。すなわち、長いメッセージはほぼ確実に情報源の真の組成を反映し、約 2^{nH} 通りのメッセージがその組成を共有している。

37.2.4 パープレキシティ

実際には、言語モデルがその性能をファイルサイズとして報告することはほとんどない。代わりに、ホールドアウトされたテキストに対する単一の数値である**パープレキシティ**を報告するが、これは単に本節の符号長を別の単位で表したものにすぎない。

37.2.4.1 分岐数としてのパープレキシティ

n 個のトークンからなるテキストと、次のトークンの確率 $q(\cdot \mid x_{<i})$ を割り当てるモデルが与えられたとき、nat 単位でのトークンあたり平均対数損失とその指数である**パープレキシティ**(Jelinek et al. 1977) は、次式で定義される。

$$\mathrm{CE} = -\frac{1}{n}\sum_{i=1}^{n} \log q(x_i \mid x_{<i}), \qquad \mathrm{PPL} = \exp(\mathrm{CE}) = \Big(\prod_{i=1}^{n} q(x_i \mid x_{<i})\Big)^{-1/n}. \tag{37.22}$$

分布の引数を省略して書いた CE は、この特定のテキストに対するモデルの経験損失を表しており、情報源から抽出されたテキストにわたるその期待値が情報源とモデルの間の交差エントロピーである。第 2 の表式から、パープレキシティは観測されたトークンに割り当てられた確率の幾

何平均の逆数であることがわかる。8.3.2 節で導入したように、パープレキシティは**実効分岐数**（effective branching factor）として直ちに解釈できる。毎ステップで k 個のトークンに一様に確率を割り振るモデルのパープレキシティは k になり、決定論的な系列に対する完全な予測器のパープレキシティは 1 になる。

37.2.2.2 節における『The Time Machine』の結果は、文字レベルのテキストでこれを具体的に示している。何の知識も持たないモデルは全 27 記号から一様に推測するため、パープレキシティは 27 になる。文字頻度を学習するだけでパープレキシティは 16.91（1 文字あたりおおよそ 17 通りの等確率な選択肢）に下がり、3 文字のコンテキストを使えば 5.84 まで下がる。平均すると、このモデルが次の文字について抱える不確実性は、6 面サイコロの出目に対する不確実性とほぼ同程度なのである。なお、先ほどのコードでは 2 の「文字あたりビット数」乗としてパープレキシティを計算したのに対し、Equation 37.22 では e の「文字あたり nat 数」乗を用いているが、$\mathrm{PPL} = b^{\mathrm{CE}_b}$ が任意の対数の底 b に対して成り立つため、両者は同じ数値を与える（演習問題 8）。

37.2.4.2 パープレキシティの下限

符号長がエントロピー率を下回れないのと同様に、パープレキシティにも基本的な下限がある。定常情報源に対して、期待パープレキシティは少なくとも $2^{H_{\text{rate}}}$ である。CE は nat 単位で測られるのに対し、H_{rate}（Equation 37.19）はビット単位で測られるため、係数 $\ln 2$ によって両者の単位を変換する。

命題（パープレキシティの下限） **$X_1, \dots, X_n$ を定常情報源から抽出された系列とし、q を任意のモデル、CE と PPL をこのテキストに対する経験損失とパープレキシティ（Equation 37.22、モデル q のもとでの値）とする。このとき次式が成り立つ。**

$$\mathbb{E}[\mathrm{CE}] \geq H_{\text{rate}} \ln 2, \qquad \mathbb{E}[\mathrm{PPL}] = \mathbb{E}\big[e^{\mathrm{CE}}\big] \geq 2^{H_{\text{rate}}}. \tag{37.23}$$

証明 任意のプレフィックス $x_{<i}$ を固定する。$x_{<i}$ が与えられたときのモデルの次トークン期待損失は、真の条件付き分布 $p(\cdot \mid x_{<i})$ と $q(\cdot \mid x_{<i})$ の間の交差エントロピーであり、ギブスの不等式により情報源の条件付きエントロピー以上である。プレフィックスにわたって平均をとると $\mathbb{E}[-\log q(X_i \mid X_{<i})] \geq \mathbb{E}[-\log p(X_i \mid X_{<i})]$ が得られる。系列の対数確率は $\log p(x_{1:n}) = \sum_{i} \log p(x_i \mid x_{<i})$ と分解されるため、i について足し合わせて n で割ると $\mathbb{E}[\mathrm{CE}] \geq \frac{\ln 2}{n} H(X_1, \dots, X_n) \geq H_{\text{rate}} \ln 2$ となる。ここで第 2 の不等号は、記号あたりのブロックエントロピーの単調減少性から従う。最後に、e^{x} は凸関数であるから、イェンセンの不等式により $\mathbb{E}[e^{\mathrm{CE}}] \geq e^{\mathbb{E}[\mathrm{CE}]} \geq 2^{H_{\text{rate}}}$ が成り立つ。$\blacksquare$

要するに、いかなるモデルも平均的には、情報源が本来持つランダム性が許す以上にパープレキシティを低くすることはできない。規模感をつかむため、あるテキストのエントロピー率が文字あたり 1 ビット（英語に対する Shannon の推定範囲内の値）だったとしよう。すると、いかなる文字レベルモデルも期待パープレキシティ 2（1 文字につき公平なコイン投げ 1 回分）を下回ることはできず、先ほどの 3 次カウントモデルの 5.84 はその下限よりはるかに上に位置することになる。実際のテキストに対しては、この下限は上界を通してしか知ることのできないエントロピー率そのものと同じだけ不確かである。

この命題が下界を与えているのはあくまで**期待値**であり、1 本の短いテキストに対するパープレキシティはこの下限を容易に下回り得ることに十分注意しよう。再び、`a` を確率 0.9 で出力する情報源（下限は $2^{H} = 1.38$）を考える。真のモデルのもとでさえ、10 記号の文字列は 1.38 未満のスコアを、`b` を 1 つも含まないときはいつでも記録し、それは確率 $0.9^{10} = 35\%$ で起こる。この情報源からの長いテキストであっても、パープレキシティはほぼ半分の確率で依然として 2^{H} を下回るが、その差は長さが伸びるにつれて消失していく。より一般に、エルゴード情報源からの 1 本の長いテキストを真のモデルでスコア付けしたときのパープレキシティは $2^{H_{\text{rate}}}$ へと（$n \to \infty$ のとき）収束するが、この事実はここでは認めることとする。

37.2.4.3 トークナイザ間のパープレキシティ

パープレキシティはトークンあたりで平均されるため、その値はトークナイザに直接依存する。したがって、2 つのモデルを生のパープレキシティで比較できるのは、両者が同じトークナイザを共有している場合に限られる。8.3.2.4 節では 3 種類のトークナイザにわたってこの落とし穴を実演し、総サプライザルをテキストのバイト数で割ることでこれを解決した。本節の言葉で言えば、この対処法は単にパープレキシティを元のテキストの単位あたりの符号長へと戻しているにすぎない。n_{char} 文字のテキストが n_{tok} 個のトークンに分割されているとき、総符号長は $n_{\text{tok}} \log_2 \text{PPL}$ ビットとなり、次式が得られる。

$$\text{bpc} = \frac{n_{\text{tok}}}{n_{\text{char}}} \log_2 \text{PPL}, \tag{37.24}$$

ここで分母はテキスト自体のみに依存する。27 記号のテキストのようにすべての文字が 1 バイトである場合、文字あたりビット数は本編で用いたバイトあたりビット数と完全に一致する。

符号長からパープレキシティに至るまで、本節で扱ったすべての量は 1 つの決定的な要件に依拠している。それは、エンコーダとデコーダの双方が各記号に対するモデルの確率 $q(x_i \mid x_{<i})$ を計算できなければならないという点である。生成モデルの出力を実データと比較するときのように、2 つの分布にサンプルを通してしかアクセスできない場合には、そのような符号は存在しない。37.3 節と 37.4 節では、手元にサンプルしかない状況で分布を比較する手法を展開する。

37.2.5 まとめ

モデルの対数損失は、適切な符号と組み合わせることでファイルサイズそのものになる。記号ごとの接頭符号はいずれも各符号語を整数ビットに丸めなければならず、そのため最適な期待符号長は $H_2(P)$ ビットと $H_2(P)+1$ ビットの間に位置する。算術符号化はメッセージごとに 1 回だけ丸めを行うことでこの記号ごとのペナルティを取り除き、自己回帰言語モデルを含む任意の次記号モデルで駆動することで、系列をその総サプライザルに高々 2 ビットを加えた長さへと圧縮する。不一致モデル Q のもとでは期待符号長は交差エントロピー $\text{CE}_2(P, Q)$ となり、KL ダイバージェンスはモデル Q を P の代わりに用いるオーバーヘッドを測る。『The Time Machine』に対しては、適応的 3 次カウントモデルがテキストを文字あたり 2.55 ビットに圧縮して `zlib` を上回る一方、4 次では希少なコンテキストの学習に費やすビット数が 4 文字目による節約分を上回るため悪化し、最小記述長の原理を具体的に示している。記憶を持つ定常情報源に対しては、いかなる一意復号可能符号も平均して記号あたりエントロピー率 H_{rate} 未満のビット数にはできない。最後に、パープレキシティはこの符号長を単に指数化したものであり、実効分岐数として働き、期待値の下界 $2^{H_{\text{rate}}}$ に従い、異なるトークナイザ間では文字あたりビット数またはバイトあたりビット数へ変換し直した後にのみ比較できる。

37.2.6 演習問題

1. **マクミランの定理**。符号語のあらゆる連結がただ 1 通りにしか区切れないとき、その符号は一意復号可能である。
    1. 符号 $\{0, 01, 011\}$ が一意復号可能であるがプレフィックスフリーではないことを示し、クラフトの不等式を満たすことを確かめよ。
    2. すべての一意復号可能符号がクラフトの不等式 Equation 37.13 を満たすことを証明せよ。ヒント：$\big(\sum_i 2^{-\ell_i}\big)^m$ を、m 個の符号語からなるすべての列にわたる $2^{-(\text{total length})}$ の和として展開し、それらの列を合計長 L（ただし $m \le L \le m\ell_{\max}$）ごとにまとめる。一意復号可能性により高々 2^L 通りの合計長 L の列しか存在し得な

いため、和全体は高々 $m\ell_{\max}$ となる。あとは両辺の m 乗根をとって $m \to \infty$ とすればよい。

2. **ハフマン符号**。ハフマンのアルゴリズム (Huffman 1952) は、最も確率の低い 2 つの記号をその確率の和を持つ 1 つの記号へと繰り返し統合し、各統合の 2 つの要素に 0 と 1 を割り当てることで接頭符号を構成する。

   1. これを $P = (0.6, 0.3, 0.1)$ および $(0.4, 0.3, 0.2, 0.1)$ に対して実行せよ。それぞれの符号の符号語長と期待符号長を、シャノン符号および H_2 と並べて示せ。
   2. ハフマン符号がすべての接頭符号の中で最小の期待符号長を持つことを証明せよ。まず、最適符号においては $p_i > p_j$ ならば $\ell_i \le \ell_j$ が成り立つこと、また最長の 2 つの符号語は同じ長さを持ち木の中で兄弟ノードにとれることを示せ。次に、最も確率の低い 2 つの記号 i, j を統合すると問題が 1 記号少ない場合へと帰着し、$\mathbb{E}[\ell] = \mathbb{E}[\ell'] + p_i + p_j$（ただし ℓ' は統合後の符号の符号語長）が成り立つことを示せ。帰納法により結論を導け。ヒント：2 つの符号語を入れ替えて期待符号長を比較する。
   3. a を確率 0.9 で出力する情報源に対し、いかなる記号符号も記号あたり少なくとも 1 ビットを費やすことを示せ。次に、記号の**ペア**を 4 記号の情報源としてハフマンのアルゴリズムで符号化し、本文の「元の記号あたり 0.645 ビット」を確かめよ。3 つ組に対しても同様に行え。それぞれの場合で、エントロピーからの超過分は上界 $1/n$ と比べてどうなっているか。
   4. 長さ $1, 2, \dots, k-1, k-1$ を持つ線形推測符号を考えよ。確率を $p_1 \ge \cdots \ge p_k$ と並べたとき、$p_i \ge \sum_{j>i} p_j$ がすべての i について成り立つならばハフマンのアルゴリズムがこれらの長さを返し得ること、したがってそのとき線形推測符号が最適な接頭符号であることを示せ。情報源 $(0.4, 0.3, 0.2, 0.1)$ はこの条件に反している。この情報源に対して線形推測符号は最適か。

3. [code] **厳密な算術符号化**。●37.2.2.1 節の区間縮小では、`lo` と `hi` を浮動小数点数として保持している。$n = 1000$ 記号のメッセージ（各記号は a が確率 0.9 で生じ、それ以外は b となる）を 1 つ生成せよ。

   1. このメッセージに対してモデル $q(\mathtt{a}) = 0.9$ で浮動小数点による区間縮小を実行せよ。何記号目で `hi - lo` が確率の積から初めて 0.1% を超えてずれ、また何記号目で 0 になるか。両方の答えを倍精度浮動小数点数の 53 ビットの仮数部と関連づけよ。
   2. 厳密な有理数（`fractions.Fraction`）を用いて区間縮小を繰り返せ。本文のコードセルと同様に最終区間内の 1 点を指定し、モデルを再現してメッセージを復元するデコーダを実装せよ。さらに、その符号長をメッセージのサプライザルおよび $nH_2(0.9, 0.1)$ と比較せよ。
   3. 同じメッセージを不一致モデル $q(\mathtt{a}) = 0.7$ で符号化せよ。符号長の増加分を $nD^{(2)}_{\mathrm{KL}}(P\|Q)$ と比較せよ。1 本のメッセージに対してこれら 2 つの数値が異なるのはなぜか。

4. [extended] **適応的コンテキストモデルによる圧縮**。●●本文の `adaptive_bits` は、モデルの学習コストを内部に含んだ Equation 37.18 の意味での完全な記述長である。Equation 37.17 により、その予測で駆動される算術符号器が余分に費やすビット数は高々 2 ビットにとどまる。Blier and Ollivier (2018) はこのような逐次予測（prequential）符号を用いて深層ネットワークの記述長を測っている。

   1. 本文の実験を $m = 0, 1, \dots, 8$ に拡張せよ。各 m について、適応的符号長を、テキスト全体からカウントを集めた m 次モデルの文字あたり交差エントロピーおよび `zlib` と比較せよ。
   2. 適合済みモデルの交差エントロピーが単調に下がり続ける一方で、適応的符号長が m

に関して最小値を持つ理由を説明せよ。MDL はどの次数を選ぶか。適合済みモデルの交差エントロピーは Equation 37.18 のどの項を落としているか。

3. たとえば長いコンテキストのカウントが少ないときにより短いコンテキストへフォールバックするなどして、複数の次数のコンテキストを組み合わせる適応的モデルを設計せよ。試したすべてのモデルについて文字あたりビット数を `zlib` と併せて表にまとめ、最も重要だった設計上の選択について 1 段落で述べよ。

5. **コインに対する 2 部符号。**●系列 $\mathcal{D}$（n 回のコイン投げ）には表が j 回含まれ、$\hat{\theta} = j/n$ とする。仮説 $\mathcal{M}_0$ は公平なコインであり、記述すべきパラメータを持たず $L(\mathcal{D} \mid \mathcal{M}_0) = n$ ビットとなる。仮説 $\mathcal{M}_1$ は偏り θ を持つコインである。そのエンコーダは点 θ_q（$\hat{\theta}$ に最も近い、間隔 Δ で $(0,1)$ 上に設けたグリッド上の点）を送り、これに $L(\mathcal{M}_1) = \log_2(1/\Delta)$ ビットのコストがかかり、デコーダは n と Δ を知っているものとする。

   1. 分解 Equation 37.14 を用いて $L(\mathcal{D} \mid \mathcal{M}_1) = nH_2(\hat{\theta}, 1-\hat{\theta}) + nD^{(2)}_{\mathrm{KL}}(\hat{\theta} \,\|\, \theta_q)$ を示せ（ただしこの KL ダイバージェンスはこれらのパラメータを持つベルヌーイ分布どうしのものとする）。nat 単位では、この KL ダイバージェンスは 2 次の項まで展開すると $(\theta_q - \hat{\theta})^2 / (2\hat{\theta}(1-\hat{\theta}))$ となる。この展開を導出し、Equation 37.18 による $\mathcal{M}_1$ の総符号長が $nH_2(\hat{\theta}, 1-\hat{\theta})$ を超える量は高々およそ $\log_2(1/\Delta) + n\Delta^2/(8\hat{\theta}(1-\hat{\theta})\ln 2)$ ビットであることを示せ。
   2. どの間隔 Δ がこの上界を最小化するか。デコーダは Δ を知ったうえでなければ θ_q をデコードできない。それゆえ符号がこの最小点を使えないのはなぜか。また、間隔 $\Delta = 1/\sqrt{n}$（n のみに依存する）をとれば、過剰ビット数が $\tfrac{1}{2}\log_2 n + O(1)$ ビットに収まることを（$\hat{\theta}$ が 0 と 1 から有界に離れている限りにおいて）示せ。
   3. $n = 1000$ とし、近似 $L(\mathcal{M}_1) + L(\mathcal{D} \mid \mathcal{M}_1) \approx nH_2(\hat{\theta}, 1-\hat{\theta}) + \tfrac{1}{2}\log_2 n$ ビットを用いよ。本文では $\mathcal{M}_1$ が $j = 600$ において 24 ビットの差で勝つことを示した。$j = 520$ ではどちらの仮説が勝つか。$n/2$ から j がおよそどれだけ離れれば $\mathcal{M}_1$ が勝つか、またそれは公平なコインのもとでの標準偏差 $\sqrt{n}/2$（j の標準偏差）の何倍にあたるか。

6. **2 状態連鎖のエントロピー率。**$\{\mathtt{a}, \mathtt{b}\}$ 上のマルコフ連鎖が、a から b へ確率 0.1 で遷移し、b から a へ確率 0.3 で遷移し、その定常分布から出発するものとする。

   1. 定常分布とビット単位での $H(X_1)$ を求めよ。
   2. $H(X_n \mid X_{n-1}) = \sum_s \pi_s H_2(r_s, 1-r_s)$ ビットとなること（ただし π は定常分布、r_s は状態 s からの離脱確率）、およびエントロピー率がこの値に等しいことを示せ。さらにその値を計算せよ。
   3. $\frac{1}{n}H(X_1, \ldots, X_n) = \frac{1}{n}\big(H(X_1) + (n-1)H_{\text{rate}}\big)$ であり、これが H_{rate} へと減少することを示せ。
   4. 2 つの極限を考えよ。両方の離脱確率が $\tfrac{1}{2}$ に等しいとき、連鎖は独立同分布となりエントロピー率は 1 ビットである。両方の離脱確率が r に等しいとき、エントロピー率は 0 へと（$r \to 0$ のとき）収束する一方、$H(X_1)$ は 1 ビットのままである。$H(X_1)$ はこの連鎖がどれだけよく圧縮できるかについて何を語っているか。
   5. [code] 最初の連鎖を 10^5 ステップシミュレーションし、本文の `adaptive_bits` を用いて次数 0、1、2 で符号化せよ。どの次数がエントロピー率に到達し、また 0 次の符号長は何に収束するか。

7. [code] **典型集合と高確率集合。**●37.2.3.2 節のコードは、長さ $n = 200$ の 2 進文字列（$p = 0.3$ のベルヌーイ情報源からのもの）について、記号あたりのサプライザルが $\delta = 0.1$ ビット以内の誤差でビット単位のエントロピー H の周囲に収まるとき典型的であると判定する。その結果、$2^{188.8}$ 通りの典型文字列が見つかり、これは AEP の命題の上下界 $0.989 \cdot 2^{156.3} \le |A_\delta| \le 2^{196.3}$ の内側にあるものの、$2^{nH} = 2^{176.3}$ を大きく上回ってい

る。

1. 最も確率の高い文字列はどれか。その記号あたりサプライザルを計算し、典型的であるか判定せよ。この文字列が圧縮において問題にならないのはなぜか。
2. A を典型集合とし、B を $P(B) \geq 1-\eta$ を満たす任意の文字列集合とする。$P(A \cap B) \geq 1-\eta-P(A^c)$ を示し、$|B| \geq \left(1-\eta-P(A^c)\right) 2^{n(H-\delta)}$ を結論づけよ。
3. 確率が少なくとも 0.9 となる最小の集合は、確率の高い順に文字列を集めることで得られる。本文のコードと同様に 1 の個数 j ごとに文字列をグループ化し、この集合の要素数の $\frac{1}{n}\log_2$ を $n = 200$、1000、5000、20000 に対して計算せよ。これを H および $\frac{1}{n}\log_2 |A_\delta|$（同じ n における値）と比較せよ。2 つの集合のうち 2^{nH} のように増大するのはどちらか。また、δ を固定した典型集合がそうならないのはなぜか。

David Tse、EE376A/Stats376A: Information Theory, Homework 2 の問題 6 を改題。

8. **トークナイザ間のパープレキシティ。**パープレキシティはトークンあたりで計算されるため、テキストがどのようにトークンへ分割されるかに依存する。

1. b^{CE_b} が任意の対数の底 b に対して同じ値になることを示せ（ただし CE_b は底 b の単位でのトークンあたり交差エントロピーとする）。
2. 2 つの言語モデルを、10^6 文字からなる同一のテキスト上で評価する。モデル A のトークナイザはこのテキストを 2.5×10^5 トークンに分割し、A はトークンあたりパープレキシティ 20 を達成する。モデル B のトークナイザは 6.7×10^5 トークンに分割し、B はトークンあたりパープレキシティ 8 を達成する。Equation 37.24 を用いて両方の結果を文字あたりビット数に変換せよ。どちらのモデルがより精度よくテキストを予測しているか。
3. テキストのエントロピー率を文字あたり 1 ビット（英語に対して Shannon (1951) が推定した範囲内の値）とする。Equation 37.23 により、期待されるトークンあたりパープレキシティは少なくとも 2^H である（ただし H はビット単位での**トークン**あたりエントロピー率）。A のトークナイザを用いたとき、および B のトークナイザを用いたとき、任意のモデルが期待値として達成できる最小のパープレキシティはそれぞれいくらか。各モデルは自身の下限からどれだけ離れているか。
4. 2 つのパープレキシティは直接比較できない一方で、2 つの文字あたりビット数は直接比較できる理由を 2 文で説明せよ。

37.3 f ダイバージェンス

本書を通じて、最尤法（maximum likelihood）によってモデルを当てはめるときは常に、データとの一致度を KL ダイバージェンスというただ 1 つの数値で測ってきた。しかし 37.1 節の末尾で見たように、KL ダイバージェンスには決定的な盲点がある。実際に生じる事象にモデルが確率 0 を割り当てた瞬間、その値が無限大へと跳ね上がるのである。分類器がこの問題に直面することはめったにない。ソフトマックス層は厳密な 0 を決して出力しないからである。これに対し、低次元のノイズベクトルを画像空間へ写す生成ネットワークは、直ちにこの問題に突き当たる。生成されるサンプルは画像空間内の薄い曲面上に乗っており、実データを完全に外してしまうことがあるためである。では、2 つの分布を比較する方法には他にどのようなものがあり、ほぼ正しいモデルと大きく外れたモデルを見分けられるのはどの比較法だろうか。

最も単純な例を見るだけでも、何が問題になるかは明らかである。データが 1 点に集中して $P = \delta_0$ となっており、モデルはそこから距離 d だけ離れた 1 点にすべての質量を置いて $Q = \delta_d$ となっているとしよう。直感的には、$d = 0.01$ のモデルはほぼ正しく、$d = 100$ のモデルは大きく外れているのだから、有用な損失であればこの 2 つを区別できるはずである。5 つの標準的な比較法がどのような値を返すかを Table 37.2 に示す。

Table 37.2: 点質量 δ_0 と δ_d の 5 通りの比較。MMD の行では長さスケール 1 の RBF カーネルを用いている。

比較法	$d = 0.01$	$d = 1$	$d = 100$
KL ダイバージェンス $D_{\mathrm{KL}}(P\|Q)$	∞	∞	∞
イェンセン・シャノン・ダイバージェンス	$\log 2$	$\log 2$	$\log 2$
全変動距離	1	1	1
最大平均値差異 MMD^2	0.0001	0.79	2.00
ワッサースタイン距離 W_1	0.01	1	100

この表をよく見ると、5 行のうち 3 行は d が変化しても微動だにしない。これら 3 つのいずれかで訓練されるモデルは、$d = 0.01$ でも $d = 100$ でも同じ損失を受け取るため、d に関する勾配は 0 になる。第 16 章で扱った初期の敵対的生成ネットワーク（generative adversarial network、GAN）はまさにこの窮地に陥っていた。その目的関数は、定数の違いを除けばイェンセン・シャノン・ダイバージェンスの行に帰着するからである。

値が凍りついたこれら 3 つの行は、1 つの同じ族に属している。KL ダイバージェンスは、P と Q をその密度の比 p/q によって比較する。この密度比（density ratio）を保ったまま、そこに適用する凸関数を変えると、**f ダイバージェンス**（f-divergence）と呼ばれる広い族が得られる。この族に属するダイバージェンスはすべて、非負性の証明も、サンプルから推定する手法も、距離 d に対して不感になる構造的な理由も共通している。一方、d に応じて変化する残りの 2 行は密度比を一切使わない。これらについては 37.4 節で学ぶ。

この族を定式化したうえで、本節では深層学習の随所に現れる 3 つの問いに答えていく。第 1 に、KL ダイバージェンスにおいて最尤法のようにデータを第 1 引数に置くか、変分推論（variational inference）のようにモデルを第 1 引数に置くかで違いは生じるのか（37.3.2 節）。第 2 に、ダイバージェンスが小さいことは、P と Q を見分けようとする者にとって実際に何を保証するのか（37.3.3 節）。第 3 に、両方の分布がサンプルを通じてしかわからないとき、それでもなお計算できるものは何か（37.3.4 節）。37.2 節はこの最後の問いで締めくくられており、敵対的訓練（adversarial training）はその答えの上に成り立っている。

本節を通じて対数はすべて自然対数とするため、KL ダイバージェンスおよびイェンセン・シャノン・ダイバージェンスの単位は nat となる。データ分布または目標分布を P、モデルを Q と書き、それぞれの確率質量関数または密度を p と q、その**密度比**を $u = p/q$ と書く。

```
from scipy import integrate, optimize
```

Table 37.2 のすべての項目はコードで直接確かめられる。次のセルでは、δ_0 と δ_d を確率ベクトル $(1, 0)$ と $(0, 1)$ として 2 点からなる台（support）$\{0, d\}$ 上で表し、各比較法をその定義から計算する。これらの定義のうち 3 つは本節で、残りの 2 つは次節で導入する。

```
def kl_div(p, q):
    """D_KL(P||Q) on a finite support: 0 log 0 = 0, and inf if q = 0 < p."""
    s = p > 0
    with np.errstate(divide='ignore'):
        return np.sum(p[s] * np.log(p[s] / q[s]))
```

```
def five_comparisons(atoms, p, q, ell=1.0):
    """KL, JS, TV, RBF MMD^2 and W1 between pmfs p, q on sorted atoms."""
    m = (p + q) / 2                                  # the mixture for JS
    K = np.exp(-(atoms[:, None] - atoms[None, :]) ** 2 / (2 * ell ** 2))
    F_gap = np.abs(np.cumsum(p) - np.cumsum(q))[:-1]  # |F_P - F_Q| between atoms
    return {'KL': kl_div(p, q), 'JS': (kl_div(p, m) + kl_div(q, m)) / 2,
            'TV': np.abs(p - q).sum() / 2, 'MMD^2': (p - q) @ K @ (p - q),
            'W1': np.sum(F_gap * np.diff(atoms))}

delta_0, delta_d = np.array([1.0, 0.0]), np.array([0.0, 1.0])  # on atoms (0, d)
for d in (0.01, 1.0, 100.0):
    vals = five_comparisons(np.array([0.0, d]), delta_0, delta_d)
    print(f'd = {d:6}: ' + '  '.join(f'{k} {v:.4g}' for k, v in vals.items()))
```

```
d =   0.01: KL inf  JS 0.6931  TV 1  MMD^2 0.0001  W1 0.01
d =    1.0: KL inf  JS 0.6931  TV 1  MMD^2 0.7869  W1 1
d =  100.0: KL inf  JS 0.6931  TV 1  MMD^2 2  W1 100
```

37.3.1 f ダイバージェンス族

KL ダイバージェンスを一般化する前に、2 つの分布を比較する尺度が最低限満たすべき性質を明確にしておこう。そうしたうえで、KL ダイバージェンスの非負性の証明が本質的にどの要素を使っているかを問うと、f ダイバージェンス族が自然に浮かび上がる。

37.3.1.1 ダイバージェンスと距離関数

少なくとも「分布間の距離」と呼べるものは、2 つの分布が一致するときに 0 になり、異なるときには真に正の値をとるべきである。この性質を持つ 2 つの分布の関数を**ダイバージェンス**と呼ぶ。

$$D(P,Q) \geq 0, \qquad D(P,Q) = 0 \quad \text{if and only if} \quad P = Q.$$

それ以外の条件は一切課されない。一方、2 つのベクトル間のユークリッド距離 $\|\mathbf{x} - \mathbf{y}\|$（1.3 節）のような**距離関数**（metric）は、さらに $D(P,Q) = D(Q,P)$ という対称性を持ち、三角不等式（triangle inequality）$D(P,R) \leq D(P,Q) + D(Q,R)$ を満たさなければならない。KL ダイバージェンスはダイバージェンスである。実際、ギブスの不等式（37.1 節）により $D_{\mathrm{KL}}(P\|Q) \geq 0$ が保証され、等号が成り立つのは $P = Q$ の場合に限られる。しかし、距離の公理は 2 つとも満たしておらず、1 変量ガウス分布の組を考えればその両方が直ちにわかる。同節で計算したように、2 つのガウス分布間の KL ダイバージェンスは次式で与えられる。

$$D_{\mathrm{KL}}\big(\mathcal{N}(\mu_1,\sigma_1^2)\,\big\|\,\mathcal{N}(\mu_2,\sigma_2^2)\big) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}. \tag{37.25}$$

$P = \mathcal{N}(0,1)$ と $Q = \mathcal{N}(0,4)$ に対してこの式を適用すると、一方の向きでは $D_{\mathrm{KL}}(P\|Q) \approx 0.318$ nat、逆向きでは $D_{\mathrm{KL}}(Q\|P) \approx 0.807$ nat となるため、KL ダイバージェンスは対称ではない。三角不等式が破れる理由はまた別にある。平均が 0、1、2 で分散が 1 の 3 つのガウス分布をとると、隣り合う組に対してはそれぞれ $D_{\mathrm{KL}} = \frac{1}{2}$ となる一方、両端の組に対しては 2 となり、$2 > \frac{1}{2} + \frac{1}{2}$ となってしまう。KL ダイバージェンスは平均間の距離の二**乗**に比例して増大するため、真の長さというよりは二乗ユークリッド距離のように振る舞うのである。

どちらの性質の欠如も修復する必要はない。ダイバージェンスの最小化には対称性も三角不等式も使わないし、KL ダイバージェンスの非対称性はそれ自体が実質的なモデリング上の選択となる（37.3.2 節）。それでも、証明の中で P から Q、さらに Q から R へと比較をつなぐ必要があるときには、三角不等式が不可欠になる。幸い、これから構成する族には真の距離関数も含まれている。

37.3.1.2 f ダイバージェンスのテンプレート

KL を正当なダイバージェンスにしているのはギブスの不等式である。その証明を詳しく見てみよう。そこで使われる 2 つの性質を共有する関数であれば、他のどのような関数からも同様にダイバージェンスが得られるからである。

まず、KL ダイバージェンスを、Q のもとでの密度比 $u(x) = p(x)/q(x)$ の関数の期待値として書き直す。

$$D_{\mathrm{KL}}(P\|Q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} = \sum_{x} q(x)\, \frac{p(x)}{q(x)} \log \frac{p(x)}{q(x)} = \mathbb{E}_{x\sim Q}\big[u(x)\log u(x)\big].$$

このように見ると、KL ダイバージェンスは各結果を Q で重み付けし、その密度比に関数 $u\log u$ を通じてペナルティを課している。37.1 節から通しで用いている例である離散分布の組 $P = (0.6, 0.3, 0.1)$ と $Q = (0.2, 0.5, 0.3)$ でこの式を確かめてみよう。このとき密度比は $u = 3$、0.6、$\tfrac{1}{3}$ であり、それぞれに重み 0.2、0.5、0.3 が掛かる。

$$\begin{aligned} &0.2\,(3\log 3) + 0.5\,(0.6\log 0.6) + 0.3\,\big(\tfrac{1}{3}\log\tfrac{1}{3}\big) \\ &\qquad = 0.659 - 0.153 - 0.110 = 0.396 \text{ nats}, \end{aligned}$$

これは先ほど求めた KL ダイバージェンスと一致する。

次に、非負性の証明を見てみよう。37.1 節では、$-\log(q/p)$ に対して P のもとでイェンセンの不等式を 1 回適用した。代わりに Q のもとで関数 $u\log u$ に対して同じ 1 ステップの議論を行うと、そこで使われている事実は、関数が凸関数であることと $u = 1$ で 0 に等しいことの 2 つだけである。したがって、$u\log u$ をこの 2 つの性質を持つ任意の関数に置き換えても何ら支障はない。$f : (0,\infty) \to \mathbb{R}$ を $f(1) = 0$ を満たす凸関数とし、$f(0) = \lim_{u\downarrow 0} f(u)$ と定義する（これは $+\infty$ であってもよい）。当面は、$p(x) = 0$ が $q(x) = 0$ なるすべての点で成り立つと仮定し、これを $P \ll Q$ と書く。こうすれば、Q のもとでの期待値は P のすべてを捉えられる。このとき、P の Q に対する **f ダイバージェンス**は次式で定義される (Csiszár 1967)。

$$D_f(P\|Q) = \mathbb{E}_{x\sim Q}\left[f\!\left(\frac{p(x)}{q(x)}\right)\right]. \tag{37.26}$$

各点において、比 $u = p/q$ は P が x を Q に比べてどれだけ過剰または過少に表しているかを測り、f は $u = 1$ のときに 0 になるペナルティを割り当てる。この f を**ダイバージェンス関数**（divergence function）と呼ぶ。多くの文献ではこれをダイバージェンスの生成関数（generator）と呼び、以下のコードでもこれらの関数を `generators` という名前の辞書に格納している。しかし本文では、16.2 節と同様、**生成器**（generator）という語はサンプルを生成するネットワークのために取っておく。

ダイバージェンスがいつ 0 になるかを特定するには、凸性（convexity）を少しだけ強めた条件が必要になる。すなわち、関数 f が 1 **において狭義凸**（strictly convex）であるとは、$(1, 0)$ を通る

ある直線がそれ以外のすべての点で f より真に下側にあること、言い換えれば $f(u) > c\,(u-1)$ がすべての $u \neq 1$ とある傾き c に対して成り立つことをいう。

命題（非負性）。 任意の凸関数 f（ただし $f(1)=0$）および $P \ll Q$ に対し、

$$D_f(P\|Q) \geq 0,$$

が成り立つ。さらに f が 1 において狭義凸ならば、$D_f(P\|Q)=0$ となるのは $P=Q$ のとき、かつそのときに限る。

証明。 イェンセンの不等式（35.3.3.1 節）によれば、$\mathbb{E}[f(U)] \geq f(\mathbb{E}[U])$ が任意の凸関数 f と確率変数 U に対して成り立つ。これを $U = u(x)$（ただし $x \sim Q$）に適用する。証明の鍵はただ 1 つの恒等式、すなわち Q のもとで密度比の期待値が 1 になるという事実 $\mathbb{E}_Q[u] = \sum_x q(x)\,p(x)/q(x) = \sum_x p(x) = 1$ にある。したがって、次が成り立つ。

$$D_f(P\|Q) = \mathbb{E}_Q[f(u)] \geq f(\mathbb{E}_Q[u]) = f(1) = 0.$$

等号成立条件については、支持直線（supporting line）$c\,(u-1)$ として、f より 1 以外のすべての点で真に下側にあるものをとる。すると $D_f(P\|Q) \geq \mathbb{E}_Q[c\,(u-1)] = 0$ となり、その差は $f(u) - c\,(u-1)$ の期待値であって、これは $u \neq 1$ となるすべての点で真に正である。ゆえに $D_f(P\|Q)=0$ ならば、$u(x)=1$ がすべての x（ただし $q(x)>0$）で成り立ち、すなわち $P=Q$ となる。■

この証明の原動力となった恒等式 $\mathbb{E}_Q[u]=1$ からは、以降で繰り返し用いるもう 1 つの便利な帰結が得られる。すなわち、定数倍 $c\,(u-1)$ を f に加えても、$\mathbb{E}_Q[c\,(u-1)]=0$ であるため、いかなるダイバージェンスも変化しない。たとえば関数 $u\log u$ と $u\log u - u + 1$ は、同じ KL ダイバージェンスを定める。

ここまでは $P \ll Q$ を仮定してきたが、冒頭の点質量（point mass）の例はこの条件を満たしていない。生成モデルもこれと同じ状況、すなわち生成サンプルとデータが互いに素な領域を占めた状態から訓練を始めるため、D_f を拡張して q の 0 を扱えるようにしなければならない。$p(x)=q(x)=0$ となる結果からの寄与は 0 とする。$p(x)=0<q(x)$ となる点での寄与は、すでに定義にあるとおり単に $q(x)\,f(0)$ である。$q(x)=0<p(x)$ となる点では比が発散するため、その寄与を $q\,f(p/q)$ の $q \downarrow 0$ における極限として定義する。$t=p/q$ と置けばこれは $p\,f(t)/t$ と書き直され、その極限は $p(x)\,f'(\infty)$（ただし $f'(\infty) := \lim_{t\to\infty} f(t)/t$）となる。凸関数 f に対しては、割線の傾き $f(t)/t$ と導関数 $f'(t)$ が無限遠で同じ極限を持つため、この記法が正当化される。$f(0)$ と $f'(\infty)$ はいずれも $+\infty$ の値をとり得る。

ここで、この規則を点質量 δ_0 と δ_d（ただし $d \neq 0$ は任意）に適用しよう。$x=0$ では $q=0<p=1$ であり、$x=d$ では $p=0<q=1$ であるから、次を得る。

$$D_f(\delta_0\|\delta_d) = f(0) + f'(\infty) \qquad \text{for every } d \neq 0.$$

右辺から d が消えていることに注目してほしい。この式からは、Table 37.2 単独で見たものよりはるかに強い事実がわかる。すなわち、**あらゆる** f ダイバージェンスは互いに素な台の上で定数になる。ニアミスと大きな外れを区別できないため、それを損失として訓練されるモデルは、台どうしを隔てる距離から勾配を一切受け取れないのである。KL ダイバージェンスの場合、$f(u)=u\log u$ から $f(0)=0$ および $f'(\infty)=\lim_{t\to\infty}\log t=\infty$ が得られ、Table 37.2 の 1 行目が再現される。値が凍りついた残りの 2 行も、それぞれのダイバージェンス関数を書き下せば直ちに導かれる。

37.3.1.3 代表的なダイバージェンス関数

Table 37.3 の各行は f の 1 つの選択に対応しており、Figure 37.7 にはそのうち 6 つの曲線を並べて描いている。列 $f''(1)$ は $u=1$ における曲率（curvature）を表しており、次項で中心的な役割を果たす。

Table 37.3: 代表的なダイバージェンス関数。各行は 1 つのダイバージェンスの関数 f、$u=1$ における曲率、そのダイバージェンスが対称か、およびそれ自体（またはその平方根）が距離関数であるかを示す。

ダイバージェンス	$f(u)$	$f''(1)$	対称か	距離関数か
カルバック・ライブラー（順方向）	$u\log u$	1	いいえ	いいえ
逆方向 KL	$-\log u$	1	いいえ	いいえ
ピアソン χ^2	$(u-1)^2$	2	いいえ	いいえ
二乗ヘリンガー d_{H}^2	$(\sqrt{u}-1)^2$	$\frac{1}{2}$	はい	d_{H} が距離関数
全変動	$\frac{1}{2}\lvert u-1\rvert$	なし（角）	はい	はい
イェンセン・シャノン	$\frac{u}{2}\log u - \frac{u+1}{2}\log\frac{u+1}{2}$	$\frac{1}{4}$	はい	$\sqrt{\mathrm{JS}}$ が距離関数
α ダイバージェンス（$\alpha\neq 0,1$）	$\dfrac{u^{\alpha}-\alpha u+\alpha-1}{\alpha(\alpha-1)}$	1	$\alpha=\frac{1}{2}$ のみ	$\alpha=\frac{1}{2}$ で平方根が距離関数

逆方向 KL（reverse KL）は $f(u)=-\log u$ を用いる。というのも、$\mathbb{E}_Q[-\log(p/q)] = \mathbb{E}_Q[\log(q/p)] = D_{\mathrm{KL}}(Q\|P)$ が成り立つからである。したがって、KL ダイバージェンスの引数を入れ替えることは単に別の f を選ぶことにほかならず、両方の向きが同じ表の中に並んでいる。任意の f ダイバージェンスの引数を入れ替える一般的な手順は、演習問題 1a で扱う。

全変動（total variation）は $f(u)=\frac{1}{2}|u-1|$ を用い、これを整理すると $\sum_x q(x)\,\frac{1}{2}|p(x)/q(x)-1| = \frac{1}{2}\sum_x |p(x)-q(x)|$ となる。これを $\mathrm{TV}(P,Q)$ と書く。確率ベクトル間の ℓ_1 距離の半分にあたり、通しで用いている分布の組では $\mathrm{TV}=\frac{1}{2}(0.4+0.2+0.2)=0.4$ となる。この数値が P と Q の識別において何を意味するかは、37.3.3 節で詳しく見る。点質量に対しては、このダイバージェンス関数から $f(0)+f'(\infty)=\frac{1}{2}+\frac{1}{2}=1$ が得られ、Table 37.2 の 3 行目が説明される。

ピアソンの χ^2 ダイバージェンスは放物線 $f(u)=(u-1)^2$ を用い、$\chi^2(P\|Q) = \sum_x q(x)\,(p(x)/q(x)-1)^2 = \sum_x (p(x)-q(x))^2/q(x)$ を与える。通しで用いている分布の組では、その値は $0.16/0.2+0.04/0.5+0.04/0.3=1.0133$ となる。Q が小さい箇所で不一致がどれほど大きく重み付けされるかに注目しよう。$q=0.2$ である第 1 の結果だけで、全体の 0.8 を占めている。

イェンセン・シャノン・ダイバージェンス (Lin 1991) は、両方の分布をその均等な混合分布 $M=\frac{1}{2}(P+Q)$（確率質量関数を m とする）と比較することで KL ダイバージェンスを対称化する。

$$\mathrm{JS}(P,Q)=\frac{1}{2}D_{\mathrm{KL}}(P\,\|\,M)+\frac{1}{2}D_{\mathrm{KL}}(Q\,\|\,M). \tag{37.27}$$

M は P と Q の少なくとも一方が質量を置くすべての点に質量を置くため、比 p/m と q/m が発散することはない。実際、$p/m=2p/(p+q)\le 2$ であるから、各 KL 項は高々 $\log 2$ である。したがって JS は対称かつ有界で $0\le \mathrm{JS}(P,Q)\le \log 2$ を満たし、上限 $\log 2$ に達するのは

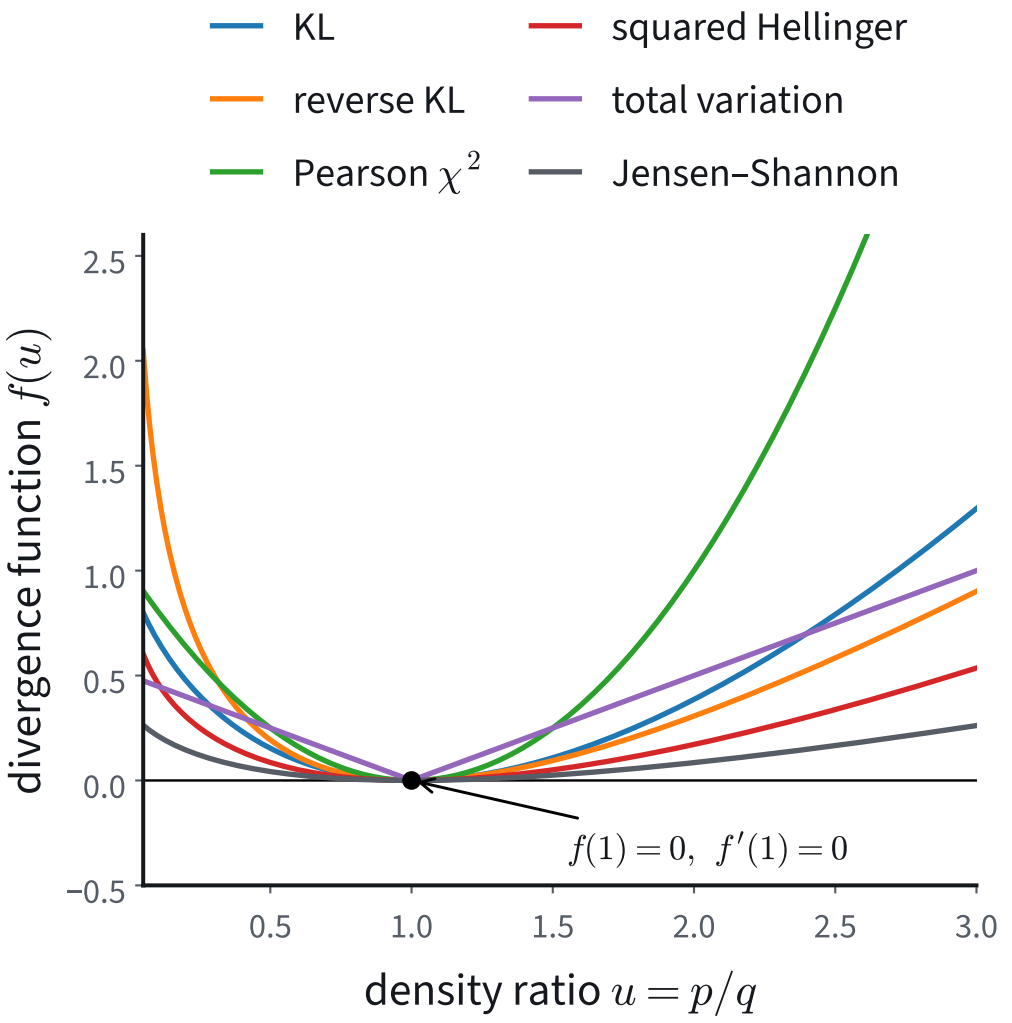


Figure 37.7: 6 つのダイバージェンス関数。滑らかな曲線はそれぞれダイバージェンスを変えない範囲で $f'(1)(u-1)$ だけずらしてあり、すべて $u=1$ で 0 に接している。$u=1$ の近傍では滑らかな曲線は表に示した曲率 $1, 1, 2, \frac{1}{2}, \frac{1}{4}$ を持つ放物線になる一方、全変動は鋭い角を持つ。$u=1$ から離れると不一致に対するペナルティの課し方は大きく異なり、逆方向 KL は $u \to 0$ で発散し、KL と χ^2 は超線形に増大し、全変動は線形に増大する。

P と Q が互いに素な台を持つ場合に限られる（このとき $p/m = 2$ が $p > 0$ なるすべての点で成り立つ）。驚くべきことに、その平方根は真の距離関数になる (Endres and Schindelin 2003)。Equation 37.27 を比 u を用いて書き直すと表のダイバージェンス関数が得られ（演習問題 1c）、これは $f(0) = f'(\infty) = \frac{1}{2}\log 2$ を満たす。その和は $\log 2$ であり、Table 37.2 の 2 行目に一致する。こうして、ただ 1 つの恒等式 $D_f(\delta_0 \| \delta_d) = f(0) + f'(\infty)$ によって、値が凍りついた 3 つの行すべてが説明された。

二乗ヘリンガーダイバージェンス（squared Hellinger divergence）は $d_{\mathrm{H}}^2(P,Q) = \sum_x \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2$ で定義される。その平方根 d_{H} はベクトル $\sqrt{p}$ と $\sqrt{q}$ の間のユークリッド距離にほかならないため、d_{H} は直ちに距離関数となる（演習問題 2b）。

表の最後の行は、この族の中に含まれる連続的な族である α **ダイバージェンス** (Amari 2016) であり、次の関数によって生成される。

$$f_\alpha(u) = \frac{u^\alpha - \alpha u + \alpha - 1}{\alpha(\alpha - 1)}. \tag{37.28}$$

すべての $\alpha \neq 0, 1$ に対して、この関数は $f_\alpha(1) = 0$ と曲率 $f_\alpha''(1) = 1$ を持つ凸関数である。$\alpha \to 1$ では $u\log u - u + 1$ に近づき、$\alpha \to 0$ では $-\log u + u - 1$ に近づく。$u - 1$ の定数倍を加えてもダイバージェンスは変わらないため、これら 2 つの極限から順方向 KL（forward KL）と逆方向 KL が復元される。両者の中間にあたる $\alpha = \frac{1}{2}$ では $2(\sqrt{u} - 1)^2$（二乗ヘリンガーの 2 倍）が得られ、$\alpha = 2$ では χ^2 の半分が得られる。したがって、ただ 1 つのパラメータ α によって、逆方向 KL からヘリンガー、順方向 KL、さらには χ^2 までが滑らかに補間される（演習問題 1d ではこの族をレーニイダイバージェンスと関連づける）。

表の最初の 6 行を、通しで用いている分布の組に対して両方の引数の順序で評価してみよう。KL ダイバージェンスは 0.396 nat と 0.365 nat となって 37.1 節の結果と一致し、TV と $\chi^2(P\|Q)$ も上の手計算と一致するはずである。

```
def f_divergence(f, p, q):
    """D_f(P||Q) = sum_x q(x) f(p(x)/q(x)); nats for KL/JS, else unitless."""
    return np.sum(q * f(p / q))

generators = {
    'KL':          lambda u: u * np.log(u),
    'reverse KL':  lambda u: -np.log(u),
    'chi^2':       lambda u: (u - 1) ** 2,
    'Hellinger^2': lambda u: (np.sqrt(u) - 1) ** 2,
    'TV':          lambda u: 0.5 * np.abs(u - 1),
    'JS':          lambda u: u / 2 * np.log(u) - (u + 1) / 2 * np.log((u + 1) / 2),
}

p = np.array([0.6, 0.3, 0.1])
q = np.array([0.2, 0.5, 0.3])
for name, f in generators.items():
    print(f'{name:12s} D_f(P||Q) = {f_divergence(f, p, q):.4f}   '
          f'D_f(Q||P) = {f_divergence(f, q, p):.4f}')
```

```
KL           D_f(P||Q) = 0.3961   D_f(Q||P) = 0.3653
reverse KL   D_f(P||Q) = 0.3653   D_f(Q||P) = 0.3961
chi^2        D_f(P||Q) = 1.0133   D_f(Q||P) = 0.8000
Hellinger^2  D_f(P||Q) = 0.1862   D_f(Q||P) = 0.1862
```

```
TV          D_f(P||Q) = 0.4000   D_f(Q||P) = 0.4000
JS          D_f(P||Q) = 0.0911   D_f(Q||P) = 0.0911
```

予想どおり KL の行は非対称であり、逆方向 KL の行は単に KL の行の 2 列を入れ替えたものになっている。χ^2 の行も非対称である（1.0133 対 0.8000）。これに対し、二乗ヘリンガー（0.1862）、TV（0.4）、JS はいずれも対称であり、JS は 0.0911 nat と上限 $\log 2 \approx 0.693$ を大きく下回っている。6 つのダイバージェンスが同じ分布の組を見て、0.09 から 1.01 まで 1 桁にわたる値を返したことになる。ところが、KL を 2 倍、二乗ヘリンガーを 4 倍、JS を 8 倍すると、スケールし直した 3 つの値は 0.79、0.74、0.73 となる。密度比のまったく異なる 3 つの関数がなぜほぼ一致するのか、そして係数 2、4、8 はどこから現れたのだろうか。

37.3.1.4 局所二次近似

Figure 37.7 の各曲線は、$u = 1$ から離れた領域ではまったく異なる振る舞いを見せる。しかし $u = 1$ の近傍では、定数倍の違いを除けばただ 1 つの関数へと重なり合う。これはおそらくこの族全体で最も驚くべき性質だろう。本節の以降の内容はこの項に依存しないため、初読時には読み飛ばしてもよい。

1 次項 $c\,(u-1)$ を加えてもいかなるダイバージェンスも変わらないため、値 $f(1) = 0$ も傾き $f'(1)$ も D_f には影響しない。滑らかなダイバージェンス関数どうしを区別する最初の性質は、曲率 $f''(1)$ である。$u = 1$ の近傍では、あらゆる滑らかな f は放物線 $\tfrac{1}{2}f''(1)(u-1)^2$ のように見え、$(u-1)^2$ の Q のもとでの期待値はまさに χ^2 ダイバージェンス $\mathbb{E}_Q[(u-1)^2] = \sum_x (p(x)-q(x))^2/q(x)$ にほかならない。そこで $r = \max_x |u(x)-1|$ により、密度比が 1 からどれだけ離れているかを測ることにする。

命題（局所二次近似）。 **$P \ll Q$、$0 < r < 1$ とし、f は $[1-r, 1+r]$ 上で 3 回微分可能でそこにおいて $|f'''| \le B$ を満たし、さらに $|u(x)-1| \le r$ がすべての x（ただし $q(x) > 0$）で成り立つとする。このとき、次が成り立つ。**

$$\left| D_f(P\|Q) - \frac{f''(1)}{2}\,\chi^2(P\|Q) \right| \le \frac{Br}{6}\,\chi^2(P\|Q). \tag{37.29}$$

証明。 $f(1) = 0$ より、剰余項付きのテイラーの定理から $f(u) = f'(1)(u-1) + \tfrac{1}{2}f''(1)(u-1)^2 + R(u)$（ただし $|R(u)| \le B|u-1|^3/6$）が得られる。項ごとに期待値 $\mathbb{E}_Q$ をとる。1 次項は $\mathbb{E}_Q[u-1] = 0$ により消えるため、異なる f の間の 1 次の違いはすべて打ち消される。2 次項は $\tfrac{1}{2}f''(1)\,\chi^2(P\|Q)$ となり、剰余項は $\mathbb{E}_Q|R| \le \tfrac{B}{6}\,\mathbb{E}_Q|u-1|^3 \le \tfrac{Br}{6}\,\mathbb{E}_Q(u-1)^2 = \tfrac{Br}{6}\,\chi^2$ を満たす。$\blacksquare$

要するに、$D_f \approx \tfrac{1}{2}f''(1)\,\chi^2$ が r に比例する相対誤差のもとで成り立つ。すなわち、P が Q に近いときは常に、滑らかな f ダイバージェンスはすべて互いにスカラー倍の関係にある。先ほど KL、二乗ヘリンガー、JS に掛けた係数 2、4、8 は単に $2/f''(1)$ であり、3 つの積がいずれも近似していた対象は $\chi^2(P\|Q) = 1.013$ である。通しで用いている分布の組ではこれらはまだ 4 分の 1 ほど小さいが、それも無理はない。最大の密度比が 3 であるため $r = 2$ となり、命題の仮定 $r < 1$ の外にあるからである。二次近似が精密になっていく様子を観察するため、P を Q に向かって線分 $P_t = Q + t\,(P-Q)$ に沿って動かしてみよう。するとすべての $u-1$ が t 倍され、$r = 2t$ となる。4 つの滑らかなダイバージェンスについて比 $D_f(P_t\|Q)/(\tfrac{1}{2}f''(1)\,\chi^2(P_t\|Q))$ を出力すると、この比は 1 に向かって t に比例する速さで収束するはずである。

```
    curvature = {'KL': 1.0, 'reverse KL': 1.0, 'Hellinger^2': 0.5, 'JS': 0.25}
    for t in (0.5, 0.1, 0.01):
        p_t = q + t * (p - q)                    # moves P_t toward Q as t -> 0
        chi2 = f_divergence(generators['chi^2'], p_t, q)
        ratios = [f_divergence(generators[name], p_t, q) / (c / 2 * chi2)
                  for name, c in curvature.items()]
        print(f't = {t:4}: ' + '  '.join(f'{name} {r:.3f}'
                                         for name, r in zip(curvature, ratios)))
```

```
t =  0.5: KL 0.844  reverse KL 0.747  Hellinger^2 0.789  JS 0.783
t =  0.1: KL 0.956  reverse KL 0.917  Hellinger^2 0.936  JS 0.935
t = 0.01: KL 0.995  reverse KL 0.990  Hellinger^2 0.993  JS 0.993
```

$t = 0.5$ では、4 つのダイバージェンスはそれぞれの二次近似のおよそ 75% から 85% に位置し、$t = 0.1$ では 91% を超え、$t = 0.01$ では 99% を超える。最後の 2 行の間では t が 10 分の 1 に減少し、1 からの不足分もそれぞれほぼ 10 分の 1 に減少しており、命題の 1 次因子 r と整合している。上界自体は控えめな評価になっている。たとえば $t = 0.1$ における KL では $r = 0.2$ かつ $|f'''(u)| = 1/u^2 \le 1/0.8^2$ であり、上界が許容する相対誤差は約 10% であるのに対し、実際の差はわずか 4% にとどまる。最適化における教訓は実践的である。滑らかな f ダイバージェンスどうしが実質的な違いを見せるのは遠く離れた分布に対してのみであり、それはモデルがまだデータから遠い訓練初期の領域なのである。

パラメトリックなモデル族の内部では、この共通の局所二次形式を支配するのは 36.6 節のフィッシャー情報行列（Fisher information matrix）$\mathbf{I}(\theta) = \mathbb{E}\big[\nabla_\theta \log q_\theta \, \nabla_\theta \log q_\theta^\top\big]$ である。微小なパラメータの変位 $\mathbf{h}$ に対し、$q_{\theta+\mathbf{h}}$ の q_θ に対する密度比は $u \approx 1 + \mathbf{h}^\top \nabla_\theta \log q_\theta$ と展開される。したがって $\chi^2(q_{\theta+\mathbf{h}} \| q_\theta) = \mathbb{E}[(u-1)^2] \approx \mathbf{h}^\top \mathbf{I}(\theta)\, \mathbf{h}$ となり、あらゆる滑らかな D_f はヘッセ行列 $f''(1)\,\mathbf{I}(\theta)$ を $\mathbf{h} = \mathbf{0}$ において持つ。35.5 節の自然勾配（natural gradient）は通常 KL ダイバージェンスから動機づけられるが、どの滑らかな f ダイバージェンスのもとでも同じ更新則を与えることになる。表の中で唯一の例外は全変動である。そのダイバージェンス関数は $u = 1$ に角を持っており二階導関数が存在しないため、それによって挙動がどう変わるかは演習問題 4b で P が Q に近づく極限を通じて調べる。

37.3.2 順方向 KL と逆方向 KL

本書の大半のモデルの背後にある訓練基準である最尤法は、データを第 1 引数に置いた KL ダイバージェンスを最小化する。これに対し、変分推論はモデルを第 1 引数に置いた KL ダイバージェンスを最小化する。これら両方の順序は Table 37.3 に別々の行として現れており、モデル Q_θ（密度を q_θ とする）をモデル族の外にある目標分布 P に当てはめるとき、モデルを正反対の方向へと押しやる。

- **順方向 KL**、すなわち $D_{\mathrm{KL}}(P \,\|\, Q_\theta) = \mathbb{E}_P[\log(p/q_\theta)]$ は、**真の分布**のもとで期待値をとる。$q_\theta(x) \to 0$ が $p(x) > 0$ なる点で生じると常に、ダイバージェンスは発散する。したがって順方向 KL は**ゼロ回避型**（zero-avoiding）、すなわち**モード被覆型**（mode-covering）である。モデルはデータのすべてのモード、つまり p が集中するあらゆる領域を覆うように広がらなければならず、たとえそのために P が決して訪れない空の領域に質量をこぼすことになっても避けられない。
- **逆方向 KL**、すなわち $D_{\mathrm{KL}}(Q_\theta \,\|\, P) = \mathbb{E}_{Q_\theta}[\log(q_\theta/p)]$ は、**モデル**のもとで期待値をとる。ここでは、真の分布が質量を持たない領域にモデルが質量を置くと重いペナルティが課される一方、モデルが無視したモードは目的関数に一切寄与しない。したがって逆方向 KL は**ゼロ強制型**（zero-forcing）、すなわち**モード探索型**（mode-seeking）であり、モデルは目標分布のうちうまく説明できる部分に集中する。

実際には、どちらの向きを選ぶかは、何が実際に計算できるかによって決まることが多い。順方向 KL には P からのサンプルとモデルの密度が必要であり、その最小化は最尤法にほかならない。36.6.2.1 節で示したように、負の対数尤度（negative log-likelihood）の平均は経験分布（empirical distribution）からモデルへの交差エントロピーに等しく、これは順方向 KL と θ に依らない定数だけしか違わない。一方、逆方向 KL にはモデルからのサンプルと目標分布の密度が必要であり、目標密度が正規化定数（normalizing constant）を除いてしかわからない場合でも機能する。変分推論（36.7 節）はまさにこの設定にあり、その目標分布 P は正規化定数の計算が困難な事後分布（posterior）であるため、$D_{\mathrm{KL}}(Q_\theta \| P)$ を計算可能な近似分布 Q_θ にわたって最小化する。

しかし、両方の分布でサンプルと密度がともに利用できる場合には、向きの選択は本質的な設計上の判断となる。教師モデルから蒸留される生徒ネットワーク（26.1 節（英語版））は、教師を第 1 引数に置くと教師が出力し得るあらゆる候補を覆うように広がり、生徒を第 1 引数に置くと教師の最も尤もらしい出力に集中する。同様に、15.3 節の KL ペナルティは学習される方策（policy）を第 1 引数に置くことで、参照方策が決してとらない行動から遠ざけている。

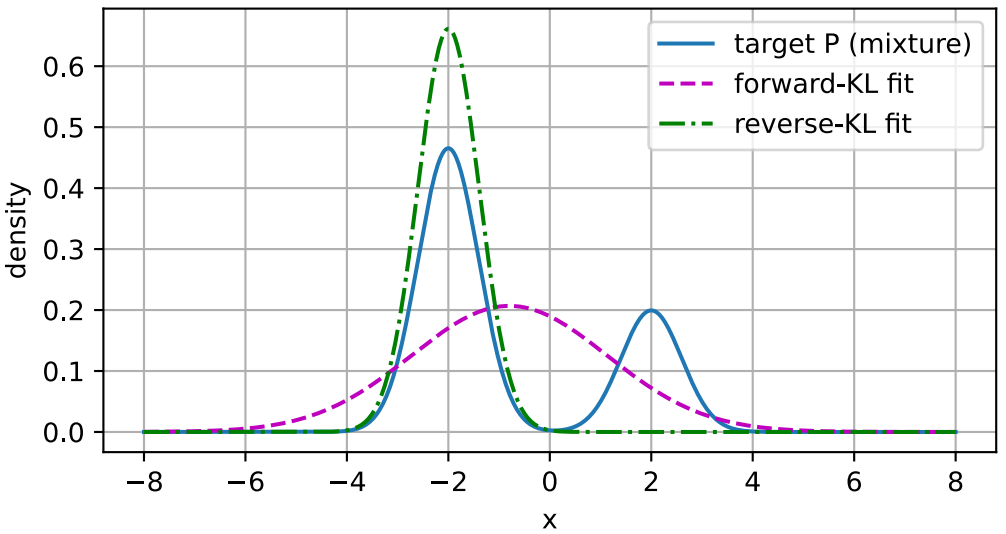
モデル族が P を含んでいれば、どちらの向きでも P が復元される。両者の道が分かれるのはモデルが**誤特定**（misspecified）されているときであり、それを最も明快に示す例が二峰性混合分布への単一ガウス分布の当てはめである。順方向 KL の場合、最適化すら必要ない。順方向 KL は定数から $\mathbb{E}_P[\log q(X)]$ を引いたものに等しいため、その最小点（minimizer）は経験分布を P に置き換えた最尤法によるガウス分布となり、指数分布族（exponential family）における最尤法は十分統計量（sufficient statistics）の期待値を一致させる（36.4.3.2 節）。すなわち、分散が有限である限り $\mu^* = \mathbb{E}_P[X]$ および $(\sigma^*)^2 = \mathrm{Var}_P(X)$ となる。逆方向 KL にはこのような閉形式解（closed-form solution）はなく、各モードの近くに別々の極小値を持つことがある。

二峰性混合分布 $P = 0.7\,\mathcal{N}(-2, 0.6^2) + 0.3\,\mathcal{N}(2, 0.6^2)$ に対しては、順方向 KL による当てはめ結果をあらかじめ計算できる。その平均は $0.7(-2) + 0.3(2) = -0.8$ である。分散は成分内分散と成分間分散の和であり、後者は重み w_1、w_2 と平均間の距離 Δ に対して $w_1 w_2 \Delta^2$ に等しいため、$0.36 + 0.7 \cdot 0.3 \cdot 4^2 = 3.72$ となる。逆方向 KL による当てはめについては、導関数を用いないオプティマイザを $\theta = (\mu, \log\sigma)$ 上で原点と小さい方のモードの近くという 2 つの初期点から走らせ、各 KL をグリッド上の数値積分によって評価する。次のセルでは、それぞれの当てはめについて平均、標準偏差、および両方向のダイバージェンスを出力する。

```
w, mus, sigmas = np.array([0.7, 0.3]), np.array([-2.0, 2.0]), np.array([0.6, 0.6])
x = np.linspace(-8.0, 8.0, 4001)

def log_components(x, w, mus, sigmas):
    """log(w_k N(x; mu_k, sigma_k^2)) for every component k; shape (k, len(x))."""
    return (np.log(w)[:, None] - 0.5 * np.log(2 * np.pi * sigmas[:, None] ** 2)
            - (x - mus[:, None]) ** 2 / (2 * sigmas[:, None] ** 2))

def log_mix(x, w, mus, sigmas):
    """Log-density of the mixture sum_k w_k N(mu_k, sigma_k^2), computed stably."""
    c = log_components(x, w, mus, sigmas)
    c_max = c.max(axis=0)
    return c_max + np.log(np.exp(c - c_max).sum(axis=0))

log_p = lambda x: log_mix(x, w, mus, sigmas)

def log_q(x, theta):
    mu, log_sigma = theta
    return (-0.5 * np.log(2 * np.pi) - log_sigma
```

```
            - (x - mu) ** 2 / (2 * np.exp(2 * log_sigma)))

def kl(log_a, log_b, x):
    """D_KL(A||B) by quadrature on the grid x; in nats."""
    return integrate.trapezoid(np.exp(log_a) * (log_a - log_b), x)

fwd = lambda th: kl(log_p(x), log_q(x, th), x)     # D_KL(P || Q_theta)
rev = lambda th: kl(log_q(x, th), log_p(x), x)     # D_KL(Q_theta || P)

th_fwd = optimize.minimize(fwd, x0=[0.0, 0.0], method='Nelder-Mead').x
th_rev = optimize.minimize(rev, x0=[0.0, 0.0], method='Nelder-Mead').x
th_rev2 = optimize.minimize(rev, x0=[2.5, -0.5], method='Nelder-Mead').x

print(f'{"fit":26s}{"mu":>8s}{"sigma":>7s}'
      f'{"forward KL":>12s}{"reverse KL":>12s}')
for name, th in [('forward-KL fit', th_fwd), ('reverse-KL fit', th_rev),
                 ('reverse-KL fit, 2nd start', th_rev2)]:
    print(f'{name:26s}{th[0]:+8.3f}{np.exp(th[1]):7.3f}'
          f'{fwd(th):12.3f}{rev(th):12.3f}')

d2l.plot(x, [np.exp(log_p(x)), np.exp(log_q(x, th_fwd)),
             np.exp(log_q(x, th_rev))], 'x', 'density',
         legend=['target P (mixture)', 'forward-KL fit', 'reverse-KL fit'],
         figsize=(6, 3))
```

```
fit                           mu  sigma  forward KL  reverse KL
forward-KL fit              -0.800  1.929       0.558       1.274
reverse-KL fit              -1.998  0.603       5.981       0.356
reverse-KL fit, 2nd start   +1.995  0.607      14.534       1.202
```



順方向 KL による当てはめは手計算と一致し、$\mu = -0.800$、$\sigma = 1.929 \approx \sqrt{3.72}$ となる。これは両方のモードにまたがる幅広い釣鐘曲線であり、P がほとんど質量を持たないモード間の谷にもかなりの質量を置いている。1 つ目の逆方向 KL による当てはめは $\mu = -1.998$、$\sigma = 0.603$ に

落ち着き、主要な成分にぴったりと寄り添う一方で小さい方の成分にはほとんど質量を残さない。その逆方向 KL である 0.356 nat は $\log(1/0.7) \approx 0.357$ に近く、これは 70% の成分をあたかも分布全体であるかのように扱うコストにあたる。小さい方のモードの近くから出発すると、オプティマイザは代わりに $\mu = +1.995$ に収束し、その逆方向 KL は 1.202 nat と $\log(1/0.3) \approx 1.204$ に近くなる。したがって、変分近似がどのモードを捉えるかは初期化に依存する。

逆方向 KL はモデルのもとで平均をとるため、モデルが訪れないモードはそもそも一切検査されない。目的関数が気づくのは、選ばれたモードの重みが 0.7（または 0.3）であって 1 ではないということだけである。変分事後分布が不確実性を過小評価しやすいことで知られる理由はまさにここにあり、この例でも逆方向 KL による当てはめの標準偏差は 0.60 であるのに対し、目標分布の標準偏差は 1.93 である。

最後の 2 列は、同一のモデルに対して 2 つの基準がどれほど鋭く対立するかを浮き彫りにしている。最尤法の基準である順方向 KL は、主要モードへの当てはめに 5.98 nat（幅広い当てはめの 0.558 nat の 10 倍以上）、小さい方のモードへの当てはめに 14.5 nat というペナルティを課す。いずれの場合も、取りこぼされたモードはモデルの平均から標準偏差の 6 倍以上離れた位置にあり、そこでは $\log q$ が大きな負の値をとるからである（演習問題 3c）。これに対し、逆方向 KL は最初の 2 つの当てはめに 0.356 nat 対 1.27 と逆の順序をつける。逆方向 KL が許容できないのは、幅広い当てはめがモード間の空の谷に置いた質量なのである。

密度が 0 になる場合の境界規則は、この対比を極限の形で見事に捉えている。順方向 KL では $f(u) = u\log u$ が $f'(\infty) = \infty$ を持つため、P が正の質量を持つ領域にモデルが質量を 0 しか置かないと無限大のコストがかかる一方、$f(0) = 0$ であることは、P が質量を持たない領域にモデルが質量を置いても境界項が一切加わらないことを意味する。逆方向 KL である $f(u) = -\log u$ では、これら 2 つの値がそのまま入れ替わる。

37.3.3 全変動、ピンスカーの不等式、データ処理

あるダイバージェンスが 0.4 に等しいとわかっても、その数値が実際に何を保証するのかを知らなければ、それ自体から読み取れることはほとんどない。全変動についてはその操作的な意味が完全に明快であり、ピンスカーの不等式はその意味の多くを KL ダイバージェンスへと引き継ぐ。

37.3.3.1 全変動

ダイバージェンス関数 $\frac{1}{2}|u-1|$ は Table 37.3 の中で最も単純な項目であり、比の 1 からの絶対距離だけにペナルティを課す。これに対応するダイバージェンスは ℓ_1 距離（p と q の間の距離）の半分であるが、その数値を直ちに解釈可能にする第 2 の同値な表現を持つ。すなわち全変動とは、P と Q が任意の事象 A に割り当てる確率の差の最大値である。

$$\mathrm{TV}(P,Q) = \frac{1}{2}\sum_{x}|p(x)-q(x)| = \sup_{A}|P(A)-Q(A)|. \tag{37.30}$$

もし $\mathrm{TV}(P,Q) = 0.03$ であれば、いかなる事象をもってしても 2 つの分布の確率に 3 パーセントポイントを超える差をつけることはできない。この上限を達成するのは事象 $A^\star = \{x : p(x) > q(x)\}$、すなわち P が Q よりも高い確率を与える結果の集合である。これを確かめるために任意の事象 A をとると、$P(A)-Q(A) = \sum_{x\in A}(p(x)-q(x)) \le \sum_{x\in A^\star}(p(x)-q(x))$ が成り立つ。A から $A^\star$ へ切り替えても、非負の項が加わり非正の項が除かれるだけだからである。さらに $\sum_x (p(x)-q(x)) = 0$ より、$p-q$ の正の部分と負の部分は全体の大きさが等しいため、右辺は $\frac{1}{2}\sum_x|p(x)-q(x)|$ に等しくなり、同じ上界が $Q(A)-P(A)$ にも当てはまる。

通しで用いている分布の組では、$A^\star$ は第 1 の結果のみからなり、先ほど求めたとおり TV $= 0.6 - 0.2 = 0.4$ となる。2 つの連続密度に対する同じ幾何学的構造を Figure 37.8 に示す。全変動は真の距離関数でもある。どちらの表現からも対称性は明らかであり、三角不等式は ℓ_1 の式から直ちに従う（演習問題 2a）。

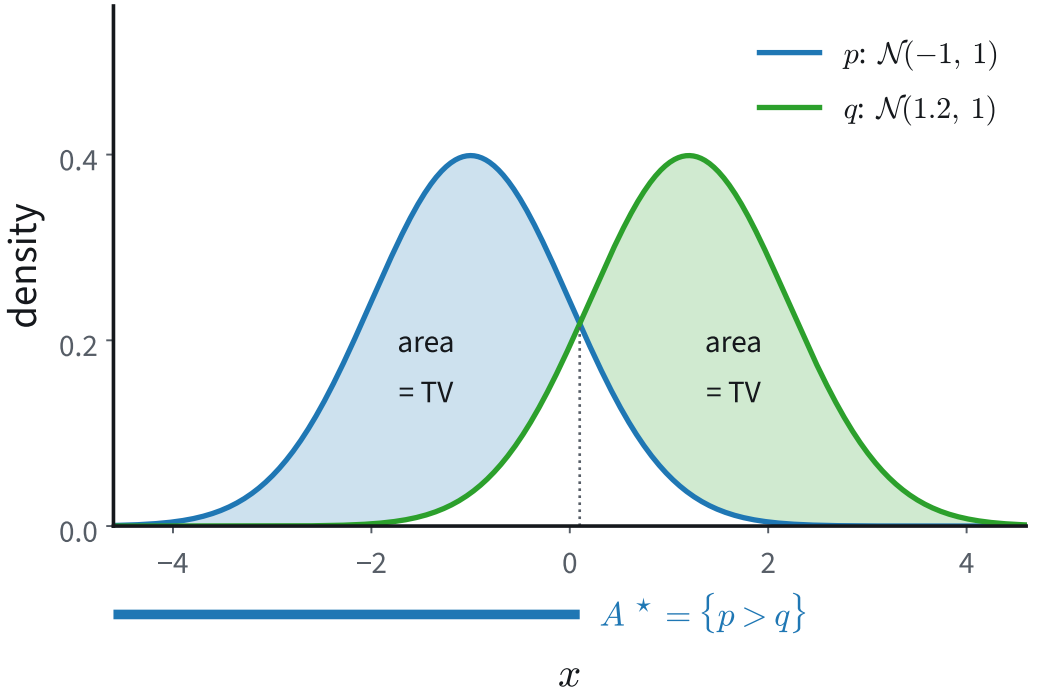


Figure 37.8: 等分散の 2 つの密度とそれらが食い違う領域。$p > q$ となる領域と $q > p$ となる領域は面積が等しく、その共通の面積が全変動距離である。上限を達成する事象は $A^\star = \{p > q\}$（ここでは交点より左の半直線）であり、そこでは P、それ以外では Q と判定するのが最適な 1 サンプル検定となる。

上限による表現は、TV に具体的な統計的意味を与える。等確率 $\frac{1}{2}$ で P または Q から抽出された 1 個のサンプルを検定者に渡し、どちらの分布から生成されたかを当てさせる場面を想像しよう。任意の決定論的な検定は、「P」と答える領域 A を定め、それ以外では「Q」と答えるものであり、次の確率で正解する。

$$\tfrac{1}{2}P(A) + \tfrac{1}{2}(1 - Q(A)) = \tfrac{1}{2}(1 + P(A) - Q(A)) \le \tfrac{1}{2}(1 + \mathrm{TV}(P, Q)),$$

ここで等号は $A = A^\star$ のときに成り立つ。あらゆる検定はなんらかの A に対するこの形をとるため（ランダム化検定は決定論的な検定の平均をとるにすぎず、それを上回ることはできない）、最適な検定者であっても公平なコイン投げより高々 $\mathrm{TV}/2$ だけ高い確率でしか正解できない。通しで用いている分布の組では、最適な検定は第 1 の結果に対して「P」と答え、70% の確率で正解する。GAN の識別器（discriminator、実データと生成サンプルを見分けようとするもの）も機械生成テキストの検出器もこの種の検定者であり、全変動は 1 サンプルから達成できる正解率に厳格な上限を課す。

37.3.3.2 ピンスカーの不等式

全変動はあらゆる統計的検定の限界を与える一方、ニューラルネットワークの損失は対数確率（log-probability）から構築されるため、実際に計算したり制御したりできるのは通常 KL ダイバージェンスである。ピンスカーの不等式はこの両者を橋渡しする。すなわち、KL ダイバージェンスが小さければ、いかなる検定も P と Q を確実に見分けることはできないと保証される。15.2 節における方策の信頼領域（trust region）はその典型例であり、単調改善の保証は全変動を用いて証明されるが、実装上は KL に対する制約を通じて強制される。

命題（ピンスカーの不等式）。 (Pinsker 1964; Csiszár 1967) **有限集合または可算集合上の分布 P と Q に対し、次が成り立つ。**

$$\mathrm{TV}(P,Q) \;\le\; \sqrt{\tfrac{1}{2}\, D_{\mathrm{KL}}(P\|Q)}. \tag{37.31}$$

証明。ステップ 1：2 値の場合。 $\mathrm{Ber}(a)$ で表の出る確率が a のコインを表す。$a \in [0,1]$ および $b \in (0,1)$ に対し、$D_{\mathrm{KL}}(\mathrm{Ber}(a)\|\mathrm{Ber}(b)) = a\log\frac{a}{b} + (1-a)\log\frac{1-a}{1-b}$ かつ $\mathrm{TV}(\mathrm{Ber}(a),\mathrm{Ber}(b)) = |a-b|$ である。したがって、$\rho(a) = D_{\mathrm{KL}}(\mathrm{Ber}(a)\|\mathrm{Ber}(b)) - 2(a-b)^2$ が非負であることを示せばよい。b を固定すると、明らかに $\rho(b)=0$ であり、1 階導関数 $\rho'(a) = \log\frac{a}{b} - \log\frac{1-a}{1-b} - 4(a-b)$ も $a=b$ で 0 になり、$0<a<1$ における二階導関数は次式となる。

$$\rho''(a) = \frac{1}{a(1-a)} - 4 \ge 0,$$

これは $a(1-a) \le \frac{1}{4}$ による。$a=b$ において値と傾きの両方が 0 になる凸関数は $(0,1)$ 上のいたるところで非負であり、連続性（$0\log 0 = 0$ とする）により端点 $a=0$ および $a=1$ においても非負である。b が 0 または 1 のときは、$a=b$ でない限り KL ダイバージェンスが無限大になるため、不等式は自明に成り立つ。

ステップ 2：2 値への帰着。 $A^\star = \{p > q\}$ とし、$a = P(A^\star)$、$b = Q(A^\star)$ と置くと $\mathrm{TV}(P,Q) = a - b$ となる。$A^\star$ 内のすべての結果を 1 つのバケットにまとめ、$A^\star$ 外のすべての結果をもう 1 つのバケットにまとめることで標本空間を粗視化すると、KL ダイバージェンスは**減少**する（または変わらない）ことしか起こり得ない。これは対数和不等式（log-sum inequality）から従う。すなわち、非負の数 v_i と w_i に対し、

$$\sum_i v_i \log\frac{v_i}{w_i} \;\ge\; \Big(\sum_i v_i\Big)\log\frac{\sum_i v_i}{\sum_i w_i},$$

が成り立つ。これ自体、イェンセンの不等式を $t \mapsto t\log t$ に 1 回適用したものである。これを $A^\star$ とその補集合にそれぞれ適用すると、

$$D_{\mathrm{KL}}(P\|Q) \;\ge\; D_{\mathrm{KL}}(\mathrm{Ber}(a)\|\mathrm{Ber}(b)) \;\ge\; 2(a-b)^2 = 2\,\mathrm{TV}(P,Q)^2,$$

が得られ、これを整理すれば Equation 37.31 となる。$\blacksquare$

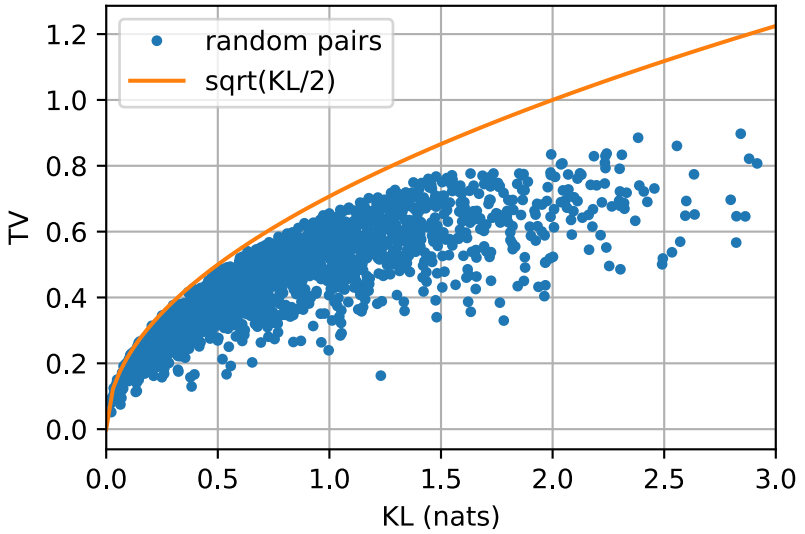
通しで用いている分布の組では、ピンスカーの不等式は $0.4 \le \sqrt{0.396/2} = 0.445$ となり、ステップ 2 を手計算で追うことができる。第 2 と第 3 の結果を統合すると、この組は $(0.6, 0.4)$ と $(0.2, 0.8)$ という 2 枚のコインに縮約される。両者の TV 距離は 0.4 のままであり、KL ダイバージェンスは 0.396 から $0.6\log 3 + 0.4\log\frac{1}{2} = 0.382$ nat へと減少し、ステップ 1 はそれを下から $2\,(0.4)^2 = 0.32$ で評価する。

偏り 0.6 のコインを公平なコインと比較すると、KL は $0.6\log 1.2 + 0.4\log 0.8 = 0.0201$ nat であるのに対し、$2\,(0.1)^2 = 0.02$ となる。つまり、ほぼ等しいコインどうしではピンスカーの不等式はほぼ等号に近い。しかし決定的に重要な点として、有用な逆向きの不等式は存在しない。まれにしか起きないたった 1 つの結果に $q = 0$ を置くモデルは、たとえ TV 距離がごくわずかであっても無限大の KL を持つ。また、TV は決して 1 を超えないため、大きく離れた分布に対してもこの上界は自明なものになる。たとえば平均が 50 だけ離れた分散 1 の 2 つのガウス分布は TV ≈ 1 であるのに対し、Equation 37.25 から KL $= 50^2/2 = 1250$ nat となる。

典型的な分布に対してピンスカーの上界はどれほど精密であり、また定数 $\frac{1}{2}$ を改良することはできるだろうか。第 1 の問いを確かめるため、単体上から一様に $10{,}000$ 組の 5 値分布のペアをサンプリングする。第 2 の問いを確かめるため、2 枚のコインを $a = b = \frac{1}{2}$ の周りで互いに近づける。ステップ 1 はそこでは 2 次の項まで一切の緩みを持たなかったため、比 $\mathrm{TV}/\sqrt{\mathrm{KL}/2}$ は 1 まで登りつめるはずである。

```
rng = np.random.default_rng()
pr, qr = rng.dirichlet(np.ones(5), 10000), rng.dirichlet(np.ones(5), 10000)
tv = 0.5 * np.abs(pr - qr).sum(axis=1)
kl_pq = np.sum(pr * np.log(pr / qr), axis=1)
ratio = tv / np.sqrt(kl_pq / 2)
print(f'TV / sqrt(KL/2) over 10,000 random pairs: median {np.median(ratio):.2f}, '
      f'max {ratio.max():.3f}')
for delta in (0.1, 0.01, 0.001):
    a, b = np.array([0.5, 0.5]), np.array([0.5 + delta, 0.5 - delta])
    print(f'coins 1/2 vs 1/2 + {delta:5}: TV / sqrt(KL/2) = '
          f'{delta / np.sqrt(kl_div(a, b) / 2):.6f}')
grid = np.linspace(0, 3, 100)
d2l.plot([kl_pq[:2000], grid], [tv[:2000], np.sqrt(grid / 2)], 'KL (nats)',
         'TV', legend=['random pairs', 'sqrt(KL/2)'], fmts=('.', '-'),
         xlim=[0, 3], figsize=(4.5, 3))
```

```
TV / sqrt(KL/2) over 10,000 random pairs: median 0.78, max 0.986
coins 1/2 vs 1/2 +   0.1: TV / sqrt(KL/2) = 0.989881
coins 1/2 vs 1/2 +  0.01: TV / sqrt(KL/2) = 0.999900
coins 1/2 vs 1/2 + 0.001: TV / sqrt(KL/2) = 0.999999
```



散布図のすべての点は曲線 $\sqrt{\mathrm{KL}/2}$ の下側に収まっている。中央値にあたるペアは上界の 0.8 付近に位置し、10,000 組の無作為なペアのうち最も精密なものは 1 から数パーセント以内に迫る。KL > 2 になると曲線は 1 を超え、上界は自明になる。ほぼ公平なコインに対しては、比は 0.9899、0.9999、0.999999 と上昇し、定数 $\frac{1}{2}$ をこれ以上厳しくできないことが確認できる（演習問題 4c）。Pinsker の元の証明では定数がこれより弱く、最良の係数 $\frac{1}{2}$ は Csiszár (1967)、およびそれとは独立に Kullback と Kemperman によって得られた。

37.3.3.3 データ処理

特徴抽出器や量子化器、あるいはその他の後処理ステップによって、2 つの分布をより見分けやすくすることはできるだろうか。ピンスカーの不等式の証明では結果を統合すると KL は減少するほかなく、37.1.2.4 節の連鎖律でもペアの一方の変数を捨てると同じことが起きた。同じ原理があらゆる f ダイバージェンスとあらゆる形態の処理に対して成り立つ。**通信路**（channel）$\kappa(y \mid x)$ とは、決定論的かランダム化されているかを問わず、入力 x を出力 y へ変換する任意の規則のことである。P をこの通信路に通すと出力分布 $(P\kappa)(y) = \sum_x \kappa(y \mid x)\, p(x)$ が生じ、Q を通すと同じように $Q\kappa$ が生じる。

命題（f ダイバージェンスに対するデータ処理不等式）。 任意の通信路 κ、凸関数 f（ただし $f(1) = 0$）、および有限集合または可算集合上の $P \ll Q$ に対し、次が成り立つ。

$$D_f(P\kappa \,\|\, Q\kappa) \;\le\; D_f(P\|Q). \tag{37.32}$$

証明。$\sum_y \kappa(y \mid x) = 1$ より、$D_f(P\|Q) = \sum_y \sum_x \kappa(y \mid x)\, q(x)\, f\!\left(p(x)/q(x)\right)$ と展開できる。出力 y として $(Q\kappa)(y) > 0$ を満たすものを固定する。正規化された重み $\kappa(y \mid x)\, q(x)/(Q\kappa)(y)$ は x に関して和をとると 1 になり、この重みのもとで密度比 $p(x)/q(x)$ の期待値をとると $(P\kappa)(y)/(Q\kappa)(y)$ が得られる。したがって内側の和にイェンセンの不等式を適用すると、下から $(Q\kappa)(y)\, f\!\left((P\kappa)(y)/(Q\kappa)(y)\right)$ で抑えられ、これは y 番目の項として $D_f(P\kappa\|Q\kappa)$ に現れるものにほかならない。出力 y が $(Q\kappa)(y) = 0$ となる場合には、$\kappa(y \mid x) = 0$ が $q(x) > 0$ なるすべての点で成り立つため $(P\kappa)(y) = 0$ でもあり、両辺のいずれにも寄与しない。最後に y にわたって和をとれば証明が完了する。■

どのダイバージェンス関数を選んでも、データを処理することによって 2 つの分布がより識別しやすくなることは決してない。深層学習では、この不等式を逆向きに読むことが多い。すなわち、学習された特徴量、量子化された出力、あるいはその他のデータ要約の上で測ったダイバージェンスは、常に元の入力間のダイバージェンスの下界になる。ピンスカーの不等式の証明で結果を統合した操作は、単に決定論的な通信路 $x \mapsto \mathbf{1}(x \in A^\star)$ にほかならない。通しで用いている分布の組では、この操作により KL は 0.396 から 0.382 へと下がり、χ^2 も 1.013 から $0.16/0.2 + 0.16/0.8 = 1$ へと下がる。また、KL の特別な場合は 37.6.2.2 節における相互情報量（mutual information）のデータ処理不等式（data-processing inequality）の原動力でもある。

可逆な変換は両方向の通信路として働くため、Equation 37.32 において等号を強制する。すなわち、あらゆる f ダイバージェンスは結果のラベルの付け替えや座標軸のスケール変換に対して不変である。対照的に、Table 37.2 の最後の 2 行は軸のスケールを変えると値が変わる。それらの距離は、d に対する幾何学的な感度を得る引き換えに、スケール不変性を手放しているのである。

37.3.4 フェンシェル共役とクリティック下界

分布 Q からサンプルを生成する生成ネットワークと、真の分布 P からのサンプルからなるデータセットが手元にあるとしよう。どちらのサンプル集合からも、Equation 37.26 が必要とする密度比 $p(x)/q(x)$ を直接知ることはできない。それでもサンプルだけから D_f を推定できるだろうか。実は、任意の f ダイバージェンスは P と Q のもとでの期待値だけを用いて下から評価でき、その下界によってあらゆるダイバージェンスが敵対的訓練の目的関数へと変わる。

37.3.4.1 下界の導出

中心となる道具は、35.7.4 節で導入した凸共役（convex conjugate）である。凸関数 f に対し、その共役は次式で定義される。

$$f^*(t) = \sup_{u} \big(ut - f(u)\big),$$

すなわち $-f^*(t)$ は、f の傾き t の接線の縦軸切片となる（Figure 37.9）。$f(u) = u\log u$ の場合、上限は $\log u + 1 = t$ の点で達成され、$f^*(t) = e^{t-1}$ となる。凸関数は常にそのすべての接線の上側にあるため、フェンシェル・ヤングの不等式が得られる。

$$f(u) \geq ut - f^*(t) \quad \text{for all } t, \tag{37.33}$$

ここで等号は、t が f の点 u における傾きと一致するとき、すなわち $t = f'(u)$ のときに常に成り立つ。

ここで Equation 37.33 を**ダイバージェンスの内部で各点ごとに**適用し、傾きが x に応じて変化できるようにしよう。具体的には、**クリティック**と呼ばれる任意の関数 $T(x)$ をとり、$u = p(x)/q(x)$ および $t = T(x)$ を代入する。

命題（クリティック下界）。 **f を $f(1) = 0$ を満たす微分可能な凸関数とし、$P \ll Q$ とし、T を $\mathbb{E}_P|T| < \infty$ かつ $\mathbb{E}_Q|f^*(T)| < \infty$ を満たす任意のクリティックとする。このとき、次が成り立つ。**

$$D_f(P\|Q) \;\geq\; \mathbb{E}_{x\sim P}[T(x)] - \mathbb{E}_{x\sim Q}[f^*(T(x))]. \tag{37.34}$$

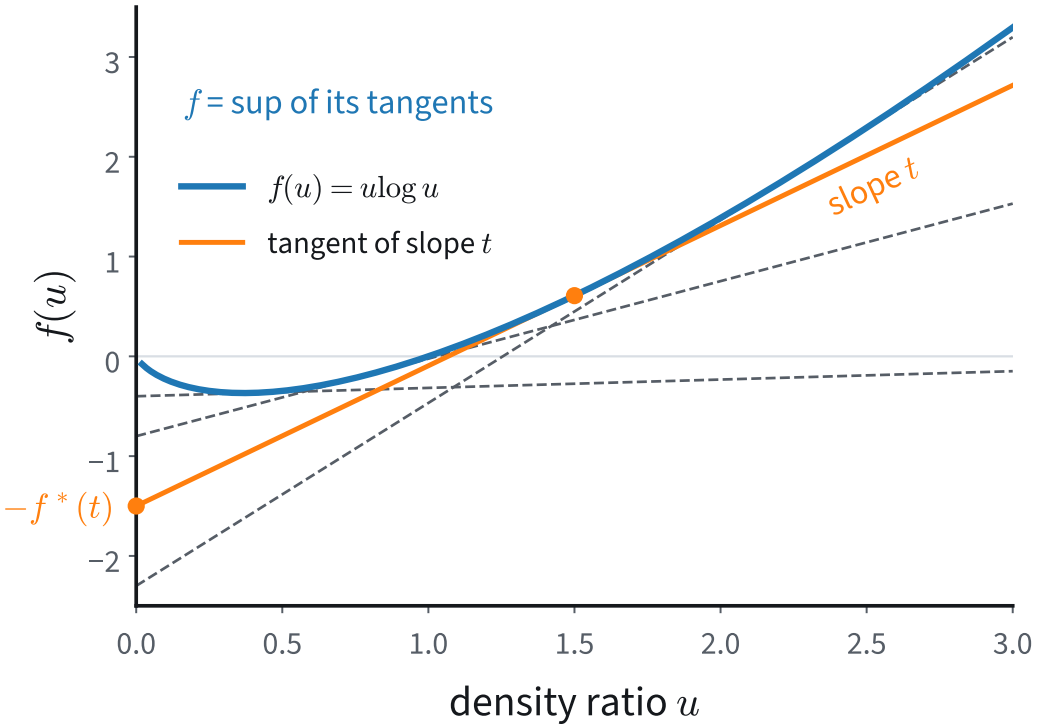


Figure 37.9: 凸なダイバージェンス関数（ここでは $f(u) = u \log u$）は、その接線群の上側包絡線である。傾き t の接線は縦軸切片 $-f^*(t)$（ここでは $-e^{t-1}$）を持つ。1 本の接線だけを残せばフェンシェル・ヤングの下界が得られ、傾きにわたって上限をとれば $f(u)$ が復元される。この不等式を密度比に各点ごとに適用するとクリティック下界が得られる。

さらに、すべてのクリティックにわたる右辺の上限は $D_f(P\|Q)$ に等しく、$T^\star(x) = f'(p(x)/q(x))$ によって $T^\star$ が有限となるすべての点で達成される (Nguyen et al. 2010; Nowozin et al. 2016)。

証明。 Equation 37.33 に $u = p(x)/q(x)$ と $t = T(x)$ を代入し、両辺に $q(x) \geq 0$ を掛ける。

$$q(x)\, f\!\left(\frac{p(x)}{q(x)}\right) \;\geq\; p(x)\, T(x) - q(x)\, f^*(T(x)).$$

x にわたって和（または積分）をとれば Equation 37.34 が得られる。フェンシェル・ヤングの不等式は $t = f'(u)$ で等号が成り立つため、$T^\star(x) = f'(p(x)/q(x))$ と選べばすべての x で不等式が等式になり、したがって期待値においても等号が成り立つ。■

Equation 37.34 の右辺にある 2 つの項は、実データにわたる T の期待値と生成サンプルにわたる $f^*(T)$ の期待値という単純な期待値であり、明示的な密度を一切必要としない。したがって T をニューラルネットワークとしてパラメータ化し、ミニバッチ上で訓練してこの下界を最大化できる。KL ダイバージェンスの場合、この下界はすべてのクリティックに対して $D_{\mathrm{KL}}(P\|Q) \geq \mathbb{E}_P[T] - \mathbb{E}_Q[e^{T-1}]$（任意の T）となる。Nowozin et al. (2016) による **f-GAN**（これにちなんで Equation 37.34 は f-GAN 下界とも呼ばれる）は、この変分表現を 2 人ゲームへと変換する。すなわち、クリティックは下界を最大化してダイバージェンスを追跡させ、Q を生成する生成器はそれを最小化する。このようなペアの実装と訓練は 16.2 節で扱う。

37.3.4.2 初期の GAN

Goodfellow et al. (2014) による初期の GAN は、この枠組みにおいて $f(u) = u\log u - (u+1)\log(u+1)$ と選ぶだけでそのまま導かれる。その凸共役は $f^*(t) = -\log(1 - e^t)$（ただし $t < 0$）である。クリティックを識別器 $D(x) \in (0,1)$ の対数として $T = \log D$ と書けば、クリティック下界はおなじみの二値交差エントロピー（binary cross-entropy）のゲームになる。

$$\mathbb{E}_P[T] - \mathbb{E}_Q[f^*(T)] = \mathbb{E}_{x\sim P}[\log D(x)] + \mathbb{E}_{x\sim Q}[\log(1 - D(x))].$$

最適クリティック $T^\star = f'(p/q) = \log\big(p/(p+q)\big)$ は、ベイズ最適な識別器 $D^\star = p/(p+q)$ に対応する。イェンセン・シャノンのダイバージェンス関数 f_{JS}（Table 37.3）と比較すると、$f(u) = 2f_{\mathrm{JS}}(u) - (u-1)\log 2 - \log 4$ であることがわかる。すなわち、イェンセン・シャノン関数の 2 倍に、$u-1$ の 1 次項（ダイバージェンスを変化させない）と定数 $f(1) = -\log 4$（値を平行移動させるだけである）を加えたものになっている。したがって $D^\star$ において、このミニマックスゲームは $2\,\mathrm{JS}(P,Q) - \log 4$ を達成し、通しで用いている分布の組ではその値は $2(0.0911) - 1.386 = -1.204$ に等しい。

37.3.4.3 下界の限界

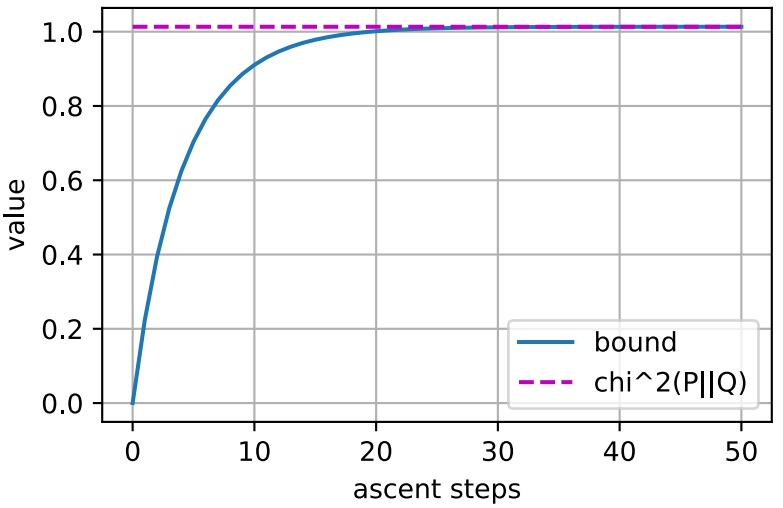
実際には、クリティック下界は 2 つの限界に直面し、そのうち第 1 の限界は直接測定できる。下界が等号を達成するのは $T^\star$ のみであるため、訓練が不十分なクリティックは真のダイバージェンスより小さい値を返すのである。クリティックが訓練を進めるにつれてその差をどう縮めていくかを観察してみよう。

通しで用いている離散分布の組では、期待値は単純な和であり、クリティックは各結果に 1 つずつ対応する 3 つの数値からなるベクトルにすぎない。χ^2 ダイバージェンスをとり、$f(u) = (u-1)^2$、共役 $f^*(t) = t + t^2/4$、最適クリティック $T^\star = 2(p/q - 1)$ としよう（ここでは共役の上限をすべての実数 u にわたってとっている。演習問題 6a でこれを導出し、$T^\star$ において何も損

なわれないことを確かめる）。$T = 0$（このとき下界は 0）から出発し、ステップサイズ 1 で下界に対して勾配上昇法（gradient ascent）を実行する。この下界の $T(x)$ に関する導関数は $p(x) - q(x)\,(1 + T(x)/2)$ である。命題により、この勾配上昇は下から $\chi^2(P\|Q) = 1.0133$ に向かって登っていかなければならない。妥当性確認として、最後の行では正しい向きを指しつつ 2 倍だけ行き過ぎた過信クリティック $2T^\star$ も評価する。

```
def fgan_bound(T, p, q, f_star):
    """E_P[T] - E_Q[f*(T)]: a lower bound on D_f(P||Q) for any critic T."""
    return np.sum(p * T) - np.sum(q * f_star(T))

chi2_star = lambda t: t + t ** 2 / 4    # conjugate of f(u) = (u - 1)^2
exact = f_divergence(generators['chi^2'], p, q)
T, bounds = np.zeros(3), []             # untrained critic: one number per outcome
for step in range(51):
    bounds.append(fgan_bound(T, p, q, chi2_star))
    T = T + (p - q * (1 + T / 2))       # ascent step 1 along p - q f*'(T)
shown = (0, 1, 2, 5, 10, 20, 50)
print(f'exact chi^2(P||Q) = {exact:.4f}')
print('steps ' + ''.join(f'{s:8d}' for s in shown))
print('bound ' + ''.join(f'{bounds[s]:8.4f}' for s in shown))
T_star = 2 * (p / q - 1)                # the optimal critic f'(p/q)
print(f'overconfident critic 2 T*: bound = '
      f'{fgan_bound(2 * T_star, p, q, chi2_star):.4f}')
d2l.plot(np.arange(51), [bounds, [exact] * 51], 'ascent steps', 'value',
         legend=['bound', 'chi^2(P||Q)'], figsize=(4.5, 2.8))
```

```
exact chi^2(P||Q) = 1.0133
steps        0       1       2       5      10      20      50
bound   0.0000  0.2240  0.3935  0.7036  0.9107  1.0013  1.0133
overconfident critic 2 T*: bound = 0.0000
```



5 ステップ後、クリティックは 0.70（真のダイバージェンス 1.01 の約 70%）を下界として保証し、10 ステップ後には 90%、20 ステップ後には 99% に達する。どの時点でも下界が真の値を上回ることはない。過信クリティック $2T^\star$ でさえ 0 を返し、未訓練のクリティックと変わらない。

期待値が厳密に計算できる限り、敵対的なダイバージェンス推定値は常に過小評価であり、その数値自体からはどれだけ取りこぼしているかは決してわからない。期待値を有限標本の平均に置き換えた場合、それが（期待値の意味で）正当な下界であり続けるのは、クリティックの訓練に使われていない新たなサンプル上で評価したときに限られる。

勾配更新の式からは、取りこぼされたダイバージェンスがどこに潜んでいるかもわかる。$T(x)$ は $q(x)$ に比例する速さで動くため、モデルのサンプルが最もまれな箇所でクリティックの学習は最も遅くなる。通しで用いている分布の組ではそれは第 1 の結果であり、まさに密度比が最大でダイバージェンスの大半が集中している場所である（演習問題 6b）。

第 2 の限界は、冒頭の互いに素な台の問題へと、今度はクリティックの視点から立ち返らせる。$T^\star = f'(p/q)$ を仮定 $P \ll Q$ の外へ形式的に拡張すると、密度比はデータの台の上では ∞、モデルの台の上では 0 に等しくなるため、$T^\star$ は前者で定数値 $f'(\infty)$、後者で定数値 $f'(0)$ をとる。生成器は自身のサンプル上で評価したクリティックの勾配を用いてパラメータを更新するため、そこで局所的に定数となっているクリティックからは勾配が 0 しか得られない。生成器には、どちらの方向へ動けばよいかを知らせる信号が一切届かないのである。

これら 2 つの限界はいずれも同じ根源、すなわち密度比 p/q に由来している。有限個のサンプルから p/q が直接手に入ることは決してなく、台が離れるやいなやこの比は 0 または ∞ へと潰れ、2 つの分布がどれだけ離れているかという手がかりをすべて消し去ってしまう。37.4 節では、密度比に一切頼らない比較法へと目を向ける。

37.3.5 まとめ

KL ダイバージェンスは広い族の 1 つの要素にすぎない。一般に f ダイバージェンスは、密度比 p/q の凸関数の Q のもとでの期待値をとるものである。イェンセンの不等式によりすべての要素の非負性が保証され、滑らかな要素はすべて $f''(1)$ という曲率の定数倍を除いて P と Q が近いときに一致し、いかなるデータの後処理も f ダイバージェンスを増加させることはない。KL の向きの選択はモデリング上の判断である。順方向 KL（最尤法）はすべてのモードを覆うように広がる一方、逆方向 KL（変分推論）は単一のモードに集中し得る。操作的には、P と Q を見分ける最良の 1 サンプル検定は確率 $\frac{1}{2}(1 + \mathrm{TV})$ で成功し、ピンスカーの不等式は TV を上から $\sqrt{D_{\mathrm{KL}}/2}$ で抑える。また凸共役を用いれば、任意のクリティックによって D_f を P と Q にわたる期待値のみから下界評価でき、敵対的訓練はこの仕組みによってサンプルからダイバージェンスを推定している。それでもこの族のすべての要素は、密度比が持つ同じ盲点を共有している。互いに素な台の上では、台どうしがどれほど近くても遠くても、D_f は定数 $f(0) + f'(\infty)$ に凍りついてしまうのである。

37.3.6 演習問題

1. **ダイバージェンス関数と引数の順序。** Table 37.3 の各行は、テンプレート中の f（Equation 37.26）の 1 つの選択に対応している。
   1. ダイバージェンス関数 f に対し、$f^\diamond(u) = u\,f(1/u)$ と定義する。$f^\diamond$ が $f^\diamond(1) = 0$ を満たす凸関数であること、および $D_f(Q\|P) = D_{f^\diamond}(P\|Q)$ が成り立つことを示せ。さらに $f^\diamond$ を KL 関数 $u\log u$ と χ^2 関数 $(u-1)^2$ に対して計算し、後者が定めるダイバージェンスを x に関する和として書き下せ。
   2. $c\,(u-1)$ を f に加えてもダイバージェンスは変化しない。D_f は、$f^\diamond(u) = f(u) + c\,(u-1)$ がなんらかの定数 c に対して成り立つとき常に対称であることを示せ。Table 37.3 で対称と記されている各行についてこの条件を確かめ、さらに $D_{f_\alpha}(Q\|P) = D_{f_{1-\alpha}}(P\|Q)$ が関数 Equation 37.28 に対して成り立つことを示せ。この恒等式を用いて、α ダイバージェンスの行の「対称か」列の記述を確かめよ。

3. Equation 37.27 を比 $u = p/q$ を用いて展開し、Table 37.3 のイェンセン・シャノン関数を導出せよ。
4. **レーニイダイバージェンス。** $\alpha > 0$、$\alpha \neq 1$ に対し、$D_\alpha(P\|Q) = \frac{1}{\alpha-1}\log\sum_x p(x)^\alpha\, q(x)^{1-\alpha}$ と定義する (Rényi 1961)。$\sum_x p^\alpha q^{1-\alpha} = 1 + \alpha(\alpha-1)\, D_{f_\alpha}(P\|Q)$ であり、D_α が D_{f_α} の単調増加関数であることを示せ。ここから $D_\alpha \geq 0$ であること、および D_α がデータ処理不等式 Equation 37.32 を満たすことを導け。さらに $\lim_{\alpha\to 0} D_\alpha(P\|Q)$ を求め、それが 0 になる（p と q が同じ結果の集合上で正であるとき）理由を説明せよ。したがって、レーニイ族が逆方向 KL に $\alpha \to 0$ で到達**しない**一方、D_{f_α} は到達するのはなぜか。

2. **ダイバージェンスの中の距離関数。** Table 37.3 の最後の列には、どのダイバージェンスが距離関数であるかが記されている。
1. TV に対する三角不等式を、ℓ_1 距離の半分という表現から証明せよ。
2. ヘリンガー距離 $d_{\mathrm{H}} = \sqrt{d_{\mathrm{H}}^2}$ がベクトル $\left(\sqrt{p(x)}\right)_x$ と $\left(\sqrt{q(x)}\right)_x$ の間のユークリッド距離であり、したがって d_{H} が距離関数であることを示せ。また、2 値上の 3 つの分布であって、d_{H}^2 と JS の両方が三角不等式に反するものを求めよ。

3. **モードを取りこぼすコスト。** モデル Q がデータ分布 P の一部を取りこぼしている状況を考える。
1. $P = (1-w)\,Q + wR$（ただし $0 < w < 1$）とし、R はある集合（それに対して Q が確率 0 を割り当てるもの）上の分布とする。$D_{\mathrm{KL}}(P\|Q)$、$D_{\mathrm{KL}}(Q\|P)$、$\mathrm{TV}(P,Q)$、$d_{\mathrm{H}}^2(P,Q)$ を w の関数として計算せよ。取りこぼされた質量にペナルティを課すのはどちらの向きの KL か。また、有限の値をとるダイバージェンスは $w \to 0$ でどのように振る舞うか。
2. 次に $P = w_1\mathcal{N}(m_1, s^2) + w_2\mathcal{N}(m_2, s^2)$（平均間の距離を $\Delta = |m_2 - m_1|$ とする）とし、モデル $Q = \mathcal{N}(m_1, s^2)$ は第 1 成分のみを保持するものとする。$D_{\mathrm{KL}}(Q\|P) \leq \log(1/w_1)$ がすべての Δ に対して成り立ち、極限 $\Delta \to \infty$ で等号となることを示し、この極限を (a) と関連づけよ。さらにこの上界を用いて、混合分布へのガウス分布の当てはめで出力された 0.356 nat と 1.202 nat という 2 つの逆方向 KL の値を説明せよ。
3. 2 つの成分が十分に離れており、ほとんどすべての x において無視できない密度を持つ成分が高々 1 つであると仮定する。このとき $D_{\mathrm{KL}}(P\|Q) \approx w_2\Delta^2/(2s^2) + w_1\log w_1 + w_2\log w_2$ となることを示せ。この近似式を先ほどの実験の混合分布に対して計算し、`kl(log_p(x), log_q(x, [-2.0, np.log(0.6)]), x)` で確かめよ。実験の順方向 KL の列には比較対象となる 2 つの値が出力されている。すなわち、5.981 nat（パラメータが $(-1.998, 0.603)$ であり厳密な $(-2, 0.6)$ とはわずかに異なる逆方向 KL の当てはめにおける値）と、0.558 nat（順方向 KL の当てはめにおける値）である。

4. **局所二次の領域。** 命題 Equation 37.29 は D_f と $\frac{1}{2}f''(1)\,\chi^2$ の差を上から評価している。
1. 演習問題 7b（37.1 節）では、ガウス分布間の KL ダイバージェンスの 2 つの向きが初めて食い違う次数を求めた。KL 関数と逆方向 KL 関数のどの性質がその次数を説明するか。証明のテイラー展開をもう 1 項先まで進め、局所二次近似の実験が出力する比が $1 + c\,t\,f'''(1)/f''(1) + O(t^2)$（ただし $c = \mathbb{E}_Q[(u-1)^3]/(3\chi^2(P\|Q))$）であり、通しで用いている分布の組では約 0.49）となることを示せ。4 つの関数に対して $f'''(1)/f''(1)$ を計算せよ。$t = 0.01$ において逆方向 KL の不足分が KL の 2 倍になり、また二乗ヘリンガーと JS が同じ値を出力するのはなぜか。
2. 全変動の関数 $\frac{1}{2}|u-1|$ は 1 において二階導関数を持たない。$P_t = Q + t\,(P-Q)$（ただし $t \in (0,1]$）と置くとき、$\mathrm{TV}(P_t, Q)$ と $\chi^2(P_t\|Q)$ は t に対してどのようにスケールするか。このことから、ピンスカーの不等式 Equation 37.31 が TV を KL の定数倍ではなく平方根で抑える理由を説明せよ。
3. P と Q を表の出る確率が b と $b+\delta$ のコインとする。命題を用いて $D_{\mathrm{KL}}(P\|Q) =$

$\delta^2/(2b(1-b)) + O(\delta^3)$ を示せ。演習問題 5a（37.2 節）では、ベルヌーイ分布の KL ダイバージェンスから直接同じ展開を得ている。因子 $1/(b(1-b))$ はこのコインのフィッシャー情報量である（36.6 節）。$\mathrm{TV}/\sqrt{D_{\mathrm{KL}}/2}$ の $\delta \to 0$ における極限を求め、この極限でピンスカーの不等式が等号を達成する b の値を求めよ。ピンスカーの不等式に続くコインの実験では $b = \frac{1}{2}$ をとり、比として 0.999999 が $\delta = 10^{-3}$ で出力されている。この比が 1 に近づく速さが δ^2 であって δ ではないのはなぜか。

5. **ヘリンガーによる評価と検定。**●ピンスカーの不等式は $D_{\mathrm{KL}} > 2$ になると自明になる。$\mathrm{BC}(P,Q) = \sum_x \sqrt{p(x)q(x)} = 1 - \frac{1}{2}d_{\mathrm{H}}^2(P,Q)$ をバタチャリア係数とする。
   1. $\frac{1}{2}d_{\mathrm{H}}^2(P,Q) \le \mathrm{TV}(P,Q)$ を、$|p-q| = |\sqrt{p}-\sqrt{q}|\,(\sqrt{p}+\sqrt{q})$ を用いて証明せよ。
   2. $\mathrm{TV}(P,Q) \le d_{\mathrm{H}}(P,Q)\sqrt{1 - \frac{1}{4}d_{\mathrm{H}}^2(P,Q)}$ を 33.1 節のコーシー・シュワルツの不等式を用いて証明し、この上界を $\mathrm{TV}^2 \le 1 - \mathrm{BC}^2$ と書き直せ。
   3. イェンセンの不等式を用いて $\mathrm{BC}(P,Q) \ge e^{-D_{\mathrm{KL}}(P\|Q)/2}$ を示し、ブレタニョール・フーバーの不等式 $\mathrm{TV}(P,Q) \le \sqrt{1 - e^{-D_{\mathrm{KL}}(P\|Q)}}$ を導け (Bretagnolle and Huber 1978; Canonne 2022)。D_{KL} がどの値をとるとき、これは Equation 37.31 より精密になるか。
   4. ある検定が n 個の独立標本を受け取り、それらはすべて P からのものであるか、すべて Q からのものであるかのいずれかが確率 $\frac{1}{2}$ ずつで生じるものとする。(c) と 37.1.2.4 節の独立標本にわたる KL の加法性を用いて、いかなる検定も少なくとも確率 $\frac{1}{4}e^{-nD_{\mathrm{KL}}(P\|Q)}$ で誤ることを示せ。誤り確率を高々 η に抑えるには何個のサンプルが必要か。ピンスカーの不等式からはこの種の評価が得られないのはなぜか。
6. **共役とクリティック。**クリティック下界 Equation 37.34 にはダイバージェンス関数の共役 f^* が必要である。
   1. 共役 $f^*(t) = t + t^2/4$（χ^2 関数 $f(u) = (u-1)^2$ に対して $u \in \mathbb{R}$ にわたる上限をとったもの）と最適クリティック $T^\star = 2(p/q - 1)$ を導出せよ。関数の本来の定義域 $u \in (0,\infty)$ 上で計算をやり直し、上限が $t + t^2/4$（$t \ge -2$ のとき）および -1（$t \le -2$ のとき）となることを示せ。
   2. $\mathbb{R}$ 上の共役を用いる。すべてのクリティック T に対し、$\chi^2(P\|Q)$ と下界の差が $\frac{1}{4}\mathbb{E}_Q\big[(T-T^\star)^2\big]$ に等しいことを示せ。ここから、定数倍したクリティック $T = cT^\star$ が保証する下界は $\chi^2(P\|Q)\,(2c - c^2)$ であることを導き、実験が $2T^\star$ に対して出力する値の理由を説明せよ。さらに、実験の勾配上昇 1 ステップによって $T(x) - T^\star(x)$ が $1 - q(x)/2$ 倍されることを示せ。10 ステップ後に差の大半を占めているのは通しで用いている分布の組のどの結果であり、それはなぜか。この勾配上昇はどのステップサイズに対して収束するか。
   3. 全変動の関数 $f(u) = \frac{1}{2}|u-1|$ の共役を $u \in (0,\infty)$ 上で計算せよ。値が $[-\frac{1}{2}, \frac{1}{2}]$ に収まるクリティックだけで下界を最大化するのに十分であることを示し、$\mathrm{TV}(P,Q) = \sup_g\big(\mathbb{E}_P[g] - \mathbb{E}_Q[g]\big)$（上限はすべての関数 g のうち $|g| \le \frac{1}{2}$ を満たすものにわたる）を結論づけよ。これは 37.4.1 節で学ぶ種類の上限表現である。この上限を達成するのはどの g か。

37.4 テスト関数と輸送

サンプルがデータを外しているモデルは、f ダイバージェンスからは何も学習できない。互いに素なサポート上では、この族に属するダイバージェンスはすべて定数に帰着し（37.3.1.2 節）、定数の勾配は 0 だからである。ただしこの診断は、2 つの点質量という極端な例から導かれたものであった。現実の確率分布は通常広がりを持っており、2 つのガウス分布は裾が決して 0 にならないため至るところで重なり合う。わずかな重なりがあれば、勾配は戻ってくるのだろうか。ま

た、f ダイバージェンスには実用上の負担も残されていた。生のサンプルからは得られない密度比 p/q に依存するため、ダイバージェンスを推定するだけでもクリティックネットワークを訓練しなければならなかったのである（37.3.4 節）。クリティックを一切用いずに、サンプルだけから 2 つの分布を比較することはできるだろうか。

最初の問いに数値で答えるため、データ分布を $P = \mathcal{N}(0, 1)$ とし、モデルを同じ釣鐘曲線を標準偏差の d 倍だけ平行移動させた $Q = \mathcal{N}(d, 1)$ とする（d はモデルの唯一のパラメータである）。以下では、必要なライブラリをインポートしたあと、このペアに対して Table 37.2 の 5 つの比較指標と、それぞれの d に関する傾き（訓練中にモデルが受け取る勾配）を評価する。コード中で用いる公式は前節および本節で導出するが、ひとまず出力される数値だけに注目してほしい。以下ではこれを冒頭の表と呼ぶ。

```
from scipy import optimize
from scipy.stats import norm
```

```
def js(d, x=np.linspace(-12, 20, 6401)):
    """JS divergence of N(0, 1) and N(d, 1) by quadrature on the grid x."""
    log_p, log_q = norm.logpdf(x), norm.logpdf(x, loc=d)
    log_m = np.logaddexp(log_p, log_q) - math.log(2)    # log of the mixture
    terms = np.exp(log_p) * (log_p - log_m) + np.exp(log_q) * (log_q - log_m)
    return terms.sum() * (x[1] - x[0]) / 2

comparisons = {                         # N(0, 1) against N(d, 1), RBF ell = 1
    'KL': lambda d: d ** 2 / 2,
    'JS': js,
    'TV': lambda d: 2 * norm.cdf(d / 2) - 1,
    'MMD^2': lambda d: 2 / math.sqrt(3) * (1 - math.exp(-d ** 2 / 6)),
    'W1': lambda d: d}
h = 1e-3                                # step of the central difference
print('   d' + ''.join(f'{name:>8s} {"slope":>7s}' for name in comparisons))
for d in (0.5, 1.0, 2.0, 4.0, 8.0):
    print(f'{d:4.1f}' + ''.join(
        f'{f(d):8.3f} {(f(d + h) - f(d - h)) / (2 * h):7.1e}'
        for f in comparisons.values()))
```

```
   d      KL   slope      JS   slope      TV   slope   MMD^2   slope      W1   slope
 0.5   0.125 5.0e-01   0.030 1.2e-01   0.197 3.9e-01   0.047 1.8e-01   0.500 1.0e+00
 1.0   0.500 1.0e+00   0.111 2.0e-01   0.383 3.5e-01   0.177 3.3e-01   1.000 1.0e+00
 2.0   2.000 2.0e+00   0.337 2.2e-01   0.683 2.4e-01   0.562 4.0e-01   2.000 1.0e+00
 4.0   8.000 4.0e+00   0.633 6.9e-02   0.954 5.4e-02   1.074 1.1e-01   4.000 1.0e+00
 8.0  32.000 8.0e+00   0.693 2.0e-04   1.000 1.3e-04   1.155 7.2e-05   8.000 1.0e+00
```

2 つのガウス分布の間隔が標準偏差の半分であれば、5 つの傾きはすべて 0.1 から 1 の間に収まっており、どの目的関数を用いてもモデルを効果的に訓練できる。ところが $d = 8$ になると、イェンセン・シャノン・ダイバージェンスは 0.693 で、全変動（total variation）は 1.000 で飽和し（互いに素な点質量で見た定数 $\log 2$ および 1 と同じ値である）、どちらの傾きも約 10^{-4} まで落ち込む。裾が重なっていても救われないのである。KL ダイバージェンスは依然として d そのものに等しい健全な傾きを保つが、それはガウスモデルがあらゆるデータ点に正の密度を割り当てるからにすぎない。サンプルが低次元曲面上に乗る陰的生成器（implicit generator）には、そのような密度は存在しない。残るのは、点質量に対しても有益な情報を保っていた 2 つの距離である。MMD^2 は最初こそ増加するものの、$d = 8$ までには頭打ちとなり、値 1.155 に対して傾きは 10^{-4} 未満になる。あらゆる間隔において線形のまま d に一致し、傾き 1 を保ち続けるのは

W_1 だけである。

MMD も W_1 も密度比 p/q には依存しない。その代わりに MMD は、ある関数クラスを P と Q がどのように**平均**するかを比較する。平均はサンプルから直接推定でき、どの関数クラスを選ぶかによって、2 つの分布がどれだけ離れていても有用な信号を生み出せるかが決まる。一方、W_1 は幾何学的な問いから出発する。一方の分布をもう一方へと変形するには、確率質量をどれだけ遠くまで運ばなければならないかという問いである。驚くべきことに、これもまた平均の比較になることがわかる。本節の残りの議論は 2 つの結果を中心に構成される。固定された特徴の組の上では、平均の比較は補助ネットワークを訓練することなく閉形式で計算できる。他方、質量を動かす最小コストは、傾きが高々 1 の関数全体にわたる平均の差の最大値に等しく、これが Wasserstein GAN（16.2 節）の背後にある双対性（duality）である。本節を通して P はデータ、Q はモデルを表し、実直線上の具体例を用いて議論を進める。

37.4.1 積分確率指標と MMD

37.4.1.1 テスト関数のクラス

全変動はれっきとした距離（metric）であるが、それでも冒頭の表では平坦になってしまった。その変分形式を見れば、なぜそうなったのか、そしてどう修正すればよいかの両方がわかる。全変動とは、任意の事象に関する確率の P と Q の最大の不一致、すなわち任意の事象 A にわたる（37.3.3 節）$\mathrm{TV}(P,Q) = \sup_A |P(A) - Q(A)|$ である。事象の確率はその指示関数（indicator function）の期待値にほかならないため、全変動はある関数クラス上でとった 2 つの平均の差の上限（supremum）となっている。ここで関数を指示関数だけに限定すべき理由はない。任意のクラス $\mathcal{G}$ に属する**テスト関数**（test function）g が与えられたとき、**積分確率指標**（integral probability metric、IPM）(Müller 1997) は次のように定義される。

$$\mathrm{IPM}_{\mathcal{G}}(P,Q) = \sup_{g \in \mathcal{G}} \left(\mathbb{E}_{x\sim P}[g(x)] - \mathbb{E}_{x\sim Q}[g(x)] \right). \tag{37.35}$$

$\mathcal{G}$ を、空間内の各点にスコアをつける監査人のパネルだと考えてみよう。IPM は、パネル内の監査人が見つけ出せる平均スコアの最大の差を報告する。ここで考えるクラスはすべて対称である。すなわち、$-g$ が $\mathcal{G}$ に属するのが g が属するときであるため、この差は常に非負であり、P と Q に関して対称になる。「距離（metric）」という言葉が約束するように $P = Q$ のときに限って IPM が 0 になるかどうかは、異なる任意の 2 分布を識別できるほど $\mathcal{G}$ が豊かであるかどうかにかかっている。37.3.4 節のクリティックも同様の監査の役割を果たしていたが、決定的な違いがある。変分下界 $D_f(P\|Q) \geq \mathbb{E}_P[T] - \mathbb{E}_Q[f^*(T)]$ では、凸共役（convex conjugate）f^* が制約のないクリティック T を自動的に抑制していた。これに対し IPM では、g が P と Q の両方で直接平均されるため、g を制約するものは $\mathcal{G}$ に属すること以外には何もない。したがって、すべては適切なクラスを選ぶことにかかっている。

ここでは次の 3 つの関数クラスが主役となる。

- $|g| \leq \frac{1}{2}$ を満たす有界関数からは全変動が復元される。指示関数から $\frac{1}{2}$ を引いた関数はこの上界を満たし、定数は平均の差をとると相殺され、$g = \frac{1}{2}\,\mathrm{sign}(p-q)$ が上限を達成する。
- 固定された特徴の組に関して線形な関数からは、次に構築する**最大平均値差異**（maximum mean discrepancy、MMD）(Gretton et al. 2012) が得られる。
- 傾きが高々 1 の関数からはワッサースタイン距離（Wasserstein distance）が得られる。これについては 37.4.2 節で詳しく扱う。

なぜ毎回、可能な限り最大のクラス $\mathcal{G}$ を選ばないのだろうか。有用なクラスは互いに競合する 2 つの要件を満たす必要があり、$\mathcal{G}$ を大きくしすぎると第 1 の要件が破綻するからである。第 1 に、標本平均が真の期待値に追随しなければならない。サンプルを用いれば密度比なしに期待値を直

接推定できるとはいえ、監査人のパネルが豊かすぎると、有限個の標本間に常に見せかけの差を見つけ出してしまう。たとえば、**同じ**連続分布から抽出した 1,000 点ずつの 2 つの独立標本は、共通の点を 1 つも持たない。第 1 の標本の指示関数は、第 1 の標本上では平均 1、第 2 の標本上では平均 0 となるため、経験全変動は 1 となり、これは P と Q が同一であるかどうかに依存しない。これを避けるには、そのような鋭く尖った関数を排除できるほど $\mathcal{G}$ が十分に小さいか、滑らかでなければならない。

第 2 に、P と Q が離れるにつれて差も増大し続けなければならず、そのためには距離とともに増大する関数が $\mathcal{G}$ に含まれている必要がある。全変動にはそのような関数が存在しない。テスト関数が $\tfrac{1}{2}$ で上から抑えられているため、差が 1 を超えることは決してないからである。最大の不一致が生じるのは事象 $\{p > q\}$ 上であり、冒頭の表の 2 つのガウス分布ではこれは半直線 $x < d/2$ となる。両方のガウス分布のもとでこの半直線の確率を計算すると、$\mathrm{TV} = \Phi(d/2) - \Phi(-d/2) = 2\Phi(d/2) - 1$ が得られる（ここで Φ は標準正規分布の累積分布関数（cumulative distribution function、CDF）である）。この値は 0.954（$d = 4$）に達し、10^{-4} 以内まで 1 に迫る（$d = 8$）。階段関数が 2 つの釣鐘曲線をほぼ完全に分離してしまえば、それ以上引き離してもスコアはほとんど変化しない。全変動はこれら両方の要件を満たしておらず、続く 2 つのクラスもこの両要件に照らして評価していく。

37.4.1.2 平均特徴とカーネル

最小の監査人パネルから出発し、同じ平均を持ちながら分散が異なる 1 組のペア $P = \mathcal{N}(0, 1)$ と $Q = \mathcal{N}(0, 4)$ で各段階を検証しながら、より優れたクラスを一から構築していこう。

最も単純な選択として、線形テスト関数 $g(x) = w\,x$（$|w| \le 1$）から始める。このとき $\mathbb{E}_P g - \mathbb{E}_Q g = w\,(\mathbb{E}_P X - \mathbb{E}_Q X)$ であるから、$|w| \le 1$ にわたって上限をとると、2 つの平均の距離 $|\mathbb{E}_P X - \mathbb{E}_Q X|$ が得られる。このパネルは分散に対して盲目であり、検証用のペアに対しては 0 を報告する。

位置だけでなく広がりも捉えるため、2 つ目の特徴を追加しよう。特徴写像 $\varphi(x) = (x, x^2)$ を定義し、線形結合 $g(x) = \mathbf{w}^\top \varphi(x)$（$\|\mathbf{w}\| \le 1$）をとる。$\mathbf{m}_P = \mathbb{E}_P[\varphi(x)]$ で P の**平均特徴**（mean feature）を表せば（同様に $\mathbf{m}_Q$ を Q に対して定める）、平均の差は $\mathbb{E}_P g - \mathbb{E}_Q g = \mathbf{w}^\top(\mathbf{m}_P - \mathbf{m}_Q)$ となる。コーシー・シュワルツの不等式により、単位球上における線形関数の最大値は、その係数ベクトルのユークリッドノルムにほかならない。

$$\mathrm{MMD}(P, Q) = \sup_{\|\mathbf{w}\| \le 1} \mathbf{w}^\top(\mathbf{m}_P - \mathbf{m}_Q) = \|\mathbf{m}_P - \mathbf{m}_Q\|. \tag{37.36}$$

このように MMD は P と Q の平均特徴ベクトル間の距離となり、上限をとる操作は完全に消え去った。検証用のペアでは平均特徴は $\mathbf{m}_P = (0, 1)$ と $\mathbf{m}_Q = (0, 4)$ であり、$\mathrm{MMD}^2 = 0^2 + 3^2 = 9$ となる。こうして、平均と二次モーメントの両方の違いを検出できるようになった。

次に、二乗ノルムを展開しよう。2 つの期待値の積は独立なサンプルに対する積の期待値に等しいため、$\|\mathbf{m}_P\|^2 = \mathbb{E}\langle \varphi(x), \varphi(x') \rangle$ と書ける（ただし $x, x' \sim P$ は独立）。独立な $y, y' \sim Q$ および交差項についても同様に展開すると、$\|\mathbf{m}_P - \mathbf{m}_Q\|^2 = \mathbb{E}\langle \varphi(x), \varphi(x') \rangle + \mathbb{E}\langle \varphi(y), \varphi(y') \rangle - 2\,\mathbb{E}\langle \varphi(x), \varphi(y) \rangle$ が得られる。ここで特徴 $\varphi(x)$ は決して単独では現れず、その内積だけが現れている点に注目しよう。そこで**カーネル関数**（kernel）を $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$ と定義すると、次式が得られる。

$$\mathrm{MMD}^2(P, Q) = \mathbb{E}_{x, x' \sim P}[k(x, x')] + \mathbb{E}_{y, y' \sim Q}[k(y, y')] - 2\,\mathbb{E}_{x \sim P, y \sim Q}[k(x, y)]. \tag{37.37}$$

この式には直観的な幾何学的意味がある。最初の 2 項はそれぞれ P の**内部**および Q の**内部**における平均類似度を測り、第 3 項は両者の**間**の類似度を測っている。P と Q からのサンプルが混ざり合っているときは、標本内類似度と標本間類似度が一致して 3 つの項が打ち消し合い、別々のクラスタに分離しているときは内部の項が支配的になる。最初の段階は**線形カーネル** $k(x,x') = xx'$ に、2 番目の段階は**二次カーネル** $k(x,x') = xx' + x^2 x'^2$ に対応している。

二次カーネルでさえまだ死角が残っている。平均と分散が一致する任意の 2 分布に対して距離が 0 になってしまうのである。x^3、次いで x^4 と順に追加していってもきりがない。そこで決定的な一歩となるのが、有限個の特徴のリストを書き下すことなく、カーネルを類似度の尺度として直接選ぶことである。**RBF カーネル** $k(x,x') = e^{-(x-x')^2/(2\ell^2)}$ は、2 点がおよそ**長さスケール**（length scale）ℓ 以内にあるときに両者を類似しているとみなす。直接選んだとはいえ、このカーネルも依然として特徴の内積になっている。実際、$e^{-x^2/(2\ell^2)}\,e^{-x'^2/(2\ell^2)}\,e^{xx'/\ell^2}$ と因数分解して最後の因子をべき級数展開すると、無限に多くの特徴 $\varphi_j(x) = x^j e^{-x^2/(2\ell^2)}/(\ell^j\sqrt{j!})$ $(j = 0,1,2,\dots)$ にわたる内積が現れる。これらは重みが急速に減衰するあらゆる次数の減衰単項式である。いまやパネルには無限に多くの監査人が含まれているが、その評決を下すのに必要なのは相変わらず Equation 37.37 の 3 つの平均だけである。$\mathbb{R}^n$ においては、単に $(x-x')^2$ を $\|x-x'\|^2$ に置き換えるだけで、他は何も変わらない。

このカーネルを用いれば死角は一切残らない。$\mathrm{MMD}(P,Q) = 0$ となるのは $P = Q$ のとき、かつそのときに限られ (Gretton et al. 2012)、RBF-MMD は真の距離となる。この性質を持つカーネルは**特性的**（characteristic）であるといい、線形カーネルと二次カーネルはそうではない。

ガウス分布に対しては、RBF の期待値を閉形式で計算できる。$Z \sim \mathcal{N}(\mu, s^2)$ とすると、標準的なガウス積分により次式が得られる。

$$\mathbb{E}\big[e^{-Z^2/(2\ell^2)}\big] = \Big(1 + \frac{s^2}{\ell^2}\Big)^{-1/2} \exp\!\Big(-\frac{\mu^2}{2(\ell^2 + s^2)}\Big). \tag{37.38}$$

独立な 2 つのガウス分布の差は再び分散の和を持つガウス分布になるため、Equation 37.37 の 3 つの項をすべて直ちに読み取れる。検証用のペアである $\mathcal{N}(0,1)$ 対 $\mathcal{N}(0,4)$ において $\ell = 1$ とすると、差 $x - x'$、$y - y'$、$x - y$ はいずれも平均 0 で、分散はそれぞれ 2、8、5 となり、$\mathrm{MMD}^2 = 3^{-1/2} + 9^{-1/2} - 2\cdot 6^{-1/2} \approx 0.094$ が得られる。冒頭の表の平行移動した 2 つのガウス分布に対しては、同じ積分から $\frac{2}{\sqrt{3}}\big(1 - e^{-d^2/6}\big)$ が導かれる（演習問題 1a）。Table 37.2 の点質量に対しては平均をとる必要すらなく、Equation 37.37 から直ちに $1 + 1 - 2k(0,d) = 2 - 2e^{-d^2/2}$ が得られ、同表の 4 行目と一致する。いずれの場合も d に依存するのは交差項だけであり、d が長さスケールの数倍を超えると急速に減衰する。したがって差は平坦になってしまい、第 2 の要件を満たさない。RBF の監査人は滑らかではあるが、近視眼的なのである。

37.4.1.3 サンプルからの推定

カーネルが真価を発揮するのは第 1 の要件、すなわちサンプルからの信頼性の高い推定である。P と Q からの有限標本が与えられると、Equation 37.37 の各期待値は点のペアにわたる平均になる。不偏推定量（unbiased estimator）を得るため (Gretton et al. 2012)、各点をそれ自身と組にする対角項は除外する（残すと標本内平均に上向きの統計的バイアスが生じるためである）。以下では、個数 $N = 1{,}000$ で $\mathcal{N}(0,1)$ から抽出した標本と、1,000 個を $\mathcal{N}(0,4)$ から抽出した標本に対し、線形・二次・RBF の各カーネルのもとでこの推定量を評価する。それぞれの母集団での真値は、先ほど計算したとおり 0、9、0.094 である。RBF カーネルについてはさらに 2 組のペアも試す。$\mathcal{N}(0,1)$ からの 2 つ目の独立標本（母集団値は 0）と、平行移動させた $\mathcal{N}(0.5,1)$ からの標本（母集団値は冒頭の表の 0.047）である。目的は、標本推定値がこれらの母集団値にどれだけ密接に追随するかを確かめることにある。

```
def mmd2_unbiased(x, y, k):
    """Unbiased MMD^2 between samples x and y for the kernel k(a, b)."""
    kxx = k(x[:, None], x[None, :])
    kyy = k(y[:, None], y[None, :])
    n, m = len(x), len(y)
    return ((kxx.sum() - np.trace(kxx)) / (n * (n - 1))
            + (kyy.sum() - np.trace(kyy)) / (m * (m - 1))
            - 2 * k(x[:, None], y[None, :]).mean())

linear = lambda a, b: a * b                         # features phi(x) = x
quadratic = lambda a, b: a * b + a ** 2 * b ** 2    # features (x, x^2)
rbf = lambda a, b: np.exp(-(a - b) ** 2 / 2)        # RBF kernel, ell = 1

rng = np.random.default_rng()
n = 1000
x_p = rng.standard_normal(n)                # P = N(0, 1)
y_var = 2 * rng.standard_normal(n)          # N(0, 4)
y_same = rng.standard_normal(n)             # N(0, 1) again
y_shift = rng.standard_normal(n) + 0.5      # N(0.5, 1)
for name, y, k, pop in [
        ('linear,    N(0,1) vs N(0,4)', y_var, linear, 0.0),
        ('quadratic, N(0,1) vs N(0,4)', y_var, quadratic, 9.0),
        ('RBF,       N(0,1) vs N(0,4)', y_var, rbf,
         3 ** -0.5 + 9 ** -0.5 - 2 * 6 ** -0.5),
        ('RBF,       N(0,1) vs N(0,1)', y_same, rbf, 0.0),
        ('RBF,       N(0,1) vs N(0.5,1)', y_shift, rbf,
         2 / np.sqrt(3) * (1 - np.exp(-1 / 24)))]:
    print(f'{name:30s} MMD^2 = {mmd2_unbiased(x_p, y, k):+.4f}  '
          f'(population {pop:.4f})')
```

```
linear,    N(0,1) vs N(0,4)    MMD^2 = -0.0033  (population 0.0000)
quadratic, N(0,1) vs N(0,4)    MMD^2 = +10.2618  (population 9.0000)
RBF,       N(0,1) vs N(0,4)    MMD^2 = +0.1180  (population 0.0942)
RBF,       N(0,1) vs N(0,1)    MMD^2 = +0.0001  (population 0.0000)
RBF,       N(0,1) vs N(0.5,1)  MMD^2 = +0.0351  (population 0.0471)
```

線形カーネルは 2 つの平均が一致しているため 0 に近い数値を返す。具体的には、その不偏推定値は 2 つの標本平均の差の二乗からその差の分散の推定値（およそ $(1+4)/1{,}000 = 0.005$）を引いたものに等しく、その結果、-0.005 を大きく下回ることはまれである一方、約 20 回に 1 回の実行で $+0.015$ を超える。二次カーネルは 4 乗が標本誤差を増幅するため、9 の前後 3 程度の範囲に落ち着く。RBF カーネルのもとでは、分散が異なるペアの推定値は約 0.094 ± 0.03、平均が異なるペアでは約 0.047 ± 0.02 となり、標本誤差が信号の無視できない割合を占めうる。しかし、**同じ**分布 $\mathcal{N}(0,1)$ からの 2 つの独立標本の間では、3 つの平均に含まれる 1 次の標本誤差が打ち消し合うため、RBF 推定値は驚くほどばらつきが小さく、0 から数千分の 1 以内に収まる（わずかに負になることもある）。同じ分布からの 2 標本間に全変動が 1 という見せかけの差を報告した場面でも、RBF カーネルは何の差も見出さないのである。

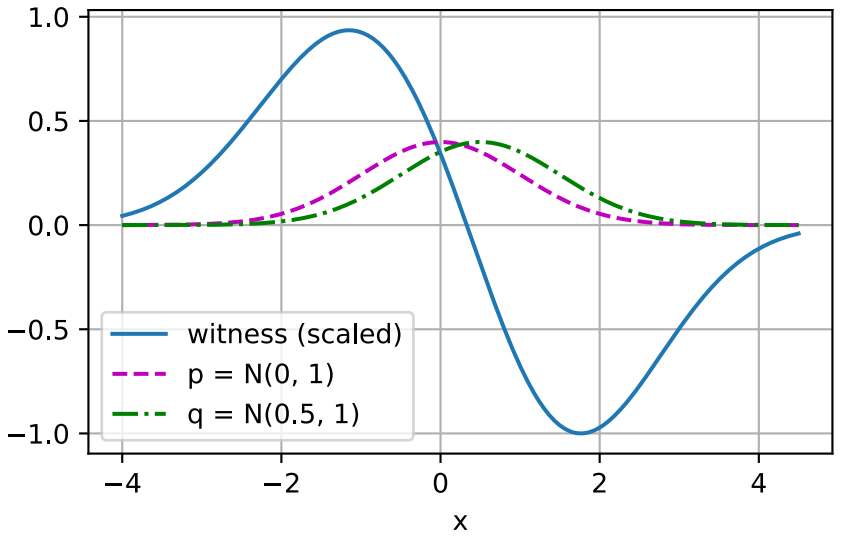
要するに、RBF 推定量は非ゼロの差を測るときには比較的大きな誤差を伴うものの、$P = Q$ のときには並外れた精度を発揮する。この性質により、RBF 推定量は強力な **2 標本検定**（two-sample test）となる。各標本あたり $1{,}000$ 点あれば、RBF の MMD^2 が 100 分の数程度であっても 2 つの分布が異なることの強い証拠となり、しかもクリティックの訓練も密度の評価も一切不要である。

37.4.1.4 証人関数

IPM は上限として定義されるため、実際にどのテスト関数が最大の差を達成したのかを問うのは自然である。MMD の場合、Equation 37.36 の上限は $\mathbf{w}^\star \propto \mathbf{m}_P - \mathbf{m}_Q$ によって達成される。得られるテスト関数は**証人関数**（witness）と呼ばれ、$g^\star(x) \propto \langle \mathbf{m}_P - \mathbf{m}_Q, \varphi(x) \rangle = \mathbb{E}_P[k(x, X)] - \mathbb{E}_Q[k(x, Y)]$ となる。これもまたカーネルの評価しか必要としない。平行移動したガウス分布 $\mathcal{N}(0, 1)$ と $\mathcal{N}(0.5, 1)$ では、2 つの密度は $x = 0.25$ を挟んで鏡像の関係にあるため、母集団における証人関数はちょうど 0.25 で符号を変える。以下では、グリッド上でサンプルから証人関数を推定し、2 つの密度と並べてプロットする。あわせて、スケール前の証人関数に対する $\mathbb{E}_P g - \mathbb{E}_Q g$、すなわち $g = \langle \mathbf{m}_P - \mathbf{m}_Q, \varphi \rangle$ に対する値も出力する。この値は $\|\mathbf{m}_P - \mathbf{m}_Q\|^2 = \mathrm{MMD}^2$ に等しく、ここでは自己ペアを含めて推定している。

```
g = lambda s: (rbf(s[:, None], x_p[None, :]).mean(axis=1)    # the unscaled
               - rbf(s[:, None], y_shift[None, :]).mean(axis=1))  # witness
grid = np.linspace(-4.0, 4.5, 400)
witness = g(grid)
print(f'witness changes sign at x = {grid[np.argmax(witness < 0)]:.2f}')
print(f'E_P g - E_Q g = {g(x_p).mean() - g(y_shift).mean():.4f}')
d2l.plot(grid, [witness / np.abs(witness).max(), norm.pdf(grid),
                norm.pdf(grid, loc=0.5)], 'x', '',
         legend=['witness (scaled)', 'p = N(0, 1)', 'q = N(0.5, 1)'],
         figsize=(5, 3))
```

```
witness changes sign at x = 0.35
E_P g - E_Q g = 0.0360
```



予想どおり、証人関数は P の密度が高い領域で正、Q の密度が高い領域で負となり、$x = 0.25$ 付近で 0 を横切る（各標本 1,000 点では、実行ごとにゼロ交差点の位置が 0.1〜0.2 ほど変動する）。出力された差は前のセルの不偏推定値と約 0.001 以内で一致しており、わずかなずれは自己ペアに由来する。

証人関数のプロットの裾は、ゼロ交差点以上に重要な事実を物語っている。差 $\mathbb{E}_P[g] - \mathbb{E}_Q[g]$ を

縮めるには、生成モデルは各サンプルを g の上り坂の方向へ押し上げなければならないため、モデルのサンプル位置における証人関数の傾きこそが、そのサンプルが受け取る勾配信号となる。2 つの釣鐘曲線の間では証人関数は急峻であり、データへ向かう強い引力を生み出す。ところが裾の方へ長さスケールの数倍も離れると、証人関数は平坦になって 0 に近づき、遠く離れたモデルのサンプルには勾配がほとんど届かなくなる。冒頭の表の MMD^2 の列を、1 つのサンプルの視点から目の当たりにしているわけである。

したがって実用上、固定カーネル MMD は 2 つの異なる役割を担う。ノンパラメトリックな 2 標本検定としての役割と、敵対的最適化を回避する生成モデルの訓練目的関数としての役割である (Li et al. 2015)。たとえば、画像生成器の評価に用いられる Kernel Inception Distance（16.5 節）は、画像の深層特徴量に対して三次カーネルで不偏 MMD 推定量を適用したものにほかならない。長さスケール ℓ を慎重に選ぶことは不可欠であり、$\ell = 1$ では $d = 8$ における傾きが 7×10^{-5} まで崩壊してしまった（演習問題 1b ではより大きなスケールを調べる）。実際には、ℓ をサンプル間のペアごとの距離の中央値に設定するか、複数の長さスケールにわたって RBF カーネルを足し合わせることが多く、16.2 節で GAN の**生成器**を MMD で訓練する際にもこの戦略が用いられている。

37.4.2 最適輸送とワッサースタイン距離

冒頭の表には、決して平坦にならない列が 1 つだけあった。W_1 が一定の傾き 1 を保ち続けたのは、f ダイバージェンスが無視し、固定カーネルが局所的にしか捉えられない基礎空間の幾何学を取り込んでいるからである。W_1 のもとで点質量 $\delta_{0.01}$ が δ_0 に近いのは、単にその質量が距離 0.01 だけ移動すれば済むためである。この着想は 1781 年にまで遡る。当時 Gaspard Monge は、盛り土を同じ体積の穴へと最小の総仕事量で運ぶにはどうすればよいかを問うた。その最小仕事量こそ、2 つの分布がどれだけ離れているかを測る自然な尺度となる。

37.4.2.1 輸送計画

実直線上の 6 点 $0, 1, \dots, 5$ をサポートとし、P が左側に、Q が右側に集中している 2 つの離散分布を考えよう（Figure 37.10 左上）。P を Q の形に組み替えるには、各出発点 x から各到着点 y へどれだけ質量を動かすかを指定しなければならない。このような**輸送計画**（transport plan）とは、同時分布 γ（ペア (x, y) 上で定義され、周辺分布が P と Q に一致するもの）のことである。すなわち、各 x から出ていく質量の合計は $P(x)$ に等しく、各 y に到着する質量の合計は $Q(y)$ に等しい。確率論では、輸送計画は P と Q の**カップリング**とも呼ばれる。**1 次ワッサースタイン距離**（Wasserstein-1 distance。土砂移動距離、earth mover's distance とも呼ばれる）は、有効なすべての計画にわたる移動距離の期待値の最小値である。$\Pi(P, Q)$ を、有限な一次モーメントを持つ P と Q（$\mathbb{R}^n$ 上）のすべてのカップリングの集合と書くことにすれば、次式で与えられる。

$$W_1(P, Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}\big[\|x - y\|\big]. \tag{37.39}$$

添字の 1 は距離の 1 乗を表しており、距離の二乗を用いる版である W_2 もまもなく登場する。2 つの点質量 δ_0 と δ_d の間では、可能な計画は全質量を 0 から d へ移すただ 1 つしかないため、$W_1(\delta_0, \delta_d) = |d|$ となり、勾配は ± 1 である（$d \neq 0$ のとき）。冒頭の表の平行移動した 2 つのガウス分布では、すべての点を d だけ平行移動させるのがコスト d の有効な計画であり、$\mathbb{E}_\gamma|x - y| \geq |\mathbb{E}_\gamma[x - y]| = d$ が任意のカップリング γ に対して成り立つため、これより優れた計画は存在しない。いずれの場合も W_1 は間隔 d そのものを復元し、f ダイバージェンスに欠けていた明瞭な勾配信号を与える。

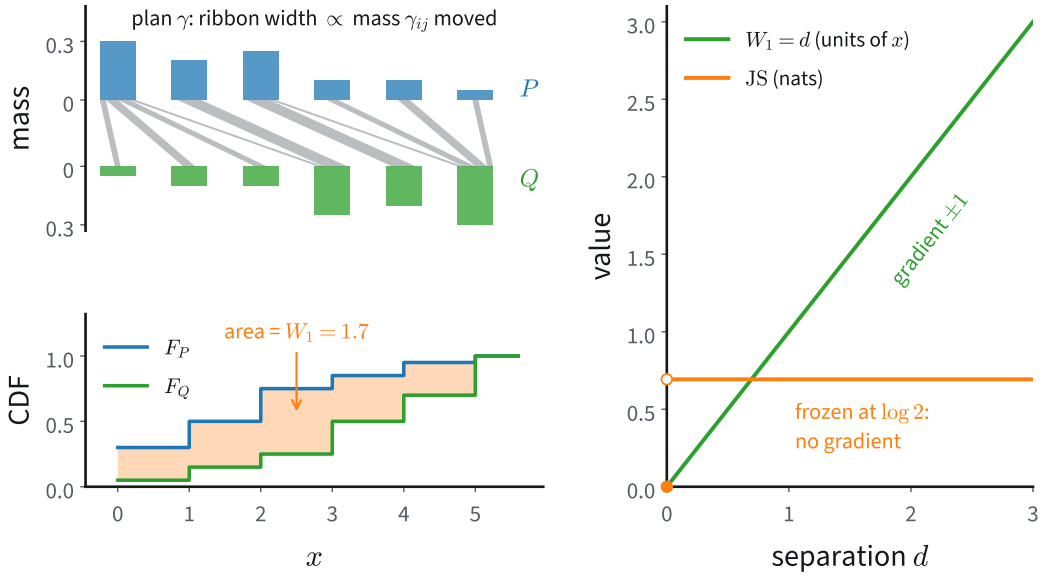


Figure 37.10: 直線上の輸送。左上：6 点上の 2 つの分布間の輸送計画。各リボンは P（上）の点から Q（下）の点へ質量を運び、その幅は移動する質量に比例する。左下：P と Q の CDF。両者の間の塗りつぶされた面積は 1.7 である。右：距離 d だけ離れた 2 つの点質量に対し、イェンセン・シャノン・ダイバージェンス（単位は nat）は $\log 2$ で平坦なままである一方、W_1（単位は x）は d に対して線形に増大する。

37.4.2.2 カントロヴィッチ・ルビンシュタイン双対性

計算手順として見ると、Equation 37.39 は手強そうに見える。同時分布 γ にわたって最小化しており、サンプルからパラメータ化することさえ厄介だからである。しかし有限点集合上では、これはおなじみの最適化問題、すなわち線形計画問題（linear program）にすぎない。未知変数は輸送計画の各成分であり、輸送コストも周辺分布の制約もそれらの成分に関して線形である。あらゆる線形計画問題には、制約に対する乗数に関して最大化する**双対**問題（dual problem）が存在する（35.7.1 節）。最適輸送（optimal transport）においては、この双対問題が同時分布にわたる探索を単一の関数に関する最大化へと置き換える。この再定式化こそが、深層学習において W_1 を実用的なものにしている。

命題（カントロヴィッチ・ルビンシュタイン双対性） 分布 P および Q（$\mathbb{R}^n$ 上で有限な一次モーメントを持つもの）に対し、

$$W_1(P,Q) = \sup_{\|g\|_{\mathrm{Lip}} \le 1} \left(\mathbb{E}_{x\sim P}[g(x)] - \mathbb{E}_{y\sim Q}[g(y)] \right), \tag{37.40}$$

が成り立つ。ここで上限はすべての 1 リプシッツ関数 g、すなわち $|g(x) - g(y)| \le \|x - y\|$ をすべての x, y について満たす関数にわたってとる。

カップリングにわたる下限（infimum）が、単一のテスト関数にわたる上限へと姿を変えた。すなわち W_1 は、先ほど予告した 3 つ目のクラスである傾き高々 1 の関数上の IPM（Equation 37.35）にほかならない。1 リプシッツな監査人は急激なジャンプができないため、指示関数のように個々のサンプル点を丸暗記することはできない。それでいて有界関数や RBF カーネルとは異なり際限なく増大できるため、第 2 の要件も満たしている。たとえば冒頭の表の平行移動した 2 つのガウ

ス分布では、$g(x) = -x$ は 1 リプシッツであり、期待値は 0（P のもと）と $-d$（Q のもと）になるため、差 $d = W_1(P, Q)$ を達成する。さらに都合のよいことに、モデルのサンプルがデータからどれだけ遠く離れていようとも、その勾配は至るところ大きさ 1 でデータの方向を指し示す。

ニューラルネットワークはそのようなテスト関数を近似できるため、**WGAN** (Arjovsky et al. 2017) は双対目的関数 Equation 37.40 を直接訓練するにあたり、近似的に 1 リプシッツとなるよう制約したニューラルクリティックを g の役割に据える。16.2 節ではこの目的関数を他の敵対的損失と比較し、勾配ペナルティ（gradient penalty）によって傾きを 1 に引き寄せるクリティックを訓練する。また 16.4 節では、後発の手法がペナルティの中心を 0 に置く理由を示す。傾き 1 を好むペナルティはすでにデータと一致している生成器までも押し動かし続けるのに対し、0 を中心とするペナルティはサンプル位置でのクリティックの勾配を小さく保つだけで、リプシッツ制約を課さないためである。なお、クリティックが近似的にしかリプシッツでなく、かつニューラルアーキテクチャに制限されているとき、その報告するスコアはもはや厳密な W_1 ではないことに留意しておこう。

37.4.2.3 直線上の輸送

1 次元の場合についてカントロヴィッチ・ルビンシュタイン双対性（Kantorovich–Rubinstein duality）を証明しよう。1 次元ではこの議論から、最適化を一切必要としない閉形式の公式も同時に得られる（高次元での証明は Peyré and Cuturi (2019) を参照）。直観的には、各点 $t \in \mathbb{R}$ をどれだけの質量が横切るかを数えるだけでよい。t の左側（t を含む）には、P が質量 $F_P(t)$ を持ち、Q が質量 $F_Q(t)$ を持つため、有効な輸送計画はいずれも少なくとも $|F_P(t) - F_Q(t)|$ の質量を t を越えて運ばなければならない。さらに、x から y へ運ばれる微小質量は、あらゆる点 t（x と y の間にあるもの）を横切るため、その移動距離 $|x - y|$ は横切る点からなる区間の長さに等しい。すべての微小質量について足し合わせれば、総輸送コストは t に関する積分（t を横切る総質量の積分）となる。

命題（直線上の W_1） 分布 P および Q（$\mathbb{R}$ 上で CDF F_P、F_Q と有限な一次モーメントを持つもの）に対し、

$$W_1(P, Q) = \int_{-\infty}^{\infty} |F_P(t) - F_Q(t)|\, dt = \sup_{\|g\|_{\mathrm{Lip}} \le 1} \big(\mathbb{E}_P[g] - \mathbb{E}_Q[g]\big), \tag{37.41}$$

が成り立つ。したがって、実直線上ではカントロヴィッチ・ルビンシュタイン双対性が成り立つ。

証明 ステップ 1：あらゆる計画のコストは少なくとも CDF 間の面積以上である。 $(X, Y) \sim \gamma$ に対し、$|X - Y| = \int \mathbf{1}(t \text{ lies between } X \text{ and } Y)\, dt$ と書く。被積分関数は非負であるため、期待値と積分の順序を交換できる。

$$\begin{aligned} \mathbb{E}_\gamma |X - Y| &= \int \Big(\gamma(X \le t < Y) + \gamma(Y \le t < X) \Big)\, dt \\ &\ge \int \big| \gamma(X \le t < Y) - \gamma(Y \le t < X) \big|\, dt. \end{aligned}$$

鍵となるのは、最後の積分の内部にある差を整理することである。$\gamma(X \le t, Y \le t)$ を足して引くと、$\gamma(X \le t < Y) - \gamma(Y \le t < X) = \gamma(X \le t) - \gamma(Y \le t) = F_P(t) - F_Q(t)$ となる（γ が周辺分布 P と Q を持つため）。したがって、あらゆる輸送計画のコストは少なくとも $\int |F_P - F_Q|$ 以上である。

ステップ 2：この下界を達成する計画が 1 つ存在する。 分位点を直接対応させよう。すなわち、P の下位 10% を Q の下位 10% へ送り、以下同様に直線上を上っていく。形式的に

は、U を $(0,1)$ 上の一様確率変数とし、$X = F_P^{-1}(U)$ および $Y = F_Q^{-1}(U)$ とおく。ここで分位点関数（quantile function）$F^{-1}(\lambda) = \inf\{x : F(x) \ge \lambda\}$（$\lambda \in (0,1)$）は、$F^{-1}(\lambda) \le t$ となるのが $\lambda \le F(t)$ のとき、かつそのときに限られるという性質を満たす。$\Pr(X \le t) = \Pr(U \le F_P(t)) = F_P(t)$ であり Y についても同様であるから、これは有効なカップリングを定める。X と Y がともに増加するため、これを**単調計画**（monotone plan）と呼ぶ。このカップリングのもとでは、$\gamma(X \le t < Y) = \Pr\big(F_Q(t) < U \le F_P(t)\big)$ および $\gamma(Y \le t < X) = \Pr\big(F_P(t) < U \le F_Q(t)\big)$ となる。すべての t において、これら 2 つの確率のうち少なくとも一方は 0 である。つまり単調計画は、同じ点を挟んで両方向に質量を送ることが決してない。したがってステップ 1 の不等式は等号で成立し、最小輸送コストは CDF 間の面積に等しい。

ステップ 3：いかなる 1 リプシッツなテスト関数も、どの計画のコストも上回らない。この**弱双対性**（weak duality）の議論は任意の次元で成り立つ。任意の 1 リプシッツ関数 g と任意のカップリング γ に対し、各点で $g(x) - g(y) \le \|x - y\|$ が成り立つため、γ のもとで期待値をとれば直ちに $\mathbb{E}_P[g] - \mathbb{E}_Q[g] \le \mathbb{E}_\gamma\|x - y\|$ が得られる（35.7.1 節の弱双対性と比較されたい）。

ステップ 4：この面積を達成する 1 リプシッツなテスト関数が 1 つ存在する。 $g(x) = g(0) + \int_0^x g'(t)\,dt$（$|g'| \le 1$）という形の関数だけを考えれば十分である。このような g はすべて 1 リプシッツであり、それ以外の関数もすでにステップ 3 で上から抑えられているからである。定数 $g(0)$ は $\mathbb{E}_P[g] - \mathbb{E}_Q[g]$ の中で相殺される。$\int_0^X g'(t)\,dt = \int_0^\infty g'(t)\,\mathbf{1}(X > t)\,dt - \int_{-\infty}^0 g'(t)\,\mathbf{1}(X \le t)\,dt$ と書き換え、期待値と積分の順序を交換すると（一次モーメントが有限であることから、34.6 節のフビニの定理により正当化される）、次式が得られる。

$$\mathbb{E}_P[g] - \mathbb{E}_Q[g] = \int_{-\infty}^{\infty} g'(t)\,\big(F_Q(t) - F_P(t)\big)\,dt.$$

$g'(t) = \operatorname{sign}\big(F_Q(t) - F_P(t)\big)$ と選べば、右辺は $\int |F_P(t) - F_Q(t)|\,dt$ となる。以上 4 つのステップを合わせると、1 リプシッツ関数全体にわたる上限は少なくとも面積以上であり（ステップ 4）、かつ最小計画コスト以下であるが（ステップ 3）、その最小計画コストは面積に等しい（ステップ 1 および 2）。■

結局のところ実直線上では、W_1 は 2 つの CDF に挟まれた面積にすぎず、同じサイズの 2 つの標本に対してはソートした値の平均絶対差に帰着する（演習問題 2a）。線形計画問題全体がこれほどきれいに簡約されるのは驚くべきことである。この簡約を Figure 37.10 の 6 点の例で確かめてみよう。隣接するアトム間の 5 つの区間において、CDF の差は 0.25、0.35、0.5、0.35、0.25 であり、その和は図の塗りつぶされた面積である 1.7 となる。以下のコードでは 3 通りの方法で W_1 を計算する。5 つの CDF の差を足し合わせる方法、変数 36 個の線形計画問題（6×6 輸送行列のうち行和が p、列和が q となるものの上での問題）を解く方法、そしてステップ 2 の単調計画（各経路が対応する分位点区間の重なりを運ぶ計画）を構成してそのコストを評価する方法である。最後に、線形計画法が単調計画を返したかどうかも確認する。

```
atoms = np.array([0.0, 1.0, 2.0, 3.0, 4.0, 5.0])
p_w = np.array([0.30, 0.20, 0.25, 0.10, 0.10, 0.05])
q_w = np.array([0.05, 0.10, 0.10, 0.25, 0.20, 0.30])

F_p, F_q = np.cumsum(p_w), np.cumsum(q_w)
w1_cdf = np.sum(np.abs(F_p - F_q)[:-1] * np.diff(atoms))

C = np.abs(atoms[:, None] - atoms[None, :])      # cost matrix |x_i - y_j|
```

```
    n_atoms = len(atoms)
    A_eq = np.zeros((2 * n_atoms, n_atoms * n_atoms))
    for i in range(n_atoms):
        A_eq[i, i * n_atoms:(i + 1) * n_atoms] = 1   # row sums    = p
        A_eq[n_atoms + i, i::n_atoms] = 1            # column sums = q
    res = optimize.linprog(C.ravel(), A_eq=A_eq,
                           b_eq=np.concatenate([p_w, q_w]), method='highs')
    plan_lp = res.x.reshape(n_atoms, n_atoms)
    # Monotone plan (Step 2): source i holds the quantiles (F_p[i-1], F_p[i]],
    # destination j holds (F_q[j-1], F_q[j]]; route i -> j carries their overlap
    lo_p, lo_q = F_p - p_w, F_q - q_w
    mono = np.clip(np.minimum(F_p[:, None], F_q[None, :])
                   - np.maximum(lo_p[:, None], lo_q[None, :]), 0, None)
    print(f'W1 from the CDF gaps     : {w1_cdf:.4f}')
    print(f'W1 from the LP           : {res.fun:.4f}')
    print(f'cost of the monotone plan: {np.sum(mono * C):.4f}')
    print(f'LP plan = monotone plan  : {np.allclose(plan_lp, mono)}, '
          f'mass the LP plan moves leftward: {np.tril(plan_lp, -1).sum():.2f}')
```

```
W1 from the CDF gaps     : 1.7000
W1 from the LP           : 1.7000
cost of the monotone plan: 1.7000
LP plan = monotone plan  : False, mass the LP plan moves leftward: 0.00
```

3 つの方法はすべて $W_1 = 1.7$ で一致する。5 つの CDF の差を足し合わせるだけで、36 変数の線形計画問題を解いた場合と寸分違わぬ答えが得られるのである。ところが最後の行には小さな驚きがある。線形計画法は単調計画を返さなかったのである。その代わりに、まったく同じコストを持ち、同じく左方向へは一切質量を動かさない別の計画を見つけた。

つまり、この例では最適な W_1 計画は一意ではなく、その理由も見抜くのは容易である。至るところで $F_P \geq F_Q$ であるため、質量を左方向へ動かす必要がそもそもない。質量を右方向（$y \geq x$）にのみ動かす**いかなる**計画に対しても $|x-y| = y - x$ が成り立つため、経路がどのように交差しようとも、総コストは 2 つの平均の差 $3.35 - 1.65 = 1.7$ に等しくなる。図に描かれた単調計画は、同等に優れた多数の計画の 1 つにすぎない。

本書の他の箇所に登場する 2 つの手法では、絶対距離をその二乗に置き換える。画像生成器の評価に用いる Fréchet Inception Distance（FID）（16.5 節）と、二乗距離を最小化してノイズとデータを組にするフローマッチングのミニバッチ最適輸送（38.7 節）である。**2 次ワッサースタイン距離**（Wasserstein-2 distance）は、二乗距離の平均をとってから平方根をとった量 $W_2(P,Q) = \left(\inf_{\gamma} \mathbb{E}_{\gamma}\|x-y\|^2\right)^{1/2}$ である。二乗コストのもとでは、交差する 2 本の輸送リボンは交差を解くよりも常に真に大きなコストがかかるため、実直線上では単調計画が**唯一の**最適解となる（演習問題 2d）。平均 μ_1, μ_2、標準偏差 σ_1, σ_2 を持つ 2 つの 1 次元ガウス分布の間では、この距離は $W_2^2 = (\mu_1 - \mu_2)^2 + (\sigma_1 - \sigma_2)^2$ ときれいに求まる（演習問題 4a）。Fréchet Inception Distance はこの公式の多変量版であり、画像の特徴量に当てはめた多変量ガウス分布に対して分散を共分散行列に置き換えて適用したものである。

37.4.2.4 エントロピー正則化とシンクホーン反復

本項は初読時には読み飛ばしてもよい。1 次元を超えると一般に閉形式の CDF 公式は存在せず、両側に N 点ずつある線形計画問題は N^2 個の変数を持つ。最適輸送を高速かつ微分可能にする標準的な手法が**エントロピー正則化**（entropic regularization）である (Cuturi 2013)。コスト行列 $\mathbf{C}$ と正則化強度 $\varepsilon > 0$ が与えられたとき、$\varepsilon \sum_{ij} \gamma_{ij}(\log \gamma_{ij} - 1)$ を輸送コスト $\sum_{ij} \gamma_{ij} C_{ij}$

に加える。有効な計画はすべて $\sum_{ij}\gamma_{ij}=1$ を満たすため、この追加項は $-\varepsilon$ と γ のエントロピーの積に定数を足したものに等しく、少数の鋭い経路に全質量を集中させる計画を抑制する。エントロピー正則化こそが最適輸送を深層学習の実用的な損失関数へと変えた立役者であり、正則化輸送コストを用いて生成モデルを直接訓練することを可能にした (Genevay et al. 2018)。

驚くべきことに、正則化された問題はほぼ閉形式で解ける。ラグランジアン（35.7.1 節）を、乗数 ξ_i（各行和制約 $\sum_j\gamma_{ij}=p_i$ に対応）および乗数 ζ_j（各列和制約 $\sum_i\gamma_{ij}=q_j$ に対応）を用いて作り、γ_{ij} に関する導関数を 0 とおくと、$C_{ij}+\varepsilon\log\gamma_{ij}-\xi_i-\zeta_j=0$、すなわち $\gamma_{ij}=e^{-(C_{ij}-\xi_i-\zeta_j)/\varepsilon}$ が得られる。各経路が運ぶ質量は、そのコスト C_{ij} が $\xi_i+\zeta_j$ をどれだけ上回るかに応じて（ε を単位として）指数関数的に減衰する。行列形式で書けば次のようになる。

$$\gamma_\varepsilon=\operatorname{diag}(\mathbf{a})\,\mathbf{K}\,\operatorname{diag}(\mathbf{b}),\qquad \mathbf{K}=e^{-\mathbf{C}/\varepsilon}\ \text{(entrywise)},\tag{37.42}$$

ここで $\mathbf{a}=e^{\xi/\varepsilon}$ と $\mathbf{b}=e^{\zeta/\varepsilon}$ は正のスケーリングベクトルである。正則化輸送計画は、単なる**ギブスカーネル** $\mathbf{K}$（正の重みからなる行列であり、MMD のカーネルとは無関係である）を行ごと・列ごとにリスケーリングしたものにすぎない。求めるべき未知数は $\mathbf{a}$ と $\mathbf{b}$ だけであり、それぞれの周辺制約が一方を決定する。すなわち、$\mathbf{b}$ を固定すれば成分ごとに $\mathbf{a}=\mathbf{p}/(\mathbf{K}\mathbf{b})$ とおくことで行和 $\mathbf{p}$ が満たされ、$\mathbf{a}$ を固定すれば $\mathbf{b}=\mathbf{q}/(\mathbf{K}^\top\mathbf{a})$ とおくことで列和 $\mathbf{q}$ が満たされる。

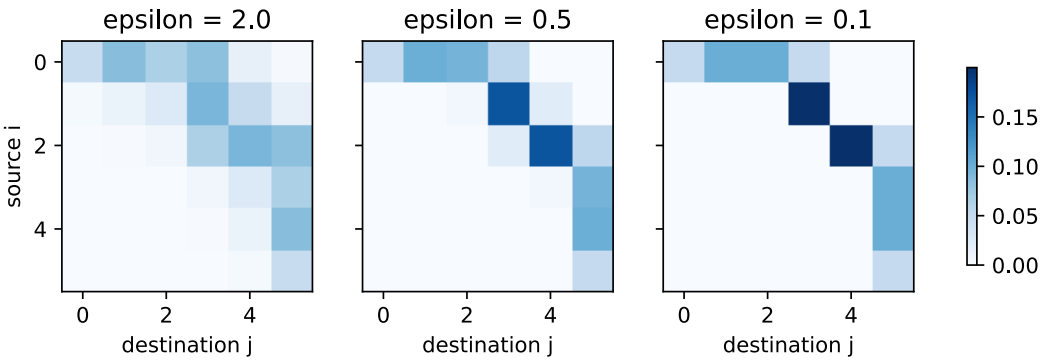
シンクホーン反復（Sinkhorn iterations）はこの 2 つの更新を交互に繰り返し、両方の周辺分布が一致するまで $\mathbf{b}$、次いで $\mathbf{a}$ を更新する。$\mathbf{a}$ の更新直後には行和は厳密に一致しているため、列和が $\mathbf{q}$ と誤差 10^{-9} 以内で一致した時点で停止する。各ステップで必要なのは行列・ベクトル積が 2 回だけであり、GPU 上で高速に動作するうえ、反復を通じた微分も容易である。以下では、正則化計画の（エントロピーペナルティを除いた）輸送コスト $\sum_{ij}(\gamma_\varepsilon)_{ij}C_{ij}$ を指して**エントロピーコスト**（entropic cost）として報告する。$\varepsilon\to0$ とすると、このコストは正則化なしの厳密な最適値に近づくが、収束が劇的に遅くなり (Peyré and Cuturi 2019)、行列 $\mathbf{K}$ の成分がアンダーフローを起こしうる。そのため実用上の実装では、対数空間で $\log\mathbf{a}$ と $\log\mathbf{b}$ を更新する（演習問題 5b）。

ε をどれほど小さくする必要があり、また小さな ε がどれほど高コストになるかを確かめるため、二乗コスト $C_{ij}=(x_i-x_j)^2$ を用いた 6 点の例にシンクホーン反復を適用しよう。この設定では Figure 37.10 の単調計画が唯一の最適解であり、10 本の有効な経路のコストは $W_2^2=3.5$ である。以下のセルでは線形計画問題でこの最適値を確認したあと、3 つの ε の値でシンクホーン反復を実行し、反復回数、エントロピーコスト、および 10^{-3} を超える質量を運ぶ経路の本数を報告する。

```
def sinkhorn(p, q, C, eps, tol=1e-9, max_iters=10 ** 6):
    """Entropic OT by Sinkhorn iterations, stopped once the column sums are
    within tol of q (L1); returns the plan, its cost and the iterations."""
    K = np.exp(-C / eps)                        # the Gibbs kernel
    a = np.ones_like(p)
    for it in range(1, max_iters + 1):
        b = q / (K.T @ a)                       # match the column sums
        a = p / (K @ b)                         # match the row sums ...
        if np.abs(b * (K.T @ a) - q).sum() < tol:  # ... columns still right?
            break
    plan = a[:, None] * K * b[None, :]
    return plan, np.sum(plan * C), it

C2 = (atoms[:, None] - atoms[None, :]) ** 2     # squared cost (x_i - y_j)^2
```

```
res2 = optimize.linprog(C2.ravel(), A_eq=A_eq,
                        b_eq=np.concatenate([p_w, q_w]), method='highs')
print(f'squared cost, LP optimum : {res2.fun:.5f}')
plans, eps_values = [], (2.0, 0.5, 0.1)
for eps in eps_values:
    plan, cost, iters = sinkhorn(p_w, q_w, C2, eps)
    plans.append(plan)
    print(f'epsilon = {eps}: {iters:6d} iterations, entropic cost = '
          f'{cost:.5f}, routes above 1e-3 = {(plan > 1e-3).sum()}')
d2l.show_heatmaps(d2l.tensor(np.stack(plans)[None]), 'destination j',
                  'source i', titles=[f'epsilon = {e}' for e in eps_values],
                  figsize=(9, 2.8), cmap='Blues')
```

```
squared cost, LP optimum : 3.50000
epsilon = 2.0:     47 iterations, entropic cost = 4.06489, routes above 1e-3 = 25
epsilon = 0.5:    186 iterations, entropic cost = 3.56963, routes above 1e-3 = 14
epsilon = 0.1: 154325 iterations, entropic cost = 3.50002, routes above 1e-3 = 10
```



線形計画法により $W_2^2 = 3.5$ が確認できる。$\varepsilon = 2$ では、正則化計画は各出発点の質量を多数の到着先ににじませ（25 本の経路が全 36 本中 10^{-3} 超の質量を運ぶ）、コストを 4.06 まで押し上げる。$\varepsilon = 0.5$ では有効な経路が 14 本に絞られ、コストは 3.57 となる。$\varepsilon = 0.1$ になると単調計画の 10 本の経路だけが生き残り、コストは 3.50002 に達して厳密な最適値との差が 2×10^{-5} 以内に収まる。ヒートマップでは、ぼやけた帯がくっきりとした階段状のパターンへと鋭さを増していく。ε の自然なスケールは競合する経路間のコスト差であり、このグリッド上ではオーダー 1 である。ε をそのスケールの 10 分の 1 に設定すれば、最適値を精度よく復元できる。

反復回数を見ると、その代償が明らかになる。シンクホーン反復が要した回数は約 50 回（$\varepsilon = 2$）および約 190 回（$\varepsilon = 0.5$）であったのに対し、約 150,000 回（$\varepsilon = 0.1$）へと跳ね上がった。ε を 5 分の 1 に縮めただけで、計算量は 800 倍以上に膨れ上がったのである。シンクホーン反復が高速なのは、ぼやけた輸送計画を許容できる場合に限られる（演習問題 6b でこの増大を定量的に追跡する）。

37.4.3 固定カーネルと訓練されたクリティック

MMD と W_1 は、2 つの要件をちょうど真っ二つに分け合っている。固定カーネルは小さく滑らかな関数クラスを定め、MMD は 3 つの標本平均に帰着し、N 個の点からの推定誤差は**いかなる**次元においても $O(N^{-1/2})$ で減衰する (Gretton et al. 2012)。残念ながら、だからといってカーネルが次元の呪い（curse of dimensionality）を免れるわけではない。誤差の収束レートは次元に依存しないものの、信号の大きさは次元に依存するからである。固定カーネルのもとでは、与え

られた 2 分布間の母集団 MMD は次元が高くなるにつれて縮小するのが一般的であり (Ramdas et al. 2015)、Kernel Inception Distance や MMD 損失が生のピクセル上ではなく学習された特徴量上で計算される理由はそこにある。また固定カーネルは、長さスケールの数倍を超えるとデータを見失ってしまう。これとは対照的に、1 リプシッツ関数はあらゆる間隔で安定した勾配を与え、まさにデータから遠く離れたモデルが必要とする性質を備えている。しかし、指示関数のように不連続なジャンプはできないとはいえ、すべての 1 リプシッツ関数からなるクラスは高次元では依然として膨大である。次元 $n > 2$ における 2 標本間では、経験 W_1 の誤差は $O(N^{-1/n})$ でしか減衰しない (Weed and Bach 2019)。100 次元空間では、その推定誤差を半分にするのに 2^{100} 倍ものデータが必要になる計算である。高次元のサンプルからそのように直接 W_1 を推定する者はいない。代わりに WGAN のクリティックははるかに小さなニューラルネットワークのクラス上を探索し、シンクホーン反復はより小規模な離散マッチング問題を扱うのである。

カーネルも敵対的クリティックも、まずはモデルからサンプルを抽出することから始まる。生成モデルによってはサンプリング自体がボトルネックとなる場合があり、続く 37.5 節では、モデルからサンプルを一切抽出せずに 2 つの分布を比較する方法を問う。

37.4.4 まとめ

裾の重なりは f ダイバージェンスを救わない。2 つの単位ガウス分布が標準偏差の 8 倍離れると、イェンセン・シャノン・ダイバージェンスと全変動の傾きはいずれも 10^{-4} 付近まで落ち込む。積分確率指標は、テスト関数のクラスを P と Q がどう平均するかを比較することで密度比を回避し、どのクラスを選ぶかによってサンプル効率と距離が見渡せる範囲の両方が決まる。固定された特徴に関して線形なテスト関数からは、平均特徴ベクトル間の距離である MMD が得られる。MMD はクリティックを訓練することなく 3 つのカーネル平均として閉形式で評価でき、次元によらず $O(N^{-1/2})$ のレートで推定できるが、次元が増すにつれて MMD 自体が縮小する傾向があり、また RBF カーネルは長さスケールの数倍を超えると盲目になる。傾きが高々 1 の関数からは W_1、すなわち P を Q へ輸送する「質量 × 距離」の最小コストが得られ、これはあらゆる間隔で傾き 1 を維持し、Wasserstein GAN のクリティックの基盤をなす。1 次元では W_1 は 2 つの CDF に挟まれた面積に帰着し、高次元では訓練されたクリティックを用いるか、小規模な離散問題に対してエントロピー正則化付きのシンクホーン反復を用いる必要がある。

37.4.5 演習問題

1. [code] **閉形式の MMD** ガウス積分 Equation 37.38 を長さスケール ℓ の RBF カーネルに適用し、$P = \mathcal{N}(0, 1)$ および $Q = \mathcal{N}(d, 1)$ とせよ。
    1. $\text{MMD}^2 = \frac{2\ell}{\sqrt{\ell^2+2}}\big(1 - e^{-d^2/(2(\ell^2+2))}\big)$ となることを示し、$\ell = 1$ のとき冒頭の表の MMD^2 の列と一致することを確かめよ。小さなシフト d に対して MMD^2 を最大化する長さスケールはいくらか。また $\ell \to 0$ および $\ell \to \infty$ のときどうなるか。
    2. d に関する傾きを $d = 8$ において $\ell = 1$、4、16 について求めよ。得られた 3 つの値は、データから遠く離れた位置から出発するモデルの長さスケールについて何を物語っているか。
2. [code] **1 次元の輸送●**実直線上では、W_1 は CDF 間の面積 Equation 37.41 である。
    1. 同じサイズの標本 $x_1, \dots, x_N$ と $y_1, \dots, y_N$ 上の 2 つの経験分布に対し、証明中の単調計画を用いて $W_1 = \frac{1}{N}\sum_i |x_{(i)} - y_{(i)}|$（ソートした値の平均絶対差）となることを示せ。サイズ 7 の 2 つの無作為標本において、6 点の例の線形計画問題と照合してこれを確認せよ。
    2. MMD の実験における `x_p` と `y_shift` の間、および `x_p` と `y_same` の間で、ソート標本による W_1 を計算せよ。次に、20 組の独立標本のペア（各サイズ N、$\mathcal{N}(0, 1)$ から

抽出）にわたって $N \in \{10^2, 10^3, 10^4\}$ のそれぞれで平均をとり、その減少レートを求めよ。この推定値は決して負にならない。$P = Q$ のとき、なぜこのことから不偏性が否定されるのか。

3. `q_w` を $(0.40, 0.05, 0.05, 0.10, 0.10, 0.30)$ に変更して 2 つの CDF が交差するようにし、W_1 を計算するセルを再実行せよ。証明のステップ 4 と同様に、テスト関数 g を $g(0) = 0$ かつ隣接アトム間の傾きが $\operatorname{sign}(F_Q(t) - F_P(t))$ となるように構成し、$\mathbb{E}_P[g] - \mathbb{E}_Q[g]$ が LP の値と一致することを確かめよ。LP の解 `plan_lp` において質量を運ぶすべての経路 (i, j) について、$g(x_i) - g(x_j)$ と $|x_i - x_j|$ を比較せよ。任意の最適計画と任意の最適テスト関数において、なぜ質量を運ぶすべての経路で両者が一致しなければならないのか。
4. **二乗コストのもとでの一意性**。ある計画が x から y' へ、かつ x' から y へ質量を動かしており、$x < x'$ かつ $y < y'$ であるため 2 本のリボンが交差しているとする。二乗コストのもとでは、同じ質量を x から y へ、x' から y' へ送ることで単位質量あたり $2(x' - x)(y' - y)$ だけコストが節約されることを示し、単調計画が唯一の最適計画であることを導け。4 つの移動がすべて右向きであるとき、コスト $|x - y|$ のもとで同じ入れ替えはどれだけコストを節約するか。6 点の例に対して線形計画法が返した計画 `plan_lp` の中から交差する 2 本のリボンを見つけ、自分の答えを確認せよ。

3. **直線上の最適クリティック** Equation 37.41 の証明のステップ 4 は、傾き $g'(t) = \operatorname{sign}(F_Q(t) - F_P(t))$ を持つテスト関数が W_1 を達成することを示している。

1. $P = \mathcal{N}(0, 1)$ および $Q = \mathcal{N}(d, 1)$（$d > 0$）に対し、至るところで $F_P > F_Q$ となり、ステップ 4 が本文で直観的に見出したクリティック $g(x) = -x$ を返すことを示せ。
2. $P = \mathcal{N}(0, 1)$ および $Q = \mathcal{N}(0, 4)$ に対し、2 つの CDF が 0 でのみ交差すること、$g(x) = -|x|$ が最適であること、および $W_1 = \sqrt{2/\pi} \approx 0.80$ となることを示せ（$X \sim \mathcal{N}(0, \sigma^2)$ に対して $\mathbb{E}|X| = \sigma\sqrt{2/\pi}$ である）。このペアに対して線形カーネルは 0 を報告していた。このクリティックは Q のサンプルをどちらの方向へ動かすか。
3. $P = \frac{1}{2}\delta_0 + \frac{1}{2}\delta_3$、$Q = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_3$ とする。$W_1(P, Q)$ を求め、ステップ 4 で用いた形の最適クリティックをすべて記述せよ。さらに一般に、そのようなクリティックが最適となるのは、$g'(t) = \operatorname{sign}(F_Q(t) - F_P(t))$ がほとんどすべての t（質量が横切らなければならない $F_P(t) \neq F_Q(t)$ の点）で成り立つとき、かつそのときに限られ、それ以外の点での傾きは任意であることを示せ。
4. WGAN のクリティックは、データと生成サンプルの間の点における $(|g'(x)| - 1)^2$（点 x で評価）にペナルティを課して訓練されることが多い (Gulrajani et al. 2017)。この選択を (c) を用いて説明せよ。$P = Q$ のときどのクリティックが最適となり、このペナルティはそのうちのどれを選好するか。訓練においてその帰結がどうなるかは 16.4 節で示される。

4. [code] **ガウス分布間の W_2 とフレシェ距離**実直線上では、単調計画 $X = F_P^{-1}(U)$、$Y = F_Q^{-1}(U)$ が二乗コストに対して最適である。

1. $P = \mathcal{N}(\mu_1, \sigma_1^2)$ と $Q = \mathcal{N}(\mu_2, \sigma_2^2)$ に対し、単調計画が $Y = \mu_2 + (\sigma_2/\sigma_1)(X - \mu_1)$ で与えられることを示し、$W_2^2(P, Q) = (\mu_1 - \mu_2)^2 + (\sigma_1 - \sigma_2)^2$ を導け。同じ計画を用いて、等分散の場合に $W_1 = |\mu_1 - \mu_2|$（冒頭の表の W_1 の列）を復元せよ。
2. 両方の標準偏差を σ、平均の間隔を d とする。このとき $W_2^2 = d^2$ がすべての σ に対して成り立つ一方、2 つのガウス分布間の KL ダイバージェンスは $d^2/(2\sigma^2)$ である。$\sigma \to 0$ としてガウス分布が Table 37.2 の点質量へと近づくとき、両者はそれぞれ何を報告するか。
3. $\mathbb{R}^n$ において、W_2 の二乗（$\mathcal{N}(\mu_1, \Sigma_1)$ と $\mathcal{N}(\mu_2, \Sigma_2)$ の間）は、二乗**フレシェ距離**

$\|\mu_1-\mu_2\|^2+\mathrm{tr}\big(\Sigma_1+\Sigma_2-2\,(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\big)$ に等しい (Dowson and Landau 1982)。$n=1$ のとき (a) に帰着すること、および共分散行列が対角行列のとき各座標にわたる (a) の和に帰着することを確かめよ。さらに、対角行列の場合を直接証明せよ（二乗コストは座標ごとの和であるため、いかなる計画のコストも座標ごとの最適値の和を下回らず、各座標の単調計画を組み合わせればその下界を達成する）。

4. Fréchet Inception Distance（FID）は 2 つの特徴量分布のそれぞれにガウス分布を当てはめ、当てはめた分布間の二乗フレシェ距離を報告する。P を混合分布 $0.7\,\mathcal{N}(-2,0.6^2)+0.3\,\mathcal{N}(2,0.6^2)$ とし、同じ平均と分散を持つ $Q=\mathcal{N}(-0.8,3.72)$ を考える。当てはめた 2 つのガウス分布間のフレシェ距離はいくらか。それぞれの分布から抽出した 10^5 個のソート済みサンプルから $W_2^2(P,Q)$ を推定せよ（約 0.63 になるはずである）。フレシェ距離が 0 であることは何を保証し、何を保証しないか。

5. [code] **エントロピー輸送とシンクホーンダイバージェンス●●**6 点の例の輸送 LP を、独立な計画からのダイバージェンスによって次のように正則化する。

$$\mathrm{OT}_\varepsilon(P,Q)=\min_{\gamma}\sum_{i,j}\gamma_{ij}C_{ij}+\varepsilon\,D_{\mathrm{KL}}(\gamma\,\|\,p\,q^\top),$$

ここで γ は行和 p、列和 q を持つ計画全体を動く。

1. このような任意の計画に対して $D_{\mathrm{KL}}(\gamma\,\|\,p\,q^\top)=-H(\gamma)+H(P)+H(Q)$（$H$ はエントロピー）が成り立ち、したがって `sinkhorn` が最小点を返すことを示せ。また $\mathrm{OT}_\varepsilon(P,Q)\to\sum_{i,j}p_iq_jC_{ij}$（$\varepsilon\to\infty$ のとき）となること、および $\mathrm{OT}_\varepsilon(P,P)>0$ がすべての $\varepsilon>0$ に対して（P が点質量でない限り）成り立つことを示せ。$C_{ij}=|x_i-x_j|$ のもとで、アトム上の分布 P のうち極限 $\sum_{i,j}p_iq_jC_{ij}$ を最小化するものは Q の中央値（`q_w` ではアトム 3 と 4）に全質量を集中させることを示せ。大きな ε を用いた OT_ε で訓練されたモデルは何を学習するか。
2. シンクホーンダイバージェンス $S_\varepsilon(P,Q)=\mathrm{OT}_\varepsilon(P,Q)-\tfrac12\mathrm{OT}_\varepsilon(P,P)-\tfrac12\mathrm{OT}_\varepsilon(Q,Q)$ はこの統計的バイアスを取り除く (Genevay et al. 2018; Feydy et al. 2019)。`p_w` と `q_w` に対し、絶対値コスト `C` のもとで $\varepsilon\in\{10^{-3},10^{-2},\dots,10^{3}\}$ における $S_\varepsilon(P,Q)$ を計算せよ。$\varepsilon=10^{-3}$ では $\mathbf{K}$ の成分がアンダーフローを起こすため、そこでは `scipy.special.logsumexp` を用いて $\log\mathbf{a}$ と $\log\mathbf{b}$ 上でシンクホーン更新を実行せよ。S_ε が W_1 に近づき（$\varepsilon\to0$ のとき）、また $\tfrac12\mathcal{E}(P,Q)$ に近づく（$\varepsilon\to\infty$ のとき）ことを確認せよ。ここで $\mathcal{E}(P,Q)=2\mathbb{E}|X-Y|-\mathbb{E}|X-X'|-\mathbb{E}|Y-Y'|$ はエネルギー距離 (Székely and Rizzo 2013) であり、Equation 37.37 のカーネルを $-|x-y|$ に置き換えた式である。2 つ目の極限を (a) から導出せよ。

6. [code] **小さな ε におけるシンクホーン反復●**6 点の例の `sinkhorn`、`p_w`、`q_w` を用いよ。

1. **早期停止**。二乗コスト `C2`、$\varepsilon=0.1$、`max_iters=5000` で `sinkhorn` を呼び出し、列和が収束する前にループが終了するようにせよ。返されるコストは最適値 3.5 を下回る。返された行列の列和から `q_w` を引いた値を出力せよ。どの到着先への質量が不足しており、なぜそれによってコストが下がるのか。行和 `p_w` と列和 `q_w` を持つ行列のコストが 3.5 未満になり得ないのはなぜか。
2. **反復回数**。`sinkhorn` が `C2` のもとで $\varepsilon\in\{2,1,0.5,0.3,0.2,0.15,0.1\}$ に対して要する反復回数を数え、回数の対数を $1/\varepsilon$ に対してプロットせよ。小さな ε における点列はどのような増大を示し、$\varepsilon=0.05$ に対していくつの反復回数を予測するか。
3. **同値な最適解**。絶対値コスト `C` のもとでは、質量を右方向にのみ動かすあらゆる計画が `p_w` と `q_w` に対して最適となる。`sinkhorn` を `C` のもとで $\varepsilon=0.1$ および 0.02 で実

行せよ。何本の経路が質量を運ぶか。極限の計画が $\gamma_{ij} = a_i b_j$（$j \geq i$ のとき）および 0（それ以外）という形を持つことを示せ。有効な計画上では、正則化項は $-\varepsilon$ とエントロピーの積に定数を足したものである。これを用いて、なぜ極限の計画が最適計画の中で最大エントロピーを持つ計画になるのかを説明せよ。

37.5 スコアに基づく比較

ソフトマックス層は 10 個のロジットの指数をとってその和で割ることで確率へと変換するが、その和の計算に頭を悩ませる者はいない。では、ニューラルネットワークに 10 個のクラスではなく、起こり得るすべての入力に対してスコアを割り当てさせる場合を想像してみよう。慣例により符号は反転させる。ネットワークはもっともらしい入力（ロジットであれば高くなる場所）には低く、もっともらしさの低い入力には高くなるエネルギー $E(\mathbf{x})$ を出力し、$\tilde{q}(\mathbf{x}) = e^{-E(\mathbf{x})}$ に比例する密度を定める。すると 10 個のクラスにわたる和は、起こり得るすべての入力にわたる積分 $Z = \int \tilde{q}(\mathbf{x})\, d\mathbf{x}$ へと置き換わるが、高次元においてこの積分はまったく計算困難である。このような**エネルギーベースモデル**は、ある入力が別の入力に比べてどれだけもっともらしいかは教えてくれるが、その実際の確率は教えてくれない。

これまでに構築してきた分布の比較手法は、いずれもこの状況に対処できない。最尤法には正規化された密度 $q = \tilde{q}/Z$ が必要である。クリティック（37.3.4 節）、カーネル、輸送計画（37.4 節）のいずれに基づくにせよ、サンプルに基づく手法は Z を回避できるが、モデルからサンプルを抽出する必要があり、エネルギー関数はそのままではサンプルを与えてくれない。各サンプルは、より低いエネルギーへ向けてノイズを伴う小さなステップを多数繰り返す長いランダムウォークによって生成しなければならず、その各ステップでネットワークを通じた勾配計算が必要になる（38.6.2 節）。

しかし、勾配を 1 回計算するだけなら安価である。必要なコストはわずか 1 回の逆伝播であり、実はそれだけで十分なのである。$\log \tilde{q}$ の入力 $\mathbf{x}$ に関する勾配は対数密度の傾きを測るものであり、定数の傾きは 0 であるため Z を完全に消去する。この勾配は**スコア**（またはスコア関数）と呼ばれ、本節ではスコアのみを用いて分布を比較する。2 つのスコア間の平均二乗差はスコアマッチングの目的関数を与え、これは第 17 章で扱う拡散モデル（diffusion model）の基盤をなしている。とはいえ、傾きの比較には、スコアに基づくモデルを訓練する者なら誰もが理解しておくべき重要な盲点が伴う。傾きは純粋に局所的な量であるため、遠く離れたモード間で確率質量がどう配分されているかを知ることができないのである。この盲点がどれほど深刻かを以下で定量化していく。

前節までと同様、サンプルを通じてのみ観測されるデータ分布を P と表し、モデルを Q と表す。ただし本節では、モデルについてはそのスコア s_Q のみが既知であるとする。ここで 2 つの中心的な問いを立てる。傾きの差は 2 つの分布について何を教え（何を見落とす）のか、そしてそもそもデータが有限個の点集合であって傾きを持たないとき、どうやって傾きどうしを比較できるのか、という問いである。38.6.1 節はこの 2 つの答えの上に直接築かれる。また 3 つ目の問いとして、サンプリングできないモデルから与えられたサンプルが生成されたかどうかを検定する方法も探るが、この部分は初読時には飛ばしても構わない。最後に、本章で扱ったすべての比較手法を訓練目的関数の表へとまとめ、それらをどう使い分けるかについての実践的な指針を示す。

```
from scipy import integrate
```

本節の通し例は、1 次元のガウス分布および 2 つのガウス分布の混合分布である。まずは 3 つの小さなヘルパ関数から始める。`log_components` と `log_mix` は最大の項をくくり出すことで裾がアンダーフローしないように混合ガウス分布（Gaussian mixture）の対数密度を数値的に安定して評価し（35.8.2 節）、`kl` はグリッド上の 2 つの密度間の KL ダイバージェンスを nat 単位で計算

する。

```
def log_components(x, w, mus, sigmas):
    """log(w_k N(x; mu_k, sigma_k^2)) for every component k; shape (k, len(x))."""
    return (np.log(w)[:, None] - 0.5 * np.log(2 * np.pi * sigmas[:, None] ** 2)
            - (x - mus[:, None]) ** 2 / (2 * sigmas[:, None] ** 2))

def log_mix(x, w, mus, sigmas):
    """Log-density of the mixture sum_k w_k N(mu_k, sigma_k^2), computed stably."""
    c = log_components(x, w, mus, sigmas)
    c_max = c.max(axis=0)
    return c_max + np.log(np.exp(c - c_max).sum(axis=0))

def kl(log_a, log_b, x):
    """D_KL(A||B) by quadrature on the grid x; in nats."""
    return integrate.trapezoid(np.exp(log_a) * (log_a - log_b), x)
```

37.5.1 スコアとフィッシャーダイバージェンス

37.5.1.1 スコア

空間内のあらゆる点において、スコアは「密度はどの方向に、どれだけ急峻に上昇するか」という単純な幾何学的疑問に答える。形式的には、分布 P が微分可能な密度 p を持つとき、**スコア**とは**データ点に関する**対数密度の勾配であり、次式で定義される。

$$s_P(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}). \tag{37.43}$$

古典統計学では「スコア」が対数尤度の**パラメータ**に関する勾配を指すため（36.6 節）、混同しないよう注意してほしい。ここでは $\mathbf{x}$ に関して微分する。1 次元ガウス分布 $\mathcal{N}(\mu, \sigma^2)$ の場合、スコアは単純に次の式となる。

$$s(x) = -\frac{x - \mu}{\sigma^2},$$

これは x を平均 μ へと引き戻す、剛性 $1/\sigma^2$ のばねのように働く。混合分布 $p = \sum_k w_k\, p_k$ に対して連鎖律を適用すると、$s(x) = \sum_k r_k(x)\, s_k(x)$ が得られる。ここで $r_k(x) = w_k p_k(x)/p(x)$ は、x が成分 k から生じた事後確率、すなわち**負担率**（responsibility）である。したがって各点は、各成分のばねに重みを付けて合成した力を受ける。どのモードの近傍でも、そのモードの負担率は 1 に近いため、点はそのモード単独のばねの力を受ける。一方、モード間の谷に沿っては、Figure 37.11 に示すように、負担率が一方のアトラクタから次のアトラクタへと滑らかに支配権を引き渡す。以下ではこの式をそのまま実装し、数値安定性のために負担率を対数空間で計算する。

```
def mix_score(x, w, mus, sigmas):
    """Score of a Gaussian mixture: responsibility-weighted component scores."""
    r = np.exp(log_components(x, w, mus, sigmas) - log_mix(x, w, mus, sigmas))
    return np.sum(r * -(x - mus[:, None]) / sigmas[:, None] ** 2, axis=0)
```

エネルギーベースモデルにおいてスコアがこれほど有用なのは、いかなる正規化定数も無視できるからである。$q(\mathbf{x}) = \tilde{q}(\mathbf{x})/Z$ であれば、次式が成り立つ。

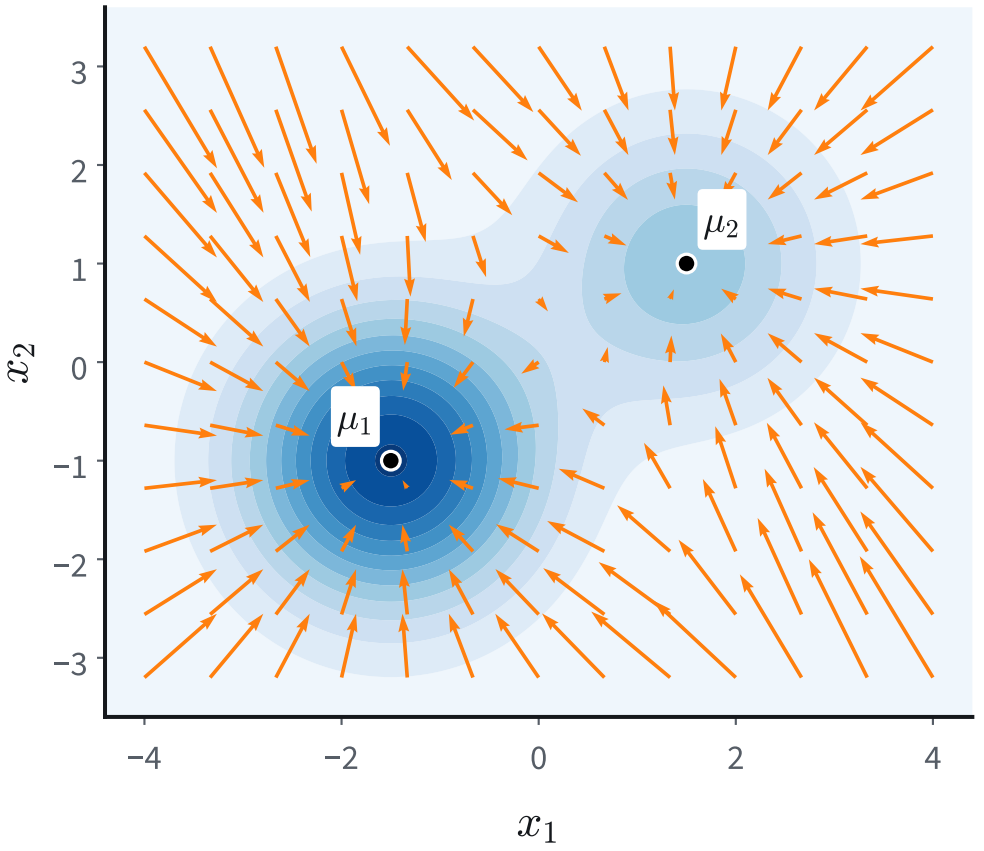


Figure 37.11: 2 成分混合ガウス分布のスコア場。矢印は対数密度の勾配を表す。各モードの近傍ではそのモードの平均へ引き寄せるばねとして働き、モード間の谷に沿っては負担率が一方のアトラクタからもう一方のアトラクタへと点を引き渡す。参考として背後に密度の等高線を描いている。

$$\nabla_{\mathbf{x}} \log q(\mathbf{x}) = \nabla_{\mathbf{x}} \log \tilde{q}(\mathbf{x}) - \nabla_{\mathbf{x}} \log Z = \nabla_{\mathbf{x}} \log \tilde{q}(\mathbf{x}),$$

これは Z が $\mathbf{x}$ に依存しないためである。たとえば、非正規化 4 次密度 $\tilde{q}(x) = e^{-x^4/4}$ のスコアは $-x^3$ であり、$Z \approx 2.56$ であるという事実を一切知ることなく、あらゆる点でこのスコアを評価できる。

37.5.1.2 フィッシャーダイバージェンス

スコアからは Z が消えるため、スコアから構成した比較尺度は非正規化モデルにそのまま適用できる。最も自然な選択は、データのもとで平均した 2 つのスコア場の平均二乗差であり、これは**フィッシャーダイバージェンス**として知られる。

$$D_{\mathrm{F}}(P\|Q) = \frac{1}{2}\, \mathbb{E}_{\mathbf{x}\sim P}\Big[\big\| s_P(\mathbf{x}) - s_Q(\mathbf{x}) \big\|^2\Big]. \tag{37.44}$$

明らかに $D_{\mathrm{F}}(P\|Q) \geq 0$ であり、両方の密度が同一の連結領域上で狭義に正である限り、これは $P = Q$ のとき、かつそのときに限り 0 となる真のダイバージェンスである（37.3.1.1 節）。連結領域上では、スコアが一致すると $\log p - \log q$ が定数とならざるを得ず、両方の密度の積分は 1 でなければならないため、その定数は 0 に限られる。

2 つのガウス分布を比較すると、フィッシャーダイバージェンスと KL ダイバージェンスの重要な違いが浮かび上がる。両方の分布が分散 σ^2 を持ち、平均が距離 Δ だけ離れているとき、両者のスコアは定数 Δ/σ^2 だけすべての x において異なるため、次式が得られる。

$$D_{\mathrm{F}} = \frac{\Delta^2}{2\sigma^4} \qquad \text{whereas} \qquad D_{\mathrm{KL}} = \frac{\Delta^2}{2\sigma^2},$$

ここで 37.1 節のガウス分布に対する KL の公式を用いた。$\sigma = 1$ のとき、2 つの値は一致する。ここで座標 x を 2 倍にスケール変換し、Δ と σ の両方を 2 倍にしてみよう。KL ダイバージェンスは変化しないのに対し、フィッシャーダイバージェンスは元の値の 4 分の 1 に減少する。無次元量（nat 単位で測られる）である KL ダイバージェンスとは異なり、フィッシャーダイバージェンスは傾きの二乗であるがゆえに $1/x^2$ の単位を持つ。したがってその数値はデータのスケーリングに依存するため、あるデータセット上のスコアマッチング損失を別のデータセット上の損失と直接比較することはできず、nat 単位の符号化バジェットとして解釈することもできない。分散が異なる場合については、演習問題 1a で導出する。

37.5.1.3 遠く離れたモードの混合重み

Equation 37.44 の期待値は、データ分布 P のもとで傾きの差を平均する。データが十分に離れた複数のモードを持つとき、この 2 つの要素は相性が悪い。まったく同じガウス成分から構成され、混合重み（mixing weight）だけを入れ替えた 2 つの混合分布を考えてみよう。

$$P_w = w\,\mathcal{N}(-m, 1) + (1 - w)\,\mathcal{N}(m, 1), \qquad w = 0.7 \text{ against } w = 0.3.$$

分布としては、この 2 つの混合分布は大きく離れている。モードどうしが十分に離れると、両者の KL ダイバージェンスは離散重みベクトル $(0.7, 0.3)$ と $(0.3, 0.7)$ の間の KL ダイバージェンスに近づき、その値は $0.7 \log \frac{7}{3} + 0.3 \log \frac{3}{7} = 0.339$ nat となって m がどれほど大きくなっても変わらない。

しかし、両者のスコアはほとんど異ならない。どちらのモードの近傍でも、**両方**の混合分布においてそのモードの負担率は 1 に近いため、両方のスコア場はともにそのモード単独のばねへと帰着する。陽に書き下すと（演習問題 2b で導出するように）、P_w のスコアは次のようになる。

$$s_w(x) = -x + m\tanh(mx - c_w), \qquad c_w = \tfrac{1}{2}\log\frac{w}{1-w}.$$

この式は、十分左側ではばね $-(x+m)$ のように、十分右側ではばね $-(x-m)$ のように振る舞い、$x = c_w/m$ 付近で両者の間を遷移する。混合重み w が、この遷移の位置を通じて**のみ**現れることに注目してほしい。$m = 5$ のとき、遷移の位置は $x = 0.085$（$w = 0.7$ の場合）および $x = -0.085$（$w = 0.3$ の場合）となり、まさに 2 つのモードに挟まれた空っぽな谷の真ん中にある。そこでの密度は $e^{-m^2/2}/\sqrt{2\pi} \approx 1.5 \times 10^{-6}$ にすぎず、データは 2 つのスコアが食い違う唯一の場所をほとんど決して訪れない。m を大きくしながら両方のダイバージェンスを数値積分で計算してみよう。KL ダイバージェンスが 0.339 nat 付近にとどまる一方、フィッシャーダイバージェンスはおよそ $e^{-m^2/2}$ のように急減すると予想される。

```
xs = np.linspace(-14.0, 14.0, 14001)
w_a, w_b, unit = np.array([0.7, 0.3]), np.array([0.3, 0.7]), np.ones(2)
for m in (1.0, 3.0, 5.0):
    mus_m = np.array([-m, m])
    log_a, log_b = log_mix(xs, w_a, mus_m, unit), log_mix(xs, w_b, mus_m, unit)
    score_gap = mix_score(xs, w_a, mus_m, unit) - mix_score(xs, w_b, mus_m, unit)
    fisher = 0.5 * integrate.trapezoid(np.exp(log_a) * score_gap ** 2, xs)
    print(f'm = {m:.0f}: KL = {kl(log_a, log_b, xs):.3f} nats, '
          f'Fisher = {fisher:.1e}')
```

```
m = 1: KL = 0.184 nats, Fisher = 1.1e-01
m = 3: KL = 0.337 nats, Fisher = 7.1e-03
m = 5: KL = 0.339 nats, Fisher = 4.0e-06
```

予想どおり、KL ダイバージェンスが 0.339 nat に落ち着く一方、フィッシャーダイバージェンスは約 0.1 から約 4×10^{-6} へと 4 桁以上も急落する。$m = 5$ では $e^{-m^2/2} = e^{-12.5} \approx 4 \times 10^{-6}$ のオーダーとなり、ほぼ空の谷から予測されたとおりである。純粋にスコアのみから構築された損失は、遠く離れたモード間で確率質量がどう分けられているかを見ることができない。先ほど**連結**領域にこだわったのはこのためである。2 つの非連結な領域上では、スコアが同一であっても $\log p - \log q$ が各領域で異なる定数をとり得る。その結果、フィッシャーダイバージェンスの最小化によって訓練されたモデルは、すべてのモードの位置と形状を正しく捉えながら、自らの質量の 30% を、データが 70% を置いている場所に割り当ててしまうことがあり得る (Wenliang and Kanagawa 2020)。

幸い、ノイズを加えれば谷は埋まる。両方の混合分布に標準偏差 2 の独立なガウスノイズを加えると、各成分の分散は 1 から 5 へと広がる。標準偏差の 10 倍離れていたモードどうしは標準偏差の 4.5 倍の距離にまで縮まり、谷における密度は 1.5×10^{-6} から約 0.015 へと上昇する。こうなれば、2 つのスコアが食い違う領域をデータが実際に訪れるようになる（演習問題 2c でこの改善幅を定量化する）。拡散モデルがスコアマッチングを行う前に多数のスケールにわたるノイズでデータを摂動させる大きな理由はここにあり、17.1 節では訓練済みのニューラルネットワーク上でまさにこの盲点を計測する。

37.5.2 スタインの恒等式

フィッシャーダイバージェンスには、さらに直接的な 2 つ目の障壁がある。データ分布のスコア s_P に依存するにもかかわらず、データはサンプルとしてしか観測できないという点である。有限

個の点集合には傾きが存在しない。s_P がどこに現れるかを見るため、1 次元で二乗を展開してみよう。

$$D_{\mathrm{F}}(P\|Q) = \tfrac{1}{2}\,\mathbb{E}_P[s_P^2] - \mathbb{E}_P[s_P\, s_Q] + \tfrac{1}{2}\,\mathbb{E}_P[s_Q^2]. \tag{37.45}$$

第 3 項はモデルが提供する関数のデータに関する期待値であるため、サンプルから直接推定できる。第 1 項は Q に一切依存しないため、モデルの当てはめやランキングにおいては無関係な定数である。厄介なのはすべて中央の交差項にあり、そこでは未知のデータスコアとモデルスコアが掛け合わされている。驚くべきことに、あらゆる滑らかな密度が満たすたった 1 つの恒等式によって、この交差項は完全に消去される。以下では 1 次元の密度に対してこの恒等式を述べて証明し、それを 2 回活用する。1 回目はデータ分布に適用して交差項を取り除くためであり、2 回目はモデル分布に適用してサンプルが Q から得られたものかを検定するためである。

命題（スタインの恒等式） **$p > 0$ を $\mathbb{R}$ 上の微分可能な密度とし、そのスコアを $s = (\log p)'$ とする。また g を微分可能な関数とし、$g(x)\,p(x) \to 0$（$x \to \pm\infty$ のとき）、$\mathbb{E}_p|g'| < \infty$、および $\mathbb{E}_p|g\,s| < \infty$ を満たすとする。このとき、次が成り立つ。**

$$\mathbb{E}_{x\sim p}\big[g'(x) + g(x)\,s(x)\big] = 0. \tag{37.46}$$

証明： 定義より $s = p'/p$ であるため、$p' = s\,p$ となる。部分積分（34.6.2.3 節）を行うと、次式が得られる。

$$\int g'\,p\,dx = \big[g\,p\big]_{-\infty}^{\infty} - \int g\,p'\,dx = -\int g\,s\,p\,dx,$$

ここで、g に関する仮定より境界項は消え、両方の積分は収束する。■

標準正規分布では $s(x) = -x$ であり、スタインの恒等式は $\mathbb{E}[g'(X)] = \mathbb{E}[X g(X)]$ となる。たとえば $g(x) = x$ と置けば、直ちに $\mathbb{E}[X^2] = 1$ が得られる。このガウス分布の場合はスタインの補題（Stein's lemma）(Stein 1981) であり、一般形は (Stein 1986) によるものである。この恒等式がこれほど強力なのは、傾きに関する知識を期待値に関する等式へと変換してくれるからであり、期待値こそサンプルから推定できる量だからである。

37.5.2.1 スコアマッチング

データ分布 P は、そこからのサンプルしか観測できないものの、滑らかな密度を持つと仮定しよう。スタインの恒等式を P に適用し（$s = s_P$）、モデルのスコアをテスト関数に選ぶと（$g = s_Q$）、$\mathbb{E}_P[s_Q' + s_Q\, s_P] = 0$ が得られる。したがって Equation 37.45 の交差項は $-\mathbb{E}_P[s_Q']$ となり、s_P が完全に消去される。

$$D_{\mathrm{F}}(P\|Q) = \mathbb{E}_{x\sim P}\Big[\tfrac{1}{2}\,s_Q(x)^2 + s_Q'(x)\Big] + \tfrac{1}{2}\,\mathbb{E}_P[s_P^2]. \tag{37.47}$$

部分積分によって、導関数を持たないデータ側からモデル側へと微分が移されたのである。末尾の項 $\frac{1}{2}\mathbb{E}_P[s_P^2]$ はどのモデルに対しても同じ値であるため、フィッシャーダイバージェンスの最小化は最初の期待値の最小化と等価になる。それはモデル自身が提供する量のデータ点にわたる平均であり、Z も s_P も必要としない。この手法は**スコアマッチング**と呼ばれ、Hyvärinen (2005) が非正規化モデルを当てはめるために導入し、自然画像の画像パッチ上で実証したものである。

37.5.1 節で導入した非正規化 4 次モデルを見ると、この手法が実際にどれほど鮮やかに機能するかがわかる。スケールパラメータ $\theta > 0$ を $\tilde{q}_\theta(x) = e^{-\theta x^4/4}$ によって導入すると、$s_Q(x) = -\theta x^3$ および $s'_Q(x) = -3\theta x^2$ となる。データにわたる標本平均を上線で表すと、Equation 37.47 のモデルに依存する部分は θ に関する単純な二次関数で推定される。

$$\overline{\tfrac{1}{2}\, s_Q^2 + s'_Q} = \tfrac{1}{2}\,\theta^2\,\overline{x^6} - 3\,\theta\,\overline{x^2}, \qquad \text{which is minimized at} \qquad \hat{\theta} = \frac{3\,\overline{x^2}}{\overline{x^6}}.$$

わずか 2 つの標本モーメントだけで、$e^{-\theta x^4/4}$ を一度も積分することなくパラメータを当てはめることができる。最尤法では $Z(\theta)$ が必要となり異なる推定量が導かれるが（演習問題 4）、ガウスモデルの場合は両者の推定量は一致する（演習問題 3）。

1 次元では、s'_Q は導関数をもう 1 回とるだけで済む。しかし $\mathbb{R}^n$ ではヤコビ行列（Jacobian）のトレース、すなわち n 個の二階導関数（座標ごとに 1 つ）の和となるため、スコアの計算では 1 回で済んだ逆伝播が n 回の逆伝播へと膨れ上がり、画像に対してはあまりにも高コストになる。38.6.1 節ではスコアマッチングを $\mathbb{R}^n$ へと拡張し、データにノイズを加えることでこれらの二階導関数の計算を回避できることを示したうえで、その結果から拡散モデルの訓練損失を導出する。

37.5.2.2 サンプルに対する検定

スタインの恒等式は逆向きにも使える。すなわち、与えられたサンプルがあるモデル Q から得られたものかどうかを検定する用途である。モデル自身にこの恒等式を適用すると、条件を満たすすべてのテスト関数 g に対して、量 $g' + g\,s_Q$ は Q のもとで平均 0 を持つ。g を $g' + g\,s_Q$ へ写す線形作用素は、Q の**スタイン作用素**（Stein operator）と呼ばれる。$g' + g\,s_Q$ の標本平均が、ある g に対して 0 から大きく離れていれば、そのサンプルが Q からサンプリングされたものではないという根拠が得られる。この検査に必要なものに注目してほしい。サンプル点において評価したモデルのスコアだけでよく、正規化定数 Z もモデルからのサンプルも一切不要である。

単一のテスト関数でどれだけの違いを捉えられるだろうか。モデルを $Q = \mathcal{N}(0,1)$ とすると、そのスコアは $-x$ であるため、スタイン作用素は g を $g'(x) - x\,g(x)$ へと写す。$g(x) = 1$ を選ぶと $-x$ となり標本平均を検査し、$g(x) = x$ を選ぶと $1 - x^2$ となり二次モーメントを 1 と照合する。単純なモーメントとしての解釈を持たない 3 つ目の選択肢として、$g(x) = \sin x$ を試してみよう。以下では 200,000 個の点を 3 つの分布、すなわち真のモデル $\mathcal{N}(0,1)$、平均をずらした分布 $\mathcal{N}(0.5,1)$、および裾を広げた分布 $\mathcal{N}(0,4)$ からサンプリングし、各テスト関数の標本平均をその標準誤差とともに出力する。最初の 2 つのモーメントだけから、平均をずらしたサンプルでは -0.5 と $1 - 1.25 = -0.25$、広げたサンプルでは 0 と $1 - 4 = -3$ になると予想される。

```
rng = np.random.default_rng()
n = 200_000
# Stein operator of the model Q = N(0, 1), whose score is -x:  g'(x) - x g(x)
tests = [('g = 1', np.ones_like, np.zeros_like),
         ('g = x', lambda t: t, np.ones_like),
         ('g = sin x', np.sin, np.cos)]
samples = [('N(0, 1)', rng.normal(0.0, 1.0, n)),
           ('N(0.5, 1)', rng.normal(0.5, 1.0, n)),
           ('N(0, 4)', rng.normal(0.0, 2.0, n))]
print(f'{"sample":10s}' + ''.join(f'{name:>18s}' for name, _, _ in tests))
for s_name, x in samples:
    vals = [g_prime(x) - x * g(x) for _, g, g_prime in tests]
    print(f'{s_name:10s}' + ''.join(f'{v.mean():+10.3f} ± {v.std() / np.sqrt(n):.3f}'
```

```
                                    for v in vals))
```

```
sample                 g = 1             g = x          g = sin x
N(0, 1)       -0.001 ± 0.002    -0.002 ± 0.003    -0.001 ± 0.002
N(0.5, 1)     -0.503 ± 0.002    -0.249 ± 0.004    -0.145 ± 0.002
N(0, 4)       +0.002 ± 0.004    -3.006 ± 0.013    -0.410 ± 0.003
```

モデル自身のサンプルに対しては、すべての標本平均が 0 から標準誤差の数倍以内に収まる。平均をずらしたサンプルに対しては 3 つのテスト関数すべてが容易にずれを捉え、それぞれ約 -0.50、-0.25、-0.15 という値を報告する。一方、広げたサンプルでは様相が異なる。平均が依然として 0 であるため、$g = 1$ は 0 に近い値（0.01 程度以内）を報告し、分散における 4 倍もの誤差を完全に見逃してしまう。不一致を検出できるのは $g(x) = x$（-3 付近）と $g(x) = \sin x$（-0.4 付近）だけである。個々のテスト関数はサンプルの 1 つの側面しか調べないため、誤った分布であってもその側面においてモデルと容易に一致し得るのである。

なぜ $g = 1$ が平均のシフトを捉えながら分散の違いを見逃すのかを理解するため、任意の滑らかなデータ分布 P がそれ自身のスタインの恒等式 $\mathbb{E}_P[g'] = -\mathbb{E}_P[g\, s_P]$ を満たすことに注目しよう。これをモデルのスタイン期待値に代入すると、次式が得られる。

$$\mathbb{E}_P[g' + g\, s_Q] = \mathbb{E}_P[(s_Q - s_P)\, g]. \tag{37.48}$$

つまりテスト関数の期待値は、2 つのスコアの差に g の重みを付けて平均したものにほかならない。平均をずらしたサンプルでは、そのスコア差は定数 $-x + (x - 0.5) = -0.5$ となり、$g = 1$ がこれを余すところなく捉える。一方、広げたサンプルではスコア差が $-x + x/4 = -\tfrac{3}{4}x$ という奇関数になるため、任意の対称な密度に対して $g = 1$ の重みで積分すると 0 になってしまう。いかなる単一のテスト関数も、起こり得るすべてのスコア差と相関を持つことはできないのである。

37.5.3 カーネルスタイン不一致度

1 つのテスト関数では 1 つの性質しか検査できないため、37.4.1 節で最大平均値差異（maximum mean discrepancy、MMD）に対して行ったのと同じ手法をとる。すなわち、再生核特徴空間（reproducing kernel feature space）の単位球に含まれるすべての関数という関数族全体にわたって一度に検定を行い、最大の違反量をとるのである。驚くべきことに、無限に多くのテスト関数にわたるこの上限もまた、データ点のペアに対するカーネルの評価値から構成される閉形式へと帰着する。

$\mathcal{A}_Q$ を Q のスタイン作用素とし、$(\mathcal{A}_Q g)(x) = g'(x) + s_Q(x)\, g(x)$ で定義する。スタインの恒等式によれば $\mathbb{E}_Q[\mathcal{A}_Q g] = 0$ が成り立つため、P がモデルの恒等式にどれだけ大きく違反しているかを $\sup_g \mathbb{E}_P[\mathcal{A}_Q g]$ によって測る。

テスト関数として $g(x) = \langle \mathbf{w}, \varphi(x) \rangle$ の形のものを考える。ここで φ は特徴写像であり、$\|\mathbf{w}\| \le 1$ とする。線形性により、スタイン作用素は特徴量に直接作用して $(\mathcal{A}_Q g)(x) = \langle \mathbf{w}, \psi(x) \rangle$ となる。ただし $\psi = \varphi' + s_Q\, \varphi$ である。コーシー・シュワルツの不等式により、単位球上における $\langle \mathbf{w}, \mathbb{E}_P[\psi] \rangle$ の上限は単純に $\big\| \mathbb{E}_P[\psi] \big\|$ となる。このノルムを二乗すると $\mathbb{E}\langle \psi(x), \psi(x') \rangle$（ただし $x, x' \sim P$ は 2 つの独立なサンプル）が得られる。内積を展開すると 4 つの項が現れ、それぞれがカーネル $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$ とその偏微分によって次のように表される。

$$\begin{aligned} \langle \varphi'(x), \varphi(x') \rangle &= \partial_x k, \qquad \langle \varphi(x), \varphi'(x') \rangle = \partial_{x'} k, \\ \langle \varphi'(x), \varphi'(x') \rangle &= \partial_x \partial_{x'} k. \end{aligned}$$

これら 4 つの項にスコアの重みを付けてまとめることで、**スタインカーネル**（Q に対するカーネル）が次のように定義される。

$$\begin{aligned} k_Q(x,x') = {} & s_Q(x)\, s_Q(x')\, k(x,x') + s_Q(x)\, \partial_{x'} k(x,x') \\ & + \partial_x k(x,x')\, s_Q(x') + \partial_x \partial_{x'} k(x,x'). \end{aligned} \tag{37.49}$$

すると、二乗**カーネルスタイン不一致度**（kernel Stein discrepancy、KSD）は、データからの 2 つの独立なサンプルにわたるスタインカーネルの期待値となる。

$$\mathrm{KSD}^2(P,Q) = \mathbb{E}_{x,x' \sim P}\big[k_Q(x,x')\big]. \tag{37.50}$$

この期待値は P のみに依存するため、標本 $x_1, \dots, x_N$（P から抽出）から次の不偏推定量が得られる。

$$\widehat{\mathrm{KSD}}^2 = \frac{1}{N(N-1)} \sum_{i \neq j} k_Q(x_i, x_j),$$

これは MMD で用いたのと同じ非対角ペアの平均である。統計学では、異なる組にわたるこの形の平均は U 統計量（U-statistic）と呼ばれ、以下の関数名 `ksd2_ustat` もそこに由来する。

KSD が実際に何を測っているかを理解するため、Equation 37.48 を φ の各座標に適用しよう。すると $\mathbb{E}_P[\psi] = \mathbb{E}_P[(s_Q - s_P)\,\varphi]$ が得られ、ノルムを二乗すると次式になる。

$$\mathrm{KSD}^2(P,Q) = \mathbb{E}_{x,x' \sim P}\big[(s_Q - s_P)(x)\, k(x,x')\, (s_Q - s_P)(x')\big]. \tag{37.51}$$

すなわち KSD は、フィッシャーダイバージェンス $D_{\mathrm{F}}(P\|Q)$ をカーネルで平滑化した親戚にあたる。各点でのスコア差を個別に二乗する代わりに、データ点のペアにわたってスコア差を掛け合わせ、$k(x,x')$ で重みを付けるのである。残念ながら、これは KSD がモード重みの盲点を受け継ぐだけでなく、さらに悪化させることを意味する。37.5.1 節の離れた混合分布に対しては、ペアの**両方**の点がスコアの異なる空っぽの谷に落ちなければならなくなるため、KSD^2 はフィッシャーダイバージェンスの減衰レートの二乗にあたる e^{-m^2} 程度で減衰する（演習問題 2e）。

両方のスコアが既知であるとき、Equation 37.51 を使えば母集団での正確な値を予測できる。データを $P = \mathcal{N}(0,1)$ とし、3 つの候補モデルに対して検定してみよう。真のモデル $\mathcal{N}(0,1)$ に対してはスコア差が消えるため、$\mathrm{KSD}^2 = 0$ となる。平均をシフトしたモデル $\mathcal{N}(1,1)$ に対しては至るところでスコア差が $-(x-1) - (-x) = 1$ となるため、$\mathrm{KSD}^2 = \mathbb{E}[k(X,X')]$ となる。RBF カーネル $k(x,x') = e^{-(x-x')^2/(2\ell^2)}$（長さスケール ℓ）のもとでは、これは 37.4.1 節で出会ったガウス積分と同じである。$X - X' \sim \mathcal{N}(0,2)$ より期待値は $(1 + 2/\ell^2)^{-1/2}$ に等しく、$\ell = 1$ において $\mathrm{KSD}^2 = 1/\sqrt{3} \approx 0.577$ を与える。

3 つ目の候補は、非正規化 4 次密度 $\tilde{q}(x) = e^{-x^4/4}$（37.5.1 節）であり、その正規化定数は一切必要ない。スコアは $-x^3$ であるためスコア差は $x - x^3$ となり、結果として得られる Equation 37.51 の二重積分は初等的な閉形式を持たないため、以下の `ksd2_population` ではグリッド上の数値積分でこれを評価する（ここでそれができるのは、この合成ベンチマークでは真のデータ密度がわかっているからにほかならない）。

1 次元の RBF カーネルに対し、Equation 37.49 に現れる導関数は $\partial_{x'} k = \frac{x - x'}{\ell^2}\, k$、$\partial_x k = -\frac{x - x'}{\ell^2}\, k$、および $\partial_x \partial_{x'} k = \left(\frac{1}{\ell^2} - \frac{(x-x')^2}{\ell^4}\right) k$ である。以下のコードでは、`dk_dy` が第 2 引数に

関する導関数を表し、kappa が値 $k_Q(x_i, x_j)$ の行列を格納する。ここでは 2,000 個の点からなる $\mathcal{N}(0,1)$ の単一標本を 3 つのモデルすべてに対して検定し、ksd2_ustat には各モデルのスコア関数だけを渡す。

```
def ksd2_ustat(x, score, ell=1.0):
    """U-statistic KSD^2 with the RBF kernel exp(-(a-b)^2 / (2 ell^2))."""
    d = x[:, None] - x[None, :]
    k = np.exp(-d ** 2 / (2 * ell ** 2))
    dk_dx = -d / ell ** 2 * k
    dk_dy = d / ell ** 2 * k
    d2k = (1 / ell ** 2 - d ** 2 / ell ** 4) * k
    s = score(x)
    kappa = (s[:, None] * s[None, :] * k + s[:, None] * dk_dy
             + dk_dx * s[None, :] + d2k)
    n = len(x)
    return (kappa.sum() - np.trace(kappa)) / (n * (n - 1))

def ksd2_population(score, ell=1.0):
    """KSD^2 of the data P = N(0, 1) against a model with this score, from
    the score difference s_Q - s_P weighted by the kernel, by quadrature."""
    t = np.linspace(-10.0, 10.0, 2001)
    h = np.exp(-t ** 2 / 2) / np.sqrt(2 * np.pi) * (score(t) + t)
    k = np.exp(-(t[:, None] - t[None, :]) ** 2 / (2 * ell ** 2))
    return h @ k @ h * (t[1] - t[0]) ** 2

x_p = rng.standard_normal(2000)          # the data sample, P = N(0, 1)
for name, s in [('true model  N(0,1)', lambda t: -t),
                ('wrong model N(1,1)', lambda t: -(t - 1)),
                ('quartic exp(-x^4/4)', lambda t: -t ** 3)]:
    print(f'KSD^2 vs the {name:19s}: {ksd2_ustat(x_p, s):+.4f}  '
          f'(population {ksd2_population(s):.4f})')
```

```
KSD^2 vs the true model  N(0,1) : +0.0010  (population 0.0000)
KSD^2 vs the wrong model N(1,1) : +0.5274  (population 0.5774)
KSD^2 vs the quartic exp(-x^4/4): +0.2835  (population 0.3208)
```

真のモデルに対しては、推定値は 0 から数千分の 1 以内に収まる（0 の不偏推定量であれば起こり得るように、わずかに負になることさえある）。シフトしたモデルに対しては、予測した $1/\sqrt{3} \approx 0.58$ の近く（通常は約 0.1 以内）に着地する。4 次モデルに対しては 0.1 の数倍程度の値となり、明らかに 0 から大きく離れている。ここでの母集団の値は 0.32 であるが、単一の標本では 0.1 以上ずれることも珍しくない。3 次のスコア $-x^3$ が裾で急激に増大するため、ごく少数の極端な点が平均を支配してしまうからである。

3 つの検定すべてを通じて変化させたのは、スコア関数だけであることに注目してほしい。データのサンプルは固定したままであり、どのモデルからも一度もサンプリングしておらず、4 次モデルの正規化定数 $Z \approx 2.56$ がコードに現れることも一度もなかった。

RBF カーネルを用いる場合、母集団における値 0 は信頼できる。スコアに関する穏やかな正則性条件のもとで、滑らかなデータ密度に対して $\mathrm{KSD}^2(P, Q) = 0$ となるのは $P = Q$ のとき、かつそのときに限られるからである (Liu et al. 2016; Chwialkowski et al. 2016)。それでも、2 成分混合分布の例は有益な教訓を与えてくれる。「理論上は非ゼロである」からといって、遠く離れた分布に対してその値が天文学的に小さくなることまで防げるわけではないのである。

KSD を $\mathbb{R}^n$ へと拡張する際に変わるのは記法だけである。g はベクトル場となり、スタイン作用

素は $s_Q^\top g + \nabla \cdot g$ となり、Equation 37.49 のスカラーの積はベクトルの内積に置き換わり、最後の項は $\sum_i \partial_{x_i} \partial_{x_i'} k$ になる。実用的には、KSD は非正規化モデルの適合度検定（goodness-of-fit test）や MCMC サンプラの診断指標として使われる。また、スタイン作用素をカーネル自身に適用すると、粒子群を Q へと押し動かす決定論的な更新方向が得られ、これはスタイン変分勾配降下法（Stein variational gradient descent）として知られるアルゴリズムである (Liu and Wang 2016)。なお、本書で扱う訓練目的関数の中に KSD を直接用いるものはない。

37.5.4 訓練目的関数と対応するダイバージェンス

前節までの 2 節にわたって分布比較の 3 つの主要な族を構築し、本節ではスコアに基づく比較を 2 つ追加した。Figure 37.12 に、これら 3 つの族とそれらが共有するダイバージェンスを整理する。この図では、3 つの族のいずれにも属さないフィッシャーダイバージェンスと KSD は省いている。KSD は積分確率指標（integral probability metric、テスト関数にわたる上限）の形をとるものの、そのテスト関数のクラスがモデルのスコアに依存し、したがって Q とともに変化するためである。

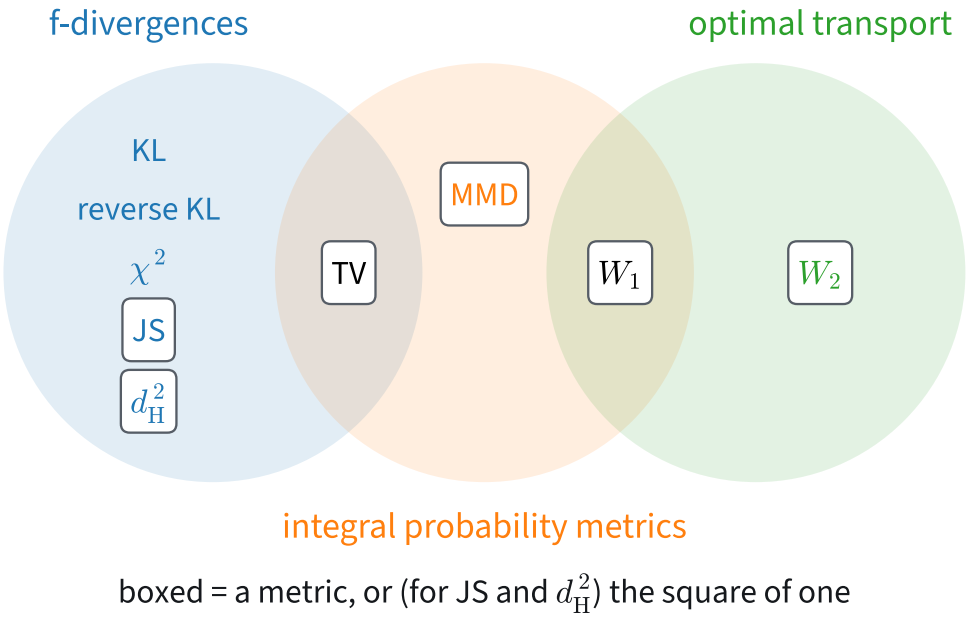


Figure 37.12: 比・テスト関数・輸送の 3 つの族と共通する要素。f ダイバージェンスは密度比の凸関数を平均し、積分確率指標はテスト関数のクラス上で平均の差を最大化し、最適輸送距離は質量を移動させる最小コストを測る。全変動は最初の 2 つの族に属し、1 次ワッサースタイン距離は後ろの 2 つの族に属する。（全変動は非ゼロの距離だけ質量を動かすたびにコスト 1 を課す輸送コストでもあるが、見やすさのため図では省いている。）枠で囲んだ名称は距離（計量）であり、JS と二乗ヘリンガーダイバージェンス d_{H}^2 は距離の二乗である。

Table 37.4 は、これらの比較手法の上に構築された訓練目的関数を整理したものである。各目的関数について、モデルとデータに求める要件と、期待値が厳密で（敵対的目的関数の場合は 37.3.4 節で実サンプルと生成サンプルを識別するよう訓練された補助ネットワークであるクリティックが最適である）理想的な極限において最小化されるダイバージェンスを示している。4 列目

では、37.3 節の冒頭で導入した診断テストケースを再訪する。すなわち、単一点に集中したデータ $P = \delta_0$ と、モデルのサンプル $Q = \delta_d$（距離 d だけ離れた点）の比較である。d を変化させても一定値にとどまる比較尺度は、そのようなモデルに対してデータへ向かう勾配を一切与えない。

Table 37.4: 必要とする情報順に並べた訓練目的関数と、平均が厳密かつクリティックが最適なときにそれらが最小化するダイバージェンス。点質量の列は δ_0 対 δ_d（$d \neq 0$）に対する値であり、$f(0)$ と $f'(\infty)$ はそれぞれ $f(u)$ の 0 における極限と $f(u)/u$ の無限大における極限を表す。また MMD の項目では長さスケール ℓ の RBF カーネルを用いている。

目的関数が必要とする情報	目的関数	ダイバージェンス	距離 d 離れた 2 つの点質量	解説箇所
データ点におけるモデル密度	最尤法（自己回帰モデル、フロー）	順方向 KL	∞	37.3.2 節
クリティックを通じた両者のサンプル	オリジナル GAN	イェンセン・シャノン	$\log 2$（定数）	37.3.4 節
クリティックを通じた両者のサンプル	f-GAN	選択した D_f	$f(0) + f'(\infty)$（定数）	37.3.4 節
両者のサンプル（固定カーネルならクリティック不要）	モーメントマッチング、2 標本検定	MMD	$\mathrm{MMD}^2 = 2 - 2e^{-d^2/(2\ell^2)}$（$\ell$ の数倍を超えると平坦）	37.4.1 節
両者のサンプル、1 リプシッツのクリティック	WGAN	W_1	d	37.4.2 節
モデルのサンプル、そこでのモデル密度、非正規化ターゲット	変分推論、変分オートエンコーダの事後分布	逆方向 KL	∞	37.3.2 節
モデルのスコア、データのサンプル	スコアマッチング、拡散モデル	フィッシャー	未定義（点質量はスコアを持たない）	37.5.1 節
モデルのスコア、データのサンプル	適合度検定、サンプラの品質評価	KSD	未定義（点質量はスコアを持たない）	37.5.3 節

この表は行がモデルの提供できる情報順に並んでいるため、上から下へ読むことで実践的な選択ガイドとして使える。モデルの正規化済み密度を評価でき、かつデータが現れ得るあらゆる場所で正であるならば、最尤法で訓練するのがよい。順方向 KL はデータしか必要とせず、不偏な標本推定値を与え、データのどのモードであっても取りこぼせばペナルティを課すからである。モデルがサンプルを生成することしかできない場合には、3 つの選択肢がある。GAN（16.1 節）のように学習されたクリティックを訓練する手法は、どの特徴量が実サンプルと生成サンプルを区別するかが事前にはわからない画像などの複雑なデータのサンプル限定モデルに対する標準的な選択肢である。今日では、その大半は拡散モデルから蒸留された 1 ステップ生成器や潜在オートエンコーダのデコーダである。学習されたクリティックの欠点は、推定されたダイバージェンスに下向きの統計的バイアスがかかり、モデルがデータから遠く離れると平坦になってしまうことである（37.3.4 節）。最適な 1 リプシッツのクリティックにはそのような平坦な領域はない。W_1 があらゆる距離で一定の傾きを保つからである。ただし、訓練されたクリティックが最適なクリティックと同じように振る舞うとは限らない。16.2.4.2 節の実験では、訓練されたロジスティッククリティックは遠く離れた生成サンプルに対してもなお傾きを持ち、通常の両側勾配ペナルティによって傾き 1 付近に保たれた W_1 クリティックは生成器をデータまで運んだあと、そのまま通り過ぎさせてしまう。固定カーネルを用いれば 2 つ目のネットワークの訓練を完全に回避でき、

任意の次元で $O(N^{-1/2})$ の推定誤差を達成できるため（37.4.1 節）、2 標本検定やサンプルの評価に理想的である。もっとも、固定カーネルの MMD は生データの次元が増すにつれて弱まり、長さスケールの数倍を超えると勾配が消失するため、通常は学習された特徴量に対して適用される。

最後の 3 行は、正規化定数が未知である設定を扱っている。変分推論（variational inference）のように、**ターゲット**密度が非正規化である一方、**モデル**がサンプルと計算可能な密度の両方を提供する場合には、単一のモードへ崩壊するリスクを伴いつつも逆方向 KL が適用できる。モデル自体がそのスコアを通じてのみ既知である場合には、拡散モデルが行うように（17.1 節）遠く離れたモードの重みを見失わないよう複数スケールにわたるノイズを加えることを条件として、（スコアマッチングを通じた）フィッシャーダイバージェンスが自然な選択肢となる。

この表は各目的関数が**どの**比較尺度を目標とするかを教えてくれるが、有限のデータからそれをどれだけ高い信頼性で推定できるかは教えてくれない。直近の 3 節で扱ったサンプルに基づく推定量（クリティックによる下界、MMD^2、および KSD^2）はいずれも推定誤差を伴い、おまけにクリティックによる下界には低く出るバイアスまである。続く 2 節では、2 つの分布の比較から 2 つの確率変数間の従属性の計測へと焦点を移す。37.6 節では KL ダイバージェンスを用いて一方の確率変数がもう一方についてどれだけの情報を持つかを定義し、37.7 節では n 組のペアからなる標本が実際にどれだけの従属性を保証できるかを問う。

37.5.5 まとめ

モデルの正規化定数が計算困難であっても、入力に関する対数密度の勾配であるスコアを通じればデータと比較できる。微分によって正規化定数が消えるからである。フィッシャーダイバージェンスはデータスコアとモデルスコアの二乗差を平均し、スタインの恒等式は 1 回の部分積分によって未知のデータスコアを消去してスコアマッチングを導く。ただし、スコアは純粋に局所的な量であるため、遠く離れたモード間で確率質量がどう分割されているかを見ることはできない。標準偏差の 10 倍離れた 2 つの混合分布に対しては、フィッシャーダイバージェンスが 4×10^{-6} まで落ち込む一方で、KL ダイバージェンスは 0.339 nat のまま保たれる。逆にスタインの恒等式をモデル側に適用すると、モデルのスコアのみを用いてサンプルを検定するカーネルスタイン不一致度が得られる。結局のところ、適切な訓練目的関数はモデルが何を提供できるか、すなわち正規化済み密度か、サンプルか、スコアかによって決まるのである。

37.5.6 演習問題

1. **フィッシャーダイバージェンスとスタインのモーメント。** $\mathcal{N}(\mu, \sigma^2)$ のスコアが $-(x-\mu)/\sigma^2$ であったことを思い出そう。
   1. フィッシャーダイバージェンス Equation 37.44 を $\mathcal{N}(\mu_1, \sigma_1^2)$ と $\mathcal{N}(\mu_2, \sigma_2^2)$ の間で閉形式として計算し、分散が等しいときに $(\mu_1 - \mu_2)^2/(2\sigma^4)$ へ帰着することを確かめよ。フィッシャーダイバージェンスは対称か。
   2. スタインの恒等式 Equation 37.46（$\mathcal{N}(0,1)$ の場合）を $g(x) = x^k$ に適用して漸化式 $\mathbb{E}[X^{k+1}] = k\,\mathbb{E}[X^{k-1}]$ を導出し、それを用いて $\mathbb{E}[X^4]$ と $\mathbb{E}[X^6]$ を計算せよ。恒等式の条件を満たす任意の密度に対して、$g(x) = 1$ は何を与えるか。
   3. 分布 P（$\mathbb{R}$ 上）がすべての次数の有限モーメントを持ち、$\mathbb{E}_P[g'(X) - X g(X)] = 0$ をすべての多項式 g に対して満たすとする。正規分布がそのモーメントによって一意に定まるという事実を用いて、なぜ $P = \mathcal{N}(0,1)$ となるかを説明せよ。次に $P(X=1) = P(X=-1) = \tfrac{1}{2}$ とし、すべての多項式 g（次数が高々 2 のもの）に対してはこの恒等式を満たすが、$g(x) = x^3$ に対しては成り立たないことを示せ。このことは、固定された次数以下の多項式上でのみスタインの恒等式を検査する適合度検定に対して何を意味するか。
2. [code] **混合重みに対するスコアの不感性。**● $P_w = w\,\mathcal{N}(-m, 1) + (1-w)\,\mathcal{N}(m, 1)$ とせよ。

1. 本文では $D_{\mathrm{F}}(P_{0.7}\|P_{0.3})$ と $D_{\mathrm{KL}}(P_{0.7}\|P_{0.3})$ を $m \in \{1,3,5\}$ について計算した。この計算を $m \in \{1,\dots,6\}$ に拡張せよ。KL ダイバージェンスが $0.4\log\frac{7}{3} \approx 0.339$ に近づくことを示し、$\log D_{\mathrm{F}}$ を m^2 に対して直線当てはめすることでフィッシャーダイバージェンスの減衰率を求めよ。
2. 成分の負担率からスコアの公式 $s_w(x) = -x + m\tanh(mx - c_w)$ を導出せよ。$s_{0.7}$ と $s_{0.3}$ の差が $x=0$ において最大値 $0.8\,m$ をとることを示し、$u = mx$ をフィッシャーダイバージェンスの積分に代入することで、その積分が $C\,m\,e^{-m^2/2}$（大きな m における漸近形であり、C は重みのみに依存する）としてスケールすることを示せ。これを (a) で当てはめた減衰率と比較せよ。この結果は、スコアマッチングによって学習される混合重みについて何を意味するか (Wenliang and Kanagawa 2020)。
3. 独立な $\mathcal{N}(0,\sigma^2)$ のノイズを両方の混合分布のサンプルに加え、各成分を $\mathcal{N}(\pm m, 1+\sigma^2)$ に変える。(a) を $m=5$ において $\sigma \in \{0, 0.5, 1, 2\}$ について繰り返せ。ノイズを加えることは各ダイバージェンスにどのような影響を与えるか。強いノイズは混合重みを明らかにするが各モードの形状をぼかし、弱いノイズはその逆となる。複数のノイズレベルにわたってスコアマッチングを行うモデルが、なぜその両方を復元できるのか (Song and Ermon 2019)。
4. P_t と Q_t を、独立な $\mathcal{N}(0,t)$ のノイズを両方に加えた後の 2 つの分布とする。等しい分散 σ^2（ノイズによって $\sigma^2 + t$ に増加）を持つ 2 つのガウス分布に対し、本文の閉形式から $\frac{d}{dt} D_{\mathrm{KL}}(P_t\|Q_t) = -D_{\mathrm{F}}(P_t\|Q_t)$ が成り立つことを確かめよ。この恒等式は滑らかな密度のあらゆる対に対して成り立つ。2 つの混合分布に対し、有限差分を用いて $m=5$ および $t \in \{1,4,9\}$ でこれを確認し、$D_{\mathrm{KL}}(P\|Q) = \int_0^\infty D_{\mathrm{F}}(P_t\|Q_t)\,dt$ を導け。どのノイズレベルにおいてフィッシャーダイバージェンスが最大になるか。この積分は、単一のノイズレベルでのスコアマッチングとすべてのノイズレベルにわたるスコアマッチングの違いについて何を教えてくれるか (Y. Song, Durkan, et al. 2021)。
5. カーネルスタイン不一致度も同じ盲点を共有する。2 つの混合分布に対して Equation 37.51 をグリッド上の数値積分で評価し（$\ell = 1$ の RBF カーネルを用い、$m \in \{1,3,5\}$ とする）、本文のフィッシャーダイバージェンスと比較せよ（$m=5$ ではおよそ 10^{-11} になるはずである）。本文の `ksd2_population` を改変し、データ分布としての $\mathcal{N}(0,1)$ を $P_{0.7}$ に置き換えよ。積分値に寄与するためにはペアの両方の点がどこに落ちなければならず、その領域の幅は ℓ に比べてどれほどか。ここから KSD^2 が e^{-m^2} のように減衰することを導け。より小さな長さスケールを選べば改善するか。

3. **ガウスモデルに対するスコアマッチング**。データ分布 P が平均 a と分散 v を持つとし、モデルを $Q = \mathcal{N}(\mu,\sigma^2)$ ととって、$J = \mathbb{E}_P\big[\frac{1}{2}s_Q^2 + s_Q'\big]$ を Equation 37.47 のモデルに依存する部分とする。
   1. $J = (v + (a-\mu)^2)/(2\sigma^4) - 1/\sigma^2$ となることを示し、$\mu = a$ かつ $\sigma^2 = v$ において（P のそれ以外の性質によらず）最小化されることを確かめよ。これを順方向 KL ダイバージェンスを最小化するガウス分布と比較せよ（37.3.2 節）。
   2. 最小点では $J = -1/(2v)$ となるため、P からその最良のガウス近似へのフィッシャーダイバージェンスは $\frac{1}{2}(\mathbb{E}_P[s_P^2] - 1/v)$ である。P に対するスタインの恒等式を $g(x) = x - a$ に適用し、続けてコーシー・シュワルツの不等式を用いることで $\mathbb{E}_P[s_P^2] \geq 1/v$ を証明し、等号成立は P がガウス分布のとき、かつそのときに限られることを示せ。与えられた分散のもとで、平均二乗スコアが最小となる密度はどれか。
   3. モデルを $\tilde{q}_\theta(x) = \exp(\theta_1 x + \theta_2 x^2)$ と再パラメータ化せよ。J が θ の二次関数となり、したがってスコアマッチングがデータのモーメントに関する 2 本の線形方程式を解くことに帰着することを示せ。また、任意の指数分布族のモデル $\tilde{q}_\theta(x) = \exp\big(\theta^\top \phi(x)\big)$ に対しても同様の性質が成り立つことを示せ。これに対し、最尤法では代わりに何を計算する必要があるか。
4. [code] **非正規化 4 次モデル**。$\tilde{q}_\theta(x) = e^{-\theta x^4/4}$（$\theta > 0$）とせよ。

1. スコアを計算せよ。スタインの恒等式を $g(x) = x$ および $g(x) = x^3$ に適用し、正規化定数を計算することなく、このモデルのもとで $\mathbb{E}[X^4] = 1/\theta$ および $\mathbb{E}[X^6] = (3/\theta)\,\mathbb{E}[X^2]$ が成り立つことを示せ。多項式 g を用いたスタインの恒等式によって $\mathbb{E}[X^2]$ も決定できるか。
2. 本文ではスコアマッチング推定量 $\hat{\theta} = 3\,\overline{x^2}/\overline{x^6}$（上線は標本平均を表す）を導出した。正規化定数が $Z(\theta) = \theta^{-1/4} Z(1)$ のようにスケールすること、および最尤法が $\hat{\theta} = 1/\overline{x^4}$ を与えることを示せ。(a) を用いて、両方の推定量がデータの異なる関数であるにもかかわらず、モデルからの大標本上ではともに真の θ へ収束することを論じよ。
3. U が形状 $\frac{1}{4}$、尺度 1 のガンマ分布に従うとき、ランダムな符号を付けた $(4U/\theta)^{1/4}$ は密度 $\tilde{q}_\theta/Z(\theta)$ を持つ。$1{,}000$ 個の点を $\theta = 1$ でサンプリングして両方の推定値を計算し、これを $2{,}000$ 回繰り返せ。どちらの推定量の方が標準偏差が小さく、何倍異なるか。スコアマッチングは Z の計算を回避する引き換えに、何を犠牲にしているか。
4. 今度はデータが標準正規分布に従い（KSD の実験と同様）、どの 4 次モデルも厳密には一致しない場合を考える。大標本上では、(b) の 2 つの推定量はそれぞれ $3\,\mathbb{E}[X^2]/\mathbb{E}[X^6] = \frac{1}{5}$ と $1/\mathbb{E}[X^4] = \frac{1}{3}$ に収束する。両方の当てはめのスコアにおいて `ksd2_population` を評価し、さらにそれを最小化する θ を求めよ。3 つの規準が 3 つの異なる 4 次モデルを選んだことになる。なぜこれは矛盾ではなく、どのような条件のもとでなら 3 つすべてが一致するか。

5. [code] **スタインカーネルの長さスケール**。KSD の実験と同様にデータを $P = \mathcal{N}(0,1)$ とし、長さスケール ℓ の RBF カーネルを用いよ。
   1. モデル $\mathcal{N}(\mu, 1)$ に対し、Equation 37.51 を用いて $\mathrm{KSD}^2 = \mu^2\,(1 + 2/\ell^2)^{-1/2}$ となることを示せ。$\ell \to \infty$ のときの極限はいくらか。また、どの単一のテスト関数 g を選べば、その同じ極限が $\mathbb{E}_P[g' + g\,s_Q]$ の二乗として得られるか。
   2. モデル $\mathcal{N}(0, 4)$（スコアは $-x/4$）に対し、$\mathrm{KSD}^2 = \frac{9}{16}\,\ell\,(\ell^2 + 2)^{-3/2}$ となることを示せ。（ヒント：XX' を独立変数 $X + X'$ と $X - X'$ を用いて書き直せ。）この値がデータの標準偏差である $\ell = 1$ で最大となり、$\ell \to \infty$ のとき 0 に近づくことを示せ。(a) の答えを用いてこの極限を説明せよ。
   3. 両方の公式を `ksd2_population` によって $\ell \in \{0.25, 1, 4, 16\}$ で確認し、さらにスコア $-x^3$ を持つ 4 次モデルに対して対応する 4 つの値を計算せよ。いかなる単一の長さスケールも、あらゆる誤ったモデルに対して同時に最適とはなり得ないのはなぜか。

37.6 相互情報量

ここまでの節では、データ分布とモデルのように、同じ変数上の 2 つの分布を比較してきた。これに対し、深層学習における多くの問いは変数の対を扱う。画像とそのキャプション、同じ写真から切り出した 2 つの領域、あるいは文とその翻訳などである。一方の変数は、もう一方の変数についてどれだけの情報を与えてくれるのだろうか。

従属性の標準的な尺度である相関係数は、線形な関係しか検出できない。$X \sim \mathcal{N}(0,1)$ とし、$Y = X^2 + 0.3\,\epsilon$ とおく。ここで $\epsilon \sim \mathcal{N}(0,1)$ は独立なノイズである。X^3 は 0 の周りで対称であるため $\mathrm{Cov}(X, Y) = \mathbb{E}[X^3] = 0$ となり、X と Y の相関は 0 である。しかし実際には、X は Y をほぼ決定している。X がわかれば、Y は標準偏差 0.3 のノイズの分しか変動しないのに対し、単体での Y はおよそ 0 から 9 まで広がる。本節で定義する量は、この従属性を難なく捉える。実際、この対には相関 0.9 の 2 変量ガウス分布よりも多くの共有情報量が割り当てられる。

求めているのは、2 つの変数が独立であるとき、かつそのときに限り 0 となり、従属性が強まるにつれて大きくなり、さらにどちらの変数を別の座標で表し直しても変化しない尺度である。これを構築するための道具はすでに揃っている。KL ダイバージェンス $D_{\mathrm{KL}}(P \| Q)$（37.1 節）であ

る。ギブスの不等式により、この量は常に非負であり、$P = Q$ の場合にのみ 0 になる。P を対の真の同時分布とし、Q を各成分が独立であった場合に従う積分布とすれば、**相互情報量**（mutual information）$I(X;Y)$ が得られる。

相互情報量の定義自体は単純であり、その真価はそこから導かれる帰結にある。本節では 3 つの基本的な結果を確立する。第 1 に、変数を処理しても他の何らかの対象との相互情報量は増加しないため、ニューラルネットワークのどの層も、その直下の層より多くの入力情報をもつことはない。第 2 に、分類器がある誤り率を達成するには、入力がラベルについて最低限の情報量をもっていなければならない。第 3 に、任意の確率モデルのテスト損失は、その情報量に対する下界を直接保証する。続いて、Shannon の 2 つの根源的な問い（雑音通信路はどれだけの情報を運べるか、非可逆な圧縮器はどれだけ少ない情報の保持で済むか）と、圧縮から発展した第 3 の問い（学習された表現は何を保持すべきか）へと進む。本節を通じて背後の分布は既知であると仮定する。サンプルのみから相互情報量を推定するのはより難しい問題であり、37.7 節で扱う。

これまでと同様、単位には nat を用いる。ここで 1 ビット $= \ln 2 \approx 0.693$ nat である。対数はすべて自然対数とし、一般の式では log、数値を評価する際には ln と書く。また 2 値エントロピーを $H_b(p) = -p \log p - (1-p)\log(1-p)$ で表す。

37.6.1 従属性の尺度

すべての定義を具体的に理解するため、まず 1 つの 2×2 確率表の上でそれぞれの量を計算する。この確率表は、本節を通じて通し例として繰り返し登場する。

37.6.1.1 結合エントロピーと条件付きエントロピー

X を公平なビットとし、Y を X が確率 $\delta = 0.1$ で反転したコピーとする。X をノイズのある回線で送られたビット、Y を受信されたビットと読むこともできるし、Y を 2 値ラベル、X を 90% の確率でそれに一致するノイズ付き特徴量と読むこともできる。表は対称であるため、以下の議論はどちらの解釈を選んでも変わらない。行を $x \in \{0,1\}$、列を $y \in \{0,1\}$ に対応させると、同時分布は次のようになる。

$$p_{X,Y} = \begin{pmatrix} 0.45 & 0.05 \\ 0.05 & 0.45 \end{pmatrix},$$

行と列の和をとると、周辺分布 $p_X = p_Y = (\tfrac{1}{2}, \tfrac{1}{2})$ が得られる。この対を**反転ビット**（flipped bit）と呼ぶ。以下では、添字によって分布の変数名を表し、条件付き分布 $p(y \mid x)$ のように引数から分布が明らかな場合には添字を省略する。

個々のエントロピー $H(X)$ と $H(Y)$ に加え、確率変数の対には 37.1.1.3 節で導入した 2 つの自然なエントロピーがある。**結合エントロピー**（joint entropy）は、対を単一の確率変数（ここでは 4 つの結果をもつ変数）とみなしたときのエントロピーである。

$$H(X,Y) = -\mathbb{E}_{(x,y)\sim P_{X,Y}}[\log p_{X,Y}(x,y)]. \tag{37.52}$$

この表では、$H(X,Y) = -2(0.45 \ln 0.45) - 2(0.05 \ln 0.05) = 1.018$ nat となる。これは 1 つの公平なビットがもつ 0.693 nat よりは大きいが、独立な 2 つの公平なビットがもつ 1.386 nat よりは小さい。

条件付きエントロピー（conditional entropy）は、Y について X が既知となった後に残る驚きを測る量であり、条件付き分布の各行 $p(y \mid x) = p_{X,Y}(x,y)/p_X(x)$ のエントロピーの平均で与えられる。

$$H(Y \mid X) = -\mathbb{E}_{(x,y)\sim P_{X,Y}}[\log p(y \mid x)] = \sum_{x} p_X(x)\, H(Y \mid X = x). \tag{37.53}$$

反転ビットでは、2 つの行は $(0.9, 0.1)$ と $(0.1, 0.9)$ であり、それぞれのエントロピーは $H_b(0.1) = 0.325$ nat であるから、$H(Y \mid X) = 0.325$ nat となる。X を観測した後でも、Y は 0.325 nat の驚きを（元の 0.693 nat のうち）依然として保持している。

これら 3 つの数値が $1.018 = 0.693 + 0.325$ を満たしていることに注目しよう。これは連鎖律 $H(X, Y) = H(X) + H(Y \mid X)$（37.1.1.3 節）にほかならない。すなわち、対がもつ驚きは、X の驚きに、Y の追加的な驚き（X が既知となった後のもの）を加えたものに等しい。

37.6.1.2 相互情報量

条件付きエントロピーは、Y の驚きのうち X では説明できない部分を捉える。いま求めたいのは、2 つの変数が共有する部分である。もし X と Y が独立であれば、その同時分布は周辺分布の積 $P_X \otimes P_Y$ となり、すべてのセルに $p_X(x)\, p_Y(y)$ が入る。**相互情報量**とは、実際の同時分布からこの独立な積分布への KL ダイバージェンスである。

$$I(X; Y) = D_{\mathrm{KL}}(P_{X,Y} \,\|\, P_X \otimes P_Y) = \mathbb{E}_{(x,y)\sim P_{X,Y}}\left[\log \frac{p_{X,Y}(x, y)}{p_X(x)\, p_Y(y)}\right]. \tag{37.54}$$

反転ビットでは、周辺分布の積はすべてのセルに 0.25 を置く。したがって、独立な積に対する同時確率の比は、対角成分では $0.45/0.25 = 1.8$（X と Y が一致する場合）となり、非対角成分では $0.05/0.25 = 0.2$（不一致の場合）となる。この対数比を同時分布のもとで平均すると次を得る。

$$I(X; Y) = 0.9 \ln 1.8 + 0.1 \ln 0.2 = 0.529 - 0.161 = 0.368 \text{ nats}.$$

一致は独立性が予測する頻度の 1.8 倍起こり、不一致は 1/5 の頻度しか起こらない。相互情報量とは、単にこの倍率の対数の期待値である。KL ダイバージェンスは誤ったモデルを用いた際に生じる余分な符号長も測るため（37.2 節）、$I(X; Y)$ は X と Y を独立として扱う符号が 1 対あたりに浪費する nat 数でもある。17.9 節では、ネットワークが 2 つのマスクトークンを独立に予測する際に、まさにこのペナルティが現れる。

Equation 37.54 は表を 1 行ずつ読むこともできる。分子と分母を $p_X(x)$ で割ると比は $p(y \mid x)/p_Y(y)$ となり、対に関する期待値は、条件付き分布と周辺分布の間の KL ダイバージェンスを x に関して平均した形に書き直せる（対象は Y の分布である）。

$$I(X; Y) = \mathbb{E}_{x\sim P_X}\Big[D_{\mathrm{KL}}\big(p(y \mid x) \,\|\, p_Y(y)\big)\Big]. \tag{37.55}$$

この形式では、相互情報量は x を観測することで Y の分布がその周辺分布から平均してどれだけ変化するかを測る。反転ビットでは、各条件付き行 $(0.9, 0.1)$ および $(0.1, 0.9)$ は、それぞれ $0.9 \ln 1.8 + 0.1 \ln 0.2 = 0.368$ nat だけ周辺分布 $(0.5, 0.5)$ から離れている。この非対称な見方は、入力とノイズ付きエンコーダの出力のように 2 つの変数が異なる役割を担う場合に特に自然であり、37.6.4 節で再び立ち返る。

命題（相互情報量の性質）。 任意の対 (X, Y) に対して $I(X; Y) = I(Y; X) \geq 0$ が成り立ち、等号は X と Y が独立であるとき、かつそのときに限り成り立つ。関係する各エントロピーが有限であるとき、次が成り立つ。

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X).$$

証明。 X と Y を入れ替えても Equation 37.54 の比は変わらないため、対称性が従う。非負性と等号成立条件は、$P_{X,Y}$ と $P_X \otimes P_Y$ にギブスの不等式を適用すれば直ちに従う。すなわち KL ダイバージェンスは非負であり、すべての点で $p_{X,Y}(x,y) = p_X(x)\,p_Y(y)$ となるとき、かつそのときに限り 0 になるが、これは独立性の定義そのものである。エントロピーの恒等式を得るには、比の対数を 3 つの項に分解して期待値をとればよい。

$$I(X;Y) = \mathbb{E}\left[\log p_{X,Y}(x,y)\right] - \mathbb{E}\left[\log p_X(x)\right] - \mathbb{E}\left[\log p_Y(y)\right] = -H(X,Y) + H(X) + H(Y).$$

2 通りの順序による連鎖律 $H(X,Y) = H(Y) + H(X \mid Y) = H(X) + H(Y \mid X)$ を代入すれば、残りの 2 つの形式が得られる。$\blacksquare$

したがって、X が Y について語る情報量は、Y が X について語る情報量と正確に一致する。最後の恒等式は相互情報量を不確実性の減少として解釈するものであり、X を観測すると Y の驚きは $H(Y)$ から $H(Y \mid X)$ へと低下する（反転ビットでは 0.693 nat から 0.325 nat へ下がる）。また非負性から劣加法性の上界 $H(X,Y) \le H(X) + H(Y)$ が導かれ、等号は独立性が成り立つときにのみ成立する。すなわち、変数の対がもつ驚きは、各成分を別々に考えたときの驚きの和を決して超えない。Figure 37.13 はこれらの関係を視覚的に表したものであり、命題の各恒等式は重なり部分の面積の異なる読み取り方にすぎない。

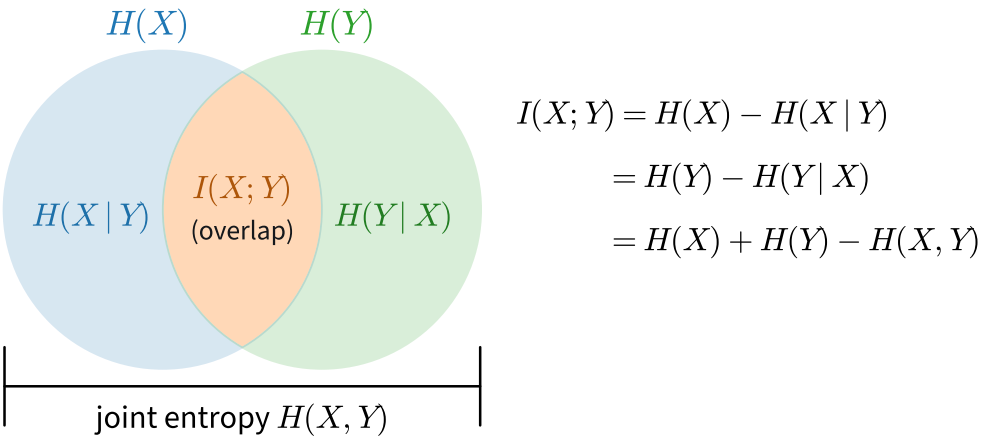


Figure 37.13: 相互情報量 $I(X;Y)$ をエントロピー $H(X)$ と $H(Y)$ の重なりとして表した図。三日月形の部分は条件付きエントロピー $H(X \mid Y)$ と $H(Y \mid X)$、和集合は結合エントロピー $H(X,Y)$ であり、$I(X;Y)$ の 3 つの等価な表現は面積を読み取ることで得られる。

以下の関数 `mutual_information` は任意の離散同時確率表に対してダイバージェンス形式を実装したものであり、本節全体で再利用する。またこのセルでは、各行のエントロピーを平均することで恒等式 $I(X;Y) = H(Y) - H(Y \mid X)$ も確認している。

```
p_xy = np.array([[0.45, 0.05],    # rows x, columns y: a fair bit X and
                 [0.05, 0.45]])   # its copy Y, flipped with probability 0.1

def entropy(p):
    """Entropy of a p.m.f. of any shape, in nats (0 log 0 = 0)."""
    p = p[p > 0]
    return float(-(p * np.log(p)).sum())

def mutual_information(p_xy):
    """I(X;Y) of a discrete joint p.m.f. (rows x, columns y), in nats."""
    p_x = p_xy.sum(axis=1, keepdims=True)
    p_y = p_xy.sum(axis=0, keepdims=True)
    m = p_xy > 0                                  # 0 log 0 = 0
    return float((p_xy[m] * np.log(p_xy[m] / (p_x * p_y)[m])).sum())

p_x, p_y = p_xy.sum(axis=1), p_xy.sum(axis=0)     # marginals from the joint
H_rows = sum(p_x[i] * entropy(p_xy[i] / p_x[i]) for i in range(2))   # H(Y|X)
I = mutual_information(p_xy)
print(f'I(X;Y) = {I:.4f} nats (KL form) = {entropy(p_y) - H_rows:.4f} '
      f'(H(Y) - H(Y|X)) = {I / np.log(2):.3f} bits')
```

```
I(X;Y) = 0.3681 nats (KL form) = 0.3681 (H(Y) - H(Y|X)) = 0.531 bits
```

どちらの式も 0.3681 nat、すなわち 0.531 ビットを返す。10 回に 1 回反転するコピーであっても、元のビットについて半ビット強の情報を伝えているのである。

37.6.1.3 ガウス対

ここまでは離散的な確率表を扱ってきたが、深層学習における大半の変数（活性化値、勾配、埋め込みなど）は連続である。連続な場合の中で、手計算で完全に求められるケースが 1 つある。2 変量ガウス分布である。これは 2 つ目の通し例として、本節後半のレート・歪み曲線および情報ボトルネック（information bottleneck）の双方の土台となる。連続密度に対しては、離散エントロピーは微分エントロピー h（37.1.4.1 節）に置き換わる。微分エントロピーは負になり得るうえ測定単位に依存するが、KL ダイバージェンスによる相互情報量の定義にはそのどちらの欠点もない。連鎖律の代数変形は離散性を用いていないため、各量が有限である限り $I(X;Y) = h(X) + h(Y) - h(X,Y) = h(Y) - h(Y \mid X)$ が成り立つ。ここで $h(Y \mid X)$ は条件付き密度 $p(y \mid x)$ の微分エントロピーを x に関して平均した量を表す。

命題（ガウス分布の相互情報量）。 **(X,Y) が分散 $\sigma_X^2, \sigma_Y^2 > 0$、相関係数 ρ（ただし $\rho^2 < 1$）の 2 変量ガウス分布に従うならば、次が成り立つ。**

$$I(X;Y) = -\tfrac{1}{2}\log\left(1-\rho^2\right). \tag{37.56}$$

証明。 $I(X;Y) = h(Y) - h(Y \mid X)$ を用いる。分散 σ^2 のガウス変数は微分エントロピー $\tfrac{1}{2}\log(2\pi e\sigma^2)$（37.1.4.3 節）をもつため、$h(Y) = \tfrac{1}{2}\log(2\pi e\sigma_Y^2)$ である。$X = x$ で条件付けると、Y は引き続きガウス分布に従い、その分散は $\sigma_Y^2(1-\rho^2)$ となって x に依存しない。これは Y の最良線形予測（X に基づく予測）を差し引いた残差の分散であり、2 変量ガウス分布ではこの残差は X と独立である（36.3.3.3 節）。したがって $h(Y \mid X) = \tfrac{1}{2}\log\left(2\pi e\sigma_Y^2(1-\rho^2)\right)$ となり、差をとれば $-\tfrac{1}{2}\log(1-\rho^2)$ が得られる。■

この結果は $1-\rho^2$、すなわち Y の分散のうち X によって説明されずに残る割合のみに依存する。

Y の単位が相殺されていることに注目されたい。これは 37.6.1.5 節で証明する座標不変性の最初の現れである。

$\rho = 0$ のとき変数は独立であり、$I(X;Y) = 0$ となる。$\rho \to \pm 1$ の極限では同時密度が直線上に退化し、$I(X;Y) \to \infty$ となる。これら両極端の間では、$\rho = 0.5$ で 0.144 nat、$\rho = 0.9$ で 0.830 nat、$\rho = 0.99$ で 1.959 nat となる。周辺分布が標準正規分布で $\rho = 0.9$ である 2 変量ガウス分布は本節を通じて繰り返し登場するため、これを $\rho = 0.9$ **のガウス対**と呼ぶ。

極限 $\rho \to \pm 1$ が示すように、相互情報量は実際に無限大になり得る。この現象は後ほど、ニューラルネットワークの決定論的な層と情報ボトルネックのエンコーダという 2 つの場面で再び登場する。連続確率変数 X に対して $Y = X$ であれば、X を知ることは Y を小数点以下無限桁まで特定することを意味する。Y が X の決定論的な関数であり、どの単一の値も正の確率をもたない場合にも同様のことが成り立つ。対照的に、Y が有限のエントロピーをもつ離散変数である場合には、相互情報量が無限大になることはあり得ない。このとき $I(X;Y) = H(Y) - H(Y \mid X) \le H(Y)$ が X によらず成り立つ（ここで $H(Y \mid X)$ は $p(\cdot \mid x)$ のエントロピーを x について平均した量である）。たとえば標準正規分布に従う X に対し、X の符号はちょうど $\ln 2$ nat の情報量をもつ。この同じ上限によって、16.1 節のイェンセン・シャノン・ダイバージェンスも $\ln 2$ 以下に抑えられる。そこで示したとおり、この量はサンプルとその生成元を識別する 2 値ラベルの間の相互情報量に等しいからである。

37.6.1.4 自己相互情報量

相互情報量はすべての結果の対にわたって平均をとるため、どの特定の対が従属性を生み出しているかは見えなくなる。しかし実際には、特定の 1 対に関心があることも多い。たとえば単語 **new** を見たとき、次の単語が **york** になる確率は偶然よりはるかに高くなるだろうか。Equation 37.54 の和の中にある被加数は、この問いに直接答えるものであり、**自己相互情報量**（pointwise mutual information）と呼ばれる。

$$\operatorname{pmi}(x, y) = \log \frac{p_{X,Y}(x, y)}{p_X(x)\, p_Y(y)}, \tag{37.57}$$

この量は、特定の対 (x, y) が実際に生じる頻度を、X と Y が独立であった場合に生じる頻度と比較する。正の pmi は 2 つの結果が引き合う（偶然の予測より高い頻度で共起する）ことを示し、負の pmi は反発し合うことを示す。そして $I(X;Y) = \mathbb{E}_{(x,y)\sim P_{X,Y}}[\operatorname{pmi}(x, y)]$ はその平均的な引力である。反転ビットでは、一致する 2 つのセルで pmi は $\ln 1.8 = +0.588$、不一致の 2 つのセルで $\ln 0.2 = -1.609$ となり、先ほど平均をとった対数と正確に一致する。

自己相互情報量は自然言語処理における古典的な道具であり (Church and Hanks 1990)、本来結びつきの強い単語対である**連語**（collocation）を、単に両方とも頻出語であるために共起する単語対から切り分ける。小さなコーパスで連続する単語対（バイグラム）を数え上げ、**new**、**machine**、**the** のそれぞれの直後に **york**、**learning**、**day** が何回現れるかを集計したとしよう。

出現回数	york	learning	day
new	38	3	9
machine	1	46	3
the	11	21	168

生の出現回数はここでは誤解を招く。「the day」が群を抜いて最多の対となっている（出現回数は 168 回であるのに対し、「new york」は 38 回）のは、単に **the** と **day** がともに高頻度語だから

である。以下のコードは、各同時頻度を 2 つの周辺単語頻度の積で割り、4 つの対に対する pmi を出力するとともに、表全体の相互情報量を求める。

```
counts = np.array([[38., 3., 9.],      # new york / new learning / new day
                   [1., 46., 3.],      # machine ...
                   [11., 21., 168.]])  # the ...
first, second = ['new', 'machine', 'the'], ['york', 'learning', 'day']

p = counts / counts.sum()
pmi = np.log(p / (p.sum(axis=1, keepdims=True) * p.sum(axis=0, keepdims=True)))
for (i, j) in [(0, 0), (1, 1), (2, 2), (2, 0)]:
    print(f'pmi({first[i]}, {second[j]}) = {pmi[i, j]:6.3f} nats '
          f'(count {counts[i, j]:.0f})')
print(f'I(first; second) = {float((p * pmi).sum()):.4f} nats')
```

```
pmi(new, york) =  1.517 nats (count 38)
pmi(machine, learning) =  1.372 nats (count 46)
pmi(the, day) =  0.336 nats (count 168)
pmi(the, york) = -1.109 nats (count 11)
I(first; second) = 0.4146 nats
```

周辺頻度で補正すると、真の連語が上位に浮上する。「new york」のスコアは 1.52 nat、「machine learning」は 1.37 であるのに対し、高頻度対「the day」は 0.34 にとどまる。一方、「the york」は -1.11 nat と負になり、独立性が予測する頻度の約 1/3 しか現れない。表全体の相互情報量である 0.41 nat は、1 語目が 2 語目を平均してどれだけ強く制約しているかを要約している。これと同じ周辺頻度の補正は、学習された単語埋め込み（19.1 節（英語版））にも再び現れる。隣接語の観測対を K 個のランダムな対から識別するよう skip-gram モデルを訓練することは、シフトされた自己相互情報量 $\mathrm{pmi}(x, y) - \log K$ を成分とする行列を暗黙のうちに行列分解することに相当する (Levy and Goldberg 2014)（演習問題 7（37.7 節））で導出する)。

37.6.1.5 相互情報量と相関

相関係数 $\rho(X, Y) = \mathrm{Cov}(X, Y)/(\sigma_X \sigma_Y)$(36.2 節)は線形な従属性しか検出できず、$\rho(X, Y) = 0$ は独立性を意味しない。対照的に、相互情報量は変数が独立であるとき、かつそのときに限り 0 になる。冒頭の例に戻ると、この違いが具体的になる。

$X \sim \mathcal{N}(0, 1)$ とし、独立なノイズを用いて $Y = X^2 + 0.3\,\epsilon$（ただし $\epsilon \sim \mathcal{N}(0, 1)$）とおく。この対の相関は $\rho = 0$ である。$I(X; Y) = h(Y) - h(Y \mid X)$ と書けば、1 次元積分を用いて相互情報量を評価できる。$X = x$ で条件付けると、Y に残るランダム性は加法的ノイズのみであり、分布を既知の定数 x^2 だけ平行移動しても微分エントロピーは変わらないため、$h(Y \mid X) = h(0.3\,\epsilon) = \tfrac{1}{2}\ln(2\pi e \cdot 0.09) = 0.215$ nat となる。$h(Y)$ を数値計算する前に、解析的な上界を求めておこう。Y の分散は $\mathrm{Var}(X^2) + 0.09 = 2.09$ であり、与えられた分散をもつすべての密度の中で微分エントロピーを最大化するのはガウス分布であるから（37.1.4.3 節）、$h(Y) \le \tfrac{1}{2}\ln(2\pi e \cdot 2.09) = 1.79$ nat、したがって $I(X; Y) \le 1.79 - 0.215 = 1.57$ nat が得られる。Y の周辺密度は x にわたる連続混合分布 $p_Y(y) = \int \mathcal{N}(x; 0, 1)\,\mathcal{N}(y; x^2, 0.09)\,dx$ であって閉形式をもたないため、数値積分によって $h(Y)$ を求める。

```
normal = lambda t, var: np.exp(-t**2 / (2 * var)) / np.sqrt(2 * np.pi * var)
xs = np.linspace(-6, 6, 6001)              # quadrature grid for X
ys = np.linspace(-2.5, 30, 6501)           # grid for Y = X^2 + 0.3 eps
dx, dy = xs[1] - xs[0], ys[1] - ys[0]
```

```
p_Y = np.concatenate([                    # p_Y(y) = int N(x;0,1) N(y;x^2,0.09) dx
    (normal(xs, 1) * normal(c[:, None] - xs**2, 0.09)).sum(axis=1) * dx
    for c in np.array_split(ys, 13)])     # 13 chunks keep memory small
h_Y = float(-(p_Y[p_Y > 0] * np.log(p_Y[p_Y > 0])).sum() * dy)
h_noise = 0.5 * np.log(2 * np.pi * np.e * 0.09)   # h(Y|X) = h(0.3 eps)
print(f'h(Y) = {h_Y:.3f}, h(Y|X) = {h_noise:.3f}, '
      f'I(X;Y) = h(Y) - h(Y|X) = {h_Y - h_noise:.3f} nats')
```

```
h(Y) = 1.356, h(Y|X) = 0.215, I(X;Y) = h(Y) - h(Y|X) = 1.141 nats
```

この対は 1.141 nat を共有している。これがガウス分布による上界 1.57 nat を大きく下回るのは、Y の密度（0 付近に集中し右に長い裾をもつ）がガウス分布から大きく外れているためである。それでいて、線形相関が 0 であるにもかかわらず、この値は 0.830 nat（$\rho = 0.9$ のガウス対）を上回っている。Figure 37.14 に 3 つのケースを並べて比較する。

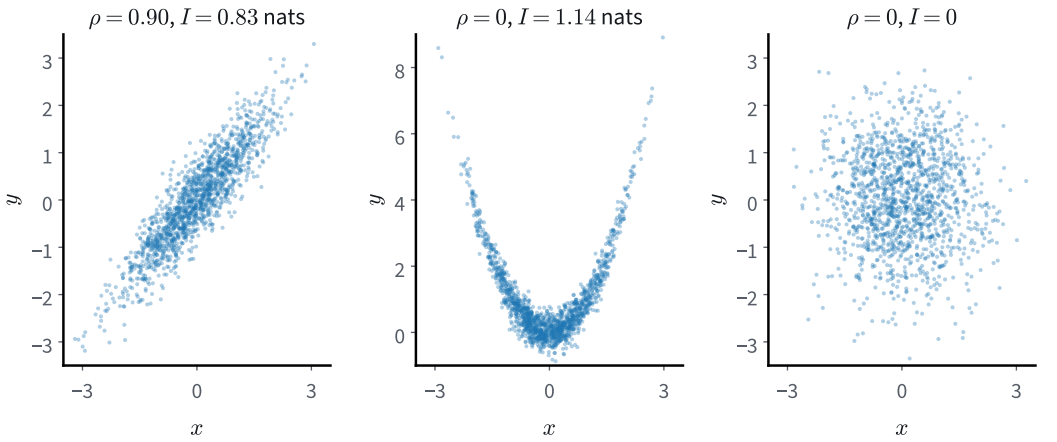


Figure 37.14: 相関と相互情報量による 3 つの対の比較。(a) 相関 0.9 のガウス対。(b) $Y = X^2 + 0.3\epsilon$。X が Y をほぼ決定しているにもかかわらず対称性により相関は 0 になる（(b) では縦軸のスケールが異なることに注意）。(c) 独立な対。相関は (b) と (c) を区別できないが、相互情報量は (b) を (a) より上位に置く。

相関と相互情報量には、もう 1 つ根本的な違いがある。座標変換に対する振る舞いである。気温を摂氏ではなく華氏で測ったり、価格を対数スケールで記録したりしても、その変数が他の対象についてどれだけの情報をもつかはまったく変わらない。したがって、原理に基づいた従属性の尺度はこうした変換のもとで不変であるべきである。

命題（再パラメータ化不変性）。ϕ と ψ をそれぞれ X の値と Y の値に対する可逆写像とする。このとき $I(\phi(X); \psi(Y)) = I(X; Y)$ が成り立つ。

証明。離散変数の場合、可逆写像は結果のラベルを付け替えるだけであり、$(\phi(X), \psi(Y))$ の同時分布は付け替え後の結果に対して (X, Y) の同時分布と同じ確率を割り当てるため、Equation 37.54 の和は変化しない。連続変数の場合、ここでは仕組みを示すために滑らかな場合を証明し、滑らかさの仮定は 37.6.2.2 節で取り除く。ϕ と ψ が連続微分可能かつ連続微分可能な逆写像をもつとし、$(x, y) \mapsto (\phi(x), \psi(y))$ によって変数変換する。変換後の対の $(\phi(x), \psi(y))$ における同時密度は $p_{X,Y}(x, y)\,|\det D\phi(x)|^{-1} |\det D\psi(y)|^{-1}$ となる一方、2 つの周辺密度にも因子 $|\det D\phi(x)|^{-1}$ と $|\det D\psi(y)|^{-1}$ が掛かる。これらのヤコビ行列の行列式因子は密度比の中で

相殺されるため、Equation 37.54 の被積分関数は対応する点で同じ値をとり、その期待値も変わらない。$x \mapsto x^3$ における原点のような確率 0 の例外点の集合は積分に寄与しない。■

微分エントロピーがこの不変性（再パラメータ化不変性（reparameterization invariance））をもたないのは、ヤコビ行列の行列式因子を打ち消す相手が存在しないためであり、それゆえ h は単位に依存する。また相関は（符号を除けば）アフィン変換のもとでしか保存されない。たとえば可逆写像 $x \mapsto x^3$ を $\rho = 0.9$ のガウス対の第 1 座標に適用しても、相互情報量は 0.830 nat のままであるが、相関は 0.9 から $3\rho/\sqrt{15} = 0.697$ へと低下する（$\mathbb{E}[X^3 Y] = 3\rho$ かつ $\mathrm{Var}(X^3) = 15$ が標準正規分布に従う X に対して成り立つため）。相互情報量が問うているのは、2 つの変数が**何らかの**座標系において互いについての情報をもつかであって、たまたま直線上に並んでいるかではない。

37.6.2 データ処理と予測

ここで、冒頭で予告した 3 つの結果を導出しよう。最初の 2 つは根本的な限界を表すものである。すなわち、データを処理しても情報は生み出せず、正確な予測には最低限の情報量が不可欠である。どちらも相互情報量の連鎖律という単一の恒等式から従う。3 つ目の結果は、任意の確率モデルの交差エントロピー損失を、入力が保持する情報量の下界へと直接変換するものである。

37.6.2.1 条件付き相互情報量と連鎖律

反転ビットの例をもう 1 段階拡張してみよう。Z が、Y をさらに独立に確率 0.2 で反転させることで生成されるとする。これは公平なビットを直列につながれた 2 本のノイズ回線で送る状況、あるいはラベルに続いてノイズ付き特徴量とそのノイズ付きコピーが得られる状況に相当する（Figure 37.15）。直列の 2 回反転は単一の反転として振る舞う。ビットが反転して届くのは 2 本の回線のうちちょうど一方だけが反転させるとき、かつそのときに限り、その確率は $0.1 \cdot 0.8 + 0.9 \cdot 0.2 = 0.26$ である。したがって対 (X, Z) は $\delta = 0.26$ の反転ビットであり、$I(X;Z) = \ln 2 - H_b(0.26) = 0.120$ nat となる。2 本目のノイズ回線によって、0.368 nat あった Y の情報量の 2/3 が失われたことになる。

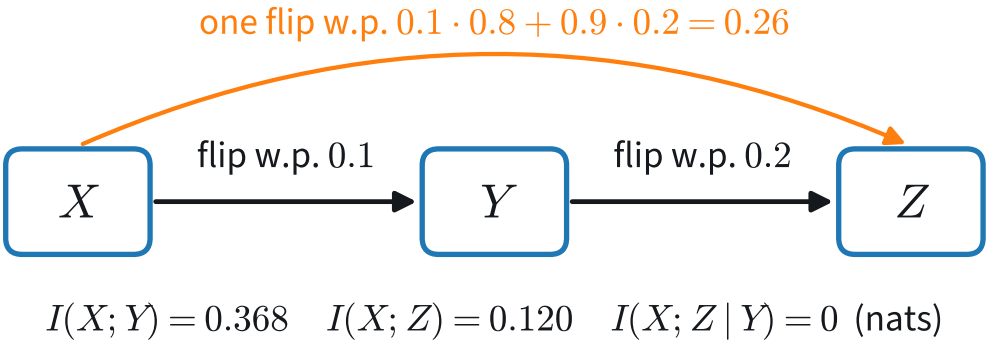


Figure 37.15: 直列の 2 回反転。公平なビット X が確率 0.1 で反転して Y となり、Y が確率 0.2 で反転して Z となる。2 回の反転は合成すると確率 0.26 の単一の反転になる。2 回目の反転で情報量の 2/3 が失われ、$I(X;Z \mid Y) = 0$ は、Z が X について、Y がすでに知っている以上の情報を何ももたないことを表す。

変数が 3 つになると、新たな問いが生じる。X と Y は、Z がすでに既知であるときにどれだけの情報を共有するだろうか。エントロピーに条件付き版があるのと同様に、相互情報量にも条件

付き相互情報量（conditional mutual information）がある。

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z), \tag{37.58}$$

ここで $H(X \mid Y, Z)$ は、X の対 (Y, Z) が与えられたもとでの条件付きエントロピーである。各条件付きエントロピーを期待値として書いて対数をまとめると、ダイバージェンス形式 $I(X;Y \mid Z) = \mathbb{E}_z\big[D_{\mathrm{KL}}(P_{X,Y|z} \,\|\, P_{X|z} \otimes P_{Y|z})\big]$ が得られる。これは各 z の値に対する条件付き分布の上で通常の相互情報量を計算し、平均をとったものにほかならない。KL ダイバージェンスの期待値であるため、$I(X;Y \mid Z)$ は常に非負であり、X と Y が Z を与えたもとで条件付き独立であるとき、かつそのときに限り 0 となる。

この定義により、対 (Y, Z) が X について保持する情報量を 2 つの部分に分解できる。$H(X \mid Z)$ を $I(X;Y,Z) = H(X) - H(X \mid Y, Z)$ の中で引いて足すと、次式を得る。

$$I(X;Y,Z) = \underbrace{H(X) - H(X \mid Z)}_{I(X;Z)} + \underbrace{H(X \mid Z) - H(X \mid Y, Z)}_{I(X;Y \mid Z)}.$$

この恒等式は**相互情報量の連鎖律**（chain rule for mutual information）であり、各項を KL 形式で書けば完全な一般性のもとで成り立つ。言葉で述べれば、(Y, Z) が X について教えてくれる情報量は、Z が教えてくれる情報量に、Y が Z の既知のもとでつけ加える情報量を足したものに等しい。まったく同様に、先に Y で条件付けてもよい。直列の 2 回反転では、2 通りの順序によって同じ合計が 2 通りに分解される。Z は Y と独立なノイズから生成されるため、Y が既知となれば、Z は X についてそれ以上の情報を何も加えない。すなわち $I(X;Z \mid Y) = 0$ であり、Y を先に置くと $I(X;Y,Z) = I(X;Y) + 0 = 0.368$ nat となる。一方、Z を先に置くと、同じ 0.368 nat が $I(X;Z) + I(X;Y \mid Z) = 0.120 + I(X;Y \mid Z)$ と分割され、$I(X;Y \mid Z) = 0.248$ nat が残る。ここでは、Z で条件付けることで、Y が X について明かす情報量が減少した。Z がすでにその情報の一部を保持しているからである。しかし次項で見るように、条件付けによって相互情報量が**増加**することもある。

37.6.2.2 データ処理不等式

ニューラルネットワークでは、各隠れ層はその直下の層から計算され、さらにその層も入力から計算される。このような処理によって、前の層に含まれていなかった入力についての情報を新たに生み出すことはできるだろうか。2 本のノイズ回線では答えは否であったが、その理由はビットという特殊な設定とは無関係である。

3 つの確率変数 X, Y, Z が**マルコフ連鎖**（$X \to Y \to Z$ と書く）をなすとは、Z が (X, Y) に Y を通じてのみ依存すること、すなわち $p(z \mid x, y) = p(z \mid y)$ が成り立つことである。これは逐次処理の構造をそのまま捉えている。まず Y が X から生成され（観測値、特徴マップ、あるいは隠れ層）、続いて Z が（場合によっては新たなランダム性を伴って）Y のみから計算される。同値な言い方をすれば、X と Z は Y が与えられたもとで条件付き独立であり、$I(X;Z \mid Y) = 0$ が成り立つ。

命題（データ処理不等式）。 *$X \to Y \to Z$ がマルコフ連鎖であり、関係する各エントロピーが有限であるならば、次が成り立つ。*

$$I(X;Z) \le I(X;Y),$$

等号が成り立つのは $I(X;Y \mid Z) = 0$ のとき、すなわち $X \to Z \to Y$ もマルコフ連鎖であるとき、かつそのときに限る (Cover and Thomas 1999)。

証明。 $I(X;Y,Z)$ を連鎖律により両方の順序で展開する。

$$I(X;Z) + I(X;Y \mid Z) = I(X;Y,Z) = I(X;Y) + I(X;Z \mid Y).$$

マルコフ性より最後の項は 0 となり、条件付き相互情報量の非負性から $I(X;Z) = I(X;Y) - I(X;Y \mid Z) \le I(X;Y)$ が得られる。等号は $I(X;Y \mid Z) = 0$ のとき、かつそのときに限り成り立つ。$\blacksquare$

決定論的であれランダム化されていようと、Y に対するいかなる処理も、X について保持する情報量を増やすことはできない。直列の 2 回反転では、この不等式は単に $0.120 \le 0.368$ と表される。

この不等式（データ処理不等式（data-processing inequality））はマルコフ連鎖の構造に厳密に依拠しており、条件付け一般の性質ではないことに注意されたい。マルコフ連鎖の外では、条件付けによって本来存在しなかった従属性が生じることがある。X と Y を独立な公平なビットとし、$Z = X \oplus Y$ をその排他的論理和とする。周辺分布では X と Y が共有する情報量は 0 であるが、いったん Z が明かされると、Y は X を完全に決定する。以下のコードは、直列の 2 回反転とこの XOR の 3 つ組の双方について相互情報量を計算する。

```
def cond_mi(p_abc):
    """I(A;B|C) of a joint p.m.f. indexed [a, b, c], in nats."""
    p_ac, p_bc, p_c = p_abc.sum(1), p_abc.sum(0), p_abc.sum((0, 1))
    m = p_abc > 0                                  # 0 log 0 = 0
    ratio = ((p_abc * p_c[None, None, :])[m]
             / (p_ac[:, None, :] * p_bc[None, :, :])[m])
    return float((p_abc[m] * np.log(ratio)).sum())

flip = lambda d: np.array([[1 - d, d], [d, 1 - d]])   # p(output | input)
p_xyz = 0.5 * flip(0.1)[:, :, None] * flip(0.2)[None, :, :]   # X -> Y -> Z
I_xy, I_xz = mutual_information(p_xyz.sum(2)), mutual_information(p_xyz.sum(1))
I_xz_y, I_xy_z = cond_mi(p_xyz.transpose(0, 2, 1)), cond_mi(p_xyz)
print(f'I(X;Y) = {I_xy:.4f}, I(X;Z) = {I_xz:.4f}, '
      f'I(X;Z|Y) = {I_xz_y:.4f}, I(X;Y|Z) = {I_xy_z:.4f}')
print(f'I(X;Y,Z) by the chain rule: {I_xz + I_xy_z:.4f} = {I_xy + I_xz_y:.4f}')
p_xor = np.zeros((2, 2, 2))
for a, b in np.ndindex(2, 2):
    p_xor[a, b, a ^ b] = 0.25          # independent fair bits, Z = X xor Y
print(f'XOR: I(X;Y) = {mutual_information(p_xor.sum(2)):.4f}, '
      f'I(X;Y|Z) = {cond_mi(p_xor):.4f} nats')
```

```
I(X;Y) = 0.3681, I(X;Z) = 0.1201, I(X;Z|Y) = 0.0000, I(X;Y|Z) = 0.2480
I(X;Y,Z) by the chain rule: 0.3681 = 0.3681
XOR: I(X;Y) = 0.0000, I(X;Y|Z) = 0.6931 nats
```

2 回反転では、$I(X;Z \mid Y) = 0$ によってマルコフ性が数値的に確認でき、$I(X;Y,Z)$ の 2 通りの連鎖律分解はいずれも 0.3681 nat を与える。排他的論理和では、Z で条件付けることにより、X と Y の相互情報量は 0 から 1 ビット分の全情報量である $\ln 2 = 0.693$ nat へと跳ね上がる。

処理によって情報がまったく失われないのはどのような場合だろうか。命題より、等号が成り立つのは $X \to Z \to Y$ もマルコフ連鎖であるとき、かつそのときに限る。すなわち Z が、Y から得られる X についての情報をすべて保持している場合である。可逆変換 $Z = g(Y)$ は明らかにこれを達成する。Y を Z から復元できるからである。しかし写像が Y の一部を捨てたとしても、X についての情報を失わないことがある。捨てる部分が X と無関係でありさえすればよい。た

とえばコイン投げの系列における表の合計回数は、その系列がコインの偏りについて語る情報をすべて保存している。出た順序はコインの偏りについて何の情報ももたないからである。この性質をもつ任意の要約量 $T(Y)$（すなわち $p(x \mid y)$ が y に $T(y)$ を通じてのみ依存し、したがって $I(X;T(Y)) = I(X;Y)$ となるもの）を**十分統計量**（sufficient statistic）と呼ぶ。指数分布族（36.4 節）の十分統計量はまさにこの種の要約量であり、そこでは Y がデータの役割を、X がパラメータ θ（$I(\theta;Y)$ が適切に定義されるよう事前分布を備えたもの）の役割を果たす。

データ処理不等式そのものには、エントロピーの有限性も確率密度の存在も必要ない。相互情報量は KL ダイバージェンスであり、ランダムであれ決定論的であれデータのいかなる写像も KL ダイバージェンスを増加させないため（可算集合については 37.3.3.3 節で証明しており、完全な一般性のもとで成り立つ）、演習問題 4 ではこの事実からデータ処理不等式の一般的な証明を導く。さらに、この不等式を可逆写像とその逆写像に 1 回ずつ適用すれば、滑らかさを仮定することなく 37.6.1.5 節の再パラメータ化不変性も得られる。

深層ネットワークにデータ処理不等式を適用する際には、誤解が生じやすい。ネットワークの各層はマルコフ連鎖 $X \to Z_1 \to Z_2 \to \cdots$ をなすため、$I(X;Z_\ell)$ は深さ ℓ とともに増加することはない。ここから本節の終わりまでは、Y を連鎖の中間変数ではなく予測対象のラベルを表すものとする。ネットワークがラベルについて知り得ることはすべて入力を経由しなければならないため、$Y \to X \to Z_1 \to Z_2 \to \cdots$ もマルコフ連鎖となり、$I(Y;Z_\ell)$ もまた ℓ とともに増加し得ない。いかなるアーキテクチャも入力に存在しなかったラベル情報を無から生み出すことはできず、前の層が捨てた情報を後の層が取り戻すこともできない。

これは、深さが役に立たないという意味では**ない**。隠れ層はラベルについての情報を増やすのではなく、単純な最終層で読み取れるように既存の情報を再編成するのである。ラベル Y を公平なビットとし、入力 $X \in \mathbb{R}^2$ は半径 1 の円周上の一様分布（$Y=0$ のとき）または半径 2 の円周上の一様分布（$Y=1$ のとき）に従うとする。入力はラベルを完全に決定するため $I(Y;X) = \ln 2$ であるが、2 つの同心円を分離できる直線は存在しない。最良の線形分類器は決定境界を内側の円に接するように置き、内側の円全体と、直線の外側に位置する外側の円の 1/3 を正しく分類して、正解率 $2/3$ を達成する。ここで単一の非線形特徴量 $Z = \|X\|^2$ を適用すると、その値は 1 と 4 の 2 通りしかとらない。両者の間に単純な閾値を設ければすべての点を完璧に分類できるが、$I(Y;Z) = \ln 2$ は $I(Y;X)$ より大きくなったわけではない。この変換は情報を一切増やさなかったが、すべての情報を線形に読み取れる形へと変えたのである。演習問題 2 では、制限された分類器の族が抽出できる分だけを数える利用可能情報量（usable information）の概念を定式化する。

入力側の連鎖 $X \to Z_1 \to Z_2 \to \cdots$ については、最後に 1 つ注意点がある。決定論的な層では $H(Z_\ell \mid X) = 0$ である。表現が離散化または量子化されていれば $I(X;Z_\ell)$ は単にエントロピー $H(Z_\ell)$ に帰着し、連続であれば $I(X;Z_\ell)$ は通常無限大になる（37.6.1.3 節）。したがって、層が入力についてどれだけ保持しているかの尺度として $I(X;Z_\ell)$ が意味をもつのは、層がノイズを注入するか離散出力を生成する場合に限られる。

37.6.2.3 ファノの不等式

正確な予測には相互情報量が不可欠である。誤り率 5% を $1{,}000$ クラス分類で達成する分類器は、ラベルのエントロピーの大半を含む入力を受け取っていなければならず、ファノの不等式はその必要量を正確に教えてくれる。

命題（ファノの不等式）。 **Y が k 個の要素からなる集合に値をとるとし、$\hat{Y} = g(X)$ を Y に対する X から計算された同じ集合上の任意の推定値とし、$P_e = P(\hat{Y} \neq Y)$ をその誤り確率とする。このとき次が成り立つ。**

$$H_b(P_e) + P_e \log(k-1) \geq H(Y \mid X) = H(Y) - I(X;Y). \tag{37.59}$$

左辺には、推定値 $\hat{Y}$ を修復するコストとしての明快な符号化解釈がある。まず $H_b(P_e)$ nat を使って $\hat{Y}$ が誤っているかどうかを指定し、誤っている場合（確率 P_e）には高々 $\log(k-1)$ nat を使って残りの $k-1$ 個のラベルのどれが正解かを指定する。推定値と修復情報の両方をもっていれば Y が正確にわかるため、修復のコストは Y について $\hat{Y}$ が残す不確実性を下回ることはない。そしてデータ処理不等式により、その不確実性は少なくとも X が残す不確実性以上である。

証明では、誤りが生じたかどうかを表す指示変数というただ 1 つの道具を使って、この直観を厳密な不等式へと仕立てる。残りは連鎖律だけである。

証明。 $W = \mathbf{1}\{\hat{Y} \neq Y\}$ を誤りの指示変数とする。推定値が与えられたもとでの誤りの有無と真のラベルの双方に関する残りの不確実性 $H(W, Y \mid \hat{Y})$ を 2 通りに展開する。第 1 に、W は $(Y, \hat{Y})$ によって完全に定まるためエントロピーを増やさず、$H(W, Y \mid \hat{Y}) = H(Y \mid \hat{Y})$ となる。第 2 に、連鎖律より $H(W, Y \mid \hat{Y}) = H(W \mid \hat{Y}) + H(Y \mid W, \hat{Y})$ が得られ、各項を個別に上から抑える。

- $H(W) - H(W \mid \hat{Y}) = I(W; \hat{Y}) \geq 0$ であるから、条件付けによってエントロピーは増加しない。したがって $H(W \mid \hat{Y}) \leq H(W) = H_b(P_e)$ である。
- $W = 0$ のときラベルは $\hat{Y}$ に等しく、不確実性は 0 である。$W = 1$ のとき（確率 P_e で生じる）、ラベルは $k-1$ 個ある $\hat{Y}$ 以外の値のいずれかであるため、そのエントロピーは高々 $\log(k-1)$ である。W について平均をとると $H(Y \mid W, \hat{Y}) \leq P_e \log(k-1)$ が得られる。

両方の上界を合わせると $H(Y \mid \hat{Y}) \leq H_b(P_e) + P_e \log(k-1)$ が得られる。最後に、$\hat{Y}$ は X のみから計算されるため 3 つ組 $Y \to X \to \hat{Y}$ はマルコフ連鎖をなし、データ処理不等式より $I(Y; \hat{Y}) \leq I(Y; X)$、すなわち同値な形として $H(Y \mid \hat{Y}) \geq H(Y \mid X)$ が従う。$\blacksquare$

第 2 段階における残りのラベル数 $k-1$ は、$\hat{Y}$ が同じラベル集合の中に値をとることに依拠している点に注意しよう。もし分類器に「棄権」という回答を許せば、$\log k$ で $\log(k-1)$ を置き換えなければならず、上の不等式は成り立たなくなる。実際、公平なビット Y が X と独立であり、分類器が常に棄権するとすれば、$P_e = 1$ となって左辺は $H_b(1) + \ln 1 = 0$ となり、$H(Y \mid X) = \ln 2$ を下回ってしまう。

最も単純な場合では、ファノの不等式は等号で成り立つ。反転ビットにおいて推定値を $\hat{Y} = X$ とすると、$P_e = 0.1$ かつ $k = 2$ であるから、左辺は $H_b(0.1) + 0.1 \ln 1 = 0.325$ となり、$H(Y \mid X)$ と厳密に一致する。

アーキテクチャや訓練手順がいかなるものであろうと、ファノの限界を超える分類器は存在しない。均衡した 1,000 クラスでは $H(Y) = \ln 1000 = 6.91$ nat であり、Equation 37.59 を $I(X; Y)$ について整理すると、各目標誤り率を達成するのに必要な最小の相互情報量がわかる。

```
H_b = lambda p: -p * np.log(p) - (1 - p) * np.log(1 - p)   # binary entropy
k = 1000
for p_e in (0.01, 0.05, 0.2, 0.5):
    need = np.log(k) - H_b(p_e) - p_e * np.log(k - 1)
    print(f'P_e = {p_e:4.2f}: I(X;Y) >= {need:.3f} of log k = {np.log(k):.3f} nats')
```

```
P_e = 0.01: I(X;Y) >= 6.783 of log k = 6.908 nats
P_e = 0.05: I(X;Y) >= 6.364 of log k = 6.908 nats
P_e = 0.20: I(X;Y) >= 5.026 of log k = 6.908 nats
P_e = 0.50: I(X;Y) >= 2.761 of log k = 6.908 nats
```

誤り率 5% に到達するには少なくとも 6.36 nat（ラベルエントロピー 6.91 nat の 92%）が必要であり、半分の確率で誤る分類器でさえ 2.76 nat、すなわち全体の 2/5 を必要とする。

37.6.2.4 損失による保証

ファノの不等式は分類誤り率を相互情報量の下界へと変換する。しかし言語モデルはハードな誤り率では評価されない。最尤法で訓練されるほぼすべてのモデルと同様、交差エントロピー損失を報告するのである。この損失からも相互情報量の下界が得られ、その証明はわずか 3 行で済む。

命題（損失は情報量を保証する）。 **$q(y \mid x)$ を入力に対するラベルの任意のモデルとし、交差エントロピー損失を $L = \mathbb{E}[-\log q(Y \mid X)]$（$P_{X,Y}$ のもとでの期待値）とする。このとき次が成り立つ。**

$$I(X;Y) \;\geq\; H(Y) - L, \tag{37.60}$$

さらに、均衡した k クラスに対しては $I(X;Y) \geq \log k - L$ が成り立つ。

証明。 定義より $I(X;Y) = H(Y) - H(Y \mid X)$ である。各 x に対し、$q(\cdot \mid x)$ の $p(\cdot \mid x)$ のもとでの交差エントロピーは、$p(\cdot \mid x)$ の真の条件付きエントロピーを $D_{\mathrm{KL}}(p(\cdot \mid x) \,\|\, q(\cdot \mid x)) \geq 0$ だけ上回る（37.1 節）。x について平均をとれば $H(Y \mid X) \leq L$ が得られ、これを $H(Y)$ から引けば証明が完了する。■

ホールドアウトしたテスト損失は L を推定するため、任意の分類器や言語モデルのテスト損失は、入力がもつ情報量の下界を直接保証する。たとえば均衡したラベルに対する 10 クラス分類器のテスト損失が 0.3 nat であれば、入力は少なくとも $\ln 10 - 0.3 = 2.0$ nat（ラベルの 2.30 nat のうち）を保持していることが保証される。この下界はモデルの較正が不十分であったり訓練が不出来であったりしても成り立ち、弱いモデルは単に小さな下界を保証するにすぎない。これは分類器をデコーダとして用いた Barber and Agakov (2003) の変分下界であり、37.7.3 節ではこれをより広い相互情報量推定量の族へと発展させる。

この保証がタイトになるのは、$q(y \mid x)$ が真の条件付き分布 $p(y \mid x)$ に一致するとき、かつそのときに限る。反転ビットにおいて $\hat{Y} = X$ を確信度 c で予測するモデルの期待損失は $L = -0.9 \log c - 0.1 \log(1 - c)$ であり、以下ではこれを 4 つの確信度について評価する。

```
for c in (0.6, 0.8, 0.9, 0.99):
    L = -(0.9 * np.log(c) + 0.1 * np.log(1 - c))   # cross-entropy of q
    print(f'confidence {c:4.2f}: loss {L:.4f}, '
          f'certifies I(X;Y) >= {np.log(2) - L:.4f} nats')
```

```
confidence 0.60: loss 0.5514, certifies I(X;Y) >= 0.1418 nats
confidence 0.80: loss 0.3618, certifies I(X;Y) >= 0.3314 nats
confidence 0.90: loss 0.3251, certifies I(X;Y) >= 0.3681 nats
confidence 0.99: loss 0.4696, certifies I(X;Y) >= 0.2236 nats
```

較正されたモデル（$c = 0.9$）だけが全情報量である 0.368 nat を保証する。過信したモデル（$c = 0.99$）は誤るたびに大きなペナルティを支払い 0.22 nat しか保証できず、自信不足のモデル（$c = 0.6$）は 0.14 nat にとどまる。しかし 4 つのモデルはすべて同一のハード予測を出力し 10% の確率で誤るため、誤り率しか見ないファノの不等式はどのモデルに対しても 0.368 nat すべてを保証する。交差エントロピー損失は予測確率まで見ているため、較正のずれにペナルティを科すのである。どちらの結果も同じ条件付きエントロピー $H(Y \mid X)$ を上から抑え（ファノの不等式は誤り率により、損失による保証は損失により）、それぞれの上界を $I(X;Y)$ の下界へと変換している。実は両者の結びつきは見た目以上に深い。演習問題 6 で示すように、ファノの下

界自体がある特定のモデルに対する損失による保証にほかならず、自身がいつ不確かであるかを把握している適切に較正された分類器は、それより真に多くの情報量を保証できる。

37.6.3 通信と圧縮

Shannon は 2 つの根源的な工学的問いを立てたが、その答えは相互情報量の最大値と最小値に帰着する。雑音通信路はどれだけの情報を確実に運べるか、そして非可逆な圧縮器はどれだけ少ない情報の保持で済むか、という問いである。表現学習に関心がある読者は初読時には通信路容量（channel capacity）を流し読みしても構わないが、レート・歪み理論（rate–distortion theory）は本節の最後のパートが直接その上に構築されるため不可欠である。

37.6.3.1 通信路容量

各ビットを独立に確率 δ で反転させる通信回線を、どれだけの情報が生き残れるだろうか。無記憶**通信路**（channel）は、各入力記号に独立に適用される条件付き分布 $p(y \mid x)$ によって規定され、送信される記号 1 つを通信路の 1 回の使用と数える。通し例の反転ビットは、入力 X と出力 Y をもつまさにそのような通信路である。通信路のノイズは固定されているが、送信者は入力分布 $p(x)$ を自由に選ぶことができる。通信路の**容量**（capacity）とは、入力分布のあらゆる選び方にわたって達成可能な最大の相互情報量であり、1 回の使用あたりの nat 数で測られる。

$$C = \max_{p(x)} I(X;Y). \tag{37.61}$$

ほとんど 0 ばかりを送る送信者は、そもそも入力自体のエントロピーが小さいため、通信路を十分に活用できていない。シャノンの通信路符号化定理は、この量に操作的な意味を与える。符号の**レート**を通信路の 1 回の使用あたりに伝達するメッセージの nat 数と定義したとき、Shannon は、記号の長いブロックをまとめて符号化することで、C 未満の任意のレートにおいて誤り確率をいくらでも小さく抑えた高信頼な通信が可能である一方、C を超えるいかなるレートでも高信頼な通信は不可能であることを示した (Shannon 1948; Cover and Thomas 1999)。この定理は証明なしで認めることとする。

2 元対称通信路（binary symmetric channel）として知られるビット反転通信路では、容量は 1 行で計算できる。$I(X;Y) = H(Y) - H(Y \mid X)$ と書く。$p(y \mid x)$ のどの行も順序を除けば $(1-\delta, \delta)$ であるから、入力分布によらず $H(Y \mid X) = H_b(\delta)$ となる。一方、$H(Y) \le \ln 2$ であり、等号は Y が一様分布であるとき、かつそのときに限り成り立つ。$P(Y=1) = \delta + P(X=1)(1-2\delta)$ より、$\delta \neq \tfrac{1}{2}$ である限り、出力が一様になるのは入力が一様であるとき、かつそのときに限る。したがって次式を得る。

$$C = \ln 2 - H_b(\delta) \quad \text{nats per use}, \tag{37.62}$$

この最大値は一様な入力によって達成される。$\delta = 0.1$ では、（入力が最初から一様であった）反転ビットの 0.368 nat が復元され、直列の 2 本の回線（$\delta = 0.26$）では容量は 0.120 nat へと低下する。$\delta = \tfrac{1}{2}$ では出力は入力と独立になり、容量は 0 になる。$\delta = 1$ では容量は全情報量である $\ln 2$ nat に戻る。すべてのビットが反転していると知っている受信者は単にそれを反転し直せばよいからである。ノイズが情報を破壊するのは、受信者がそれを予測できない範囲に限られる。以下のコードでは、まず $\delta = 0.1$ の通信路に対して入力分布を動かし、続いて一様入力のもとで反転確率 δ を動かす。

```
inputs = (0.1, 0.3, 0.5, 0.7, 0.9)
I_in = [mutual_information(np.array([[1 - a], [a]]) * flip(0.1)) for a in inputs]
print('P(X=1):', '  '.join(f'{a:6.2f}' for a in inputs))
print('I(X;Y):', '  '.join(f'{v:6.4f}' for v in I_in))
deltas = (0.0, 0.1, 0.26, 0.5, 1.0)
C = [np.log(2) - entropy(np.array([d, 1 - d])) for d in deltas]
print('delta: ', '  '.join(f'{d:6.2f}' for d in deltas))
print('C:     ', '  '.join(f'{c:6.4f}' for c in C))
```

```
P(X=1):   0.10    0.30    0.50    0.70    0.90
I(X;Y): 0.1463  0.3160  0.3681  0.3160  0.1463
delta:    0.00    0.10    0.26    0.50    1.00
C:      0.6931  0.3681  0.1201  0.0000  0.6931
```

1 を 10% の割合でしか送らない入力は、そもそも $H_b(0.1) = 0.325$ nat しかもっておらず、そのうち通信路を通過できるのは 0.146 nat、すなわち一様入力が達成する量の 2/5 にすぎない。Figure 37.16 の左パネルに、容量を δ の関数として描く。ニューラルネットワークの内部にも独自の通信路が存在する。ドロップアウトは特徴量を確率 α で消去し、それ以外の場合には通過させるが、演習問題 7 では 2 値の特徴量がこの通信路を通して高々 $(1-\alpha)\ln 2$ nat しか送れないことを示す。

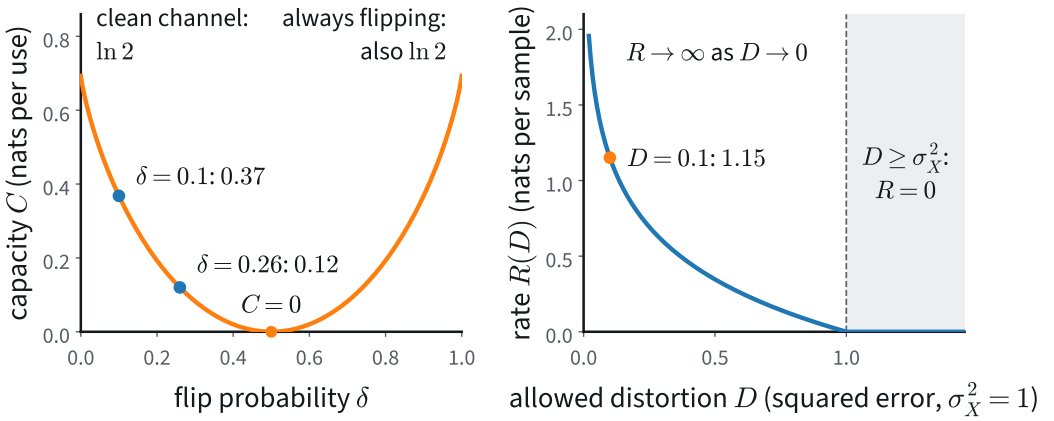


Figure 37.16: nat 単位で表した Shannon の 2 つの操作曲線。左：2 元対称通信路の容量 $C = \ln 2 - H_b(\delta)$。反転ビット（$\delta = 0.1$）と直列の 2 回反転（$\delta = 0.26$）に印をつけてある。確率 $1/2$ で反転する通信路は何も運べない一方、常に反転する通信路は $\ln 2$ すべてを運ぶ。右：分散 $\sigma_X^2 = 1$ の情報源に対するガウス分布のレート・歪み関数。レートは $D \to 0$ で発散し、D が情報源の分散に達すると 0 になる。

37.6.3.2 レート・歪み

可逆圧縮には 1 記号あたり $H(X)$ nat が必要であり（37.2 節）、連続情報源を厳密に可逆符号化しようとすれば無限に多くの nat が必要になる。しかし実用上は、近似的な再構成で十分なことがほとんどである。画像コーデックはピクセルをわずかに変化させてもよいし、学習された表現は無関係な細部を意図的に捨ててもよい。非負の**歪み**（distortion）尺度 $d(x, \hat{x}) \geq 0$ を選んで $\hat{x}$ が x をどれだけうまく再構成しているかを測り、平均歪みを高々 D まで許容するとしよう。1 サンプルあたりどれだけ少ない nat 数で済むだろうか。**レート・歪み関数**（rate–distortion

function）は、歪みの許容範囲内に収めるためにエンコーダ $p(\hat{x} \mid x)$ が保持しなければならない最小の相互情報量としてこの問いに答える。

$$R(D) = \inf_{p(\hat{x}|x):\ \mathbb{E}[d(X,\hat{X})] \le D} I(X;\hat{X}). \tag{37.63}$$

許容歪みを厳しくするほど、より多くの情報量が求められる。シャノンのレート・歪み定理は通信路符号化定理と対をなしており、ここでのレートは符号が情報源の 1 サンプルあたりに費やす nat 数を測る。すなわち、$R(D)$ を上回る任意のレートで動作するブロック符号は平均歪み D を達成できる一方、$R(D)$ 未満のいかなる符号もそれを達成できない (Shannon 1959)。

実際には、ハードな歪み制約をペナルティに置き換え、$I(X;\hat{X}) + \lambda\,\mathbb{E}[d(X,\hat{X})]$ を乗数 $\lambda > 0$ のもとですべてのエンコーダにわたって最小化するほうが扱いやすいことが多い。各エンコーダは歪み・レート平面において曲線 $R(D)$ 上またはその上方の 1 点に対応し、ペナルティ付き目的関数は傾き $-\lambda$ の直線上で一定となる。$R(D)$ は凸関数であるため、最小値は傾き $-\lambda$ の直線が曲線に下から接する点で達成される。λ を大きくするほど、より高いレートでより小さい歪みが選ばれる。37.6.4 節の情報ボトルネックは、まさにこのラグランジアン形式で定式化される。

命題（ガウス情報源のレート・歪み）。 **情報源 $X \sim \mathcal{N}(0,\sigma_X^2)$ と二乗誤差 $d(x,\hat{x}) = (x-\hat{x})^2$ のもとで、次が成り立つ。**

$$R(D) = \frac{1}{2}\log\frac{\sigma_X^2}{D} \quad (0 < D < \sigma_X^2), \qquad R(D) = 0 \quad (D \ge \sigma_X^2). \tag{37.64}$$

この式を歪みについて解くと $D = \sigma_X^2 e^{-2R}$ となる。レートを 1 ビット（$\ln 2$ nat）増やすごとに、二乗誤差は 1/4 に減少する。

証明。 主たるステップは下界の導出であり、37.1.4.3 節のただ 1 つの事実に依拠する。すなわち、分散 s^2 をもつすべての密度の中で微分エントロピーを最大化するのはガウス分布であり、その最大値 $\frac{1}{2}\log(2\pi e s^2)$ は s^2 とともに単調に増加する。$\mathbb{E}[(X-\hat{X})^2] \le D$ を満たす任意のエンコーダをとると、次が成り立つ。

$$\begin{aligned} I(X;\hat{X}) &= h(X) - h(X \mid \hat{X}) = h(X) - h(X-\hat{X} \mid \hat{X}) \\ &\ge h(X) - h(X-\hat{X}) \\ &\ge \tfrac{1}{2}\log(2\pi e\sigma_X^2) - \tfrac{1}{2}\log(2\pi e D) = \tfrac{1}{2}\log\frac{\sigma_X^2}{D}. \end{aligned}$$

2 つ目の等号は、既知の値 $\hat{X}$ を引いても条件付き密度が平行移動するだけでエントロピーは変わらないことから従う。1 つ目の不等号は条件付けがエントロピーを増加させないという事実を用い、2 つ目の不等号は再構成誤差 $X-\hat{X}$ の分散が高々 $\mathbb{E}[(X-\hat{X})^2] \le D$ であることからガウス分布の最大エントロピー上界を適用している。$D \ge \sigma_X^2$ のときは、何も送信せず常に平均 0 を出力する定数エンコーダがすでに許容歪みを満たすため、$R(D) = 0$ となる。

残るは $D < \sigma_X^2$ に対してこの下界を達成するエンコーダの構成である。$\sigma_X^2 = 1$ とおき、独立なガウスノイズを加えて $Z = X + \sigma\epsilon$（ただし $\epsilon \sim \mathcal{N}(0,1)$、$\sigma > 0$）とする。このとき $\mathrm{Cov}(X,Z) = 1$ かつ $\mathrm{Var}(Z) = 1 + \sigma^2$ であるから、二乗相関は $\rho_{XZ}^2 = 1/(1+\sigma^2)$ となる。Equation 37.56 を適用すると、このエンコーダが保持する相互情報量は次で与えられる。

$$I(X; X+\sigma\epsilon) = \tfrac{1}{2}\log\left(1+\sigma^{-2}\right). \tag{37.65}$$

ここで X を条件付き平均 $\hat{X} = \mathbb{E}[X \mid Z] = Z/(1+\sigma^2)$ によって再構成する。これは X の Z に対する傾き $\mathrm{Cov}(X,Z)/\mathrm{Var}(Z)$ の最適な線形回帰である。その平均二乗誤差は条件付き分散 $D = 1 - 1/(1+\sigma^2) = \sigma^2/(1+\sigma^2)$ であり、開区間 $(0,1)$ 全体を σ の変化に伴って動く。$\hat{X}$ は Z の可逆な定数倍であるため、$I(X;\hat{X}) = \tfrac{1}{2}\log(1+\sigma^{-2}) = \tfrac{1}{2}\log(1/D)$ が成り立つ。一般の分散 σ_X^2 に対しては、単に X と $\hat{X}$ を定数倍すればよい。■

この証明で用いたエンコーダ（独立なガウスノイズで汚された信号）は、信号対雑音比（signal-to-noise ratio）σ^{-2} の**ガウス通信路**（Gaussian channel）として知られ、Equation 37.65 はその容量である（演習問題 8）。拡散モデル（第 17 章）の順方向ノイズ付加過程はまさにこの通信路にほかならない。データ分布がいかなるものであっても、信号対雑音比 s までノイズを加えられた座標は、元のクリーンな値について高々 $\tfrac{1}{2}\log(1+s)$ nat しか保持しない。以下のコードは、100 万個のサンプル上でこのエンコーダを $\sigma^2 = 1/9$、1、3（それぞれ $D = 0.1$、0.5、0.75 に対応）においてシミュレーションし、二乗誤差を検証する。

```
rng = np.random.default_rng()
x = rng.standard_normal(1_000_000)
for s2 in (1 / 9, 1.0, 3.0):                                # sigma^2
    z = x + np.sqrt(s2) * rng.standard_normal(x.size)       # encoder
    x_hat = z / (1 + s2)                                    # E[X | Z]
    D = s2 / (1 + s2)
    mse = np.mean((x - x_hat)**2)
    print(f'sigma^2 = {s2:5.3f}: D = {D:.3f}, MSE {mse:.3f}, '
          f'I(X;Z) = R(D) = {0.5 * np.log(1 + 1 / s2):.3f} nats')
```

```
sigma^2 = 0.111: D = 0.100, MSE 0.100, I(X;Z) = R(D) = 1.151 nats
sigma^2 = 1.000: D = 0.500, MSE 0.500, I(X;Z) = R(D) = 0.347 nats
sigma^2 = 3.000: D = 0.750, MSE 0.750, I(X;Z) = R(D) = 0.144 nats
```

経験二乗誤差は小数第 3 位の数単位以内で D と一致する。$D = 0.1$ の歪みには 1 サンプルあたり 1.15 nat、$D = 0.5$ には 0.35 nat（半ビット）、$D = 0.75$ には 0.14 nat のコストがかかる。

このノイズ付きエンコーダが $R(D)$ を達成しているのは相互情報量としてであり、まだ実際のファイルサイズとしてではないことに留意されたい。実用的な符号はサンプルを離散的なビット列へと写像しなければならず、サンプルの長いブロックをまとめて符号化することで初めてレート・歪み曲線に漸近する。1 度に 1 サンプルずつ量子化するいかなるスカラー符号も、曲線より真に上にとどまる。最も単純な例は X の符号のみを保持するもので、$\ln 2 = 0.693$ nat を費やして $\pm\sqrt{2/\pi}$（$|X|$ の平均）を再構成し、二乗誤差 $1 - 2/\pi = 0.363$ を生じるが、レート・歪み曲線がその歪みに対して要求するのはわずか 0.506 nat である。

Figure 37.16 の右パネルに曲線全体を示す。$D \to 0$ につれて必要レートは限りなく上昇し（連続変数の厳密なコピーがもつ無限の情報量を反映している）、0 まで落ちるのは D が σ_X^2 に達したときである。$D = 0$ 付近でのこの急激な立ち上がりは、現代の画像生成器がモデル化の前に画像を圧縮する理由を説明している。知覚できないほど微細なピクセルの細部がビットの大半を消費してしまうため、17.7 節ではそうした細部をオートエンコーダに委ねるのである。

37.6.4 情報ボトルネック

優れた表現は、ラベルの予測に役立つものを残し、それ以外を捨てるべきである。一見すると、何かを捨てることは逆効果に思えるかもしれない。データ処理不等式によれば、情報を捨ててもベイズ最適な予測器が改善することは決してないからである。しかし実際には、無関係な細部には現実のコストが伴う。犬の写真には、照明や芝生、さらにはフレーム内のどこにたまたま犬が立っているかまでもが符号化されている。そうした細かな違いをすべて保存するには余分な nat が

かかり、有限枚の画像で訓練された分類器はそれらをラベルへの真の手がかりと容易に取り違えてしまう。情報ボトルネックは、このトレードオフを 2 つの相互情報量によって定式化する。それはラベル自体が歪みを定義するレート・歪み問題にほかならない。

37.6.4.1 ボトルネック目的関数

Z を入力 X から確率的エンコーダ $p(z \mid x)$ によって生成される表現とし、Y を目的ラベルとする。Z が連続であるとき、エンコーダは確率的でなければならない。決定論的な写像では通常 $I(X;Z) = \infty$ となってしまうからである（37.6.1.3 節）。**情報ボトルネック** (Tishby et al. 1999) は、表現 Z が X の情報をできる限り削ぎ落としつつ、Y についての情報をできる限り多く保持することを目指す。

$$\min_{p(z|x)} \; I(X;Z) - \beta\, I(Y;Z). \tag{37.66}$$

ここで $I(X;Z)$ はレート（表現が入力について保持する情報量）であり、$I(Y;Z)$（重み $\beta > 0$ を掛ける）は表現がラベルについて保持する情報量である。エンコーダは X しか参照しないため、変数はマルコフ連鎖 $Y \to X \to Z$ をなし、データ処理不等式により $I(Y;Z) \le \min\{I(X;Y),\, I(X;Z)\}$ という制約が課される。表現は入力以上にラベルを知ることは決してできず、また入力情報を保持せずにうまく予測することもできない。

乗数 β がこのトレードオフを制御しており、その両極端は容易に理解できる。$\beta \le 1$ である限り、何も保持しないことが最適である。実際、上の不等式から $I(X;Z) - \beta I(Y;Z) \ge (1-\beta)\, I(X;Z) \ge 0$ であり、X を完全に無視する定数エンコーダが $I(X;Z) = I(Y;Z) = 0$ を達成する。定数エンコーダが最適になるとき、ボトルネックは**崩壊**（collapse）するという。逆に β が大きくなるとラベルの項が支配的になり、エンコーダは Y の予測に役立つあらゆる情報を保持するようになる。その極限では $I(Y;Z)$ は上限 $I(X;Y)$ に近づき、表現は**最小十分統計量**（minimal sufficient statistic）、すなわち X の要約のうち Y を X 自身と同等の精度で予測できる最も粗いものへと近づく。もちろん、あらゆるニューラルネットワークがそうであるようにエンコーダがパラメトリックな族に制限されている場合には、その上限の手前で頭打ちになることがある。

命題（レート・歪み問題としてのボトルネック）。 *$Y \to X \to Z$ がマルコフ連鎖であるならば、次が成り立つ。*

$$I(Y;Z) = I(X;Y) - \mathbb{E}_{(x,z)\sim P_{X,Z}}\Big[D_{\mathrm{KL}}\big(p(y \mid x) \,\|\, p(y \mid z)\big) \Big]. \tag{37.67}$$

証明。 $I(Y;X,Z)$ を連鎖律により両方の順序で展開すると $I(Y;Z) + I(Y;X \mid Z) = I(Y;X) + I(Y;Z \mid X)$ となる。マルコフ性より $I(Y;Z \mid X) = 0$ である。残った条件付き相互情報量を KL 形式で書くと $I(Y;X \mid Z) = \mathbb{E}_{(x,z)}\big[D_{\mathrm{KL}}(p(y \mid x,z) \,\|\, p(y \mid z))\big]$ となり、マルコフ性により $p(y \mid x,z)$ は $p(y \mid x)$ に簡約される。$\blacksquare$

Equation 37.67 を目的関数に代入すると、Equation 37.66 は $I(X;Z) + \beta\, \mathbb{E}[d(X,Z)] - \beta I(X;Y)$ へと書き換わる（ただし歪みは $d(x,z) = D_{\mathrm{KL}}(p(y \mid x) \,\|\, p(y \mid z))$ である）。最後の項 $\beta I(X;Y)$ はエンコーダに依存しないため、情報ボトルネックはまさにレート・歪み問題のラグランジアン形式である。設計者が手作業で歪み尺度を作る代わりに、ラベルがそれを自動的に与えてくれる。符号 z は、x に対する忠実な表現であるのは、それが Y を x と同じ精度で予測するときなのである。ただし通常のレート・歪み問題とは異なり、この歪みは $p(y \mid z)$ を通じてエンコーダ自身に依存するため、最適化は格段に難しくなる。

37.6.4.2 ガウス情報ボトルネック

ある特別な設定では、情報ボトルネックのすべての量を閉形式で解くことができる (Chechik et al. 2005)。(X, Y) を $\rho = 0.9$ のガウス対（$I(X;Y) = 0.830$ nat）とし、37.6.3.2 節の加法的ノイズエンコーダを用いる。

$$Z = X + \sigma\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,1),$$

このエンコーダはノイズのスケール σ のみでパラメータ化される。レート $I(X;Z)$ はすでに Equation 37.65 で与えられている。ラベルについては $\mathrm{Cov}(Y,Z) = \mathrm{Cov}(Y,X) = \rho$ かつ $\mathrm{Var}(Z) = 1+\sigma^2$ であるから、$\rho_{YZ}^2 = \rho^2/(1+\sigma^2)$ となる。Equation 37.56 を適用すると、両方の相互情報量が求まる。

$$I(X;Z) = \tfrac{1}{2}\log\left(1+\sigma^{-2}\right), \qquad I(Y;Z) = -\tfrac{1}{2}\log\left(1 - \frac{\rho^2}{1+\sigma^2}\right). \tag{37.68}$$

σ が ∞ から 0 へ減少するにつれて、表現は X について何も知らない状態から X を完全に知る状態へと滑らかに移行し、ラベル情報量は 0 からデータ処理不等式の上限である 0.830 nat まで上昇する。

このトレードオフを直接可視化するため、レートを $u = I(X;Z)$ と書く（R はレート・歪み関数 $R(D)$ のために取っておく）。Equation 37.68 を逆に解くと $\sigma^{-2} = e^{2u} - 1$、ゆえに $1/(1+\sigma^2) = 1 - e^{-2u}$ となり、ラベル情報量をレートの関数として直接表せる。

$$I(Y;Z) = -\tfrac{1}{2}\log\left(1 - \rho^2(1-e^{-2u})\right), \qquad \frac{d\,I(Y;Z)}{du} = \frac{\rho^2}{\rho^2 + (1-\rho^2)\,e^{2u}}.$$

導関数は ρ^2（$u=0$ における値）から 0 に向かって単調に減少するため、この曲線は狭義凹関数である。原点を傾き $\rho^2 = 0.81$ で出発するため、最初の 1 nat の入力情報から得られるラベル情報は高々 0.81 nat であり、その後に続く 1 nat ごとの増分は $I(Y;Z)$ が $I(X;Y)$ に近づくにつれて逓減していく。Chechik et al. (2005) は、ガウス対に対して線形・非線形を問わずいかなるエンコーダもこの曲線を上回れないことを証明しており、これが厳密な最適フロンティアとなる（この結果は証明なしで用いる）。任意の β を選ぶと、ラグランジアンの最小点はこのフロンティア上のただ 1 つの**動作点**（operating point）を選び出す。

命題（動作点と崩壊）。$\rho^2 < 1$ のガウス情報ボトルネックにおいて、ラグランジアン $u - \beta\, I(Y;Z)$ は u に関して凸であり、その導関数は $1-\beta\rho^2$（$u=0$ における値）となる。$\beta \le 1/\rho^2$ では最小点は $u = 0$（定数エンコーダ）である。$\beta > 1/\rho^2$ では、最小点とそのラベル情報量は次で与えられる。

$$u^{*} = \frac{1}{2}\log\frac{\rho^2(\beta-1)}{1-\rho^2}, \qquad I(Y;Z)^{*} = \frac{1}{2}\log\frac{\beta-1}{\beta(1-\rho^2)}. \tag{37.69}$$

証明。 $I(Y;Z)$ は u に関して凹であるため、ラグランジアン $u - \beta I(Y;Z)$ は凸であり、導関数は $1 - \beta\rho^2/(\rho^2 + (1-\rho^2)e^{2u})$ である。$u=0$ ではこの導関数は $1-\beta\rho^2$ に等しい。$\beta\rho^2 \le 1$ のときは常に、導関数は $u=0$ で非負かつ狭義単調増加であるため、$u \ge 0$ における最小値は $u=0$ で達成される。そうでない場合、導関数を 0 とおくと $\rho^2 + (1-\rho^2)e^{2u} = \beta\rho^2$、すなわち $e^{2u} = \rho^2(\beta-1)/(1-\rho^2)$ が得られる。$\rho^2 e^{-2u} = (1-\rho^2)/(\beta-1)$ を $I(Y;Z) = -\tfrac{1}{2}\log(1-\rho^2+\rho^2 e^{-2u})$ に代入すれば 2 つ目の式が得られる。■

先ほどは一般のデータ処理不等式の議論から $\beta \le 1$ での崩壊が保証された。しかし $\rho = 0.9$ のガウス対では、実際には崩壊が $\beta \le 1/\rho^2 = 1.235$ に至るまで続く。この閾値以下では、レートを 1 nat 増やすごとに目的関数に 1 が加わる一方で減少量は高々 $\beta\rho^2 \le 1$ にとどまる。フロンティアの傾きがその初期値 ρ^2 を決して超えないからである。閾値を超えると、β の増加に伴って動作点はフロンティアに沿って上限へと滑り上がっていく。$\beta = 2$ ではレート $u^* = \tfrac{1}{2}\ln(0.81/0.19) = 0.725$ nat とラベル情報量 0.484 nat が得られ、$\beta = 16$ ではそれぞれ 2.079 nat と 0.798 nat に達する。Figure 37.17 は、横軸にレート $I(X;Z)$、縦軸にラベル情報量 $I(Y;Z)$ をとった**情報平面**（information plane）上に、このフロンティアと各動作点を描いたものである。

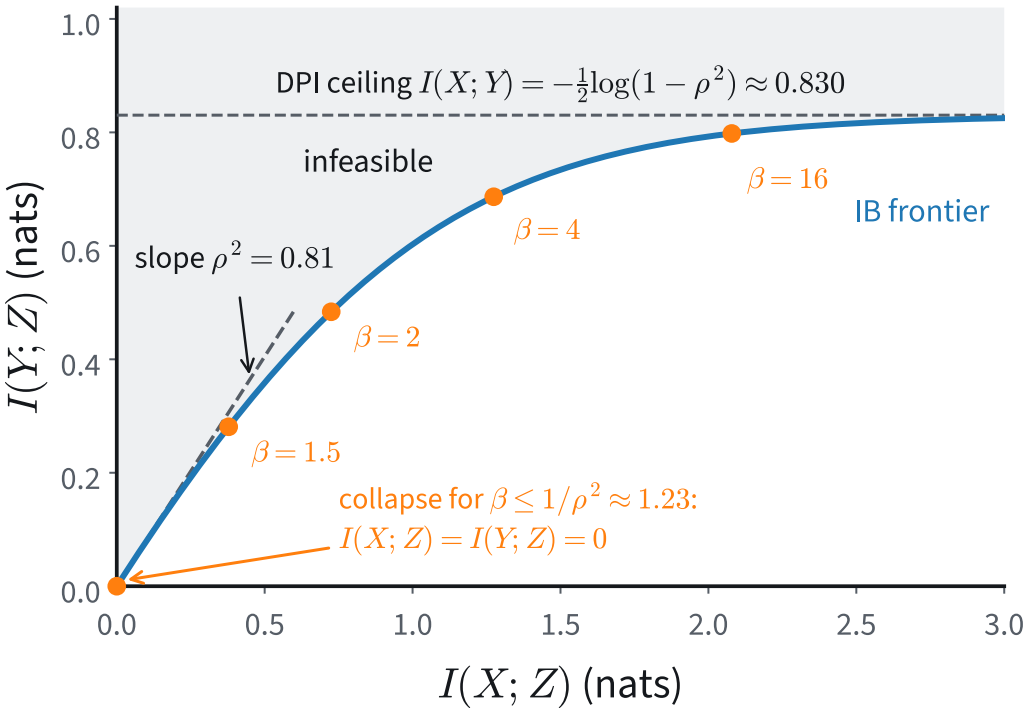


Figure 37.17: $\rho = 0.9$ のガウス情報ボトルネックの情報平面。凹フロンティアは達成可能な最大の $I(Y;Z)$ を各レート $I(X;Z)$ に対してたどり、その上方の網掛け領域は実行不可能である。破線の上限はデータ処理不等式の限界 $I(X;Y) = -\tfrac{1}{2}\log(1-\rho^2) \approx 0.830$ nat を表す。フロンティアは原点を傾き ρ^2 で出発し、乗数は傾きが $1/\beta$ に等しくなる点を選択する。$\beta \le 1/\rho^2 \approx 1.23$ ではその点は原点へと崩壊する。

これほど鋭い相転移は数値的にも確かめておく価値がある。命題によれば、最適エンコーダは $\beta = 1.2$ では何も保持しないが、$\beta = 1.25$ では情報を保持するはずである。以下では閉形式解を用いずに、4,001 個のノイズスケールからなるグリッド上でラグランジアンを直接最小化する。

```
rho = 0.9
sigma = np.logspace(-2, 2, 4001)                  # encoder Z = X + sigma eps
I_xz = 0.5 * np.log(1 + sigma**-2)
I_yz = -0.5 * np.log(1 - rho**2 / (1 + sigma**2))
for beta in (1.2, 1.25, 1.5, 2, 4, 16):
```

```
    i = np.argmin(I_xz - beta * I_yz)            # Lagrangian on the grid
    u = (0.5 * np.log(rho**2 * (beta - 1) / (1 - rho**2))
         if beta * rho**2 > 1 else 0.0)          # closed-form optimum
    print(f'beta = {beta:5.2f}: I(X;Z) grid {I_xz[i]:.3f}, closed form {u:.3f};'
          f' I(Y;Z) = {I_yz[i]:.3f}')
```

```
beta =  1.20: I(X;Z) grid 0.000, closed form 0.000; I(Y;Z) = 0.000
beta =  1.25: I(X;Z) grid 0.032, closed form 0.032; I(Y;Z) = 0.026
beta =  1.50: I(X;Z) grid 0.379, closed form 0.378; I(Y;Z) = 0.281
beta =  2.00: I(X;Z) grid 0.725, closed form 0.725; I(Y;Z) = 0.484
beta =  4.00: I(X;Z) grid 1.275, closed form 1.274; I(Y;Z) = 0.687
beta = 16.00: I(X;Z) grid 2.078, closed form 2.079; I(Y;Z) = 0.798
```

グリッド探索と閉形式解は 0.001 nat の精度で一致し、命題が予測したとおりの位置に崩壊の閾値があることが確認できる。$\beta = 1.2$ では定数エンコーダが最適であるのに対し、$\beta = 1.25$ ではエンコーダはすでに 0.032 nat を保持している。ノイズスケール σ をもつまったく同じこのエンコーダが、二乗誤差に対するレート・歪み曲線 $R(D)$ も $D = \sigma^2/(1+\sigma^2)$ において達成していることに注目しよう。レート・歪み理論はエンコーダが X をどれだけうまく再構成するかで評価するのに対し、情報ボトルネックは Y をどれだけうまく予測するかで評価するのである。

37.6.4.3 変分情報ボトルネック

ガウス対に対しては、ボトルネック目的関数に含まれる 2 つの相互情報量をどちらも解析的に計算できた。しかし実データでは真の同時分布が未知であるため、どちらも計算できない。観測できるのはサンプル (x, y) と、設計者が選んだエンコーダ $p(z \mid x)$ のみである。**変分情報ボトルネック**（variational information bottleneck）は、各相互情報量をサンプルから推定できる扱いやすい上下界に置き換えることで、この目的関数を訓練可能にする (Alemi et al. 2017)。

ラベルの項については、必要な下界がすでに手元にある。デコーダ $q(y \mid z)$（表現からラベルを予測する分類器）を導入し、その交差エントロピー損失を $L_Z = \mathbb{E}[-\log q(Y \mid Z)]$ とする。損失による保証 Equation 37.60 を対 (Z, Y) に適用すれば、直ちに $I(Y;Z) \geq H(Y) - L_Z$ が得られる。

一方、レート $I(X;Z)$ は**最小化**したい量であるため、今度は上界が必要になる。レートを条件付き形式 Equation 37.55、すなわち $I(X;Z) = \mathbb{E}_x\big[D_{\mathrm{KL}}(p(z \mid x) \,\|\, p(z))\big]$ で書こう。エンコーダ $p(z \mid x)$ は既知であるが、集約された周辺分布 $p(z)$ は $p(z \mid x)$ を未知のデータ分布で平均したものであるため未知である。そこで $p(z)$ を、標準正規分布などの任意に選んだ固定の事前密度 $r(z)$ に置き換える。恒等式 $\log \frac{p(z|x)}{r(z)} = \log \frac{p(z|x)}{p(z)} + \log \frac{p(z)}{r(z)}$ を用いて x と z について期待値をとると、次式を得る。

$$\mathbb{E}_x\Big[D_{\mathrm{KL}}(p(z \mid x) \,\|\, r(z))\Big] = I(X;Z) + D_{\mathrm{KL}}(p(z) \,\|\, r(z)) \;\geq\; I(X;Z). \tag{37.70}$$

選んだ事前分布 r と真の周辺分布 $p(z)$ の間にずれがあると、レートの上界は膨らむ方向にしか動かない。演習問題 9 では、ガウスエンコーダに対してこのギャップを明示的に計算する。両方の上下界を組み合わせると、Equation 37.70 の左辺にある KL 項に βL_Z を加えたものが、ボトルネック目的関数 Equation 37.66 の上界を（定数 $\beta H(Y)$ の差を除いて）与えることがわかる。どちらの項も訓練例に関する単純な期待値であるため、ニューラルネットワークは確率的勾配降下法（stochastic gradient descent）によってこの上界を直接最小化できる。

変分オートエンコーダ（variational autoencoder）を訓練したことがあれば、Equation 37.70 の左辺にはすぐ見覚えがあるだろう。変分オートエンコーダ（17.7 節）とは、入力自身をラベルと

したまったく同じ構成にほかならない。すなわち、エンコーダ $p(z \mid x)$、固定の事前分布 $r(z)$、そして別のクラスラベルを予測する代わりに x を z から再構成するデコーダからなる。その訓練損失は、再構成誤差に重み付きの KL ペナルティを加えたものである。本節の視点から見れば、その KL ペナルティはレートの上界であり、潜在コードが入力について蓄えられる nat 数を制限する一方、再構成誤差が歪みの役割を担っている。異なるのはトレードオフの重みを置く位置だけである。Equation 37.66 が保持される情報量に β を掛けるのに対し、変分オートエンコーダは KL レート項に重みを置くため、あちらでの大きな KL 重みはこちらでの小さな β に対応する。

デコーダの交差エントロピー損失と事前分布への KL ダイバージェンスは、本節の中でサンプルのみから評価できる唯一の量であり、一方の相互情報量には下界を、もう一方には上界を与えた。それ以外に計算してきた数値はすべて同時分布が既知であることに頼っていたが、実際の画像やテキストでは同時分布など決して手に入らない。37.7 節では、サンプルのみから相互情報量についてどこまで推定できるかを問う。

37.6.5 まとめ

相互情報量は、同時分布とその周辺分布の積との間の KL ダイバージェンス $I(X;Y) = D_{\mathrm{KL}}(P_{X,Y} \,\|\, P_X \otimes P_Y)$ であり、X と Y が独立であるとき、かつそのときに限り 0 となり、いずれの変数の可逆な再パラメータ化のもとでも不変である。線形相関とは異なり非線形な従属性を検出でき、相関が 0 であるにもかかわらず 1.14 nat を X と $Y = X^2 + 0.3\epsilon$ に割り当てる。データ処理不等式により、逐次処理は情報を生み出すことはできないが、ニューラルネットワークの層は分類器が読み取りやすいように既存の情報を並べ替えることができる。ファノの不等式は任意の目標誤り率を相互情報量の最小要件へと変換し、任意の確率モデルの交差エントロピー損失 L は $I(X;Y) \geq H(Y) - L$ を保証する。Shannon の通信路容量とレート・歪み関数は、それぞれ入力に関する相互情報量の最大値と、非可逆エンコーダに関する最小値として現れる。最後に、情報ボトルネック $\min I(X;Z) - \beta I(Y;Z)$ は表現学習をラベルによって歪みが与えられるレート・歪み問題として定式化する。ガウス対に対しては β が $1/\rho^2$ を超えるまで崩壊し、変分形式においてはそのレートペナルティが変分オートエンコーダでおなじみの KL 正則化項となる。

37.6.6 演習問題

1. [code] **低頻度対と PMI**。自己相互情報量 Equation 37.57 は、共起回数を 2 つの単語の頻度で補正する（37.6.1.4 節）。
    1. $\mathrm{pmi}(x,y) \leq \min\{-\log p_X(x), -\log p_Y(y)\}$ を示せ。2 つの上界のそれぞれが達成されるのはどのようなときか。
    2. `counts` を拡張し、バイグラム「sri lanka」の中でのみ出現回数 2 で現れる 4 番目の第 1 語 **sri** と 4 番目の第 2 語 **lanka** を追加せよ。「new york」、「machine learning」、「the day」、「sri lanka」の pmi を再計算し、さらに出現回数 1 でも同様に計算せよ。出現回数 0 のセルでは pmi $= -\infty$ となって `(p * pmi).sum()` が `nan` を返すため、`mutual_information` と同様に非ゼロのセルに限定して和をとること。
    3. 「new york」と「sri lanka」がもつ寄与 $p_{X,Y}(x,y)\,\mathrm{pmi}(x,y)$（$I(X;Y)$ に対する寄与）を比較せよ。続いてすべての出現回数に 1 を加えて両方の pmi の値を再計算せよ。連語をランク付けするにはこれらのスコアのうちどれを用いるべきか。また、それはなぜか。
2. **情報量と利用可能情報量**。ラベル Y は公平なビットである。$Y = 0$ のとき入力 $X \in \mathbb{R}^2$ は半径 1 の円周上の一様分布に従い、$Y = 1$ のときは半径 2 の円周上の一様分布に従う。表現として $Z = \|X\|^2$ を計算する。

1. $I(Y;Z) = I(Y;X) = \ln 2$ が成り立つことを示せ（ただし $x \mapsto \|x\|^2$ は可逆ではない）。37.6.2.2 節のデータ処理不等式における等号成立条件のうち、どれが成り立っているか。
2. いかなる線形分類器 $\hat{y} = \mathbf{1}(\mathbf{w}^\top \mathbf{x} + b > 0)$ もすべての入力を正しく分類できない一方で、Z に対する閾値処理はすべて正しく分類できることを示せ。
3. ネットワークの各層はマルコフ連鎖 $X \to Z_1 \to Z_2 \to \cdots$ をなす。Z_ℓ を読み取るあらゆる種類の分類器の中で最良のものの誤り率は、ℓ が大きくなっても低下し得ないことを示せ。それにもかかわらず、Z_ℓ に当てはめた線形分類器の正解率は ℓ とともに上昇することが多い。なぜここに矛盾はないのか。
4. Xu et al. (2020) は、予測器族 $\mathcal{V}$（予測器 $q(y \mid x)$ の集合）が抽出できる情報量を測っている。$H_{\mathcal{V}}(Y \mid X) = \inf_{q \in \mathcal{V}} \mathbb{E}[-\log q(Y \mid X)]$ とおき、$H_{\mathcal{V}}(Y)$ を入力を無視する $\mathcal{V}$ 内の予測器にわたる同じ下限とする。**利用可能情報量**は次で定義される。

$$I_{\mathcal{V}}(X \to Y) = H_{\mathcal{V}}(Y) - H_{\mathcal{V}}(Y \mid X).$$

   $\mathcal{V}$ を与えられた入力上のロジスティック回帰とする。$I_{\mathcal{V}}(X \to Y) = 0$ かつ $I_{\mathcal{V}}(Z \to Y) = \ln 2$ を示せ。X については、期待ロジスティック損失が $(\mathbf{w}, b)$ に関して凸であり、$\mathbf{w}$ を回転させても不変であることを用いよ。37.6.2.2 節のどの不等式に $I_{\mathcal{V}}$ は反するか。

3. **シナジーとベン図**。X と Y を独立な公平なビットとし、$Z = X \oplus Y$ をその排他的論理和とする。本節ではこれについて $I(X;Y) = 0$ および $I(X;Y \mid Z) = \ln 2$ を計算した。
   1. マルコフ連鎖 $X \to Z \to Y$、$X \to Y \to Z$、$Y \to X \to Z$ のうち成り立つものはどれか。したがってデータ処理不等式はこの 3 つ組について何を主張するか。
   2. Figure 37.13 の 3 変数版では、3 つの円盤すべてに共通する領域が**交互作用情報量** $I(X;Y;Z) = I(X;Y) - I(X;Y \mid Z)$ に対応するはずである。$I(X;Y;Z) = H(X) + H(Y) + H(Z) - H(X,Y) - H(X,Z) - H(Y,Z) + H(X,Y,Z)$ を証明し、$I(X;Y;Z)$ が 3 つの引数に関して対称であると結論せよ。
   3. $I(X;Y;Z)$ を (a) の 3 つ組と、1 つの公平なビットの 3 つのコピー $X = Y = Z$ のそれぞれに対して求めよ。これら 2 つの値は、エントロピーの 3 円盤図に対して何を意味するか。

   Thomas M. Cover、Joy A. Thomas 著 Elements of Information Theory, 2nd ed. 演習問題 2.25、および David J. C. MacKay 著 Information Theory, Inference, and Learning Algorithms 演習問題 8.8 に基づく。

4. [code] **f ダイバージェンスからのデータ処理不等式**。入力をランダムな出力へ写す任意の通信路 κ と、入力の任意の 2 つの分布 P および Q に対し、出力分布は $D_f(P\kappa \,\|\, Q\kappa) \le D_f(P \,\|\, Q)$ をすべての f ダイバージェンスについて満たす（37.3.3.3 節）。KL ダイバージェンスと $\chi^2(P\|Q) = \sum_x p(x)^2/q(x) - 1$ はその 2 つの例である。
   1. $X \to Y \to Z$ をマルコフ連鎖とする。$P = P_{X,Y}$、$Q = P_X \otimes P_Y$ とし、通信路 κ を (x,y) から (x,z)（ただし $z \sim p(z \mid y)$）への写像とする。$P\kappa = P_{X,Z}$ かつ $Q\kappa = P_X \otimes P_Z$ を示し、$I(X;Z) \le I(X;Y)$ を結論せよ。
   2. 本節のデータ処理不等式の命題のうち、この証明経路からは得られない部分はどれか。
   3. 確率 0.1 と 0.2 の直列の 2 回反転について、2 回目の反転の前後それぞれで、同時確率表とその周辺分布の積との間の KL ダイバージェンスおよび χ^2 ダイバージェンスを計算せよ。

4. 公平な入力を確率 δ の反転通信路に通したとき $\chi^2(P_{X,Y} \,\|\, P_X \otimes P_Y) = (1-2\delta)^2$ となることを示し、(c) の数値を確認せよ。さらに確率 δ' で反転させると、このダイバージェンスに $(1-2\delta')^2$（ここでは 0.36）が掛かると結論せよ。(c) における比 $I(X;Z)/I(X;Y)$ も同じ倍率になるか。

5. **ファノの不等式とその逆。**●ファノの不等式 Equation 37.59 は、推定値 $\hat{Y} = g(X)$（ラベル Y が k 個の値をとるときの任意の推定値）の誤り率を上からではなく下から抑える。

   1. $\hat{Y}(x) = \arg\max_y p(y \mid x)$ を最尤ラベルとする。その誤り率が $P_e \le 1 - e^{-H(Y|X)}$ を満たすことを示せ。まず $\max_y p(y \mid x)$ を下から $\sum_y p(y \mid x)^2 = \mathbb{E}[p(Y \mid X) \mid X = x]$ で抑え、次にイェンセンの不等式を $\mathbb{E}[p(Y \mid X)] = \mathbb{E}\big[e^{\log p(Y|X)}\big]$ に 1 回適用せよ。
   2. 観測 X が $I(X;Y) = 9$ nat の情報を、均衡した 10,000 クラスのラベル Y について保持しているとする。Equation 37.59 と (a) を用いて、最良の分類器の誤り率を上下から挟み込め。不等式 Equation 37.59 は P_e について閉形式解をもたないため、数値的に解くこと。この区間の幅は、相互情報量から下流タスクの誤り率を予測することについて何を示唆しているか。

6. **損失による保証とファノの不等式。**均衡した $k = 10$ クラスの分類器がテスト誤差 $P_e = 0.08$ およびテスト損失 $L = 0.30$ nat をもつとする。

   1. $I(X;Y)$ について、ファノの不等式 Equation 37.59 は何 nat を保証し、損失による保証 Equation 37.60 は何 nat を保証するか。
   2. 任意の推定値 $\hat{Y} = g(X)$（誤り確率 P_e）に対し、モデル $q(y \mid x)$ が確率 $1 - P_e$ を $y = g(x)$ に置き、残りの P_e を他の $k-1$ 個のラベルに均等に配分するとする。その交差エントロピー損失が $H_b(P_e) + P_e \log(k-1)$ となることを示し、ファノの不等式はこのモデルの損失による保証であると結論せよ。これは 37.6.2.4 節で評価した 4 つのモデルのうちどれに当たるか。
   3. (a) の分類器は、その判定結果から (b) が構成するモデルよりも小さい損失を報告している。それが起こるためには、分類器の確信度はどのように振る舞わなければならないか。2 値ラベルをもつ対 (X, Y) と誤り率 P_e をもつモデルであって、ファノの不等式より真に多くの情報量を保証する具体例を挙げよ。

7. **2 元通信路。**●2 値エントロピー H_b を用い、nat 単位で計算せよ。

   1. **Z 通信路**は 0 をそのまま通過させ、1 を確率 $\tfrac{1}{2}$ で 0 に変える。$I(X;Y) = H(Y) - H(Y \mid X)$ を $a = P(X=1)$ の関数として計算し、$a = \tfrac{2}{5}$ で最大となってその最大値が $\ln \tfrac{5}{4}$ nat に等しいことを示せ。最良の入力が信頼できない記号を送る割合を半分未満にするのはなぜか。また、まったく送らないわけではないのはなぜか。
   2. 特徴量が値 ± 1 をとり、率 α のドロップアウト（確率 α で 0 に置き換え、それ以外は定数倍する処理）を通過するとする。この **2 元消失通信路**の容量が $(1-\alpha)\ln 2$ nat であることを示せ。
   3. 2 元対称通信路の容量が $\alpha = 0.5$ のドロップアウトと同じになる反転確率 δ を求めよ。ビットの反転が消去よりも大きく容量を損なうのはなぜか。

8. **ガウス通信路。**$Y = X + N$ とし、ノイズ $N \sim \mathcal{N}(0, \sigma_N^2)$ は X と独立であり、入力は電力制約 $\mathbb{E}[X^2] \le P$ に従うとする。

   1. 確率密度をもつ任意の入力に対して $I(X;Y) = h(Y) - h(N)$ が成り立つことを示せ。
   2. 分散 s^2 をもつ密度の中で微分エントロピーが最大になるのはガウス分布であり、その値は $\tfrac{1}{2}\log(2\pi e s^2)$ である。これを用いて $h(Y) \le \tfrac{1}{2}\log\big(2\pi e(P + \sigma_N^2)\big)$ を示し、

容量が $C = \frac{1}{2}\log(1 + P/\sigma_N^2)$ であってガウス入力により達成されると結論せよ。

3. 信号対雑音比 $P/\sigma_N^2 = 1, 10, 100$ における C を nat 単位およびビット単位で求めよ。また高い信号対雑音比においては、P を 2 倍にするごとに約 $\frac{1}{2}\ln 2$ nat（半ビット）が加わることを示せ。
4. 拡散モデルは、$\mathbb{E}[X^2] = 1$ となるようスケールされたデータの座標 X を $\sqrt{a}\,X + \sqrt{1-a}\,\epsilon$（ただし $0 < a < 1$、また $\epsilon \sim \mathcal{N}(0,1)$ は独立）に置き換える。ノイズを加えられた座標が保持する情報は高々 $\frac{1}{2}\log\frac{1}{1-a}$ nat（X についての情報量）であることを X の分布によらず示し、この上界を $a = 0.5$ と $a = 0.01$ で評価せよ。この上界が達成されるのはどのようなデータに対してか。

9. **変分オートエンコーダのレート上界。** $X \sim \mathcal{N}(0,1)$ とし、エンコーダを 37.6.4.2 節と同じく $Z = X + \sigma\epsilon$（したがって $p(z \mid x) = \mathcal{N}(z; x, \sigma^2)$）とし、事前分布を $r = \mathcal{N}(0,1)$ とする。$D_{\mathrm{KL}}(\mathcal{N}(\mu, s^2) \,\|\, \mathcal{N}(0,1)) = \frac{1}{2}(\mu^2 + s^2 - 1 - \log s^2)$ を用いてよい。

   1. Equation 37.70 の 3 つの項を計算し、恒等式を確かめよ。$\sigma = 1$ において、KL ペナルティはレートを何 nat 過大評価しているか。
   2. 符号を $Z' = Z/\sqrt{1+\sigma^2}$ へと定数倍し直せ。$I(X; Z') = I(X; Z)$ が成り立ち、かつ上界が等号で成り立つようになることを示せ。
   3. レートは符号の単位に依存しないが、この上界は単位に依存する。与えられたエンコーダに対して上界を等号にする事前分布 r はどれか。したがって KL ペナルティを最小化することは、レートを下げるほかに符号に対してどのような作用を及ぼすか。

37.7 相互情報量の推定

これまでに計算してきた相互情報量（mutual information）は、いずれも閉じた形の同時密度（joint density）から求められたものであった。しかし実際には、データに確率密度が添えられていることは決してない。その代わりに与えられるのは、発話音声とその書き起こし、あるいは検索クエリとそれに答えた文書といったペアの集まりであり、そうしたサンプルだけから、ペアの一方が他方についてどれだけの情報を明らかにするかを定量化したいのである。表現学習（representation learning）はこの考え方をさらに一歩推し進め、共有される情報量が大きくなるようにモデルを訓練することで、推定値そのものを訓練の目的関数へと変える。

よく知られた対照損失（contrastive loss）を例にとると、この課題が具体的になる。SimCLR や CLIP などのシステムは、1 枚の写真から切り出した 2 つのランダムクロップや、画像とそのキャプションの組のように、同一データ例のペアになった**ビュー**を用いて訓練する。一方のビューが与えられると、モデルは InfoNCE 損失として知られる交差エントロピー損失（cross-entropy loss）を最小化することで、バッチ内の候補（たとえば 128 個）の中から真の対応相手を識別するよう学習する。論文ではしばしば、この損失を $\ln 128 = 4.85$ から差し引き、その差を 2 つのビュー間の相互情報量として報告する。しかし、2 つのビューが実際にどれほど多くの情報を共有していようとも、その差が 4.85 nat を超えることは決してない。毎回正しい相手を選び出すモデルであっても、128 個の候補を識別できることを証明したにすぎないからである。このような量を用いてモデルを訓練すると同時にそれを測定値としても解釈する以上、真の測定値と下界（lower bound）とを区別し、さらに下界の緩み（slack）と学習されたモデルの誤差とを切り分ける必要がある。

サンプルから相互情報量を推定する方法は大きく 2 つある。直接的なアプローチは、ヒストグラムのビン内または最近傍（nearest neighbor）領域内に入るサンプル数を数える手法であり、低次元変数間の小さな情報量に対してはうまく機能するが、ベクトルに対しては破綻する。第 2 のアプローチは、未知の密度を学習されたモデルに置き換え、得られる期待値が $I(X; Y)$ の下界となるように構成したうえで、その下界を最大化する。本節の残りの議論を支える柱は 2 つの結果である。すなわち、分布に関する仮定なしに任意の推定量が保証できる限界という根本的な制約

（37.7.2.4 節）と、InfoNCE 損失の背後にある変分下界およびその対数的な上限（37.7.4 節）である。

```
from scipy.spatial.distance import cdist
from scipy.special import digamma, logsumexp
```

37.7.1 既知の情報量を持つペア

推定量を公平に評価するには、厳密な相互情報量があらかじめわかっているベンチマークが必要である。本節を通じて、37.6.1.3 節で導入した相関（correlation）ρ を持つ標準 2 変量ガウス分布のペアを用いる。このペアが共有する情報量は次の式で与えられる。

$$I(X;Y) = -\tfrac{1}{2}\log\left(1-\rho^2\right) \tag{37.71}$$

単位は nat であり、その値は 0.83 nat（$\rho = 0.9$ のとき）および 1.96 nat（$\rho = 0.99$ のとき）となる。なお、本節の対数はすべて自然対数である。

これから導出する変分下界は、37.6.1.4 節の**自己相互情報量**（pointwise mutual information）、すなわち $\operatorname{pmi}(x,y) = \log \frac{p(x,y)}{p(x)\,p(y)}$ に依拠している。これは、特定のペア (x,y) が独立性（independence）のもとと比べて同時分布（joint distribution）のもとでどれだけ起こりやすいかを測る量である。標準ガウス分布のペアに対して、pmi は次のような単純な二次関数（quadratic function）になる。

$$\operatorname{pmi}(x,y) = -\tfrac{1}{2}\log(1-\rho^2) - \frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)} + \frac{x^2+y^2}{2}. \tag{37.72}$$

この定義から直ちに 2 つの恒等式が導かれる。第 1 に、同時分布からサンプリングされたペアにわたって $\operatorname{pmi}(x,y)$ の平均をとると $I(X;Y)$ になる。第 2 に、その指数をとった $e^{\operatorname{pmi}(x,y)}$ を x と y の**独立な**サンプルにわたって平均すると 1 になる。これは $p(x)\,p(y)\,e^{\operatorname{pmi}(x,y)} = p(x,y)$ を積分すると 1 になるからである。両方の恒等式をサンプル上で確かめてみよう。第 2 の恒等式については、配列内で各 x を次のサンプルの y と組にすることで従属性を断ち切る。これらを**非対応ペア**（mismatched pairs）と呼ぶことにする。各非対応ペアは $P_X \otimes P_Y$ からの独立なサンプルであり、下界の計算に独立なペアが必要になるたびにこの単純な手法を再利用する。

```
def gauss_pair(rng, rho, n):
    """Samples of the standard Gaussian pair; n is a count or a shape."""
    x = rng.standard_normal(n)
    y = rho * x + np.sqrt(1 - rho**2) * rng.standard_normal(n)
    return x, y

def mi_gauss(rho):
    """I(X;Y) of the pair, in nats."""
    return -0.5 * np.log(1 - rho**2)

def pmi_gauss(x, y, rho):
    """Pointwise mutual information pmi(x, y) of the pair, in nats."""
    c = 1 - rho**2
    return (-0.5 * np.log(c) - (x**2 - 2 * rho * x * y + y**2) / (2 * c)
            + (x**2 + y**2) / 2)
```

```
rng = np.random.default_rng()
x, y = gauss_pair(rng, 0.9, 200_000)
print(f'I(X;Y) = {mi_gauss(0.9):.3f} nats, mean pmi over pairs = '
      f'{pmi_gauss(x, y, 0.9).mean():.3f}')
print(f'mean exp(pmi) over independent pairs = '
      f'{np.exp(pmi_gauss(x, np.roll(y, 1), 0.9)).mean():.3f}')
```

```
I(X;Y) = 0.830 nats, mean pmi over pairs = 0.833
mean exp(pmi) over independent pairs = 0.994
```

どちらの標本平均も、理論値と約 0.01 以内の精度で一致している。本節のすべての変分下界はこれら 2 つの恒等式の上に築かれており、統計的な困難のほぼすべてを引き起こすのは第 2 の恒等式の方である。$\rho = 0.9$ では e^{pmi} の振る舞いは穏やかであり、独立なペアにわたるその分散（variance）は $e^{2I} - 1$（演習問題 1）であって、ここではわずか 4.3 にすぎない。ところが $I = 6$ になると、その分散は $160{,}000$ へと爆発し、同じ $200{,}000$ サンプルの平均であっても約 0.9 もの不確実性が残る。

37.7.2 ノンパラメトリック推定

データから確率分布を推定する最も直接的な方法は、空間の各領域に何個のサンプルが入るかを数えることである。設計上の重要な選択は、それらの領域をどう定義するかにある。まずは最も単純な選択である固定グリッドから始めよう。

37.7.2.1 ヒストグラム

ヒストグラム推定量は、各変数の値域を b 個の等幅区間に分割し、$b \times b$ 個の各セルに入るサンプル数を数えて、得られた経験頻度を離散版の相互情報量の公式に代入する。この推定量は単純であるにもかかわらず、互いに逆方向に働く 2 つの競合する統計的バイアス（estimator bias）を抱えている。

第 1 のバイアスは、推定値を**下方向**に引き下げる。連続値をそのビンのインデックスに置き換えることは X に決定論的な関数を適用することにほかならず、データ処理不等式（data processing inequality、37.6.2.2 節）により、X の任意の関数が Y と共有する情報量は高々 X 自身が共有する情報量までにとどまる。したがってビン分割は情報を失わせることしかなく、従属性の幅が 1 つのビンより狭いときにはその損失が深刻になる。たとえば $\rho = 0.99$ では、y の x が与えられたもとでの条件付き標準偏差（standard deviation）は $\sqrt{1 - \rho^2} = 0.14$ である。おおよそ $[-4.5, 4.5]$ に広がるサンプルを 10 個のビンに分割するとセルの幅は 0.9 となり、条件付きの広がりの 6 倍以上になってしまう。

第 2 のバイアスは、推定値を**上方向**に押し上げる。プラグイン推定値は、経験同時度数表と、その行・列の周辺分布の積との間の KL ダイバージェンスである。そのため常に非負であり、表が厳密に積へ分解される場合に限り 0 になる。たとえ X と Y が完全に独立であっても、ランダムなサンプリングの揺らぎによって厳密な分解は妨げられるため、ノイズが従属性を装ってしまう。各セルに含まれるサンプル数が少ないほど、この上向きのバイアスは悪化する。

これら 2 つの誤差に加えて、ヒストグラムには厳密な上限がある。推定値はビン分割された X のエントロピーを超えることができず、b 個のビンが担える情報量は高々 $\log b$ nat である。ビン数が 10 個ならわずか 2.30 nat であり、X と Y がどれほど強く結びついていてもこれを超えることはない。本節で扱うすべての推定量において、この上限の変種に直面することになる。いかなる手法も、識別可能な状態（state）の数の対数を超える情報量を報告することはできないのである。

この推定量を `binned_mi` として実装し、4 つの分布から生成した 200,000 個のサンプルに対して $b = 10$、40、160 を用いてテストしてみよう。対象とする 4 つの分布は、独立なペア、$\rho = 0.9$ および $\rho = 0.99$ のガウス分布のペア、そして放物線ペア $Y = X^2 + 0.3\,\epsilon$（37.6.1.5 節）である。この放物線ペアは線形相関が 0 でありながら 1.141 nat を共有する。

```
def binned_mi(x, y, bins):
    """Plug-in MI of the bins x bins histogram of (x, y), in nats."""
    p = np.histogram2d(x, y, bins=bins)[0] / len(x)       # cell frequencies
    q = p.sum(axis=1, keepdims=True) * p.sum(axis=0, keepdims=True)
    nz = p > 0                                            # occupied cells
    return float((p[nz] * np.log(p[nz] / q[nz])).sum())

mi_parabola = 1.141  # by quadrature, sec_mdl-mi-nonlinear-correlation
n = 200_000
x = rng.standard_normal(n)
pairs = {'rho = 0': (gauss_pair(rng, 0.0, n), 0.0),
         'rho = 0.9': (gauss_pair(rng, 0.9, n), mi_gauss(0.9)),
         'rho = 0.99': (gauss_pair(rng, 0.99, n), mi_gauss(0.99)),
         'parabola': ((x, x**2 + 0.3 * rng.standard_normal(n)), mi_parabola)}
print('pair        truth  b = 10  b = 40  b = 160')
for name, ((u, v), truth) in pairs.items():
    print(f'{name:10}  {truth:5.3f}'
          + ''.join(f'  {binned_mi(u, v, b):6.3f}' for b in (10, 40, 160)))
```

```
pair        truth  b = 10  b = 40  b = 160
rho = 0     0.000   0.000   0.003   0.035
rho = 0.9   0.830   0.629   0.813   0.845
rho = 0.99  1.959   1.090   1.800   1.955
parabola    1.141   0.551   0.989   1.136
```

表には両方のバイアスがはっきりと現れている。$b = 10$ では、従属なすべてのペアで値が低く出ており、約 0.6 nat（真の値は 0.83 nat、$\rho = 0.9$ のとき）にとどまり、$\rho = 0.99$ と放物線ペアでは真の値の約半分にしかならない。$b = 160$ では、独立なペアに 100 分の数 nat ほどの値が計上され、$\rho = 0.9$ のペアでは実際に真の値を上回ってしまっている。これは粗いビン分割だけでは決して起こり得ない現象である。$\rho = 0.99$ かつ $b = 160$ では推定値が真の値の約 0.01 以内に収まっているが、これは逆向きの誤差がたまたま打ち消し合った結果にすぎず、そのすぐ上の行では同じグリッドが過大推定を起こしている。真の値が未知のとき、どの列を信頼すべきかをこの表から知る術はない。

3 つ目の弱点は座標への敏感さである。どちらかの変数に可逆変換を施しても相互情報量は変化しないが（37.6.1.5 節）、固定幅のグリッドは歪んでしまう。x を x^3 に置き換える操作を、$\rho = 0.9$ のガウス分布のペアに施したとしよう。標準正規分布からの 200,000 個のサンプルのうち最大値はおおよそ ± 4.5 であるため、x^3 の範囲はおおよそ ± 90 に広がる。すると 40 個の等幅ビンの幅はそれぞれ約 4.5 となり、サンプルの約 10 個中 9 個が 0 付近の 2〜3 個のビンに詰め込まれてしまう。明快な解決策は、ソートしたサンプル内での各値の位置を表す**順位**（rank）をビン分割することである。任意の狭義単調増加写像は順位を保存し、どの順位ビンにも自動的に同じ数のサンプルが入る。

```
x, y = gauss_pair(rng, 0.9, n)
rank = lambda a: np.argsort(np.argsort(a))       # position in sorted order
print(f'true I = {mi_gauss(0.9):.3f} nats; histograms with b = 40:')
```

```
for name, u in (('x', x), ('x**3', x**3)):
    print(f'{name:4}: correlation {np.corrcoef(u, y)[0, 1]:.3f}, '
          f'raw {binned_mi(u, y, 40):.3f}, '
          f'ranks {binned_mi(rank(u), rank(y), 40):.3f} nats')
```

```
true I = 0.830 nats; histograms with b = 40:
x   : correlation 0.901, raw 0.819, ranks 0.819 nats
x**3: correlation 0.699, raw 0.439, ranks 0.819 nats
```

真の相互情報量は 0.830 nat のままであるにもかかわらず、x を 3 乗すると線形相関は 0.90 から約 0.7 へと低下し、生の値によるヒストグラム推定値は 0.3〜0.5 nat の間へと激減する（正確な値はビンの範囲を決める少数の極端なサンプルに依存する）。これに対し、順位をビン分割すれば、x と x^3 で桁の隅々まで完全に同一の推定値が得られ、真の値との差も 0.02 nat 以内に収まる。

順位を用いれば座標の問題は解決するが、高次元におけるヒストグラムの破綻はどうにも救いようがない。5 次元ベクトルのペアには 10 次元のグリッドが必要になり、各軸にわずか 10 個のビンを置くだけでも 10^{10} 個のセルが生じ、現実的なあらゆるサンプルサイズにおいてそのほとんどすべてが空になってしまう。ベクトルを扱うには、データ自身に解像度を決めさせ、サンプルが密な場所では小さな領域を、疎な場所では大きな領域を使うような推定量が必要である。

37.7.2.2 KSG 推定量

KSG として知られる Kraskov et al. (2004) の推定量は、固定されたヒストグラムのセルを、各データ点の周囲の適応的な最近傍領域に置き換える。これは低次元変数のペアに対する標準的なノンパラメトリック手法である（本項と次項は初読時には読み飛ばしてもよい）。

KSG では座標差の最大値（チェビシェフ距離、すなわち ℓ_∞ 距離）を用いて距離を測るため、半径 r の球は 2 次元では正方形、より高次元では超立方体になる。各サンプル点 i の周囲に、同時空間 (x, y) 内で m 番目の最近傍点に達する最小のボックスを描き、そのボックスの半幅を r_i とする。構成法から、このボックスには割合 m/n（全 n サンプル中）のサンプルが含まれ、(x_i, y_i) 近傍の同時確率（joint probability）質量の局所的な推定値を与える。

次に、まったく同じスケール r_i で 2 つの周辺確率を測る。$n_x^{(i)}$ を縦方向のストリップ（帯状領域）$|x - x_i| < r_i$ の厳密な内部に入る他のサンプルの数とし、$n_y^{(i)}$ を横方向のストリップ $|y - y_i| < r_i$ の厳密な内部に入る他のサンプルの数とする。半幅 r_i を共有しているからこそ、生のカウント数を確率密度の代わりに使えるのである。もしボックス全体で密度がほぼ一定であれば、ボックスが持つ確率質量は約 $p(x_i, y_i)\,(2r_i)^2$ となり、一方で 2 つのストリップが持つ確率質量はそれぞれ約 $p(x_i)\,2r_i$ と $p(y_i)\,2r_i$ となる。ボックス内の割合を 2 つのストリップ内の割合の積で割ると、幾何学的な因子 $2r_i$ が約分されて消える。その対数をとれば、純粋にカウントだけで計算された自己相互情報量の局所推定値が得られる。Figure 37.18 に、1 つの点におけるこれら 3 つのカウントを示す。

これら局所的な対数比を全 n 点にわたって平均すると、バイアス補正（bias correction）にあたる 1 つの有限標本補正を伴って $I(X;Y)$ の推定値が得られる。小さなカウントの対数をとると対数確率（log probability）の推定値にバイアスが生じる。すなわち、m 番目の近傍点（全 $n-1$ 個の近傍点のうち）に達するボックスについて、対数確率質量の期待値は $\mathbb{E}[\log p] = \psi(m) - \psi(n)$ となり、$\log(m/n)$ とは一致しない。ここで $\psi = (\log\Gamma)'$ はディガンマ関数である（導出は省略する）。対数を ψ に置き換えることで、KSG 推定量が次のように得られる。

$$\hat{I}_{\text{KSG}} = \psi(m) + \psi(n) - \frac{1}{n}\sum_{i=1}^{n}\left(\psi\left(n_x^{(i)} + 1\right) + \psi\left(n_y^{(i)} + 1\right)\right). \tag{37.73}$$

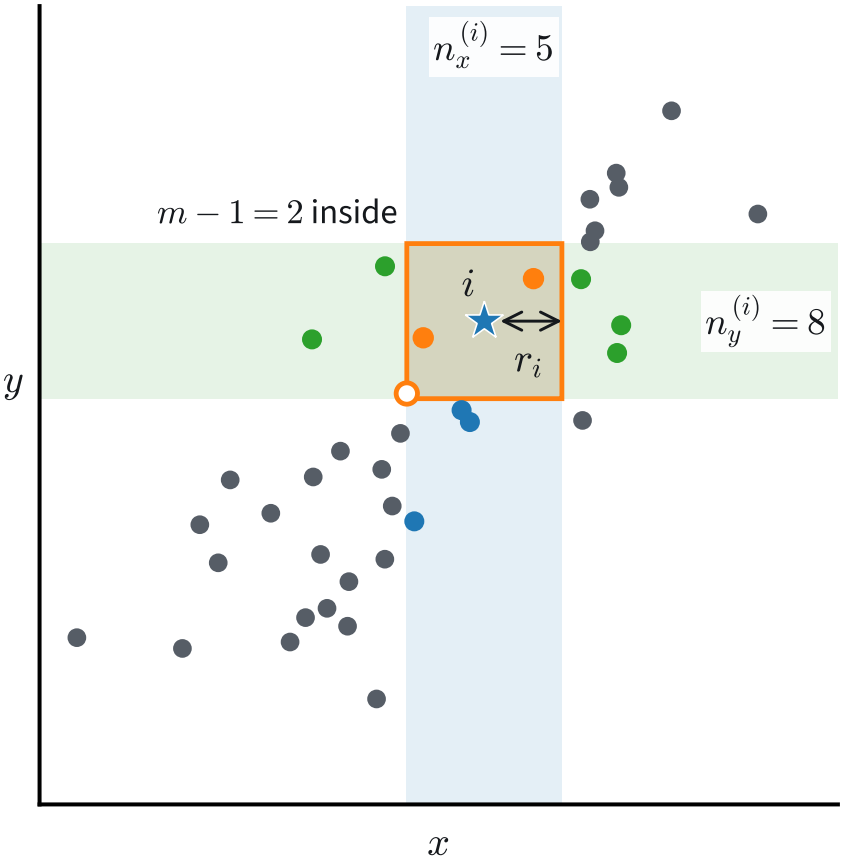


Figure 37.18: サンプル点 i（星印）における $m = 3$ の KSG カウント。半幅 r_i のボックスは同時平面内で 3 番目の最近傍点（左辺上の丸印。横ストリップ内にあるため $n_y^{(i)}$ にのみカウントされる）に達しており、2 つの近傍点が厳密に内部に入っている。同じ半幅の縦ストリップには $n_x^{(i)}$ 個、横ストリップには $n_y^{(i)}$ 個の他のサンプルが入る（青と緑の点は一方のストリップにのみ含まれる）。推定量の各被加数は近似的に $\log \frac{m/n}{(n_x/n)(n_y/n)}$ であり、カウントから計算された自己相互情報量となっている。

$\psi(j) \approx \log j$（j が大きいとき）が成り立つため、各被加数は近似的に $\log \frac{m\,n}{n_x^{(i)}\,n_y^{(i)}}$ となり、ボックス内の割合とストリップ内の割合の積との局所的な対数比に一致する。ストリップの項の中にある $+1$ はボックスと基準を揃えるためのものである。つまり、$\psi(m)$ は境界が m 番目の近傍点を通るボックスに対応するため、厳密な内部に $n_x^{(i)}$ 個の点を含むストリップは、あたかもその境界が $(n_x^{(i)}+1)$ 番目の点を通るかのように扱われる。両方のストリップのカウントには狭義の不等号を用い、点 i 自身を除外するため、ストリップの境界上にちょうど乗っている近傍点はカウントされない。すべての半径 r_i が局所的なサンプル密度に自動的に適応するため、調整すべきビン幅やカーネルのバンド幅は存在しない。

$n \times n$ のチェビシェフ距離行列を x と y のそれぞれについて `cdist` で求め、同時距離として要素ごとの最大値をとれば、すべての近傍カウントを一度に計算できる。

```
def ksg_counts(x, y, m=5):
    """Strip counts n_x, n_y of every sample point; x, y are (n, d)."""
    dx, dy = cdist(x, x, 'chebyshev'), cdist(y, y, 'chebyshev')
    d = np.maximum(dx, dy)                       # joint Chebyshev distance
    np.fill_diagonal(d, np.inf)
    r = np.partition(d, m - 1, axis=1)[:, m - 1]  # half-width: m-th neighbor
    n_x = (dx < r[:, None]).sum(axis=1) - 1      # strict, self excluded
    n_y = (dy < r[:, None]).sum(axis=1) - 1
    return n_x, n_y

def ksg_mi(x, y, m=5):
    """KSG nearest-neighbor estimate of I(X;Y) in nats; x, y are (n, d)."""
    n_x, n_y = ksg_counts(x, y, m)
    return float(digamma(m) + digamma(len(x))
                 - (digamma(n_x + 1) + digamma(n_y + 1)).mean())
```

Equation 37.73 の各被加数は近傍カウントのみに依存するため、カウントを用いること自体が KSG の報告し得る値に厳密な上限を課す。

命題（KSG のサンプルサイズによる上限）同一の点からの同時距離がどの 2 点についても等しくない任意の n 点のサンプルに対して、次が成り立つ。

$$\hat{I}_{\mathrm{KSG}} \le \psi(n) - \psi(m), \tag{37.74}$$

等号は $Y = X$ のときに成り立つ。

証明距離の同値（タイ）がないため、ちょうど $m-1$ 個のサンプルが点 i に対してその m 番目の近傍点より厳密に近い位置にある。同時チェビシェフ距離は 2 つの座標距離のうち大きい方であるため、これら $m-1$ 個の各点は $|x_j - x_i| < r_i$ と $|y_j - y_i| < r_i$ の両方を満たし、したがって両方のストリップに含まれる（すなわち $n_x^{(i)} \ge m-1$ かつ $n_y^{(i)} \ge m-1$）。ψ は狭義単調増加であるため、各被加数は高々 $\psi(m) + \psi(n) - 2\psi(m) = \psi(n) - \psi(m)$ である。$Y = X$ のときは同時距離と周辺距離が一致するため、両ストリップにはボックスの厳密な内部にある $m-1$ 個の点のみが含まれ、すべての被加数がこの上界を達成する。■

$n = 2{,}000$、$m = 5$ のとき、この上限は $\psi(2000) - \psi(5) = 6.09$ nat であり、$\log(n/m) = 5.99$ に近い。$Y = X$ かつ X が連続で真の相互情報量が無限大になる場合であっても、KSG が報告する値は 6.1 nat であり、サンプルサイズを 2 倍にしてもその上限は約 $\log 2$ しか上がらない。ここでも再び、ヒストグラムの $\log b$ という上限が、ビンを適応的な近傍に置き換えた形で現れて

いる。近傍をカウントする推定量が報告できるのは、サンプルから作れる互いに異なる近傍の数（おおよそ n/m）の対数が上限なのである。

37.7.2.3 高次元における KSG

残念ながら高次元では、幾何学的な理由から、KSG はサンプルサイズによる上限よりはるかに低いところで破綻する。ガウス分布のペアの d 個の独立なコピー（それぞれ $\rho = 0.9$）を考え、X と Y を d 次元ベクトルとする。対数密度（log density）比が d 個の独立な項の和に分解されるため、相互情報量は独立な座標にわたって加法的であり、真の値は $d \times 0.83$ nat となる。

KSG は同時密度と周辺密度が各近傍ボックス内でほぼ一定であることに依拠しているが、この仮定は次元が増えるにつれて崩壊する。$n = 2{,}000$、$m = 5$ のとき、各ボックスはサンプルの割合 $m/n = 0.0025$ を捕捉する。もし点が $2d$ 次元空間に一様に広がっていれば、そのようなボックスはどの軸に沿ってもデータ幅の割合 $(m/n)^{1/(2d)}$ にわたって広がることになり、0.05（$d = 1$ のとき）から、0.47（$d = 4$）、さらには 0.83（$d = 16$）へと跳ね上がる。ところが実際の従属性は $y = \rho x$ の周囲の細い帯の内部に集中している。x が与えられたとき、y の各座標の標準偏差は $\sqrt{1-\rho^2} = 0.44$ であり、約 0.07 倍（サンプル全体の広がりである約 -3.3〜3.3 に対する比率）にすぎない。その帯より何倍も広いボックスは、粗すぎるヒストグラムのビンとまったく同様に、従属性を塗りつぶしてしまう。2,000 個のサンプルに対して $d = 1, 2, 4, 8, 16$ で KSG を実行し、推定値とストリップ内の平均カウント数を、上限を達成する $Y = X$ の場合とあわせて記録してみよう。

```
n, rho = 2000, 0.9
print(' d  true I    KSG  mean strip count')
for dim in (1, 2, 4, 8, 16):
    x, y = gauss_pair(rng, rho, (n, dim))        # dim independent pairs
    n_x, n_y = ksg_counts(x, y)
    print(f'{dim:2d}  {dim * mi_gauss(rho):6.2f}  {ksg_mi(x, y):5.2f}  '
          f'{(n_x.mean() + n_y.mean()) / 2:16.1f}')
x = rng.standard_normal((n, 1))
print(f'Y = X:  KSG = {ksg_mi(x, x):.2f}, '
      f'ceiling psi(n) - psi(m) = {digamma(n) - digamma(5):.2f} nats')
```

```
 d  true I    KSG  mean strip count
 1    0.83   0.83              71.3
 2    1.66   1.60              50.0
 4    3.32   2.51              31.3
 8    6.64   2.85              26.0
16   13.29   2.75              26.8
Y = X:  KSG = 6.09, ceiling psi(n) - psi(m) = 6.09 nats
```

1 次元および 2 次元では、KSG は真の値から約 0.1 nat 以内に収まっている。しかし $d = 4$ になると、約 2.5 nat（真の値は 3.3 nat）へと落ち込み、1 nat 近く低くなる。その後は頭打ちになり、$d = 8$ でも $d = 16$ でも、KSG が報告する値は 3 nat 弱にとどまる（その間に真の相互情報量は 6.6 から 13.3 へと倍増している）。ここで推定を阻んでいるのはサンプルサイズによる上限ではないことに注目したい。3 nat という値は、出力された 6.1 nat という上限（$Y = X$ のとき）の半分未満だからである。

ストリップ内の平均カウント数を見れば、なぜ推定値が頭打ちになるかがわかる。各被加数は近似的に $\log \frac{m\,n}{n_x\,n_y}$ であるため、推定値が上昇できるのは周辺ストリップ内の点が減っていく場合に限られる。ところがそのカウント数は約 70 点（$d = 1$）から約 31 点（$d = 4$）へと減った後、26 点付近で横ばいになる。$d = 16$ において $d = 1$ のときと同じくらい局所的なボックスを保つに

は $m/n = 0.05^{32}$ が必要となり、およそ 10^{42} 点ものサンプルが求められる計算になる。局所的な近傍を維持するために必要なサンプルサイズは、次元に対して指数関数的に増大するのである。

37.7.2.4 分布によらない推定の限界

もっと巧妙な推定量を使えば、このボトルネックを回避できるかもしれない。その望みを検証するため、厳密な測定よりも弱い条件である**下界**を求めてみよう。具体的には、関数 B（n 個のサンプルの関数）であって、$B \le I(X;Y)$ が確率 0.95 以上でペア (X,Y) の**あらゆる分布に対して**成り立つものを求める（$1/2$ より大きい任意の信頼水準（confidence level）でもまったく同様に成り立つ）。2 つのビューが多量の情報を共有していることに確信を持ちたい表現学習にとっても、求められているのはまさに下界である。

考え得る最も単純なケースで、すでに根本的な障壁が明らかになる。$Y = X$ とおき、X を離散変数とすれば、$I(X;Y) = H(X)$ となる。推定量が離散記号を固有の距離や順序を持たない任意のラベルとして扱うなら、**どの**記号が現れたかを利用することはできない。調べられるのは重複のパターン、すなわち 1 回だけ現れた異なる値がいくつあり、2 回現れたものがいくつあるか、といった情報だけである。この要約統計量はサンプルの**タイプ**と呼ばれ、ここでは B がタイプのみに依存すると仮定する。以下の短い 4 つのステップによって限界が示される（実質的な働きをするのは 3 番目のステップである）。

1. X を M 個の記号上の一様分布（uniform distribution）とし、n 個のサンプルを抽出する。任意の 1 組のサンプルが衝突する確率は $1/M$ であるため、ユニオンバウンドにより $n(n-1)/2$ 組のうち**少なくとも 1 組**が一致する確率は高々 $n(n-1)/(2M)$ である。
2. 0.95 の保証を、$M = n^2$ 個の記号上の一様分布（エントロピーは $2\log n$）に適用する。**この**分布から抽出されたサンプル上では、$B \le 2\log n$ が少なくとも確率 0.95 で成り立つ。
3. 第 1 ステップより、サイズ n のサンプルを $M = n^2$ 個の記号から抽出するとき、それが n 個のすべて異なる値からなる確率は $\frac{1}{2}$ より大きい。確率の和が 1 を超える 2 つの事象は必ず重なりを持つため、$B \le 2\log n$ を満たすすべて異なる値からなるサンプルが少なくとも 1 つ存在する。
4. n 個のすべて異なる値からなるサンプルはいずれもまったく同じタイプを持つため、実際にどの分布から生成されたかにかかわらず、**あらゆる**重複なしサンプルに対して B は同じ値（高々 $2\log n$）を返さなければならない。

ここで、この限界を $M = 2^{256}$ 個の記号上の一様分布（エントロピーは $256\ln 2 = 177$ nat）で試してみよう。$n = 10^6$ 個のサンプルでは、すべてのサンプルが互いに異なる確率は少なくとも $1 - 4.3\times10^{-66}$ であり、そのようなサンプルに対して B が保証できるのは高々 $2\log 10^6 = 27.6$ nat にすぎない。100 万個のサンプルがあっても 177 nat を保証することはできず、それどころか 30 nat すら保証できないのである。演習問題 4 では定数をさらに精密化し、どれだけのサンプル数が必要になるかを問う。

McAllester and Stratos (2020) は、推定量がサンプルのタイプのみに依存するという制限を設けずに一般の定理を証明した。

> **定理（McAllester–Stratos、非形式的）** 分布によらない $I(X;Y)$ の任意の下界であって、一定の信頼水準のもとで n 個のサンプルから成り立つものは $O(\log n)$ である。

この障壁の背後には相互情報量の不変性がある。X または Y にどのような可逆なラベルの付け替えを施しても $I(X;Y)$ は変化しないため（37.6.1.5 節）、座標について一切の仮定を置かない推定量はあらゆるラベルの付け替えに対して整合的に振る舞わなければならず、手がかりとして残るのは重複のパターンだけになる。$O(\log n)$ の壁を超えるには、データについて何らかの仮定を置かなければならない。KSG は座標間の距離に意味があり密度が滑らかであると仮定するが、その代償として有限サンプルでの保証を持たない推定値となる。37.7.3 節の変分下界は $\log n$ よ

り大きな数値を報告し得るが、実際にそのような値が出たときにどこまで信頼できるかについては 37.7.4.4 節で確認する。

37.7.2.5 圧縮フェーズをめぐる論争

いま見てきた 2 つの限界、すなわち座標に依存するヒストグラムと対数的なサンプル上限は、深層学習における有名な論争に光を当てる。Shwartz-Ziv and Tishby (2017) は、$4{,}096$ 通りの 2 進パターン（12 ビット）上で小さな分類器を訓練し、訓練を通じて各隠れ層 Z の活性化値（activation）をビン分割して、$I(X;Z)$ と $I(Y;Z)$ の推定値をプロットした（X は入力、Y はラベル）。著者らは、両方の量が上昇する短い**適合**（fitting）フェーズに続いて、$I(X;Z)$ が減少する一方で $I(Y;Z)$ が高いまま保たれる長い**圧縮**（compression）フェーズが観測されたとし、この圧縮こそが深層ネットワークが汎化（generalization）する理由を説明するものだと提唱した。その後、Saxe et al. (2018) は、この圧縮フェーズは tanh のように飽和する活性化関数では現れるものの、ReLU 活性化関数では消滅し、それでも ReLU ネットワークはまったく遜色なく汎化することを示した。

ヒストグラムは実際には何を測っていたのだろうか。決定論的なネットワークでは、ビン分割された活性化値 Z_{bin} は入力 X の決定論的な関数である。したがって $H(Z_{\text{bin}} \mid X) = 0$ となり、ビン分割による推定値は $I(X;Z_{\text{bin}}) = H(Z_{\text{bin}})$、すなわち単にビン分割された活性化値のエントロピーへと帰着する。この量は入力数の対数である $\ln 4096 = 8.3$ nat を決して超えることができず、異なる入力が同じヒストグラムのセルに潰れるたびに減少する。これは、訓練によって飽和する tanh ユニットが両端の 2 つのビン（± 1 付近）へと追い込まれるときにまさに起きる現象である。真の相互情報量 $I(X;Z)$ 自体は微動だにしない。異なる入力が異なる活性化値を生み出す限り、有限の入力集合に対しては $I(X;Z) = H(X)$ であり、連続入力に対しては $I(X;Z)$ は無限大である（37.6.1.3 節）。プロットされた曲線が追跡していたのは、固定されたビン上で活性化値がどのようにクラスタ化するかという推定量側の性質であり、ネットワークの真の相互情報量ではない（演習問題 8）。

37.7.3 変分下界

カウントに基づく手法では高次元における大きな相互情報量を測定できないため、変分法では目標そのものを切り替える。$\log \frac{p(x,y)}{p(x)\,p(y)}$ に現れる未知の密度を直接推定する代わりに、補助モデルを導入し、そのモデルがどれほど不完全であっても期待スコアが必ず $I(X;Y)$ の**下界**となるように構成するのである。その期待値を最大化するようにモデルを訓練すれば、$I(X;Y)$ を決して超えることなく下界を引き締められる。こうして推定は最適化（optimization）へと変わる。その出力には性質の異なる 2 種類の誤差が含まれる。最適モデルに届かない可能性のある補助モデル由来の近似誤差と、母集団の期待値を有限バッチの平均で置き換えなければならないために生じる標本誤差（sampling error）である。

ここでは、Poole et al. (2019) によって単一の枠組みに統一された 4 つの変分下界を順に調べる。1 つ目は正規化されたデコーダを用いる下界（Barber–Agakov）である。続く 2 つは非正規化のスコア関数を用い、Donsker–Varadhan（DV）下界と、Nguyen・Wainwright・Jordan（NWJ）の下界を与える。4 つ目の InfoNCE は独立した節を設けるに値するほど重要である。いずれの証明も、ただ 1 つの道具、すなわち KL ダイバージェンスは常に非負であるというギブスの不等式（Gibbs' inequality、37.1 節）に依拠している。

37.7.3.1 Barber–Agakov 下界

最も単純な変分下界は、未知の事後分布（posterior distribution）$p(x \mid y)$ を計算可能なモデル $q(x \mid y)$ に置き換えるものである。

命題（Barber–Agakov 下界） 任意の条件付き密度 $q(x \mid y)$ に対して、次が成り立つ。

$$I(X;Y) \geq H(X) + \mathbb{E}_{(x,y)\sim P_{X,Y}}\left[\log q(x \mid y)\right], \tag{37.75}$$

等号は $q(x \mid y) = p(x \mid y)$ のとき、かつそのときに限り成り立つ。連続な X に対しては、微分エントロピー（differential entropy）$h(X)$ が $H(X)$ を置き換える。

証明 $I(X;Y) = H(X) - H(X \mid Y)$ から出発し、Equation 37.75 の右辺を差し引くと次のようになる。

$$\begin{aligned} I(X;Y) - H(X) - \mathbb{E}\left[\log q(x \mid y)\right] &= \mathbb{E}\left[\log \frac{p(x \mid y)}{q(x \mid y)}\right] \\ &= \mathbb{E}_{y}\left[D_{\mathrm{KL}}(p(\cdot \mid y) \,\|\, q(\cdot \mid y))\right] \geq 0. \end{aligned}$$

不等号は、各 y に対してギブスの不等式を適用し、P_Y にわたって平均をとることから従う。等号はすべての y に対して KL ダイバージェンスが 0 になるとき、すなわち $q(x \mid y) = p(x \mid y)$ のとき、かつそのときに限り成り立つ。■

Barber–Agakov 下界 (Barber and Agakov 2003) は、エンコーダ・デコーダの情報理論的な解釈を与える。もしデコーダ $q(x \mid y)$ が低い交差エントロピーで X を Y から再構成できるなら、X と Y は多くの情報量を共有しているに違いない。実用上の制約は $H(X)$ の項であり、これは通常未知である。幸いにも $H(X)$ は q に依存しないため、q に関して下界を**最大化**する際にはこの値を知る必要が一切ない。$H(X) = -\mathbb{E}[\log p(x)]$ と書き直せば等価な比の形式 $I(X;Y) \geq \mathbb{E}\left[\log\left(q(x \mid y)/p(x)\right)\right]$ が得られ、これが InfoNCE 下界を証明する際の鍵となるステップになる。

本節を通じて、モデルはその下界の値を**保証する**（certify）と表現する。この保証は母集団の期待値のレベルで成り立つものである点に留意されたい。有限バッチの平均はその保証値を推定しているにすぎず、その上下に変動し得る。

標準的な分類器は身近な特殊ケースである。相互情報量の対称性により、X と Y を入れ替えて分類器 $q(y \mid x)$ をラベルのデコーダとして扱うことができる。これにより 37.6.2.4 節の損失による保証（loss certificate）が復元される。すなわち、サンプル数の均衡した k 個のクラスとホールドアウト交差エントロピー L に対して、$I(X;Y) \geq \log k - L$ が成り立つ。$L \geq 0$ であるため、10 クラスの分類器は $\ln 10 = 2.30$ nat を超える情報量を決して保証できない。InfoNCE もこれとまったく同じ形をとることになる。

ガウス分布のペアを見ると、不一致なデコーダによってどれほどの情報が失われるかがよくわかる。二乗誤差（squared error）のみで予測器を訓練することは、単位分散のガウスデコーダ $q(x \mid y) = \mathcal{N}(x; ay, 1)$ を用いることに対応する。$h(X) = \frac{1}{2}\log(2\pi e)$ であるから、Barber–Agakov 下界を計算すると $\frac{1}{2} - \frac{1}{2}\mathbb{E}[(X - aY)^2] = \frac{1}{2} - \frac{1}{2}(1 - 2a\rho + a^2)$ となり、$a = \rho$ で最大値 $\rho^2/2$ をとる。$\rho = 0.9$ のとき、このデコーダが保証するのは 0.41 nat（真の値は 0.83 nat）にとどまり、ρ がどれほど 1 に近づいても $\frac{1}{2}$ nat を超えて保証することは決してない。失われた情報量は、デコーダの分散を 1 に固定し、真の条件付き分散（conditional variance）$1 - \rho^2$ に合わせなかったことに対するペナルティである（演習問題 2）。

37.7.3.2 Donsker–Varadhan 下界

2 つ目の下界は、Barber–Agakov の 2 つの要件をともに取り除くものであり、$H(X)$ も正規化された密度 q も必要としない。相互情報量は同時分布から周辺分布の積への KL ダイバージェンス

であるため（37.6 節）、KL ダイバージェンスに対する任意の下界は直ちに $I(X;Y)$ の下界を与える。Donsker–Varadhan 下界は、任意のスコア関数 T（37.3.4 節と同様、**クリティック**と呼ばれる）を用いることで、正規化の制約（constraint）を一切課さずにこれを実現する。

命題（Donsker–Varadhan 下界） 分布 P、Q および任意の関数 T（ただし $\mathbb{E}_P[T]$ が定義され $\mathbb{E}_Q[e^T] < \infty$ を満たすもの）に対して、次が成り立つ。

$$D_{\mathrm{KL}}(P\|Q) \geq \mathbb{E}_P[T] - \log \mathbb{E}_Q\left[e^T\right], \tag{37.76}$$

等号は $T = \log \frac{dP}{dQ} + c$（$c$ はある定数）のとき、かつそのときに限り成り立つ。$P \ll Q$ かつダイバージェンスが有限のとき、T にわたる上限は $D_{\mathrm{KL}}(P\|Q)$ を達成し、KL ダイバージェンスの Donsker–Varadhan 表現を与える (Donsker and Varadhan 1983)。

証明 P が Q に関して絶対連続でない場合、左辺は無限大となり不等式は自明に成り立つ。そうでない場合、クリティックを用いて Q に重み付けを施す。すなわち、**傾斜分布**（tilted distribution）$\tilde{Q}$ を密度 $\tilde{q}(x) = q(x)\, e^{T(x)}/Z$ によって定義する。ここで、$Z = \mathbb{E}_Q[e^T]$ は正規化定数（normalizing constant）である。このとき、次の式が成り立つ。

$$\begin{aligned} D_{\mathrm{KL}}(P\|Q) - (\mathbb{E}_P[T] - \log Z) &= \mathbb{E}_P\left[\log \frac{p}{q} - T + \log Z\right] \\ &= \mathbb{E}_P\left[\log \frac{p}{\tilde{q}}\right] = D_{\mathrm{KL}}(P\|\tilde{Q}) \geq 0 \end{aligned}$$

最後の不等号はギブスの不等式による。等号は $P = \tilde{Q}$、すなわち $p/q = e^T/Z$、言い換えれば $T = \log(p/q) + \log Z$ のとき、かつそのときに限り成り立つ。■

この下界の緩みは単に $D_{\mathrm{KL}}(P\|\tilde{Q})$ であるため、クリティックが正確であることは、Q を e^T で傾斜させたときに P が再現されることとちょうど同値である。また、T を任意の定数 c だけシフトしても 2 つの項の間で相殺されることにも注目したい。$P = P_{X,Y}$ および $Q = P_X \otimes P_Y$ とおけば、任意のクリティック $T(x,y)$ に対する相互情報量の下界が次のように得られる。

$$I(X;Y) \geq \mathbb{E}_{P_{X,Y}}[T] - \log \mathbb{E}_{P_X \otimes P_Y}\left[e^T\right],$$

その最適なクリティックは、任意の加法定数の違いを除いて $\mathrm{pmi}(x,y)$ である。実際には、第 1 の期待値は観測されたペアにわたって T を平均することで推定し、第 2 の期待値は同じミニバッチから作った非対応ペアにわたって e^T を平均することで推定する。T をニューラルネットワークでパラメータ化すると **MINE**（mutual information neural estimation）が得られる (Belghazi et al. 2018)。

37.7.3.3 NWJ 下界

Donsker–Varadhan 下界は期待値の意味ではタイトであるが、$\mathbb{E}_Q[e^T]$ の外側にある対数がミニバッチ上で厄介な統計的バイアスを生む。log は凹関数であるため、イェンセンの不等式より、バッチ平均の対数は平均的に真の期待値の対数より小さくなる。より小さな数を差し引くことになるため、バッチ推定値 $\overline{T} - \log \overline{e^T}$ は平均的に**上方向**へと押し上げられるが、これは下界の推定量としては望ましくない性質である。

初等的な接線不等式 $\log z \leq z/e$（すべての $z > 0$ で成り立ち、$z = e$ で等号成立）を使えば、この対数を取り除くことができる。これを $Z = \mathbb{E}_Q[e^T]$ に対して Equation 37.76 に適用すると、すべてのクリティック T について次の式が得られる。

$$I(X;Y) \geq \mathbb{E}_{P_{X,Y}}[T] - e^{-1}\,\mathbb{E}_{P_X \otimes P_Y}\left[e^{T}\right]. \tag{37.77}$$

これが **NWJ 下界**である (Nguyen et al. 2010)。固定された任意のクリティック T に対して、NWJ が DV を上回ることは決してなく、両者が一致するのは $Z = e$ のときに限られる。この下界は $T^* = 1 + \text{pmi}$ においてタイトになる。DV がクリティックへの定数の加算に対して不変であるのに対し、NWJ はその定数を 1 に固定する。$e^{-1}\,\mathbb{E}_Q[e^T] = \mathbb{E}_Q[e^{T-1}]$ と書き直せば、NWJ は単に 37.3.4 節の一般の f ダイバージェンスの双対下界を KL ダイバージェンスに特殊化した式 $D_{\mathrm{KL}}(P\|Q) \geq \mathbb{E}_P[T] - \mathbb{E}_Q[e^{T-1}]$（同節では凸共役（convex conjugate）を $u\log u$ に対してとることで導出した）にほかならないことがわかる。KL ダイバージェンスに関しては、Donsker–Varadhan がその一般の下界を狭義に精密化しているのである。

第 2 の期待値を対数が囲んでいないため、ミニバッチ上で Equation 37.77 を評価すると下界の**不偏**（unbiased）推定値が得られる。しかし、不偏であることは信頼できることを意味しない。最適なクリティックにおいて、$e^{T^*-1} = e^{\text{pmi}}$ は密度比（density ratio）$p(x,y)/(p(x)p(y))$ である。独立なペアにわたるその期待値は 1 だが、X と Y が強く従属しているとき、その期待値は 0 に近い多数の値の中に混じるごくわずかな巨大値によって支配される。そうしたまれなスパイクをたまたま取り逃したバッチは、第 2 項を著しく過小評価し、結果として下界を大きく超えてしまう。この振る舞いについては 37.7.4.4 節で定量化し、実際に測定する。

37.7.4 InfoNCE

37.7.4.1 対照損失

DV 下界や NWJ 下界の指数項 e^T は正規化されていないため、極端なペアが 1 つあるだけでバッチ全体が支配されてしまう。InfoNCE はバッチの**内部で**正規化して候補どうしを競わせることで、この不安定性を回避する。具体的には、1 つの**正**のペア $(x, y^+) \sim P_{X,Y}$ をサンプリングし（x を**アンカー**と呼ぶ）、あわせて $N-1$ 個の**負例** $y_2, \ldots, y_N$ を周辺分布 P_Y から独立にサンプリングする。全 N 個の候補をクリティック $T(x,y)$ でスコア付けし、そのスコアをソフトマックスに通すことで、どの候補が真の相手であるかを予測する。その結果得られる交差エントロピーが **InfoNCE 損失** (Oord et al. 2018) であり、次の式で定義される。

$$\mathcal{L}_{\text{NCE}} = -\mathbb{E}\left[\log \frac{e^{T(x,y^+)}}{e^{T(x,y^+)} + \sum_{j=2}^{N} e^{T(x,y_j)}}\right], \tag{37.78}$$

ここで期待値は、正のペアと $N-1$ 個の負例の両方にわたってとる。対応する相互情報量の推定値は、この損失を $\log N$ から差し引いたものである。

$$\hat{I}_{\text{NCE}} = \log N - \mathcal{L}_{\text{NCE}}. \tag{37.79}$$

論文によっては分母を平均 $\frac{1}{N}\sum_{j=1}^{N} e^{T(x,y_j)}$ として書くこともあり、その場合は対数比の期待値がそのまま $\hat{I}_{\text{NCE}}$ に等しくなる。概念的には、$\mathcal{L}_{\text{NCE}}$ は固定されたラベルではなくバッチ内の他のデータ例をクラスとする単なる N クラス交差エントロピーにほかならない。これは CPC、SimCLR、CLIP の背後にある訓練の目的関数であり (Oord et al. 2018; Chen et al. 2020; Radford et al. 2021)、そこでは x と y が同一データ例の 2 つのビューまたは 2 つのモダリティを表し、バッチ内の残りのデータ例が負例の役割を果たす。

Figure 37.19 にこの設定を示す。非自明な最小のケースである $N = 2$ では、モデルが直面する負例は 1 つだけであり、損失は真の相手と 1 つのディストラクタを見分ける二値交差エントロピ

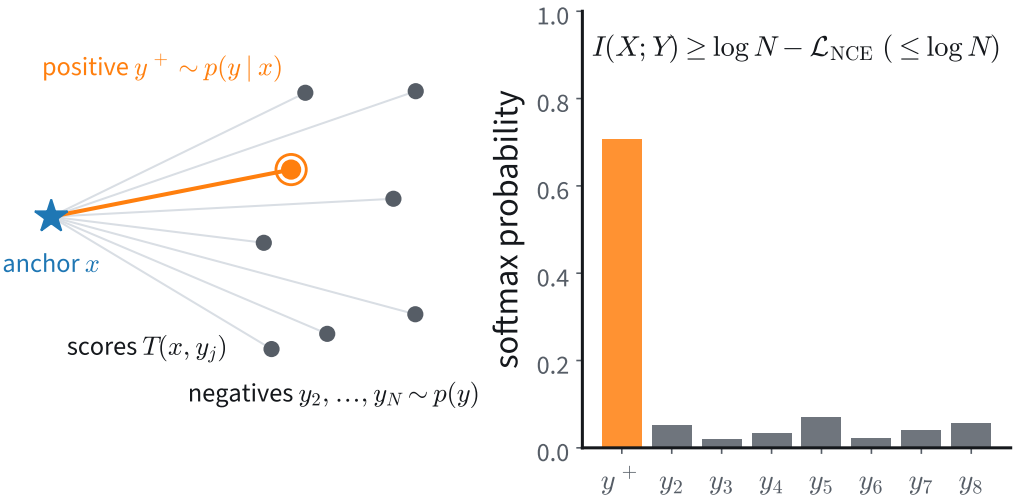


Figure 37.19: InfoNCE のタスク。左：アンカー x が、それと同時にサンプリングされた正例の相手 y^+、および周辺分布から独立にサンプリングされた $N-1$ 個の負例に対してクリティックによりスコア付けされる。各直線はスコア $T(x, y_j)$ を表す。右：N 個のスコアに対するソフトマックスが正例を識別しなければならない。得られる交差エントロピーは $I(X;Y) \geq \log N - \mathcal{L}_{\text{NCE}}$ を保証するが、この下界は決して $\log N$ を超えることができない。

ー（binary cross-entropy）に帰着する。このとき完璧なモデルであっても 2 つの選択肢から正しい方を選んだことしか示せないため、推定値は $\ln 2 = 0.69$ nat で頭打ちになる。

37.7.4.2 InfoNCE 下界

次の命題は、InfoNCE 損失の最小化が相互情報量の下界を保証することを確認するとともに、$\log N$ という上限がどこから生じるかを示すものである (Poole et al. 2019)。

命題（InfoNCE 下界）任意の可測なクリティック T および任意の $N \geq 1$ に対して、次が成り立つ。

$$I(X;Y) \geq \log N - \mathcal{L}_{\text{NCE}}. \tag{37.80}$$

さらに、右辺は高々 $\log N$ である。

証明候補に $y_1 = y^+, y_2, \ldots, y_N$ と番号を付け、単一の確率的なバッグ $V = (Y_1, \ldots, Y_N)$ にまとめる。ここで負例 $Y_{2:N}$ は (X, Y_1) と独立である。証明の方針は、Y をバッグ V に置き換え、条件付き分布（conditional distribution）として X が与えられたもとでのバッグ上の正規化された分布を構成し、Barber–Agakov 下界を適用することである。

ステップ 1：独立な負例は情報量を増やさない。相互情報量の連鎖律（37.6 節）より、$I(X; Y_1, Y_{2:N}) = I(X; Y_1) + I(X; Y_{2:N} \mid Y_1)$ が成り立つ。$Y_{2:N}$ は (X, Y_1) と独立であるため、X とも（Y_1 が与えられたもとで）独立のままである。したがって第 2 項は 0 となり、$I(X;V) = I(X;Y)$ が得られる。

ステップ 2：ソフトマックス比が正規化されたモデルを定義する。ここが鍵となる計算である。再スケーリングしたソフトマックススコアを次のように定義する。

$$\gamma(x, v) = \frac{e^{T(x, y_1)}}{\frac{1}{N}\sum_{j=1}^{N} e^{T(x, y_j)}},$$

これは $(0, N]$ に値をとる。$p(v)$ を、第 1 成分も含めて N 個すべてを P_Y から独立にサンプリングしたバッグの分布とする。全 N 個の成分は独立同分布であるため、それらを置換しても $p(v)$ は変化しない。したがって、全 N 個の比 $e^{T(x,Y_i)} / \left(\frac{1}{N}\sum_j e^{T(x,Y_j)}\right)$ は $p(v)$ のもとですべて同じ期待値を持ち、固定された任意の x に対して次の式が成り立つ。

$$\begin{aligned}\mathbb{E}_{p(v)}\left[\gamma(x, V)\right] &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{p(v)}\left[\frac{e^{T(x,Y_i)}}{\frac{1}{N}\sum_j e^{T(x,Y_j)}}\right] \\ &= \mathbb{E}_{p(v)}\left[\frac{\frac{1}{N}\sum_i e^{T(x,Y_i)}}{\frac{1}{N}\sum_j e^{T(x,Y_j)}}\right] = 1.\end{aligned}$$

よって、$q(v \mid x) = p(v)\,\gamma(x, v)$ はバッグ上の有効な条件付き確率（conditional probability）分布となる。

ステップ 3：バッグに対して Barber–Agakov を適用する。 Y_1 は周辺分布 P_Y を持ち $Y_{2:N}$ と独立であるため、V の周辺分布も $p(v)$ である。Equation 37.75 の比の形式を、2 変数の役割を入れ替えて (X, V) に適用すると（I の対称性を用いる）、次の式が得られる。

$$I(X; V) \geq \mathbb{E}_{p(x,v)}\left[\log \frac{q(v \mid x)}{p(v)}\right] = \mathbb{E}\left[\log \gamma(X, V)\right] = \log N - \mathcal{L}_{\text{NCE}}.$$

最後の等号は、$\log \gamma$ が $\log N$ と正例に割り当てられた対数ソフトマックス確率との和に等しいことから成り立つ。これをステップ 1 と合わせれば下界が証明される。最後に、正例のソフトマックス確率は 1 を超えられないため $\mathcal{L}_{\text{NCE}} \geq 0$ であり、したがって右辺は高々 $\log N$ である。■

直観的には、N 個の候補から真の相手を選び出すことは、たとえ誤り率が 0 であっても高々 $\log N$ nat、すなわち N 個の選択肢からの 1 回の選択に含まれる情報量しか示さない。10 クラス分類器が $\ln 10$ nat を超える情報量を決して保証できなかったのと同様に、いかなるクリティックもそのタスクからそれ以上の情報量を絞り出すことはできない。N を増やせばこの上限が引き上げられるため、これが CPC、SimCLR、CLIP といった対照手法で大きなバッチを用いる動機の 1 つとして挙げられる。ただし、それが話のすべてではない。本節の最後で見るように、表現を優れたものにするのはよりタイトな下界ではないからである。なお、この損失は $T^*(x, y) = \text{pmi}(x, y) + c(x)$ という形の任意のクリティックによって最小化される。ここで $c(x)$ はアンカーのみに依存する任意の関数である（演習問題 5）。

37.7.4.3 厳密なクリティックのもとでの上限

厳密なクリティック $T = \text{pmi}$ を用いれば、$\hat{I}_{\text{NCE}}$ は真の $I(X; Y)$ に沿って $\log N$ までまっすぐ追随し、そこで鋭く折れ曲がって平坦になると考えるのが自然だろう。しかし $T = \text{pmi}$ を Equation 37.79 に代入すると、より滑らかで、かつ厳しい現実を突きつける振る舞いが明らかになる。

$$\hat{I}_{\text{NCE}} = I(X; Y) - \mathbb{E}\left[\log \frac{1}{N}\sum_{j=1}^{N} e^{\text{pmi}(x, y_j)}\right]. \tag{37.81}$$

推定値は、真の相互情報量から緩みの項（命題により非負）を引いたものに等しい。簡単な概算によって、この緩みの大きさを見積もることができる。対数の内側には N 個の項の平均がある。そのうち $N-1$ 個の負例は x と独立であるためそれぞれの平均は 1 であり（本節の冒頭で確かめた第 2 の恒等式）、平均全体に対しておおよそ 1 だけ寄与する。一方、正例は $e^{\mathrm{pmi}(x,y^+)}/N$ だけ寄与し、典型的なペアに対してこれはおおよそ e^I/N となる。確率的な平均をこの典型値で置き換えると、次の近似が得られる。

$$\hat{I}_{\mathrm{NCE}} \approx I - \log\left(1 + \frac{e^I}{N}\right) = -\log\left(e^{-I} + \frac{1}{N}\right). \tag{37.82}$$

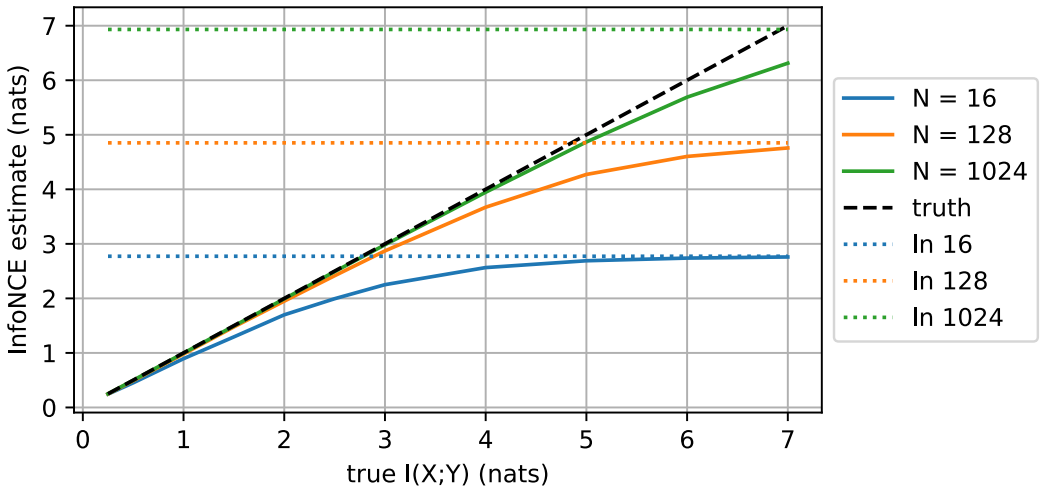
右辺は I と $\log N$ の滑らかな最小値である。これは I に密接に追随し（$e^I \ll N$ のとき）、$\log N$ に漸近し（$e^I \gg N$ のとき）、遷移領域ではその両方よりかなり下へと曲がり込む。ちょうど $I = \log N$ の地点では $\log N - \log 2$ となり、0.69 nat 低くなる。最初の予想が見落としていたのは、平均が 1 となる負例とともに正例自身が分母の中に入っているという事実である。正例が分母を支配するようになるのは、e^I が N を大きく上回ってからにすぎない。

裾の重い平均をその期待値で置き換えるのは大雑把な近似にすぎないため、Equation 37.82 を実験的に確かめておくべきだろう。$N = 128$ かつ $I = 4$ のとき、この式は 3.6 nat と予測する（上限 $\ln 128 = 4.85$ にはまだ十分な余裕があるように見えるにもかかわらずである）。以下では、バッチ内の全 $N \times N$ 個のペアを同時にスコア付けする。第 i 行はアンカー x_i をすべての y_j と組にするため、その対角成分が正例となり、$N-1$ 個の非対角成分が負例となる。各行は候補を共有しているものの、下界は依然として有効である（演習問題 6）。$I(X;Y)$ を変化させながら、$N = 16$、128、1024（上限はそれぞれ 2.77、4.85、6.93 nat）に対する経験的な推定値を真の値と比較してみよう。

```
def infonce_bound(f):
    """log N minus the InfoNCE loss of an N x N score matrix f[i, j]."""
    return np.log(len(f)) + np.mean(np.diag(f) - logsumexp(f, axis=1))

def exact_scores(x, y, rho):
    """The exact critic T = pmi on all N x N pairs (x_i, y_j) of a batch."""
    return pmi_gauss(x[:, None], y[None, :], rho)

true_mi = np.array([0.25, 0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 6, 7])
rhos = np.sqrt(1 - np.exp(-2 * true_mi))                # invert mi_gauss
Ns = (16, 128, 1024)
ests = np.array([[np.mean([infonce_bound(exact_scores(*gauss_pair(rng, r, N), r))
                           for _ in range(200 if N <= 128 else 50)])
                  for r in rhos] for N in Ns])              # (N, level)
ceilings = [[np.log(N)] * len(true_mi) for N in Ns]
d2l.plot(true_mi, [*ests, true_mi, *ceilings], 'true I(X;Y) (nats)',
         'InfoNCE estimate (nats)',
         legend=[f'N = {N}' for N in Ns] + ['truth'] + [f'ln {N}' for N in Ns],
         fmts=('C0-', 'C1-', 'C2-', 'k--', 'C0:', 'C1:', 'C2:'),
         figsize=(5.5, 3), legend_loc='outside')
for I in (1, 4, 7):
    k = list(true_mi).index(I)
    print(f'I = {I}:  ' + '   '.join(f'N = {N}: {e:.2f}'
                                      for N, e in zip(Ns, ests[:, k])))
```

```
I = 1:  N = 16: 0.89   N = 128: 0.98   N = 1024: 1.00
```

```
I = 4:  N = 16: 2.56   N = 128: 3.67   N = 1024: 3.95
I = 7:  N = 16: 2.76   N = 128: 4.76   N = 1024: 6.31
```



どの曲線にも鋭い折れ曲がりはない。いずれも早い段階で対角線から離れ、それぞれの上限に向かって滑らかに曲がっており、出力された 9 個の値すべてが 0.1 nat 以内の誤差で Equation 37.82 に一致している。$I = 1$ では、どのバッチサイズでも真の値の約 0.1 nat 以内に収まる。$I = 4$ になる頃には、サイズ 128 のバッチが報告する値は 3.7 nat にとどまり、$N = 1024$ でさえわずかに遅れをとり始める。$I = 7$（上限 6.93 をわずかに超えた地点）では、$N = 1024$ の報告値は 6.3 にすぎない。

なぜこうなるかは、2 つの厳密な事実から説明できる。第 1 に、対数の内側の平均は少なくとも正例自身の項 $e^{\mathrm{pmi}(x,y^+)}/N$ 以上であるため、緩みは少なくとも $\mathbb{E}[\mathrm{pmi}(x,y^+)] - \log N = I - \log N$ となり、ここから $\log N$ の上限が復元される。第 2 に、この緩みが消えるのは極限 $N \to \infty$ においてのみである（演習問題 5）。厳密な上限 $\log N$ は、この有限バッチによる緩みの最も大雑把な兆候にすぎない。

この滑らかな圧縮により、推定値から真の値を逆算することも難しくなる。$N = 128$ のとき、測定値 3.7 は真の 4 nat に対応し、4.75 付近の測定値は 7 nat に対応するため、4 nat から無限大までのあらゆる真の値が、目盛りの上端わずか 1.2 nat の幅に押し込められてしまう。

画像分類（image classification）を例にとると、これらの数値の意味が実感できる。ある画像の 2 つのビューが、それぞれ均衡した 1,000 個のカテゴリの中からそのクラスを一意に決定するとしよう。ラベルはどちらのビューの関数でもあるため、データ処理不等式（37.6.2.2 節）より、2 つのビューは少なくともラベルのエントロピーである $\ln 1000 = 6.91$ nat を共有する。候補数 $N = 128$ の InfoNCE 損失ではそれに近い値すら決して保証できず、$N = 1024$ を用いても推定値は 6.3 にしかならない（真の共有情報量は 7 であるにもかかわらず）。2 つのビューが細粒度分類に十分な情報を保持しているかどうかは、このような数値からは到底読み取れないのである。

37.7.4.4 厳密なクリティックによる比較

これで DV、NWJ、InfoNCE を同じ土俵で比較できる。バッチサイズを $N = 128$ とし、いずれにも厳密な最適クリティック $T^* = 1 + \mathrm{pmi}$ を与える（定数を加えても InfoNCE は変化しない）。DV と NWJ は独立なペアをバッチ内の N 個の非対応ペアから取り出すが、先ほど指摘した裾の重い振る舞いを踏まえれば、コードを 1 行も実行する前に何が起きるかを予測できる。

典型的な同時ペアの pmi は I に近い。直積分布 $P_X \otimes P_Y$ のもとでは、そのようなペアは確率お

およそ e^{-I} で生じる。なぜなら $p(x)\,p(y) = p(x,y)\,e^{-\mathrm{pmi}}$ だからである。言い換えれば、非対応ペアのうちおおよそ e^{I} 個に 1 個が、本物のペアで通るほどもっともらしい見た目をしている。こうした**希少ペア**（rare pairs）はそれぞれ巨大な重み $e^{\mathrm{pmi}} \approx e^{I}$ を持ち、それら全体で期待値 $\mathbb{E}[e^{\mathrm{pmi}}] = 1$ のすべてを担っている。サイズ N のバッチには平均しておおよそ Ne^{-I} 個の希少ペアが含まれる。その数は約 6 個（$I = 3$）、2 個（$I = 4$）、そしてわずか 0.3 個（$I = 6$）となる。

バッチ内に希少ペアが複数含まれている限り、e^{pmi} の標本平均は 1 付近にとどまり、3 つの下界はすべて一致するはずである。しかし期待個数 Ne^{-I} が 1 を下回ると、大半のバッチには希少ペアが 1 つも含まれなくなる。このようなバッチを**枯渇**（starved）バッチと呼ぼう。枯渇バッチでは e^{pmi} の平均が 0 に近くなるため、NWJ は T^{*} のバッチ平均からほぼ何も引かない値（約 $I + 1$ であり、1 nat 高すぎる）を報告し、DV に至っては極小の数の対数を引くことで何 nat も過大に飛び出してしまう。

NWJ は不偏であるため、そうした枯渇バッチと釣り合いをとる何かが存在するはずである。その釣り合いをとるのが**ヒット**バッチ、すなわちたまたま $e^{\mathrm{pmi}} \geq N$ となる非対応ペアを含んだバッチである。そのようなペアが 1 つあるだけで、バッチ平均には少なくとも 1（e^{pmi} の平均への寄与）が加わり、NWJ を少なくとも 1 nat 引き下げて I 付近、ときにはそれよりはるかに低い値へと落とす。$I = 6$ では希少ペアは $e^{\mathrm{pmi}} \approx e^{6} = 403 > N$ を持つため、それを 1 つでも含むバッチはすべてヒットバッチになる。バッチあたりの希少ペアが平均 0.3 個であれば、およそ 4 バッチに 1 つがヒットバッチとなり、残りの大半は枯渇バッチになるはずである。

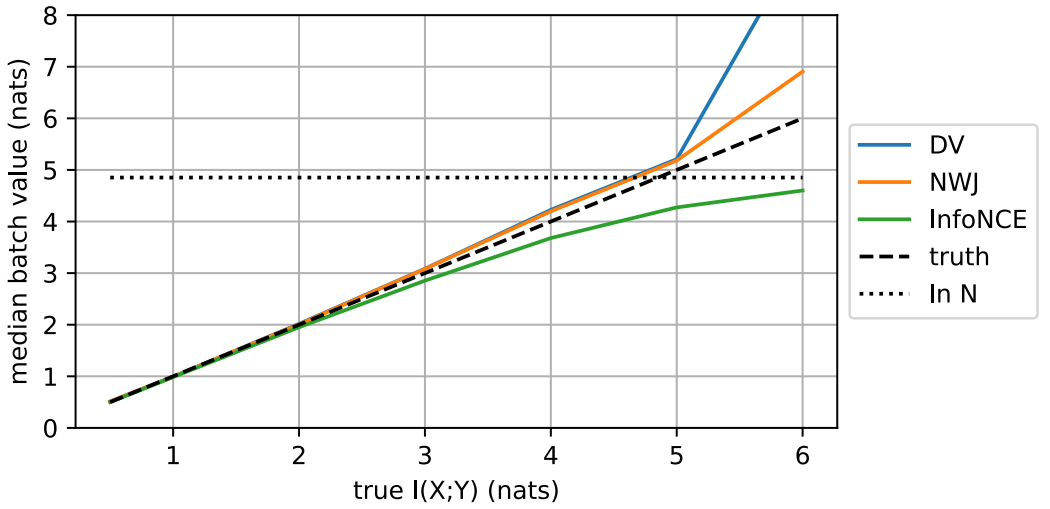
InfoNCE はこれとはまったく異なる振る舞いを見せる。バッチ内で正規化するため、もっともらしい負例が現れてもソフトマックスの高々 1 行にしか影響しない。そのため、Equation 37.82 が予測するとおりの地点、すなわち約 4.6 nat（$I = 6$ のとき）で安定するはずである。切り替わりの境界 $Ne^{-I} \approx 1$ が $I \approx \ln N$ で生じることに注目したい。バッチサイズの対数は、InfoNCE を制限するのとまったく同じだけ DV や NWJ も制限しているのであり、DV と NWJ は目に見える上限の代わりに爆発する分散の陰にその限界を隠しているにすぎない。

200 個のバッチを各 I の水準でサンプリングし、これらの予測を検証してみよう。裾の重い分布は標本平均（sample mean）や標準偏差を歪めるため、各下界について中央値（median）と四分位範囲を報告し、NWJ については標本平均も併記する。最後の 2 列は枯渇バッチ（平均が $e^{\mathrm{pmi}} < 0.1$）の割合とヒットバッチ（$\max e^{\mathrm{pmi}} \geq N$）の割合を記録し、プロットは 3 つの下界の中央値を比較している。

```
def quartiles(a):
    """Median and quartiles of a list of batch values, as text."""
    q1, med, q3 = np.percentile(a, [25, 50, 75])
    return f'{med:5.2f} [{q1:5.2f}, {q3:5.2f}]'

N, true_mi = 128, np.array([0.5, 1, 2, 3, 4, 5, 6])
medians = []
# Last two columns: share of batches whose mean e^pmi over mismatched pairs
# is below 0.1, and share holding a mismatched pair with e^pmi >= N.
print(f'{"I":>3}  {"DV median [IQR]":20}  {"NWJ median [IQR]":20}  '
      f'{"NWJ mean":>8}  {"InfoNCE median [IQR]":20}  {"<0.1":>4}  {">=N":>4}')
for I in true_mi:
    rho, batches = np.sqrt(1 - np.exp(-2 * I)), []
    for _ in range(200):
        x, y = gauss_pair(rng, rho, N)
        T_joint = 1 + pmi_gauss(x, y, rho)              # T* on true pairs
        T_prod = 1 + pmi_gauss(x, np.roll(y, 1), rho)   # T* on mismatched
        e_pmi = np.exp(T_prod - 1)                      # e^pmi, mismatched
```

```
            batches.append((T_joint.mean() - (logsumexp(T_prod) - np.log(N)),  # DV
                            T_joint.mean() - e_pmi.mean(),                     # NWJ
                            infonce_bound(exact_scores(x, y, rho)),
                            e_pmi.mean() < 0.1, e_pmi.max() >= N))
        dv, nwj, nce, low, rare = np.array(batches).T
        medians.append([np.median(a) for a in (dv, nwj, nce)])
        print(f'{I:3.1f}  {quartiles(dv)}  {quartiles(nwj)}  {nwj.mean():8.2f}'
              f'  {quartiles(nce)}  {low.mean():4.0%}  {rare.mean():4.0%}')
    d2l.plot(true_mi, [*np.array(medians).T, true_mi, [np.log(N)] * len(true_mi)],
             'true I(X;Y) (nats)', 'median batch value (nats)',
             legend=['DV', 'NWJ', 'InfoNCE', 'truth', 'ln N'], ylim=[0, 8],
             fmts=('C0-', 'C1-', 'C2-', 'k--', 'k:'), figsize=(5.5, 3),
             legend_loc='outside')
```

```
  I  DV median [IQR]       NWJ median [IQR]      NWJ mean  InfoNCE median [IQR]  <0.1   >=N
0.5   0.51 [ 0.43,  0.58]   0.51 [ 0.43,  0.58]      0.50   0.50 [ 0.45,  0.54]    0%    0%
1.0   0.99 [ 0.88,  1.13]   0.99 [ 0.88,  1.12]      0.99   0.99 [ 0.92,  1.04]    0%    0%
2.0   2.02 [ 1.81,  2.17]   2.01 [ 1.80,  2.16]      1.97   1.95 [ 1.90,  2.01]    0%    1%
3.0   3.08 [ 2.75,  3.47]   3.08 [ 2.70,  3.36]      3.02   2.85 [ 2.79,  2.92]    0%    2%
4.0   4.23 [ 3.74,  4.95]   4.20 [ 3.71,  4.57]      3.76   3.68 [ 3.62,  3.74]    8%    6%
5.0   5.21 [ 4.54,  8.28]   5.18 [ 4.44,  5.94]      4.83   4.27 [ 4.22,  4.31]   32%   36%
6.0   9.47 [ 5.33, 29.04]   6.91 [ 5.10,  7.00]      5.93   4.60 [ 4.58,  4.64]   57%   28%
```



$I = 3$ までは、3 つの中央値はいずれも真の値の約 0.2 nat 以内に収まり、枯渇バッチもヒットバッチもほとんど現れず、ばらつきが最も小さいのは InfoNCE である。そこを超えると、予測どおりに枯渇バッチが現れ始める。その割合は高々約 10 分の 1（$I = 4$）、約 3 分の 1（$I = 5$）、そして半数以上（$I = 6$）へと増えていく。

$I = 6$ の行では、NWJ の中央値は約 $7 = I + 1$ まで上昇し、上側四分位数もそのすぐ上に重なっている。つまり全バッチの半数が丸々 1 nat ほど過大推定しているのである。全体の 4 分の 1 前後を占めるヒットバッチが標本平均を引き戻し、6 から約 0.5 nat 以内に収めている。DV はさらに深刻であり、中央値は真の値より数 nat 高く、上側四分位数は数十 nat に達して曲線がプロットの上端から突き抜けてしまっている。これとは対照的に、InfoNCE は 4.6 nat で盤石に安定しており、四分位範囲も 0.1 未満である。

NWJ のばらつきは解析的にも計算できる。その非対応ペアの項は e^{pmi} のバッチ平均であり、分散は $(e^{2I}-1)/N$ となる。$I=6$ かつ $N=128$ では標準偏差が 36 nat にも達するが、これは 200 回の試行ではめったに含まれないほどまれな極端なヒットバッチによって引き起こされている（演習問題 1）。

以上のことは、母集団の期待値に対して成り立つ変分不等式と何ら矛盾するものではない。むしろ、変分下界が 37.7.2.4 節の分布によらない限界とどのように整合するかを示している。InfoNCE は $\log N$（ただし N は高々サンプルサイズ）を超えることができないため、構成上おのずと対数的な限界に従う。DV と NWJ はより大きな数値を自由に報告できるが、そうした値を報告するときには、そのバッチ推定値はもはや信頼できる下界ではなくなっているのである。

37.7.4.5 クリティックの訓練

実際の応用では真の pmi は未知であるため、クリティックをデータから学習しなければならない。対照損失が真の相互情報量より小さな値を報告したとき、その不足分が不完全なクリティックに由来するのか、それとも下界自体の緩みに由来するのかはどう見分ければよいだろうか。これに答えるため、CLIP や SimCLR で使われているものとまったく同じ形式のデュアルエンコーダ型クリティックを訓練してみよう。すなわち、2 つの小さなネットワーク $g, h : \mathbb{R} \to \mathbb{R}^{16}$ が x と y を埋め込み（embedding）へと写し、その内積（inner product）$T(x,y) = g(x)^{\top} h(y)$ をクリティックとする。するとバッチ内の全 $N \times N$ 個のスコアは 2 つの埋め込み行列の行列積（matrix multiplication）をたった 1 回計算するだけで求まり、ネットワークの評価回数を N^2 回から $2N$ 回へと削減できる。ここでは互いに素な 4,096 ペアのプールを $\rho = 0.99$（真の情報量は 1.96 nat）で 2 つ生成し、一方を訓練用、他方を評価用とする。

```
rho, n_pool = 0.99, 4096
x_tr, y_tr = (a.astype(np.float32) for a in gauss_pair(rng, rho, n_pool))
x_ev, y_ev = (a.astype(np.float32) for a in gauss_pair(rng, rho, n_pool))
```

各エンコーダは、64 個の ReLU ユニットからなる隠れ層を 2 つ持ち、Glorot 一様初期化（initialization）を用いた多層パーセプトロン（multilayer perceptron、MLP）である。得られる $N \times N$ のスコア行列上では、InfoNCE 損失は単なる標準的な多クラス交差エントロピーであり、第 i 行に対する正解クラスは対角成分の正のペアのインデックスである i となる。

```
def encoder(d_out=16, hidden=64):
    """An MLP that embeds a scalar in d_out dimensions."""
    net = torch.nn.Sequential(torch.nn.Linear(1, hidden), torch.nn.ReLU(),
                              torch.nn.Linear(hidden, hidden), torch.nn.ReLU(),
                              torch.nn.Linear(hidden, d_out))
    for layer in net[::2]:                          # the Linear layers
        torch.nn.init.xavier_uniform_(layer.weight)
        torch.nn.init.zeros_(layer.bias)
    return net

def scores(g, h, x, y):
    """All N x N critic values T(x_i, y_j) = g(x_i) . h(y_j)."""
    return g(x[:, None]) @ h(y[:, None]).T

def infonce_loss(f):
    """Cross-entropy of each row's softmax at its positive, the diagonal."""
    return torch.nn.functional.cross_entropy(f, torch.arange(len(f)))
```

```
    g, h = encoder(), encoder()
```

訓練プールからサンプリングしたペアのミニバッチを用いて、1,000 ステップの Adam（学習率 10^{-3}、バッチサイズ $N = 128$）で訓練し、訓練済みモデルを `critic_np` としてラップして `exact_scores` と同様に NumPy 配列を受け取れるようにする。

```
    opt = torch.optim.Adam([*g.parameters(), *h.parameters()], lr=1e-3)
    N = 128
    for step in range(1000):
        idx = rng.choice(n_pool, N, replace=False)
        loss = infonce_loss(scores(g, h, torch.tensor(x_tr[idx]),
                                   torch.tensor(y_tr[idx])))
        opt.zero_grad()
        loss.backward()
        opt.step()

    def critic_np(a, b):
        """The trained critic on NumPy inputs, as an N x N NumPy array."""
        with torch.inference_mode():
            return scores(g, h, torch.tensor(a), torch.tensor(b)).numpy()
```

クリティックの誤差と下界の緩みをきれいに切り分けるため、学習されたクリティックと厳密なクリティックの両方を**同じ**ホールドアウトバッチ上で評価する。以下では、評価プールをサイズ $N \in \{2, 8, 32, 128\}$ の互いに素なバッチに分割し、それらにわたって InfoNCE 下界の平均をとる。なお、$N = 128$ で訓練したクリティックを他の任意のバッチサイズで評価できるのは、最適なクリティック $\mathrm{pmi}(x, y) + c(x)$ が N に依存しないからである。

```
    def infonce_eval(score, x, y, N):
        """Mean InfoNCE bound over the disjoint batches of N pairs in (x, y)."""
        return np.mean([infonce_bound(score(x[i:i + N], y[i:i + N]))
                        for i in range(0, len(x), N)])

    exact_np = lambda a, b: exact_scores(a, b, rho)   # the exact critic T = pmi

    print('   N   ln N   learned   exact')
    for N_ev in (2, 8, 32, 128):
        print(f'{N_ev:4d}  {np.log(N_ev):5.2f}  '
              f'{infonce_eval(critic_np, x_ev, y_ev, N_ev):7.2f}  '
              f'{infonce_eval(exact_np, x_ev, y_ev, N_ev):6.2f}')
    print(f'true I(X;Y) = {mi_gauss(rho):.2f} nats')
```

```
   N   ln N   learned   exact
   2   0.69     0.58    0.58
   8   2.08     1.42    1.42
  32   3.47     1.78    1.80
 128   4.85     1.89    1.91
true I(X;Y) = 1.96 nats
```

どのバッチサイズにおいても、学習されたクリティックは厳密なクリティックと 100 分の数 nat 以内で一致している。したがって、真の値 1.96 nat との残りのギャップはすべて下界そのものに属しており、N が増えるにつれて徐々に縮まり、両方のクリティックが約 1.9 nat（$N = 128$ のとき）に達する。$N = 8$ の行を詳しく見てみよう。上限 $\ln 8 = 2.08$ はすでに真の値 1.96 を上

回っているため、約 0.55 nat の不足は上限に突き当たったせいではない。これは下界の滑らかな最小値による緩みであり、Equation 37.82 が約 0.6 nat と予測する量である。もちろん、スカラーのガウス分布ペアはクリティックにとって容易なタスクであり、高次元ではクリティックの近似誤差は無視できなくなるうえ、既知の正解（ground truth）がなければ 2 つの誤差要因を切り分けることはできない。

SimCLR などの対照システムでは、クリティックにわずかな変更を加える。埋め込み z_x と z_y を単位長に正規化し、そのコサイン類似度（cosine similarity）を**温度**（ソフトマックス温度、softmax temperature）τ で割って $T(x,y) = \mathrm{sim}(z_x, z_y)/\tau$ とおくのである (Chen et al. 2020)。温度はソフトマックスがどれだけ鋭く候補を分離するかを制御すると同時に、それ自体が第 2 の上限を導入する。コサイン類似度は $[-1, 1]$ に収まるため、どの 2 つのスコアの差も $2/\tau$ を超えることはなく、したがってコサイン型クリティックは $2/\tau$ nat 以上を決して保証できない（N がどれほど大きくても同様である。演習問題 3)。このため下界は 4 nat 未満（$\tau = 0.5$ のとき）や 20 nat 未満（$\tau = 0.1$ のとき）に制限される。ゆえに温度が大きいと、バッチサイズが制約になるよりずっと手前で下界が頭打ちになってしまう。

実用上はどの推定量を選ぶべきだろうか。高々数 nat を共有する 1 次元または 2 次元変数のペアには、最近傍カウントが適している。その領域では KSG が真の値の約 0.1 nat 以内に収まっていた。一方、高次元ベクトルに対しては、本節のどの推定量も大きな相互情報量を正確には測れないため、問いは 2 つのビューが**どれだけ**の情報を共有するかから、**少なくともどれだけ**共有するかへと移る。InfoNCE はその問いに高い信頼性で答えてくれる。メモリが許す限り多くの負例を用いて訓練し、$\log N - \mathcal{L}_{\mathrm{NCE}}$ を厳密に下界として解釈すればよい。$\log N$ から 1 nat 前後以内の測定値が得られた場合、そこからわかるのは両ビューが少なくともそれだけの情報を共有しているということだけであり、逆に低い測定値は単にクリティックの弱さを反映している可能性がある。DV と NWJ については、独自の得意領域が存在しない。バッチ推定値が信頼できる $\log N$ より数 nat 低い領域では、InfoNCE もほぼ同等の精度をはるかに高い安定性で達成し、それより高い領域では典型的なミニバッチが過大推定を起こすからである。

最後に、よりタイトな下界が自動的により優れた表現を生むわけではない。Tschannen et al. (2020) は、下流タスク（downstream task）の性能が、最大化する相互情報量の下界のタイトさとほとんど相関しないことを示した。むしろ緩い下界の方が強力な表現をもたらすことも多く、エンコーダやクリティックのアーキテクチャの選択は推定される情報量と同等以上に重要である。こうして 2 つの根本的な問いが未解決のまま残されている。高次元において大きな相互情報量をどうすれば信頼性高く推定できるのか、そして緩い変分下界を最大化することでなぜこれほど有効な表現が得られるのか、という問いである。

37.7.5 まとめ

有限のサンプルだけから相互情報量を直接測定できるのは、変数が低次元で共有情報量が控えめな場合に限られる。その領域を超えると、あらゆる推定量は識別可能な結果の数の対数によって制約される。すなわち、ヒストグラムでは $\log b$（ビン数が b 個のとき）、KSG の最近傍カウントではおおよそ $\log(n/m)$（しかも高次元ではその上限よりはるかに低いところで飽和する）、そして分布によらない任意の下界では $O(\log n)$ となる。変分法は、デコーダまたはクリティックでパラメータ化された下界を最大化することで密度推定を回避する。対照学習の交差エントロピー目的関数である InfoNCE は低分散で扱いやすく、厳密なクリティックのもとでおおよそ $-\log(e^{-I} + 1/N)$ を与え、決して $\log N$（候補数 N）を超えることがない。Donsker–Varadhan と NWJ には厳密な上限はないものの、そのミニバッチ推定値が信頼できるのは $\log N$ より 2 nat ほど低い領域までに限られる。ガウス分布のベンチマークでは、学習されたデュアルエンコーダクリティックが厳密な pmi クリティックと 100 分の数 nat 以内で一致し、観測されたギャップが下界そのものに由来することが確認された。とはいえ、情報量の下界のタイトさそれ自体が、学習された表現の質を予測するわけではない。

37.7.6 演習問題

1. **指数をとった pmi の重い裾。**●(X, Y) を相関 ρ、$I = -\frac{1}{2}\log(1-\rho^2)$ を持つガウス分布のペアとし、$Q = P_X \otimes P_Y$ とする。Q のもとでは e^{pmi} の平均は 1 であるが（任意の ρ に対して成り立つ）、典型的な独立ペアの e^{pmi} は 0 の近くになる（ρ が 1 に近いとき）。

   1. $\mathbb{E}_Q[e^{2\,\mathrm{pmi}}] = \int p^2/q = 1/(1-\rho^2) = e^{2I}$ であり、したがって e^{pmi} が分散 $e^{2I}-1$ を Q のもとで持つことを示せ。**ヒント：**被積分関数は精度行列 $2\Sigma^{-1} - \mathbf{I}$ を持つガウス密度に比例する。ここで Σ はペアの共分散行列であり、$\det(2\Sigma^{-1} - \mathbf{I}) = 1$ が成り立つ。
   2. バッチにランダムな置換を施すと、平均して 1 つの y が元の対応する x と組のまま残る。このような不動点が置換ペアにわたるバッチ平均に e^{pmi}/N（すなわち e^{pmi} の平均への寄与）を加え、その期待値が e^{2I}/N（すなわち 0.43、23、1270。それぞれ $I = 2, 4, 6$ かつ $N = 128$ のとき）になることを示せ。なぜ `np.roll` はこの問題を回避できるのか。
   3. NWJ クリティック $T^* = 1 + \mathrm{pmi}$ に対し、非対応ペアが独立なサンプルであるときバッチ推定値の平均が I となり、その非対応ペアの項の分散が $(e^{2I}-1)/N$（標準偏差はそれぞれ 0.65、4.8、36 nat。ただし $I = 2, 4, 6$ かつ $N = 128$ のとき）になることを示せ。さらに、コードセルのようにずらして作ったペア (x_j, y_{j+1}) でも、隣り合う項どうしが変数を共有しているにもかかわらず、$N \geq 3$ に対して同じ分散になることを示せ。**ヒント：**共有されているペアで条件づけを行う。

2. **ガウスデコーダを用いた Barber–Agakov。**相関 ρ を持つ標準 2 変量ガウス分布のペア（相互情報量は $-\frac{1}{2}\log(1-\rho^2)$）と、デコーダ $q(x \mid y) = \mathcal{N}(x; ay, s^2)$（パラメータは $a \in \mathbb{R}$ および $s^2 > 0$）を考える。下界 Equation 37.75 において、微分エントロピー $h(X) = \frac{1}{2}\log(2\pi e)$ を用いよ。

   1. Equation 37.75 の右辺を a と s^2 の関数として計算せよ。両方のパラメータに関して最大化し、最大値が真の相互情報量に一致することを示せ。
   2. 本文では $s^2 = 1$ に固定し（これは二乗誤差のみで訓練することに相当する）、最良の下界 $\rho^2/2$ を求めた。真の相互情報量とのギャップが、命題の証明どおり $D_{\mathrm{KL}}(\mathcal{N}(0, 1-\rho^2) \,\|\, \mathcal{N}(0,1))$ となることを示せ。$\rho \to 1$ のとき、このギャップはどのように振る舞うか。
   3. $\hat{x}(y)$ を X の任意の予測器とし、その平均二乗誤差を D とする。デコーダ $\mathcal{N}(x; \hat{x}(y), D)$ を用いて $I(X;Y) \geq -\frac{1}{2}\log D$ を示せ。$X \sim \mathcal{N}(0, v)$ のとき、対応する下界はどうなるか。これをガウス分布のレート・歪み関数 $R(D) = \frac{1}{2}\log(v/D)$（$D < v$ のとき。37.6.3.2 節参照。すなわち X の任意の記述が二乗誤差 D を達成するのに必要な最小の情報量）と比較せよ。

3. **コサインクリティックの温度。**SimCLR 型のシステムは候補を $T(x, y) = s(x, y)/\tau$ によってスコア付けする。ここで $s(x, y) = \mathrm{sim}(z_x, z_y) \in [-1, 1]$ はコサイン類似度であり、$\tau > 0$ は温度（ソフトマックス温度）である。1 つのアンカー x に対し、$s_j = s(x, y_j)$（ただし $y_1 = y^+$ が正例）と書き、ℓ を期待値 Equation 37.78 の内部の損失とする。

   1. $\mathcal{L}_{\mathrm{NCE}} \geq \log\left(1 + (N-1)e^{-2/\tau}\right)$ を示し、したがって $\hat{I}_{\mathrm{NCE}} < 2/\tau$ がすべての N に対して成り立つことを導け。上限 $\log N - \log\left(1 + (N-1)e^{-2/\tau}\right)$ を $N = 4096$ かつ $\tau \in \{1, 0.5, 0.1\}$ に対して計算せよ（それぞれ約 2.0、4.0、8.3 nat となるはずである）。バッチサイズではなく温度が下界を制限するのはどのようなときか。

   2. $\partial \ell / \partial s_j = p_j/\tau$ がすべての負例 $j \geq 2$ に対して成り立つことを示せ（ただし p_j は候補 j のソフトマックス確率である）。また、$\partial \ell / \partial s_1$ を p_j を用いて表せ。

3. $\tau\ell \to \max\{0, \max_{j\ge 2} s_j - s_1\}$（$\tau \to 0$ のとき）を示し、さらに次を示せ。

$$\ell = \log N - \frac{1}{\tau}\Big(s_1 - \frac{1}{N}\sum_{j=1}^{N} s_j\Big) + O(\tau^{-2}) \quad \text{as } \tau \to \infty. \tag{37.83}$$

それぞれの極限において、どの負例が勾配を駆動するか (Wang and Liu 2021)。

4. **サンプルからのエントロピー下界。**●37.7.2.4 節では McAllester and Stratos (2020) の定理を非形式的に述べ、エントロピーに対する版を証明した。本問ではその証明を精密化して応用する。サンプル S として、n 個の独立同分布な値を離散分布 p から抽出する。S の**タイプ**は、各 $i \ge 1$ について、ちょうど i 回現れる異なる値が S の中にいくつあるかを記録したものであり、それらがどの値であるかは忘れる。B をタイプの関数であって、あらゆる離散分布に対して下側信頼限界となるもの、すなわち $P_{S\sim p^n}\big(B \le H(p)\big) \ge 0.95$ がすべての p に対して成り立つものとする。また、$\mathcal{T}_0$ を n 個のすべて異なる値からなるサンプルのタイプとする。

   1. 本文では n^2 個の値上の一様分布にこの保証を適用した。任意の整数 $M \ge n^2/1.9$ でも同様に成り立ち、したがって $B(\mathcal{T}_0) \le \log M$ となることを示せ。これにより $2\log n$ からどれだけ改善されるか。
   2. p を $M = 2^{256}$ 個の値上の一様分布とする。n がどれだけ大きくなれば $2\log n$ が $H(p)$ に達するか。この結果は、暗号鍵のエントロピーをサンプルから測定することについて何を物語っているか。

5. **最適な InfoNCE クリティック。**●1 つのアンカー x と 1 つのバッグ $v = (y_1, \dots, y_N)$（ただし $y_1 = y^+$）に対し、$\ell(x, v) = -\log\big(e^{T(x,y_1)} / \sum_{j=1}^{N} e^{T(x,y_j)}\big)$ とおくと、$\mathcal{L}_{\text{NCE}}$（Equation 37.78）はその期待値となる。

   1. $T(x,y)$ を $T(x,y) + c(x)$ に置き換えても（c は任意の関数）、$\mathcal{L}_{\text{NCE}}$ が変化しないことを示せ。
   2. バッグをシャッフルし、正例が一様ランダムな位置 J に置かれ、残りの $N-1$ 個の位置に負例が入るようにする。このとき次が成り立つことを示せ。

$$P(J = i \mid x, y_1, \dots, y_N) = \frac{e^{\text{pmi}(x,y_i)}}{\sum_{j=1}^{N} e^{\text{pmi}(x,y_j)}}. \tag{37.84}$$

   ギブスの不等式を用いて、すべてのクリティック $T(x,y) = \text{pmi}(x,y) + c(x)$ が $\mathcal{L}_{\text{NCE}}$ を最小化することを導け。
   3. このクリティックに対して、次が成り立つことを示せ。

$$\hat{I}_{\text{NCE}} = I(X;Y) - \mathbb{E}\left[\log \frac{1}{N}\sum_{j=1}^{N} e^{\text{pmi}(x,y_j)}\right]. \tag{37.85}$$

   大数の法則を用いて、補正項が $N \to \infty$ で 0 に収束することを（$I(X;Y)$ が有限のとき）論じよ。$\rho = 0.99$ かつ $N = 8$ においてモンテカルロ法でこの補正項を評価し、厳密なクリティックの値（約 1.4）と真の値（1.96）とのギャップ（37.7.4.5 節）と比較せよ。

4. 近似 Equation 37.82 では、平均に対する負例の寄与を 1 に、$\mathrm{pmi}(x, y^+)$ を I に置き換えている。イェンセンの不等式を用いて、第 1 の置き換えは推定値を引き下げ、第 2 の置き換えは推定値を引き上げることを示せ。37.7.4.3 節の表では、どちらの効果が勝っているか。

6. **バッチ内負例と対称損失。** 37.7.4.5 節のコードセルでは、全 $N \times N$ 個のペア（N 個の独立なペア (x_i, y_i) からなるバッチから構成）をスコア付けし、第 i 行の非対角成分をアンカー x_i の負例として用いている。
   1. 各行が InfoNCE の命題の仮定を満たすこと、すなわちその候補 y_j（$j \neq i$）が P_Y からのサンプルであって (x_i, y_i) と独立であることを示せ。ここから、行損失の平均 $\mathcal{L}$ が期待値の意味で $I(X;Y) \geq \log N - \mathcal{L}$ を保証することを導け。
   2. 各行は候補を共有しているため、それぞれの損失は従属である。このことは下界の妥当性、行損失の平均の期待値、あるいはその分散に影響するか。
   3. CLIP は対称損失、すなわち行方向の交差エントロピーと列方向の交差エントロピーの平均を用いる。これも同様に $I(X;Y)$ の下界を与え、同じ上限 $\log N$ を持つことを示せ。
   4. 同時クリティック $T(x, y)$（連結したペアにネットワークを適用するもの）はバッチのスコア計算に N^2 回のネットワーク評価を要するのに対し、分離可能なクリティック $g(x)^\top h(y)$ は $2N$ 回で済む。CLIP のバッチサイズ $N = 32{,}768$ に対して両方の回数を計算せよ。分離可能なクリティックはペアのあらゆる関数を表現できるわけではない。ガウス分布のペアの pmi は 3 次元の埋め込みで分離可能に表せること、そして InfoNCE には 2 次元で十分であることを示せ。

7. **シフトされた PMI としてのネガティブサンプリング。●**word2vec の訓練手法であるネガティブサンプリング付き skip-gram は、中心語 w と文脈語 o をそれぞれのベクトルの内積 $s(w, o) = \mathbf{u}_o^\top \mathbf{v}_w$ によってスコア付けする。そして、コーパス内に現れるペアと、文脈語が K 個のノイズ語（分布 P からサンプリング）の 1 つであるペアとを識別するよう学習する。$p(w, o)$ をコーパス内の全（中心語、文脈語）ペアにおけるそのペアの相対頻度とし、周辺分布を $p(w)$ および $p(o)$ とする。コーパス全体で平均すると、損失は次のようになる。

$$\ell = -\sum_{w,o} p(w,o) \log \sigma(s(w,o)) - K \sum_{w} p(w) \sum_{o} P(o) \log \sigma(-s(w,o)), \quad (37.86)$$

ここで $\sigma(s) = 1/(1 + e^{-s})$ はシグモイド関数である。埋め込みは十分な次元を持ち、すべてのスコア $s(w, o)$ を独立に選べると仮定せよ。

   1. ℓ が $s^*(w,o) = \log \frac{p(w,o)}{p(w)\,P(o)} - \log K$ で最小化され、これが $\mathrm{pmi}(w,o) - \log K$ に等しくなること（P がユニグラム分布 $p(o)$ のとき）を示せ (Levy and Goldberg 2014)。
   2. word2vec は出現頻度の 0.75 乗に比例する確率でノイズ語をサンプリングする。このとき s^* が $\mathrm{pmi}(w,o) - \log K$ から o のみに依存する項だけずれることを示せ。この項が負になるのはどのような文脈語に対してか。また、「sri lanka」（演習問題 1、37.6 節参照）のような低頻度語のペアに対してこの項はどう作用するか。
   3. s^* を演習問題 5b の最適な InfoNCE クリティックと比較せよ。InfoNCE では自由なまま残される加法項を、ネガティブサンプリングの二値分類が固定するのはなぜか。

8. [extended] **小規模ネットワークにおける圧縮フェーズ。●●**Shwartz-Ziv and Tishby (2017) は、幅 12-10-7-5-4-3-2 の全結合 tanh ネットワークを、4096 通りの 2 進入力パターン（12 ビット）上で調べた。各隠れ層 Z について、すべてのユニットの活性化値を 30 個の等幅区間（$[-1, 1]$ の分割）にビン分割し、ビン分割した値から $I(X;Z)$ と $I(Y;Z)$ を計算した。著者らが報告した圧縮フェーズをめぐる論争については 37.7.2.5 節で述べたとおりである。

1. X を 4096 通りのパターン上の一様分布とし、Z_{bin} を決定論的ネットワークのビン分割された層とする。$I(X;Z)$ のビン分割推定値が $H(Z_{\text{bin}})$ に等しく、決して $\ln 4096 \approx 8.32$ nat を超えないことを示せ。訓練中にこの値が減少するのはどのようなときか。
2. 確率ベクトル $\mathbf{w} \in \mathbb{R}^{12}$ を 1 つ固定し、$Y = 1$ となる条件を $\mathbf{w}^{\top}\mathbf{x}$ が全 4096 パターンにわたる中央値を超えるときと定める。2 つの出力ユニット上にソフトマックスを置いた tanh ネットワークを、ランダムに選んだパターンの 85% 上で交差エントロピー損失を用いた確率的勾配降下法（SGD）により訓練せよ。訓練を通じて、すべての隠れ層の情報平面 $\big(I(X;Z), I(Y;Z)\big)$ における軌跡をプロットせよ。
3. 隠れユニットを ReLU に変えて (b) を繰り返せ。ただし各ユニットは 30 個の等幅区間（0 とその実行全体での最大の活性化値との間）にビン分割するものとする。
4. 訓練中に $|z| > 0.99$ となる tanh ユニットの割合を記録せよ。(b) におけるビン分割した $I(X;Z)$ の減少は、この割合に連動しているか。(c) で活性化関数を変更してもなお維持される圧縮についての結論は何か (Saxe et al. 2018)。

37.8 関連文献

本章の主要な節では、明快に証明でき、かつ小さな例で確認できる結果に対象を絞ってきた。以下の各小項では、それぞれの節の議論を引き継ぎ、本筋には収まらなかった周辺の考え方を取り上げる。時間がない場合は、日々の実践で特に直接現れる 3 つのトピック、すなわちモデルの較正（calibration）、スケーリング則（scaling law）における削減不能損失の定数、および相互情報量（mutual information）推定量のバイアス・分散のトレードオフから読み始めるとよい。ビッツバック符号化（bits-back coding）も、その仕掛け自体がきわめて巧妙であるため一読の価値がある。通信路容量（channel capacity）と情報幾何学（information geometry）は後回しにしても差し支えなく、本書の以降の箇所がこのページの内容に依存することはない。

37.8.1 エントロピーと損失

37.8.1.1 訓練後の較正

原理的には、母集団の対数損失（log-loss）を最小化すれば真の条件付きクラス確率が復元され（37.1.3.1 節）、予測は自動的に較正される。しかし、訓練済みのニューラルネットワークがその最適解に落ち着くことはまれである。Guo et al. (2017) が指摘したことで知られるように、現代のネットワークは特定の方向に偏って誤る、すなわち過信（overconfidence）に陥る傾向がある。その主な原因はワンホットの訓練ラベルにある。ワンホットラベルのもとでは、最大ロジットを他のロジットより無限に大きく引き離すほど損失が下がるため（37.1 節）、ネットワークが訓練セットを完全に分類できるようになった後は、ロジットどうしの差をさらに広げ続けることでしか損失を下げられない。訓練の終盤において同著者らは、ホールドアウトの分類誤差が横ばいのままであるにもかかわらず、ホールドアウトの対数損失が上昇し始めることを観察している。ネットワークは意思決定ではなく確率推定に対して過学習しているのである。温度スケーリング（temperature scaling、3.4 節）は、ホールドアウトの検証セット上で当てはめた単一のスカラーによって、この問題の大部分を解消する。これは蒸留（distillation）で用いられるものと同じ温度パラメータであり、ここでは過信した予測を軟化させる目的で転用されている。

このような事後的な再較正（post-hoc recalibration）はどこまで有効だろうか。$\bar{\mathbf{p}}(\mathbf{q})$ を、モデルが確率ベクトル $\mathbf{q}$ を出力するすべての入力における実際の経験クラス頻度とする。任意の交差エントロピーはエントロピーと KL ダイバージェンスの和に分解されるため、予測ベクトル $\mathbf{q}$ ごとに条件付きでこの恒等式を適用すると、ホールドアウトの対数損失は次の 2 つの項に分解される。

$$\mathbb{E}\big[-\log q_{y}\big] = \mathbb{E}_{\mathbf{q}}\big[D_{\mathrm{KL}}\big(\bar{\mathbf{p}}(\mathbf{q}) \,\|\, \mathbf{q}\big)\big] + \mathbb{E}_{\mathbf{q}}\big[H\big(\bar{\mathbf{p}}(\mathbf{q})\big)\big].$$

第 1 項は較正のずれ（miscalibration）を nat 単位で測るものであり、モデルの予測確率が主張するとおりの頻度で実際に事象が起こるとき、かつそのときに限り 0 になる。第 2 項は、モデルの予測を知った後でもなお真のラベルについて残る不確実性である。予測確率を事後的にどのように再写像しても、第 1 項を 0 に近づけることはできるが、第 2 項を減らすことは決してできない。温度スケーリングはこのような再写像の 1 パラメータ族にほかならず、それゆえきわめて簡潔であるが、一部のクラスでは過信し別のクラスでは過小確信（underconfidence）に陥っているモデルを 1 つのスカラーで修正することはできない。より一般に、あらゆる固有スコアリングルール（proper scoring rule）は、ダイバージェンス項とエントロピー項への同様の分解をもつ (Gneiting and Raftery 2007)。

37.8.1.2 参照に対する相対的なエントロピー

37.1.4.1 節では、微分エントロピー（differential entropy）の厄介な性質を見た。単位をメートルからセンチメートルに変えるだけで、値が $\ln 100$ だけ増加してしまうのである。これに対し、KL ダイバージェンスはまったく変化しない。比の内部にある両方の密度が同じ係数でスケールされるからである。参照密度 m に対して測り、$-\int p \log(p/m)\, dx = -D_{\mathrm{KL}}(p \| m)$ と定義すれば、エントロピーにも同じ座標不変性を与えることができる。通常の微分エントロピー h は、たまたま選んだ単位系における $m = 1$ という特別な場合にすぎない。1 メートルあたり 1 の一様密度と 1 センチメートルあたり 1 の一様密度は同じ測度ではないため、先の $\ln 100$ のずれの正体は隠れた参照測度の変更だったのである。Jaynes (1968) は、次第に細かくなるグリッド上の離散エントロピーの極限としてこの相対形を導出しており、これを用いると最大エントロピー原理（maximum-entropy principle）は「制約が許す限り参照測度の近くにとどまれ」という自然な形をとる。符号の違いを除けば、変分オートエンコーダ（variational autoencoder、37.6.4.3 節）における KL ペナルティは事前分布（prior）が m の役割を果たすこの種のエントロピーであり、潜在空間（latent space）の滑らかな再パラメータ化のもとで不変に保たれる理由もそこにある。

37.8.2 符号と圧縮

37.8.2.1 モデルを用いない圧縮

37.2 節で扱ったすべての圧縮器は、次の記号を予測するために明示的な確率モデルに依拠している。レンペル・ジブ（Lempel–Ziv）族はその要件を完全に不要にする。LZ77 (Ziv and Lempel 1977) は繰り返し現れる任意の部分文字列をそれ以前の出現箇所を指す短いポインタに置き換え、LZ78 (Ziv and Lempel 1978) はファイルを走査しながらフレーズの辞書をオンザフライで構築していく。注目すべきことに、両アルゴリズムは**普遍**（universal）である。すなわち、任意の定常エルゴード情報源（stationary ergodic source、大まかに言えば 1 つの長いサンプルから長期的な統計的性質が明らかになる情報源）に対し、情報源に関する事前知識を一切与えなくても、ファイルが大きくなるにつれて平均符号長が真のエントロピー率（entropy rate）へと近づく。もっとも実際には、その極限への収束は遅い。『The Time Machine』に対して、（LZ77 とハフマン符号化を組み合わせた）`zlib` は 1 文字あたり 2.84 ビットを要するのに対し、直前の 3 文字から各文字を予測する 37.2.2.2 節の単純なカウントモデルは、すでに 2.55 ビットまで到達している。

37.8.2.2 ビッツバック符号化

変分オートエンコーダのような潜在変数モデル（latent-variable model）では、37.2.2.1 節の算術符号器を直接駆動することはできない。符号器には厳密な周辺確率（marginal probability）$p(x)$ が必要であるのに対し、モデルが与えるのは ELBO（36.7 節）のみであり、これは $\log p(x)$ に対する、近似事後分布（approximate posterior）$q(z \mid x)$ から構成された下界だからである。驚くべきことに、圧縮ストリームのデコードは数学的にはサンプリングと同じであるという巧妙な双対性（duality）のおかげで、この下界もなお達成可能なファイル長に対応する。$q(z \mid x)$ 向

けに設計されたデコーダにランダムなビット列を入力すると、デコーダは約 $\log_2 1/q(z \mid x)$ ビットを消費し、潜在サンプル z を q からの抽出結果として出力する。

ビッツバック符号化は、この知見を実用的な方式へと昇華させる。送信者がそれ以前のデータ点を収めた圧縮ファイルをすでに持っているとする。優れた圧縮は冗長性を取り除くため、そのファイル末尾のビット列はランダムなコイン投げと区別がつかないように見える。新しいデータ点 x を符号化するため、送信者はファイル末尾から $\log_2 1/q(z \mid x)$ ビットを借り受け、それを $q(z \mid x)$ で「デコード」して潜在変数 z を抽出する。次に送信者は、x を $p(x \mid z)$ で符号化し、z を事前分布 $p(z)$ で符号化して、その両方をファイルに追記する。x を展開するため、受信者はこの手順を逆にたどる。まず z を事前分布からデコードし、次に x を $p(x \mid z)$ からデコードしたうえで、z を $q(z \mid x)$ によって**再符号化**し、借りていたビット列を元の場所へとそっくり戻す。全体として、ファイルは $\log_2 1/p(x \mid z) + \log_2 1/p(z)$ ビットだけ増え、$\log_2 1/q(z \mid x)$ ビットだけ縮むため、q のもとでの正味の期待増加量は負の ELBO となる。受信者は送信者の操作を後入れ先出し（last-in, first-out）の順序で取り消さなければならないため、基盤となる符号器はスタックのように振る舞う必要がある。ビッツバックの議論自体は Wallace (1990) および Hinton and Camp (1993) に遡り、Frey and Hinton (1997) はキューとして振る舞う従来の算術符号化（arithmetic coding）の上に最初の動作する符号器を構築した。非対称記数系（asymmetric numeral systems）(Duda 2013) は本来的にスタックベースであり、これを用いて Townsend et al. (2019) は、小規模な変分オートエンコーダで MNIST をその負の ELBO のすぐ近くのレートまで圧縮できることを実証した。

37.8.2.3 2 部符号を超える記述長

37.2.2.3 節の 2 部符号（two-part code）は、選択した精度でモデルパラメータを明示的に符号化するため、総記述長（description length）がパラメータ空間をどれだけ細かく離散化するかに敏感になる。**逐次予測**（prequential）符号は、パラメータの送信を一切行わない。その代わり、37.2.2.2 節の適応カウントモデルとちょうど同じように、それまでに観測したデータ点に当てはめたモデルを用いて各データ点を符号化するため、モデルの学習コストはデータの符号長に直接吸収される。Blier and Ollivier (2018) は逐次予測符号化を用いて深層ニューラルネットワークの訓練ラベルを圧縮し、2 部符号や変分符号（ネットワークの重みにビッツバック符号化を適用する手法）よりもはるかに短い符号を得た。大幅に過剰パラメータ化されたネットワークであっても、この尺度で測ればラベルを確かに圧縮しているのである。Grünwald (2007) は、逐次予測符号と正規化最尤法（normalized maximum likelihood、事後的に選んだクラス最良のモデルに対する最悪ケースのリグレットを最小化する符号）の双方を詳しく展開している。この圧縮の視点を理論的な極限まで推し進めると、データを出力する最短のコンピュータプログラムの長さであるコルモゴロフ複雑性（Kolmogorov complexity）へと至る。これは以上の考え方をすべて統一する概念であるが、計算不能であることでよく知られている (Li and Vitányi 2019)。

37.8.2.4 スケーリング則の定数

11.6 節では、多様なスケールにわたる言語モデルのホールドアウト損失を、定数 E と 2 つのべき乗則（power law）の項（一方はパラメータ数に応じて減衰し、もう一方は訓練トークン数に応じて減衰する項）の和としてモデル化している (Hoffmann et al. 2022)。E を自然テキストの真のエントロピーと読みたくなるかもしれないし、実際、最良の場合には E はトークンあたりのエントロピー率（37.2.3.1 節）、すなわちいかなる期待損失も下回ることのできない下限に対応する。しかし実際には、E はそれよりはるかに上にある損失から外挿して当てはめられた回帰定数にすぎず、エントロピー率の直接的な測定値ではない。先行研究である Kaplan et al. (2020) は加法定数を一切含まない同様のスケーリング曲線を当てはめていたが、損失はエントロピーの下限を下回れないため、この近似は十分に大きなスケールではいずれ破綻せざるを得ない。また、E がトークン単位で測られている点にも留意すべきである。異なるトークナイザのもとで当てはめられ

た定数は、双方をバイトあたりビット数（8.3.2.4 節）へ変換しない限り直接比較することはできない。

37.8.3 分布の比較

37.8.3.1 レーニイダイバージェンスとプライバシー

乱択アルゴリズムを繰り返し実行したとき、1 件の訓練レコードについてどれだけの情報が漏洩するだろうか。差分プライバシー（differential privacy）は、その 1 件のレコードだけが異なる 2 つのデータセット上でのアルゴリズムの出力分布間のダイバージェンスとして、この情報漏洩を定式化する。差分プライバシー付き SGD は各勾配ステップでノイズを注入し何千回もの反復を実行するため、独立なステップにわたってきれいに合成できるダイバージェンスが必要になる。次数 α のレーニイダイバージェンス（Rényi divergence）(Rényi 1961) は次の式で定義される。

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \sum_x p(x)^\alpha q(x)^{1-\alpha},$$

このダイバージェンスはその性質を備えている。f ダイバージェンスが和の内側で凸関数を適用するのに対し、ここでは対数が和の**外側**にあることがポイントである。独立な座標にわたって和が因数分解されるため、対数によってその積が和へと変換され、結果として D_α は KL ダイバージェンスと同様に、また χ^2 ダイバージェンスや全変動距離（total variation distance）とは異なり、独立なステップにわたってきれいに加法的に合成される。レーニイ差分プライバシー（Rényi differential privacy）(Mironov 2017) は、ノイズを加えた 1 ステップあたりの D_α を上から抑え、その上界を訓練実行全体にわたって足し合わせる。KL ダイバージェンスもステップ間で加算できるが、対数密度比 $\log(p(x)/q(x))$ を平均の意味でしか制御できない。一方、D_α は α に関して単調に増加し、$\alpha \to 1$ での KL ダイバージェンスから $\alpha \to \infty$ での最悪ケースの対数比までを補間するため、より大きな次数 α を選べば、まれに起こる情報漏洩の大きい出力に対しても保護を与えられる。これらの極限と不等式の網羅的な整理は Erven and Harremoës (2014) にまとめられており、演習問題 1d（37.3 節）では D_α と α ダイバージェンスの関係を扱う。

37.8.3.2 情報幾何学

37.3 節では簡単なテイラー展開により、$q_{\theta+\mathbf{h}}$ と q_θ の間の任意の滑らかな f ダイバージェンスが、2 次の精度でフィッシャー情報行列（Fisher information matrix）に関する二次形式 $\frac{1}{2} f''(1)\, \mathbf{h}^\top \mathbf{I}(\theta)\, \mathbf{h}$ に帰着することを見た。これは一見すると代数的な偶然のように思えるかもしれないが、チェンツォフの定理（Chentsov's theorem）は、それ以外の結果にはなり得なかったことを示している。正定値二次形式はあらゆる微小ステップ $\mathbf{h}$ に長さを割り当て、分布の空間をリーマン多様体（Riemannian manifold）へと変える（37.3.1.1 節の距離が 2 つの分布間の大域的な距離を測るのに対し、こちらは局所的なステップを測る）。有限集合上の分布について、Chentsov は、生のデータを十分統計量（sufficient statistic）に置き換えたときに不変に保たれるリーマン計量（Riemannian metric）は、正の定数倍を除いてフィッシャー情報量**のみ**であることを証明した (Čencov 1982)。あらゆる f ダイバージェンスはこの不変性を共有している。θ を知らなくても十分統計量からデータを再サンプリングできるため、37.3.3.3 節のデータ処理不等式（data-processing inequality）が両方向に適用され、等号が強制されるからである。したがって、任意の滑らかな f ダイバージェンスの局所二次近似は、フィッシャー情報量の定数倍でなければならない。この幾何構造のもとで長さを測った勾配ステップを踏むと自然勾配（natural gradient）が得られ、K-FAC はこれを層あたり 1 つのクロネッカー積（Kronecker product）で近似する（35.5.5 節）。また、15.2 節の KL 信頼領域（trust region）も、2 次の精度ではまさにこの計量における球である。Amari (2016) はこの幾何学的視点を全面的に展開し、最尤法と EM ア

ルゴリズム（36.7 節）の両ステップがいかに直交射影（orthogonal projection）として理解できるかを示している。

37.8.3.3 シンクホーンダイバージェンスとスライスワッサースタイン距離

最適輸送距離を高次元ミニバッチ上の実用的な損失へと変える際には、計算コストと統計的バイアス（statistical bias）という 2 つの障壁に直面する。エントロピー正則化（entropic regularization、37.4.2.4 節）は計算上の問題を解決するものの、微妙なバイアスを持ち込む。ある分布からそれ自身への正則化輸送コストが厳密に非ゼロとなるため、これを最小化するよう訓練されたモデルはデータよりも狭い分布へと収縮しやすいのである。演習問題 5（37.4 節）のシンクホーンダイバージェンス（Sinkhorn divergence）は、2 つの自己輸送コストの平均を差し引くことでこの問題を解消する。その演習問題の先には、これを信頼できる訓練目的関数たらしめる定理がある。有界領域上でユークリッド距離またはその二乗を基礎コスト（ground cost）として用いると、シンクホーンダイバージェンスは非負であり、2 つの分布が一致するとき、かつそのときに限り 0 となり、さらにモデルがデータへと分布収束するとき、かつそのときに限り 0 に収束する (Feydy et al. 2019)。

スライスワッサースタイン距離（sliced Wasserstein distance）は、まったく異なる道筋をとる。2 つの高次元サンプルを 1 本の 1 次元直線上に射影すれば、最適輸送は射影された値を単にソートするだけの操作に帰着する（37.4.2 節）。この 1 次元のコストを多数のランダムな方向にわたって平均したものが、スライスワッサースタイン距離である (Bonneel et al. 2015)。ただし高次元における難点は、2 つの分布が実際に異なっている部分空間にランダムな方向が重なることがめったにない点にある。これを回避するため、Deshpande et al. (2018) は識別器（discriminator）を訓練して有用な特徴量を抽出し、生のピクセルではなくその学習された特徴空間上でスライスワッサースタイン距離を計算している。

37.8.3.4 Wasserstein GAN におけるリプシッツクリティック

37.4.2 節のカントロヴィチ・ルビンシュタイン双対（Kantorovich–Rubinstein dual）はリプシッツ定数 1 の関数（1-Lipschitz function）全体にわたって最適化を行うが、通常のニューラルネットワークはそのままではこの制約を満たさない。16.2 節ではこれを課す 3 つの主要な手法（重みクリッピング、スペクトル正規化、勾配ペナルティ）を比較し、16.4 節では勾配ペナルティを詳しく調べている。とりわけ Gulrajani et al. (2017) のペナルティは、実サンプルと生成サンプルの間を補間したランダムな点においてクリティックの勾配ノルムを 1 に近づけるものであり、これは対応づけられた点を結ぶ輸送半直線（transport ray）に沿って最適クリティックが傾き 1 をもつという幾何学的事実を反映している（演習問題 3d、37.4 節）。

37.8.3.5 スタイン変分勾配降下法

分布 Q がそのスコア s_Q を通じてのみ既知である場合、そこからサンプルを抽出するには通常、長い確率的ランダムウォークを実行する必要がある（37.5 節）。37.5.3 節では、カーネルスタイン不一致度（kernel Stein discrepancy）が、サンプル分布 P がそのようなモデル Q からどれだけ離れているかを測ることを見た。スタイン変分勾配降下法（Stein variational gradient descent）(Liu and Wang 2016) は、この不一致度を、粒子の集合を Q へ向かって能動的に輸送する決定論的アルゴリズムへと変換する。分布 P に従う任意の初期点群から出発し、各粒子 y は次の速度場（velocity field）に沿って微小ステップで移動する。

$$v(y) = \mathbb{E}_{x\sim P}\big[k(x,y)\, s_Q(x) + \nabla_x k(x,y)\big].$$

期待値の内側にある 2 つの項は相補的な役割を果たす。第 1 項は y の近傍におけるスコアのカーネル重み付き平均であり、確率密度のより高い領域へと粒子を上り方向に導く。第 2 項は y を近

傍の粒子から押し離す斥力であり、ランジュバン動力学におけるランダムノイズの役割を引き継いでいる。これがなければ、すべての粒子が単一のモードへと潰れてしまうだろう。37.5.3 節の再生核ヒルベルト空間の単位球へと正規化すると、この速度場は $D_{\mathrm{KL}}(P\|Q)$ に対する最急降下方向となり、KL ダイバージェンスを $\|v\| = \mathrm{KSD}(P, Q)$ の速さで減少させる。驚くべきことに、同じスタイン作用素（Stein operator）が適合度検定（goodness-of-fit test）と粒子輸送則の両方の役割を果たしているのである。依然として多数の微小ステップは必要であるものの、更新は決定論的であり粒子どうしが協調的に広がるため、元論文における小規模ニューラルネットワークの重みのように、わずか数十個の粒子で事後分布を近似する用途にこの手法は適している。

37.8.4 相互情報量

37.8.4.1 容量に迫る符号

シャノンの通信路符号化定理（Shannon's channel coding theorem、37.6.3.1 節）は、通信路容量未満のあらゆるレートで高信頼な通信が可能であることを保証するが、Shannon の証明はデコードに指数時間を要するランダム符号帳（random codebook）に依拠している。その存在定理と実用的なハードウェアとの隔たりを埋めるには数十年を要した。疎な因子グラフ（factor graph）上でメッセージを伝播させることで効率的にデコードされる低密度パリティ検査（low-density parity-check、LDPC）符号は、実用上きわめて容量に近い性能を達成する (Richardson and Urbanke 2008)。ポーラ符号（polar code）(Arıkan 2009) はさらに一歩進み、対称 2 元入力通信路において容量を達成することが証明された初の明示的な符号族であり、符号化とデコードの双方が $O(n \log n)$ 回の演算で動作する（n はブロック長）。現代の 5G 無線規格では、これら両方の符号族が実用化されている。

37.8.4.2 学習型非可逆圧縮

37.6.3.2 節では、ガウス情報源に対する厳密なレート歪み（rate–distortion）曲線を導出した。自然写真に対しては閉じた形のレート歪み関数が知られていないため、学習型の画像コーデックはいずれも、その未知のフロンティアより上方に位置する実用的な動作点を表すことになる。Ballé et al. (2017) は、エンコーダ、デコーダ、および量子化された潜在表現上の学習された確率分布を定義する**エントロピーモデル**という 3 つの構成要素を同時に訓練する。丸め操作はほとんど至るところで勾配が 0 になるため、訓練中は量子化を加法的な一様ノイズに置き換える。その目的関数は $R + \lambda D$ を最小化する。ここで、レート R はエントロピーモデルのもとでの潜在変数の算術符号化長であり（37.2 節）、D はデコードされた画像の再構成歪みである。λ を変化させると傾き $-\lambda$ の点が選ばれ、コーデックのレート歪み曲線が描かれる。後続研究において Ballé et al. (2018) は、**超事前分布**（hyperprior）と呼ばれる第 2 のより小さな潜在変数でエントロピーモデルを条件づけることにより、画像の複雑さの空間的な変動を捉え、同じ視覚品質のもとでビットレートを大幅に引き下げている。

37.8.5 相互情報量の推定

37.8.5.1 上界

37.7 節で導出した変分推定量は、いずれも相互情報量の**下界**である。これは共有情報量を最大化したいときには有用だが、本来破棄すべき機微属性や保護属性（protected attribute）に関する情報を学習表現が**ほとんど**保持していないことを保証するには使えない。真の上界を得るには、確率的エンコーダが提供する条件付き密度 $p(z \mid x)$ を利用できる必要がある。変分情報ボトルネック（variational information bottleneck、37.6.4.3 節）はそのような上界の 1 つであり、固定された周辺分布の近似 $r(z)$ を用いる。対照対数比上界（contrastive log-ratio upper bound、CLUB）(Cheng et al. 2020) は $r(z)$ の必要性を完全に取り除く手法であり、ミニバッチ内の対応ペアにわたる $\log p(z \mid x)$ の平均から、非対応ペアにわたる平均を差し引く。$p(z \mid x)$ が真の

エンコーダ密度であるとき、CLUB は期待値の意味で上界となる。しかし、表現と保護属性の間には解析的な条件付き密度が存在しないため、代わりにニューラルネットワークを当てはめなければならない。その場合、当てはめた条件付き分布が周辺分布の積よりも真の同時分布（joint distribution）に対して KL ダイバージェンスの意味で近い場合に限り上界が成り立つが、この条件はそもそも未知であった当の密度なしには検証できない。

37.8.5.2 下界のバイアスと分散

厳密に最適なクリティックが与えられた場合でさえ、変分下界は鋭いトレードオフに直面する。NWJ は平均的には不偏であるが高い分散に悩まされるのに対し、InfoNCE は低分散である代わりに $\log N$ で頭打ちになる（37.7.4.4 節）。Poole et al. (2019) は、これら 2 つの下界の間を滑らかに補間する単一の混合パラメータを導入し、このトレードオフを連続的に調整できるようにしている。さらに深刻なことに、J. Song and Ermon (2020) は、分散の爆発が特定の目的関数に固有の副産物ではなく本質的な現象であることを証明した。すなわち、NWJ と Donsker–Varadhan の両下界において、たとえ厳密なクリティックを用いても、推定量の分散は真の相互情報量に対して指数関数的に増大する。同著者らが提案する対策はクリティックの密度比をクリップするものであり、統計的バイアスが生じる代償として分散を抑え込む。要するに、利用可能なあらゆる対策は、誤差を解消するのではなく分散とバイアスの間で移し替えているにすぎない。

実データでは正解の相互情報量がわかっていることはまれであるため、同著者らは既知の答えを必要としない自己無撞着性による妥当性確認（self-consistency sanity check）を提案している。信頼できる推定量であれば、一方の変数にその決定論的関数を付加しても推定値は変化せず、単一の対 (X, Y) をその独立な 2 つのコピーに置き換えれば推定値は 2 倍になるはずである。画像データセット上では、同著者らが評価した標準的な推定量はいずれもこれらの基本的なテストに不合格となっている。実践上の経験則として、数 nat 規模の相互情報量の推定値が報告されている場合には必ずバッチサイズの対数と比較し、手元のデータでこれらの整合性チェックを通過することを確認したうえで初めて推定量を信頼すべきである。

37.8.5.3 ノイズ付きネットワークとボトルネックの視点

37.7.2.5 節の圧縮論争は、Tishby and Zaslavsky (2015) がニューラルネットワークの各層を情報ボトルネックのフロンティア（37.6.4 節）上にプロットすることを提案したことに端を発する。このフロンティアは、達成可能な最大の $I(Y;Z)$ を与えられた $I(X;Z)$ ごとに追跡するものである。しかしその項で見たとおり、この着想は数学的な難点に突き当たる。連続入力を受け取る決定論的ネットワークでは $I(X;Z)$ が形式的に無限大となり、追跡すべき有限の圧縮曲線が存在しなくなるのである。Goldfeld et al. (2019) は、隠れ層の活性化値に小さなガウスノイズを注入して $I(X;Z)$ を有限にすることで、この問いを良設定（well-posed）にした。このノイズ付きモデルのもとでは実際に訓練中に $I(X;Z)$ が減少し得るが、著者らはその減少の要因が、同一クラスの隠れ表現どうしが互いに引き寄せられる幾何学的なクラスタリングにあることを突き止めている。以前から決定論的ネットワークに対するビン分割ベースの相互情報量推定が捉えていた正体も、まさにこのクラスタリングにほかならないと同著者らは論じている。

37.8.6 情報理論全般の参考資料

書籍

- Elements of Information Theory — Cover & Thomas：エントロピー、KL ダイバージェンス（同書では相対エントロピーと呼ばれる）、相互情報量、およびそれらの連鎖律を第一原理から展開するとともに、情報源符号化（source coding）、通信路容量、レート歪み理論を扱う大学院レベルの古典的標準書である。37.1 節、37.2 節、37.6 節に対応する。

- Information Theory, Inference, and Learning Algorithms — David MacKay：オンラインで無料公開されている書籍である。全編を通じて情報理論と機械学習を織り交ぜて論じており、37.2 節に対応する記号符号（symbol code）とストリーム符号（算術符号化を含む）の章を備える。
- Information Theory: From Coding to Learning — Polyanskiy & Wu：統計学と機械学習に軸足を置いた現代的な大学院向けテキストであり、f ダイバージェンス、変分表現、データ処理の扱いは 37.3 節および 37.6 節に対応する。同書の母体となった MIT 6.441 の講義ノートも無料公開されている。
- Computational Optimal Transport — Peyré & Cuturi：データサイエンス向けにワッサースタイン距離、エントロピー正則化、シンクホーン反復を解説しており、37.4 節の最適輸送パートを補完する。全文は arXiv でも公開されている。
- The Minimum Description Length Principle — Peter Grünwald：2 部符号と普遍符号を通じて統計学習をデータ圧縮として展開し、37.2.2.3 節の内容を掘り下げている。中心的な考え方は無料公開されているチュートリアルでも解説されている。

講義・チュートリアル・ソフトウェア

- Stanford EE376A Information Theory — Tsachy Weissman：エントロピー、接頭符号（prefix code）、典型集合（typical set）を扱うわかりやすい講義であり、学生が筆記した公開講義ノートが提供されている。37.1 節および 37.2 節に対応する。
- Visual Information Theory — Christopher Olah：符号語長から出発してエントロピー、交差エントロピー、相互情報量へと積み上げていく直観的かつ視覚的な入門記事であり、37.1 節、37.2 節、37.6 節に対応する。
- From GAN to WGAN — Lilian Weng：生成モデルにおける KL、イェンセン・シャノン、ワッサースタインの各損失を比較する明快な解説であり、37.3 節および 37.4 節と並行した内容になっている。
- POT: Python Optimal Transport — Flamary et al.：厳密な最適輸送とエントロピー正則化付きシンクホーン反復の両方のソルバを提供するオープンソースの Python ライブラリであり、手元のデータセット上で 37.4.2 節の距離を計算するのに役立つ。
- Generative Modeling by Estimating Gradients of the Data Distribution — Yang Song：スコア関数がなぜ正規化定数（normalizing constant）を回避できるのか、またスコアマッチングがいかにしてスコア関数をデータに当てはめるのか（37.5 節の範囲）を平易に解説したうえで、サンプリングと拡散モデル（diffusion model）へと進むチュートリアルである。

基礎論文

- Language Modeling Is Compression — Delétang et al.：算術符号器と言語モデルの次トークン確率を組み合わせる研究であり、まさに 37.2.2.2 節で実演した着想そのものを扱っている。
- f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization — Nowozin et al.：37.3.4 節で導出したとおり、任意の f ダイバージェンスがそのフェンシェル共役（Fenchel conjugate）による下界を通じて敵対的訓練（adversarial training）の目的関数を与えることを示している。
- A Kernel Two-Sample Test — Gretton et al.：37.4.1 節で扱った最大平均値差異（maximum mean discrepancy）、その不偏 2 サンプル推定量、およびそれに基づく仮説検定を導入した論文である。
- Wasserstein GAN — Arjovsky et al.：サポートが重ならない場合でも勾配が有用な情報を保つよう、イェンセン・シャノン・ダイバージェンスを 1 次ワッサースタイン距離に置き換えた研究であり、37.4.2 節に対応する。
- Estimation of Non-Normalized Statistical Models by Score Matching — Aapo Hyvärinen：対数密度の勾配をデータの勾配に一致させ、未知のデータスコアを部分積分（inte-

gration by parts）によって消去することで非正規化モデルを当てはめる手法を提案しており、これらが 37.5 節の 2 つの重要なステップとなっている。

- A Kernelized Stein Discrepancy for Goodness-of-fit Tests — Liu, Lee & Jordan：37.5.3 節で示したように、ペアごとのカーネル評価を用いて、スコアのみを通じて知られるモデルからサンプルが得られたかどうかを検定する。
- The Information Bottleneck Method — Tishby, Pereira & Bialek：X を圧縮しつつ Y に関する情報を保持するという問題を定式化しており、37.6.4 節で学んだ枠組みの原典である。
- Estimating Mutual Information — Kraskov, Stögbauer & Grassberger：最近傍距離から連続変数の相互情報量を推定する 37.7.2.2 節の推定量を提案した論文であり、scikit-learn の `mutual_info_regression` もこれに基づいている。
- On Variational Bounds of Mutual Information — Poole et al.：Barber–Agakov、Donsker–Varadhan、NWJ、InfoNCE の各下界を単一の枠組みで統一し、それらのバイアス・分散のトレードオフを解析しており、37.7.3 節に対応する。
- Representation Learning with Contrastive Predictive Coding — van den Oord et al.：37.7.4 節で展開したとおり、自己教師あり表現学習（self-supervised representation learning）に向けて InfoNCE 損失を相互情報量の下界として導入した論文である。

Chapter 38

ダイナミクス：微分方程式と生成フロー

残差ネットワーク（residual network）は層ごとに入力へ小さな更新を加え、勾配降下法は重みをステップごとに下り方向へ少しずつ動かし、拡散モデル（diffusion model）は学習されたベクトル場（vector field）に沿ってノイズを画像へと変換する。これら 3 つはいずれも、状態が変化する速さを定める規則である微分方程式（differential equation）を離散化したものである。ここからただちに 2 つの実践的な問いが生じる。サンプラはどれだけ少ないステップ数で実行できるのか。また、画像の例だけから速度やスコアの方向をどのように学習できるのだろうか。

訓練画像にノイズを徐々に加えていくことで、2 つ目の問いに逆方向から答えられる。ノイズを帯びた個々の粒子は予測不能にさまようが、粒子群全体の密度は決定論的に時間発展し、その時間発展を逆転させればガウスノイズをデータへと戻せる。これまでの章では、この仕組みまでは踏み込んでいない。誤差逆伝播法（backpropagation）は ODE の解ではなく固定された計算グラフを微分するものであり、静的な確率では移動する密度を追跡できず、$\sqrt{\Delta t}$ オーダーのブラウン増分は通常の連鎖律（chain rule）を破綻させるからである。

本章は 3 つの段階で展開する。第 1 に、決定論的な ODE によって単一の状態を厳密かつ数値的に動かす。ここでは、最も速く減衰するモードが勾配降下法およびすべての陽的ルンゲ・クッタ法のソルバのステップサイズに上限を課す一方（38.2 節）、より真っ直ぐな輸送経路によって少数ステップでの生成が可能になる（38.7.2 節）。その過程で、駆動される線形 ODE から状態空間モデル（state space model）の漸化式が導かれ（38.1.2.3 節、12.4 節）、深い残差ネットワーク（7.4 節）はニューラル ODE および連続正規化フロー（continuous normalizing flow）へとつながる（38.3 節）。第 2 に、確率解析（stochastic calculus）によって連鎖律を修復し（38.4 節）、フォッカー・プランク方程式を通じて粒子群の密度を追跡することで、時間反転に必要なのは**スコア** $\nabla \log p_t$ だけであることを示す（38.5 節）。第 3 に、ノイズを加えた対や補間された対に対する最小二乗回帰（least-squares regression）により、場の正解ラベルなしにスコアや周辺速度（marginal velocity）を直接学習する（38.6 節、38.7 節）。

第 17 章へ直行したい読者は、最初の一読では連続随伴法（continuous adjoint method、38.3.2 節）と最適輸送（optimal transport、38.7.3 節）を読み飛ばしてよい。本章のほとんどの実験では、離散化誤差を明確に切り分けられるよう、閉形式のベクトル場に対してサンプラを検証する。なお、固有値分解（33.3 節）、ヤコビ行列（34.4.1 節）、およびガウス分布の条件付けを前提とし、測度論（measure theory）は仮定しない。

記法

Table 38.1 に、複数の節にわたって繰り返し現れる記号をまとめる。各節ではその節で必要な記号を改めて示し、一度しか使わない文字はその場で定義する。表中の文字が 2 つの意味を持つ場合は、それぞれの意味がどの箇所で成り立つかを該当する行に記している。

Table 38.1: ダイナミクスの章で用いる記法。

記号	意味	初出箇所
$\mathbf{x}(t) \in \mathbb{R}^d$、$\dot{\mathbf{x}}$	状態、その時間微分	38.1 節
$\mathbf{x}_0$	$t = 0$ における状態（軌跡の始点、ノイズ付加過程ではデータ点）	38.1 節
$\mathbf{f}(\mathbf{x}, t)$	ODE のベクトル場、SDE のドリフト	38.1 節、38.4 節
Φ_t、$\Phi_{s\to t}$	フロー写像 $\Phi_t(\mathbf{x}_0) = \mathbf{x}(t)$、時刻 s の状態から時刻 t の状態への写像	38.1 節
L	$\mathbf{f}$ のリプシッツ定数	38.1 節
T	終端時刻（ホライズン）	38.1 節
$\mathbf{A}$、$\lambda_j = a_j + \mathrm{i}\, b_j$、$\mathbf{V}$、$\Lambda$	線形場 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ の行列、その固有値、固有ベクトルを並べた行列、固有値の対角行列	38.1 節
λ	$\mathbf{A}$ またはヤコビ行列の固有値。逆時間サンプラでは注入するノイズ振幅の定数倍 λg	38.1 節、38.5 節
$\mathbf{K}$	回転を表す行列 $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$（$\dot{\mathbf{x}} = \mathbf{K}\mathbf{x}$）	38.1 節
$\mathbf{J}$	$\mathbf{f}$ のヤコビ行列	38.1 節
$\mathbf{x}^\star$	ベクトル場の不動点 $\mathbf{f}(\mathbf{x}^\star) = \mathbf{0}$（勾配フローでは損失の臨界点）	38.1 節
$\mathbf{u}(t)$、$\mathbf{B}$	入力とその行列（$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}(t)$）	38.1 節
h、$t_n = nh$、$\mathbf{x}_n$	ステップサイズ（38.4 節と 38.5 節では Δt と表記）、時間グリッド、t_n における状態（厳密解またはソルバの反復点）	38.1 節、38.2 節
$\bar{\mathbf{A}}$、$\bar{\mathbf{B}}$	入力を区間幅 h にわたり一定に保った厳密ステップ $\mathbf{x}_{n+1} = \bar{\mathbf{A}}\mathbf{x}_n + \bar{\mathbf{B}}\mathbf{u}_n$	38.1 節
$R(z)$、$z = h\lambda$	ソルバの安定性関数	38.2 節
ε_n	大域誤差 $\lVert \mathbf{x}_n - \mathbf{x}(t_n) \rVert$	38.2 節
$\mathcal{L}$、η、$\mathbf{H}$	損失、学習率、ヘッセ行列	38.1 節、38.2 節
N	終端時刻までのステップ数。ソルバのステップ $n = 1, \dots, N$（$Nh = T$、最初の 2 節ではこの総数にも n を用いる）。また 38.6 節では DDPM のノイズレベル $t = 1, \dots, N$	38.3 節
θ	ネットワークのパラメータ	38.3 節
$\mathbf{a}(t)$	随伴状態	38.3 節
p_t	時刻 t における状態の密度	38.3 節
ϵ	38.3 節ではトレース推定量のランダムプローブ。38.4 節以降では $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$ における標準ガウスノイズ（38.7 節ではノイズ側端点 $\mathbf{x}_1$ とも表記）	38.3 節、38.4 節
$\mathbf{X}_t$（1 次元では X_t）	時刻 t における確率過程の状態を表す確率変数（$\mathbf{x}$ はそのとる値）	38.4 節
$\mathbf{W}_t$	ブラウン運動	38.4 節
g、$g(t)$、$\mathbf{G}$	定数または時間変化する SDE のノイズ振幅（拡散係数）、ノイズ行列	38.4 節
θ	OU 過程 $dX = -\theta X\, dt + g\, dW$ のレート。38.3 節のスカラー例では $\dot{x} = \theta x$ のパラメータ	38.3 節、38.4 節
α_t、σ_t	$\mathbf{X}_t$ の、$\mathbf{X}_0$ を条件とする平均係数と標準偏差（σ_t はノイズレベル）	38.4 節

記号	意味	初出箇所
ϕ	状態の滑らかな関数。伊藤の補題における関数、およびフォッカー・プランク方程式の背後にあるテスト関数（他の箇所では角度）	38.4 節
ψ	ベクトル解析の準備における滑らかなスカラー場	38.5 節
$\mathbf{s}_t = \nabla \log p_t$、$\mathbf{s}_\theta$	スコア、学習されたスコア	38.5 節、38.6 節
$\mathbf{u}_t$、$\mathbf{u}_\theta$	p_t を輸送する速度（確率フロー速度、入力 $\mathbf{u}(t)$ とは無関係）、学習された速度	38.5 節、38.7 節
$\mathbf{j}$	確率流	38.5 節
$\bar{t} = T - t$	逆向きの時刻	38.5 節
w_k、$r_k(\mathbf{x})$、v_k	混合重み、負担率、各成分の分散	38.5 節
$\beta(t)$、β_t、$\bar{\alpha}_t$	分散保持 SDE のレート。DDPM のノイズスケジュール（およそ $\beta(t)$ とステップ幅の積であり、38.4 節では β_k と表記）とその累積積 $\bar{\alpha}_t = \prod_{s \le t}(1-\beta_s)$（本章の α_t^2 に相当）	38.4 節、38.6 節
$\bar{\mathbf{x}}_t$、$\bar{\sigma}_t$	DDIM 座標（リスケールされた点 $\mathbf{x}_t/\sqrt{\bar{\alpha}_t}$ とそのノイズ対信号比 $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$）	38.6 節
$w(t)$、γ	損失におけるノイズレベル t の重み、ガイダンス強度	38.6 節
π	カップリング（データとノイズの組の同時分布、ペアリングとも呼ぶ）	38.7 節
i、$\mathbf{I}$	虚数単位（立体）、単位行列	章全体

38.1 常微分方程式

本章の導入部で述べたように、勾配ステップ、残差ブロック（residual block）、および拡散サンプラの 1 ステップには共通点がある。いずれも、状態がどれほどの速さで変化するかを表す矢印に沿った微小な移動である。次節ではそのようなステップがどれほど良い近似かを問うが、本節ではそもそもそれが**何の**ステップなのかを問う。天気図の上の風のように、すべての点で速度が指定されており、その中に気球を放つとしよう。気球の経路は一意に定まるだろうか。同じ場所から放たれた 2 つの気球が別々の道を進むことはあるだろうか。1 時間前に気球がどこにあったかを知ることができるだろうか。そして、気球は最終的にどこへ行き着くのだろうか。

1687 年の Newton の法則はまさにこの種の規則であり、物体の状態がどう変化するかを述べるものであって、将来どこにあるかを直接与えるものではない。このような規則がただ 1 つの解を持つことは自明ではない。それを保証する定理は 1890 年代に Picard と Lindelöf によって証明されたものであり、速度がある点から隣の点へと急激に変化しすぎないことを要請する。後ほど、一見無害でありながらこの条件を満たさない規則に出会うことになる。

モデルにはこの定理が不可欠である。ネットワークが速度を与え、運動によって入力が運ばれた先をモデルの出力とするとき、解の一意性によって出力は入力の関数となり、運動を逆向きにたどればその逆関数が得られる。これは生成フロー（generative flow）が必要とする性質である。線形の規則であればさらに踏み込んで、解を具体的に書き下すことができる。その行列の固有値は、運動が減衰するか、増大するか、あるいは回転するかを決定しており、状態空間モデル（state space model、12.4 節）の記憶や訓練の収束もそこから読み取れる。

まずは手計算で解ける 2 つの運動から始め、上記の定理を証明したうえで、線形方程式を完全に解く。以降の議論はすべて 2 つの結果に基づいている。すなわち、緩やかに変化する速度規則は各点を通る軌跡（trajectory）をただ 1 つ持ち、線形規則の運動が減衰するかどうかはその固有値の実部によって決まるということである。減衰する前にいったん増大する減衰系と振り子を扱う最後の 2 つの項は、初読時には飛ばしてもよい。

38.1.1　ベクトル場とフロー

38.1.1.1　減衰、回転、スパイラル

最も単純な運動は**減衰**（decay）である。「距離に比例する速さでゼロへ向かって動く」という規則は方程式 $\dot{x} = -x$ で表される。ここでドットは時間に関する導関数を表す。初期条件 $x(0) = x_0$ からの解は次のとおりである。

$$x(t) = e^{-t} x_0,$$

このことは微分してみれば確かめられ、$\dot{x} = -e^{-t}x_0 = -x$ となる。単位時間が経過するごとに、原点までの距離は $e^{-1} \approx 0.37$ 倍になる。

次は**回転**（rotation）である。平面上において、位置ベクトルを直角に回転させたものを速度としよう。

$$\dot{\mathbf{x}} = \mathbf{K}\mathbf{x}, \qquad \mathbf{K} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

すなわち $\dot{x}_1 = -x_2$ および $\dot{x}_2 = x_1$ である。$\mathbf{x}^{\top}\mathbf{K}\mathbf{x} = -x_1 x_2 + x_2 x_1 = 0$ であるから、速度は位置と直交しており、したがって $\mathbf{x}$ の長さは決して変化しない。

$$\frac{d}{dt}\|\mathbf{x}\|^2 = 2\,\mathbf{x}^{\top}\dot{\mathbf{x}} = 2\,\mathbf{x}^{\top}\mathbf{K}\mathbf{x} = 0.$$

点は円周上を動き、その速さ $\|\mathbf{K}\mathbf{x}\|$ は半径 $\|\mathbf{x}\|$ に等しいため、単位時間あたり 1 ラジアンの速さで回転する。したがって 1 周するのに 2π かかる。

回転に半分の速さの減衰を加えると、本節および続く 2 つの節を通して例として用いる方程式が得られる。

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \qquad \mathbf{A} = -\tfrac{1}{2}\mathbf{I} + \mathbf{K} = \begin{pmatrix} -0.5 & -1 \\ 1 & -0.5 \end{pmatrix}.$$

回転の項は長さを変えず、減衰の項が長さを縮めるため、$\frac{d}{dt}\|\mathbf{x}\|^2 = 2\mathbf{x}^{\top}\mathbf{A}\mathbf{x} = -\|\mathbf{x}\|^2$ となる。これは長さの二乗に対する先ほどの減衰方程式にほかならず、$\|\mathbf{x}(t)\|^2 = e^{-t}\|\mathbf{x}_0\|^2$、すなわち $\|\mathbf{x}(t)\| = e^{-t/2}\|\mathbf{x}_0\|$ が従う。点は単位角速度で回転しながら速さ $\frac{1}{2}$ の率で長さを失い、スパイラルを描いて原点へと吸い込まれていく（Figure 38.1）。$\mathbf{A}$ の固有値は $-0.5 \pm \mathrm{i}$ であり、その実部 $-\frac{1}{2}$ が減衰率を、虚部 ± 1 が角速度を与えている。これは偶然ではない。38.1.2 節で示すように、この種のあらゆる線形方程式は、各固有値に対応する減衰・増大・回転の和として表されるのである。

38.1.1.2　初期値問題

運動の規則の大半は線形ではないため、それらをすべて扱える一般的な形式が必要である。一般の問題は、速度の規則と出発点を与える次の形で書かれる。

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t), \qquad \mathbf{x}(0) = \mathbf{x}_0. \tag{38.1}$$

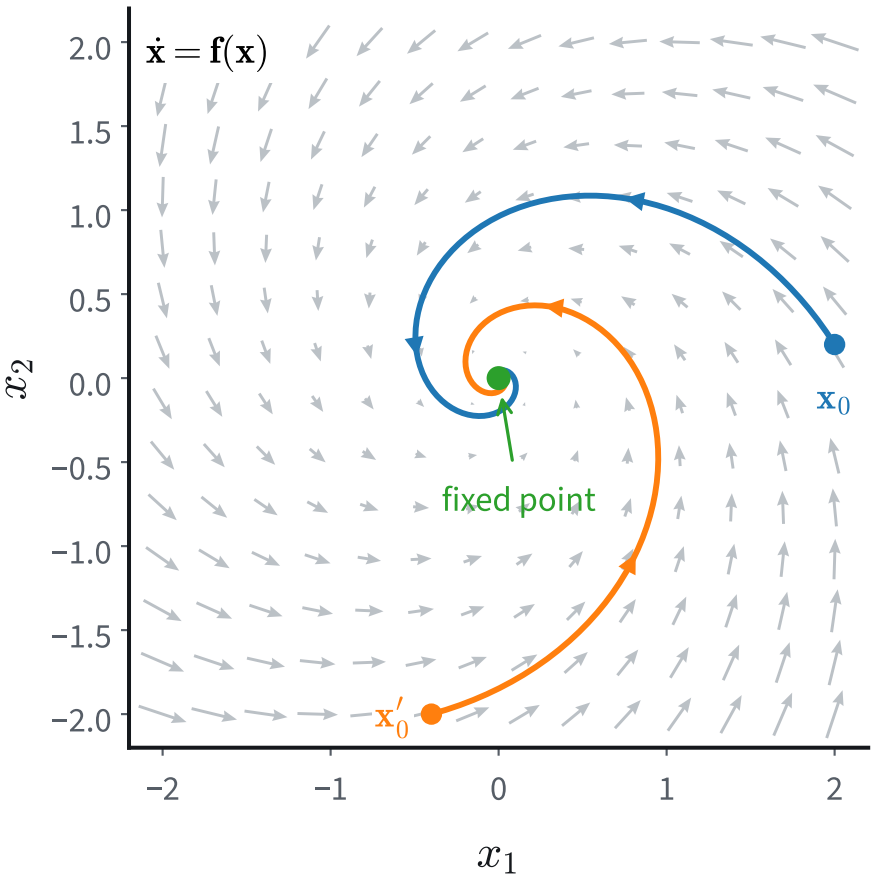


Figure 38.1: 速度場としての ODE。薄いグリッド状の矢印は場 $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$（ただし $\mathbf{A} = \left(\begin{smallmatrix} -0.5 & -1 \\ 1 & -0.5 \end{smallmatrix}\right)$）、固有値は $-0.5 \pm \mathrm{i}$）を表す。色付きの 2 本の曲線は軌跡であり、それぞれが各瞬間で局所的な矢印に沿って進み、場がゼロになる唯一の点（不動点）である原点へとスパイラル状に収束する。

ここで、$\mathbf{x}(t) \in \mathbb{R}^d$ は状態である。関数 $\mathbf{f} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ は**ベクトル場**（vector field）であり、各時刻・各点に矢印を対応させる。**軌跡**とは微分可能な経路であって、各瞬間における接ベクトルがその時点の位置にある矢印と一致するものをいう。

ベクトル $\mathbf{x}_0$ は**初期値**（initial value）であり、Equation 38.1 は**初期値問題**（initial-value problem）と呼ばれる。$\mathbf{f}$ が t に依存しないとき、系は**自律的**（autonomous）であるといい、矢印は固定されたまま状態だけが動く。減衰、回転、スパイラルは、それぞれ $\mathbf{f}(\mathbf{x}) = -\mathbf{x}$、$\mathbf{Kx}$、$\mathbf{Ax}$ という自律的な場である。一方、気球が飛んでいるあいだに風向きが変わる天気図は自律的ではない。

両辺を 0 から t まで積分すると（34.6 節）、同値な**積分形**（integral form）が得られる。

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(s), s)\, ds. \tag{38.2}$$

解は、初期位置に累積された速度を足し合わせたものになる。未知の経路が両辺に現れているため、これは依然として方程式であって明示的な解の公式ではないが、以下の存在証明ではこの形を用いる。

38.1.1.3 存在と一意性

初期値問題はつねに解を持ち、しかもただ 1 つだけ持つのだろうか。必ずしもそうとは限らない。$\dot{x} = \sqrt{|x|}$ を初期値 $x_0 = 0$ で考えよう。定数経路 $x \equiv 0$ はこの方程式の解である。さらに、任意の待ち時間 $c \geq 0$ に対して、次の関数もまた解となる。

$$x_c(t) = \begin{cases} 0, & t \leq c, \\ (t-c)^2/4, & t > c, \end{cases}$$

実際、$\dot{x}_c = (t-c)/2 = \sqrt{x_c}$ が $t > c$ で成り立つからである。静止している粒子は、いつ出発してもよいことになる。解は出発時刻ごとに 1 つずつ軸から扇状に広がっており（Figure 38.2 の (a)）、以下ではこの図形を**扇**（fan）と呼ぶ。この場で定義されたモデルは、入力に対してただ 1 つの出力を対応させることができない。原因は $\sqrt{|x|}$ の傾きが 0 で非有界であることにある。わずかな変位 x が、それよりはるかに大きな速度 $\sqrt{|x|}$ を生み出してしまうのである。

性質のよい場に対しては、積分形 Equation 38.2 を反復することで解を構成できる。すなわち、経路を 1 つ推測し、それに沿って場が指定する速度を読み取り、初期位置からそれらを積分して新しい経路を得て、この操作を繰り返すのである。これを **Picard 反復**（Picard iteration）という。$\dot{x} = x$、$x_0 = 1$ の場合、定数経路 $\varphi_0 \equiv 1$ から出発して $\varphi_{k+1}(t) = 1 + \int_0^t \varphi_k(s)\, ds$ とおくと、次のようになる。

$$\varphi_1(t) = 1 + t, \qquad \varphi_2(t) = 1 + t + \tfrac{t^2}{2}, \qquad \varphi_3(t) = 1 + t + \tfrac{t^2}{2} + \tfrac{t^3}{6}.$$

この反復列は解 e^t に収束する。この例では反復を 1 回進めるごとに e^t のテイラー展開の項が 1 つずつ加わっていく。他の場では反復列の形は異なるものの、場がある点から別の点へと急激に変化しすぎない限り、やはり収束する。次の定理は、この「急激に変化しすぎない」という意味を厳密に定式化する。

定理（Picard–Lindelöf） **$\mathbf{f}(\mathbf{x}, t)$ が t について連続であり、$\mathbf{x}$ について大域的リプシッツ連続**（globally Lipschitz）**であるとする。すなわち、ある定数 L が存在して次を満たすとする。**

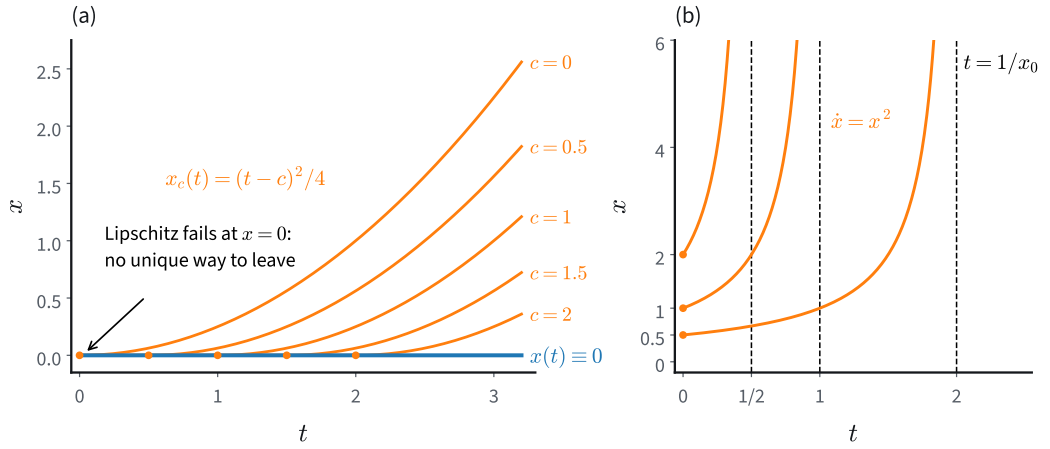


Figure 38.2: 以下の定理が適用できない 2 つの場。(a) $\dot{x} = \sqrt{|x|}$ の $x(0) = 0$ を通る解。定数解 $x \equiv 0$ に加え、放物線状の解 $x_c(t) = (t - c)^2/4$ が任意の出発時刻 c ごとに存在する。原点において $\sqrt{|x|}$ の傾きが非有界であるため、速度の変化の速さを抑える定数が存在せず（定理のリプシッツ条件が破れる）、解は一意ではない。(b) $\dot{x} = x^2$ の $x_0 = 2$、1、$1/2$（黒丸）から出発する解。ここでは場の傾きは各点の近傍で有界だが、すべての x に共通する 1 つの定数では抑えられず、各解 $x_0/(1 - x_0 t)$ は無限大に到達する時刻 $t = 1/x_0$（破線）までしか存在しない。

$$\|\mathbf{f}(\mathbf{x}, t) - \mathbf{f}(\mathbf{y}, t)\| \le L \, \|\mathbf{x} - \mathbf{y}\| \qquad \text{for all } \mathbf{x}, \mathbf{y}, t.$$

このとき、初期値問題 Equation 38.1 には区間 $[0, T]$ 上でただ 1 つの解が、任意の $\mathbf{x}_0$ および任意の $T > 0$ に対して存在する (Coddington and Levinson 1955)。

これこそ、場によってモデルを定義することを可能にする定理である。定数 L は、点から点へと移るときに矢印がどれほど速く変化しうるかを上から抑える。減衰では $L = 1$ であり、線形の場 $\mathbf{Ax}$ では $L = \|\mathbf{A}\|$ となる。ここで、ここでも本節の残りでも、行列のノルムとはスペクトルノルム（spectral norm）、すなわちその行列がベクトルを引き伸ばす最大の倍率を指す（33.4.2.2 節）。先ほどの扇は、そのような L が 0 の近傍に存在しないときに起こる現象である。以下の証明では、Picard 反復列が収束し、しかも唯一の解に収束することを示す。

証明連続な経路 φ（区間 $[0, \delta]$ 上で定義されるもの）がこの問題の解であるのは、それが次の **Picard 作用素**（Picard operator）の不動点（fixed point）であるとき、かつそのときに限る。

$$(P\varphi)(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\varphi(s), s) \, ds.$$

言葉で言えば、P は Picard 反復の 1 ステップであり、推測した経路に沿って速度を読み取り、初期位置からそれらを積分して足し合わせる操作である。（解は積分形を満たし、逆に連続な不動点は微分積分学の基本定理（fundamental theorem of calculus）によって微分可能であり、正しい導関数と正しい初期値を持つ。）

証明の核心となる非自明なステップは、リプシッツ条件による評価を積分の内側へ通すことである。$[0, \delta]$ 上の 2 つの経路に対して $d(\varphi, \psi) = \max_{t \le \delta} \|\varphi(t) - \psi(t)\|$ とおく。このとき $t \le \delta$ に対して次が成り立つ。

$$\|(P\varphi)(t) - (P\psi)(t)\| \le \int_0^t L\,\|\varphi(s)-\psi(s)\|\,ds \le L\,\delta\; d(\varphi,\psi).$$

$\delta = 1/(2L)$ ととれば、P を 1 回適用するごとに 2 つの候補経路のあいだの距離は半分になる。

定数経路 $\varphi_0 \equiv \mathbf{x}_0$ から出発し、$\varphi_{k+1} = P\varphi_k$ と定める。すると隣り合う反復点の差は高々 $2^{-k}d(\varphi_1,\varphi_0)$ となり、これらの差の和は幾何級数のように抑えられる。したがって反復列は距離 d に関して、すなわち $[0,\delta]$ 上で一様に、ある経路 φ へと収束する。ここでは解析学の事実を 1 つ証明なしで認めることにする。それは、連続な経路の一様極限は連続であるという事実である。$d(P\varphi_k, P\varphi) \le \tfrac{1}{2}d(\varphi_k,\varphi) \to 0$ であるから、経路の列 $P\varphi_k = \varphi_{k+1}$ は $P\varphi$ にも φ にも収束する。よって $\varphi = P\varphi$ が成り立ち、この極限は解である。また、2 つの解 φ,ψ があれば $d(\varphi,\psi) = d(P\varphi,P\psi) \le \tfrac{1}{2}d(\varphi,\psi)$ を満たすため、両者は一致する。

最後に、$t=\delta$ において $\varphi(\delta)$ から再び同じ議論を始める。ステップ幅 $1/(2L)$ は解がどこにあるかに依存しない。なぜなら L は至るところで同じだからであり、したがってこの長さの区間を $\lceil 2LT \rceil$ 個つなげば $[0,T]$ 全体が覆われ、各区間上にただ 1 つの解が定まる。$\blacksquare$

この定理は、リプシッツ評価が存在することと、その評価が大域的に成り立つことの両方を要請しており、それぞれの条件が異なる破綻を排除している。リプシッツ評価がない場合に何が起こるかは、扇の例が示している。一方、大域的な評価がなければ、解は爆発しうる。場 $\dot{x} = x^2$ は滑らかであり、手計算で解くことができる。$\frac{d}{dt}(-1/x) = \dot{x}/x^2 = 1$ より、量 $-1/x$ は単位速度で増加する。すなわち $-1/x(t) = t - 1/x_0$ であり、したがって次式が得られる。

$$x(t) = \frac{x_0}{1 - x_0 t},$$

この解は、初期値 $x_0 > 0$ から出発すると有限時刻 $t = 1/x_0$ で無限大に到達する（Figure 38.2 の (b)）。なぜこの例が定理の対象外なのかは、証明の最後のステップを見ればわかる。区間 $|x| \le \rho$ 上では、場は $L = 2\rho$ でリプシッツ連続である（$|x^2 - y^2| = |x+y|\,|x-y|$ による）。そのため、解が大きくなるにつれて再出発の区間幅 $1/(2L) = 1/(4\rho)$ が縮んでいき、それらを無限個足し合わせても有限時刻にしかならないことが起こりうる。有界集合上でのみリプシッツ連続な場は、しばらくのあいだしか解の存在を保証しない。

38.1.1.4　フロー写像

生成モデルは初期点の雲全体を一度に動かすため、ここからは場を固定して初期点を変化させることにしよう。まずは単純な場合として、時間変化しない場を取り上げる（時間変化する風の例は本項の最後で再び扱う）。自律的かつ大域的リプシッツ連続な場に対して、次を定義する。

$$\Phi_t(\mathbf{x}_0) = \mathbf{x}(t)$$

これは、時刻 t における、$\mathbf{x}_0$ から出発した唯一の軌跡の状態を表す。この**フロー写像**（flow map）$\Phi_t : \mathbb{R}^d \to \mathbb{R}^d$ は、空間全体を矢印に沿って時間 t だけ前へ進める。

減衰の場合、フロー写像は $\Phi_t(x) = e^{-t}x$ である。時間 s だけ流してからさらに時間 t だけ流すと $\Phi_t(\Phi_s(x)) = e^{-t}e^{-s}x = e^{-(t+s)}x = \Phi_{t+s}(x)$ となり、また $\Phi_{-t}(x) = e^{t}x$ は Φ_t を元に戻す。同じ法則があらゆる自律的な場に対して成り立つ。

$$\Phi_0 = \mathrm{id}, \qquad \Phi_{t+s} = \Phi_t \circ \Phi_s. \tag{38.3}$$

その理由は、矢印が時間とともに変化しないことにある。経路 $\tau \mapsto \mathbf{x}(s+\tau)$ は $\Phi_s(\mathbf{x}_0)$ から出発して同じ矢印に沿うため、一意性によりその点からの軌跡に一致する。減衰の例からは、この法則が負の時刻に対しても成り立つことが示唆される。実際そのとおりであり、場を逆向きにたどることは、別の場に同じ定理を適用することに帰着するからである。

系（フロー写像は全単射である） $\mathbf{f}(\mathbf{x})$ が自律的かつ大域的リプシッツ連続であるとする。このとき Φ_t はすべての $t \in \mathbb{R}$ に対して存在し、各 Φ_t は $\mathbb{R}^d$ 上の全単射（bijection）であって逆写像 Φ_{-t} を持つ。一方、$\mathbf{f}(\mathbf{x}, t)$ が時刻に依存し、かつ定理の仮定を満たすならば、任意の $T > 0$ に対して時刻 0 の状態を時刻 T の状態へと運ぶ写像は $\mathbb{R}^d$ 上の全単射である。

証明 場 $\mathbf{g}(\mathbf{y}) = -\mathbf{f}(\mathbf{y})$ は同じ定数 L でリプシッツ連続である。経路 $\mathbf{x}(t)$ が $\dot{\mathbf{x}} = \mathbf{f}$ の $[-T, 0]$ 上の解であるのは、$\mathbf{y}(s) = \mathbf{x}(-s)$ が $\dot{\mathbf{y}} = \mathbf{g}$ の $[0, T]$ 上の解であるとき、かつそのときに限る。なぜなら $\dot{\mathbf{y}}(s) = -\dot{\mathbf{x}}(-s) = -\mathbf{f}(\mathbf{y}(s))$ だからである。$\mathbf{g}$ に定理を適用すれば、任意の初期値に対して時間を逆向きにたどる一意な解が得られるため、Φ_t はすべての実数 t に対して定義される。さらに合成則 Equation 38.3 の証明では s や t の符号を使っていなかったから、$\Phi_{-t} \circ \Phi_t = \Phi_t \circ \Phi_{-t} = \Phi_0 = \mathrm{id}$ が従う。

この時間反転の議論には、矢印が静止している必要はなかった。時刻に依存する場に対しては、時刻 T からフィルムを逆再生すればよい。経路 $\mathbf{x}(t)$ が $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ の $[0, T]$ 上の解であるのは、$\mathbf{y}(s) = \mathbf{x}(T - s)$ が $\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, s)$（ただし $\mathbf{g}(\mathbf{y}, s) = -\mathbf{f}(\mathbf{y}, T - s)$）の解であるとき、かつそのときに限る。この $\mathbf{g}$ もまた時刻について連続かつ同じ定数 L でリプシッツ連続であるため、定理により各点 $\mathbf{z}$ からは逆向きの経路がただ 1 つ出発する。これを順方向に読めば、各点 $\mathbf{z}$ に時刻 T で到達する唯一の軌跡にほかならない。すなわち、すべての終端状態はただ 1 つの初期状態から生じる。$\blacksquare$

時間とともに変化する場で失われるのは、Equation 38.3 の形そのものだけである。正午からの 1 時間の風と真夜中からの 1 時間の風は異なる写像を与えるため、写像は経過時間だけでなく開始時刻にも依存する。そこで $\Phi_{s \to t}(\mathbf{x})$ によって、時刻 t における状態を表そう（ただし軌跡は位置 $\mathbf{x}$ を時刻 s に通るものとする）。s から r へ進み、さらに r から t へ進むことは、先ほどと同じ一意性の議論により s から t へ進むことに等しい。

$$\Phi_{s \to s} = \mathrm{id}, \qquad \Phi_{r \to t} \circ \Phi_{s \to r} = \Phi_{s \to t}. \tag{38.4}$$

これら 3 つの時刻はどの順序で並んでいてもよいため、$\Phi_{t \to s}$ は $\Phi_{s \to t}$ の逆写像となる。自律的な場では $\Phi_{s \to t} = \Phi_{t - s}$ であり、Equation 38.4 は元の法則に帰着する。時刻に依存する場に対しても、Φ_t という記号を $\Phi_{0 \to t}$ の略記として引き続き用いる。この 2 時刻間の法則は単なる形式的な書き換えではない。17.8 節の少数ステップサンプラは、ネットワークが $\Phi_{s \to t}$ そのものを模倣するように訓練し、その一部はこの法則自体を訓練信号として用いる。すなわち、ネットワークによる 2 回の短いジャンプの合成が、1 回の長いジャンプと一致しなければならない。

ここから得られる帰結の 1 つとして、2 つの異なる軌跡が同じ時刻に同じ点に居合わせることは決してない。もしそのような合流点から時間を逆向きにたどれば、定理によってただ 1 つの経路しか許されないからである。リプシッツ連続な風に対しては、これで冒頭の最初の 3 つの問いに答えたことになる。気球の経路は一意に定まり、同時に同じ場所から放たれた 2 つの気球は離れずに進み、天気図を逆再生すれば 1 時間前にどこにあったかがわかる。

ノイズからデータへの可逆な写像であり、しかもその逆写像を計算できることこそ生成フローが必要とする性質であり、多くのニューラルネットワークが定める場はこの条件を満たす。たとえば $\mathbf{f}(\mathbf{x}) = \mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$ のように、線形変換とリプシッツ連続な活性化関数から構成される場は大域的リプシッツ連続であり、その定数 L は高々、重みのノルムと活性化関数のリプシッツ定数の積（この例では $\|\mathbf{W}_2\| \, \|\mathbf{W}_1\|$）で抑えられる。したがって、場が時刻にも依存しうる 38.3 節のモデルは適定（well-posed）であり、先ほどの系によって入力から出力への写像は全単射となる。

38.1.2 線形 ODE と安定性

冒頭で挙げた最後の問い、すなわち運動が最終的にどこへ行き着くかという問いには、ある 1 つの族に対して完全な答えがある。線形の場は閉形式で解くことができ、さらに場がゼロになる点の近傍では、他の大半の場の振る舞いをも記述する。

38.1.2.1 行列指数関数

スカラー方程式 $\dot{x} = ax$ の解は $x(t) = e^{at}x_0$ である。次の線形系を考えよう。

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \qquad \mathbf{x}(0) = \mathbf{x}_0, \qquad \mathbf{A} \in \mathbb{R}^{d \times d},$$

これに対しても、指数部に行列を置いた同じ形の解を探す。任意の正方行列 $\mathbf{C}$ に対し、e^{z} を定めるべき級数によって**行列指数関数**（matrix exponential）を次のように定義する。

$$e^{\mathbf{C}} = \sum_{k=0}^{\infty} \frac{\mathbf{C}^k}{k!} = \mathbf{I} + \mathbf{C} + \frac{\mathbf{C}^2}{2} + \frac{\mathbf{C}^3}{6} + \cdots . \tag{38.5}$$

この級数はあらゆる行列に対して収束する。ある行列で引き伸ばしてから別の行列で引き伸ばしたときの倍率は高々それぞれの倍率の積、すなわち $\|\mathbf{C}\mathbf{D}\| \le \|\mathbf{C}\|\,\|\mathbf{D}\|$ で抑えられるため、$\|\mathbf{C}^k\| \le \|\mathbf{C}\|^k$ が成り立ち、級数の各項は収束するスカラー級数 $e^{\|\mathbf{C}\|}$ によって上から抑えられるからである。

命題（行列指数関数は線形 ODE を解く） $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$ **は** $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$、$\mathbf{x}(0) = \mathbf{x}_0$ **の一意な解である。**

証明級数 Equation 38.5 に $\mathbf{C} = \mathbf{A}t$ を代入したものを項別に微分する（任意の有界な t 区間上で、微分後の級数も先ほどと同じ優級数の評価により一様収束するため、項別微分が正当化される）。

$$\frac{d}{dt} e^{\mathbf{A}t} = \sum_{k=1}^{\infty} \frac{\mathbf{A}^k\, t^{k-1}}{(k-1)!} = \mathbf{A} \sum_{j=0}^{\infty} \frac{(\mathbf{A}t)^j}{j!} = \mathbf{A}\, e^{\mathbf{A}t}.$$

よって $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$ はこの ODE を満たし、$t = 0$ では級数が $\mathbf{I}$ になるため $\mathbf{x}(0) = \mathbf{x}_0$ も満たす。一意性は Picard–Lindelöf の定理から従う。実際、場 $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ は $L = \|\mathbf{A}\|$ でリプシッツ連続である。$\blacksquare$

行列のべき乗の無限級数は、このままでは振る舞いを読み取れる式ではない。固有値分解（eigendecomposition）を用いると、これは d 個のスカラー指数関数へと変わる。33.3 節で見たように、$\mathbf{A}$ が d 個の線形独立な固有ベクトル $\mathbf{v}_j$（$\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j$）を持つならば、$\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_d)$ および $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$ とおくことで $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ が成り立つのであった。

するとべき乗は $(\mathbf{A}t)^k = \mathbf{V}(\Lambda t)^k\mathbf{V}^{-1}$ のように簡約される。隣り合う $\mathbf{V}^{-1}\mathbf{V}$ がすべて打ち消し合うからであり、級数 Equation 38.5 の和をとると次式が得られる。

$$e^{\mathbf{A}t} = \mathbf{V} e^{\Lambda t} \mathbf{V}^{-1}, \qquad e^{\Lambda t} = \operatorname{diag}\!\left(e^{\lambda_1 t}, \dots, e^{\lambda_d t}\right). \tag{38.6}$$

固有値の節ではこの公式を引用するにとどめ、その導出は本節に委ねられていた。この式は、系が非結合化されることを意味している。初期状態を固有基底で $\mathbf{x}_0 = \sum_j c_j \mathbf{v}_j$ と展開すれば、次が得られる。

$$\mathbf{x}(t) = \sum_{j=1}^{d} c_j\, e^{\lambda_j t}\, \mathbf{v}_j .$$

各固有方向に沿って、系はスカラー方程式 $\dot{x} = \lambda_j x$ となり、他の方向とは独立に時間発展する。各項 $c_j e^{\lambda_j t}\mathbf{v}_j$ は系の**モード**と呼ばれる。

スパイラルの例では、一般公式よりもさらに見通しよく計算できる。その行列は $\mathbf{A} = -\tfrac{1}{2}\mathbf{I} + \mathbf{K}$ という和の形をしており、数に対して $e^{p+q} = e^{p}e^{q}$ と分解できるように指数関数を分解したい。行列において $e^{\mathbf{P}+\mathbf{Q}} = e^{\mathbf{P}}e^{\mathbf{Q}}$ という法則が成り立つのは $\mathbf{PQ} = \mathbf{QP}$ のときである。このときには $(\mathbf{P}+\mathbf{Q})^{k}$ が数の場合と同様に二項定理で展開できるからである。単位行列はあらゆる行列と可換であるため、$e^{\mathbf{A}t} = e^{-t/2}e^{\mathbf{K}t}$ となる。また $\mathbf{K}^2 = -\mathbf{I}$ より、$\mathbf{K}$ のべき乗は $\mathbf{I}, \mathbf{K}, -\mathbf{I}, -\mathbf{K}$ を巡回し、$e^{\mathbf{K}t}$ の級数は偶数次の項と奇数次の項、すなわち $\cos t$ と $\sin t$ の級数に分かれる。

$$e^{\mathbf{A}t} = e^{-t/2}\left(\cos t\,\mathbf{I} + \sin t\,\mathbf{K}\right) = e^{-t/2}\begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}.$$

場そのものから導いたとおり、このフローは $e^{-t/2}$ 倍に縮小しながら t ラジアンだけ回転する。$\mathbf{x}_0 = (2, 0)$ から出発すると、$t = 2$ における状態は $e^{-1}(2\cos 2,\, 2\sin 2) = (-0.306,\, 0.669)$ となり、その長さは $2e^{-1} = 0.736$ である。この $\mathbf{A}$ に対して固有ベクトルと固有値 $-0.5 \pm \mathrm{i}$ は複素数になるが、積 $\mathbf{V}e^{\Lambda t}\mathbf{V}^{-1}$ は実数となる。

ここで一言注意しておくと、2 つの行列が可換でないときにはこの分解は成り立たない。ともに二乗するとゼロ行列になる $\mathbf{P} = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ と $\mathbf{Q} = \mathbf{P}^{\top}$ に対して、$e^{\mathbf{P}}e^{\mathbf{Q}} = (\mathbf{I}+\mathbf{P})(\mathbf{I}+\mathbf{Q}) = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 1 \end{smallmatrix}\right)$ であるのに対し、$e^{\mathbf{P}+\mathbf{Q}}$ の両対角成分は $\cosh 1 \approx 1.54$ となる。

$e^{\mathbf{A}t}$ に至る第 3 の道として、級数も固有ベクトルも必要としない複利の極限 $(\mathbf{I} + t\mathbf{A}/n)^{n}\,(n \to \infty)$ がある。各因子は、微小時間 t/n だけ場に沿って状態を進める操作である。この極限はどれほどの速さで収束するだろうか。固有基底ではこの積は $\mathbf{V}\,\mathrm{diag}\big((1 + \lambda_j t/n)^{n}\big)\mathbf{V}^{-1}$ となるため、1 つの固有値について調べれば十分である。$\log(1+z) = z - z^2/2 + \cdots$ より、次が成り立つ。

$$\left(1 + \frac{\lambda t}{n}\right)^{n} = \exp\!\left(n \log\!\left(1 + \tfrac{\lambda t}{n}\right)\right) \approx e^{\lambda t}\left(1 - \frac{(\lambda t)^2}{2n}\right),$$

したがって、各 $(1 + \lambda_j t/n)^{n}$ は $e^{\lambda_j t}$ に収束し、その大きさに対する相対誤差は約 $|\lambda_j t|^2/(2n)$ となる。$t = 2$ におけるスパイラルでは、この比率は両固有値とも $2.5/n$ であり、$|e^{\lambda t}| = e^{-1}$ であるから、誤差は約 $0.9/n$ と予測される。次のコードセルでは、固有値分解から $e^{\mathbf{A}t}$ を計算し、3 つの n の値において複利計算による近似と比較する。

```
A = np.array([[-0.5, -1.0], [1.0, -0.5]])      # spiral: eigenvalues -0.5 +- i
t, x0 = 2.0, np.array([2.0, 0.0])
lam, V = np.linalg.eig(A)                      # A = V diag(lam) V^{-1}
expAt = (V @ np.diag(np.exp(lam * t)) @ np.linalg.inv(V)).real
x = expAt @ x0
print('eigenvalues of A:', lam.round(4))
print('x(2) =', x.round(4), f' length {np.linalg.norm(x):.6f}')
for n in [2 ** 10, 2 ** 16, 2 ** 24]:          # compounding (I + tA/n)^n
    err = np.abs(np.linalg.matrix_power(np.eye(2) + (t / n) * A, n) - expAt).max()
    print(f'n = 2^{int(np.log2(n))}: error {err:.1e}, n * error = {n * err:.2f}')
```

```
eigenvalues of A: [-0.5+1.j -0.5-1.j]
x(2) = [-0.3062  0.669 ]  length 0.735759
n = 2^10: error 8.8e-04, n * error = 0.90
n = 2^16: error 1.4e-05, n * error = 0.90
n = 2^24: error 5.4e-08, n * error = 0.90
```

状態 $(-0.3062,\, 0.6690)$ とその長さ 0.735759 は、手計算で求めた値と一致している。複利計算は予測された速度で収束し、それより速くはならない。誤差の最大成分に n を掛けた値は、$n = 2^{10}$ から $n = 2^{24}$ まで 0.90 のままである。これは金利を行列にした複利計算にほかならない。指数関数の定義としては優れているが、数値計算法としては不向きである。約 1700 万回の複利期間を経てもなお、誤差は 5×10^{-8} 残っている。

38.1.2.2 固有値による安定性

非結合化された形を見れば、線形系が落ち着くかどうかが一目でわかる。各固有値を $\lambda_j = a_j + \mathrm{i}\, b_j$ と書こう。そのモードの大きさは $|e^{\lambda_j t}| = e^{a_j t}$ である（$e^{\mathrm{i} b_j t} = \cos b_j t + \mathrm{i} \sin b_j t$ の絶対値が 1 であることによる）。実部が増大か減衰かを決め、虚部が回転を決める。対角化可能な $\mathbf{A}$ に対して、ここから次の分類が得られる。

- $\operatorname{Re} \lambda_j < 0$ がすべての j について成り立つならば、すべてのモードが減衰し、いかなる初期値からも $\mathbf{x}(t) \to \mathbf{0}$ となる。このとき原点は**漸近安定**（asymptotically stable）である。最も遅いモードが減衰率 $-\max_j \operatorname{Re} \lambda_j$ を決める。
- $\operatorname{Re} \lambda_j > 0$ がある j について成り立つならば、そのモードは指数関数的に増大し、$\mathbf{v}_j$ 方向の成分を持つあらゆる初期値は遠ざかっていく。このとき原点は**不安定**（unstable）である。
- その中間にあたる $\operatorname{Re} \lambda_j = 0$ のモードは、増大も減衰もしない。静止したままであるか、先ほどの回転の例のように永遠に周回し続ける。

複素共役ペア $a \pm \mathrm{i} b$ は一般の率を持つスパイラルであり、角速度 b で回転しつつ a の符号に応じて減衰または増大する。これを見るために、$\lambda = a + \mathrm{i} b$ の固有ベクトルを実ベクトルを用いて $\mathbf{v} = \mathbf{p} + \mathrm{i}\mathbf{q}$ と書く（$\mathbf{p}, \mathbf{q}$ は $b \neq 0$ より線形独立である）。$\mathbf{A}\mathbf{v} = (a + \mathrm{i} b)\mathbf{v}$ の実部と虚部をとると、次のようになる。

$$\mathbf{A}\mathbf{p} = a\mathbf{p} - b\mathbf{q}, \qquad \mathbf{A}\mathbf{q} = b\mathbf{p} + a\mathbf{q}.$$

したがって $\mathbf{A}$ は $\mathbf{p}$ と $\mathbf{q}$ が張る平面をそれ自身に写し、基底 $(\mathbf{q}, \mathbf{p})$ においては行列 $a\mathbf{I} + b\mathbf{K}$ で表される。するとスパイラルのときと同じ計算から次式が得られる。

$$e^{(a\mathbf{I} + b\mathbf{K})t} = e^{at}\left(\cos bt\, \mathbf{I} + \sin bt\, \mathbf{K}\right) :$$

すなわち、e^{at} 倍のスケーリングと角度 bt の回転の合成である。この同じブロックを連続時間に流すのではなく離散時間で反復したものは、33.3.1.6 節に登場する。

2 次元では以上から 6 通りの相図が得られ、Figure 38.3 にそれぞれの代表的な場を 1 つずつ描いた。実固有値の場合、原点は**安定ノード**（stable node、両方とも負）、**不安定ノード**（unstable node、両方とも正）、または**鞍点**（saddle、正負が 1 つずつ）となる。複素共役ペアの場合は、**安定スパイラル**（stable spiral）、**不安定スパイラル**（unstable spiral）、あるいは実部がゼロのときの**渦心点**（center）となる。（固有値 0 は除外している。この場合は直線上のすべての点が静止点になる。）本節を通して用いている例は安定スパイラルである。

訓練において最もよく出会うのは上段のケースである。二次損失 $\frac{1}{2}\mathbf{x}^{\top}\mathbf{H}\mathbf{x}$ に対する連続時間の勾配降下法は線形の場 $\dot{\mathbf{x}} = -\mathbf{H}\mathbf{x}$ となり、対称行列 $\mathbf{H}$ の固有値はすべて実数であるため、このよう

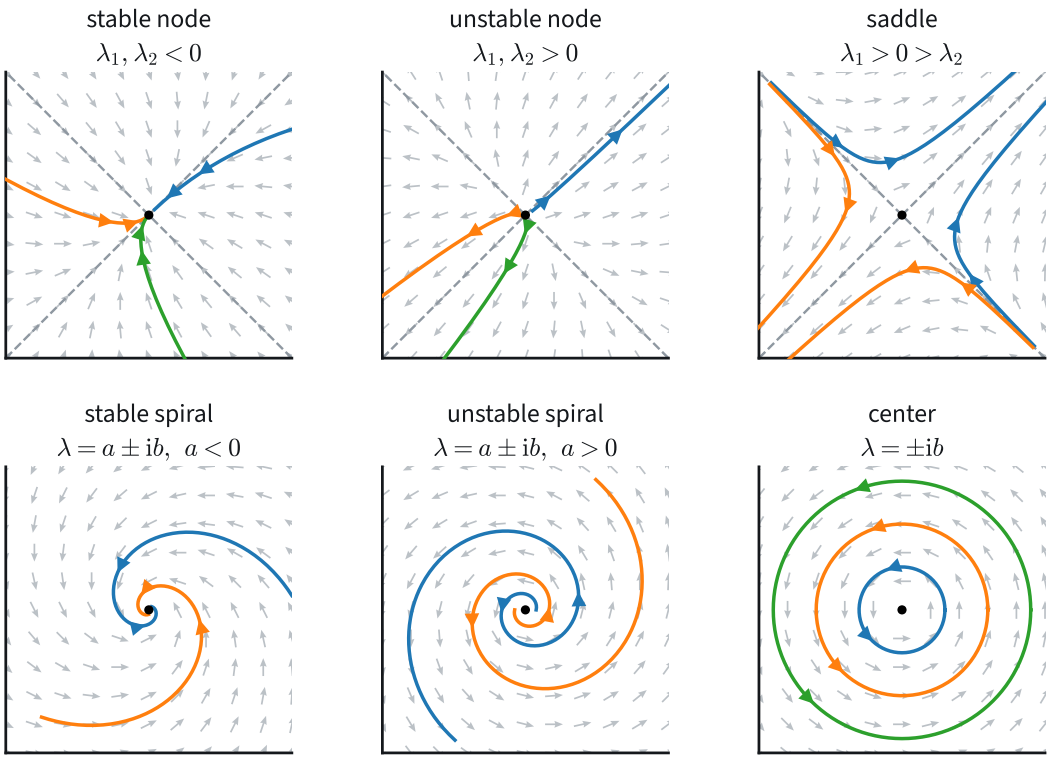


Figure 38.3: 6 種類の平面線形系。上段は実固有値の場合で、安定ノード（両方とも負）、不安定ノード（両方とも正）、鞍点（異符号）であり、破線は固有方向を表す。軌跡はこれらに沿って流入または流出する。下段は複素共役ペア $a \pm ib$ の場合で、安定スパイラル（$a < 0$）、不安定スパイラル（$a > 0$）、渦心点（$a = 0$、閉軌道）である。

な損失からはノードと鞍点しか生じない。回転が生じるには勾配ではない場が必要であり、たとえば最小構成の GAN における 2 人のプレイヤーは、渦心点となって平衡点の周りを周回し続ける（16.4 節）。

A の固有ベクトルが足りず対角化できない場合には、モードに多項式因子が加わって $t^k e^{\lambda t}$ の形になる。これが安定性の判定を左右するのは虚軸上の固有値に限られ、最小の具体例を演習問題 3 で扱う。

38.1.2.3 入力を伴う線形系

ここまでは、外部から系を押す力は存在しなかった。系列を読み込む状態空間モデル（12.4 節）の状態は、入力 $\mathbf{u}(t)$ を伴う線形方程式に従い、この入力は行列 $\mathbf{B}$ を介して状態に加わる。

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0.$$

行列指数関数はこの方程式も解くことができ、ある 1 つの着眼点から解が求まる。入力がなければ、量 $e^{-\mathbf{A}t}\mathbf{x}(t)$ は $\mathbf{x}_0$ に静止したままである。これは自由運動に沿って時刻 0 へと引き戻された状態にほかならない。入力があるとき、この量は入力が押す速さでしか動けない。実際、$\frac{d}{dt}e^{-\mathbf{A}t} = -e^{-\mathbf{A}t}\mathbf{A}$ であるから、積の微分法則より $\frac{d}{dt}\big(e^{-\mathbf{A}t}\mathbf{x}\big) = e^{-\mathbf{A}t}(\dot{\mathbf{x}} - \mathbf{A}\mathbf{x}) = e^{-\mathbf{A}t}\mathbf{B}\mathbf{u}$ となり、右辺にはもはや $\mathbf{x}$ が含まれない。0 から t まで積分して $e^{\mathbf{A}t}$ を掛けると、次式が得られる。

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-s)}\,\mathbf{B}\mathbf{u}(s)\,ds. \tag{38.7}$$

これが**定数変化法**（variation of constants）の公式である。定数 $\mathbf{x}_0$（自由解 $e^{\mathbf{A}t}\mathbf{x}_0$ に現れるもの）が時間とともに変化することを許したためにこう呼ばれる。状態は、減衰した初期状態に過去のすべての入力（それぞれ到着してからの経過時間だけ減衰したもの）を足し合わせたものになる。

系列モデルは、あらゆる瞬間ではなくステップごとに 1 サンプルずつ入力を受け取る。そこで時間を長さ h のステップに区切り、グリッド時刻 $t_n = nh$ とグリッド状態 $\mathbf{x}_n = \mathbf{x}(t_n)$ を定め、各ステップのあいだ入力は一定、すなわち $\mathbf{u}(s) = \mathbf{u}_n$（$s \in [t_n, t_n + h)$）であると仮定しよう。これを**ゼロ次ホールド**（zero-order hold）という。$\mathbf{x}_n$ から時刻 t_n に出発する 1 ステップに対して Equation 38.7 を適用すると、定数 $\mathbf{u}_n$ が積分の外に出る。さらに $\tau = t_{n+1} - s$ と変数変換すれば残る積分は $\int_0^h e^{\mathbf{A}\tau}d\tau$ であり、これはすべてのステップで同じ行列になる。したがってグリッド上の状態は、いかなる近似も伴わずに次の線形漸化式（linear recurrence）に従う。

$$\mathbf{x}_{n+1} = \bar{\mathbf{A}}\,\mathbf{x}_n + \bar{\mathbf{B}}\,\mathbf{u}_n, \qquad \bar{\mathbf{A}} = e^{\mathbf{A}h}, \qquad \bar{\mathbf{B}} = \int_0^h e^{\mathbf{A}\tau}\,d\tau\;\mathbf{B} = \mathbf{A}^{-1}\big(e^{\mathbf{A}h} - \mathbf{I}\big)\mathbf{B}, \tag{38.8}$$

ただし最後の表式には $\mathbf{A}$ が正則（可逆）である必要がある。1 次元ではこれらは $\bar{a} = e^{h\lambda}$ および $\bar{b} = (e^{h\lambda} - 1)\,b/\lambda$ となり、12.4.1.2 節の状態空間モデルを構築する基礎となる係数である（そこではステップ幅を Δ、状態を h と書いている）。

ここで押さえておくべき要点は、$|e^{h\lambda}| = e^{h\,\mathrm{Re}\,\lambda} < 1$ が任意のステップ幅 $h > 0$ に対して、$\mathrm{Re}\,\lambda < 0$ である限りつねに成り立つことである。すなわち、元の方程式が安定であれば厳密な漸化式もつねに安定になる。これに対し、複利計算 1 期間分の因子 $1 + h\lambda$（途中で矢印を見直すことなく時間 h だけ場に沿って進むもの）と比較してみよう。$\dot{x} = -50x$ において $h = 0.05$ とすると、厳密な因子は $e^{-2.5} = 0.08$ であるのに対し、$1 + h\lambda = -1.5$ は状態の符号を反転させ

たうえに絶対値を拡大してしまう。このように場に沿って一歩ずつ進む手法がオイラー法であり、因子 -1.5 が訓練に何をもたらすかは 38.2 節で見る。

この漸化式が厳密であるのは各ステップのあいだ入力が一定のときに限られるが、実際の系列モデルはステップ間でも変動する信号のサンプルを読み込む。では、このホールドによる誤差はどれほど生じるのだろうか。次のコードセルでは、$\dot{x} = \lambda x + \sin t$ に対して $x(0) = 1$ から $T = 2$ まで、入力を各ステップ開始時の値に保持したスカラー漸化式を実行し、閉形式解と比較する。遅い減衰率 $\lambda = -1$ と速い減衰率 $\lambda = -50$ の両方について計算し、$1 + h\lambda$ を厳密な因子 $e^{h\lambda}$ と並べて表示する。

```
def zoh(lam, h, u, x0=1.0, T=2.0):
    """Exact steps of dx/dt = lam x + u(t), u held at u(t_n) over each step."""
    a, b = np.exp(h * lam), np.expm1(h * lam) / lam
    x = x0
    for n in range(round(T / h)):
        x = a * x + b * u(n * h)
    return x

def exact_sin(lam, T=2.0):
    """Solution of dx/dt = lam x + sin t with x(0) = 1, at time T."""
    c = 1 + lam ** 2
    return np.exp(lam * T) * (1 + 1 / c) - (lam * np.sin(T) + np.cos(T)) / c

print(f'{"lam":>5} {"h":>5} {"e^(h lam)":>10} {"1 + h lam":>10}'
      f' {"error at T":>11}')
for lam in [-1.0, -50.0]:
    for h in [0.5, 0.1, 0.02]:
        err = abs(zoh(lam, h, np.sin) - exact_sin(lam))
        print(f'{lam:5.0f} {h:5.2f} {np.exp(h * lam):10.3g}'
              f' {1 + h * lam:10.2f} {err:11.1e}')
```

```
lam     h  e^(h lam)  1 + h lam  error at T
 -1  0.50      0.607       0.50     6.8e-02
 -1  0.10      0.905       0.90     9.7e-03
 -1  0.02       0.98       0.98     1.8e-03
-50  0.50   1.39e-11     -24.00     1.6e-03
-50  0.10    0.00674      -4.00     5.8e-04
-50  0.02      0.368       0.00     9.2e-05
```

漸化式そのものは何ら近似をしていないため、生じる誤差は各ステップ中も正弦波が静止しているとみなすホールドの誤差だけである。この誤差はステップ幅にほぼ比例して縮まり、$\lambda = -1$ では 0.068、0.0097、0.0018（それぞれ $h = 0.5$、0.1、0.02 のとき）となる。注目すべき行は $\lambda = -50$、$h = 0.5$ のケースであり、ステップ幅は状態が過去を忘れる時間スケール $1/|\lambda|$ の 25 倍もの長さになっている。それでも厳密な因子は 1.4×10^{-11}、誤差は 0.0016 に収まっているのに対し、因子 $1 + h\lambda = -24$ を用いていれば状態は爆発していただろう。12.4.1.2 節のモデルはこの事実に支えられている。減衰率が負に保たれている限り、訓練がどのようなステップサイズを選ぼうとも漸化式が発散することはない。

38.1.2.4 過渡的増大

固有値による安定性判定には、まだ 1 つ補足しておくべき注意点がある。すべてのモードが減衰するとき状態はゼロへと向かうが、その途中で長さ $\|\mathbf{x}(t)\|$ が単調に減少するとは限らないのである。次の三角行列がその例を与える。

$$\dot{\mathbf{x}} = \begin{pmatrix} -1 & 10 \\ 0 & -2 \end{pmatrix} \mathbf{x}, \qquad \mathbf{x}_0 = (0, 1),$$

固有値は -1 と -2 で、どちらも負である。第 2 行には x_2 しか現れないため、$x_2(t) = e^{-2t}$ となる。すると第 1 行は $\dot{x}_1 = -x_1 + 10e^{-2t}$ という入力付きの減衰方程式になり、Equation 38.7 から次式が得られる。

$$x_1(t) = 10 \int_0^t e^{-(t-s)} e^{-2s}\, ds = 10\left(e^{-t} - e^{-2t}\right).$$

第 1 座標は 0 から出発し、その導関数 $10(2e^{-2t} - e^{-t})$ がゼロになる $e^{-t} = \frac{1}{2}$ まで上昇する。そこ（$t = \log 2 \approx 0.69$）では第 1 座標は $10(\frac{1}{2} - \frac{1}{4}) = 2.5$、また $x_2 = \frac{1}{4}$ となるため、長さ 1 で出発した状態の長さは $\sqrt{2.5^2 + 0.25^2} \approx 2.51$ に達する。それ以降 x_1 は減衰に転じ、長期的にはより遅いモードの減衰率に従う $10e^{-t}$ のように減衰する（Figure 38.4）。

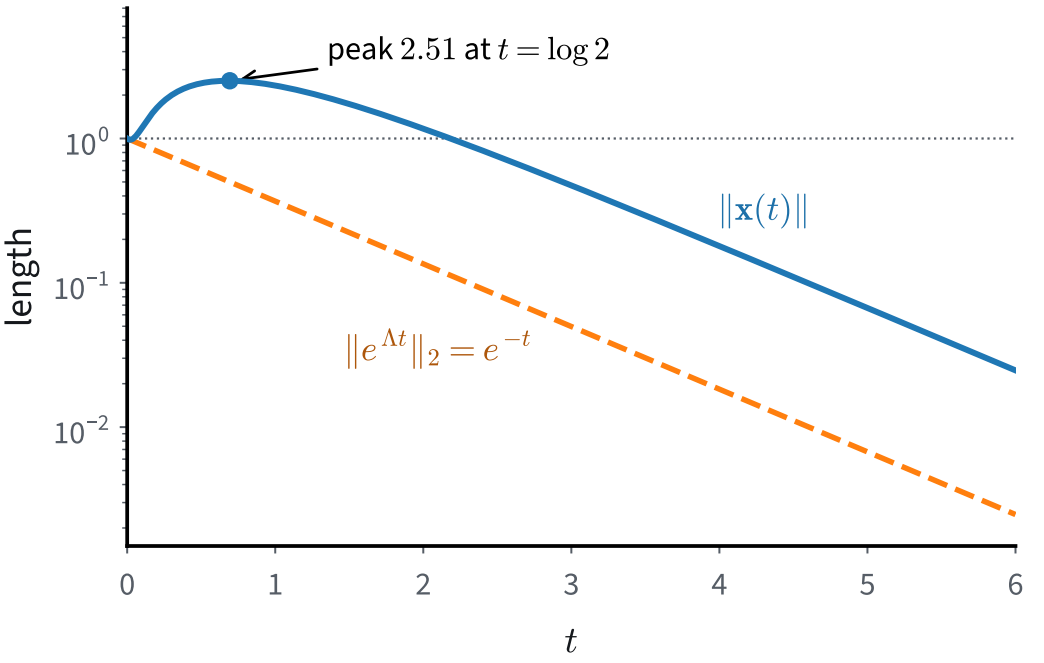


Figure 38.4: 三角行列からなる線形系の状態の長さ $\|\mathbf{x}(t)\|$（初期値 $\mathbf{x}_0 = (0, 1)$、対数軸）。2.51 まで上昇してから（$t = \log 2$）、$10e^{-t}$ のように減衰する。破線は $\|e^{\Lambda t}\|_2 = e^{-t}$（ただし $\Lambda = \mathrm{diag}(-1, -2)$）であり、同じ固有値を持つ対角行列のもとで単位長の初期状態が時刻 t にとりうる最大の長さを表す。2 本の曲線は最終的に同じ傾きに漸近するが、固有値が決めるのはこの終端の傾きだけである。

その理由は、離散時間の場合を詳しく扱った 33.4.2.3 節と同じである。固有ベクトル $\mathbf{v}_1 = (1, 0)$ と $\mathbf{v}_2 = (0.995, -0.0995)$ が張る直線のなす角はわずか 5.7° しかなく、短いベクトル $\mathbf{x}_0 = 10.0\,\mathbf{v}_1 - 10.05\,\mathbf{v}_2$ が 2 本の長いベクトルの差として表されているからである。これら 2 つの成分は異なる速さで減衰するため、両方が消え去る前に打ち消し合いが崩れる。固有値が与えるの

は漸近的な減衰率であって、過渡的増大（transient growth）の大きさではない。これは訓練曲線を眺めるときに心に留めておく価値がある。最終的に収束するモメンタムの実行であっても、最初の数ステップでは増大しうるのである（演習問題 5）。

38.1.2.5 不動点における線形化

閉形式解を持つ非線形の場はほとんどないが、場がゼロになる点の近傍では線形の場のように振る舞う。自律的な場の**不動点**とは、状態 $\mathbf{x}^\star$ であって $\mathbf{f}(\mathbf{x}^\star) = \mathbf{0}$ を満たすものであり、そこに置かれた粒子は決して動かない。連続微分可能な $\mathbf{f}$ と微小な変位 $\delta(t) = \mathbf{x}(t) - \mathbf{x}^\star$ に対して、テイラーの定理（34.2 節）より次式が得られる。

$$\dot{\delta} = \mathbf{f}(\mathbf{x}^\star + \delta) = \mathbf{J}\,\delta + o(\|\delta\|), \qquad \mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^\star).$$

不動点の近傍では、変位よりも速く消える剰余項を除けば、場はそのヤコビ行列（Jacobian matrix）$\mathbf{J}$ が定める線形の場に一致する。この剰余項は無視してよいのだろうか。ほとんどの場合は無視できる。不動点は、$\mathbf{J}$ のどの固有値も実部がゼロでないとき**双曲型**（hyperbolic）であるという。双曲型不動点に対しては、ここでは証明なしに引用する Hartman–Grobman の定理 (Hartman 1960; Grobman 1959) により、$\mathbf{x}^\star$ の近傍における非線形場の軌跡は、連続な座標変換を除いて $\dot{\delta} = \mathbf{J}\delta$ の軌跡と一致する。したがって、ヤコビ行列の固有値から先ほどの分類によって局所的な安定性が決まる。

連続時間の勾配降下法 $\dot{\mathbf{x}} = -\nabla \mathcal{L}(\mathbf{x})$ に当てはめれば、これは見慣れた事実を別の言葉で言い換えたものにすぎない。不動点は損失の臨界点（critical point）であり、ヤコビ行列はヘッセ行列（Hessian matrix）にマイナスを付けた $-\mathbf{H}$ である。したがって、あらゆる方向に正の曲率を持つ極小点は安定ノードとなり、損失の鞍点はフローにおいても鞍点となる。

本節で具体的に計算する唯一の非線形の場として、減衰振り子を取り上げよう。重力加速度と長さの比 g/ℓ が 1 になるように時間の単位を選ぶと、最下点から測った角度 ϕ は $\ddot{\phi} = -\sin\phi - c\dot{\phi}$ に従う（$c \geq 0$ は減衰係数）。角速度 $\omega = \dot{\phi}$ を導入すれば 1 階の系になり、そのヤコビ行列と合わせて書くと次のようになる。

$$\dot{\phi} = \omega, \qquad \dot{\omega} = -\sin\phi - c\,\omega, \qquad \mathbf{J}(\phi, \omega) = \begin{pmatrix} 0 & 1 \\ -\cos\phi & -c \end{pmatrix}.$$

不動点は $\omega = 0$ かつ $\sin\phi = 0$ を満たす点であり、$\phi = 0$ で真下に垂れ下がった状態と、$\phi = \pi$ で真上に倒立して釣り合った状態の 2 つである。

- 最下点では、$\mathbf{J}$ の特性多項式は $\lambda^2 + c\lambda + 1$ であり、その根は $\lambda = -c/2 \pm \mathrm{i}\sqrt{1 - c^2/4}$（$c < 2$ のとき）である。$c = 0.5$ では根は $-0.25 \pm 0.97\,\mathrm{i}$ となり、安定スパイラルである。これは手を離した振り子の減衰振動にあたり、周期約 $2\pi/0.97$ で揺れながら振幅が $e^{-t/4}$ のように減衰していく。
- 最上点では、特性多項式は $\lambda^2 + c\lambda - 1$ であり、その根 $(-c \pm \sqrt{c^2 + 4})/2$ の積は -1 であるため、任意の c に対して 1 つは正、もう 1 つは負となる。$c = 0.5$ では根は 0.78 と -1.28 であり、鞍点である。ある 1 本の曲線上の初期状態だけが最上点へと漸近する。これはちょうどよい初速度で振り上げられ、鉛筆を先端で立てたように釣り合って静止する振り子の運動に対応している。

ヤコビ行列とその固有値を求める同じ計算によって、ただ 1 つの例外を除き、あらゆる滑らかな場の不動点を分類できる。例外とは、実部が正の固有値は存在しないものの、虚軸上に固有値がある場合であり、このとき線形化だけでは安定性を判定できない。場 $\dot{x} = -x^3$ と $\dot{x} = x^3$ はど

ちらも原点でのヤコビ行列が 0 である。前者は安定で $x(t) = x_0/\sqrt{1 + 2x_0^2 t} \to 0$ となるが、後者は $t = 1/(2x_0^2)$ で無限大へと発散する。

この 2 つを見分けられたのは、たまたま方程式が解けたからにすぎない。1892 年の Lyapunov の着想は、解を求めずにこのような場合を判定する手段を与える。すなわち、運動によって減少することしかできない量を見つければよいのである。不動点 $\mathbf{x}^\star$ に対する **Lyapunov 関数**（Lyapunov function）とは、連続微分可能な状態の関数 V であって、$V(\mathbf{x}^\star) = 0$ かつ近傍の他のすべての点で $V(\mathbf{x}) > 0$ を満たし、いかなる軌跡に沿っても値が増加しないものをいう。

$$\frac{d}{dt} V(\mathbf{x}(t)) = \nabla V(\mathbf{x}(t))^{\top} \mathbf{f}(\mathbf{x}(t)) \le 0.$$

真ん中の式は連鎖律（chain rule）によるものであり、計算に必要なのは場だけであって軌跡そのものは不要である。V が小さい場所から出発した状態は、V がより大きい場所へは決して到達できないため、$\mathbf{x}^\star$ の近傍に留まり続け、不動点は安定となる。もし $\mathbf{x}^\star$ 以外の点で真の不等号が成り立つならば、近傍の状態は不動点へ収束する。先ほどの 2 つの 3 次の場に対して $V(x) = x^2$ ととると、その時間微分は $2x\dot{x}$ である。$\dot{x} = -x^3$ のもとではこれは $-2x^4$ となり、原点は安定である。一方、$\dot{x} = x^3$ のもとでは $+2x^4$ となり、今度はあらゆる軌跡に沿って x^2 が増大するため原点は不安定である。

実は、この考え方はすでに本節で使っていた。スパイラルでは長さの二乗 $V = \|\mathbf{x}\|^2$ が $\dot{V} = -V$ を満たしていた。連続時間の勾配降下法では、孤立した極小点での値を基準として測った損失が、その極小点の近傍における Lyapunov 関数になる。実際、$\frac{d}{dt}\mathcal{L} = \nabla \mathcal{L}^{\top} \dot{\mathbf{x}} = -\|\nabla \mathcal{L}\|^2$ が成り立つからである（演習問題 6）。そして振り子には運動エネルギーと位置エネルギーの和であるエネルギー $E = \tfrac{1}{2}\omega^2 + 1 - \cos\phi$ があり、最下点でゼロとなって次の速さで変化する。

$$\dot{E} = \omega\,\dot{\omega} + \sin\phi\,\dot{\phi} = \omega(-\sin\phi - c\,\omega) + \omega\sin\phi = -c\,\omega^2 \le 0.$$

減衰はエネルギーを奪うことしかできない。この事実は最下点から遠く離れた任意の大きさの揺れに対しても成り立ち、ヤコビ行列が何も語れない領域まで有効である。ただし折り返し点（$\omega = 0$）では一瞬だけエネルギーの変化が止まるため、不等号は狭義ではない。それにもかかわらず減衰振り子が最終的に静止することを示すにはもう一段の議論を要するが、ここでは割愛する。ヤコビ行列には機械的な計算手順があるのに対し、Lyapunov 関数にはそれがない。人間が見つけ出す必要があり、エネルギーや損失がその最初の候補となるのが常である。

ここまでは、すべて数式が仕事をしてくれた。しかしネットワークによって与えられる場には解の公式など存在せず、コンピュータにできるのは一歩ずつ進むこと、すなわち矢印を読み取り、少し進み、また矢印を読み取るという操作だけである。そのようなステップは模倣しようとしている本来の軌跡からどれほど逸れるのだろうか。また、元の方程式は破綻していないのにステップ計算だけが破綻することはあるのだろうか。

38.1.3　まとめ

常微分方程式（ordinary differential equation）は各点における速度を指定するものであり、最初の問いはそれによって運動が一意に定まるかどうかである。場が大域的リプシッツ連続であれば運動は定まる。すなわち、順方向にも逆方向にも各点を通る軌跡がただ 1 つ存在するため、フロー写像は全単射となり、場によってモデルを定義できる。線形系は $e^{\mathbf{A}t} = \mathbf{V} e^{\Lambda t} \mathbf{V}^{-1}$ によって解かれ、固有値の実部が減衰を、虚部が回転を決める。各ステップで一定に保たれた入力に対しては、方程式は厳密な漸化式へと変換され、元の方程式が安定であれば任意のステップサイズで安定となる。固有値が与えるのは長期的な減衰率であって過渡的増大の大きさではなく、双曲型不動点においてはヤコビ行列を通じて非線形の場を分類する。ヤコビ行列では判定できない場合

や変位が大きい場合には、エネルギーや損失のようにいかなる軌跡も増加させられない量である Lyapunov 関数が安定性を決定する。ただし、解の公式が存在しないときには、以上の議論だけでは軌跡を具体的に計算することはできない。

38.1.4 演習問題

1. **1 つの指数と 2 種類の破綻**。$\dot{x} = |x|^{\alpha}$（$\alpha > 0$）を考えよ。
   1. $0 < \alpha < 1$ のとき、$x(0) = 0$ を満たす解をすべて求め、$\alpha = \tfrac{1}{2}$ で Figure 38.2 の扇が得られることを確かめよ。Picard–Lindelöf の定理のどの仮定が、どの点で破れているか。
   2. $\alpha = 1$ のときは定理が適用できる。L はいくらか。また $x(0) = 0$ からの解は何か。
   3. $\alpha > 1$ かつ $x_0 > 0$ のとき、本文で $\dot{x} = x^2$ を解いたのと同様にして（$x^{1-\alpha}$ を微分せよ）方程式を解き、解が無限大に到達する時刻を求めよ。今度はどの仮定が破れているか。それにもかかわらず $x(0) = 0$ からの解は一意であることを示せ。（2 つの解はいずれも短い時間だけ $|x| \le 1$ 内に留まり、そこでは場は定数 α でリプシッツ連続である。短い区間上で証明の一意性のステップを適用し、再出発を繰り返す。）
   4. $\alpha = \tfrac{1}{2}$ の場合で時間を反転させよ。方程式 $\dot{x} = -\sqrt{x}$（$x \ge 0$）は、適切な単位のもとで底の穴から水が抜けるバケツの水位を表す（Torricelli の法則）。すべての解が有限時間で 0 に達し、そのままそこに留まることを示せ。空になったバケツを見てもいつ水が尽きたかはわからないが、フロー写像に関する系のどの主張が成り立たなくなっているか。
2. **軌跡の乖離**。ニューラル ODE が積分するような場（38.3 節）として、$\mathbf{f}(\mathbf{x}) = \mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$ の形の場を考えよ。
   1. $\mathbf{f}$ がスペクトルノルムに関して定数 $L = \|\mathbf{W}_2\|\,\|\mathbf{W}_1\|$ でリプシッツ連続であることを示せ。証明では tanh のどの性質を使うか。tanh を ReLU に置き換えるとこの上界は変化するか。
   2. $\mathbf{x}(t)$ と $\mathbf{y}(t)$ を、リプシッツ定数 L を持つ自律的な場の 2 つの軌跡とする。$\|\mathbf{x}(t) - \mathbf{y}(t)\|^2$ を微分し、次を示せ。
      $$e^{-Lt}\|\mathbf{x}(0) - \mathbf{y}(0)\| \le \|\mathbf{x}(t) - \mathbf{y}(t)\| \le e^{Lt}\|\mathbf{x}(0) - \mathbf{y}(0)\|.$$
      下界は、異なる点から出発した 2 つの軌跡について何を意味するか。また上下の評価は、Φ_T とその逆写像のリプシッツ定数について何を意味するか。
3. **重複固有値**。行列 $\mathbf{A} = \left(\begin{smallmatrix} \lambda & 1 \\ 0 & \lambda \end{smallmatrix}\right)$（$\lambda$ は実数）は単一の固有値とただ 1 つの固有方向しか持たないため、Equation 38.6 は適用できない。
   1. $\mathbf{A} = \lambda \mathbf{I} + \mathbf{N}$ と書き、$\mathbf{N}^2 = \mathbf{0}$ を用いて $e^{\mathbf{A}t} = e^{\lambda t}\left(\begin{smallmatrix} 1 & t \\ 0 & 1 \end{smallmatrix}\right)$ を示せ。ここで指数関数を分解してよいのはなぜか。
   2. $\mathbf{x}_0 = (0, 1)$ から出発すると第 1 座標は $t e^{\lambda t}$ となる。$\lambda < 0$ のとき、その最大値とそれをとる時刻を求めよ。どの λ に対して、因子 t は λ の符号だけから下される判定を覆すか。
   3. 38.1.2.4 節の行列の 2 つの固有値を互いに近づけよ。$\left(\begin{smallmatrix} -1 & 1 \\ 0 & -1-\varepsilon \end{smallmatrix}\right)$ および $\mathbf{x}_0 = (0, 1)$ に対して $x_1(t)$ とその $\varepsilon \to 0$ における極限を計算せよ。この極限で 2 つの固有方向のなす角はどうなるか。またそのことは、対角化できない行列について何を示唆しているか。
4. [code] **ゼロ次ホールド**。スカラー方程式 $\dot{x} = \lambda x + b\,u(t)$（$\lambda < 0$）を考えよ。（12.4 節の離散化規則に関する演習問題では、この方程式を漸化式に変換する他の 3 つの手法とこのホールドを比較している。）
   1. Equation 38.8 を 1 次元に特殊化して $\bar{a} = e^{h\lambda}$ および $\bar{b} = (e^{h\lambda} - 1)\,b/\lambda$ を導き、$\bar{b}$ の $\lambda \to 0$ における極限を求めよ。定数入力に対して、漸化式と元の方程式が同じ値に落ち着くことを示せ。

2. 時間変化する入力のサンプル $u(t_n)$ に対して、漸化式は階段関数で駆動される方程式を厳密に解いている。この階段関数が平均的には $h/2$ だけ遅れた入力に相当することを説明し、h の 1 次の範囲で時刻 T における誤差の大きさが $\frac{h}{2}\left|\int_0^T e^{\lambda(T-s)}\, b\, u'(s)\, ds\right|$ になることを導け。$\lambda=-1$、$b=1$、$u=\sin$、$T=2$ に対してこの値を計算せよ。38.1.2.3 節のコードセルが $\lambda=-1$ に対して出力した 3 つの誤差とどれほど近いか。
3. 同項のコードセルを変更し、各ステップの中点で入力をサンプリングした $u(t_n+h/2)$ を用いるようにせよ。このとき $\lambda=-1$ に対する誤差は h に応じてどのように減少するか、またそれはなぜか。

5. [code] **過渡的増大と非正規性。**●本文では過渡的増大を起こす行列を 1 つ示した。本問ではその増大が固有ベクトル間の角度にどう依存するかを調べ、さらにモメンタムにおいて同じ現象に再び出会う。
1. $\mathbf{A}_\nu = \left(\begin{smallmatrix} -1 & \nu \\ 0 & -2 \end{smallmatrix}\right)$ および $\mathbf{x}_0=(0,1)$ に対して $x_1(t)=\nu(e^{-t}-e^{-2t})$ を示し、x_1 の最大値とそれをとる時刻を求めよ。また、ν のうち $\|\mathbf{x}(t)\|$ が一度でも $\|\mathbf{x}_0\|$ を超える最小の値を数値的に求めよ。
2. $\mathbf{A}_\nu$ の 2 つの固有ベクトルのなす角を ν の関数として計算し、$\mathbf{x}_0$ を固有基底で表したうえで、2 つの大きな成分の打ち消し合いとしてこの増大を説明せよ。
3. $\max_t \|e^{\mathbf{A}_\nu t}\|_2$ を $\nu=1,5,10,20$ に対して数値的に計算し、$(0,1)$ からの軌跡のピーク値と比較せよ。前者が後者より小さくなりえないのはなぜか。
4. $\mathcal{L}(x)=\frac{\lambda}{2}x^2$ に対するヘビーボールモメンタムは漸化式 $x_{k+1}=(1+\beta-\eta\lambda)x_k-\beta x_{k-1}$ で表される。これは演習問題 6（38.2 節）において減衰振動子から導出されるものである。この漸化式は線形変換 $(x_{k+1},x_k)=\mathbf{M}(x_k,x_{k-1})$（ただし $\mathbf{M}=\left(\begin{smallmatrix}1+\beta-\eta\lambda & -\beta\\ 1 & 0\end{smallmatrix}\right)$）である。$\mathbf{M}$ が正規行列でないこと、すなわち $\mathbf{M}\mathbf{M}^\top\neq\mathbf{M}^\top\mathbf{M}$ を示せ（正規行列は直交する固有ベクトルを持ち、過渡的増大を起こさない。33.4.2.3 節参照）。$\beta=0.9$ とし、$\eta\lambda=0.1$ と $\eta\lambda=3$ における固有値を計算して、どちらの場合も絶対値が $\sqrt{\beta}\approx 0.95$ であることを確かめよ。両方を静止状態 $x_0=x_{-1}=1$ から実行せよ。どちらの実行で $\|(x_{k+1},x_k)\|$ が減衰する前に増大するか、何ステップ増大してピークはいくつになるか、そして最初の 1 ステップで損失はどうなるか。この結果は、モメンタムの最初の数反復を見て発散と判断することについて何を示唆しているか。

6. **勾配フローは周回できない。** $\mathcal{L}$ を 2 回連続微分可能かつ下に有界な関数とし、勾配フロー $\dot{\mathbf{x}}=-\nabla\mathcal{L}(\mathbf{x})$ を考えよ。
1. $\frac{d}{dt}\mathcal{L}(\mathbf{x}(t))=-\|\nabla\mathcal{L}(\mathbf{x}(t))\|^2$ を示せ。ここから、いかなる軌跡も一度離れた点には再び戻らないこと、したがって勾配フローは閉軌道も渦心点も持たないことを導け。
2. 上の恒等式を 0 から T まで積分し、$\min_{t\le T}\|\nabla\mathcal{L}(\mathbf{x}(t))\|^2\le(\mathcal{L}(\mathbf{x}_0)-\inf\mathcal{L})/T$ を示せ。この上界の導出において不要だった損失への仮定はどれか。
3. ゲーム $\min_{x_1}\max_{x_2} x_1x_2$（$x_1$ は降下し、x_2 は値 x_1x_2 を上昇させる）の場は $\dot{x}_1=-x_2$、$\dot{x}_2=x_1$、すなわち回転 $\mathbf{K}$ である。この場がいかなる関数の負の勾配でもないことを示せ。（$\partial f_1/\partial x_2$ と $\partial f_2/\partial x_1$ を比較せよ。）このとき小問 (a) は、単一の損失を訓練することと 2 人のプレイヤーを互いに競わせて訓練することの違いについて何を語っているか。

7. **線形化の限界。**
1. 場 $\dot{x}=x^2$ は、本文の 2 つの 3 次の場と同様に原点でのヤコビ行列が 0 である。38.1.1.3 節で求めたその解は任意の初期値 x_0 に対して成り立つ。状態が原点へと近づくような初期値を求めよ。原点は安定か、不安定か、それとも片側からのみ安定か。Hartman–Grobman の定理はこれら 3 つの場について何を述べるか。
2. 減衰振り子において、最下点の静止点がスパイラルからノードへと変化する減衰係数 c を求めよ。$c=0.5$、$c=2$、$c=4$ に対して最下点近傍の相図の概形を描き、手を離した振り子がそれぞれの場合にどう動くかを述べよ。
3. 損失 $\mathcal{L}(w_1,w_2)=\frac{1}{2}(w_1w_2-1)^2$ は最小構成の深層線形ネットワークの損失であり、

2 つの重みの積を目標値 1 に当てはめるものである。その勾配フロー $\dot{\mathbf{w}} = -\nabla \mathcal{L}(\mathbf{w})$ を書き下し、すべての不動点を求め、原点を分類せよ。曲線 $w_1 w_2 = 1$ 上のすべての点において、ヤコビ行列の固有値が 0 と $-(w_1^2 + w_2^2)$ になることを示せ。不動点が曲線をなしているとき、その上のいかなる点も双曲型になりえないのはなぜか。また、それでも負の固有値から何がわかるか。

38.2 数値ソルバ

ネットワークを訓練したことがあれば誰でも、学習率（learning rate）が大きすぎたために計算が発散して爆発する様子を目にしたことがあるだろう。鋭い谷やプラトーを持つ損失曲面（loss surface）のせいにしたくなるかもしれない。しかし、考えられる限り最も単純な損失上でも同じことが起こる。ボウル型の損失 $\mathcal{L}(x) = 25x^2$ に対する勾配降下法（gradient descent）は更新ステップ $x \leftarrow x - \eta \cdot 50x$ をとるが、このボウルは唯一の最小点を持ち何ひとつ難しい点がないにもかかわらず、$\eta = 0.04$ を超えるあらゆる学習率で反復点は際限なく増大する。損失に問題がないとすれば、何が問題なのだろうか。

その答えは、より古い分野からもたらされる。勾配降下法とは、「傾きに比例する速さで下り方向へ進め」と述べる微分方程式（differential equation）$\dot{x} = -50x$ を 1 ステップずつ解く操作である。これは 38.1 節で解いた減衰方程式を 50 倍速くしたものであり、その厳密解 $x(t) = e^{-50t} x_0$ はどのような初期値から出発してもゼロへと減衰する。Euler は 1768 年に、矢印の向きに短い時間 h だけ進んでは矢印を読み直すことでこのような方程式を解く手法を提案したが、学習率はこの h の役割を果たしている。1 ステップ進めるごとに x は $1 - 50h$ 倍になる。$h = 0.01$ ではこの因子は 0.5 であり、計算される解は真の解と同様に減衰する。一方、$h = 0.05$ では -1.5 となり（厳密な因子は $e^{-2.5} = 0.08$ である）、計算される解は符号を反転させながら増大していく。方程式自体は安定であるが、ステップが安定ではないのである。前節では方程式の固有値（eigenvalue）がその厳密な運動を決めていたが、本節では同じ固有値が、その運動を追跡できるステップ幅の上限を決めることになる。

深層学習は、ステップを進めるあらゆる場面で、方程式とその離散ステップとの間にあるこの乖離に直面する。すなわち、勾配降下法（9.1 節）、残差ネットワーク（residual network）の各層（7.4 節）、そして拡散モデル（diffusion model）やフローマッチング（flow matching）モデルのサンプラ（17.5 節）である。サンプラは各ステップでネットワークを 1〜2 回評価するため、ステップ数が画像 1 枚を生成するコストを決める。

逐次的な数値解法の価値は、3 つの問いによって決まる。真の解からどれほど逸れるのか。同じ計算量でそのずれをどうすれば減らせるのか。そして、誤差が減衰せずに増大し始めるまでにステップ幅をどこまで大きくとれるのか。実は、これら 3 つの問いすべてにたった 1 つの対象が答えてくれる。1 ステップで解に掛けられる因子であり、先のボウルに対するオイラー法では $1 - 50h$ であった量である。本節の最後では勾配降下法に立ち返り、より優れたソルバを用いれば訓練はどうなるのかを問う。

38.2.1 前進オイラー法

軌跡（trajectory）とは、始点に道中で蓄積された速度を足し合わせたものである。短いステップの上で、Euler の提案は考えうる限り最も素朴な近似をとる。すなわち、速度がステップの始点での値をずっと保ち続けるとみなすのである。まず、答えのわかっている方程式でこの手法がどう振る舞うかを見てみよう。

$\dot{x} = x$ を $x_0 = 1$ から $T = 1$ まで解くことを考える。真の解は $e = 2.718$ である。区間を n 個のステップ（各ステップの長さ $h = 1/n$）に分割し、各ステップではその始点での速度で進むと

すると、$x \leftarrow x + hx = (1+1/n)x$ となる。n ステップ後には $x_n = (1+1/n)^n$ が得られる。これは 1 ドルを年利 100%、年 n 回複利で運用したときの元利合計にあたる。

n	1	2	4	8	16	32	365
$(1+1/n)^n$	2.000	2.250	2.441	2.566	2.638	2.677	2.7146
誤差	0.718	0.468	0.277	0.153	0.080	0.041	0.0037

隣り合う誤差の比は $1.53, 1.69, 1.82, 1.90, 1.95$ となっており、ステップ数を 2 倍にすると誤差は次第に半分へと近づいていく。それでも、毎日複利計算したところで e ドルには 3 分の 1 セント以上届かない。

この収束の遅い極限にはすでに前節で出会っている。複利型の式 $(\mathbf{I} + t\mathbf{A}/n)^n$（38.1 節）は、名前を付ける前のオイラー法そのものであり、線形な場に沿って n ステップ（ステップ幅 t/n）進めたものにほかならない。そこで求めた展開 $(1+\lambda t/n)^n \approx e^{\lambda t}(1 - \frac{(\lambda t)^2}{2n})$ において $\lambda t = 1$ とおけば、先の 1 ドルの誤差が得られる。その大きさは約 $e/(2n) = 1.36\,h$ である。このように h に比例する誤差を、以下では 1 次（次数 1）と呼ぶ。

一般のベクトル場 $\mathbf{f}(\mathbf{x}, t)$ に対し、ステップサイズ（step size）を h、時間グリッドを $t_n = nh$（これ以降、n はそれまでに進んだステップ数を表す）とおくと、**前進オイラー法**（forward Euler）は次式で定義される。

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}(\mathbf{x}_n, t_n). \tag{38.9}$$

この手法は現在地での矢印に沿って時間 h だけ進み、そこで再び場を読み取る。言い換えれば、真の解のテイラー展開（Taylor expansion）を 1 次の項で打ち切ったものであり、1 ステップあたりの誤差は切り捨てた最初の項になる。解に沿って $\|\ddot{\mathbf{x}}(t)\| \le M$ が成り立つならば、次が成り立つ。

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + h\,\mathbf{f}(\mathbf{x}(t_n), t_n) + \mathbf{r}_n, \qquad \|\mathbf{r}_n\| \le \tfrac{Mh^2}{2}.$$

剰余項 $\mathbf{r}_n$ は**局所誤差**（local error）であり、真の解の上から出発したときに 1 ステップで生じるずれを表す。本当に知りたいのは**大域誤差**（global error）、すなわち固定された終端時刻 T まで全ステップを進めた後の真の解からの距離である。局所誤差が h^2 のオーダーであっても、大域誤差が h^2 のオーダーになるとは限らない。T に到達するまでには T/h ステップを要し、それぞれの局所誤差が累積するうえ、各ステップがすでに生じた誤差を増幅しうるからである。こうして h のべきが 1 つ失われる。その証明は短く、定数そのものよりも証明の構造が重要である。

命題（オイラー法は次数 1 で収束する）。 **$\mathbf{f}$ は定数 L で $\mathbf{x}$ に関してリプシッツ連続であるとする。すなわち、$\|\mathbf{f}(\mathbf{x},t) - \mathbf{f}(\mathbf{y},t)\| \le L\|\mathbf{x}-\mathbf{y}\|$ がすべての $\mathbf{x}$、$\mathbf{y}$、t に対して成り立つとする。また、$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},t)$ の初期値 $\mathbf{x}(0) = \mathbf{x}_0$ からの解が $\|\ddot{\mathbf{x}}(t)\| \le M$ を区間 $[0,T]$ 上で満たすとする。このとき、同じ初期値から出発するオイラー法の反復点 Equation 38.9 は次を満たす。**

$$\max_{n \le T/h} \|\mathbf{x}_n - \mathbf{x}(t_n)\| \;\le\; \frac{Mh}{2L}\left(e^{LT} - 1\right) \;=\; O(h).$$

証明。 $\varepsilon_n = \|\mathbf{x}_n - \mathbf{x}(t_n)\|$ とおき、更新 Equation 38.9 から上のテイラー展開の恒等式を引くと、次を得る。

$$\varepsilon_{n+1} \le \underbrace{\|\mathbf{x}_n - \mathbf{x}(t_n)\| + h\,\|\mathbf{f}(\mathbf{x}_n,t_n) - \mathbf{f}(\mathbf{x}(t_n),t_n)\|}_{\le (1+hL)\,\varepsilon_n \text{ by Lipschitz}} + \underbrace{\|\mathbf{r}_n\|}_{\le Mh^2/2}.$$

各ステップは既存の誤差を高々 $1+hL$ 倍し、そこへ高々 $Mh^2/2$ を加える。$\varepsilon_0 = 0$ から漸化式を展開すると等比級数の和になる。

$$\varepsilon_n \le \frac{Mh^2}{2}\sum_{j=0}^{n-1}(1+hL)^j = \frac{Mh^2}{2}\cdot\frac{(1+hL)^n - 1}{hL} \le \frac{Mh}{2L}\left(e^{LT}-1\right),$$

ここで $(1+hL)^n \le e^{nhL} \le e^{LT}$ を用いた。■

$\dot{x} = x$ を区間 $[0,1]$ 上で考えると、$M = \max|\ddot{x}| = e$ かつ $L = 1$ であるから、この上界は $\frac{e}{2}(e-1)h = 2.3\,h$ となり、真の誤差 $1.36\,h$ に対して次数は正しいものの定数は悲観的である。この上界の構造はあらゆるソルバで繰り返し現れる。1 ステップあたり h^{p+1} のオーダーの局所誤差は、大域誤差としては h^p のオーダーになる。これは T/h ステップにわたって累積し、過去の誤差をダイナミクスがどれほど強く増幅するかを測る因子 e^{LT} によって膨らむためである。大域誤差が h^p のオーダーとなる手法は**次数**（order）p を持つといい、オイラー法の次数は 1 である。

因子 e^{LT} が大雑把な部分である。定数 L は場の傾きの急さだけを記録し、どちら向きに傾いているかは区別しないため、この上界では軌跡を互いに引き離す場と引き寄せる場を見分けられない。ボウルの減衰方程式 $\dot{x} = -50x$ を $[0,1]$ 上で考えると、この因子は $e^{50} \approx 5\times 10^{21}$ となり、小さなステップ幅をとれば減衰をうまく追跡できるにもかかわらず、コンピュータが実行しうるいかなるステップ幅でも役に立たない上界になってしまう。収縮する場には専用の解析が必要であり、次数を高める手法を見た後でそこへ進むことにする。

38.2.2 ルンゲ・クッタ法

オイラー法は、軌跡が曲がって離れていくにもかかわらずステップ始点での傾きに固執する。これに対し、ルンゲ・クッタ法はステップ内部の複数の点で傾きを測り、それらの測定値を組み合わせる。Figure 38.5 に、ステップ幅 $h = 1$ で $\dot{x} = x$ を $x = 1$ から 1 ステップ進めた様子を示す。

38.2.2.1 中点法、ホイン法、RK4

最も単純な改良である**中点法**（midpoint method）は、中点での傾きを測るためだけにオイラー法で半ステップ進み、得られた傾きを使って本来のステップを進める。時間に依存しないベクトル場を $\mathbf{f}(\mathbf{x})$ と書けば、更新式は $\mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}\big(\mathbf{x}_n + \tfrac{h}{2}\mathbf{f}(\mathbf{x}_n)\big)$ となる。1 次元でテイラー展開を 1 行計算すれば、なぜこれで精度が上がるのかがわかる。

$$x + hf\big(x + \tfrac{h}{2}f(x)\big) = x + hf + \tfrac{h^2}{2}f'f + O(h^3).$$

真の解の展開は $x(t+h) = x + h\dot{x} + \frac{h^2}{2}\ddot{x} + O(h^3)$ であり、ここで $\ddot{x} = \frac{d}{dt}f(x(t)) = f'f$ であるから、欠落していた h^2 の項が正しく一致している。したがって 1 ステップあたり 2 回の場評価で、局所誤差は $O(h^3)$、大域的な次数は 2 となる。

中点法は最初の評価を半ステップに使った。これに対し、**ホイン法**（Heun's method）は試行的にオイラー法で 1 ステップ分すべて進み、ステップの両端における傾きを平均する。

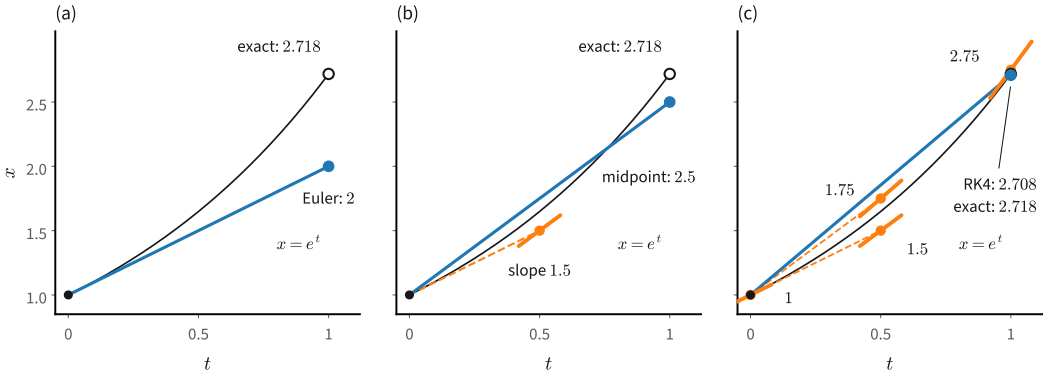


Figure 38.5: ステップ幅 $h = 1$ による $\dot{x} = x$（初期値 $x = 1$）の 1 ステップ。(a) オイラー法は始点の傾きに沿って進み、2 に到達する。(b) 中点法は中点での傾き 1.5 を測るためだけに半ステップ進み、その傾きを用いて 2.5 に到達する。(c) RK4 は 4 つの傾き 1、1.5、1.75、2.75 を測り、重み $\frac{1}{6}(1, 2, 2, 1)$ で平均して 2.708 に到達する。厳密値は $e = 2.718$ である。

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \tfrac{h}{2}\Big[\mathbf{f}(\mathbf{x}_n) + \mathbf{f}\big(\mathbf{x}_n + h\,\mathbf{f}(\mathbf{x}_n)\big)\Big]. \tag{38.10}$$

中点法が積分における中点則（midpoint rule）に対応するなら、ホイン法は台形則（trapezoid rule）に対応する。同じ 1 行のテイラー展開から、同じ 2 回の評価で同じ次数 2 を持つことがわかる。この 2 つのうち覚えておくべきなのはホイン法である。多くの拡散モデルのサンプラがこの手法を採用しているからである（17.5 節）。

4 回の傾きの測定を用いると、1901 年の Kutta に遡る科学技術計算の標準的な汎用手法である**古典的ルンゲ・クッタ法（RK4）**（classical Runge–Kutta）が得られる。

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}(\mathbf{x}_n,\, t_n),\\
\mathbf{k}_2 &= \mathbf{f}(\mathbf{x}_n + \tfrac{h}{2}\mathbf{k}_1,\, t_n + \tfrac{h}{2}),\\
\mathbf{k}_3 &= \mathbf{f}(\mathbf{x}_n + \tfrac{h}{2}\mathbf{k}_2,\, t_n + \tfrac{h}{2}),\\
\mathbf{k}_4 &= \mathbf{f}(\mathbf{x}_n + h\,\mathbf{k}_3,\, t_n + h),\\
\mathbf{x}_{n+1} &= \mathbf{x}_n + \tfrac{h}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right).
\end{aligned} \tag{38.11}$$

この更新式は、中点での 2 つの測定値に 2 倍の重みを付けて 4 つの傾きを平均している。これらの重みは 3 次多項式を厳密に積分するシンプソン則 $\int_a^b g \approx \frac{b-a}{6}\big(g(a) + 4g(\frac{a+b}{2}) + g(b)\big)$ の重みそのものであり、中点に割り当てられた重みが 2 回の測定に等分されている。

38.2.2.2 安定性関数

なぜ次数が 4 になるのだろうか。線形な場の上では、1 回の計算で答えが出る。スカラー方程式 $\dot{y} = \lambda y$（λ は実数または複素数）を考えよう。これは**テスト方程式**（test equation）と呼ばれ、あらゆる線形場を一度に代表する方程式である。というのも、$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ の各モードは、λ を行列 $\mathbf{A}$ の固有値としてこの方程式に従うからである（38.1 節）。真の解には因子 $e^{h\lambda}$ が時間 h の間に掛かる。一方、各数値手法は 1 ステップごとに y に $z = h\lambda$ の定まった関数を掛けるため、その関数を e^z と比較できる。

命題（安定性関数）。テスト方程式において $z = h\lambda$ とおくと、各手法の 1 ステップは y にそれぞれの安定性関数（stability function）$R(z)$ を掛ける。

$$R_{\text{Euler}}(z) = 1 + z, \qquad R_{\text{mid}}(z) = R_{\text{Heun}}(z) = 1 + z + \frac{z^2}{2}, \\ R_{\text{RK4}}(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}. \tag{38.12}$$

証明。 RK4 では 4 つの測定値はそれぞれ直前の測定値を代入することで得られ、$k_1 = \lambda y$、$k_2 = \lambda y(1 + \tfrac{z}{2})$、$k_3 = \lambda y(1 + \tfrac{z}{2} + \tfrac{z^2}{4})$、$k_4 = \lambda y(1 + z + \tfrac{z^2}{2} + \tfrac{z^3}{4})$ となる。したがって更新後の値は $y + \tfrac{z}{6}\left(6 + 3z + z^2 + \tfrac{z^3}{4}\right) y$ となり、上の多項式と y の積に一致する。オイラー法では $y + zy$ となる。中点法では $y + h\lambda(y + \tfrac{z}{2}y)$、ホイン法では $y + \tfrac{z}{2}(y + (1+z)y)$ となり、いずれも $(1 + z + \tfrac{z^2}{2})y$ である。■

各 R は厳密な因子 e^z のテイラー多項式をその手法の次数で打ち切ったものになっている。RK4 では $e^z - R_{\text{RK4}}(z) = z^5/120 + O(z^6)$ であるため、この方程式において 1 ステップの誤差は $O(h^5)$、T/h ステップ後の誤差は $O(h^4)$ となる。4 回連続微分可能かつ $\mathbf{x}$ に関してリプシッツ連続な非線形の場に対しても、より長いテイラー展開の照合によって同じ次数が得られ、中点法とホイン法は次数 2、RK4 は次数 4 となる (Hairer et al. 1993)。4 重に入れ子になった評価の展開は何ページにも及び、そこから得られる示唆も少ないため、ここでは証明しない。本項で押さえておくべき対象は安定性関数である。e^z に $z = 0$ の近傍でどれほど密接に追随するかがその手法の精度であり、ゼロから離れた場所でどれほど大きくなるかがどのステップ幅で安定になるかを決める。

38.2.2.3 同一コストでの次数比較

残る課題は、実際に次数を測定し、場の評価回数を揃えて各手法を比較することである。時間に依存しない場 `f` に対して、この手法族の両端にあたる 2 つのソルバはコードではそれぞれ数行で書ける。

```
def euler(f, x0, T, n):
    """Forward Euler: n steps of size h = T/n."""
    x, h = np.asarray(x0, dtype=float), T / n
    for _ in range(n):
        x = x + h * f(x)
    return x

def rk4(f, x0, T, n):
    """Classical Runge-Kutta: four slope measurements per step."""
    x, h = np.asarray(x0, dtype=float), T / n
    for _ in range(n):
        k1 = f(x)
        k2 = f(x + 0.5 * h * k1)
        k3 = f(x + 0.5 * h * k2)
        k4 = f(x + h * k3)
        x = x + (h / 6) * (k1 + 2 * k2 + 2 * k3 + k4)
    return x
```

両対数軸上では error $\approx Ch^p$ は傾き p の直線になるため、測定される傾きは 1 と 4 になるはずである。これを Figure 38.1 のスパイラル $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$（固有値は $-0.5 \pm \mathrm{i}$）で検証しよう。この軌

跡上の点は単位速度で回転しながら減衰率 $\frac{1}{2}$ で長さを失うため、$\mathbf{x}_0 = (2, 0)$ から出発したときの $T = 2$ における厳密な状態は $e^{-1}(2\cos 2,\, 2\sin 2) = (-0.306,\, 0.669)$ であり、その長さは 0.736 である（38.1 節）。

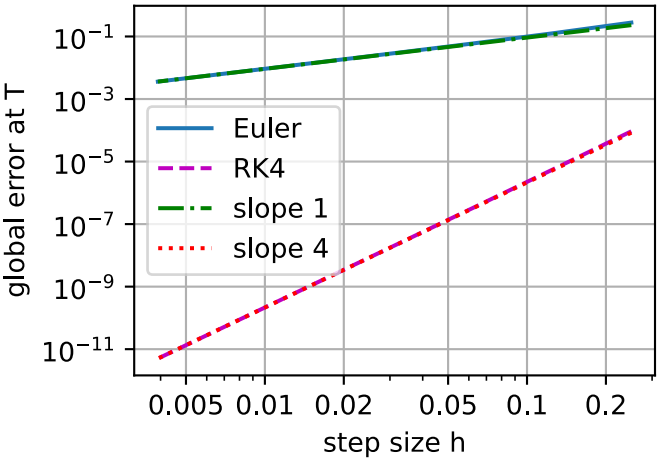
```
A = np.array([[-0.5, -1.0], [1.0, -0.5]])      # the spiral, eigenvalues -0.5 +- i
x0, T = np.array([2.0, 0.0]), 2.0
x_true = np.exp(-1) * np.array([2 * np.cos(2), 2 * np.sin(2)])  # e^{AT} x0
```

線形な場の上では、傾きだけでなく定数まで予測できる。このスパイラル上では、厳密な運動と同様にどちらのソルバの 1 ステップも回転とスケーリングの合成になる。したがって、各モードの因子がある割合だけずれるソルバは、真の終端点の長さ 0.736 に対しても同じ割合だけずれる。オイラー法におけるその割合は、複利計算で求めた $|\lambda T|^2/(2n)$ である。ここでは $|\lambda T|^2 = 5$ であるから、オイラー法の誤差は終端の長さの $2.5/n$ 倍、すなわち $1.84/n$ になるはずである。

RK4 における割合は安定性関数から導かれる。1 ステップあたりのずれの割合は $|z|^5/120$（ただし $|z| = |\lambda T|/n = \sqrt{5}/n$）であり、$n$ ステップ分の誤差が足し合わされる。その合計は終端の長さの $5^{5/2}/(120\, n^4)$ 倍、すなわち $0.343/n^4$ となる。次のコードセルでは、ステップ幅を 6 回半減させ、8 ステップから 512 ステップまで計算する。

```
f = lambda x: A @ x                            # the spiral field
ns = 2 ** np.arange(3, 10)                     # 8, 16, ..., 512 steps
hs = T / ns
err_eu = np.array([np.linalg.norm(euler(f, x0, T, n) - x_true) for n in ns])
err_rk = np.array([np.linalg.norm(rk4(f, x0, T, n) - x_true) for n in ns])
print('measured slope, Euler:', f'{np.polyfit(np.log(hs), np.log(err_eu), 1)[0]:.3f}')
print('measured slope, RK4  :', f'{np.polyfit(np.log(hs), np.log(err_rk), 1)[0]:.3f}')
print(f'error at 512 steps: Euler {err_eu[-1]:.1e}, RK4 {err_rk[-1]:.1e}')
print('RK4 with 128 steps (the same 512 evaluations):',
      f'{np.linalg.norm(rk4(f, x0, T, 128) - x_true):.1e}')
d2l.plot(hs, [err_eu, err_rk, err_eu[-1] * (hs / hs[-1]),
              err_rk[-1] * (hs / hs[-1]) ** 4],
         'step size h', 'global error at T', xscale='log', yscale='log',
         legend=['Euler', 'RK4', 'slope 1', 'slope 4'])
```

```
measured slope, Euler: 1.040
measured slope, RK4  : 4.022
error at 512 steps: Euler 3.6e-03, RK4 5.0e-12
RK4 with 128 steps (the same 512 evaluations): 1.3e-09
```



直線当てはめで得られた傾きは 1.04 と 4.02 であり、1 と 4 をわずかに上回っている。これは当てはめに $h = 0.25$ という粗いステップも含まれており、展開の次の項がまだ効いているためである。512 ステップにおける誤差は 3.6×10^{-3} と 5.0×10^{-12} であり、$1.84/n$ と $0.343/n^4$ の予測と一致する。もっとも、傾きだけを見るとコストが隠れてしまう。RK4 は 1 ステップあたり場を 4 回評価するため、公平に比較するにはステップ数を 4 分の 1 にしなければならない。同じ 512 回の場評価、すなわち 128 ステップで比較すると、RK4 の誤差は 1.3×10^{-9} に達し、オイラー法より 6 桁小さい。オイラー法の法則 $1.84/n$ でこれに並ぶには 10 億ステップ以上が必要になる。一般に、次数 p の手法では $\varepsilon^{-1/p}$ に比例するステップ数が誤差 ε の達成に必要となる。

実用的なソルバは、計算を進めながらステップサイズも適応的に選ぶ。Dormand–Prince 法のような**埋め込み型ペア**（embedded pair）は、同じ場評価から次数 4 と次数 5 の解を同時に計算し、両者の差を局所誤差の推定値として用いる (Dormand and Prince 1980)。そして推定誤差が大きい箇所ではステップを短くし、小さい箇所では長くする。演習問題 1 では、このような制御器のうち最も単純なものを構築する。

ここまでの手法はいずれも各ステップを白紙から始め、直前のステップで測定した傾きを捨ててしまう。これに対し、**多段階**（multistep）法は過去の測定値を保持する。最も単純な 2 段階の**アダムス・バッシュフォース法**（Adams–Bashforth method）は、各ステップで場を 1 回だけ評価し、直近 2 回の評価値から外挿する。

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\left[\tfrac{3}{2}\,\mathbf{f}(\mathbf{x}_n) - \tfrac{1}{2}\,\mathbf{f}(\mathbf{x}_{n-1})\right]. \tag{38.13}$$

角括弧の中身は $\mathbf{f}(\mathbf{x}_n) + \tfrac{1}{2}[\mathbf{f}(\mathbf{x}_n) - \mathbf{f}(\mathbf{x}_{n-1})]$ と書け、2 つの評価値を通る直線をさらに半ステップ先まで延長した値になっている。これはこれから進むステップの中点における傾きの推定値にあたるため、2 回目の場評価を行う代わりに記憶しておいた値から中点の傾きを得る中点法に相当し、オイラー法と同じコストで次数 2 を達成する。ただし失われる点も 2 つある。最初のステップには 1 つ前の値がないため 1 段階法で進める必要があり、また重み $\tfrac{3}{2}$ と $-\tfrac{1}{2}$ は等間隔ステップでしか成り立たないためステップサイズを変える際には再計算が必要になる。しかし 1 回の評価が大規模ネットワークの順伝播 1 回分にあたる場面では、計算量の節約がこの両方の欠点を上回るため、17.5 節の多段階サンプラはこの更新ステップを基盤として構築されている。

38.2.3　硬さと陰的手法

次数が測るのは、ステップ幅が小さいときに 1 ステップがどれほど精密かという性質である。これに対し、安定性は別の問いを立てる。すなわち、多数のステップにわたる誤差が減衰して消えるのか、それとも雪だるま式に増幅するのかという問いである。オイラー法や RK4 では、どれほど低い精度しか求めていなくても、この安定性によってステップサイズに上限が課される。

この問題を示す最小の系は、互いに分離した 2 つのモードを持つ $\dot{\mathbf{x}} = \operatorname{diag}(-50, -1)\,\mathbf{x}$ を初期値 $(1, 1)$ から $T = 1$ まで追跡するものである。その時刻までに速いモードは $e^{-50} \approx 10^{-22}$ まで減衰し、遅いモードは $e^{-1} = 0.368$ に留まっている。$t \approx 0.1$ 以降は遅いモードしか残っておらず、$h = 0.1$ のステップ幅でも十分に追跡できるはずである。ところが速い座標は冒頭の方程式 $\dot{x} = -50x$ そのものであり、積分を続ける限りオイラー法は毎ステップそこに $1 - 50h$ を掛け続ける。興味のある解が変化する速さに比べてヤコビ行列の固有値の一部に絶対値の大きな負の実部があるとき、その系は**硬い**（stiff）という。とうの昔に減衰し切ったモードが、依然としてステップ幅を決め続けてしまうのである。

どのようなステップ幅なら許されるのだろうか。それに答えるのが安定性関数である。テスト方程式 $\dot{y} = \lambda y$ の解が減衰するのは $\operatorname{Re}\lambda < 0$ のとき、かつそのときに限られるが、数値手法による解は初期値 y_0 から $R(z)^n y_0$（n ステップ後）へと変化する。したがって、計算されるモードが減衰する必要十分条件は $|R(z)| < 1$ である。前進オイラー法ではこれは $|1 + z| < 1$ であり、実数の $\lambda < 0$ に対しては $-2 < h\lambda < 0$、すなわち次式となる。

$$h < \frac{2}{|\lambda|}.$$

冒頭のボウルでは $2/|\lambda| = 2/50 = 0.04$ となり、これを超えると反復点が増大したあの学習率と一致する。

この上限より下では、オイラー法の誤差上界に欠けていた収縮する場の解析も同時に手に入る。1 ステップ進めると、すでに生じた誤差には $|1 + h\lambda| < 1$ が掛かるようになるが、一般の証明ではこの因子を $1 + hL$ でしか上から抑えられなかった。$h \le 1/|\lambda|$ のとき、あの証明の等比級数の和は $1/(h|\lambda|)$ 以下に収まり、誤差上界は指数因子が消えた $Mh/(2|\lambda|)$ となる。ボウルの減衰方程式を $x_0 = 1$ から解く場合、$M = 2500$ であるからこの上界は $25\,h$ となり、真の誤差の 3 倍以内に収まる。

38.2.3.1　後退オイラー法

ステップ幅を短くするのは 1 つの解決策である。もう 1 つの方法は一見奇妙に見えるが、ステップの**終点**における傾きを使ってオイラー法のステップを進めるというものである。

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}(\mathbf{x}_{n+1}, t_{n+1}).$$

これが**後退オイラー法**（backward Euler）である。未知数 $\mathbf{x}_{n+1}$ が両辺に現れるという厄介な点があるが、その対処はこの手法が何をもたらすかを確認するまで後回しにしよう。

命題（後退オイラー法は任意のステップ幅で安定である）。 $\operatorname{Re}\lambda < 0$ **であるテスト方程式において、後退オイラー法の安定性関数は** $R(z) = 1/(1 - z)$ **であり、** $|R(z)| < 1$ **がすべてのステップ幅** $h > 0$ **に対して成り立つ。**

証明。 1 ステップの更新は $y_{n+1} = y_n + h\lambda y_{n+1}$ であるから $y_{n+1} = y_n/(1 - z)$ となり、$|1 - z| \ge \operatorname{Re}(1 - z) = 1 - \operatorname{Re} z > 1$ が $\operatorname{Re} z < 0$ のときに成り立つ。$\blacksquare$

その理由は代数計算なしでも理解できる。前進オイラー法は現在地の傾きで外挿するため、急峻なモードの上でその傾きを長いステップにわたって保つと、ゼロを大きく通り越してしまう。一方、後退オイラー法は、その点自身の傾きを逆向きにたどると現在地に戻るような点を探す。ほぼ減衰し切った点では傾きもほぼゼロであるため、どれほど長いステップをとっても行き過ぎることはない。

ここで、先ほど後回しにした厄介な点に向き合おう。未知の点で場を評価するため、各ステップで方程式を解く必要が生じる。場が線形であれば連立一次方程式（linear system）を 1 回解き、そうでなければニュートン法の反復を数回行うことになる（演習問題 3）。このような手法は**陰的**（implicit）と呼ばれ、公式によって新しい状態を直接計算する前進オイラー法や RK4 は**陽的**（explicit）と呼ばれる。陽的手法のステップ幅が精度ではなく安定性によって制限されてしまう場面では、陰的ステップは方程式を解く手間に見合う価値を持つ。

それでは 2 モード系に対し、5 通りのステップサイズで両手法を実行してみよう。後退オイラー法ではステップごとに連立一次方程式を 1 回解く。最初の行の結果は計算するまでもなく予測できるはずである。$h = 0.1$ では、前進オイラー法は速い座標に $1 - 5 = -4$ を 10 回掛けることになる。

```
def backward_euler_linear(A, x0, T, n):
    """Backward Euler on dx/dt = A x: solve (I - h A) x_new = x at each step."""
    x, h = np.asarray(x0, dtype=float), T / n
    for _ in range(n):
        x = np.linalg.solve(np.eye(len(x)) - h * A, x)
    return x

A_stiff = np.diag([-50.0, -1.0])              # one fast mode, one slow mode
x0_stiff = np.array([1.0, 1.0])
T = 1.0
print(f'exact at T = 1: fast {np.exp(-50):.1e}, slow {np.exp(-1):.4f}')
print(f'{"h":>5} {"fwd fast":>10} {"fwd slow":>9} {"bwd fast":>10} {"bwd slow":>9}')
for n in [10, 20, 25, 50, 100]:
    fwd = euler(lambda x: A_stiff @ x, x0_stiff, T, n)
    bwd = backward_euler_linear(A_stiff, x0_stiff, T, n)
    print(f'{T / n:5.2f} {fwd[0]:10.1e} {fwd[1]:9.4f} {bwd[0]:10.1e} {bwd[1]:9.4f}')
```

```
exact at T = 1: fast 1.9e-22, slow 0.3679
    h   fwd fast  fwd slow   bwd fast  bwd slow
 0.10    1.0e+06    0.3487    1.7e-08    0.3855
 0.05    3.3e+03    0.3585    1.3e-11    0.3769
 0.04   -1.0e+00    0.3604    1.2e-12    0.3751
 0.02    0.0e+00    0.3642    8.9e-16    0.3715
 0.01    7.9e-31    0.3660    2.5e-18    0.3697
```

$h = 0.1$ では、どちらの手法も遅いモードを真の値から約 5% 以内（0.349 と 0.386 であり、真値は 0.368）で捉えており、多くの用途には十分な精度である。それにもかかわらず、前進オイラー法は速い成分に対して $4^{10} \approx 10^6$ を返してしまう（真の値は 10^{-22} である）。$h = 0.05$ ではその因子は冒頭で見た -1.5 であり、これが 20 回掛けられる。$h = 0.04$ では因子は -1 となり、速い成分は毎ステップ符号を反転させるばかりで決して縮小しない（$h = 0.02$ の行にあるゼロは厳密値であり、このとき $1 - 50h = 0$ となって 1 ステップで速いモードが完全に消去される）。前進オイラー法には $h < 0.04$ が必要だが、そのステップ幅では遅いモードの誤差はすでに約 2% 以内に達している。5% の精度で十分だったのに、安定性のせいで 2% の精度まで計算させられたのである。

後退オイラー法はあらゆるステップサイズで安定である。ただし遅いモードの誤差は 1 次手法としての $O(h)$ の誤差のままであり、安定性が精度まで高めてくれるわけではない。

38.2.3.2 安定領域

前進オイラー法にはステップサイズの上限があり、後退オイラー法にはない。では他の手法はどこに位置するのだろうか。複素 z 平面をプロットすれば、それらを一目で比較できる。ある手法の**安定領域**（stability region）とは、更新が減衰する集合 $\{z : |R(z)| < 1\}$ のことである。安定領域が開左半平面 $\mathrm{Re}\, z < 0$ 全体を含むとき、その手法は **A 安定**（A-stable）であるという。すなわち、ステップ幅によらずすべての減衰モードを減衰させる手法である (Dahlquist 1963)。後退オイラー法の安定領域は $+1$ を中心とする閉単位円板の外側全体であるため、A 安定である。

Figure 38.6 にこれまでの例をプロットした。冒頭のステップ $h = 0.05$（$\dot{x} = -50x$）はオイラー法の円板の外にある点 $z = -2.5$ に対応し、$h = 0.01$ は円板内部の $z = -0.5$ に対応する。固有値 $\lambda = -0.5 \pm \mathrm{i}$ を持つスパイラルでは $|1 + z|^2 = (1 - h/2)^2 + h^2$ となり、これが 1 未満となるのは $h < 0.8$ のとき、かつそのときに限られる。$h = 0.8$ では 2 点 $-0.4 \pm 0.8\,\mathrm{i}$ がちょうど円周上にのる。純粋な回転は z を虚軸上に置くが、虚軸はオイラー法の円板と原点でしか接しないため、前進オイラー法は減衰のない振動をあらゆるステップサイズで増大させてしまう（演習問題 2 および 4）。一方、厳密なステップ e^{z}（38.1.2.3 節）は左半平面全体で絶対値が 1 未満となり、後退オイラー法はこの振る舞いを模倣しているのに対し、前進オイラー法はそうではない。

いかなる陽的ルンゲ・クッタ法も、段数をどれだけ増やそうと A 安定にはならない。その安定性関数 R は多項式であり（演習問題 2）、定数でない多項式は z が負の実軸に沿って遠ざかると際限なく増大するからである。したがって、次数を高くしても硬さは解消されない。RK4 の安定領域は実軸上では $z = -2.79$（$R_{\mathrm{RK4}}(z) = 1$ の負の根）で途切れる。ボウルの減衰方程式に対して RK4 が許容するステップ幅は $h < 2.79/50 = 0.056$ であり、オイラー法の 0.04 と大差ないうえ、1 ステップあたり場を 4 回も評価する。硬い場に対しては陰的手法が温存され、それ以外のあらゆる場面では陽的手法が標準として使われる理由はここにある。本書の読者が最も頻繁に出会う硬い場とは、ある方向では曲率（curvature）が大きく、別の方向では曲率が小さい損失の勾配である。

38.2.4 ソルバとしての勾配降下法

冒頭では 1 つの座標だけを扱ったが、座標が複数ある場合にはヘッセ行列（Hessian）の固有値ごとに同じ因子が現れる。損失 $\mathcal{L}$ 上をつねに負の勾配に沿って連続的に下っていく運動は、次の**勾配フロー**（gradient flow）である。

$$\dot{\mathbf{x}} = -\nabla \mathcal{L}(\mathbf{x}),$$

これは $\mathcal{L}$ の臨界点（critical point）でゼロになり、その軌跡に沿って損失が単調に減少する（$\frac{d}{dt}\mathcal{L}(\mathbf{x}(t)) = \nabla\mathcal{L}^{\top}\dot{\mathbf{x}} = -\|\nabla\mathcal{L}\|^2 \le 0$）ようなベクトル場である。学習率 η の勾配降下法 $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\,\nabla\mathcal{L}(\mathbf{x}_k)$ は、このフローに対する前進オイラー法 Equation 38.9 において $h = \eta$ とおいたものにほかならない。

35.1 節では、学習率の上限 $\eta < 2/\lambda_{\max}(\mathbf{H})$ をヘッセ行列 $\mathbf{H}$ を持つ二次関数上で導出し、その上限より下での収束速度を調べている。収束速度の議論は繰り返さないが、上限については 3 本の矢印で改めて導出しよう。そこに至る道筋こそがソルバとしての視点のもたらす新しい知見だからである。$\delta = \mathbf{x} - \mathbf{x}^{\star}$ を最小点 $\mathbf{x}^{\star}$（ヘッセ行列 $\mathbf{H}$）からの変位とする。最小点の近傍ではフローが線形化され、各固有値 λ_i（$\mathbf{H}$ の固有値）が減衰率 $-\lambda_i$ を持つ減衰モード、すなわちテスト方程式のコピーになる。

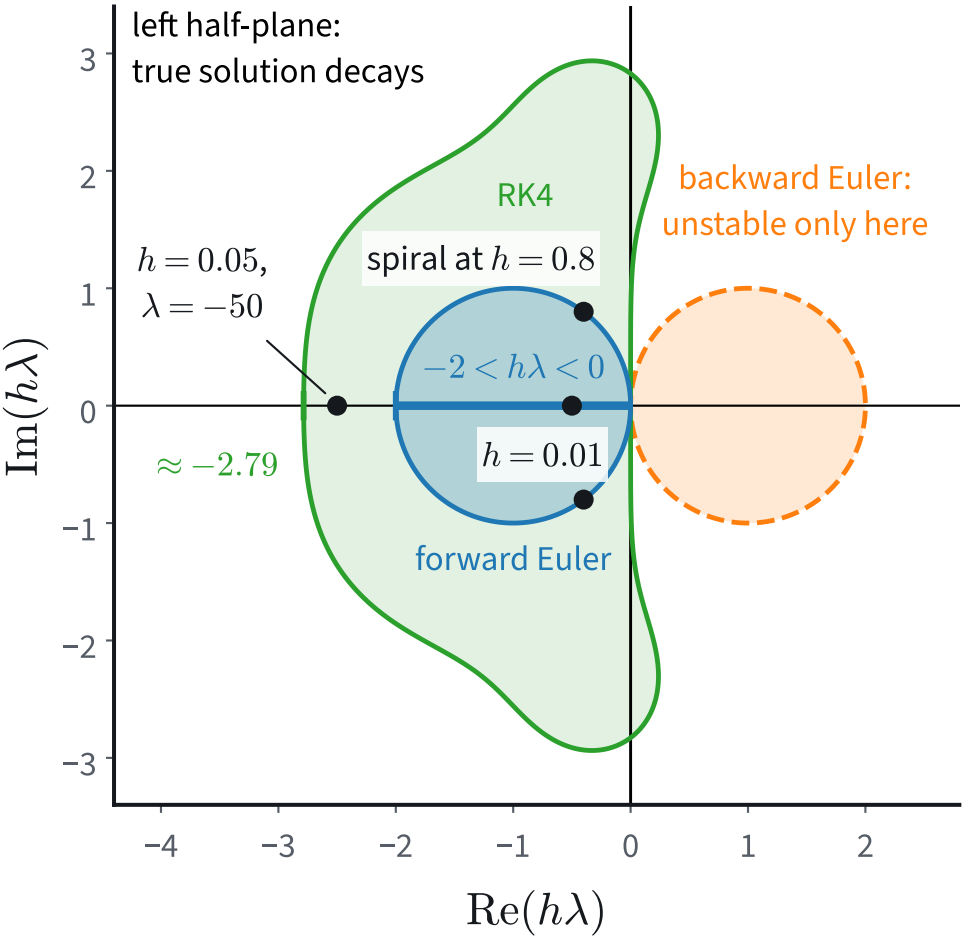


Figure 38.6: 複素平面 $z = h\lambda$ におけるテスト方程式 $\dot{y} = \lambda y$ の安定領域。減衰モードは左半平面に位置する。前進オイラー法が減衰するのは円板 $|1 + z| < 1$（実軸上のスライスは $-2 < h\lambda < 0$）の内部に限られる。RK4 はこれを実軸上で ≈ -2.79 に達する葉状の領域へと広げる。後退オイラー法は $+1$ の周りの円板を除く全域、特に左半平面全体で安定であり、これが A 安定性である。印を付けた点は、冒頭の例の $h = 0.05$ ステップと $h = 0.01$ ステップ、およびオイラー法の円板の境界上にある $h = 0.8$ のスパイラルである。

$$\dot{\mathbf{x}} = -\nabla \mathcal{L}(\mathbf{x}) \xrightarrow{\text{linearize}} \dot{\delta} = -\mathbf{H}\delta \xrightarrow{\text{Euler, } h=\eta} \delta_{k+1} = (\mathbf{I} - \eta\mathbf{H})\,\delta_k \xrightarrow{\text{stability}} \eta < \frac{2}{\lambda_{\max}(\mathbf{H})}.$$

このように、学習率の上限とはオイラー法の安定円板の境界であり、最も急峻なモードが $z = -\eta\lambda_{\max} = -2$ でそこに達する点にあたる。最初の矢印ではヘッセ行列を固定しているため、この上限は二次関数に関する主張である。実際のネットワーク上では訓練中にヘッセ行列が変化するため、この上限を超えて実行しても爆発しないことが少なくない。最大固有値が $2/\eta$ に向かって押し戻され、毎ステップではないものの長期的には損失が減少し続けるのである（35.1.2.4 節）。

この解釈が正しければ、上限はオイラー法に由来しており、別のソルバを使えば別の上限が現れるはずである。35.1 節で学習率をスイープさせた谷型の関数 $\mathcal{L}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{H}\mathbf{x}$（ただし $\mathbf{H} = \mathrm{diag}(10, 1)$ であり、同節とは平坦な座標と急峻な座標の順が逆である）でこの予測を検証してみよう。その勾配フロー $\dot{\mathbf{x}} = -\mathbf{H}\mathbf{x}$ は、減衰率 -10 の急峻なモードと減衰率 -1 の平坦なモードを持つ。初期点 $(1, 1)$ から 3 種類のソルバで 100 ステップ（ステップ幅 η）進める。前進オイラー法は勾配降下法そのものであり、上限は $2/10 = 0.2$ にある。RK4 は 1 ステップあたり勾配を 4 回評価し、その安定領域は実軸上で -2.79 まで届くため（Figure 38.6）、上限は 0.279 になるはずである。後退オイラー法 $\mathbf{x}_{k+1} = (\mathbf{I} + \eta\mathbf{H})^{-1}\mathbf{x}_k$ には上限がないはずである。次のコードセルでは、各実行について 1 ステップが 2 つのモードに掛ける因子 $R(-10\eta)$ と $R(-\eta)$、および 100 ステップ後の反復点の長さを出力する。これらの因子は単位行列（identity matrix）に 1 ステップを適用して実測している。

```
H = np.diag([10.0, 1.0])                        # Hessian of L(x) = x^T H x / 2
flow = lambda x: -H @ x                         # the gradient flow
solvers = {                                     # n steps of size eta on the flow
    'forward Euler': lambda x0, eta, n: euler(flow, x0, n * eta, n),
    'RK4': lambda x0, eta, n: rk4(flow, x0, n * eta, n),
    'backward Euler': lambda x0, eta, n: backward_euler_linear(-H, x0, n * eta, n)}
print(f'{"solver":<15}{"eta":>6}{"steep factor":>14}{"flat factor":>13}{"||x_100||":>12}')
for name, eta in [('forward Euler', 0.19), ('forward Euler', 0.21), ('RK4', 0.27),
                  ('RK4', 0.29), ('backward Euler', 0.25), ('backward Euler', 25)]:
    steep, flat = np.diag(solvers[name](np.eye(2), eta, 1))  # one step's matrix
    x_100 = solvers[name](np.array([1.0, 1.0]), eta, 100)
    print(f'{name:<15}{eta:6.2f}{steep:14.3f}{flat:13.3f}{np.linalg.norm(x_100):12.1e}')
```

```
solver            eta  steep factor  flat factor   ||x_100||
forward Euler    0.19        -0.900        0.810     2.7e-05
forward Euler    0.21        -1.100        0.790     1.4e+04
RK4              0.27         0.879        0.763     2.5e-06
RK4              0.29         1.187        0.748     2.8e+07
backward Euler   0.25         0.286        0.800     2.0e-10
backward Euler  25.00         0.004        0.038    3.2e-142
```

各ソルバは、それぞれの安定領域が途切れる場所に固有の上限を持っている。前進オイラー法は $\eta = 0.19$ では急峻なモードの因子が -0.9 となって収束し、$\eta = 0.21$ では -1.1 となって発散する。RK4 は $\eta = 0.27$ でもまだ収束し、0.29 で発散する。後退オイラー法は勾配降下法の上限を超えた $\eta = 0.25$ でも収束し、両者をはるかに上回る $\eta = 25$ でも 2 つの因子が 0.004 と 0.038 となって、100 ステップ後には反復点が測定できないほどゼロに近づく。冒頭で「損失に問題がないとすれば何が問題なのか」と問うたが、答えは「上限は商で決まる」ということである。

損失がその分母 $\lambda_{\max}$ を与え、ソルバがその分子を与える。分子は前進オイラー法では 2、RK4 では 2.79 であり、後退オイラー法にはそもそも上限が存在しない。

ではネットワークの訓練に RK4 を使うべきだろうか。答えは否である。上限は 1.4 倍高くなるものの 1 ステップに 4 回の勾配計算を要するため、勾配 1 回あたりに進められるフローの時間は高々 0.07 にとどまり、勾配降下法の 0.2 に及ばない。追加の勾配評価は勾配フローの経路をより忠実にたどるために費やされるが、オプティマイザにとって途中の経路自体には用がなく、欲しいのは経路が行き着く終着点だけである。しかも、経路から外れることは単なる誤差にとどまらない。η の 2 次の範囲で、勾配降下法の反復点は大きな勾配にペナルティを課す修正損失のフローをたどるため、オイラー法がソルバとして犯す誤差は正則化項として働くのである（演習問題 5）。

2 つの手法は破綻の仕方も異なる。オイラー法の急峻なモードの因子は -1 を通って安定範囲を抜け出すため、反復点は谷底を飛び越えながら毎回より高い位置へと着地する。一方、RK4 の因子は実軸全体で正であり、$+1$ を通って安定範囲を抜ける。$\eta = 0.29$ では因子が 1.187 となり、反復点は一度も符号を変えないまま最小点から遠ざかっていく。つまりジグザグの振動もまた、オイラー法に固有の現象なのである。その定番の処方箋であるモメンタム（momentum）は、そもそも別の方程式を解いている。Polyak のヘビーボール法は、質量と摩擦を持つ球の運動方程式 $m\ddot{\mathbf{x}} + c\dot{\mathbf{x}} = -\nabla \mathcal{L}(\mathbf{x})$ を離散化したものである (Polyak 1964)（演習問題 6）。

後退オイラー法はステップ幅を大きくするほど因子 $1/(1+\eta\lambda_i)$ が小さくなるため、性能は向上する一方である。難点は方程式の求解にある。その更新式 $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla \mathcal{L}(\mathbf{x}_{k+1})$ は、関数 $\mathcal{L}(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_k\|^2$ の勾配が $\mathbf{x}_{k+1}$ でゼロになることを意味しており、凸な損失に対しては損失そのものに適用した近接ステップ（proximal step）$\operatorname{prox}_{\eta\mathcal{L}}(\mathbf{x}_k)$（35.6.4.1 節）に一致する。二次関数上ではこれは 1 回の連立一次方程式の求解で済むが、ネットワーク上ではステップごとにそれ自体が 1 つの最適化問題となり、η が大きいときには元々解こうとしていた問題そのものになってしまう。

この谷型の関数は 38.2.3 節の硬い系を穏やかにしたものであり、減衰率として 10 と 1（50 と 1 の代わり）、ステップ幅として η（h の代わり）を用いている。任意の $\eta < 0.2$ に対して勾配降下法の平坦方向の因子 $1-\eta$ は 0.8 より大きいままであるため（出力表には 0.81（$\eta = 0.19$）と表示されている）、急峻な方向がどれほど速く収束しようと、平坦な方向は 1 ステップあたり長さの 5 分の 1 未満しか縮まない。一般には縮小する割合は $2\lambda_{\min}/\lambda_{\max}$ 未満であり、条件数（condition number）が 1000 であれば 500 分の 1 にすぎない。安定性が η の上限を $\lambda_{\max}$ によって決め、そのもとで $\lambda_{\min}$ が収束速度を決める。つまり、ヘッセ行列の条件数とは勾配フローの硬さにほかならず、二次関数における学習率の発散とはソルバの不安定性そのものなのである。

本節で扱ったすべての場では、自分たちで場を書き下していたため、最初の 1 ステップを踏み出す前から固有値がわかっていた。しかしネットワークで表される場には、そのような固有値のリストは付いてこない。38.3 節では、層をオイラー法のステップに見立てて学習された場そのものをモデルとし、そのようなモデルをどう訓練するか、そしてそれが確率密度（probability density）をどう変換するかを問う。

38.2.5 まとめ

数値ソルバは軌跡を離散的なステップで置き換えるものであり、その性能は 2 つの性質によって評価される。第 1 に、次数 p はステップ幅を小さくしたときに誤差がどれほど速く減少するかを表す。局所誤差 $O(h^{p+1})$ は累積して大域誤差 $O(h^p)$ となり、オイラー法では $p=1$、RK4 では $p=4$ である。第 2 に、安定領域はどのステップ幅なら爆発しないかを表す。前進オイラー法には $|1+h\lambda| < 1$ が必要であるのに対し、後退オイラー法は毎ステップ方程式を解くことで左半平面全体で安定となり、とうに減衰し切った速いモードがなおステップ幅を制限するとき系は硬いという。線形な場の上では、これら両方の性質が、1 ステップでモードに掛けられる因子である安定性関数 $R(h\lambda)$ から読み取れる。勾配降下法は $\dot{\mathbf{x}} = -\nabla\mathcal{L}$ に対する前進オイラー法である

ため、その学習率の上限 $\eta < 2/\lambda_{\max}(\mathbf{H})$ はオイラー法の安定領域の境界に一致し、悪条件性は硬さにほかならない。

38.2.6 演習問題

1. [code] **適応的ステップサイズ。**●埋め込み型ペアは、次数の異なる 2 つの手法から局所誤差を推定する。最も単純なペアは、ホイン法の段 $\mathbf{k}_1 = \mathbf{f}(\mathbf{x})$ と $\mathbf{k}_2 = \mathbf{f}(\mathbf{x} + h\mathbf{k}_1)$ を用いるものである。オイラー法の値 $\mathbf{x} + h\mathbf{k}_1$ とホイン法の値 $\mathbf{x} + \frac{h}{2}(\mathbf{k}_1 + \mathbf{k}_2)$ の差の大きさは $\Delta = \frac{h}{2}\|\mathbf{k}_2 - \mathbf{k}_1\|$ となる。制御器は $\Delta \le \texttt{tol}$ のときにホイン法の値を採用してそのステップを受理し、受理・棄却のいずれの場合も次のステップサイズで計算を続ける。

$$h \leftarrow h \min\!\Big(2,\ \max\!\big(0.2,\ 0.9\sqrt{\texttt{tol}/\Delta}\,\big)\Big).$$

   1. 指数 $1/2$ は、局所誤差が何次であることを仮定しているか。
   2. $h = 0.1$ から開始し、最後のステップがちょうど T で終わるように短縮する形でこの制御器を実装せよ。スパイラル場 `f` を `x0` から $T = 2$ まで、$\texttt{tol} = 10^{-2}, 10^{-3}, \dots, 10^{-6}$ に対して積分せよ。棄却されたステップの分も含めて場の評価回数を数え、厳密な終端点 `x_true` との誤差を比較せよ。評価回数は `tol` に対してどのようにスケーリングするか。
   3. `A_stiff` の場を $(1, 1)$ から $T = 10$ まで $\texttt{tol} = 10^{-3}$ で積分し、受理されたステップサイズを時間に対してプロットせよ。速いモードが減衰した後、ステップサイズはどの値に落ち着くか。この値をホイン法の安定性関数 $1 + z + z^2/2$（Equation 38.12）を用いて説明せよ。

2. **回転モード。**●本文の硬い系の例では固有値は実数であった。複素固有値 λ は $z = h\lambda$（Figure 38.6 の座標）を通じて現れ、その虚部がステップの安定性を左右しうる。

   1. Figure 38.1 のスパイラル場（固有値は $-0.5 \pm \mathrm{i}$）について、前進オイラー法は $h < 0.8$ で減衰することを本文で求めた。Equation 38.12 の安定性関数から、ホイン法が減衰する最大のステップ幅（条件は h の 3 次式になるので数値的に解け）および RK4 に対する最大のステップ幅を求めよ。ホイン法はステップあたり場を 2 回、RK4 は 4 回評価する。3 つの手法のうち、解を増大させずに場評価 1 回あたり最も長い時間を進められるのはどれか。
   2. 回転場 $\dot{\mathbf{x}} = \left(\begin{smallmatrix} 0 & -b \\ b & 0 \end{smallmatrix}\right)\mathbf{x}$ に対し、前進オイラー法は 1 ステップごとに $\|\mathbf{x}\|$ を $\sqrt{1 + h^2 b^2}$ 倍し、後退オイラー法は同じ因子で割ることを示せ。各手法において、$\|\mathbf{x}\|$ は 1 回転すなわち $2\pi/(hb)$ ステップ（$hb = 0.1$ とする）の間に何倍に変化するか。
   3. RK4 の安定性関数は Equation 38.12 の最後の式であり、虚軸上でのその絶対値の二乗が回転に対する振る舞いを決める。$|R(\mathrm{i}y)|^2 = 1 - y^6/72 + y^8/576$ が実数 y に対して成り立つことを示し、RK4 が回転を増幅させない hb の範囲を求めよ。
   4. s 段の陽的ルンゲ・クッタ法は、$\mathbf{k}_i = \mathbf{f}(\mathbf{x}_n + h\sum_{j<i} a_{ij}\mathbf{k}_j)$（$i = 1, \dots, s$）および $\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_i b_i \mathbf{k}_i$ を計算する（Equation 38.11 の RK4 と同様である）。その安定性関数が z に関する高々 s 次の多項式であることを示せ。なぜこのことから、陽的手法では A 安定性が除外されるのか。

3. [code] **硬い非線形場における陰的ステップ。**●本文の硬い系は線形かつ対角であったため、後退オイラー法の各ステップは 1 回の連立一次方程式の求解で済んだ。代わりに、減衰振り子 $\dot{\phi} = \omega$、$\dot{\omega} = -\sin\phi - c\,\omega$（38.1.2.5 節）に強い減衰 $c = 50$ を入れ、$(\phi, \omega) = (1, 0)$ から $T = 2$ まで解くことを考えよ。

   1. ヤコビ行列 $\mathbf{J}(\phi, \omega) = \left(\begin{smallmatrix} 0 & 1 \\ -\cos\phi & -c \end{smallmatrix}\right)$ の最下点 $(0, 0)$ における固有値を求め、そこから導かれる前進オイラー法と RK4 のステップサイズの上限を計算せよ。

2. 速い過渡応答が収まった後の運動はほぼ $\dot{\phi} = -\sin\phi/c$ を満たすことを示し、この 1 次元方程式から $\phi(2)$ を求めよ。
3. 後退オイラー法の 1 ステップは、$\mathbf{g}(\mathbf{y}) = \mathbf{y} - \mathbf{x}_n - h\mathbf{f}(\mathbf{y}) = \mathbf{0}$ を $\mathbf{y} = \mathbf{x}_{n+1}$ について解く。これに対するニュートン反復 $\mathbf{y} \leftarrow \mathbf{y} - (\mathbf{I} - h\mathbf{J}(\mathbf{y}))^{-1}\mathbf{g}(\mathbf{y})$ を書き下し、ステップごとに 3 回のニュートン反復（初期値は $\mathbf{x}_n$）を行う後退オイラー法を実装せよ。
4. 前進オイラー法、RK4、後退オイラー法を $n = 20$、33、36、100 の等間隔ステップ（$h = 0.1$、0.061、0.056、0.02）で実行し、`rk4` を 2^{16} ステップ用いて計算した参照解と比較せよ。どの実行が爆発するか。また、小問 (a) の上限はその結果を予測しているか。続いて各手法について終端誤差が 10^{-4} 未満となる最小の n を求め、そのときに要する場とヤコビ行列の評価回数を数えよ。どの手法では精度が n を決め、どの手法では安定性が決めているか。

4. [code] **シンプレクティック・オイラー法と作用素分割。**●調和振動子 $\dot{q} = p$、$\dot{p} = -q$ は $q^2 + p^2$ を保存する。その円軌道に対して前進オイラー法は外側へ渦を巻いて離れ、後退オイラー法は内側へ渦を巻いて縮む。これに対し、**シンプレクティック・オイラー法**の更新は q を古い p で進めてから、p を新しい q で進める。すなわち、$q_{n+1} = q_n + hp_n$、$p_{n+1} = p_n - hq_{n+1}$ とする。
   1. この更新を列ベクトルとして並べた状態に作用する行列 $\mathbf{S}_h$（(q,p) 上の変換）として書き下し、$\det \mathbf{S}_h = 1$ がすべての h に対して成り立つこと、すなわち厳密なフローと同様に面積を保存することを示せ。また、前進オイラー法の行列式が $1 + h^2$ となることも確認せよ。
   2. $\mathbf{S}_h$ の固有値が互いに異なりかつ単位円周上にのる必要十分条件が $0 < h < 2$ であることを示せ。$h = 2$ のとき、および $h > 2$ のときには何が起こるか。
   3. $q^2 + p^2 + hqp$ がこの更新によって厳密に保存されることを示せ。$(1,0)$ を通るその等位集合はどのような形をしており、計算された軌道は $h = 0.5$ と $h = 1$ において原点から最大どれだけ離れうるか。（ヒント：この二次形式の主軸の長さの比は $\sqrt{(2+h)/(2-h)}$ である。）
   4. $(1,0)$ から $T = 200$ まで、前進オイラー法、後退オイラー法、シンプレクティック・オイラー法を用いて $h = 0.1$ と $h = 1$ で積分し、各実行に沿った $\|(q,p)\|$ の最大値と最小値を報告せよ。
   5. この更新は、$\dot{q} = p$（p を固定）の時間 h にわたる厳密なフローに続けて、$\dot{p} = -q$（q を固定）の厳密なフローを適用したものである。和がこの振動子の行列となる 2 つの生成作用素 $\mathbf{P}$ と $\mathbf{Q}$ を書き下し、$e^{h\mathbf{P}}$、$e^{h\mathbf{Q}}$、$e^{h(\mathbf{P}+\mathbf{Q})}$ を計算して、$e^{h\mathbf{Q}}e^{h\mathbf{P}} = \mathbf{S}_h$ が $e^{h(\mathbf{P}+\mathbf{Q})}$ と h^2 のオーダーの項だけ異なることを示せ。38.1 節におけるスパイラルの閉形式解では、指数関数を誤差なしに分解できていた。そこで用いたどの性質がここでは成り立たないのか。対称分解 $e^{h\mathbf{Q}/2}e^{h\mathbf{P}}e^{h\mathbf{Q}/2}$ はハミルトニアンモンテカルロ (Neal 2011) の**リープフロッグ**積分器である。その次数はいくらか。

5. [code] **暗黙的勾配正則化。**●●勾配降下法は勾配フローに対する前進オイラー法であるため（38.2.4 節）、各ステップは $\mathcal{L}$ のフローから $O(\eta^2)$ だけ離れる。後退誤差解析では、修正ベクトル場 $\tilde{\mathbf{f}} = \mathbf{f} + \eta\mathbf{f}_1$ であって、その厳密な時間 η のフローがオイラー法の 1 ステップと $O(\eta^3)$ の範囲で一致するものを探す。Barrett and Dherin (2021) は、勾配降下法に対してこの修正場が再び勾配場となり、次の修正損失から導かれることを示した。

$$\tilde{\mathcal{L}}(\mathbf{x}) = \mathcal{L}(\mathbf{x}) + \frac{\eta}{4}\|\nabla\mathcal{L}(\mathbf{x})\|^2.$$

   1. $\tilde{\mathbf{f}}$ の時間 η にわたる厳密なフローを 2 次まで展開し、$\mathbf{f}_1 = -\frac{1}{2}(\partial\mathbf{f}/\partial\mathbf{x})\,\mathbf{f}$ を示せ。さらに $\tilde{\mathcal{L}}$ を $\mathbf{f} = -\nabla\mathcal{L}$ の場合に導出せよ。追加された項はパラメータ空間のどのような領域にペナルティを課すか。

2. $\mathcal{L}(x) = \frac{\lambda}{2}x^2$（ただし $0 < \eta\lambda < 1$）に対し、勾配降下法の反復点が $\frac{\tilde{\lambda}}{2}x^2$（ただし $\tilde{\lambda} = -\log(1-\eta\lambda)/\eta$）の勾配フロー上に厳密にのることを示せ。$\tilde{\lambda}$ を η で展開し、$\tilde{\mathcal{L}}$ の曲率と比較せよ。
3. この二次関数に対する勾配降下法は $1 < \eta\lambda < 2$ でも依然として収束するが、反復点の符号は交互に反転する。リプシッツ定数を持つスカラーの自律的な場の軌跡は交差できないため、そのような場の時間 η 写像は大小順序を保存する。このことから、いかなるそのような場も時刻 $k\eta$ において反復点の値をとることはできないと結論せよ。さらに、ヘッセ行列 $\mathbf{H}$ を持つ d 次元の二次関数に対し、最大固有ベクトル方向の成分を持つ任意の初期点について、反復点は、いかなる勾配場 $-\nabla\big(\frac{1}{2}\mathbf{x}^\top\tilde{\mathbf{H}}\mathbf{x}\big)$ のフロー上にも $\eta \geq 1/\lambda_{\max}(\mathbf{H})$ となった時点でのらないことを示せ。
4. $\mathcal{L}(x) = x^4/4 + x^2/2$、$x_0 = 1.5$ に対し、勾配降下法を $T = 1$ まで $\eta = 0.1, 0.05, 0.025, 0.0125$ で実行せよ。$\mathcal{L}$ および $\tilde{\mathcal{L}}$ の勾配フローを $T = 1$ まで `rk4` を十分なステップ数で用いて計算せよ（後者は各 η ごとに個別に計算すること）。勾配降下法の 2 つの終端誤差の傾きを η に対して直線当てはめせよ。

6. **離散化された ODE としてのヘビーボール法。**●ヘビーボールモメンタム $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\nabla\mathcal{L}(\mathbf{x}_k) + \beta(\mathbf{x}_k - \mathbf{x}_{k-1})$ は、方程式 $m\ddot{\mathbf{x}} + c\dot{\mathbf{x}} = -\nabla\mathcal{L}(\mathbf{x})$（質量 m と摩擦係数 c を持つ球の運動）を離散化したものである。

   1. $\ddot{\mathbf{x}}(t_k)$ を中心差分 $(\mathbf{x}_{k+1} - 2\mathbf{x}_k + \mathbf{x}_{k-1})/h^2$ で、$\dot{\mathbf{x}}(t_k)$ を後退差分 $(\mathbf{x}_k - \mathbf{x}_{k-1})/h$ で置き換えよ。その結果がヘビーボール法の更新式になることを示し、η と β を m、c、h を用いて表せ。
   2. $\mathcal{L}(x) = \frac{\lambda}{2}x^2$ に対してこの ODE は減衰振動子となり、$c = 2\sqrt{m\lambda}$ で臨界減衰となる。この条件を η、β、λ を用いて表せ。ヘビーボール法の漸化式は、その特性多項式 $z^2 - (1 + \beta - \eta\lambda)z + \beta$ が重根を持つときに臨界減衰となる。対応する β を求め、小さい $\eta\lambda$ に対して 2 つの条件を比較せよ。
   3. この二次関数上のヘビーボール法の漸化式が収束する必要十分条件は、$0 < \eta\lambda < 2(1+\beta)$（ただし $0 \leq \beta < 1$）である（演習問題 8 の小問 (a)（35.1 節）で証明を求めている）。この条件を h、m、c、λ に関する条件に書き換えよ。元の ODE はすべての $\lambda > 0$ に対して減衰するが、その離散化はそうではない。安定となる最大のステップ幅 h は、$c \to 0$ の極限において振動子の角振動数 $\sqrt{\lambda/m}$ を用いてどう表されるか。悪条件な二次関数のヘッセ行列のどの固有値が、ステップ幅の上限を決めるか。

38.3 ニューラル ODE、随伴法、連続フロー

残差ブロック（residual block）は入力に補正項を加えるが、オイラー法（38.2 節）の 1 ステップも同じ形をしている。残差ネットワーク（residual network、7.4 節）が微分方程式の数値解だとすれば、方程式そのものをモデルとすることで何が得られ、どのようなコストがかかるのだろうか。

生成モデリングがその 1 つの答えを与える。最尤法（maximum likelihood）には、モデルが生成する出力の密度が必要である。正規化フロー（normalizing flow）は逆変換可能なネットワークであり、この密度は変数変換（change of variables）の公式から従う。しかし、その公式はサンプルごとにネットワークのヤコビ行列（Jacobian）の行列式（determinant）を必要とし、そのサイズは 3072 行にもなる（32×32 ピクセルのカラー画像の場合）。連続時間（continuous-time）に移ると、この行列式はトレース（trace）へと変わり、トレースは 1 回の逆伝播（backward pass）で推定できる。

ニューラル ODE（neural ODE）は方程式そのものをモデルとする (Chen et al. 2018)。すなわち、ベクトル場（vector field）をネットワークで表し、終端時刻における解を出力とする常微分

方程式（ordinary differential equation、ODE）$\dot{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{x}, t)$ である。その訓練と密度の計算は、いずれも状態（state）と並んでもう 1 本の方程式を積分することに帰着する。訓練で用いるのは**随伴**方程式（adjoint equation）であり、これは軌跡（trajectory）のコストが制御入力に応じてどう変化するかを調べるため、1950 年代に最適制御理論で導入されたものである (Pontryagin et al. 1962)。密度の計算で用いるのは、ヤコビ行列のトレースの積分である。

残差ブロックの積み重ねと比べると、このモデルは可逆な写像、モデル自体から切り離されたステップ数、活性化値（activation）を保存せずに済む勾配（gradient）、そして計算可能な密度を手に入れる。ただし、それぞれの利点には落とし穴があり、本節ではその 4 つすべてを定量的に確かめる。すなわち、直線上のフローは 2 点を入れ替えられず、訓練されたベクトル場は使用したソルバに縛られ、状態を保存せずに計算した勾配にはドリフト（drift）が生じ、安価に求まるトレースにはノイズが乗る。

本節では、このようなモデルをソルバを通して訓練し、随伴法（adjoint method）を導出し、フローに沿った密度の変化を追う。持ち帰るべき結果は随伴方程式とトレース則 Equation 38.21 であり、拡散モデル（diffusion model）の尤度（likelihood）は後者に基づいている。トレース則だけが必要な読者は 38.3.3 節から読み始めてもよい。メモリとドリフトに関する項の内容は、これ以降の節では前提としない。

以下のセルでは、38.2 節のオイラーステップと RK4 ステップを通して訓練する。ループ `solve` はこれらのステップを N 回（ステップサイズ T/N）実行し、負の T を与えるとベクトル場を逆向きに走らせる。さらに後段で行う随伴法の検証では、スパイラルを表す行列 $\mathbf{A}$ とその初期値 $\mathbf{x}_0$、および行列指数関数（matrix exponential）$e^{\mathbf{C}s} = \mathbf{V} e^{\Lambda s} \mathbf{V}^{-1}$ を用いる（38.1 節）。

```
rng = np.random.default_rng()
A = np.array([[-0.5, -1.0], [1.0, -0.5]])      # spiral sink: eigenvalues -0.5 +- i
x0 = np.array([2.0, 0.0])

def expm_eig(C, s):
    """e^{Cs} via the eigendecomposition C = V diag(mu) V^{-1}."""
    mu, V = np.linalg.eig(C)
    return (V @ np.diag(np.exp(mu * s)) @ np.linalg.inv(V)).real

def euler_step(f, x, h):
    """One forward-Euler step of size h for dx/dt = f(x)."""
    return x + h * f(x)

def rk4_step(f, x, h):
    """One classical Runge-Kutta step of size h for dx/dt = f(x)."""
    k1 = f(x)
    k2 = f(x + h / 2 * k1)
    k3 = f(x + h / 2 * k2)
    k4 = f(x + h * k3)
    return x + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)

def solve(step, f, x, T, N):
    """N steps of size T/N from time 0 to time T; T < 0 runs the field backward."""
    for _ in range(N):
        x = step(f, x, T / N)
    return x
```

38.3.1 ニューラル ODE

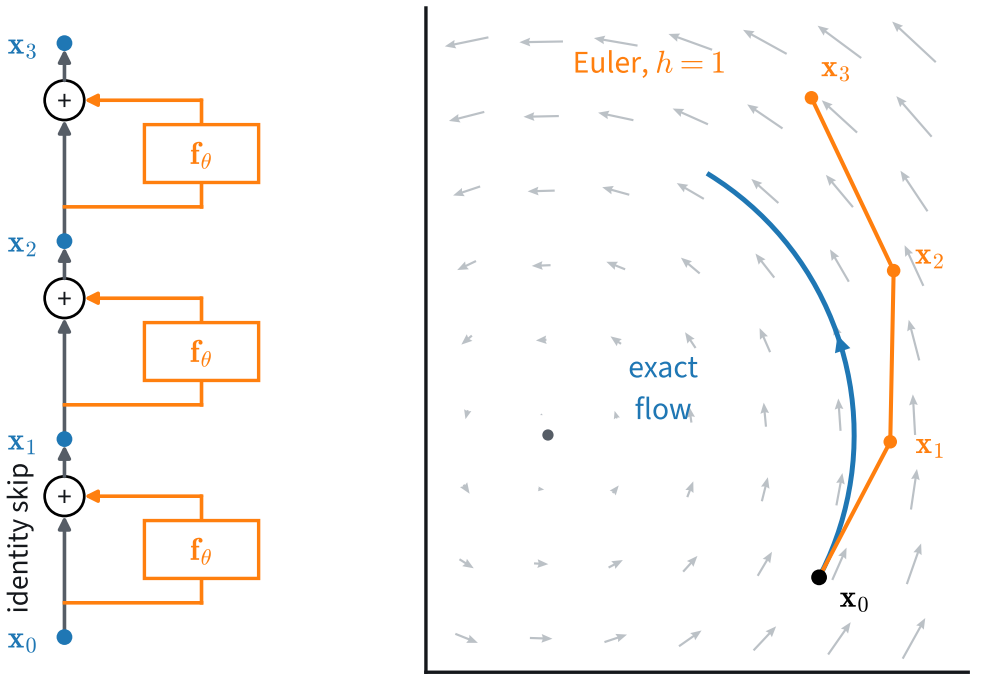
ニューラル ODE とは、残差ネットワークの深さを時間へと置き換えたものである。

38.3.1.1 残差ブロックから連続モデルへ

残差ブロック (He et al. 2016a) と、ODE $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ の前進オイラーステップを並べてみよう。

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathbf{f}_{\theta_l}(\mathbf{x}_l) \qquad \text{and} \qquad \mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}(\mathbf{x}_n, t_n). \tag{38.14}$$

$h = 1$ かつ $\mathbf{f}(\cdot, t_l) = \mathbf{f}_{\theta_l}$ とおけば、2 つの更新は同じ式になる。したがって、ブロックごとに異なる重み（weight）を持つ ResNet は、各層に対応する時刻でサンプリングされた時間依存のベクトル場を、単位ステップ幅のオイラー法で積分したものにほかならない（Figure 38.7 はすべてのブロックが 1 つのベクトル場を共有する場合を描いている）。積分の終端時刻はその深さにあたり、10 個のブロックならベクトル場を時刻 10 まで積分することになる。



$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathbf{f}_\theta(\mathbf{x}_l) \quad \Leftrightarrow \quad \mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}(\mathbf{x}_n) \quad \text{with } h = 1$$

Figure 38.7: 左：1 つのベクトル場を共有する残差ブロックの積み重ね。各ブロックは恒等スキップ結合に学習された分岐を加える。右：同じ更新をサイズ $h = 1$ の前進オイラーステップとして見たもの。同じベクトル場の厳密なフローも併記しており、オイラーステップはそこからドリフトしていく。

極限をとるため、終端時刻 T を固定し、すべての時刻で 1 つのパラメータベクトル θ を共有させたまま、ステップを細かくしていく。層の言葉で言えば、各層の補正量を小さくしながらネットワークを深くしていくことになる（ベクトル場は依然として t に依存してよい。ネットワークが t

を入力として受け取ることで、1 つのパラメータベクトルでも異なるブロックの役割を果たせる）。N 個のステップ（ステップサイズ $h = T/N$、時刻 $t_n = nh$）をとると、次式が得られる。

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}_\theta(\mathbf{x}_n, t_n), \qquad \mathbf{x}_N \xrightarrow{N\to\infty} \mathbf{x}(T) = \mathbf{x}_0 + \int_0^T \mathbf{f}_\theta(\mathbf{x}(t), t)\,dt. \tag{38.15}$$

この収束（convergence）は 38.2 節で示したオイラー法の誤差評価から従う。すなわち、滑らかで $\mathbf{x}$ についてリプシッツ連続なベクトル場に対し、N ステップ後のオイラー反復は厳密解から $O(h)$ 以内に収まる。この極限として定まる対象が**ニューラル ODE** であり、出力を解 $\mathbf{x}(T)$（方程式 $\dot{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{x}, t)$、初期条件 $\mathbf{x}(0) = \mathbf{x}_0$）とするモデルである。パラメータが定義するのはベクトル場であり、ODE ソルバ（オイラー法、RK4、適応的手法など）は後から選ぶものにすぎない。ソルバを変えれば計算される写像やそのコスト、誤差は変わるが、モデル自体は変わらない。

厳密な写像と数値計算された写像は、精度だけでなく定性的な性質まで異なり得る。ベクトル場 $\dot{x} = -x$ の厳密なフロー $\Phi_t(x) = e^{-t}x$ は、$\mathbb{R}$ 上の全単射をすべての t に対して定める。ところが、$h = 1$ のオイラーステップを 1 回適用すると $x + 1 \cdot (-x) = 0$ となり、あらゆる入力が 0 に写される。これに対し厳密な写像は、ベクトル場が t について連続かつ $\mathbf{x}$ について大域的リプシッツ連続である限り常に全単射である。なぜなら、$\mathbf{x}(T)$ からベクトル場を逆向きに走らせる問題もやはりただ 1 つの解を持ち、その解が出発点へと戻るからである（38.1.1.3 節）。

38.3.1.2 直線上での順序保存

可逆性は利点である一方、制約（constraint）でもある。ニューラル ODE は、その元となった残差ブロックの積み重ねと同等以上の表現力を持つのだろうか。実数直線上ではそうではなく、図を描けばその理由がわかる。(t, x) 平面において、リプシッツ連続なベクトル場の軌跡は決して交差しない。-1 と 1 を入れ替える 2 本の軌跡があるとすれば、途中で必ず出会わなければならない（Figure 38.8 の (a)）。

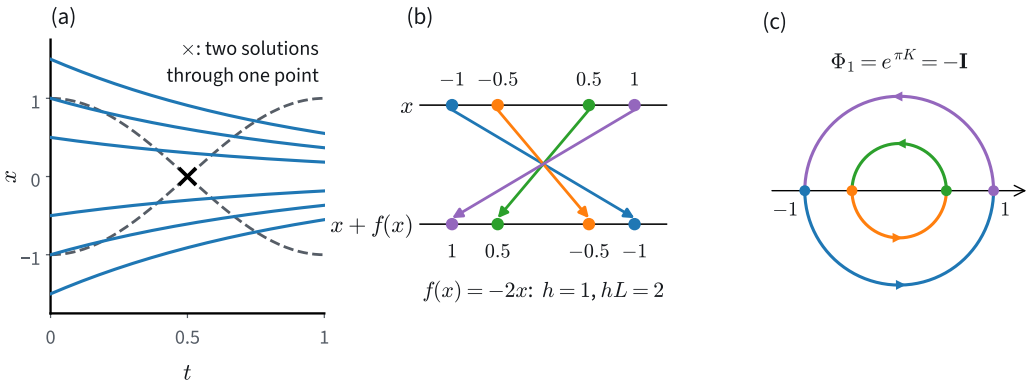


Figure 38.8: (a) 直線上ではリプシッツ連続なベクトル場（ここでは $\dot{x} = -x$）の軌跡は交差しないため、時刻 T の写像は順序を保つ。-1 と 1 を入れ替える破線の対を実現するには、1 点を通る 2 つの解が必要になる。(b) $x + f(x)$（ただし $f(x) = -2x$）という残差ステップは 1 ステップで 2 点を入れ替える。これは $hL = 2$ のオイラーステップであり、フローではない。(c) 次元を 1 つ増やすとこの障害は消える。角度 π の回転は、互いに交わらない半円に沿って $(x, 0)$ を $(-x, 0)$ へと運ぶ（色分けは (b) と同じ）。

命題（$\mathbb{R}$ 上のフローは順序を保つ） $f(x,t)$ は t について連続かつ x について大域的リプシッツ連続であるとし、$\Phi_t(x)$ を時刻 t における解（x を時刻 0 の初期値とするもの）と表す。このとき、任意の $T>0$ に対して写像 Φ_T は狭義単調増加である。

証明 $x<y$ とし、$\Phi_T(x)\geq\Phi_T(y)$ と仮定する。関数 $g(t)=\Phi_t(y)-\Phi_t(x)$ は連続で $g(0)>0$ かつ $g(T)\leq 0$ を満たすから、中間値の定理より $g(s)=0$ となる $s\in(0,T]$ が存在する。すなわち、2 本の軌跡は時刻 s にある点 z で出会う。ここで時刻 s から時間を逆向きにたどる。時間を反転させた 2 本の軌跡 $\tau\mapsto\Phi_{s-\tau}(x)$ と $\tau\mapsto\Phi_{s-\tau}(y)$ は、いずれも同じ初期値問題 $\dot{w}=-f(w,s-\tau)$、$w(0)=z$ の解であり、そのベクトル場もまたリプシッツ連続である。ピカール・リンデレーフの定理によりこの問題の解はただ 1 つであるため、逆向きの 2 本の軌跡は一致し、$\tau=s$ において $x=y$ となって矛盾（contradiction）が生じる。■

この証明の着想はただ 1 つ、軌跡が出会った時刻から逆向きに解の一意性を適用することにある。その帰結として、$\mathbb{R}$ 上のいかなるニューラル ODE も $x\mapsto -x$ を実現できない。この写像は -1 を 1 に、1 を -1 に移し、両者の順序を逆転させてしまうからである。

一方、残差ブロック $x+f(x)$（ただし $f(x)=-2x$）は、これを 1 ステップで実現する（Figure 38.8 の (b)）。このブロックは $h=1$ とした $\dot{x}=-2x$ のオイラーステップであるが、その厳密なフロー $e^{-2t}x$ が順序を逆転させることはない。両者が整合するのは、オイラーステップ $x+hf(x)$（ベクトル場のリプシッツ定数（Lipschitz constant）を L とする）が単調増加になる保証があるのは $hL<1$ のときに限られるからである。実際、$x>y$ に対して次式が成り立つ。

$$\big(x+hf(x)\big)-\big(y+hf(y)\big)\;\geq\;(x-y)-hL\,(x-y)\;=\;(1-hL)(x-y),$$

この右辺は $hL<1$ のとき正になる。いまの例では $hL=2$ であるため、ステップによって順序が逆転し得る。有限幅のステップはフローではなく、上の命題はそれを制約しない。

次元をもう 1 つ増やせば、この障害は取り除ける。回転のベクトル場 $\dot{\mathbf{x}}=\mathbf{K}\mathbf{x}$（38.1 節、ただし $\mathbf{K}=\left(\begin{smallmatrix}0&-1\\1&0\end{smallmatrix}\right)$）のフローは $e^{s\mathbf{K}}=\cos s\,\mathbf{I}+\sin s\,\mathbf{K}$ であり、平面を角度 s だけ回転させる。これを π 倍の速さで動かした $\dot{\mathbf{x}}=\pi\mathbf{K}\mathbf{x}$ の時刻 1 における写像は次のようになる。

$$\Phi_1=e^{\pi\mathbf{K}}=\cos\pi\,\mathbf{I}+\sin\pi\,\mathbf{K}=-\mathbf{I}.$$

このフローは半円に沿って $(x,0)$ を $(-x,0)$ へと運び、各軌跡は平面内で出会うことなくすれ違う（Figure 38.8 の (c)）。このように次元を追加する手法は**拡張**（augmentation）と呼ばれる (Dupont et al. 2019)。入力 x を $(x,0)$ として埋め込み（embedding）、平面上でフローを走らせたあと、第 1 座標を読み出すのである。フローとは空間の連続的な並べ替えであり、1 本の直線上には 2 点がすれ違う余地がない。

38.3.1.3 ソルバを通した訓練

ニューラル ODE の順伝播（forward pass）は N 回のソルバステップからなるループである。これを訓練する直接的な方法は、**離散化してから微分する**（discretize, then differentiate）ことである。すなわち、ソルバをそのまま計算グラフに書き下し、$\mathbf{x}_N$ に対する損失（loss）を計算して、展開されたステップ全体を逆伝播する。オイラー法をソルバに選べば、そのプログラムは 1 組の重みを共有する N ブロックの ResNet にほかならないため、通常の自動微分（automatic differentiation）で訓練（training）できる。

意図的に易しくした課題でこれを試してみよう。単位円上の 64 個の点を、中心 $(1.5,0)$、半軸 1.2 と 0.4 の楕円上の対応する点へと運ぶ課題であり、伸縮と平行移動の組み合わせにすぎない。アフィンなベクトル場 $\dot{\mathbf{x}}=\operatorname{diag}(\ln 1.2,\ln 0.4)\,\mathbf{x}+\mathbf{c}$ の厳密なフローは、適切な $\mathbf{c}$ を選べば誤差ゼロでこの変換を実現するため、もし誤差が見つかるとすれば解が存在しないせいではない。訓

練されたベクトル場は、高精度に積分したときにも目標点に到達するだろうか、それとも訓練時に使った 10 ステップでしか到達しないのだろうか。前者であれば訓練によってフローが見つかったことになり、後者であれば 10 ステップのプログラムが見つかったにすぎない。

ベクトル場は 32 個の tanh ユニットからなる隠れ層（hidden layer）を 1 つ持ち、損失は終端点での平均二乗誤差（mean squared error）とする。Adam（35.5 節）を用い、反復 0 から 400 までフルバッチで実行し、学習率（learning rate）は 0.05（最後の 100 反復では 0.005 に減衰）とする。同じ初期化（initialization）の重みから 2 回訓練する。1 回目は $N = 10$ 個のオイラーステップ（ステップサイズ $h = 0.1$）を通し、もう 1 回目は 1 ステップあたりベクトル場を 4 回評価する RK4 を 10 ステップ通して訓練する。

```
angle = np.linspace(0, 2 * np.pi, 64, endpoint=False)
X0_ring = np.stack([np.cos(angle), np.sin(angle)], axis=1)           # circle
X1_ring = np.stack([1.5 + 1.2 * np.cos(angle), 0.4 * np.sin(angle)], axis=1)
solvers = {'Euler': euler_step, 'RK4': rk4_step}
seed = int(rng.integers(2**31))               # one initialization for both solvers
```

出発点の円 `X0_ring`、目標の楕円 `X1_ring`、および共通の初期化シード `seed` を固定したうえで、`train` は 2 層の `tanh` ベクトル場 `net` を適合させる。具体的には、10 ステップの `solve(step, net, X, 1.0, 10)` を通して逆伝播することを、`solvers` 内の各ソルバについて実行する。

```
X, Y = torch.tensor(X0_ring).float(), torch.tensor(X1_ring).float()

def train(name, step, seed):
    """Fit the field through 10 steps of one solver, from the given seed."""
    torch.manual_seed(seed)
    net = nn.Sequential(nn.Linear(2, 32), nn.Tanh(), nn.Linear(32, 2))
    for layer in net[::2]:                    # Glorot-uniform weights, zero biases
        nn.init.xavier_uniform_(layer.weight)
        nn.init.zeros_(layer.bias)
    opt = torch.optim.Adam(net.parameters(), lr=0.05)
    sched = torch.optim.lr_scheduler.MultiStepLR(opt, [300], 0.1)  # 0.005 from 300
    losses = []
    for it in range(401):
        loss = ((solve(step, net, X, 1.0, 10) - Y) ** 2).mean()
        opt.zero_grad()
        loss.backward()
        opt.step()
        sched.step()
        if it % 200 == 0:
            losses.append(loss.item())
    print(f'{name}-trained: loss', ', '.join(f'{l:.1e}' for l in losses),
          'at iterations 0, 200, 400')
    return net

trained = {name: train(name, step, seed) for name, step in solvers.items()}
```

```
Euler-trained: loss 1.5e+00, 3.1e-06, 5.2e-07 at iterations 0, 200, 400
RK4-trained: loss 1.5e+00, 4.3e-06, 5.8e-07 at iterations 0, 200, 400
```

どちらの損失もオーダー 1 から 10^{-5} 未満へと減少する。訓練に用いた損失で判断する限り、2 つのベクトル場は同等に優秀である。

続いて 2 つのテストを行う。1 つ目は先ほどの問いを直接ぶつけるものである。訓練された各ベクトル場を、それぞれのソルバを用いて $N = 100$ および $N = 1000$ ステップ（いずれも $Nh = 1$）で積分する。N が大きいほど、計算される写像はそのベクトル場の時刻 1 における厳密なフローに近づく。セルを実行する前に、2 つの終端誤差がどうなるか予想してみてほしい。2 つ目のテストは、冒頭で挙げた最初の利点である可逆性に関するものである。オイラー法で訓練したベクトル場を順方向に N ステップ進めてから逆方向に N ステップ戻し、往復後の到達点が出発点からどれだけ離れているかを測る。厳密なフローであれば、すべての点が出発点に戻るはずである。

```
dist = lambda P, Q: torch.linalg.vector_norm(P - Q, dim=1).mean().item()
print('   N   Euler-trained   RK4-trained   Euler round trip')
with torch.inference_mode():
    for N in (10, 100, 1000):
        err = [dist(solve(solvers[k], trained[k], X, 1.0, N), Y) for k in solvers]
        fwd = solve(euler_step, trained['Euler'], X, 1.0, N)
        back = solve(euler_step, trained['Euler'], fwd, -1.0, N)
        print(f'{N:4d}   {err[0]:13.1e}   {err[1]:11.1e}   {dist(back, X):16.1e}')
```

```
   N   Euler-trained   RK4-trained   Euler round trip
  10         9.0e-04       9.8e-04            5.4e-02
 100         1.7e-02       9.8e-04            5.5e-03
1000         1.9e-02       9.8e-04            5.5e-04
```

まず最後の列にある往復誤差から見よう。出発点からのずれは約 0.055（$N = 10$）、0.0055（$N = 100$）、0.00055（$N = 1000$）となり、ステップ数を 10 倍にするごとに 10 分の 1 に縮まる。前進オイラーステップと逆方向のオイラーステップの対はそれぞれ h^2 のオーダーの残差（residual）を残し、それが N 回積み重なってオイラー法の次数である $O(h)$ になる。ベクトル場を逆向きに走らせれば厳密なフローは元に戻るが、オイラー法のプログラムが元に戻るのは $h \to 0$ の極限においてのみである。なお、$hL < 1$ を満たすオイラー法のプログラムには確かに逆写像が存在する。各ステップが全単射になるからである（演習問題 4）。ただし、その逆写像を構成するのは陰的なステップであり、ステップ幅を $-h$ にしたオイラーステップではない。

次に終端誤差を見よう。訓練時と同じステップ数では、どちらのベクトル場も目標点から数千分の 1 以内に到達する。ところが驚くべきことに、ステップを細かくするとオイラー法で訓練したベクトル場はかえって悪化し、誤差が約 0.02 まで**増加**してから変化しなくなる。訓練が適合させたのは 10 ステップのプログラムだったのである。$N = 1000$ が近似しているそのベクトル場の時刻 1 における厳密なフローは、目標点から 0.02（楕円の長半径の 2% 未満）だけずれた位置に着地する。つまり、このベクトル場が定めるフローは 10 ステップのプログラムに近いものの、一致してはいない。

一方、RK4 で訓練したベクトル場は、10、100、1000 ステップのいずれでも同じ誤差を出力する。これほど滑らかなベクトル場であれば 10 ステップの RK4 で表示桁数まで正確にフローが計算されるため、プログラムへの適合がそのままフローへの適合となり、訓練後はステップ数を自由に選べる。粗いソルバを通して訓練したベクトル場はプログラムであり、高精度なソルバを通して訓練したベクトル場はフローなのである。

ただし、その精度は訓練の反復ごとに支払わなければならない。RK4 を通した訓練では 1 反復あたりベクトル場を 40 回評価し、その 40 回の評価すべてを通して逆伝播するのに対し、オイラー法を通した訓練では 10 回で済む。オイラーステップを 100 回に増やしたり 2 次のソルバを使ったりしても、評価回数が変わるだけで本質は同じである。もう 1 つの選択肢は、訓練からソルバを完全に外すことである。閉形式で既知の目標値に対する回帰（regression）によって、1 データ例あたり 1 回のベクトル場評価で適合させれば、最初からフローそのものが学習される。38.6 節と 38.7 節では、このようにしてベクトル場を学習する。

38.3.2 随伴法

先ほどの訓練では 10 ステップのプログラムの勾配を用いたが、プログラムとフローは異なる写像であることがわかった。では、両者の勾配も異なるのだろうか。そもそも有限個のステップからなるプログラムではフローを厳密に計算できないのに、フローそのものの勾配を計算することなどできるのだろうか。

実は計算できる。その着想はリバースモード（reverse mode）の微分にすでに含まれている。リバースモードは、ある層の活性化値に対する損失の導関数（derivative）を層から層へと逆向きに伝えていくものであり、34.3.2 節ではこれをノードの随伴変数と呼んだ。随伴法は、あらゆる瞬間に 1 つずつノード $\mathbf{x}(t)$ があると考えて、同じ量を時間をさかのぼって運ぶ。中心となる対象は**随伴状態**（adjoint state）$\mathbf{a}(t) = \partial \mathcal{L} / \partial \mathbf{x}(t)$、すなわち時刻 t の状態に関する損失の勾配であり、これ自体がある線形常微分方程式に従うことがわかる。

38.3.2.1 スカラーの場合

一般論を述べる前に、すべての量が閉形式で書ける具体例を 1 つ最後まで計算してみよう。$\dot{x} = \theta x$、初期条件 $x(0) = x_0$、終端点での損失を $\mathcal{L} = \frac{1}{2} x(T)^2$ とする。直接計算すれば $x(T) = e^{\theta T} x_0$ であるから、$\partial x(T) / \partial \theta = T\, x(T)$ となり、次式が得られる。

$$\frac{\partial \mathcal{L}}{\partial \theta} = x(T)\, \frac{\partial x(T)}{\partial \theta} = T\, x(T)^2 .$$

$\theta = T = x_0 = 1$ のとき、この値は $e^2 = 7.389$ である。

随伴法は別の道筋で同じ数値にたどり着く。時刻 t の状態に加えた微小変化 δ は、時刻 T までに $e^{\theta (T-t)} \delta$ へと拡大し、損失を $x(T)\, e^{\theta (T-t)} \delta$ だけ変化させるため、$a(t) = e^{\theta (T-t)} x(T)$ となる。この随伴状態は、ODE $\dot{a} = -\theta a$ を終端値 $a(T) = x(T)$（損失 $\ell(x) = \frac{1}{2} x^2$ の $x(T)$ における導関数の値）から逆向きに解いた解になっている。一方、パラメータの微小変化 $\delta\theta$ は各瞬間の速度を $x(t)\, \delta\theta$ だけ変え（$\partial f / \partial \theta = x$ であるため）、随伴状態がこれらの微小変化をそれぞれ損失の変化へと変換する。時間全体にわたって足し合わせると、次のようになる。

$$\int_0^T a(t)\, \frac{\partial f}{\partial \theta}\, dt = \int_0^T e^{\theta (T-t)} x(T)\; e^{\theta t} x_0 \, dt = \int_0^T x(T)^2 \, dt = T\, x(T)^2 . \tag{38.16}$$

被積分関数は定数になっている。順方向に読むと x が増大する一方で a は同じ因子だけ縮小し、両者の積は常に $x(T)^2$ に保たれる（Figure 38.9）。求める勾配はその下の面積である。

38.3.2.2 随伴方程式

一般の場合もスカラーの場合と同じ構造を保つ。すなわち、損失の勾配から逆向きに走る随伴状態と、その随伴状態を $\partial \mathbf{f} / \partial \theta$ に対して積分するパラメータ勾配である。

命題（随伴方程式）$\mathbf{f}$ は $\mathbf{x}$ と θ について連続微分可能とし、$\mathbf{x}(t)$ は $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \theta, t)$（パラメータを引数として明記したベクトル場 $\mathbf{f}_\theta$）の区間 $[0, T]$ における解（固定された初期値 $\mathbf{x}(0)$ から出発するもの）とする。また、$\mathcal{L} = \ell(\mathbf{x}(T))$（ただし ℓ は微分可能）とおく。このとき、随伴状態は次の線形 ODE を満たす。

$$\dot{\mathbf{a}}(t) = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{\top} \mathbf{a}(t), \qquad \mathbf{a}(T) = \nabla \ell(\mathbf{x}(T)), \tag{38.17}$$

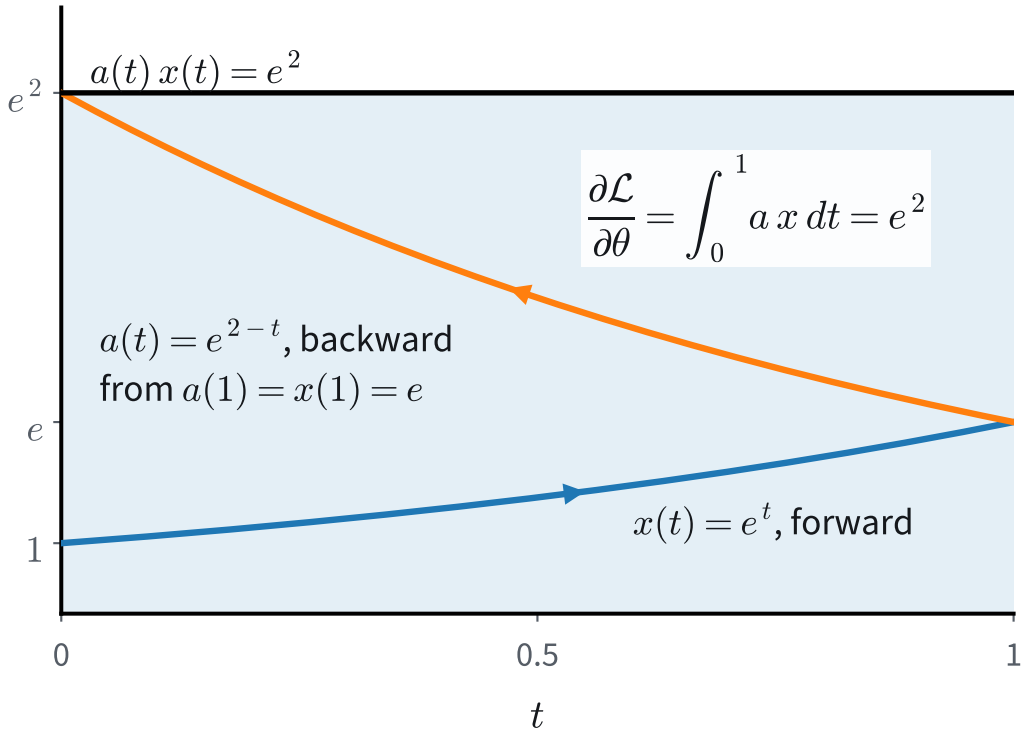


Figure 38.9: $\dot{x} = \theta x$（$\theta = 1$、$x_0 = 1$、$T = 1$）に対するスカラー随伴状態。状態は順方向に増大し、随伴状態は損失の勾配 $a(1) = x(1)$ から逆向きに積分され、両者の積は一定となる。パラメータ勾配は網掛け部分の面積 $Tx(T)^2 = e^2$ である。

ただしこれは時刻 T から逆向きに積分され、パラメータ勾配は次の積分で与えられる。

$$\frac{\partial \mathcal{L}}{\partial \theta} = \int_0^T \left(\frac{\partial \mathbf{f}}{\partial \theta}\right)^{\top} \mathbf{a}(t)\, dt, \tag{38.18}$$

ここで両方の偏微分（partial derivative）は軌跡 $(\mathbf{x}(t), \theta, t)$ に沿って評価される (Chen et al. 2018)。

証明（変分法による証明） パラメータに無限小の摂動 $\delta\theta$ を加える。$\mathbf{f}$ は C^1 級であるため解は θ について微分可能であり、軌跡の 1 次摂動 $\delta\mathbf{x}(t)$ はダイナミクスの**線形化**（linearization）に従う（ODE を θ で微分し、連鎖律（chain rule）を適用する）。

$$\delta\dot{\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\,\delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \theta}\,\delta\theta, \qquad \delta\mathbf{x}(0) = \mathbf{0}.$$

逆向き ODE Equation 38.17 によって $\mathbf{a}(t)$ を定義し、内積 $\mathbf{a}^{\top}\delta\mathbf{x}$ を時間で微分する。

$$\begin{aligned}\frac{d}{dt}\left(\mathbf{a}^{\top}\delta\mathbf{x}\right) &= \dot{\mathbf{a}}^{\top}\delta\mathbf{x} + \mathbf{a}^{\top}\delta\dot{\mathbf{x}} \\ &= -\mathbf{a}^{\top}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\,\delta\mathbf{x} + \mathbf{a}^{\top}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\,\delta\mathbf{x} + \mathbf{a}^{\top}\frac{\partial \mathbf{f}}{\partial \theta}\,\delta\theta = \mathbf{a}^{\top}\frac{\partial \mathbf{f}}{\partial \theta}\,\delta\theta.\end{aligned}$$

状態どうしの結合項は打ち消し合うが、随伴 ODE はまさにそうなるように定義したのである。0 から T まで積分し、左端で $\delta\mathbf{x}(0) = \mathbf{0}$ であることを用いると次式を得る。

$$\mathbf{a}(T)^{\top}\delta\mathbf{x}(T) = \int_0^T \mathbf{a}(t)^{\top}\frac{\partial \mathbf{f}}{\partial \theta}\, dt\ \delta\theta.$$

左辺は損失の 1 次変化 $\nabla \ell(\mathbf{x}(T))^{\top}\delta\mathbf{x}(T) = \delta\mathcal{L}$ であり、右辺は $\delta\theta$ に関して線形であって、その係数（coefficient）が Equation 38.18 である。残るは $\mathbf{a}$ が随伴状態に一致することの確認である。$\delta\theta = \mathbf{0}$ とし、微小変化 $\delta\mathbf{x}(s)$ を時刻 s で注入して同じ議論を行うと、内積 $\mathbf{a}^{\top}\delta\mathbf{x}$ は $[s, T]$ 上で一定となる。ゆえに $\delta\mathcal{L} = \mathbf{a}(T)^{\top}\delta\mathbf{x}(T) = \mathbf{a}(s)^{\top}\delta\mathbf{x}(s)$ が成り立ち、Equation 38.17 の解は $\partial\mathcal{L}/\partial\mathbf{x}(s)$ となる。■

この逆向き ODE は $\mathbf{a}$ に関して線形であり、転置されたヤコビ行列をその係数行列に持つ。各ステップでは 1 回のベクトル・ヤコビ行列積（vector–Jacobian product）$\mathbf{a}^{\top}\partial\mathbf{f}/\partial\mathbf{x}$ を計算するが、そのコストは $\mathbf{f}$ に対するリバースモードの計算 1 回分である（34.5 節）。同じパスで、パラメータが何個あろうともパラメータ勾配の被積分関数 $(\partial\mathbf{f}/\partial\theta)^{\top}\mathbf{a}$ が同時に求まる（なお、ReLU を用いたベクトル場は区分的 C^1 級にとどまるため、展開されたソルバに対して自動微分を区分ごとに適用することはできても、上の命題の適用対象には含まれない。tanh や GELU のような滑らかな活性化関数であれば命題の仮定を満たす）。

38.3.2.3 離散随伴としての誤差逆伝播法

これで、オイラー法のプログラムに対する誤差逆伝播法（backpropagation）を随伴 ODE と 1 行ずつ並べて比較できる。式を簡潔にするため、ベクトル場は自律的であるとし、そのヤコビ行列を $\mathbf{J} = \partial\mathbf{f}/\partial\mathbf{x}$ と書く。プログラム $\mathbf{x}_{n+1} = \mathbf{x}_n + h\,\mathbf{f}(\mathbf{x}_n, \theta)$ を連鎖律で微分し、$\mathbf{a}_n = \partial\mathcal{L}/\partial\mathbf{x}_n$ とおく。1 ステップのヤコビ行列は $\mathbf{I} + h\mathbf{J}(\mathbf{x}_n)$ であるから、次式が成り立つ。

$$\begin{aligned} \mathbf{a}_n &= \big(\mathbf{I} + h\mathbf{J}(\mathbf{x}_n)\big)^{\top}\mathbf{a}_{n+1}, \qquad \mathbf{a}_N = \nabla \ell(\mathbf{x}_N), \\ \frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{n=0}^{N-1} h \left(\frac{\partial \mathbf{f}}{\partial \theta}(\mathbf{x}_n)\right)^{\top} \mathbf{a}_{n+1}. \end{aligned} \tag{38.19}$$

これを、随伴 ODE Equation 38.17 を t_{n+1} から t_n へ逆向きに 1 ステップ進める陽的オイラーステップと比較してみよう。

$$\mathbf{a}_n = \mathbf{a}_{n+1} + h\,\mathbf{J}(\mathbf{x}_{n+1})^{\top}\mathbf{a}_{n+1}.$$

2 つの更新式の違いは、ヤコビ行列をどこで評価するか、すなわち順方向ステップの始点 $\mathbf{x}_n$ か終点 $\mathbf{x}_{n+1}$ かという点だけである。また、Equation 38.19 の和は積分 Equation 38.18 に対する長方形近似になっている。したがって、ソルバを通した誤差逆伝播法は**離散随伴**（discrete adjoint）であり、随伴 ODE にソルバを適用したものは**連続随伴**（continuous adjoint）を離散化したものである。文献ではこの 2 つの流儀を、それぞれ離散化してから最適化（discretize-then-optimize、先ほどの訓練における「離散化してから微分する」にあたる）および最適化してから離散化（optimize-then-discretize）と呼ぶ (Kidger 2022)。前者はプログラムの厳密な勾配であり、後者はフローの勾配を近似するものであって、オイラー法では両者の勾配は h の 1 次のオーダーで食い違う。本項の内容のなかで最も重要なのはこの事実である。

スカラーの例を見れば両方の勾配が一目でわかる。$\dot{x} = \theta x$ に対するオイラー法のプログラムは $x_N = (1+h\theta)^N x_0$ であるから、$\partial x_N/\partial\theta = Nh(1+h\theta)^{N-1}x_0 = T\,x_N/(1+h\theta)$ となり、離散随伴からは $\partial\mathcal{L}/\partial\theta = T\,x_N^2/(1+h\theta)$ が得られる。$\theta = T = x_0 = 1$ とすると、この値は 6.116（$N=10$）、7.244（$N=100$）、7.374（$N=1000$）となるのに対し、厳密値は $e^2 = 7.389$ である。その差 1.27、0.145、0.0148 は、N を 10 倍にするごとに 10 分の 1 に縮まる。

38.1 節のスパイラルのベクトル場 $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$（ただし $\mathbf{A} = \left(\begin{smallmatrix} -0.5 & -1 \\ 1 & -0.5 \end{smallmatrix}\right)$）を使えば、すべての要素が閉形式で書けたまま 2 次元で同じ検証ができる。線形なベクトル場ではヤコビ行列がどの点でも $\mathbf{A}$ で一定になるため、上に示した 2 つの漸化式は完全に一致する。したがって測るべき対象は、状態・随伴状態・和のすべてにオイラー法を用いたときに、両者がフローの厳密な勾配に対して持つ誤差である。

θ を行列 $\mathbf{A}$ の成分とし、$\mathcal{L} = \tfrac{1}{2}\|\mathbf{x}(T)\|^2$ とおく。軌跡は $e^{\mathbf{A}t}\mathbf{x}_0$、随伴状態は $\mathbf{a}(t) = e^{\mathbf{A}^{\top}(T-t)}\mathbf{x}(T)$（すなわち $\dot{\mathbf{a}} = -\mathbf{A}^{\top}\mathbf{a}$ の解）であり、$\partial\mathbf{f}/\partial A_{ij} = x_j\mathbf{e}_i$ であるから、勾配 Equation 38.18 は $\partial\mathcal{L}/\partial\mathbf{A} = \int_0^T \mathbf{a}(t)\,\mathbf{x}(t)^{\top}dt$ となる。

次のセルでは、逆方向スイープ Equation 38.19 を $T=2$ について手で書き下す。出力の 1 行目は、同じプログラムに対する中心有限差分（finite difference）とこのスイープを $N=1000$ ステップで比較するものであるが、これで確認できるのはスイープの実装に誤りがないことだけである。本当に重要なのは、厳密なフローの勾配との距離である。そのためにセルの後半では、連続随伴の積分を 2000 区間の台形公式で評価し（$\mathbf{a}$ と $\mathbf{x}$ は行列指数関数から計算）、3 通りのステップ数における離散随伴と比較する。

```
def loss_and_grad_unrolled(A, x0, T, N):
    """L = ||x_N||^2 / 2 through N unrolled Euler steps; dL/dA by hand-written
    reverse mode, the discrete adjoint."""
    h, xs = T / N, [x0]
    for _ in range(N):
```

```
        xs.append(xs[-1] + h * (A @ xs[-1]))   # forward sweep, tape the states
    a, gA = xs[-1].copy(), np.zeros_like(A)    # a_N = dL/dx_N = x_N
    for n in range(N - 1, -1, -1):             # backward sweep: one VJP per step
        gA += h * np.outer(a, xs[n])           # dL/dA += h a_{n+1} x_n^T
        a = a + h * (A.T @ a)                  # a_n = (I + hA)^T a_{n+1}
    return 0.5 * xs[-1] @ xs[-1], gA

T = 2.0
loss = lambda A: loss_and_grad_unrolled(A, x0, T, 1000)[0]
g_fd = np.array([(loss(A + 1e-5 * E) - loss(A - 1e-5 * E)) / 2e-5
                 for E in np.eye(4).reshape(4, 2, 2)]).reshape(2, 2)
gap_fd = np.abs(loss_and_grad_unrolled(A, x0, T, 1000)[1] - g_fd).max()
print(f'N =  1000: |backward sweep - central differences| = {gap_fd:.1e}')
n_quad = 2000                                  # continuous adjoint, trapezoid rule
ts = np.linspace(0.0, T, n_quad + 1)
xT = expm_eig(A, T) @ x0
AX = np.stack([np.outer(expm_eig(A.T, T - s) @ xT, expm_eig(A, s) @ x0)
               for s in ts])                   # a(s) x(s)^T
g_cont = (T / n_quad) * (AX[1:-1].sum(axis=0) + 0.5 * (AX[0] + AX[-1]))
for N in (100, 1000, 10000):
    gap = np.abs(loss_and_grad_unrolled(A, x0, T, N)[1] - g_cont).max()
    print(f'N = {N:5d}: |discrete - continuous adjoint|     = {gap:.1e}')
```

```
N =  1000: |backward sweep - central differences| = 1.2e-10
N =   100: |discrete - continuous adjoint|     = 2.9e-02
N =  1000: |discrete - continuous adjoint|     = 2.8e-03
N = 10000: |discrete - continuous adjoint|     = 2.8e-04
```

逆方向スイープは有限差分と約 10^{-10} の桁まで一致しており、これは有限差分自体の精度に等しい。一方、フローの勾配との差は 2.9×10^{-2} から 2.8×10^{-3}、さらに 2.8×10^{-4} へと減少し、スカラーの場合と同様に N を 10 倍にするごとに 10 分の 1 になる。

38.3.2.4 メモリとドリフト

随伴 ODE はヤコビ行列を評価するため、逆方向の各ステップで $\mathbf{x}(t)$ を必要とする。これを用意する方法は 2 つある。1 つは誤差逆伝播法と同様に順方向の状態をすべて保存しておく方法で、N に比例するメモリを消費する。もう 1 つは $\dot{\mathbf{x}} = \mathbf{f}$ を $\mathbf{x}(T)$ から $\mathbf{a}$ と一緒に逆向きに再積分する方法で、1 ステップあたりベクトル場の評価が 1 回増える代わりにメモリ使用量を N に対して定数に抑えられる。後者はうますぎる話に聞こえるが、最小の具体例を見ればその裏がわかる。誤差を増幅してしまうのである。

$\dot{x} = -\lambda x$（ただし $\lambda > 0$）に対し、逆方向の積分は $\dot{x} = +\lambda x$ を解くことになる。したがって、保存された $x(T)$ に含まれる絶対誤差 δ は、$t = 0$ に戻るころには $e^{\lambda T}\delta$ に膨れ上がる。

では、δ はどれほどの大きさだろうか。2 つのモードを持つ状態を想像してみよう。一方は $\lambda = 50$ という速い減衰率（38.2.3 節の硬い方程式の例における速いレート）で減衰し、もう一方はほとんど動かないとする。速いモードが対角方向を向いており、どの座標にもその成分が含まれているとしよう。時刻 T までに速いモードは減衰し切り、保存される各座標の大きさはオーダー 1 となって、倍精度（double precision）ではそれぞれ約 10^{-16} の桁で丸められる。丸め処理はモードの存在など知らないため、丸め誤差の一部は速いモードの方向に乗り、$\delta \approx 10^{-16}$ となる。その結果、再構成された初期値は 2.7×10^{-11}（$T = 0.25$）、7×10^{-6}（$T = 0.5$）、さらには 5×10^{5}（$T = 1$）もずれてしまい、状態そのものより大きくなってしまう。倍精度浮動小数点数

が保持できるのは約 16 桁であるのに対し、増幅率 $e^{50} \approx 5 \times 10^{21}$ はそれを 5 桁も上回るからである。演習問題 3 ではこの誤差の増大を実際に測定する。

固有値（eigenvalue）の実部が -0.5 であるスパイラルの例では、$T = 2$ にわたる増幅率はわずか e にとどまる。順方向に強く収縮するフローは逆方向には強く膨張するため、再積分はメモリを節約する引き換えにドリフトを招く。硬いベクトル場に対しては、**チェックポイント保存**（checkpointing、34.5.2.5 節）の方が堅牢な選択である。少数のチェックポイントでのみ状態を保存し、その間の状態は直前のチェックポイントから順方向に再計算するため、フローに逆らって積分することが一切ない。$\sqrt{N}$ ステップごとにチェックポイントを置けば、逆方向スイープが保持する状態数は約 $2\sqrt{N}$ 個、すなわち 63 個（1000 個ではなく、$N = 1000$ の場合）で済み、追加コストは順伝播 1 回分のみである。

以上で勾配を求める方法が 3 通り出そろった。どれを使うべきだろうか。答えは、ソルバを通した誤差逆伝播法に、メモリが足りないときはチェックポイント保存を組み合わせる方法である。これなら実際に実行されるプログラムの厳密な勾配が返される。連続随伴はその勾配の収束先であり、理論的な考察を行うための式である。状態の逆方向再積分は、硬くないベクトル場に限って使える最後の手段にすぎない。

38.3.3 連続正規化フロー

ニューラル ODE は点を移動させる。最尤法で訓練される生成モデル（generative model）は、それぞれの出力がどれほど確からしいかも答えなければならないため、ここからはフローに沿った確率密度の変化を追っていく。

38.3.3.1 フローに沿った密度

サンプルをフローに通して押し出せば、フローは生成器（generator）になる。$\mathbf{x}(0)$ を単純な密度 p_0 からサンプリングして積分すればよい。このような生成器を最尤法で訓練するには、出力される点の密度が必要になる。**正規化フロー**とはこのようにして訓練される可逆なネットワークであり、その密度は 36.2 節の変数変換の公式から求まる。すなわち、可逆で微分可能な写像 $\mathbf{g}$（ヤコビ行列を $\mathbf{J}_{\mathbf{g}}$ とする）に対し、次式が成り立つ。

$$\log p_{\text{out}}(\mathbf{g}(\mathbf{x})) = \log p_{\text{in}}(\mathbf{x}) - \log \left| \det \mathbf{J}_{\mathbf{g}}(\mathbf{x}) \right| . \tag{38.20}$$

対数密度（log density）は、$\mathbf{g}$ が $\mathbf{x}$ の近傍の体積を引き伸ばす倍率の対数だけ減少する。多数の層から構築された正規化フローでは、層ごと・サンプルごとに 1 回ずつ対数行列式を計算しなければならない。一般の行列式には $O(d^3)$ 回の演算がかかり、約 2×10^{10} 回にも達する（次元 $d = 3072$ の 32×32 カラー画像の場合）。そのため離散フローでは、行列式が対角成分の積になる三角ヤコビ行列や、$\det(\mathbf{I} + \mathbf{u}\mathbf{v}^{\top}) = 1 + \mathbf{v}^{\top}\mathbf{u}$ のような「単位行列＋低ランク行列」のヤコビ行列を持つ層へとアーキテクチャを制限する。これらはいずれも $O(d)$ で行列式を計算できる。

連続時間ではこの公式が単純になる。まず 1 次元から始めよう。ベクトル場 $\dot{x} = \theta x$（$\mathbb{R}$ 上）のフローは $\Phi_t(x) = e^{\theta t} x$ であり、あらゆる区間を $e^{\theta t}$ 倍に引き伸ばす。フローに運ばれる密度はその引き伸ばし率で割られるため、$\log p_t(x(t)) = \log p_0(x_0) - \theta t$ となり、したがって $\frac{d}{dt} \log p_t(x(t)) = -\theta$ である。この定数 θ はベクトル場の 1×1 ヤコビ行列であり、ゆえにそのトレースでもある。

2 次元では引き続きスパイラルを例にとろう。そのフロー $e^{\mathbf{A}t} = e^{-t/2} e^{t\mathbf{K}}$ は回転のあとに $e^{-t/2}$ 倍の縮小を施す変換である（38.1 節）。円板は半径が $e^{-t/2}$ 倍の円板へと移されるため、面積は e^{-t} 倍になり、各軌跡に沿った密度は e^{t} 倍に上昇する。ここで $\operatorname{tr} \mathbf{A} = -1$ であるから、対数密度の変化率はやはりトレースに負号をつけたものに一致する。Figure 38.10 は、この様子を面積保

存のベクトル場および面積を膨張させるベクトル場と並べて描いたものである。スパイラルの最も濃い円板（$t = 1.2$）の面積は、最も薄い円板の $e^{-1.2} \approx 0.30$ 倍になっている。

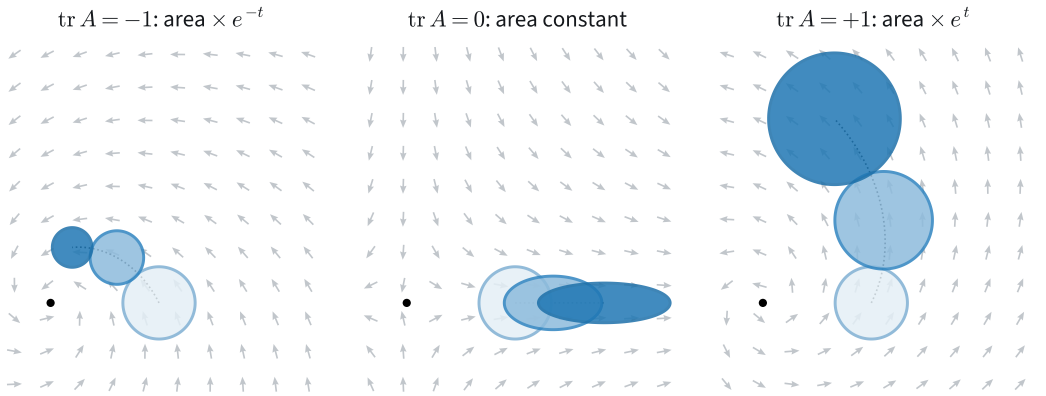


Figure 38.10: 3 つの線形ベクトル場の厳密なフローによって運ばれるサンプルの円板（$t = 0$、0.6、1.2 を表示し、色が濃いほど後の時刻を表す。点線は中心の軌跡）。軌跡に沿った密度は因子 $e^{-t\,\mathrm{tr}\,\mathbf{A}}$ 倍に変化する。フローが収縮するところでは密度が上昇し（左：スパイラル）、面積保存の鞍点では形状が変わっても密度は一定に保たれ（中央）、フローが膨張するところでは密度が低下する（右）。

38.3.3.2 瞬間的な変数変換

上の 2 つの例はいずれも線形であり、どちらでも対数密度の変化率はトレースの負値に等しかった。一般の場合を示すには、行列式に関する事実が 1 つあればよい。短いステップの間ではフロー写像（flow map）は恒等写像に近く、単位行列に近い行列の行列式はそのトレースによって決まるのである。

補題 任意の正方行列 $\mathbf{J}$ に対して、$\det(\mathbf{I} + h\mathbf{J}) = 1 + h\,\mathrm{tr}\,\mathbf{J} + O(h^2)$ が成り立つ。

証明置換による行列式の定義式（33.2.3.2 節）は、各行・各列から 1 つずつ成分を選んだ積に符号をつけて、すべての置換にわたり足し合わせたものである。恒等置換以外の任意の置換は少なくとも 2 つの非対角成分を選ぶが、それらはそれぞれ $O(h)$ の大きさを持つため、寄与は $O(h^2)$ となる。恒等置換からの寄与は $\prod_i (1 + hJ_{ii}) = 1 + h\sum_i J_{ii} + O(h^2)$ である。■

命題（瞬間的な変数変換） $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ において、$\mathbf{f}$ は $(\mathbf{x}, t)$ について連続微分可能かつ $\mathbf{x}$ について大域的リプシッツ連続であるとする。このとき各フロー写像 Φ_t は C^1 級の全単射であり、その逆写像も C^1 級となる（C^1 級のベクトル場の解は初期値に関して C^1 級であり (Hairer et al. 1993)、逆写像はベクトル場を逆向きに走らせるためである）。また、$\mathbf{x}(0) \sim p_0$ とする（ただし p_0 は正の値をとる微分可能な密度）。このとき密度 p_t（$\mathbf{x}(t)$ の確率密度）が存在して正の値をとり、各軌跡に沿って次式が成り立つ (Chen et al. 2018)。

$$\frac{d}{dt} \log p_t(\mathbf{x}(t)) = -\,\mathrm{tr}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)(\mathbf{x}(t), t). \tag{38.21}$$

証明密度の存在は Equation 38.20 を全単射 Φ_t に適用することで従い、それが正であることは、p_0 が正であり微分可能な逆写像を持つ写像のヤコビ行列が正則であることから従う。短い時間区間において、フローは写像 $\Phi_{t \to t+h}(\mathbf{x}) = \mathbf{x} + \int_t^{t+h} \mathbf{f}(\mathbf{x}(s), s)\, ds$ によって進む。$\mathbf{x}$ で微分し $\mathbf{J}$ の

連続性を用いると、そのヤコビ行列は $\mathbf{I} + h\mathbf{J} + o(h)$ となる。ただし $\mathbf{J} = \partial \mathbf{f}/\partial \mathbf{x}$ は $(\mathbf{x}(t), t)$ で評価したものであり、$o(h)$ は h より速く 0 に収束する項を表す。この写像に Equation 38.20 を適用し、上の補題を用いて展開する（補題は高次の摂動を許容するため、$h\mathbf{J}$ を $h\mathbf{J} + o(h)$ に置き換えても行列式の変化は $o(h)$ にとどまる）。十分小さい h に対して行列式は 1 に近く正になるため、絶対値は外せて次式となる。

$$\begin{aligned} \log p_{t+h}(\mathbf{x}(t+h)) &= \log p_t(\mathbf{x}(t)) - \log\det(\mathbf{I} + h\mathbf{J} + o(h)) \\ &= \log p_t(\mathbf{x}(t)) - h \operatorname{tr}\mathbf{J} + o(h), \end{aligned}$$

ここでは $\log(1+u) = u + O(u^2)$ を用いた。両辺から $\log p_t(\mathbf{x}(t))$ を引き、h で割って $h \to 0$ とすればよい。■

命題の仮定は見た目ほど厳しいものではない。求めているのはすべてのフロー写像が滑らかな全単射であることだけであり、tanh のような滑らかでリプシッツ連続な活性化関数を持つベクトル場であれば自然に満たされる。命題の実質的な内容は、トレースを含むあの 1 行の式にある。

ヤコビ行列のトレースはベクトル場の**発散**（divergence）$\nabla \cdot \mathbf{f}$ であり、フローが局所的に体積を膨張させる変化率を表す。ベクトル場がサンプルを押し広げるところ（$\operatorname{tr}\mathbf{J} > 0$）では各軌跡に沿った対数密度は減少し、サンプルを圧縮するところでは上昇する。Equation 38.21 を軌跡に沿って積分すると次式が得られる。

$$\log p_T(\mathbf{x}(T)) = \log p_0(\mathbf{x}(0)) - \int_0^T \operatorname{tr}\!\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right) dt. \tag{38.22}$$

離散フローにおける層ごとの対数行列式の和は、トレースの積分へと姿を変えた。任意の ODE ソルバで対 $(\mathbf{x}, \log p)$ を積分することによりこの量を計算するモデルが、**連続正規化フロー**（continuous normalizing flow、CNF）である (Chen et al. 2018)。このモデルでは $\mathbf{f}$ のアーキテクチャに一切の制約を課す必要がない。与えられた点 $\mathbf{x}(T)$ の密度を評価するには、その点から $\mathbf{x}(0)$ までベクトル場を逆向きに積分しながら途中でトレースを累積し、到達点での p_0 を読み取ればよい。拡散モデルの尤度もこの方法で計算される（17.6 節）。この公式には時間の向きに対する非対称性はなく、38.6 節や 38.7 節の生成器のようにノイズを終端時刻に置きデータを $t = 0$ に置く設定でも、まったく同じ公式がそのまま使える。

線形なベクトル場ではトレースの積分は $t \operatorname{tr}\mathbf{A}$ となり、Equation 38.22 を Equation 38.20（写像 $e^{\mathbf{A}t}$ に対するもの）と見比べれば $\log\det e^{\mathbf{A}t} = t \operatorname{tr}\mathbf{A}$ が得られる。対角化可能な $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ に対しては、この恒等式は固有値から 1 行で導くこともできる。

$$\det e^{\mathbf{A}t} = \det\left(\mathbf{V} e^{\mathbf{\Lambda} t}\mathbf{V}^{-1}\right) = \prod_{j} e^{\Lambda_{jj} t} = e^{t \operatorname{tr}\mathbf{A}}, \tag{38.23}$$

一方、先ほどの 2 つの公式の比較からは、対角化可能性を問わずあらゆる $\mathbf{A}$ に対してこの恒等式が従う。

上の命題は軌跡を 1 本ずつ追跡したものである。同じ法則を密度全体に対して一度に書き下せば流体力学における連続の式（continuity equation）となり、38.5 節ではノイズが加える項と合わせてこれを導出する。

38.3.3.3 トレース積分の検証

線形なベクトル場はトレースが定数で積分が単なる $t\,\mathrm{tr}\,\mathbf{A}$ になるため、命題の検証としては物足りない。経路に沿ってトレースが変化する非線形なベクトル場こそ本格的なテストになる。平面上のベクトル場 $\mathbf{f}(\mathbf{x}) = \mathbf{W}_2 \tanh(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$（隠れユニット（hidden unit）数 8、重みはランダム）を考えよう。ヤコビ行列とトレースは次のように陽に書ける。

$$\mathbf{J} = \mathbf{W}_2\,\mathrm{diag}(\mathbf{s})\,\mathbf{W}_1, \qquad \mathbf{s} = 1 - \tanh^2(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \qquad \mathrm{tr}\,\mathbf{J} = \sum_i s_i\,(\mathbf{W}_1\mathbf{W}_2)_{ii},$$

ここではトレースの巡回性 $\mathrm{tr}\,(\mathbf{W}_2\,\mathrm{diag}(\mathbf{s})\,\mathbf{W}_1) = \mathrm{tr}\,(\mathrm{diag}(\mathbf{s})\,\mathbf{W}_1\mathbf{W}_2)$ を用いた。次のセルでは、対 $(\mathbf{x}, \int \mathrm{tr}\,\mathbf{J}\,dt)$ を RK4 で $T = 1$ まで積分し、その経路を記録する。途中で現れたトレースの最小値と最大値を出力したうえで、トレース積分を $\log|\det D\Phi_T|$ と比較する。ただし、フロー写像のヤコビ行列 $D\Phi_T$ は、フロー写像自体の中心有限差分（finite difference）から計算する。

```
n_hidden = 8
W1, b1 = rng.standard_normal((n_hidden, 2)), rng.standard_normal(n_hidden)
W2 = rng.standard_normal((2, n_hidden)) / np.sqrt(n_hidden)
field = lambda x: W2 @ np.tanh(W1 @ x + b1)
trJ = lambda x: np.sum((1 - np.tanh(W1 @ x + b1) ** 2)
                       * np.einsum('ij,ji->i', W1, W2))
aug = lambda z: np.append(field(z[:2]), trJ(z[:2]))  # d/dt (x, int tr J)
xa, N = rng.standard_normal(2), 200
zs = [np.append(xa, 0.0)]                           # RK4 path of (x, int tr J)
for _ in range(N):
    zs.append(rk4_step(aug, zs[-1], 1.0 / N))
zT, tr_path = zs[-1], [trJ(z[:2]) for z in zs]
D = np.zeros((2, 2))                                # D Phi_1 by central differences
for j in range(2):
    e = np.zeros(2); e[j] = 1e-5
    D[:, j] = (solve(rk4_step, field, xa + e, 1.0, N)
               - solve(rk4_step, field, xa - e, 1.0, N)) / 2e-5
logdet = np.log(abs(np.linalg.det(D)))
print(f'tr J along the path: from {min(tr_path):+.3f} to {max(tr_path):+.3f}')
print(f'trace integral {zT[2]:+.6f}  log|det D Phi| {logdet:+.6f}  '
      f'gap {abs(zT[2] - logdet):.1e}')
```

```
tr J along the path: from +0.889 to +1.013
trace integral +0.970187  log|det D Phi| +0.970187  gap 8.3e-11
```

トレースは確かに変動しており、1 行目の出力には経路上でトレースが動いた区間が示される。その幅は重みの乱数シードに依存するが、0.1 以上動くのが普通であり、乱数によっては区間が 0 をまたいで、経路のある部分では面積を膨張させ別の部分では圧縮することもある。それでもトレース積分と対数行列式は 8 桁以上の精度で一致しており、これはソルバと有限差分自体の精度に達している。命題の証明は線形性を一切使っていなかったのである。

38.3.3.4 Hutchinson トレース推定量

上の $\mathrm{tr}\,\mathbf{J}$ の式は、隠れ層 1 層のベクトル場という特別な構造を利用したものであった。一般のネットワークでトレースを求めるには、d 個ある $\mathbf{J}$ の対角成分が必要になり、座標ごとに 1 回ずつ、計 d 回のリバースモード計算がかかる。トレースへの書き換えによって $O(d^3)$ の行列式はすで

に d 回のパスへと削減されたが、ランダム射影（projection）を使えばこの d 回のパスをさらに 1 回へと減らし、ノイズは乗るものの不偏な推定値を得ることができる (Hutchinson 1989)。

命題（Hutchinson 推定量） 任意の行列 $\mathbf{M} \in \mathbb{R}^{d\times d}$ と、任意の確率ベクトル（random vector）$\epsilon \in \mathbb{R}^d$（ただし $\mathbb{E}[\epsilon\epsilon^\top] = \mathbf{I}$ を満たすもの。たとえば標準ガウスベクトルや、独立に等確率で ± 1 をとるラデマッハベクトル）に対し、次式が成り立つ。

$$\operatorname{tr}(\mathbf{M}) = \mathbb{E}\left[\epsilon^\top \mathbf{M} \epsilon\right]. \tag{38.24}$$

証明 トレースの巡回性と、期待値（expectation）との交換を可能にするトレースの線形性により、$\mathbb{E}[\epsilon^\top \mathbf{M}\epsilon] = \mathbb{E}[\operatorname{tr}(\mathbf{M}\epsilon\epsilon^\top)] = \operatorname{tr}(\mathbf{M}\,\mathbb{E}[\epsilon\epsilon^\top]) = \operatorname{tr}(\mathbf{M})$ が成り立つ。$\blacksquare$

この推定量が $\mathbf{J}$ に触れるのは 1 回のベクトル・ヤコビ行列積 $\epsilon^\top \mathbf{J}$ を通じてのみである。これは $\mathbf{f}$ に対して ϵ をシードとするリバースモード計算を 1 回行うことに相当し、ヤコビ行列自体は一切構成しない。隠れ層が 1 つのベクトル場では、この積は次のように陽に書ける。

$$\epsilon^\top \mathbf{J} = \left((\epsilon^\top \mathbf{W}_2) \odot \mathbf{s}\right) \mathbf{W}_1,$$

その計算コストは $\mathbf{f}$ の評価とほぼ同じである。CNF はベクトル場を評価するたびにこの積を 1 回計算するため、尤度の計算に必要なのはベクトル場の評価 1 回あたり追加の逆伝播 1 回分だけである。これが Grathwohl et al. (2019) によって FFJORD と名付けられた手法である。

上の命題はプローブに関する平均がトレースに収束することを保証するが、その収束の速さまでは述べていない。プローブを 1 個しか使わない場合、まさにそのばらつきこそが問題のすべてとなる。$\mathbf{S} = (\mathbf{J} + \mathbf{J}^\top)/2$ とおくと、1 個のラデマッハプローブの分散（variance）は $2\sum_{i\neq j} S_{ij}^2$ であり、1 個のガウスプローブの分散は対角成分まで含めた同じ形の和 $2\sum_{i,j} S_{ij}^2$ となる（演習問題 5）。トレースが見ているのは $\mathbf{J}$ の対角成分だけであり、ラデマッハプローブのノイズが見ているのは非対角成分だけである。

トレースの値が予測できる $d = 50$ 次元のベクトル場でこの推定量を試してみよう。64 個の tanh ユニットを持つランダムな 1 隠れ層部分に原点への引き戻し項を加えた $\mathbf{f}(\mathbf{x}) = \mathbf{W}_2 \tanh(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) - \mathbf{x}$ を考える。ヤコビ行列は $\mathbf{W}_2 \operatorname{diag}(\mathbf{s})\,\mathbf{W}_1 - \mathbf{I}$ となるため、引き戻し項はトレースに -50 寄与し、ランダムな部分はオーダー 1 の値だけ寄与する。また引き戻し項は各推定値に $-\epsilon^\top\epsilon$ を加える。ラデマッハプローブではこの値が厳密に -50 となり、ノイズはすべてランダムな部分から生じる。一方、ガウスプローブでは $\epsilon^\top\epsilon$ 自体が確率的に揺らぎ、その代償は先ほどの 2 つの分散公式に表れている。両者の差は対角項 $2\sum_i S_{ii}^2$ であり、いま各 $S_{ii} = J_{ii}$ は -1 に近いため、ガウスプローブの分散はラデマッハプローブの分散を約 $2\cdot 50 = 100$ 上回るはずである。次のセルではそれぞれの種類のプローブを $m = 1000$ 個ずつ引き、各 $\epsilon^\top\mathbf{J}$ をリバースモードで計算する。

```
d, n_hidden, m = 50, 64, 1000
W1 = rng.standard_normal((n_hidden, d)) / np.sqrt(d)
b1 = 0.1 * rng.standard_normal(n_hidden)
W2 = rng.standard_normal((d, n_hidden)) / np.sqrt(n_hidden)
x = rng.standard_normal(d)
s = 1 - np.tanh(W1 @ x + b1) ** 2
tr_tanh = np.sum(s * np.einsum('ij,ji->i', W1, W2))  # trace of W2 diag(s) W1
E = rng.choice([-1.0, 1.0], size=(m, d))         # Rademacher probes, one per row
E_gauss = rng.standard_normal((m, d))            # Gaussian probes, one per row
```

ここで、リバースモードのベクトル・ヤコビ行列積 $\epsilon_k^\top \mathbf{J}$ を点 `x` において全 $m = 1000$ 個のラデマッハプローブ（`E`）およびガウスプローブ（`E_gauss`）に対して評価し、得られたトレース推定値

$\epsilon_k^\top \mathbf{J} \epsilon_k$ とプローブ 1 個あたりの標準偏差（standard deviation）を厳密なトレース `tr_tanh - d` と比較する。

```
W1t, b1t, W2t = (torch.tensor(w) for w in (W1, b1, W2))
field = lambda x: W2t @ torch.tanh(W1t @ x + b1t) - x
_, vjp = torch.func.vjp(field, torch.tensor(x))      # reverse mode at x
for name, P in (('Rademacher', E), ('Gaussian', E_gauss)):
    P = torch.tensor(P)
    rows = torch.func.vmap(vjp)(P)[0]                # row k: eps_k^T J
    est = (rows * P).sum(dim=1)                      # eps_k^T J eps_k
    sd = est.std(correction=0)
    print(f'{name:10s}: exact {tr_tanh - d:+6.2f}, mean of {m} probes '
          f'{est.mean():+6.2f} +- {sd / m ** 0.5:.2f}, one probe sd {sd:.2f}')
```

```
Rademacher: exact -51.30, mean of 1000 probes -51.23 +- 0.16, one probe sd 4.99
Gaussian  : exact -51.30, mean of 1000 probes -51.16 +- 0.37, one probe sd 11.62
```

どちらの平均（mean）も厳密なトレースから標準誤差（standard error）の数倍以内に収まっており、不偏推定量として期待されるとおりの振る舞いを示す。ラデマッハプローブ 1 個の標準偏差は約 5、すなわちトレースの 10 分の 1 である。厳密値の計算には 50 回のパスを要するところ、たった 1 回の逆伝播でトレースを 10% 程度の精度で特定できている。一方、ガウスプローブ 1 個の標準偏差は約 11 であり、$11^2 - 5^2 = 96$ となって予測値の 100 に近い。ラデマッハプローブの方がノイジーになることは決してなく、この例でははるかに静かである。

ただし、10% という相対精度はこのベクトル場の性質によるものであり、推定量自体の性質ではない。もし引き戻し項がなければ、すべてのラデマッハ推定値がちょうど $+50$ だけシフトし、標準偏差は 5 のまま変わらない一方、真のトレースはオーダー 1 になるため、単一のプローブからはほとんど何の情報も得られない。高次元のモデルがこれら 2 つの状況のどちらに近いかは、演習問題 5 で扱う。

尤度の計算に必要なのは 1 点でのトレースではなく、軌跡に沿った積分 $\int_0^T \operatorname{tr} \mathbf{J}\, dt$ であり、ソルバのステップごとに 1 回ずつ推定を行うことになる。このときプローブの与え方には、求解全体を通して 1 個のプローブを固定する方式と、ステップごとに新しいプローブを引き直す方式の 2 通りがある。どのような結果が予想されるだろうか。固定プローブは和 $\sum_n h\, \mathbf{J}(\mathbf{x}_n)$ を 1 つの行列とみなしてそのトレースを 1 回推定するのに等しいため、ノイズの大きさは N によらず単一推定のノイズのまま変わらない。これに対し新規プローブでは各ステップの誤差が独立になり、それぞれに $h = 1/N$ の重みがかかるため、和のノイズは $1/\sqrt{N}$ に比例して減少する。次のセルでは、先ほどと同じベクトル場のオイラー軌跡 1 本を $T = 1$ まで $N = 10$、100、1000 ステップでたどり、ラデマッハプローブを用いてそれぞれの方式を 200 回ずつ繰り返す（$\epsilon^\top \mathbf{J}$ には上で導いた陽な積の式を用いる）。

```
field = lambda x: W2 @ np.tanh(W1 @ x + b1) - x
diag_W1W2 = np.einsum('ij,ji->i', W1, W2)
for N in (10, 100, 1000):
    h, xs = 1 / N, [x]
    for _ in range(N - 1):                    # Euler trajectory x_0, ..., x_{N-1}
        xs.append(xs[-1] + h * field(xs[-1]))
    S = 1 - np.tanh(np.array(xs) @ W1.T + b1) ** 2  # s at every step, (N, n_hidden)
    exact = h * (S @ diag_W1W2).sum() - d     # h * sum_n tr J(x_n)
    held = np.broadcast_to(rng.choice([-1.0, 1.0], (200, 1, d)), (200, N, d))
    fresh = rng.choice([-1.0, 1.0], (200, N, d))
```

```
    row = f'N = {N:4d}: exact {exact:+.2f}'
    for name, P in (('held probe', held), ('fresh probes', fresh)):
        est = h * np.einsum('rnh,nh,rnh->r', P @ W2, S, P @ W1.T) - d  # eps^T eps = d
        row += f' | {name} {est.mean():+.2f}, sd {est.std():.2f}'
    print(row)
```

```
N =   10: exact -51.90 | held probe -52.11, sd 5.69 | fresh probes -51.73, sd 1.71
N =  100: exact -51.93 | held probe -51.35, sd 5.98 | fresh probes -51.94, sd 0.63
N = 1000: exact -51.93 | held probe -51.28, sd 5.88 | fresh probes -51.92, sd 0.19
```

厳密な積分値はやはり -50 に近く、どちらの方式の平均も標本誤差（sampling error）の範囲内でこれに一致する。つまり、いずれの方式でも推定は不偏である。ばらつきも予測どおりとなる。固定プローブの標準偏差はどの N でも約 5〜6 であるのに対し、新規プローブでは 1.5〜1.8（$N=10$）、約 0.5〜0.6（$N=100$）、そして 0.2 弱（$N=1000$）となり、どの行でも固定プローブの値を $\sqrt{N}$ で割った大きさに一致している。

$N=1000$ では新規プローブの方が約 30 倍もノイズが小さく、計算コストは同じである。それなのに、なぜわざわざプローブを固定するのだろうか。プローブを固定しておけば、ソルバが解くべき対象が常微分方程式のまま保たれるからである。ϵ を固定すれば推定値 $\epsilon^{\top}\mathbf{J}(\mathbf{x})\epsilon$ は状態の決定論的な関数になるため、どんなソルバでも適用でき、適応的ソルバで誤差を制御することもでき、随伴方程式から勾配も求まる。評価のたびに新しいプローブを引くと右辺にノイズが混入するため、2 回の評価を比較してステップ幅を判定するソルバは、2 回の乱数抽出を比較してしまうことになる。上のセルで新規プローブを使えたのは、オイラーステップを事前に固定していたからにすぎない。FFJORD はプローブを固定し、ノイズを受け入れる設計をとっている。

訓練においてはそのノイズがあっても問題にならない。訓練に必要なのは高精度な尤度ではなく、不偏な尤度だけだからである。トレース推定値は対数尤度（log-likelihood）に線形に入るため、推定された対数尤度とその勾配もやはり不偏となり、確率的勾配（stochastic gradient）にそれ以上の性質は必要ない（35.4 節）。プローブに由来するノイズは、ミニバッチ（minibatch）自体のノイズと同様に、ミニバッチ内および反復間にわたって平均化される。一方、1 個の固定プローブでは求められないのが、単一のデータ点に対する正確な尤度である。対数尤度における 5 の誤差は密度では $e^{5}\approx 150$ 倍もの違いになるため、尤度を評価する際には多数のプローブで平均をとるか、固定したステップのグリッド上で新規プローブを引くか、あるいは厳密なトレースを計算する。

本節で扱ったノイズはプローブのなかにしか存在せず、軌跡そのものは決定論的であった。これに対し拡散モデルはステップごとに新しいノイズを状態そのものに加えるため、$\sqrt{h}$ のスケールで変化する増分によって本節が依拠してきたテイラー展開（Taylor expansion）が成り立たなくなる。そのような方程式を積分するには、独自の微分積分学が必要になる（38.4 節）。

38.3.4 まとめ

残差ブロックはオイラーステップであり、単一のベクトル場でステップ幅を限りなく小さくするとニューラル ODE が得られる。その厳密なフローは可逆であり、ステップ数はソルバに委ねられる。ただし、これら 2 つの利点には限界がある。$\mathbb{R}$ 上のフローは順序を保つため $x\mapsto -x$ には 2 つ目の次元が必要になり、10 ステップのオイラー法を通して訓練したベクトル場は 10 ステップのプログラムに適合するのに対し、RK4 を通して訓練したベクトル場は 4 倍のベクトル場評価と引き換えにフローそのものに適合する。$\mathbf{x}(T)$ 上の損失の勾配は軌跡に沿った $(\partial \mathbf{f}/\partial\theta)^{\top}\mathbf{a}$ の積分で与えられ、随伴状態 $\mathbf{a}$ は $\nabla \ell(\mathbf{x}(T))$ から逆向きに線形 ODE を解くことで定まる。オイラー法を通した誤差逆伝播法はプログラムに対しては厳密であり、フローの勾配からは $O(h)$ 以内に収まる。状態を逆向きに再積分すればメモリを節約できるが、保存時の誤差が $e^{\lambda T}$ 倍に増幅される（減衰率 λ で減衰するモードの場合）。軌跡に沿って $\frac{d}{dt}\log p_t = -\operatorname{tr}\mathbf{J}$ が成り立ち、離散フロー

で行列式を要した箇所がトレースに置き換わる。Hutchinson 推定量はプローブ 1 個あたり 1 回のベクトル・ヤコビ行列積からトレースを統計的バイアスなしに求めるが、そこに乗るノイズは訓練では平均化されて消えるものの、単一の尤度評価では消えない。

38.3.5 演習問題

1. [code] **フローが表現できないもの。**●リプシッツ定数 L を持つベクトル場の 2 本の軌跡は $e^{-Lt}\|\mathbf{x}-\mathbf{y}\| \le \|\Phi_t(\mathbf{x})-\Phi_t(\mathbf{y})\| \le e^{Lt}\|\mathbf{x}-\mathbf{y}\|$ を満たし（演習問題 2、38.1 節）、38.3.1.2 節では $\mathbb{R}$ 上のいかなるフローも順序を逆転できないことを示した (Dupont et al. 2019)。
   1. $d=1$ とする。上の下界を用いて Φ_T が狭義単調増加であることの別証明を与え、$\Phi_T(1)-\Phi_T(-1) \ge 2e^{-LT}$ を示せ。順序と距離という 2 つの主張のうち、本文の命題から得られるのはどちらで、上の評価式を必要とするのはどちらか。
   2. 本文の訓練例は、円上の 2 点 $(0,\pm 1)$（距離 2）を楕円上の 2 点 $(1.5,\pm 0.4)$（距離 0.8）へと移す。時刻 T の厳密なフロー（ベクトル場のリプシッツ定数を L とする）がこれを実現できるのは $LT \ge \ln 2.5 \approx 0.92$ の場合に限られることを示せ。一方、長軸が 2 から 2.4 へ伸びることからは $LT \ge \ln 1.2$ しか得られないことも確認せよ。隠れ層 1 層の tanh ベクトル場は $L \le \|\mathbf{W}_2\|_2\|\mathbf{W}_1\|_2$ を満たす（tanh が 1-リプシッツ連続であるため）。RK4 で訓練したベクトル場についてこの上界を計算し、上の不等式を検証せよ。
   3. $\mathbb{R}$ を $\mathbb{R}^2$ に $x \mapsto (x,0)$ として埋め込む。任意の線形ベクトル場 $\dot{\mathbf{x}}=\mathbf{C}\mathbf{x}$ について、その時刻 1 の写像が $(x,0)$ を $(-x,0)$ へ（すべての x に対して）送るならば $e^{\mathbf{C}}=-\mathbf{I}$ が成り立つことを示せ。**ヒント：固有値 -1 が $e^{\mathbf{C}}$ に現れるには、ある固有値 μ が $\mathbf{C}$ に存在して $e^{\mu}=-1$ を満たす必要があるが、これは実数ではない。**さらに、軌跡が円ではなく楕円となるようなベクトル場を 1 つ求めよ。
   4. 訓練セルを変更し、写像 $x \mapsto -x$ を 64 個の等間隔な点（区間 $[-1,1]$）の上で、入出力次元 1 のベクトル場と $N=10$ 回のオイラーステップを用いて適合させよ。訓練後のモデルは何を出力するか。また、損失が訓練前に予測可能なある値に落ち着くのはなぜか。次元 2 のベクトル場を用い、入力を $(x,0)$ とし、第 1 座標に対して損失をとる設定でも同様の実験を行え。
   5. 1 次元モデルを $N=1$、$h=1$ で再度訓練し、それを $N=1$、および $N=100$ ステップ（サイズ $h=0.01$）で評価せよ。2 つの結果を本文の命題および条件 $hL<1$ と関連付けて説明せよ。

2. **累積損失を伴う随伴法。**●随伴方程式 Equation 38.17 と Equation 38.18 は終端点での損失を扱っている。これに対し、運動エネルギー $\int_0^T \|\mathbf{f}(\mathbf{x}(t),\theta,t)\|^2\,dt$（Finlay et al. (2020)）のような正則化項は、軌跡に沿ってコストを累積する。損失を次のように定める。

$$\mathcal{L} = \ell_T(\mathbf{x}(T)) + \int_0^T \ell(\mathbf{x}(t),\theta,t)\,dt.$$

   1. Equation 38.17 の変分的証明を拡張し、随伴方程式に湧き出し項が加わって $\dot{\mathbf{a}} = -(\partial \mathbf{f}/\partial \mathbf{x})^{\top}\mathbf{a} - \nabla_{\mathbf{x}}\ell$（終端条件 $\mathbf{a}(T)=\nabla \ell_T(\mathbf{x}(T))$）となること、および $\partial \mathcal{L}/\partial \theta = \int_0^T \left((\partial \mathbf{f}/\partial \theta)^{\top}\mathbf{a} + \nabla_{\theta}\ell\right) dt$ が成り立つことを示せ。
   2. Finlay et al. (2020) はその代わりに、座標 E（$\dot{E}=\ell$、$E(0)=0$）を状態に追加し、$E(T)$ を終端損失に加えている。Equation 38.17 を拡大状態 $(\mathbf{x},E)$ に適用し、(a) の結果が再現されることを示せ。
   3. 運動エネルギーでは $\ell = \|\mathbf{f}\|^2$ である。$\nabla_{\mathbf{x}}\ell$ をベクトル・ヤコビ行列積として表せ。このとき、随伴 ODE の 1 ステップには $\mathbf{f}$ に対するリバースモード計算が何回必要になるか。

3. [code] **軌跡の逆方向再構成。**●連続随伴では、$\mathbf{x}(t)$ を $\mathbf{x}(T)$ から $\mathbf{a}(t)$ とともに逆向きに積分することで、順方向の状態の保存を省くことができる。しかし 38.3.2 節で見たように、減衰率 λ で減衰するモードがあると、誤差 δ（保存された $\mathbf{x}(T)$ に含まれるもの）が $e^{\lambda T}\delta$ へと増幅される。

   1. $\mathbf{R}$ を 0.6 ラジアンの回転行列とし、$\mathbf{C} = \mathbf{R}\,\mathrm{diag}(-50, -1)\,\mathbf{R}^{\top}$ とおく。$\dot{\mathbf{x}} = \mathbf{C}\mathbf{x}$ を初期値 $(1, 1)$ から時刻 T まで 2000 ステップの RK4（`solve` と `rk4_step`）で積分し、続いて得られた結果から負の終端時刻を与えて同じ時間だけ逆向きに積分せよ。$T = 0.25$、0.5、1、2 に対する再構成誤差（reconstruction error）を求め、その増大を $e^{50T}\delta$ と比較せよ。どの程度の δ が当てはまるか。また、その値が 10^{-17} に近く、本文で述べた 10^{-16} より小さいのはなぜか。**ヒント：$\mathbf{x}(T)$ の各座標の大きさはどれほどか。**
   2. $\mathrm{diag}(-50, -1)$ そのものを用いて同じ実験を行え。$T = 0.5$ 以降、この行列では $\mathbf{C}$ の場合よりはるかに正確に初期値が再構成される。それはなぜか。
   3. 活性化チェックポイント（5.7.3.3 節）は軌跡の各区間の始点で状態を保存する。一方、ハイブリッド方式では各区間の終点のチェックポイントから逆向きに積分して区間内の状態を再構成する。(a) の行列 $\mathbf{C}$ に対し、ドリフト $e^{50T}\delta$（$\delta = 10^{-16}$）が 10^{-8} 未満に収まる最大の区間長はいくらか。

4. **残差ステップの対数行列式。**●●瞬間的な変数変換 Equation 38.21 は無限小ステップの極限である。有限幅の残差ステップ $F(\mathbf{x}) = \mathbf{x} + \mathbf{g}(\mathbf{x})$ では、Behrmann et al. (2019) の可逆残差ネットワークのように厳密な対数行列式が必要になる。

   1. $\mathbf{g}$ を定数 $c < 1$ のリプシッツ連続写像とする。$d = 1$ のときは 38.3.1.2 節の 1 行の議論（オイラーステップは $hL < 1$ のとき単調増加）から F が狭義単調増加、ゆえに単射であることがわかる。一般の次元 d に対し、$F(\mathbf{x}) = \mathbf{y}$ を $\mathbf{x}$ に関する不動点（fixed point）方程式として書き換え、ピカール・リンデレーフの定理の証明と同様にバナッハの不動点定理を用いて、F が $\mathbb{R}^d$ 上の全単射であることを示せ。この不動点反復はどのくらいの速さで収束するか。

   2. $\|\mathbf{J}_{\mathbf{g}}\| \le c$（ただし $\mathbf{J}_{\mathbf{g}}$ は $\mathbf{g}$ のヤコビ行列）を示し、$\mathbf{J}_{\mathbf{g}}$ の固有値を用いて次式を示せ。

$$\log\det(\mathbf{I} + \mathbf{J}_{\mathbf{g}}) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}\,\mathrm{tr}\,(\mathbf{J}_{\mathbf{g}}^{k}).$$

   3. オイラーステップでは $\mathbf{g} = h\mathbf{f}$ かつ $\mathbf{J}_{\mathbf{g}} = h\mathbf{J}$ である。$\log\det(\mathbf{I} + h\mathbf{J})$ を、1 ステップあたりに項 $h\,\mathrm{tr}\,\mathbf{J}$ を累積する Equation 38.22 と比較せよ。1 ステップあたりの差はどの程度で、T/h ステップ全体での合計の差はどの程度になるか。

   4. Hutchinson 推定量 Equation 38.24 は $\mathrm{tr}(\mathbf{J}_{\mathbf{g}}^{k})$ を $\epsilon^{\top}\mathbf{J}_{\mathbf{g}}^{k}\epsilon$ によって推定する。級数を第 K 項で打ち切るとき、プローブ 1 個あたり何回のベクトル・ヤコビ行列積が必要になるか。

5. [code] **Hutchinson 推定量の分散。**●推定量 Equation 38.24 は $\mathrm{tr}\,\mathbf{M}$ を $\epsilon^{\top}\mathbf{M}\epsilon$（単一のプローブ ϵ）によって推定する。$\mathbf{S} = (\mathbf{M} + \mathbf{M}^{\top})/2$ とおけば $\epsilon^{\top}\mathbf{M}\epsilon = \epsilon^{\top}\mathbf{S}\epsilon$ が成り立つ。

   1. $\epsilon^{\top}\mathbf{M}\epsilon$ の分散が、ガウスプローブでは $2\|\mathbf{S}\|_F^2$、ラデマッハプローブでは $2\sum_{i \ne j} S_{ij}^2$ となることを示せ。単一のラデマッハプローブが厳密な値を与えるのはどのような行列か。
   2. 引き戻し項を持つベクトル場のヤコビ行列 $\mathbf{J} = \mathbf{W}_2\,\mathrm{diag}(\mathbf{s})\,\mathbf{W}_1 - \mathbf{I}$（サイズは 50×50）を、Hutchinson の実験の重みと評価点から陽に構成せよ。両方の分散を計算し、その平方根を実験で出力されたプローブ 1 個あたりの 2 つの標準偏差と比較せよ。標準誤差を 0.01 にするには、それぞれの分布で何個のプローブが必要か。

3. CNF において、$\mathbf{M}$ はヤコビ行列 $\partial \mathbf{f}/\partial \mathbf{x}$（次元 d）である。すべての成分が同程度の大きさを持つと仮定する。プローブ 1 個の相対標準誤差 $\sqrt{\mathrm{Var}}/|\operatorname{tr}\mathbf{M}|$ は、対角成分がすべて同符号のときと符号がランダムなときで、それぞれ d に対してどのようにスケールするか。
4. 訓練時、ミニバッチ内の B 個の各点はそれぞれ独自の軌跡と独自の固定プローブを持つ。平均対数尤度のノイズは B にどう依存するか。また、どの B のときに、1 つの点が新規プローブ（$N = 100$ ステップ）で持つノイズと一致するか。

6. [code] **離散随伴と連続随伴。●●**ソルバを通した誤差逆伝播法は実行されたプログラムを微分するのに対し、随伴 ODE Equation 38.17 にソルバを適用したものは厳密なフローの勾配を近似する (Kidger 2022)。ヤコビ行列 $\mathbf{J}(\mathbf{x}) = \partial \mathbf{f}/\partial \mathbf{x}$ と順方向状態 $\mathbf{x}_n$ を持つ自律的なベクトル場について、2 つの漸化式を 1 ステップずつ比較せよ。

   1. 前進オイラー法に対し、離散随伴 Equation 38.19 は $\mathbf{a}_n = (\mathbf{I} + h\mathbf{J}(\mathbf{x}_n))^{\top}\mathbf{a}_{n+1}$ を計算する一方、随伴 ODE を t_{n+1} から t_n へ戻す陽的オイラーステップは $\mathbf{a}_n = \mathbf{a}_{n+1} + h\mathbf{J}(\mathbf{x}_{n+1})^{\top}\mathbf{a}_{n+1}$ を計算する。滑らかな $\mathbf{f}$ に対してこの 2 つの更新の差を上から評価せよ。線形なベクトル場では両者は一致し、そのときフローの勾配との差はオイラー法自身の誤差になると本文で述べた。スカラーのプログラム $x_N = (1+h\theta)^N x_0$ についてこれを厳密に確かめよ。すなわち、上の漸化式が本文で求めた勾配 $T\,x_N^2/(1+h\theta)$ を計算すること、およびこれが $T\,x(T)^2(1 - h\theta(1+\theta T)) + O(h^2)$ に等しいことを示せ。
   2. ホイン法の 1 ステップ $\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\big[\mathbf{f}(\mathbf{x}_n) + \mathbf{f}(\mathbf{u}_n)\big]$（ただし $\mathbf{u}_n = \mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n)$）を微分して、その離散随伴の漸化式を導け。この漸化式が、随伴 ODE を t_{n+1} から t_n へ進めるホイン法のステップにおいて $\mathbf{J}(\mathbf{x}_{n+1})$ を $\mathbf{J}(\mathbf{u}_n)$ に置き換えたものであることを示せ。2 つの更新は h の何次のオーダーで異なるか。
   3. 線形なベクトル場 $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ に対し、RK4 の 1 ステップは状態に $R(h\mathbf{A})$ を掛ける（ただし $R(z) = 1 + z + z^2/2 + z^3/6 + z^4/24$ は RK4 の安定性関数である。38.2 節、およびその虚軸上の振る舞いを扱う同節の演習問題 2 を参照）。RK4 の離散随伴が随伴 ODE の RK4 ステップに一致することを示せ。
   4. `loss_and_grad_unrolled` のオイラーステップを RK4 ステップに置き換え、手書きの逆方向スイープの代わりに中心有限差分で展開損失の勾配を計算せよ。連続随伴の勾配 `g_cont` との差を $N = 5$、10、20、40、80 ステップについて測定せよ。なお、`g_cont` を計算する台形公式は `n_quad = 2000` 区間を用いており約 10^{-7} の精度しかないため、先にこれを細かくしておくこと。どのような傾きが観測されるか。次数 p のソルバを通して計算された勾配は何次の精度を受け継ぐか。

7. [code] **リウヴィルの公式。●●**瞬間的な変数変換では軌跡に沿って密度を追跡した。本問では、フロー写像自身のヤコビ行列 $\mathbf{M}(t) = D\Phi_t(\mathbf{x}_0)$ を通じた別証明を与える。

   1. $\frac{d}{dt}\Phi_t(\mathbf{x}_0) = \mathbf{f}(\Phi_t(\mathbf{x}_0), t)$ を $\mathbf{x}_0$ で微分し、変分方程式 $\dot{\mathbf{M}} = \mathbf{J}(\mathbf{x}(t), t)\,\mathbf{M}$（初期条件 $\mathbf{M}(0) = \mathbf{I}$）を導け。
   2. ヤコビの公式 $\frac{d}{dt}\det\mathbf{M} = \operatorname{tr}(\mathbf{M}^{-1}\dot{\mathbf{M}})\det\mathbf{M}$ を可逆な $\mathbf{M}(t)$ に対して証明せよ。**ヒント：$\det\mathbf{M}(t+h) = \det\mathbf{M}(t)\,\det\big(\mathbf{I} + h\mathbf{M}^{-1}\dot{\mathbf{M}} + O(h^2)\big)$ と書き換え、本文の補題を用いる。**
   3. $\log|\det D\Phi_T(\mathbf{x}_0)| = \int_0^T \operatorname{tr}\mathbf{J}(\mathbf{x}(t), t)\,dt$ を結論付け、Equation 38.22 を離散の変数変換 Equation 38.20（全単射 Φ_T に適用したもの）から導け。線形なベクトル場に対して、この結果は公式 Equation 38.23 に一致する。
   4. $\dot{\mathbf{M}} = \mathbf{J}\mathbf{M}$ を 8 ユニットの tanh ベクトル場（38.3.3 節）と同時に RK4 で積分せよ（Hutchinson の実験で重みの変数名が再利用される前の、トレース積分を検証するセルで行うこと）。得られた $\log|\det\mathbf{M}(1)|$ を、そのセルが出力するトレース積分および対数行列式と比較せよ。

38.4 確率微分方程式

これまでの節で扱った軌跡はすべて始点によって定まり、フローはどこから出発したかを決して忘れない。すなわち、2 つの異なる始点は必ず 2 つの異なる終点に到達する。第 17 章の拡散モデル（diffusion model）は、始点を忘れる過程の上に構築されている。その**順過程**（forward process）は、始点の痕跡がまったく残らなくなるまで状態に新しいノイズを少しずつ混ぜていき、それを逆向きにたどる生成では既知の出発点から始められるようになる。このような過程も微分方程式を満たすが、通常の微分積分では扱えない方程式である。

まずは手計算で追えるほど小さな例から始めよう。1 つの数値 $x_0 = 2$ をとる。各ステップでその値の 99% を残し、標準偏差 $\sqrt{0.02} \approx 0.14$ の独立なガウス乱数を加える。

$$x_{k+1} = 0.99\, x_k + \sqrt{0.02}\, \xi_k, \qquad \xi_k \sim \mathcal{N}(0,1).$$

400 ステップ後には平均が $2 \cdot 0.99^{400} \approx 0.036$ となり、分散は 1 に近づく。ここでは 1 ステップあたりに削られる 2% と、新たに加わる 0.02 とが釣り合っている。他のどの始点から出発しても同じ場所に行き着き、ほぼ $\mathcal{N}(0,1)$ からのサンプルになる。この漸化式には一見奇妙な特徴がある。これがタイムステップ $h = 0.01$ で連続時間（continuous time）の過程を記述しているとするなら、減衰 $0.99 = 1 - h$ は x をオーダー h の量だけ変化させるのに対し、ノイズの大きさは $\sqrt{h}$ であって h ではない。この選択は必然である。大きさ h のノイズは $h \to 0$ とともに消えてしまうが、大きさ $\sqrt{h}$ のノイズは生き残り、その二乗はドリフトと同じオーダー h になる。通常の微分積分は微小変化の二乗を切り捨てるため、このたった 1 つの事実が新しい解析学を必要とするのである。

このようなノイズの極限がブラウン運動（Brownian motion）であり、1827 年に花粉から出た微粒子が水中で小刻みに揺れ動く様子を観察した植物学者 Robert Brown にちなんで名付けられた。Norbert Wiener は 1923 年にこれを数学的対象として定式化し、Kiyoshi Itô は 1940 年代に、たった 1 つの規則の上にその解析学を築き上げた (Itô 1944)。すなわち、短いブラウン増分の二乗を足し合わせると経過時間に一致するため、連鎖律（chain rule）に二階導関数（second derivative）の項が加わるという規則である。

本節では、コイントスからブラウン運動を構成し、上の規則からその解析学を構築する。続いて、冒頭の漸化式の背後にある過程、すなわち 17.2 節の順過程を解く。始点が与えられたとき、この過程の分布はあらゆる時刻でガウス分布になるため、拡散モデルは 1 回のサンプリングでデータを任意のノイズレベルまで汚すことができる。シミュレーションとその誤差の測定は、再読時に回してもよい。にわかには信じがたいのは二乗に関する規則だけであり、その証明は 6 行で済む。

38.4.1 ブラウン運動

38.4.1.1 順過程としてのノイズ

数学的な道具立てに入る前に、順過程が果たすべき役割を確認しておこう。$p_0 = p_{\text{data}}$ に従うデータから出発し、元のデータがどのような姿であっても、1 つの既知の分布に到達しなければならない。

$$p_0 = p_{\text{data}} \;\longrightarrow\; p_T \approx \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

そうすれば、生成は常にどこから出発すべきかがわかる。いかなる固定されたフローも、2 つのデータセットに対して同時にこれを達成することはできない。フローは可逆写像 Φ（38.1 節）であるため、もし 2 つの分布 p と q を同じ分布へ移すなら、Φ^{-1} を適用することで $p = q$ が導かれてしまうからである。

考えられる限り最も単純な 2 つの「データセット」、すなわち $x_0 = 2$ と $x_0 = -2$ における点質量に対して、3 つの過程を試してみよう。

- **ドリフトのみ**（$\dot{x} = -x$）。解 $x(t) = x_0 e^{-t}$ により、2 つの始点は $\pm 2e^{-3} = \pm 0.0996$（$t = 3$ のとき）に移る。両者はガウス分布ではなく点 0 に近づき、いかなる有限時刻においても依然として 2 つの異なる点のままである。
- **ノイズのみ**（$dX = \sqrt{2}\, dW$）。これは $X_t = x_0 + \sqrt{2}\, W_t$ を意味する。ここで W は時間とともに分散が線形に増大するノイズである（次項で構成する）。X_t の分布、すなわちその**法則**（law）は $\mathcal{N}(x_0, 2t)$ である。各点群の中心は ± 2 にとどまったまま、標準偏差 $\sqrt{2t}$ が際限なく増大する（値は 2.45、$t = 3$ のとき）。始点は広がり続ける幅との相対的な意味でしか忘れられず、極限の法則は存在しない。
- **両者の組み合わせ**（$dX = -X\, dt + \sqrt{2}\, dW$）。どちらの始点からでも $t = 3$ における法則は $\mathcal{N}(\pm 0.0996,\, 0.9975)$ になることが、38.4.3 節で証明される。2 つの法則は互いに、そして $\mathcal{N}(0, 1)$ ともほとんど区別がつかない。

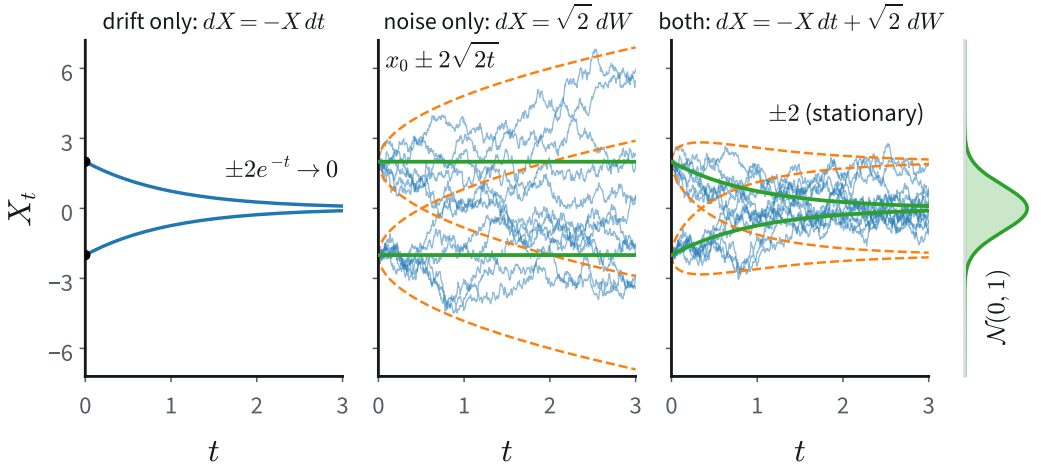


Figure 38.11: 2 つの始点 $x_0 = \pm 2$ から出発する、1 つの失敗例、1 つの利用可能な過程、そして本書で用いる過程。ドリフトのみではあらゆる始点が点 0 へ送られる。ノイズのみでは各始点が点群の中心に保たれたまま、標準偏差の 2 倍の包絡線 $x_0 \pm 2\sqrt{2t}$（破線）が際限なく広がる。両者を組み合わせると、あらゆる始点が同じ極限法則 $\mathcal{N}(0, 1)$（右端に横向きに描画）へと運ばれる。緑色の線は平均を表す。

Figure 38.11 にこれら 3 つを描く。完全に失敗するのは 1 つ目だけである。2 つ目は利用可能であり、実際に使われている。その分散 $2t$ がデータの分散よりはるかに大きくなれば、始点はほとんど影響しなくなる。この種の過程は極限法則を持たないため終端の分散を巨大にしなければならず、**分散爆発**（variance-exploding）と呼ばれる。3 つ目が本節を通じて用いる通し例であり、冒頭の漸化式はこれをシミュレーションしたものである。ドリフトが各始点をゼロへ引き寄せる一方、ドリフトが奪った分散をノイズが補填し、両者は分散 1 で釣り合う。

一般に、この種の順過程は決定論的なベクトル場にノイズ項を加えたものである。$\mathbb{R}^d$ 内のデータに対して、次のように書く。

$$d\mathbf{X} \;=\; \underbrace{\mathbf{f}(\mathbf{X}, t)\, dt}_{\text{drift}} \;+\; \underbrace{g(t)\, d\mathbf{W}}_{\text{diffusion}}, \tag{38.25}$$

この式は、微小時間 dt の間に、状態が速度 $\mathbf{f}$ と dt の積だけ移動し、ノイズ振幅 g で大きさが決まるランダムな揺らぎ（キック）を受けると読む。ドリフトが平均を操り、拡散（diffusion）が分散を注入する。これまでのところ、揺らぎ $d\mathbf{W}$ は単なる記号にすぎない。まずは 1 つの座標でこれに意味を与え、伊藤解析（Itô calculus）の最後に改めて $\mathbb{R}^d$ へ戻ることにする。

38.4.1.2 ランダムウォークからウィーナー過程へ

最も単純な確率過程（random process）である、コイントスで駆動されるランダムウォークの極限をとることで、この記号の背後に実体を与えよう。その極限から、冒頭の漸化式に平方根が現れた理由も明らかになる。

ステップ幅 Δt を固定し、公平なコインで符号を選びながら大きさ $\pm\sqrt{\Delta t}$ の独立なステップを踏むランダムウォークを考える。k ステップ後の時刻 $t = k\,\Delta t$ における位置は、k 個の独立な平均ゼロのステップ（それぞれの分散は Δt）の和である。独立な項の分散は加法性を持つため（36.2.2.1 節）、次が成り立つ。

$$\mathbb{E}\big[W_t^{(\Delta t)}\big] = 0, \qquad \mathrm{Var}\big(W_t^{(\Delta t)}\big) = k\,\Delta t = t,$$

これはステップサイズによらず成り立つ。たとえば $\Delta t = 0.01$ とすると、100 回の ± 0.1 のステップによって、分散は $100 \cdot 0.01 = 1$（時刻 1 時点）となる。同じ時間刻みで ± 0.01 のステップにすれば分散は 0.01 になり、± 1 のステップにすれば 100 になる。一般に、大きさ $c\,\Delta t^{a}$ のステップは分散 $c^2\,t\,\Delta t^{2a-1}$（時刻 t 時点）を与え、これは 0 へ収束するか（$a > \tfrac{1}{2}$ のとき）、あるいは ∞ へ発散する（$a < \tfrac{1}{2}$ のとき）。極限において有限かつ非ゼロの分散を残すのは平方根だけであり、後のすべての数値スキームのノイズ項に平方根が再び現れる理由はここにある。

$\Delta t \to 0$ とすると時刻 t までのステップ数は際限なく増え、中心極限定理（central limit theorem、36.3.2.3 節）によって位置はガウス分布に従うようになる。すなわち、全分散が t である多数の独立な平均ゼロの項の和は、近似的に $\mathcal{N}(0, t)$ となる。

この極限の過程は**ブラウン運動**または**ウィーナー過程**（Wiener process）と呼ばれる (Wiener 1923)。これは次の 4 つの性質によって定義される。

1. $W_0 = 0$;
2. **独立増分**（independent increments）：$s < t$ に対し、増分 $W_t - W_s$ は時刻 s までの経路全体と独立である。
3. **ガウス増分**（Gaussian increments）：$W_t - W_s \sim \mathcal{N}(0,\, t - s)$ が成り立つ。
4. 見本路（サンプルパス）が連続である。

これらの性質を持つ過程が存在することはウィーナーの定理（Wiener's theorem）であり、リスケールされたランダムウォークがそこへ収束することはドンスカーの定理（Donsker's theorem）である。本書では両者を既知のものとして認める。この過程は標準偏差 $\sqrt{t}$ でどこまでも広がり続け（Figure 38.12）、その増分は次の形をとる。

$$\Delta W \;=\; W_{t+\Delta t} - W_t \;=\; \sqrt{\Delta t}\,\xi, \qquad \xi \sim \mathcal{N}(0, 1), \tag{38.26}$$

すなわち、標準正規乱数を $\sqrt{\Delta t}$ 倍したものである。

独立増分の性質から、2 つの時刻における W の値同士の関係も決まる。$s \le t$ に対して $W_t = W_s + (W_t - W_s)$ と書く。増分は W_s と独立かつ平均ゼロであるため、$\mathrm{Cov}(W_s, W_t) = \mathbb{E}[W_s^2] = s = \min(s, t)$ となる。すなわち、このウォークは時刻 s における位置をそのまま先へ運び、それと無相関なノイズを加える。この公式には、1 行で終わる証明以上の価値がある。同時ガウス分布に従う値は平均と共分散によって定まるため、任意の有限個の時刻における値が同時

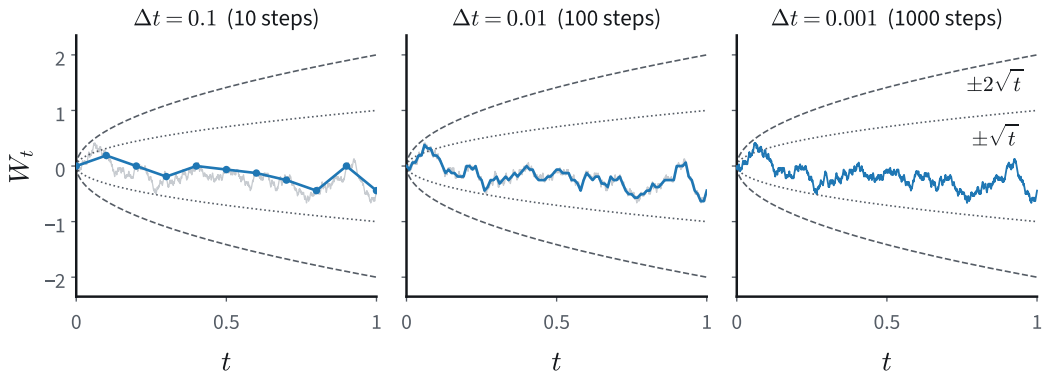


Figure 38.12: 1000 ステップ（ステップ幅 $\pm\sqrt{0.001}$）からなる 1 本のコイントス型ランダムウォークを、刻み幅 $\Delta t = 0.1$、0.01、0.001 のメッシュで観察した様子（薄い灰色は経路全体）。どの解像度で見てもステップ分散 Δt のランダムウォークになっており、メッシュを細かくすると大局的な経路を変えずに細部が埋められていく。各固定時刻における広がりは $\pm\sqrt{t}$ および $\pm 2\sqrt{t}$ の包絡線に従う。

ガウス分布に従い、平均ゼロ、共分散 $\min(s,t)$、かつ連続な経路を持つ過程はブラウン運動である。

同じ増分の性質が経路を粗く（ギザギザに）する。ブラウン経路は連続であるがいたるところ微分不可能であり、これは Paley、Wiener、Zygmund の定理である (Paley et al. 1933)。増分のスケーリング Equation 38.26 から、その直観的な理由がわかる。

$$\left|\frac{\Delta W}{\Delta t}\right| = \frac{|\xi|}{\sqrt{\Delta t}} \;\longrightarrow\; \infty \quad \text{in probability as } \Delta t \to 0,$$

したがって、ステップ幅が縮むにつれて差分商は際限なく大きくなる。これは厳密な証明ではないが（各固定時刻における導関数の存在を除外しているだけであり、「いたるところ」と言うにはすべての時刻を同時に扱う必要がある）、何が障壁になっているかは示している。

時間を c 倍、空間を $\sqrt{c}$ 倍にリスケールしても再びブラウン運動になるため（演習問題 2）、微分可能な曲線であれば拡大すると接線へと平坦化されるのに対し、ブラウン経路は拡大してもあらゆるスケールで同じ粗さを見せる。ゆえに dW/dt は存在せず、dW には積分の内部で意味を与えなければならない。その意味を与えるのが伊藤解析である。

38.4.1.3 ブラウン経路のシミュレーション

増分の表現 Equation 38.26 はそのままシミュレーションの手順にもなる。すなわち、$\sqrt{\Delta t}\,\xi$ のステップの累積和をとればよい。20,000 本の経路を区間 $[0,1]$ 上で 500 ステップにより生成し、1 つの問いを投げかけてみよう。任意の単一時刻では $W_t \sim \mathcal{N}(0,t)$ であるため、経路の 95.4% が $\pm 2\sqrt{t}$（Figure 38.12 の外側の包絡線）の内側に入る。では、すべての時刻にわたって包絡線の内側にとどまり続ける経路はどのくらいの割合だろうか。セルを実行する前に予想してみてほしい。

```
rng = np.random.default_rng()
T_bm, n_bm, n_paths = 1.0, 500, 20000
dt_bm = T_bm / n_bm
dW_bm = np.sqrt(dt_bm) * rng.standard_normal((n_paths, n_bm))
W_bm = np.cumsum(dW_bm, axis=1)              # W at t = dt, 2 dt, ..., 1
band = 2 * np.sqrt(dt_bm * np.arange(1, n_bm + 1))  # 2 sqrt(t) on the grid
inside = np.abs(W_bm) < band
print(f'inside +-2 sqrt(t) at t = 1:   {inside[:, -1].mean():.3f}'
      '   (Gaussian: 0.954)')
for stride in (10, 1):                       # same paths, coarse then fine grid
    print(f'inside at all {n_bm // stride} grid times: '
          f'{inside[:, stride - 1::stride].all(axis=1).mean():.3f}')
```

```
inside +-2 sqrt(t) at t = 1:   0.954   (Gaussian: 0.954)
inside at all 50 grid times: 0.697
inside at all 500 grid times: 0.535
```

単一の時刻 $t = 1$ における割合は、当然ながら約 95% である。しかし、等間隔な 50 個の時刻すべてで内側にとどまる経路は 10 本中約 7 本しかなく、500 個すべての時刻となると半分強にすぎない。各時刻で成り立つ帯域が、経路全体に対して成り立つわけではない。観測する時刻を増やすたびに、経路が外へ出ているのを見つける機会が増えるからである。あらゆる瞬間で監視すれば、内側にとどまる経路は 1 本もなくなる。$t = 0$ の近傍でブラウン経路は $\pm 2\sqrt{t}$ を何度も横切るからである（これは重複対数の法則（law of the iterated logarithm）と呼ばれるが、本書では証明しない）。ある 1 つの時刻における法則についての主張は、経路全体についての主張よりはるかに弱く、この 2 つは後にシミュレーションの弱誤差と強誤差として再び現れる（38.4.4.2 節）。

38.4.2 伊藤解析

38.4.2.1 二次変分

通常の微分積分でテイラー展開を 1 次で打ち切るのは、微小変化の二乗が無視できるほど小さいからである。ブラウン運動ではこれが成り立たず、どれほど成り立たないのかを示すのが本項の結果である。本節の残りの議論はすべてこの結果の上に成り立っている。

分割 $0 = t_0 < t_1 < \cdots < t_n = t$ をとり、区間幅を $\Delta t_i = t_{i+1} - t_i$、メッシュ幅を $h_{\max} = \max_i \Delta t_i$ として、この分割に沿った経路の増分の二乗を足し合わせる。微分可能な経路 x が $|x'| \le M$ を満たすとき、平均値の定理（mean value theorem）より $|\Delta x_i| \le M\,\Delta t_i$ となるため、次が成り立つ。

$$\sum_i (\Delta x_i)^2 \;\le\; M^2 \sum_i \Delta t_i^2 \;\le\; M^2\, h_{\max}\, t \;\longrightarrow\; 0$$

すなわち、メッシュ幅が縮むにつれてゼロへ収束する。滑らかな経路の増分の二乗はオーダー Δt^2 だからである。一方、ブラウン増分の二乗はオーダー Δt である。なぜなら $\mathbb{E}[(\Delta W_i)^2] = \Delta t_i$ であり、メッシュをどれほど細かくしてもこれらの和は t になるからである（Figure 38.13）。この和自体は確率変数だが、その揺らぎは極限において消失する。

命題（ブラウン運動の二次変分） メッシュ幅 $h_{\max} \to 0$ のとき、次が成り立つ。

$$Q_n \;=\; \sum_{i=0}^{n-1} \left(W_{t_{i+1}} - W_{t_i}\right)^2 \;\longrightarrow\; t \quad \text{in mean square:} \quad \mathbb{E}\big[(Q_n - t)^2\big] \to 0. \tag{38.27}$$

証明各増分は $\mathcal{N}(0, \Delta t_i)$ に従うため、任意の分割に対して $\mathbb{E}[(\Delta W_i)^2] = \Delta t_i$ および $\mathbb{E}[Q_n] = \sum_i \Delta t_i = t$ が成り立つ。ばらつきについては、ガウス分布の 4 次モーメントが $3\,\Delta t_i^2$ であるため $\operatorname{Var}\big((\Delta W_i)^2\big) = 3\Delta t_i^2 - \Delta t_i^2 = 2\Delta t_i^2$ となり、各増分は独立であるため分散は足し合わされる。

$$\mathbb{E}\big[(Q_n - t)^2\big] = \operatorname{Var}(Q_n) = \sum_{i} 2\,\Delta t_i^2 \;\le\; 2\,h_{\max} \sum_{i} \Delta t_i = 2\,h_{\max}\,t \;\longrightarrow\; 0,$$

ゆえに平均二乗収束（mean-square convergence）が従う。$\blacksquare$

この主張は改めて注目に値する。個々の増分はすべてランダムであるにもかかわらず、極限ではそれらの二乗和からランダム性が消えるのである。n 個の区間に等分割した $[0,1]$ の一様グリッド上では、上の証明よりこの和の標準偏差は $\sqrt{2/n}$ となり、その値は 0.35（$n = 16$ のとき）および 0.003（$n = 2^{18}$ のとき）である。

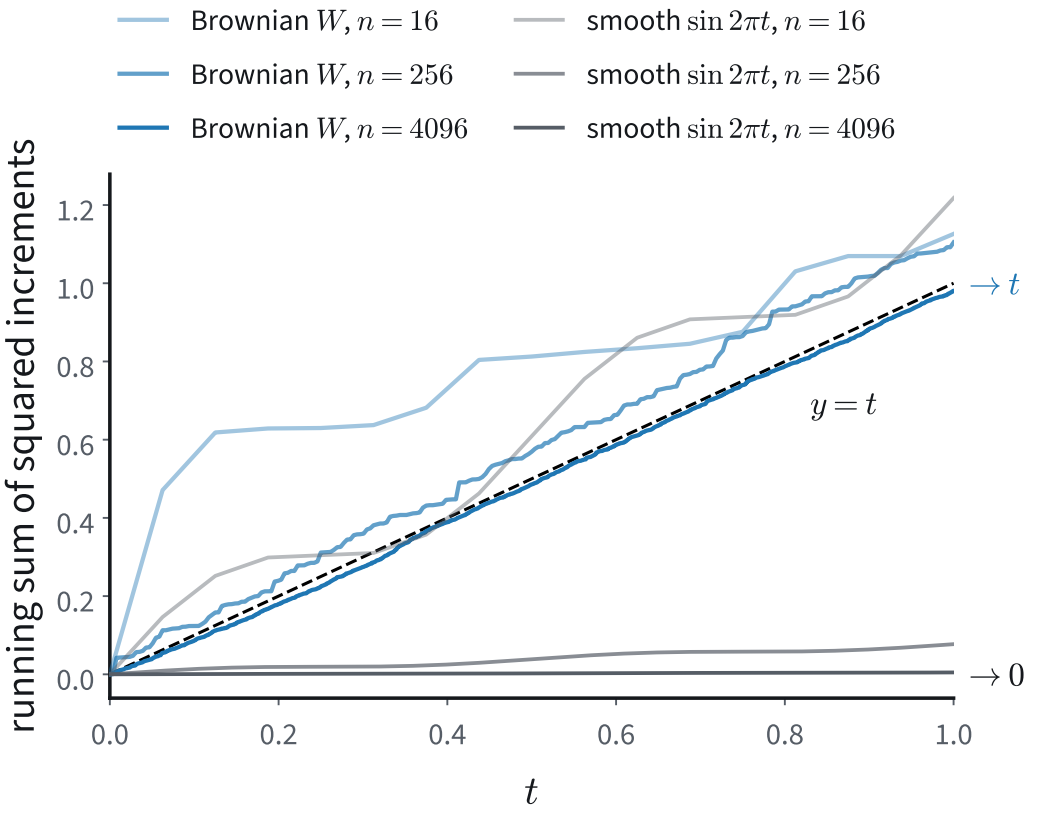


Figure 38.13: 1 本のブラウン経路に沿った増分の二乗の累積和（区間数 $n = 16$、256、4096 のメッシュ、色が濃いほど細かいメッシュを表す）。累積和は $y = t$ に近づく一方、滑らかな経路 $\sin 2\pi t$（灰色）に対する和は $2\pi^2/n$ のように縮小する。

この命題は通常、微分の積に関する実用規則である**伊藤の乗積表**（Itô multiplication table）として書かれる。

$$(dW)^2 = dt, \qquad dW\,dt = 0, \qquad (dt)^2 = 0. \tag{38.28}$$

1 つ目の規則は Equation 38.27 の略記であり、あくまで和に関する主張である。単一の増分の二乗は $\Delta t\,\xi^2$ であり、これが Δt に近い値をとることはめったにない。4 回に約 1 回は Δt の 10 分の 1 を下回り、6 回に約 1 回は Δt の 2 倍を上回る。証明が示したとおり、多数のステップにわたる和をとって初めて値が落ち着くのである。（コイントス型ウォークでは二乗したステップは毎回ちょうど Δt に等しいため、平均化を待つ必要がない。）残りの 2 つの規則は、$dW\,dt$（大きさ $\Delta t^{3/2}$）と $(dt)^2$ がいずれも Δt より速くゼロへ収束し、極限では何も寄与しないことを表している。

細かいメッシュでは揺らぎがきわめて小さくなるため、たった 1 本の経路から極限を確認できる。2^{18} 個の区間に分割した $[0,1]$ 上で 1 本の経路をサンプリングし、より粗い部分グリッド上で観察して、その二乗和を予測される標準偏差と並べて出力しよう。比較のため、滑らかな経路 $x(t) = \sin 2\pi t$ に対する同じ二乗和を $2\pi^2/n$（メッシュ幅 $1/n$ と $\int_0^1 x'(t)^2\,dt = 2\pi^2$ の積）と並べて出力する。

```
n_fine = 2**18
dW_qv = np.sqrt(1.0 / n_fine) * rng.standard_normal(n_fine)
W_qv = np.concatenate([[0.0], np.cumsum(dW_qv)])
print(f'{"n":>7} {"QV of W":>8} {"sqrt(2/n)":>10} {"QV of sin":>10}'
      f' {"2 pi^2/n":>9}')
for k in range(4, 19, 2):
    n = 2**k
    W_n = W_qv[::n_fine // n]                # the same path on a coarser grid
    s_n = np.sin(2 * np.pi * np.linspace(0, 1, n + 1))
    print(f'{n:7d} {np.sum(np.diff(W_n)**2):8.4f} {np.sqrt(2 / n):10.4f} '
          f'{np.sum(np.diff(s_n)**2):10.6f} {2 * np.pi**2 / n:9.6f}')
```

```
      n  QV of W  sqrt(2/n)  QV of sin  2 pi^2/n
     16   1.1551     0.3536   1.217927  1.233701
     64   0.9103     0.1768   0.308177  0.308425
    256   0.8957     0.0884   0.077102  0.077106
   1024   0.9560     0.0442   0.019277  0.019277
   4096   1.0044     0.0221   0.004819  0.004819
  16384   0.9893     0.0110   0.001205  0.001205
  65536   1.0007     0.0055   0.000301  0.000301
 262144   1.0017     0.0028   0.000075  0.000075
```

ブラウン運動の列は 1 のまわりに、証明が予測する標準偏差 $\sqrt{2/n}$ でばらつく。16 個の区間ではおおむね約 0.5 から 1.6 の間に収まり、$n = 2^{18}$ では約 0.01 以内の差で 1 に近づく。滑らかな経路の列はまだ 1.22 であるが（$n = 16$ のとき）、$2\pi^2/n$ に従って縮小し、3 桁までその予測と一致するようになる（$n = 256$ 以降）。

38.4.2.2 伊藤積分

経路が粗いため、dW に関する積分を直接定義せざるを得ない。最も単純な $\int_0^t W\,dW$ から始めると、通常のリーマン和（Riemann sum）による定義では答えが 1 つに定まらないことがわかる。

積分の基準となる関数が滑らかであれば、被積分関数を各区間の左端点、右端点、中点のいずれで評価しても、リーマン和はすべて同じ極限に収束する。しかしここではそうならない。W 自身を被積分関数とすると、右端点和は左端点和よりも次の量だけ大きくなる。

$$\sum_i W_{t_{i+1}}\,\Delta W_i - \sum_i W_{t_i}\,\Delta W_i = \sum_i \left(W_{t_{i+1}} - W_{t_i}\right)\Delta W_i = \sum_i (\Delta W_i)^2 \;\longrightarrow\; t,$$

ここで極限には先ほど証明した命題を用いた。「同じ」積分に対する 2 つの離散化（discretization）が、有限かつ決定論的な量だけ食い違うのである。1000 ステップからなる 1 本の経路で確かめてみよう。

```
n_lr = 1000
dt_lr = 1.0 / n_lr
t_lr = np.linspace(0, 1, n_lr + 1)
dW_lr = np.sqrt(dt_lr) * rng.standard_normal(n_lr)
W_lr = np.concatenate([[0.0], np.cumsum(dW_lr)])
left = np.sum(W_lr[:-1] * dW_lr)              # integrand at the left endpoint
right = np.sum(W_lr[1:] * dW_lr)              # integrand at the right endpoint
print(f'left sum {left:.4f}   right sum {right:.4f}')
print(f'right - left {right - left:.4f}   sum of (dW)^2 {np.sum(dW_lr**2):.4f}')
```

```
left sum -0.3614   right sum 0.6124
right - left 0.9738   sum of (dW)^2 0.9738
```

左端点和は実行するたびに変化するランダムな数値である。右端点和は、上の恒等式が示すとおり、表示されたすべての桁において増分の二乗和の分だけ左端点和を上回る。実行結果から新たにわかるのはその差の大きさである。この差は通常、約 0.1 以内の幅で 1 の近くに収まり（標準偏差は $\sqrt{2/1000} \approx 0.045$）、メッシュをさらに細かくしても埋まることはない。

両方の和は極限を持ち、ここまでの道具だけでそれを求められる。両者の和は望遠鏡和（中間の項が打ち消し合う和）になる。なぜなら、任意の分割上で $\left(W_{t_{i+1}} + W_{t_i}\right)\Delta W_i = W_{t_{i+1}}^2 - W_{t_i}^2$ を足し合わせると W_t^2 になり、さらに両者の差は t に収束するからである。したがって平均二乗の意味で、左端点和は $\tfrac{1}{2}\left(W_t^2 - t\right)$ に、右端点和は $\tfrac{1}{2}\left(W_t^2 + t\right)$ に収束する。通常の微分積分であれば $\tfrac{1}{2}W_t^2$ と答えるところだが、これはどちらの極限でもなく両者の平均である。評価点を 1 つ選ばなければならず、その選び方によって答えが変わる。2 つの和のうち、どちらを積分と定めるべきだろうか。

Itô の選択は左端点であり、これにより被積分関数はこれから起こる増分を決して見ることができない。公平な賭けを考えればその理由がわかる。ΔW_i を i 回目の公平な賭けの結果、H_i をそこに賭ける賭け金と考えよう。賭け手は賭けが決着する前の時刻 t_i までにわかっている情報だけから賭け金を決めなければならず、獲得賞金の合計は $\sum_i H_i\,\Delta W_i$ となる。増分 ΔW_i は、時刻 t_i 時点で既知のあらゆる情報（H_i を含む）と独立であるため、各回の期待獲得額は $\mathbb{E}[H_i\,\Delta W_i] = \mathbb{E}[H_i]\,\mathbb{E}[\Delta W_i] = 0$ となり、期待獲得額を正にする賭け戦略は存在しない。もし結果を盗み見て賭け金を決められるなら、平均して利益を出せてしまうだろう。右端点和はそのような賭け金を置くことに相当し、実際にその平均はゼロではなく t になる。

定義はこの規則をそのまま書き下したものである。各区間における値がその区間の増分を引く前に決まっているとき、被積分関数は**予測可能**（predictable）であるという。**単純予測可能過程**（simple predictable process）H とは、各区間で定数 $H_s = H_i$（$s \in (t_i, t_{i+1}]$）をとる過程であり、ここで H_i は時刻 t_i までの経路によって定まり、$\mathbb{E}[H_i^2] < \infty$ を満たすものとする。その**伊藤積分**（Itô integral）は獲得賞金の和として次で定義される。

$$\int_0^t H_s\,dW_s \;=\; \sum_i H_i\left(W_{t_{i+1}} - W_{t_i}\right). \tag{38.29}$$

公平な賭けの議論から、この積分の平均はゼロ、すなわち $\mathbb{E}\left[\int_0^t H\,dW\right] = 0$ である。さらに分散も計算できる。

命題（伊藤の等長性） 単純予測可能な H に対して、次が成り立つ。

$$\mathbb{E}\left[\left(\int_0^t H_s\,dW_s\right)^2\right] \;=\; \int_0^t \mathbb{E}\big[H_s^2\big]\,ds. \tag{38.30}$$

証明 Equation 38.29 の和の二乗を展開する。注意を要するのは交差項である。$i<j$ の交差項は因子 ΔW_j を含み、これは時刻 t_j 時点で既知のあらゆる量（積 $H_i\,\Delta W_i\,H_j$ を含む）と独立である。したがって期待値は積に分解でき、$\mathbb{E}\big[H_i\,\Delta W_i\,H_j\,\Delta W_j\big] = \mathbb{E}\big[H_i\,\Delta W_i\,H_j\big]\,\mathbb{E}[\Delta W_j] = 0$ となる。対角項も同様に $\mathbb{E}\big[H_i^2\big]\,\mathbb{E}\big[(\Delta W_i)^2\big] = \mathbb{E}\big[H_i^2\big]\,\Delta t_i$ へと分解される。対角項を足し合わせれば $\sum_i \mathbb{E}[H_i^2]\,\Delta t_i = \int_0^t \mathbb{E}[H_s^2]\,ds$ が得られる。$\blacksquare$

この等長性（isometry）によって、定義を階段状の過程から $H = W$ 自身のような被積分関数へも拡張できる。等長性は、写像 $H \mapsto \int H\,dW$ がノルム $\|H\|^2 = \int_0^t \mathbb{E}[H_s^2]\,ds$ を保つことを意味している。したがって、単純予測可能過程の列がこのノルムにおいて H に収束すれば、それらの積分も平均二乗収束し、その極限をもって $\int_0^t H\,dW$ の定義とする。平均ゼロの性質と等長性は極限でも保たれ、有限なノルムを持つすべての予測可能な H に対して成り立つ。また、この種の連続な被積分関数に対する左端点和はその積分に収束する。これらの主張の証明は省略する（Øksendal (2003) の第 3 章を参照）。

これにより、出発点とした積分の値が左端点和の極限として定まる。

$$\int_0^t W_s\,dW_s = \tfrac{1}{2}\left(W_t^2 - t\right). \tag{38.31}$$

これは通常の微分積分の答えから $t/2$ を引いたものである。この $-t/2$ があることで、公平な賭けが持つべき平均ゼロが実現される（$\mathbb{E}[W_t^2] = t$ であるため）。この積分に対して等長性を確かめることもできる。等長性はノイズ積分の分散を二次モーメント（second moment）の通常の積分へと変換する。ここでは $\int_0^1 \mathbb{E}[W_s^2]\,ds = \int_0^1 s\,ds = \tfrac{1}{2}$（$t=1$ のとき）となり、閉形式の結果とも一致する。実際、W_1^2 は標準正規確率変数の二乗であって分散 2 を持つため、$\mathrm{Var}\big(\tfrac{1}{2}(W_1^2-1)\big) = \tfrac{1}{4}\cdot 2 = \tfrac{1}{2}$ となる。

他の評価点を選べば別の積分が定義される。中点の選択、あるいは左端点和と右端点和の平均は、**ストラトノビッチ積分**（Stratonovich integral）を定義する (Stratonovich 1966)。上の平均 $\tfrac{1}{2}W_t^2$ が示唆するとおり、この積分は通常の連鎖律に従うが、被積分関数がこれから起こる増分の半分を見てしまうため、平均ゼロの性質を失う。本書では 2 つの理由から伊藤の流儀を採用する。第 1 に平均ゼロであるため期待値の計算が容易になること、第 2 にステップの始点で係数を評価する最も単純なシミュレーションのステップが伊藤和になることである（演習問題 6）。

38.4.2.3 ドリフト、拡散、解

積分が手に入ったことで、順過程のテンプレート Equation 38.25 に数学的な意味が備わる。**確率微分方程式**（stochastic differential equation、SDE）は次のように書かれる。

$$dX = f(X,t)\,dt + g(X,t)\,dW, \tag{38.32}$$

この微分形は次の積分方程式の略記である。

$$X_t = X_0 + \int_0^t f(X_s, s)\, ds + \int_0^t g(X_s, s)\, dW_s,$$

すなわち、通常の積分と伊藤積分の和である。短いステップにおいて係数をその始点の値に固定すると、$\mathbb{E}[dX] = f\, dt$ および $\mathrm{Var}(dX) = g^2\, dt$ となる。したがって**ドリフト** f は条件付き平均速度であり、**拡散**係数（diffusion coefficient）g は分散が注入される速度 g^2 を定める。$g \equiv 0$ のときこの方程式は 38.1 節の ODE になり、$f \equiv 0$ かつ定数 g のときは $X_t = X_0 + g\, W_t$、すなわち Figure 38.11 のノイズのみの過程になる。

では、解とは何だろうか。ODE では 1 本の曲線であったが、ここでは各ブラウン経路に対して 1 本の曲線が対応する。f と g が x に関してリプシッツ連続で高々線形にしか増大せず（$|f(x,t)| + |g(x,t)| \le C(1 + |x|)$）、かつ X_0 が W と独立で $\mathbb{E}[X_0^2] < \infty$ を満たすならば、始点とブラウン経路によって解の経路が一意に定まる（Øksendal (2003)、定理 5.2.1）。その証明は 38.1.1.3 節の Picard 反復を確率版に置き換えたものであり、積分の評価の代わりに等長性を用いる。したがって、ランダム性はすべて W の中にある。同じ始点から毎回新しいノイズを用いて方程式を何度も走らせれば、Figure 38.11 のような経路の群れが得られ、生成モデル（generative model）が関心を持つのは各時刻におけるその点群の法則である。

シミュレーションを行う際には 1 つの区別が重要になる。振幅が時間にのみ依存する場合（$g = g(t)$）、ノイズは**加法的**（additive）であるという。テンプレート Equation 38.25 や、通し例を含む本章のすべての拡散モデル SDE がこれに該当する。一方、水準に比例して変動する株価のように、振幅が状態に依存する場合（$g = g(X, t)$）、ノイズは**乗法的**（multiplicative）であるという。この違いは表面的に見えるかもしれないが、シミュレーションされた経路が真の経路をどれだけ正確に追跡できるかを決定づける（38.4.4.2 節）。

38.4.2.4 伊藤の補題

SDE を使って計算を進めるには連鎖律が必要になる。すなわち、X が SDE の解であるとき、関数 $\phi(X_t, t)$ はどの方程式を満たすのだろうか。通常の連鎖律を当てはめると $d(W^2) = 2W\, dW$ となるが、これが誤りであることはすでにわかっている。$\int_0^t 2W\, dW = W_t^2 - t$ が Equation 38.31 より成り立つからである。何かが欠けているのだ。

ランダムウォークの 1 ステップを見れば、欠けている項がどこから生じるのか、また他の関数に対してどのような形になるのかがわかる。$\phi(x) = x^2$ とし、大きさ $\pm\sqrt{\Delta t}$ の 1 ステップを点 x から各符号とも確率 $\frac{1}{2}$ で踏み出すとする。ステップ後の ϕ の平均は次のようになる。

$$\tfrac{1}{2}\left[(x + \sqrt{\Delta t})^2 + (x - \sqrt{\Delta t})^2\right] = x^2 + \Delta t = \phi(x) + \tfrac{1}{2}\phi''(x)\, \Delta t,$$

ここで $\phi'' = 2$ を用いた。この増加分はステップの二乗から生じており、コイントスではそれが毎回 Δt になるのに対し、通常の連鎖律はそれを捨ててしまうのである。

$x = 0.7$、$\Delta t = 0.25$ とすると、1 ステップ後の到達点は 0.2 または 1.2 となり、そこでの ϕ の値は 0.04 または 1.44 である。その平均 0.74 は $\phi(0.7) = 0.49$ よりも $0.25 = \Delta t$ だけ上にある（Figure 38.14）。凸関数（convex function）は対称なノイズから正の利得を受け取り、ブラウン増分の二乗はオーダー dt であるため、単位時間あたりの利得 $\frac{1}{2}\phi''$ は極限でも消えない。

一般の公式も同じ計算から導かれる。X を $dX = f\, dt + g\, dW$ の解とし、$\phi(x, t)$ を滑らかな関数とする。1 次までではもはや不十分であるため 2 次まで展開すると、ϕ の微小変化は次のようになる。

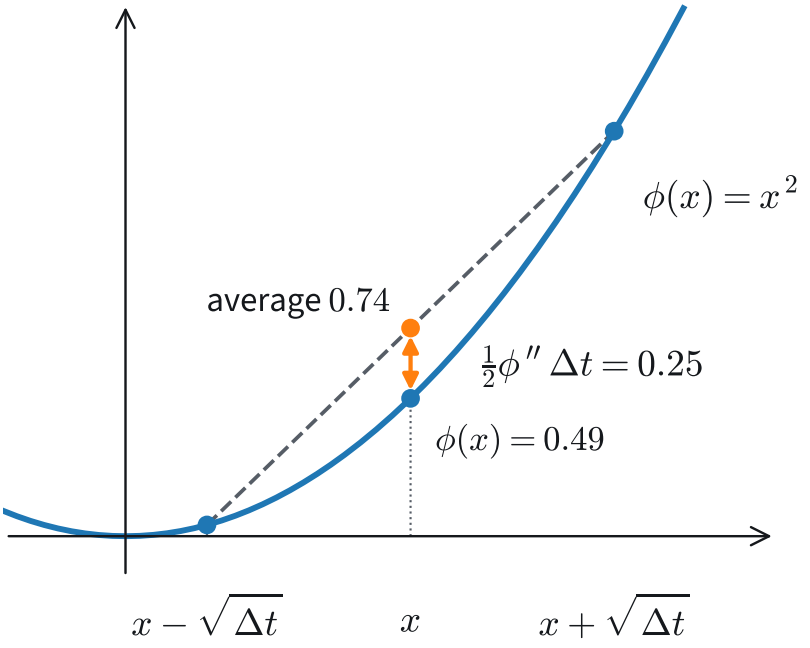


Figure 38.14: イェンセンのギャップとしての伊藤補正。$\pm\sqrt{\Delta t}$ の 1 ステップを x から踏み出すと、凸な ϕ の平均は $\phi(x)$ よりも $\frac{1}{2}\phi''(x)\,\Delta t$ だけ上に移る。ここでは $\phi = x^2$、$x = 0.7$、$\Delta t = 0.25$ であり、ギャップは 0.25 である。

$$d\phi = \phi_t\,dt + \phi_x\,dX + \tfrac{1}{2}\phi_{xx}\,(dX)^2 + \cdots .$$

$dX = f\,dt + g\,dW$ を代入し、$(dX)^2$ を乗積表 Equation 38.28 に従って展開する。

$$(dX)^2 = f^2(dt)^2 + 2fg\,dt\,dW + g^2(dW)^2 = g^2\,dt.$$

ドリフト項と交差項は消え、ノイズ項が 1 次のオーダーで生き残る。項をまとめれば確率解析の連鎖律が得られる。ここまでの議論は発見的な導出であるため、結果は証明なしの定理として述べる。

定理（伊藤の補題） 前項の条件のもとで X を $dX = f\,dt + g\,dW$ の解とし、$\phi \in C^{1,2}$、すなわち時間に関して 1 回、状態に関して 2 回連続微分可能であるとする。このとき $Y_t = \phi(X_t, t)$ は次を満たす。

$$d\phi \;=\; \bigl(\phi_t + f\,\phi_x + \tfrac{1}{2}g^2\,\phi_{xx}\bigr)\,dt \;+\; g\,\phi_x\,dW. \tag{38.33}$$

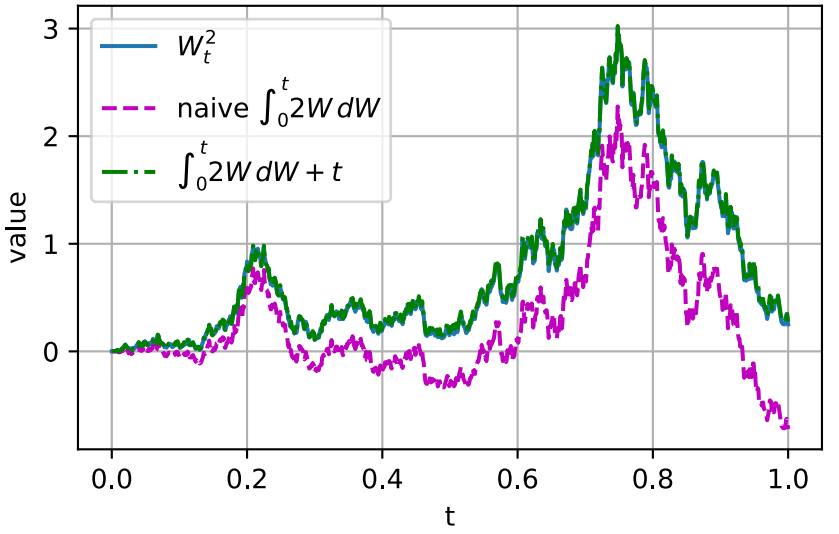
これがこの分野における実用上の基本公式であり、通常の連鎖律に、二階導関数の半分と分散が到着する速度との積を加えた形になっている。新しく加わった項 $\frac{1}{2}g^2\phi_{xx}\,dt$ は**伊藤補正**（Itô correction）と呼ばれる。証明ではテイラー展開の剰余項を平均二乗の意味で抑える必要があり（Øksendal (2003) の第 4 章）、状態に関する一階・二階の両方の導関数が不可欠である。したがって、この補題は $\phi(x) = |x|$ のような折れ目を持つ関数にはそのままでは適用できない。

例 1：W^2。 $X = W$（したがって $f = 0$、$g = 1$）および $\phi(x) = x^2$ とする。補題より $d(W^2) = 2W\,dW + dt$ となり、通常の連鎖律が落としていた dt が現れる。積分形で書けば $W_t^2 = \int_0^t 2W_s\,dW_s + t$ であり、これは再び Equation 38.31 に一致する。補題のたった 1 行が望遠鏡和の計算に取って代わったのである。

補正項が蓄積していく様子は実際に観察できる。任意のグリッド上で、W_t^2 と $\int_0^t 2W\,dW$ の左端点和との差は望遠鏡和 $W_t^2 - \sum_{t_i<t} 2W_{t_i}\Delta W_i = \sum_{t_i<t}(\Delta W_i)^2$ になるため、これは累積の二次変分（quadratic variation）に一致し、直線 t に沿って推移するはずである。先ほどの左端点和・右端点和で用いた経路を再利用しよう。

```
ito = np.concatenate([[0.0], np.cumsum(2 * W_lr[:-1] * dW_lr)])  # int 2W dW
gap = W_lr**2 - ito
print(f'final gap W_1^2 - int 2W dW = {gap[-1]:.4f}   (Ito correction: t = 1)')
print(f'max |gap_t - t| along the path = {np.abs(gap - t_lr).max():.4f}')
d2l.plot(t_lr, [W_lr**2, ito, ito + t_lr], 't', 'value',
         legend=['$W_t^2$', 'naive $\\int_0^t 2W\\,dW$',
                 '$\\int_0^t 2W\\,dW + t$'], figsize=(5, 3))
```

```
final gap W_1^2 - int 2W dW = 0.9738   (Ito correction: t = 1)
max |gap_t - t| along the path = 0.0364
```



naive とラベル付けされた曲線は伊藤積分 $\int_0^t 2W\,dW$ であり、通常の連鎖律が W_t^2 とみなしてしまう量である。この曲線は W_t^2 の下に垂れ下がるが、t を加えると元の曲線にぴったり重なる。経路全体にわたって差は通常、約 0.1 以内の幅で t の近くに収まる。Equation 38.27 の証明より、時刻 t におけるその標準偏差はこのグリッド上で $\sqrt{2t\,\Delta t} \le 0.045$ だからである。$t = 1$ における差は、前のセルで出力した増分の二乗和に一致する。

例 2：幾何ブラウン運動（geometric Brownian motion）。SDE $dX = \mu X\,dt + \nu X\,dW$ $(X_0 > 0)$ は、平均成長率 μ で増大し、自身の大きさに比例するノイズを伴う量をモデル化する。これは Samuelson による株価モデルであり (Samuelson 1965)、Black と Scholes のオプション価格公式の基礎となっている (Black and Scholes 1973)。そのノイズは乗法的である。$\phi(x) = \log x$ に補題を適用しよう。このとき $\phi' = 1/x$、$\phi'' = -1/x^2$、$g^2 = \nu^2 x^2$ である。対数関数は $x > 0$ でのみ滑らかであるため、ひとまず X が正にとどまると仮定する。

$$d\log X = \left(\mu - \tfrac{1}{2}\nu^2\right)dt + \nu\,dW, \qquad \text{so} \qquad X_t = X_0 \exp\!\left(\left(\mu - \tfrac{1}{2}\nu^2\right)t + \nu W_t\right). \tag{38.34}$$

得られた式は正の値をとり、これを W_t と t の関数とみなして補題を適用すれば元の方程式を満たすことが確かめられるため、先ほどの仮定は外せる。$-\frac{1}{2}\nu^2$ は $\phi'' < 0$ に対する補正であり、凹関数（concave function）は対称なノイズによって損をする。そのためノイズは典型的な成長率 $\mu - \frac{1}{2}\nu^2$ を平均成長率 μ よりも押し下げ、演習問題 3 では X_t の平均と中央値（median）がどれほど乖離するかを示す。後の項では、この厳密解を基準にしてシミュレーション誤差を測定する。

多次元でも構造は同じであり、結果は既知のものとして認める (Øksendal 2003)。$d\mathbf{X} = \mathbf{f}\,dt + \mathbf{G}\,d\mathbf{W}$ において、$\mathbf{X} \in \mathbb{R}^d$、$\mathbf{G}$ を行列、$\mathbf{W}$ を各座標が独立なスカラーのブラウン運動であるブラウン運動とすると、乗積表は $dW_i\,dW_j = \delta_{ij}\,dt$ となり、2 回連続微分可能な ϕ は次を満たす。

$$d\phi(\mathbf{X}) \;=\; \nabla\phi \cdot d\mathbf{X} \;+\; \tfrac{1}{2}\sum_{i,j} \left(\mathbf{G}\mathbf{G}^{\top}\right)_{ij} \frac{\partial^2 \phi}{\partial x_i\,\partial x_j}\,dt. \tag{38.35}$$

補正項は、各二階偏導関数をノイズの共分散速度 $\mathbf{G}\mathbf{G}^{\top}$ で重み付けする。本章の拡散モデル SDE は

すべての座標を対等に扱うため $\mathbf{G} = g(t)\,\mathbf{I}$ であり、このとき補正項は $\frac{1}{2}g(t)^2 \sum_i \partial^2\phi/\partial x_i^2 \; dt$、すなわち ϕ のラプラシアンの定数倍へと帰着する。次節ではこの形の補題が必要になる。

38.4.3 オルンシュタイン・ウーレンベック過程

ここで通し例に名前を与え、その解を求めよう。オルンシュタイン・ウーレンベック（OU）過程 (Uhlenbeck and Ornstein 1930) とは次の方程式で定まる過程である。

$$dX = -\theta X\,dt + g\,dW, \qquad \theta > 0, \tag{38.36}$$

すなわち、減衰 ODE $\dot{x} = -\theta x$（38.1.2 節）に定数振幅 g のノイズを加えたものである。考案者たちは、分子からの衝突で弾かれつつ摩擦によって減速する粒子の速度を表すためにこの式を書き下した。ドリフトはゼロから外れた変位をレート θ でゼロへと引き戻し、ノイズは絶えず新たな変位を生み出し続ける（Figure 38.15）。冒頭の漸化式は、$\theta = 1$、$g = \sqrt{2}$ としたこの過程を、各ステップでドリフトを固定しながらステップ幅 $h = 0.01$ のグリッド上で時間発展させたものである。1 ステップごとに $1 - \theta h = 0.99$ を掛け、標準偏差 $g\sqrt{h} = \sqrt{0.02}$ のノイズを加えている。

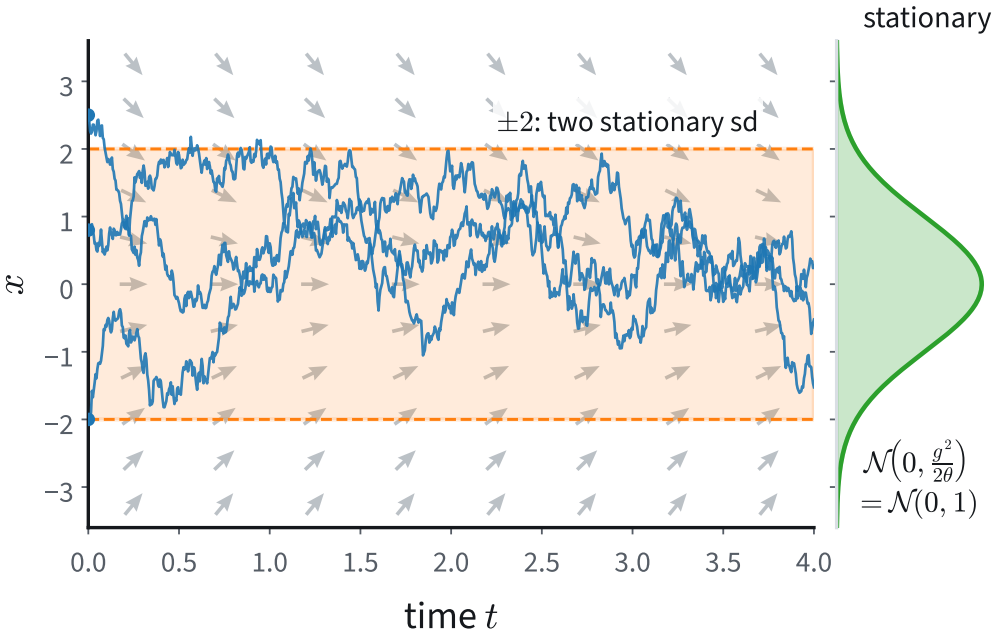


Figure 38.15: 平均回帰。ドリフト $-\theta x$（矢印）があらゆる経路を 0 へ向けてレート $\theta = 1$ で引き戻す一方、振幅 $g = \sqrt{2}$ のノイズが経路を動かし続ける。3 つの始点からの経路は $1/\theta$ の数倍の時間のうちに始点を忘れ、その後は極限法則の標準偏差の 2 倍の帯域 $\pm 2\sqrt{g^2/2\theta} = \pm 2$（stationary と表記）の内側で揺れ動く。右はその密度である。

38.4.3.1 遷移カーネル

OU 過程は、あらゆる時刻における法則が閉形式でわかる数少ない SDE の 1 つであり、拡散モデルがこの過程の上に構築される理由もそこにある。X_t の条件付き法則（始点 $X_0 = x_0$ が与えら

れたとき）を始点の関数として読んだものを**遷移カーネル**（transition kernel）と呼ぶ（畳み込みのカーネルとは無関係である）。OU 過程ではこれはガウス分布になる。本節のすべての結果の中で、本書の以降の章で最も頻繁に用いるのがこの結果である。

命題（OU 遷移カーネル） OU 過程 Equation 38.36 に対して、次が成り立つ。

$$X_t \mid X_0 = x_0 \;\sim\; \mathcal{N}\!\left(x_0\, e^{-\theta t},\; \frac{g^2}{2\theta}\left(1 - e^{-2\theta t}\right) \right). \tag{38.37}$$

証明積分因子 $e^{\theta t}$ は 38.1.2.3 節において入力を伴う減衰方程式を（定数変化法の公式により）解いたが、入力がノイズであっても同様に機能する。伊藤の補題 Equation 38.33 を $\phi(x,t) = e^{\theta t} x$ に適用する。このとき $\phi_t = \theta e^{\theta t} x$、$\phi_x = e^{\theta t}$、$\phi_{xx} = 0$ であるため、補正項はゼロになる。

$$d(e^{\theta t} X_t) = \left(\theta e^{\theta t} X_t - \theta e^{\theta t} X_t\right) dt + e^{\theta t} g\, dW_t = g\, e^{\theta t}\, dW_t .$$

ドリフトが打ち消し合い、0 から t まで積分して $e^{-\theta t}$ を掛けると次式が得られる。

$$X_t = X_0\, e^{-\theta t} + g \int_0^t e^{-\theta (t-s)}\, dW_s . \tag{38.38}$$

残った積分の被積分関数は決定論的であるため、独立なガウス増分の重み付き和の極限となり、平均ゼロのガウス分布に従う（ガウス確率変数の平均二乗極限がガウス分布になることは認める）。その分散は伊藤の等長性 Equation 38.30 から求まる。

$$\operatorname{Var}(X_t \mid X_0) = g^2 \int_0^t e^{-2\theta (t-s)}\, ds = \frac{g^2}{2\theta}\left(1 - e^{-2\theta t}\right). \qquad \blacksquare$$

解 Equation 38.38 は薄れゆく記憶として解釈できる。始点はレート θ で減衰し、時刻 s に加わったノイズの揺らぎは重み $e^{-\theta(t-s)}$ を伴って生き残る。つまり直近のノイズは効き、古いノイズは忘れ去られる。漸化式では、k ステップ前の揺らぎが重み $0.99^k \approx e^{-0.01k}$ で残る。これ以降の各節、および本編の 17.2 節では、平均係数と標準偏差という 2 つの係数を用いてこのカーネルを次のように書く。

$$\alpha_t = e^{-\theta t}, \qquad \sigma_t^2 = \frac{g^2}{2\theta}\left(1 - e^{-2\theta t}\right), \qquad \text{so that} \qquad X_t \mid X_0 = x_0 \sim \mathcal{N}\!\left(\alpha_t x_0,\, \sigma_t^2\right). \tag{38.39}$$

データは平均 $\alpha_t x_0$ を通じてのみ入り、ノイズは分散 σ_t^2 を通じてのみ入る。通し例（$\theta = 1$、$g = \sqrt{2}$、$\sigma_t^2 = 1 - e^{-2t}$）を $x_0 = 2$ から出発させると、カーネルは $\mathcal{N}(1.213,\, 0.795^2)$（$t = 0.5$）、$\mathcal{N}(0.736,\, 0.930^2)$（$t = 1$）、そして $\mathcal{N}(0.037,\, 1.000^2)$（$t = 4$）となる。あらゆる時刻・あらゆる始点に対してカーネルがガウス分布であるため、始点を条件とするノイズ付き状態については、対数密度（log-density）の勾配 $-(x - \alpha_t x_0)/\sigma_t^2$ を含め、すべてが閉形式でわかる。生成はこの勾配に依存することが後に明らかになる。

漸化式はこの過程の離散化にすぎないため、その点群はカーネルに近く従うはずだが、厳密に一致するとは限らない。10,000 個の $x_0 = 2$ のコピーに対して単純なループで漸化式を実行し、5 つの時刻における点群を保存して、その平均と標準偏差を $2\alpha_t$ および σ_t と並べて出力し、あわせて $2\alpha_t \pm 2\sigma_t$ の内側に入る割合も出力しよう。

```
theta_ou, g_ou, h_ou = 1.0, np.sqrt(2.0), 0.01
X_ou = np.full(10000, 2.0)                    # 10,000 copies of x_0 = 2
snap_steps = {round(t / h_ou): t for t in (0.5, 1.0, 1.5, 2.0, 4.0)}
X_ou_snap = {}
for k in range(1, 401):
    xi = rng.standard_normal(X_ou.size)
    X_ou = (1 - theta_ou * h_ou) * X_ou + g_ou * np.sqrt(h_ou) * xi
    if k in snap_steps:
        X_ou_snap[snap_steps[k]] = X_ou.copy()
print(f'{"t":>4} {"mean":>6} {"2alpha_t":>8} {"std":>6} {"sigma_t":>7}'
      f' {"inside":>7}')
for t, X in X_ou_snap.items():
    alpha_t = np.exp(-theta_ou * t)
    sigma_t = np.sqrt(g_ou**2 / (2 * theta_ou) * (1 - alpha_t**2))
    inside = np.mean(np.abs(X - 2 * alpha_t) < 2 * sigma_t)
    print(f'{t:4.1f} {X.mean():6.3f} {2 * alpha_t:8.3f} {X.std():6.3f} '
          f'{sigma_t:7.3f} {inside:7.3f}')
```

```
   t   mean 2alpha_t    std sigma_t  inside
 0.5  1.209    1.213  0.803   0.795   0.951
 1.0  0.733    0.736  0.943   0.930   0.955
 1.5  0.433    0.446  0.968   0.975   0.957
 2.0  0.259    0.271  0.992   0.991   0.953
 4.0  0.042    0.037  1.012   1.000   0.952
```

平均と標準偏差は、標準誤差（standard error）の 2〜3 倍の範囲内でカーネルと一致する（平均では約 0.01、標準偏差では 0.007 が $10{,}000$ サンプルに対する標準誤差である）。ガウス分布と同様に、どの時刻でも点群の約 95% が帯域の内側に入っている。単一の点 2 として出発した点群は素早く広がった後、それ以上は広がらなくなる。このサンプル数では、離散化の影響はまだ見て取れない。

38.4.3.2 定常分布

順過程は始点を忘れなければならず、カーネル Equation 38.37 は OU 過程が実際にそうであることを示している。$t \to \infty$ のとき平均 $x_0 e^{-\theta t}$ はゼロへと減衰し、分散は $g^2/(2\theta)$ に近づくため、次が成り立つ。

$$X_t \mid X_0 = x_0 \;\longrightarrow\; \mathcal{N}\!\left(0,\, \frac{g^2}{2\theta}\right) \quad \text{in distribution, for every } x_0.$$

これこそが 38.4.1.1 節で求めた、あらゆるデータに共通の終着点である。この極限は単なる終着点にとどまらない。この分布から出発した過程はそのままそこにとどまり続け、この性質を持つ法則を**定常**（stationary）と呼ぶ。これを確かめるため、$X_0 \sim \mathcal{N}(0, g^2/2\theta)$ を W と独立にとろう。すると Equation 38.38 より、X_t は $X_0 e^{-\theta t}$ と分散 σ_t^2 の独立なガウス確率変数との和になるため、平均ゼロのガウス分布に従い、その分散は次のようになる。

$$e^{-2\theta t}\,\frac{g^2}{2\theta} + \frac{g^2}{2\theta}\left(1 - e^{-2\theta t}\right) = \frac{g^2}{2\theta}.$$

始点の減衰によって失われる分散と、ノイズによって加わる分散とがちょうど釣り合っている。では、始点を忘れるにはどれほどの時間がかかるのか。その時間スケールは**緩和時間**（relaxation

time）と呼ばれる $1/\theta$ であり、その数倍の時間が経てば $e^{-\theta t}$ は無視できるほど小さくなる。通し例では $g^2/2\theta = 1$ かつ $1/\theta = 1$ であるため、定常分布（stationary distribution）は $\mathcal{N}(0,1)$ であり、$t = 4$ までに平均は 2 から 0.037 へと低下していた。

漸化式もそれ自身の定常分布を持つが、わずかに異なる値になる。1 ステップは分散 v を $(1-\theta h)^2 v + g^2 h$ へと写す。$v = 1$ から通し例の 1 ステップを進めると $0.9801 + 0.02 = 1.0001$ と 1 をわずかに上回り、この写像の不動点（fixed point）は次のようになる。

$$v_\infty = \frac{g^2 h}{1-(1-\theta h)^2} = \frac{0.02}{0.0199} = 1.005 \qquad (h = 0.01).$$

したがって漸化式は、厳密な分散 1 よりも 0.5% 高い値に落ち着く。10,000 個の点群では 0.5% のずれを見分けられないが、手計算なら見抜ける。これが離散化誤差の最初の兆候であり、38.4.4.2 節ではこの誤差が h に比例して縮小することを示す。

38.4.3.3 分散保持の正規化

定常分散 $g^2/2\theta$ は自由パラメータであり、拡散モデルでは次のように選んでこれを 1 に固定する。

$$g^2 = 2\theta \qquad \Longrightarrow \qquad X_t \mid X_0 = x_0 \;\longrightarrow\; \mathcal{N}(0,1).$$

通し例でもこの選択をとっている。このとき $\sigma_t^2 = 1 - \alpha_t^2$ となるため、ただ 1 つの係数でカーネルを記述できる。

これまでは始点を単一の数値 x_0 としていたが、今度はデータからのサンプル X_0 としよう。すると分布の意味で、解 Equation 38.38 は $X_t = \alpha_t X_0 + \sqrt{1-\alpha_t^2}\,\epsilon$（$\epsilon$ は X_0 と独立な標準ガウス確率変数）と表される。データ全体にわたってカーネルを平均したこの X_t の法則を、時刻 t における**周辺分布**（marginal）と呼ぶ。多くの論文ではこれを別の文字で表記する。すなわち、Ho et al. (2020) によるノイズ除去拡散確率モデル（denoising diffusion probabilistic model、DDPM）、つまり 17.2 節の離散時間拡散モデルに倣い、ここでの $\alpha_t^2 = e^{-2\theta t}$ を $\bar{\alpha}_t$ と書くのである。本編でも同じ記法を用いており、ここでもそれに従う。

$$X_t \;=\; \sqrt{\bar{\alpha}_t}\, X_0 \;+\; \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0,1) \text{ independent of } X_0. \tag{38.40}$$

この周辺分布は、二乗和が 1 になる重みを用いたデータと新しいノイズの重み付き混合である。データの平均がゼロ、分散が 1 であるとしよう。独立な 2 項の分散をとると次式が得られる。

$$\mathrm{Var}(X_t) = \bar{\alpha}_t \mathrm{Var}(X_0) + (1-\bar{\alpha}_t) = \bar{\alpha}_t + (1-\bar{\alpha}_t) = 1 \qquad \text{for all } t.$$

これが**分散保持**（variance-preserving、VP）の意味である。単位分散のデータに対しては、極限だけでなくあらゆる時刻において周辺分散が 1 に等しくなる。この過程は、データ側が担う割合 $\bar{\alpha}_t$ とすでにノイズへ置き換わった割合 $1-\bar{\alpha}_t$ との間で合計を変えずにデータからノイズへと分散を移していく。もっとも、この名称が文字どおり当てはまるのは単位分散のデータに限られる。分散 s^2 のデータでは周辺分散が s^2 から 1 へと変化するため（演習問題 5）、データのスケールが重要になる。17.1 節のように $[-1,1]$ へスケールした画像の分散は 1 未満である一方、潜在拡散（latent diffusion）モデルでは潜在変数をほぼ単位分散になるようリスケールする (Rombach et al. 2022)。

分散は保たれるが、分布の形状は保たれない。コイントスの結果のように、± 1 をそれぞれ確率 $\frac{1}{2}$ でとるデータを考えよう。その 4 次モーメントは 1 であるのに対し標準ガウス分布では 3 であり、過程が進むにつれて $\mathbb{E}[X_t^4] = 3 - 2\bar{\alpha}_t^2$ に従って上昇し、通し例では 2.264（$t = 0.25$）、さらに 2.963（$t = 1$）に達する（同じ演習問題を参照）。分散が微動だにしないまま、コインの分布は釣鐘型の曲線へと姿を変えるのである。

拡散モデルでは通常、レートを時間とともに変化させる。スコアベース拡散モデルの **VP-SDE** (Y. Song, Sohl-Dickstein, et al. 2021) は次式で与えられる。

$$dX = -\tfrac{1}{2}\beta(t)\, X\, dt + \sqrt{\beta(t)}\, dW, \tag{38.41}$$

これは時間依存のレート $\theta(t) = \beta(t)/2$ とノイズ振幅 $g(t)^2 = \beta(t)$ を持つ OU 過程であり、通し例は定数スケジュール $\beta = 2$ にあたる。あらゆる瞬間で $g^2 = 2\theta$ が保たれるため、スケジュール全体にわたって分散は 1 にとどまり、その周辺分布は Equation 38.40 において $\bar{\alpha}_t = e^{-\int_0^t \beta(s)\, ds}$ としたものになる（演習問題 7 でその遷移が厳密にガウス分布になることを証明する）。これが 17.2 節の分散保持経路、すなわち拡散モデルが逆転させることを学習する順過程である。

38.4.4 シミュレーション

38.4.4.1 オイラー・丸山法

ほとんどの SDE は閉形式解（closed-form solution）を持たないため、シミュレーションを行う必要がある。その方法は冒頭の漸化式がすでに示している。Equation 38.32 をグリッド $t_n = n\,\Delta t$ 上でシミュレーションするには、ODE に対する前進オイラー法（38.2 節）と同様に各ステップで係数を固定し、増分の表現 Equation 38.26 が定める大きさのガウスノイズを加える。**オイラー・丸山法**（Euler–Maruyama）の更新式 (Maruyama 1955) は次のとおりである。

$$\hat{X}_{n+1} = \hat{X}_n + f(\hat{X}_n, t_n)\,\Delta t + g(\hat{X}_n, t_n)\,\sqrt{\Delta t}\,\xi_n, \qquad \xi_n \sim \mathcal{N}(0,1)\ \text{i.i.d.}, \tag{38.42}$$

ここで $\hat{X}_n$ は X_{t_n} の近似値であり、$\hat{X}_0 = X_0$ とする。ノイズの揺らぎは $\sqrt{\Delta t}$ に比例してスケールする。他のいかなる指数でも極限でノイズが消失するか爆発してしまうからであり（38.4.1.2 節）、また $g = 0$ とすればこのスキームは前進オイラー法に一致する。

OU 過程に対して、1 ステップは次の形になる。

$$\hat{X}_{n+1} = (1 - \theta\,\Delta t)\,\hat{X}_n + g\sqrt{\Delta t}\,\xi_n = 0.99\,\hat{X}_n + 0.1414\,\xi_n, \tag{38.43}$$

2 番目の式は $\Delta t = h = 0.01$ における通し例であり、冒頭からずっと動かしてきた漸化式そのものである。

この過程に限っては、もっと良いやり方も可能であった。$g^2 = 2\theta$ のとき、1 区間にわたるカーネル Equation 38.37 は $\mathcal{N}(e^{-\theta\Delta t}x,\, 1 - e^{-2\theta\Delta t})$ であり、そこからのサンプリングは DDPM の順方向ステップ（17.2 節）、すなわち $x_k = \sqrt{1-\beta_k}\,x_{k-1} + \sqrt{\beta_k}\,\epsilon_k$（ただし $\beta_k = 1 - e^{-2\theta\Delta t}$）に一致する。したがって DDPM のステップは任意のステップサイズで厳密であり、オイラー・丸山法はその 1 次近似にあたる。$\theta = 1$、$\Delta t = 0.01$ のとき、厳密なステップは $e^{-0.01} = 0.99005$ を掛けて分散 $\beta_k = 1 - e^{-0.02} = 0.0198$ のノイズを加えるのに対し、近似では 0.99 と 0.02 であった。2 種類のベータを混同してはならない。ステップの β_k は 1 ステップでノイズに置き換わる分散の割合であり、$\beta(t)\,\Delta t$（Equation 38.41 のレート $\beta(t)$ を用いた量）におよそ等しく、こ

こでは 0.0198 対 $2 \cdot 0.01$ となっている。厳密なステップが使えるのは線形方程式に限られる（演習問題 7）。一般のドリフトには閉形式の遷移法則が存在しないため、近似スキームに頼るほかない。

以下の補助関数は、経路のバッチ全体に対して Equation 38.42 を実装したものである。増分 ΔW を引数として受け取るため、2 つのシミュレーションをまったく同じノイズで駆動できる。この機能はすぐに必要になる。ノイズが入った状態では「誤差」そのものを定義し直す必要があり、その定義には 2 種類あるからである。

```
def euler_maruyama(f, g, x0, t, dW):
    """Simulate dX = f dt + g dW on the grid t for a batch of paths."""
    X = np.empty((len(x0), len(t)))
    X[:, 0] = x0
    for k in range(len(t) - 1):
        h = t[k + 1] - t[k]
        X[:, k + 1] = (X[:, k] + f(X[:, k], t[k]) * h
                       + g(X[:, k], t[k]) * dW[:, k])
    return X
```

38.4.4.2 強収束と弱収束

シミュレーションされた経路はランダムであり、真の経路も同様である。2 つの誤差の定義は、何を比較するかという点で異なる。すなわち経路を比較するか法則を比較するかであり、これは包絡線の帯域実験で区別した 2 種類の主張に対応している。

ステップ数を $N = T/\Delta t$ とし、$\hat{X}_N$ が X_T を近似するものとする。数値スキームと厳密解を同じブラウン増分で駆動しよう。**強誤差**（strong error）$\mathbb{E}\,|\hat{X}_N - X_T|$ は経路ごとの追従性を測る。すなわち、シミュレーションされた軌跡が、同じノイズで駆動された真の軌跡をどれだけ忠実にたどるかを見る。一方、滑らかなテスト関数に対する**弱誤差**（weak error）$|\mathbb{E}\,\varphi(\hat{X}_N) - \mathbb{E}\,\varphi(X_T)|$（テスト関数は φ）は、時刻 T における法則が正しく再現されているかだけを測る。生成モデルが使うのは終端時刻のサンプルだけであるため、必要なのは弱い意味での精度である。強い意味での精度が重要になるのは、マルチレベルモンテカルロ法（multilevel Monte Carlo）のように異なる解像度の経路同士を結合する場合である。この手法は推定の大半を安価な粗い経路で行い、高価な細かい経路を少数だけ使ってそれを補正する（演習問題 8）。Figure 38.16 にこの 2 つの概念を並べて示す。

スキームが強収束次数（strong order）p を持つとは、その強誤差が高々 $C\,\Delta t^{p}$ であることをいい、弱収束次数（weak order）p も同様に定義される。オイラー・丸山法の次数は古典的な結果として知られている。

定理（オイラー・丸山法の収束次数、証明略） **f、g、φ を滑らかな関数とし、増大条件および X_0 のモーメント条件（厳密な定式化は Kloeden and Platen (1992) を参照）を満たすとする。**

1. **オイラー・丸山法は強収束次数 $\tfrac{1}{2}$ を持つ：** $\mathbb{E}\,|\hat{X}_N - X_T| \le C\,\Delta t^{1/2}$。
2. **ノイズが加法的（$g = g(t)$）であれば、強収束次数は 1 になる：** $\mathbb{E}\,|\hat{X}_N - X_T| \le C\,\Delta t$。
3. **加法的ノイズ・乗法的ノイズのいずれに対しても、弱収束次数は 1 である：** $|\mathbb{E}\,\varphi(\hat{X}_N) - \mathbb{E}\,\varphi(X_T)| \le C\,\Delta t$。

1 ステップあたりの誤差を見積もれば、これらの指数の理由がわかる。もちろんこれは厳密な証明ではない。各ステップで生じた誤差がその場にとどまるかのように足し合わせているだけであり、実際の証明では初期に生じた誤差がその後のダイナミクスによってどう伝播するかも制御しなければならない。

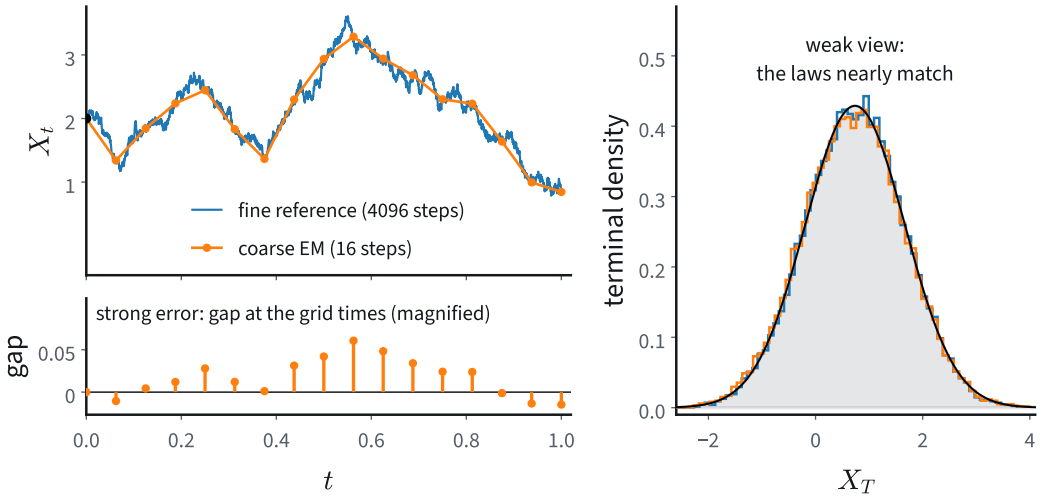


Figure 38.16: $X_0 = 2$ から出発する $[0, 1]$ 上の通し例に対する、オイラー・丸山法の強収束と弱収束。左：同じブラウン増分で駆動された 1 本の粗い経路（16 ステップ、オレンジ色）と細かい参照経路（4096 ステップ、青色）。下の帯は粗いグリッド時刻における両者の差を拡大したもので、強誤差が測るのはこの量である。右：20,000 本の経路に対する 2 つのステップサイズでの終端ヒストグラムは互いに、また厳密な法則（黒色）ともほぼ一致する（ただし 16 ステップでは分散が約 4% 大きすぎる）。

オイラー・丸山法はステップの間 g を固定するが、実際には g も経路とともに変化する。ステップ内部の時刻 s までに経路はおおよそ $g\,(W_s - W_{t_n})$ だけ動いているため、そこでのノイズ係数（noise coefficient）はおおよそ $g + g\,\partial_x g\,(W_s - W_{t_n})$ になる。このスキームが取りこぼす最大の項は第 2 項を dW で積分したものであり、これは Equation 38.31 をブラウン運動 $W_s - W_{t_n}$（時刻 t_n から新たに始まるもの）に適用した式にほかならない。

$$g\,\partial_x g \int_{t_n}^{t_{n+1}} \left(W_s - W_{t_n}\right) dW_s = \tfrac{1}{2}\, g\,\partial_x g\,\left((\Delta W)^2 - \Delta t\right).$$

ミルスタイン法（Milstein's scheme）はこの項を補い戻す (Milstein 1975)。そこに含まれるランダムな因子は、二乗に関する規則を 1 ステップずつ見たものに相当する。すなわち $(\Delta W)^2 = \Delta t$ は和をとれば成り立つが、個々のステップでは平均ゼロ・大きさ Δt のずれを伴う。$N = T/\Delta t$ 回のステップからこの種の無相関な項が積み重なり、ランダムウォークと同様に足し合わされて $\sqrt{N}\,\Delta t = \sqrt{T\,\Delta t}$ となる。これが次数 $\frac{1}{2}$ の正体である。

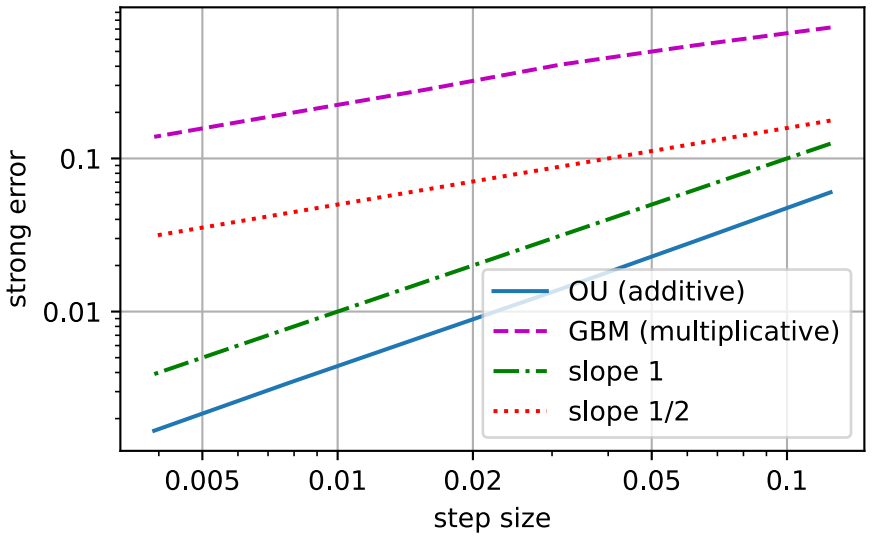
加法的ノイズでは $\partial_x g = 0$ となるため、この項自体が消える。その次に省略される確率的な項は 1 ステップあたり大きさ $\Delta t^{3/2}$ を持ち、足し合わせると $\sqrt{N}\,\Delta t^{3/2} = O(\Delta t)$ になり、決定論的な剰余項 $N\,\Delta t^2 = O(\Delta t)$ と同じオーダーとなって次数は 1 になる。一方、弱誤差で問題になるのは平均だけである。省略された確率的な項はすべて平均ゼロであり、1 ステップの統計的バイアスはオーダー Δt^2 であるため、ノイズの種類によらず N ステップ全体では $O(\Delta t)$ となる。

同じノイズのもとで両方の強収束次数を測定しよう。幾何ブラウン運動（$\mu = \nu = 1$、$X_0 = 1.5$）はその厳密解 Equation 38.34 と比較し、$X_0 = 2$ からの通し例は 4096 ステップのグリッド上の参照解と比較する。後者の厳密解 Equation 38.38 に含まれるノイズ積分自体が、グリッド上でし

か評価できないためである。粗い増分は連続する細かい増分の和として作るため、どの解像度も同じブラウン経路を見ることになる。

```
T, n_ref, n_em = 1.0, 2**12, 2000
mu_g, nu_g = 1.0, 1.0                       # GBM: multiplicative noise
ou_f, ou_g = lambda x, s: -theta_ou * x, lambda x, s: g_ou
gbm_f, gbm_g = lambda x, s: mu_g * x, lambda x, s: nu_g * x
x0_ou, x0_gbm = np.full(n_em, 2.0), np.full(n_em, 1.5)
dW_f = np.sqrt(T / n_ref) * rng.standard_normal((n_em, n_ref))
t_f = np.linspace(0, T, n_ref + 1)
ou_ref = euler_maruyama(ou_f, ou_g, x0_ou, t_f, dW_f)[:, -1]
gbm_exact = 1.5 * np.exp((mu_g - nu_g**2 / 2) * T + nu_g * dW_f.sum(axis=1))
dts, errs_ou, errs_gbm = [], [], []
for n in [2**k for k in range(3, 9)]:        # 8 to 256 steps
    dW = dW_f.reshape(n_em, n, n_ref // n).sum(axis=2)
    t = np.linspace(0, T, n + 1)
    ou = euler_maruyama(ou_f, ou_g, x0_ou, t, dW)[:, -1]
    gbm = euler_maruyama(gbm_f, gbm_g, x0_gbm, t, dW)[:, -1]
    dts.append(T / n)
    errs_ou.append(np.abs(ou - ou_ref).mean())
    errs_gbm.append(np.abs(gbm - gbm_exact).mean())
print('strong-order slopes:  OU (additive) '
      f'{np.polyfit(np.log(dts), np.log(errs_ou), 1)[0]:.2f},'
      '  GBM (multiplicative) '
      f'{np.polyfit(np.log(dts), np.log(errs_gbm), 1)[0]:.2f}')
d2l.plot(dts, [errs_ou, errs_gbm, np.array(dts), 0.5 * np.sqrt(dts)],
         'step size', 'strong error', xscale='log', yscale='log',
         legend=['OU (additive)', 'GBM (multiplicative)',
                 'slope 1', 'slope 1/2'], figsize=(5, 3))
```

```
strong-order slopes:  OU (additive) 1.03,  GBM (multiplicative) 0.48
```



測定された傾きは、加法的ノイズの OU 過程では 1 に近く、乗法的ノイズの幾何ブラウン運動では $\frac{1}{2}$ に近く、定理の 2 つの強収束次数と一致する。2 つの過程はスケールが異なるため曲線の高

さ自体は異なるが、比較すべきはその下がり方である。ステップ幅を半分にするたびに、OU 過程の誤差は半分になり、幾何ブラウン運動の誤差は約 $\sqrt{2}$ 分の 1 になる。

弱収束次数の確認にはサンプリングすら不要である。OU 過程に適用したオイラー・丸山法は線形ガウス漸化式になるため、$\hat{X}_n$ がガウス分布に従えば $\hat{X}_{n+1}$ もガウス分布に従う。したがってシミュレーションの周辺分布はあらゆるステップでガウス分布となり、その平均と分散は次の漸化式に従う。

$$m_{n+1} = (1 - \theta\,\Delta t)\, m_n, \qquad v_{n+1} = (1 - \theta\,\Delta t)^2\, v_n + g^2\,\Delta t.$$

1 ステップごとに、平均には $1 - \theta\Delta t$ が掛かるのに対し、厳密解では $e^{-\theta\Delta t}$ が掛かる（0.900 対 0.905、$\Delta t = 0.1$ のとき）。分散の漸化式は等比級数の和 $v_n = g^2\Delta t\,(1 - a^{2n})/(1 - a^2)$（ただし $a = 1 - \theta\Delta t$、$v_0 = 0$）となり、その極限は先ほど求めた不動点 v_∞ である。$T = 1$、初期値 $X_0 = 2$ として 6 通りのステップ数でこれらの閉形式を評価すれば、傾きを曖昧にする標本誤差を一切伴わずにシミュレーション法則の誤差が求まる。

```
mean_T, var_T = 2 * np.exp(-1), 1 - np.exp(-2)  # exact OU kernel at T = 1
print(f'{"n":>4} {"EM mean":>8} {"EM var":>7}'
      f'   exact {mean_T:.4f}, {var_T:.4f}')
dts, errs = [], []
for n in [10, 20, 40, 80, 160, 320]:
    h = 1 / n
    a_h = 1 - theta_ou * h                  # one-step mean multiplier
    m = 2 * a_h**n                          # EM mean after n steps
    v = g_ou**2 * h * (1 - a_h**(2 * n)) / (1 - a_h**2)   # geometric series
    print(f'{n:4d} {m:8.4f} {v:7.4f}')
    dts.append(h)
    errs.append([abs(m - mean_T), abs(v - var_T)])
s_m, s_v = np.polyfit(np.log(dts), np.log(np.array(errs)), 1)[0]
print(f'weak-order slopes:  mean {s_m:.2f},  variance {s_v:.2f}')
```

```
   n  EM mean  EM var   exact 0.7358, 0.8647
  10   0.6974  0.9247
  20   0.7170  0.8938
  40   0.7265  0.8791
  80   0.7311  0.8718
 160   0.7335  0.8682
 320   0.7346  0.8664
weak-order slopes:  mean 1.01,  variance 1.01
```

10 ステップではシミュレーション法則の平均が 0.697、分散が 0.925 となるのに対し、厳密値は 0.736 と 0.865（先ほどのカーネル $\mathcal{N}(0.736,\, 0.930^2)$）であり、320 ステップでは 0.735 と 0.866 になる。当てはめた傾きはどちらも 1.01 である。すなわち、ステップ幅を半分にするとシミュレーション法則の誤差も半分になり、弱収束次数 1 が確かめられる。

幾何ブラウン運動に対して同じ計算を行っても弱収束次数 1 が得られるため、乗法的ノイズは経路の追従では次数を半分失わせるが、法則の再現では何も失わせない。一方で、一見すると高精度に見えるスキームが誤った方程式へ収束してしまうこともある。ステップの両端でノイズ係数を平均するホイン法（Heun's method）は、伊藤解ではなくストラトノビッチ解へと収束してしまう（演習問題 6）。

シミュレーションは順過程のサンプルを生成し、カーネルは単一の始点からの法則を与える。しかし生成にはそれ以上のものが必要になる。データは単一の点ではない。ノイズ付きデータの周

辺分布 p_t はすべての始点にわたってカーネルを平均したものであり、サンプルを通じてしか知り得ない実データに対しては閉形式を持たない。それにもかかわらず、過程を逆向きに走らせるには、単一の始点に対してなら書き下せた $\log p_t$ の勾配が必要になる。そして今のところ、私たちは p_t が従う方程式さえ手にしていないのである。

38.4.5 まとめ

順過程は始点を忘れなければならないが、いかなる決定論的フローにもそれはできないため、ノイズによって駆動される。微小ステップの極限で生き残るノイズの大きさは $\sqrt{\Delta t}$ であり、その増分の二乗を足し合わせると経過時間に一致する（$(dW)^2 = dt$）。そのため dW に関する積分は被積分関数をどこで評価するかに依存し、Itô が選んだ左端点評価によって平均ゼロと等長性 $\mathbb{E}[(\int H\,dW)^2] = \int \mathbb{E}[H^2]\,ds$ が成り立つ。同じ理由から、連鎖律には補正項 $\frac{1}{2} g^2 \phi_{xx}\,dt$ が加わる。これらの道具により、オルンシュタイン・ウーレンベック過程 $dX = -\theta X\,dt + g\,dW$ が解かれる。始点 x_0 が与えられたとき、その分布はあらゆる時刻で平均 $x_0 e^{-\theta t}$、分散 $\frac{g^2}{2\theta}(1 - e^{-2\theta t})$ のガウス分布となり、$g^2 = 2\theta$ と選べば拡散モデルの分散保持順過程になる。閉形式解を持たない方程式は、オイラー法のステップに大きさ $\sqrt{\Delta t}$ のノイズを加えたオイラー・丸山法によってシミュレーションされ、経路を加法的ノイズでは次数 1、乗法的ノイズでは次数 $\frac{1}{2}$ で追跡し、法則をその両方に対して次数 1 で再現する。

38.4.6 演習問題

1. **無限の全変動。** 経路 x の分割 $0 = t_0 < t_1 < \cdots < t_n = t$ に沿った全変動は $V_n = \sum_{i} |x(t_{i+1}) - x(t_i)|$ で定義される。経路ごとの積分 $\int H\,dx$ は、和 $\sum_i H(s_i)\,\Delta x_i$（評価点 $s_i \in [t_i, t_{i+1}]$ は任意）の極限として定義され、すべての連続な H に対して、メッシュ幅が縮むときに V_n が有界にとどまるならば存在する。

    1. $x = W$ とし、$\Delta t_i = t/n$ の一様分割を用いよ。$\mathbb{E}[V_n]$ と $\mathrm{Var}(V_n)$ を計算し（$\mathbb{E}|\xi| = \sqrt{2/\pi}$、ただし $\xi \sim \mathcal{N}(0,1)$ を用いよ）、$V_n \to \infty$ が $n \to \infty$ のとき確率収束の意味で成り立つことを導け。
    2. x を連続とし、任意の分割に対して $V_n \le V$ を満たすとする。メッシュ幅が縮むにつれて $\max_i |\Delta x_i| \to 0$ となることを示し、$\sum_i (\Delta x_i)^2 \le \max_i |\Delta x_i| \cdot V \to 0$、すなわちこのような経路の二次変分がゼロになることを導け。このとき、なぜ Equation 38.27 からブラウン経路の全変動が非有界であることの 2 つ目の証明が得られるのか。
    3. x を有界な全変動を持つ連続経路とする。左端点和 $\sum_i x(t_i)\,\Delta x_i$ と右端点和 $\sum_i x(t_{i+1})\,\Delta x_i$ がともに $\frac{1}{2}\left(x(t)^2 - x(0)^2\right)$ へ収束することを示せ。$x = W$ に対する 38.4.2.2 節の 2 つの極限と比較せよ。ブラウン経路では上の議論のどのステップが破綻するか。

2. [code] **ブラウンスケーリングと到達時刻。**

    1. $c > 0$ に対し $\tilde{W}_t = W_{ct}/\sqrt{c}$ とおく。38.4.1.2 節の性質 1〜4 を確かめ、$\tilde{W}$ がブラウン運動であることを導け。これは $\pm\sqrt{\Delta t}$ ウォークのどの性質を厳密にしたものか。
    2. τ_a を $|W_t|$ が初めて水準 $a > 0$ に達する時刻とする。(a) を用いて τ_a が $a^2 \tau_1$ と同じ分布を持つこと、したがって $\mathbb{E}[\tau_a] = a^2\,\mathbb{E}[\tau_1]$ が成り立つことを示せ。
    3. $\mathbb{E}[\tau_1] = 1$ を認めたうえで（これは $\mathbb{E}[W_\tau^2] = \mathbb{E}[\tau]$ から従い、さらにそれは Equation 38.31 と停止時刻までの伊藤積分の平均がゼロであることから導かれる）、$4{,}000$ 本のランダムウォークを分散 $\Delta t = 10^{-3}$ のガウスステップでシミュレーションし、$\mathbb{E}[\tau_a]$ を $a = 0.5, 1, 2$ に対してモンテカルロ標準誤差とともに推定せよ。推定値はわずかに a^2 を上回る傾向があり、特に小さい a で顕著である。離散ウォークが水準を

横切る仕組みからこの統計的バイアスの符号を説明し、それが a と Δt に対してどのようにスケールするかを述べよ。

4. 本文の実験では、半分強の経路が $\pm 2\sqrt{t}$ の内側に 500 個すべてのグリッド時刻でとどまった。(a) を用いて、この割合がグリッド時刻の個数 n のみに依存し、区間の長さにはよらないことを示せ。$n = 50$、500、5000 に対してこの割合を測定せよ。n が 10 倍になるごとに割合はどのくらいの倍率で減少するか。またそのことは極限 $n \to \infty$ について何を示唆しているか。

3. **幾何ブラウン運動と $-\nu^2/2$ 補正。** $dX = \mu X\,dt + \nu X\,dW$ $(X_0 > 0)$ と、Equation 38.34 で与えられるその解を考えよ。

   1. $\mathbb{E}\big[e^{\lambda Z}\big] = e^{\lambda^2 v/2}$ $(Z \sim \mathcal{N}(0, v))$ を用いて、$\mathbb{E}[X_t] = X_0 e^{\mu t}$ および $\mathbb{E}[X_t^2] = X_0^2 e^{(2\mu+\nu^2)t}$ を示せ。また、X_t の中央値を求めよ。
   2. $\mu = 0.1$、$\nu = 0.6$ とする。$t \to \infty$ のとき、平均、中央値、および典型的な経路はそれぞれどうなるか。これらの主張が同時に成り立ちうるのはなぜか。
   3. どの ν に対して（固定された $\mu > 0$ のもとで）、中央値は成長しなくなるか。(b) のパラメータについて、`euler_maruyama` を用いて 100 本の経路を $t = 20$ まで $\Delta t = 10^{-2}$ でシミュレーションしてこの条件を確かめよ。終端で X_0 を下回る経路の割合はいくらか。また厳密解 Equation 38.34 はその割合をいくらと予測するか。

   Simo Särkkä と Arno Solin 著『Applied Stochastic Differential Equations』演習問題 4.3 より改編。

4. [code] **強収束次数とミルスタイン補正。** 38.4.4.2 節で説明したとおり、オイラー・丸山法はミルスタイン項 $\frac{1}{2} g\,\partial_x g\,\big((\Delta W)^2 - \Delta t\big)$ を省略しており、この項が消えない限り強収束次数は $\frac{1}{2}$ となる。

   1. Equation 38.34 より、$Y = \log X$ は加法的ノイズを持つ SDE を満たす。その SDE に `euler_maruyama` を適用し（強収束次数の実験の増分 `dW_f` とステップサイズを用いよ）、終端値の指数関数をとって `gbm_exact` に対する強誤差を測定せよ。また、その結果を説明せよ。

   2. 伊藤の補題を用いて、$dX = f(X)\,dt + g(X)\,dW$ $(g > 0)$ に対し、変数 $Y = \psi(X)$（ただし $\psi' = 1/g$）が振幅 1 の加法的ノイズを持つことを示し、そのドリフトを求めよ。

   3. ミルスタイン法 (Milstein 1975) はオイラー・丸山法のステップにその補正項を加えた次式である。

$$\hat{X}_{n+1} = \hat{X}_n + f\,\Delta t + g\,\Delta W_n + \tfrac{1}{2}\,g\,\partial_x g\,\big((\Delta W_n)^2 - \Delta t\big)\,, \tag{38.44}$$

   ここで f、g、$\partial_x g$ は $(\hat{X}_n, t_n)$ で評価する。強収束次数の実験における幾何ブラウン運動に対してこのスキームを実装し、同じ増分とステップサイズの上で強収束次数の傾きを測定せよ。$\Delta t = 2^{-8}$ において強誤差は何分の 1 に減少するか。

5. **厳密に見た分散保持。** X_t が、平均ゼロのデータ X_0 に対して Equation 38.40 に従うとする。

   1. データが単位分散を持つと仮定し、$\mathbb{E}[X_t^4] = 3 + \bar{\alpha}_t^2\,\big(\mathbb{E}[X_0^4] - 3\big)$ を示せ。さらにラデマッハデータ $X_0 = \pm 1$ に特殊化し、本文の予測式 $3 - 2\bar{\alpha}_t^2$ を導け。
   2. 超過尖度 $\mathbb{E}[X^4]/\mathrm{Var}(X)^2 - 3$ はすべてのガウス分布に対してゼロになる。t に関して、X_t の超過尖度はどのレートで減衰するか。このレートを、条件付き平均 $\mathbb{E}[X_t \mid X_0]$ がデータを忘れるレートと比較せよ。

3. 代わりにデータが分散 s^2 を持つとする。$\mathrm{Var}(X_t)$ を求めよ。$s=2$ かつ $\theta=1$ のとき、$\mathrm{Var}(X_T)$ が 1% 以内の誤差で基準分布 $\mathcal{N}(0,1)$ の分散に近づくまでには、過程をどれだけ長く走らせる必要があるか。

6. [code] **伊藤とストラトノビッチ。**●$\int_0^t W\,dW$ に対する左端点和と右端点和を足し合わせると任意の分割上で W_t^2 になるため（38.4.2.2 節）、両者の平均は $\frac{1}{2}W_t^2$ に等しい。この種の和はストラトノビッチ積分 (Stratonovich 1966) を定義し、通常の連鎖律に従う。

   1. 38.2 節のホイン法は、ノイズ項を第 2 の傾きと同様に扱えば SDE へと拡張できる。予測子はオイラー・丸山法のステップ $\tilde{X}=\hat{X}_n+f(\hat{X}_n)\,\Delta t+g(\hat{X}_n)\,\Delta W_n$ であり、修正子は両方の係数を平均する。

$$\hat{X}_{n+1}=\hat{X}_n+\tfrac{1}{2}\left[f(\hat{X}_n)+f(\tilde{X})\right]\Delta t+\tfrac{1}{2}\left[g(\hat{X}_n)+g(\tilde{X})\right]\Delta W_n. \tag{38.45}$$

      強収束次数の実験における幾何ブラウン運動に対し、その増分 `dW_f` とステップサイズを用いてこれを実装せよ。終端値を厳密な伊藤解 Equation 38.34 および $X_0\exp(\mu T+\nu W_T)$ と比較せよ。このスキームは 2 つのうちどちらに収束するか。

   2. 修正子のノイズ項を $\tilde{X}-\hat{X}_n$ の 1 次まで展開し、$(\Delta W_n)^2\approx\Delta t$ を用いて、このスキームが修正されたドリフトを持つ伊藤 SDE に対するオイラー・丸山法のように振る舞うことを示せ。そのドリフトを求め、(a) の極限が説明できることを確かめよ。

   3. Equation 38.25 の形を持つすべての SDE に対して、2 つの流儀が同じ解を与えるのはなぜか。

7. **線形 SDE の厳密な遷移。**●連続な係数を持つ線形 SDE $dX=-\theta(t)X\,dt+g(t)\,dW$（$\theta$ と g は連続）を考え、$\Theta(t)=\int_0^t\theta(u)\,du$ とおく。

   1. $\phi(x,t)=e^{\Theta(t)}x$ に伊藤の補題を適用し、X_s が与えられたとき、後の時刻の状態 X_t（時刻 t）が平均 $e^{-(\Theta(t)-\Theta(s))}X_s$、分散 $\int_s^t e^{-2(\Theta(t)-\Theta(u))}\,g(u)^2\,du$ のガウス分布に従うことを示せ。
   2. (a) を VP-SDE Equation 38.41 に特殊化し、任意の長さの遷移が単位分散の状態を単位分散の状態へ写すことを示せ。逆に、(a) の線形 SDE において、$\mathrm{Var}(X_t)=1$ が任意の $t\ge 0$ に対して（$\mathrm{Var}(X_0)=1$ であるときは常に）成り立つと仮定し、$g(t)^2=2\,\theta(t)$ がすべての t で成り立つことを示せ。
   3. 半線形 SDE $dX=(-\theta X+q(X))\,dt+g\,dW$（定数 $\theta,g>0$、滑らかな関数 q）を考えよ。指数オイラー・丸山ステップでは、q を値 $q(\hat{X}_n)$ に長さ Δt のステップの間固定し、残りの線形 SDE を (a) の方法で厳密に解く。このステップを書き下せ。それがオイラー・丸山ステップ Equation 38.42 と Δt の 1 次の範囲で一致すること、また q が定数のとき厳密になることを示せ。$q\equiv 0$ のとき、オイラー・丸山法が平均を減衰させるのは $\Delta t<2/\theta$ の場合に限られる（38.2.3 節）。指数ステップのステップサイズを制限する要因は何か。
   4. 厳密なステップは任意のステップサイズで弱誤差がゼロになる。VP-SDE のドリフトと拡散のどの性質がそれを可能にしているか。拡散モデルの生成サンプラは、ドリフトが $-\frac{1}{2}\beta(t)\,x-\beta(t)\,\nabla\log p_t(x)$ である SDE を時間逆向きにシミュレーションする。これは VP ドリフトに、順方向周辺分布の**スコア** $\nabla\log p_t$（ガウス分布でないデータに対しては x の非線形関数）の定数倍を加えたものである。そのステップを厳密にできないのはなぜか。拡散サンプラに対する指数積分器 (Zhang and Chen 2023) は、このサンプラの決定論的な対応物、すなわちスコア項を半分にした同じ 2 つの部分か

らなるドリフトを持つ ODE（38.5 節）に (c) の考え方を適用する。このようなドリフトのどの部分をステップは厳密に扱い、どの部分を近似するか。

8. [extended] **マルチレベルモンテカルロ法。●●**オイラー・丸山法の経路にわたる平均によって $\mathbb{E}[\varphi(X_T)]$ を推定するとき、誤差の原因はスキームの弱い意味での統計的バイアスとサンプリングノイズの 2 つである。マルチレベルモンテカルロ法 (Michael B. Giles 2008) は異なるステップサイズの経路同士を結合することでコストを削減する手法であり、これは 38.4.4.2 節で挙げた強い意味での精度の用途の 1 つである。本問を通じて、平均二乗強誤差 $\mathbb{E}\big[(\hat{X}_N - X_T)^2\big]^{1/2}$ は同項の強誤差と同じ次数を持つものとする。

   1. 通常のモンテカルロ法は $\varphi(\hat{X}_N)$ を M 本の独立な経路にわたって平均し、そのコストは $MT/\Delta t$ ステップである。弱い意味でのバイアスを $c_1\,\Delta t$、分散を v（対象は $\varphi(\hat{X}_N)$）と仮定せよ。平均二乗誤差を ε^2 以下にするには、オーダー ε^{-3} ステップ（M と Δt を最良に選んだ場合）のコストがかかることを示せ。
   2. P_l をオイラー・丸山法の終端における φ の値（ステップ幅 $h_l = T\,2^{-l}$）とし、$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^{L} \mathbb{E}[P_l - P_{l-1}]$ と書く。各項は M_l 組の独立なペアから推定し、強収束次数の実験と同様に、粗い経路には連続する 2 個ずつの細かい増分の和を用いる。リプシッツ連続な φ に対し、$V_l = \mathrm{Var}(P_l - P_{l-1})$ を平均二乗強誤差によって上から抑えよ。$\varphi(x) = x$ とし、同実験のパラメータにおける OU 過程と幾何ブラウン運動に対して、V_l が h_l に応じてどのようにスケールするかを予測せよ。また、V_l を $l = 1, \dots, 8$ に対して `euler_maruyama` で測定し、細かいレベルにおける傾きを当てはめよ。
   3. ラグランジュ乗数を用いて、総コスト $\sum_l M_l/h_l$ を $\sum_l V_l/M_l \le \varepsilon^2/2$ の制約のもとで最小化せよ。最も細かいレベル L を、バイアスが $\varepsilon/\sqrt{2}$ 以下にとどまるように選べ。$V_l \propto h_l^{r}$ と仮定し、コストがオーダー ε^{-2}（$r = 2$ のとき）、およびオーダー $\varepsilon^{-2}(\log \varepsilon)^2$（$r = 1$ のとき）になることを示せ。
   4. 強収束次数の実験における幾何ブラウン運動の $\mathbb{E}[X_T] = X_0 e^{\mu T}$ に対し、(c) の M_l と測定した V_l を用いて両方の推定量を実装せよ。$\varepsilon \in \{0.2, 0.1, 0.05, 0.02\}$ に対して各推定量が使用する総ステップ数を記録し、数回の反復実行にわたってその誤差を確かめよ。ε^2 とコストの積を ε に対してプロットせよ。さらにオイラー・丸山法の代わりに演習問題 4c のミルスタイン法を用いて同じ実験を行え。`euler_maruyama` はすべての経路全体をメモリに保持するため、大きい M_l はバッチに分けてシミュレーションせよ。
   5. 不連続な $\varphi(x) = \mathbf{1}(x > X_0)$ に対して (b) の測定を繰り返せ。なぜ V_l の減衰は遅くなるのか。また、減衰が遅くなることでコストはどう変化するか。

38.5 フォッカー・プランク方程式と確率フロー

前節では、ノイズで揺さぶりながら粒子を原点へと引き戻す確率微分方程式（stochastic differential equation、SDE）$dX = -X\,dt + \sqrt{2}\,dW$ であるオルンシュタイン・ウーレンベック過程（Ornstein–Uhlenbeck process）をシミュレーションした。実行するたびに異なる経路が描かれ、個々の経路を予測することはできない。それでも、実行を重ねても毎回繰り返されるものがある。10 万個の粒子を $x = 2$ から出発させ、時刻 $t = 0.5$ におけるヒストグラムを見てみよう。どの実行でも、中心が 1.21 付近、標準偏差が 0.80 付近の同じ釣鐘曲線になる。経路のランダム性は平均化によって打ち消し合い、粒子群の密度は決定論的に動く。では、その密度を動かしているのはどの方程式だろうか。

Adriaan Fokker と Max Planck は、定常的な力とランダムな揺さぶりを受ける粒子に対してその方程式を書き下し (Fokker 1914; Planck 1917)、Andrey Kolmogorov はそれがあらゆる拡散

過程の密度を支配することを示した (Kolmogorov 1931)。この方程式には 2 つの項がある。ドリフトが密度を運び、ノイズが密度を平滑化する。さらに、この平滑化は輸送、すなわち密度の高い場所から低い場所へと確率を動かすフローとして書き換えることもできる。密度のレベルで見れば、ノイズは決定論的な速度のように振る舞うのである。

この事実は生成モデル（generative model）にとって重要である。拡散モデル（diffusion model）は、ノイズ付加過程を逆向きに実行することでデータを生成する。多くの出発点が同じ終点へと至るため、ノイズを帯びた経路の終点を出発点へと戻す写像は存在しない。しかし、粒子群全体の密度がどのように変化したかはわかっており、密度が逆向きに変化するような過程を構成することはできる。その 1 つは、ノイズを外向きのフローに置き換えて得られる常微分方程式（ordinary differential equation、ODE）であり、その粒子はあらゆる時刻で元の確率過程（stochastic process）と同じヒストグラムを持つ。もう 1 つは再び確率過程であり、ノイズをデータへと変換する。いずれにおいても、ドリフトとノイズの大きさは自由に選べる。しかし 1 つだけ選べない量がある。あらゆる点と時刻において対数密度（log-density）が増加する方向である。

本節では、オルンシュタイン・ウーレンベック過程と、その方向が閉形式で書ける 2 モードの「データ」密度を用いて各ステップを確認する。実データに対してその方向は未知であり、これこそ本節が未解決のまま次節へ委ねる問いである。導出の細部を忘れたとしても、3 つの式だけは心に留めておいてほしい。フォッカー・プランク方程式（Fokker–Planck equation）Equation 38.52、その平滑化項を輸送へと書き換える恒等式 Equation 38.63、そして逆過程（reverse process）の族 Equation 38.71 である。発散定理（divergence theorem）に慣れ親しんでいる読者は 38.5.2.1 節を飛ばしてよく、末尾の尤度（likelihood）の公式とバイアスの計算は再読時に回してもよい。

38.5.1 経路から密度へ

38.5.1.1 経路と周辺分布

本節で扱うすべての SDE は次の形をとる。

$$d\mathbf{X} = \mathbf{f}(\mathbf{X}, t)\,dt + g(t)\,d\mathbf{W},$$

ここで、ドリフトは $\mathbf{f} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$、スカラーのノイズ振幅は $g(t)$、d 次元のブラウン運動（Brownian motion）は $\mathbf{W}$、ランダムな初期値は $\mathbf{X}_0 \sim p_0$ である。短いステップのあいだに、粒子は $\mathbf{f}\,dt$ に加えて、各座標で分散 $g^2\,dt$ の独立なガウス分布に従う揺さぶりの分だけ移動する。

ブラウン運動の 1 つの実現値から 1 本の経路 $\mathbf{X}_t$ が生じ、それは実行するたびに異なる。本節の対象は経路ではなく、密度 $p_t(\mathbf{x})$（確率変数 $\mathbf{X}_t$ の密度）であり、これを**時間周辺分布**（time-marginal）と呼ぶ。これは $\mathbf{X}_0$ の抽出と途中のノイズという 2 つのランダム性の源について平均をとったものであり、この平均化によって決定論的な対象となる。

通し例として用いる過程は 38.4.3 節のオルンシュタイン・ウーレンベック（OU）過程 $dX = -\theta X\,dt + g\,dW$ である。1 点 $X_0 = x_0$ から出発したとき、その周辺分布はガウスカーネルとなり、次のように書ける。

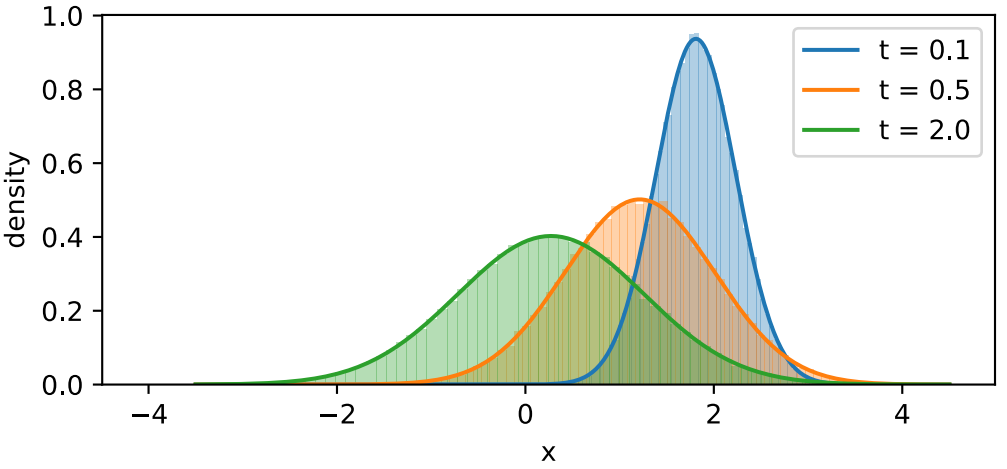
$$X_t \sim \mathcal{N}\!\left(\alpha_t x_0,\ \sigma_t^2\right), \qquad \alpha_t = e^{-\theta t}, \quad \sigma_t^2 = \frac{g^2}{2\theta}\left(1 - e^{-2\theta t}\right). \tag{38.46}$$

本節全体を通して $\theta = 1$ および $g = \sqrt{2}$ とおく。このとき分散は $g^2/(2\theta) = 1$ で飽和し、定常分布は $\mathcal{N}(0,1)$ となる。ドリフトは粒子群の中心を動かし、ノイズは粒子群を広げる。フォッカー・プランク方程式は、これら両方の作用を p_t 上の作用素へと変換する。

38.5.1.2 シミュレーションによる周辺分布

SDE の節では、このような粒子群のモーメントをカーネルと比較した。ここではヒストグラム全体を見る。100,000 個の粒子が $x_0 = 2$ から出発してステップ幅 $\Delta t = 0.002$ のオイラー・丸山法（Euler–Maruyama method、38.4.4.1 節）のステップで動き、時刻 $t = 0.1$、0.5、2 において粒子群を厳密なガウス密度と重ねて描く。

```
theta, g = 1.0, np.sqrt(2.0)            # OU: dX = -theta X dt + g dW

def ou_mean_var(x0, t, theta=theta, g=g):
    """Mean and variance of X_t started at x0: N(alpha_t x0, sigma_t^2)."""
    alpha = np.exp(-theta * t)
    return alpha * x0, g**2 / (2 * theta) * (1 - alpha**2)

rng = np.random.default_rng()
n, dt, snap_at = 100_000, 0.002, {50: 0.1, 250: 0.5, 1000: 2.0}  # step -> t
x, snaps = np.full(n, 2.0), {}          # every particle starts at x0 = 2
for k in range(1, max(snap_at) + 1):    # Euler--Maruyama
    x += -theta * x * dt + g * np.sqrt(dt) * rng.standard_normal(n)
    if k in snap_at:
        snaps[snap_at[k]] = x.copy()

grid = np.linspace(-3.5, 4.5, 400)
d2l.set_figsize((6, 2.5))
for (t, cloud), c in zip(snaps.items(), ('C0', 'C1', 'C2')):
    m, v = ou_mean_var(2.0, t)
    d2l.plt.hist(cloud, bins=80, density=True, alpha=0.35, color=c)
    d2l.plt.plot(grid, np.exp(-(grid - m)**2 / (2 * v)) / np.sqrt(2 * np.pi * v),
                 color=c, label=f't = {t}')
d2l.plt.xlabel('x'); d2l.plt.ylabel('density'); d2l.plt.legend();
```



3 つの時刻のいずれにおいても、ヒストグラムは曲線の上に重なっている。Equation 38.46 により、これらの曲線は平均 $2e^{-t}$、標準偏差 $\sqrt{1 - e^{-2t}}$ を持ち、粒子群に当てはめたものではなく公式から直接描いたものである。$t = 2$ においても平均はまだ $2e^{-2} \approx 0.27$ であり、標準偏差のおよそ 4 分の 1 だけ定常分布 $\mathcal{N}(0, 1)$ から離れている。初期位置を忘れるには時間がかかるので

ある。これは 1 回の実行結果にすぎないが、一致している 3 本の曲線にはランダム性が一切含まれていない。2 回目の実行ではすべての経路が異なるにもかかわらず、同じ曲線に一致するはずである。いよいよ、これらの曲線を動かす方程式を見つけよう。

38.5.2 フォッカー・プランク方程式

個々の経路は時間で微分できないが、$\mathbf{X}_t$ の滑らかな関数について全経路にわたる平均をとれば時間で微分できる。平均に関する記述を密度に関する記述へと書き換えるには、導関数を関数側から密度側へと移す必要があり、そのためには $\mathbb{R}^d$ 上の場に関する 3 つの事実が必要になる。

38.5.2.1 発散、ラプラシアン、部分積分

これら 3 つはいずれも 34.2 節の勾配から構成され、それぞれ後で再び必要となる物理的な意味を持つ。滑らかなベクトル場（vector field）$\mathbf{u} : \mathbb{R}^d \to \mathbb{R}^d$ を流体の速度として思い描き、$\psi : \mathbb{R}^d \to \mathbb{R}$ を滑らかなスカラー場とする。

発散。単位時間あたりに小さな箱からどれだけの流体が流出するかを知りたい。$\mathbf{u}$ の**発散**（divergence）とは、次のスカラー場である。

$$\nabla \cdot \mathbf{u} = \sum_{i=1}^{d} \frac{\partial u_i}{\partial x_i}. \tag{38.47}$$

これは $\mathbf{x}$ のまわりの無限小の箱から流体が流出する正味の流量をその箱の体積で割ったものであり、湧き出し（ソース）では正、吸い込み（シンク）では負になる。平面上の 2 つの場がその典型例を示す。膨張する場 $\mathbf{u} = (x_1, x_2)$ では $\nabla \cdot \mathbf{u} = 1 + 1 = 2$ となるため、面積 A のどの箱も場所によらず $2A$ の割合で流体を失う。一方、回転 $\mathbf{u} = (-x_2, x_1)$ では $\nabla \cdot \mathbf{u} = 0 + 0 = 0$ であり、流体は循環するのみで、どの箱も流体を増減させない。したがって、発散がゼロであることは場そのものがゼロであることを意味しない。Figure 38.17 では 1 つの箱の上でこれら両方の性質を確かめている。

多数の小さな箱からの流出を足し合わせると、内部で隣接するすべての面での出入りが打ち消し合い、外側の境界を通過する流束だけが残る。これが**発散定理**である。有界領域 Ω とその外向き単位法線ベクトル $\mathbf{n}$ に対して、次が成り立つ。

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, d\mathbf{x} = \oint_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, dS. \tag{38.48}$$

領域内部の発散の総和は、その境界を通って外へ出る正味の流量に等しい。1 次元では、これは微分積分学の基本定理（fundamental theorem of calculus）$\int_a^b u'(x)\,dx = u(b) - u(a)$ にほかならない。本節ではこの定理を証明なしに用いるが、多変数微分積分のどの教科書にも証明が載っている。

ラプラシアン。関数がその周囲の値と比べてどうなっているかを測る量がほしい。**ラプラシアン**とは勾配の発散である（関数の前に置かれた Δ はこの作用素を表し、Δt のような増分ではない）。

$$\Delta \psi = \nabla \cdot (\nabla \psi) = \sum_{i=1}^{d} \frac{\partial^2 \psi}{\partial x_i^2}. \tag{38.49}$$

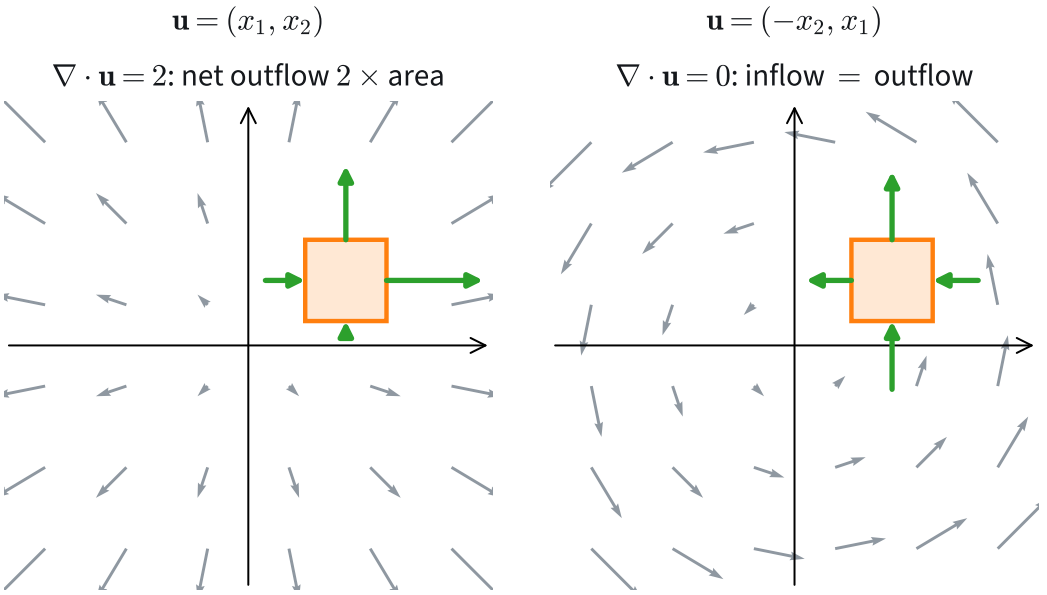


Figure 38.17: 一辺 0.5、中心 $(0.6, 0.4)$ の箱における正味の流出としての発散。緑の各矢印は辺の中点における場の法線成分を表し、流出は辺から離れる向きに、流入は辺に向かう向きに描かれている。(a) $\mathbf{u} = (x_1, x_2)$ では、流体は左辺と下辺から入り、右辺と上辺から出ていく。流出量は流入量を面積の $2\times$ だけ上回る。(b) 回転 $\mathbf{u} = (-x_2, x_1)$ では、右辺と下辺からの流入量が残り 2 辺からの流出量と等しい。

これはある点における ψ の値を、その点のまわりの半径 r の小さな球面上の平均と比較するものである。その平均は $\psi(\mathbf{x}) + \frac{r^2}{2d}\Delta\psi(\mathbf{x}) + O(r^4)$ となり、このことは演習問題 1 で証明する。1 次元で $\psi(x) = x^2$ とすると、「球面」は 2 点 $x \pm r$ の組であり、平均は $\frac{1}{2}\big((x-r)^2 + (x+r)^2\big) = x^2 + r^2$、また $\frac{r^2}{2}\psi'' = r^2$ となるため、この公式は剰余項なしで成り立つ（$x = 1$、$r = 0.3$ では平均は 1.09 である）。したがって $\Delta\psi > 0$ は、ψ が近傍の平均より下にあることを意味する。ラプラシアンがゼロであっても定数関数とは限らない。たとえば $\psi = x_1^2 - x_2^2$ は $\Delta\psi = 2 - 2 = 0$ を満たし、あらゆる円周上の平均と一致する。

部分積分。これは d 次元版の**部分積分**（integration by parts）であり、34.6.2.3 節の 1 次元の規則を拡張して、導関数を一方の因子から他方の因子へと移す道具である。ψ と $\mathbf{u}$ が滑らかであり、$\psi\,\mathbf{u}$ が無限遠で十分に速く減衰するならば（本節では、ψ が有界集合の外でゼロになるか、両因子がガウス分布の裾を持つために成り立つ）、次が成り立つ。

$$\int \psi\,(\nabla \cdot \mathbf{u})\,d\mathbf{x} = -\int \nabla\psi \cdot \mathbf{u}\,d\mathbf{x}. \tag{38.50}$$

導関数は $\mathbf{u}$ から ψ へと移り、符号が反転する。

証明積の微分法則より $\nabla\cdot(\psi\mathbf{u}) = \nabla\psi\cdot\mathbf{u} + \psi\,\nabla\cdot\mathbf{u}$ が得られる。両辺を半径 R の球体上で積分し、$R \to \infty$ とする。発散定理により、左辺は $\psi\mathbf{u}$ が半径 R の球面を通過する流束となり、減衰の仮定のもとでゼロに収束する。$\blacksquare$

まず $\mathbf{u} = \nabla p$ として適用し、次に役割を入れ替えてもう一度適用すれば、ラプラシアン全体が積分記号の向こう側へと移り、2 回の符号反転が打ち消し合う。すなわち $\int \psi\,\Delta p = -\int \nabla\psi\cdot\nabla p = \int p\,\Delta\psi$ となる。

38.5.2.2 伊藤の補題から PDE へ

ここで p_t に対する方程式を導出しよう。以下では一貫して、$\mathbf{f}$ が $\mathbf{x}$ について連続微分可能かつ t について連続であり、g が連続であり、p_t が滑らかかつ正であり、さらに t に関する導関数を積分記号の下でとれるものと仮定する。

基本的な着想は、平均を通して粒子群を観察することである。ϕ を、小さな箱の上で 1 に等しく、それより少し大きい箱の外で 0 になる滑らかなバンプ関数とすれば、$\mathbb{E}[\phi(\mathbf{X}_t)]$ はおおよそその箱の確率になる。このような平均のすべてが時間とともにどう変化するかがわかれば、密度がどう変化するかも定まる。そこで、有界集合の外でゼロになる任意の滑らかな**テスト関数**（test function）$\phi : \mathbb{R}^d \to \mathbb{R}$ をとる。

伊藤の補題（Itô's lemma、38.4.2.4 節）は SDE の経路に対する連鎖律（chain rule）である。$d\mathbf{X} = \mathbf{f}\,dt + \mathbf{G}\,d\mathbf{W}$ に対して、通常の連鎖律に 2 次の項 $\frac{1}{2}\sum_{ij}(\mathbf{G}\mathbf{G}^{\top})_{ij}\,\partial_i\partial_j\phi$ が加わる。ブラウン運動の増分の二乗 $(dW)^2$ が無視できる大きさではなく dt のオーダーを持つからである。$\mathbf{G} = g(t)\,\mathbf{I}$ のとき、この項は $\frac{1}{2}g^2$ と ϕ のラプラシアンの積になる。

$$d\phi(\mathbf{X}_t) = \big(\nabla\phi\cdot\mathbf{f} + \tfrac{1}{2}g^2\Delta\phi\big)\,dt + g\,\nabla\phi\cdot d\mathbf{W}.$$

関数は、通常の連鎖律にラプラシアン補正を加えたドリフト部分と、ノイズ部分によって変化する。ここで期待値をとろう。$g\nabla\phi$ の伊藤積分（Itô integral）は平均がゼロである。各ブラウン増分がその区間の始点における被積分関数と独立だからである（38.4.2.2 節）。したがってノイズ部分は消える。

$$\frac{d}{dt}\,\mathbb{E}[\phi(\mathbf{X}_t)] = \mathbb{E}\big[\nabla\phi\cdot\mathbf{f} + \tfrac{1}{2}g^2\Delta\phi\big]\,. \tag{38.51}$$

ϕ の平均は、伊藤の補題が与えた変化率の平均の速さで変化する。両辺は周辺密度に対する積分である。$\mathbb{E}[\,\cdot\,] = \int (\cdot)\,p_t(\mathbf{x})\,d\mathbf{x}$ と書き、d/dt を積分記号の下へ移すと次を得る。

$$\int \phi\;\partial_t p_t\;d\mathbf{x} = \int (\nabla\phi\cdot\mathbf{f})\,p_t\;d\mathbf{x} + \tfrac{1}{2}g^2\int (\Delta\phi)\,p_t\;d\mathbf{x}.$$

右辺の導関数はまだ ϕ に作用しているので、これらを p_t の側へ部分積分 Equation 38.50 によって移す。ϕ は有界集合の外でゼロになるため、境界項は現れず、p_t の減衰条件も不要である。ドリフト項は $\psi=\phi$、$\mathbf{u}=\mathbf{f}p_t$ として 1 回の部分積分を要し、拡散項は 2 回を要して符号が打ち消し合う。

$$\int (\nabla\phi\cdot\mathbf{f})\,p_t\,d\mathbf{x} = -\int \phi\,\nabla\cdot(\mathbf{f}\,p_t)\,d\mathbf{x}, \qquad \int (\Delta\phi)\,p_t\,d\mathbf{x} = \int \phi\,\Delta p_t\,d\mathbf{x}.$$

両者を代入すると、ϕ に対する 1 つの積分にまとまる。

$$\int \phi\,\Big(\partial_t p_t + \nabla\cdot(\mathbf{f}\,p_t) - \tfrac{1}{2}g^2\Delta p_t\Big)\,d\mathbf{x} = 0.$$

これは任意のテスト関数 ϕ に対して成り立つ。括弧内の関数は連続であり、あらゆるバンプ関数との積分がゼロになる連続関数は恒等的にゼロである（もし正になる点があれば、そこにバンプ関数を置くことで積分が正になってしまう）。したがって括弧内は各点でゼロとなり、p_t は**フォッカー・プランク方程式**（**コルモゴロフ前進方程式**（Kolmogorov forward equation）とも呼ばれる）を満たす。

$$\partial_t p_t(\mathbf{x}) = -\nabla\cdot\big(\mathbf{f}(\mathbf{x},t)\,p_t(\mathbf{x})\big) + \tfrac{1}{2}g(t)^2\,\Delta p_t(\mathbf{x}). \tag{38.52}$$

第 1 項はドリフトの寄与、第 2 項はノイズの寄与である。行列ノイズ $\mathbf{G}\,d\mathbf{W}$ の場合、第 2 項は $\frac{1}{2}\sum_{ij}\partial_i\partial_j(a_{ij}p_t)$（ただし $\mathbf{a}=\mathbf{G}\mathbf{G}^{\top}$）となり、何が変わるかは演習問題 5 で導く。物理学で用いられるこの方程式の多様な変種は Risken (1996) にまとめられている。

この導出が示しているのは片方向のみ、すなわち SDE の周辺分布がこの方程式を満たすということだけである。後ほどその逆、すなわち p_0 から出発する解が周辺分布**そのものである**という事実が必要になり、そのためには方程式の解がただ 1 つでなければならない。本節で用いるすべてのドリフトが満たす緩やかな条件のもとで、この一意性は実際に成り立つ。すなわち、$g>0$ であり、ドリフトが $\mathbf{x}$ について t に関して一様にリプシッツ連続（$\mathbf{x}$ 方向の傾きがすべての t に共通する 1 つの定数で抑えられること）であればよい。このとき、与えられた初期分布を持つ確率密度（probability density）の中で解は一意に定まる（厳密な主張は Bogachev et al. (2015) を参照）。以下ではこの一意性を証明なしに用い、使う箇所ではその旨を明記する。

38.5.2.3　熱方程式と 2 つの項

ドリフトを切ってみよう。するとフォッカー・プランク方程式は**熱方程式**（heat equation）になる。

$$\mathbf{f} = \mathbf{0}: \qquad \partial_t p_t = \tfrac{1}{2} g^2 \, \Delta p_t. \tag{38.53}$$

密度は、近傍の平均より下にある場所ではどこでも上昇する。一般的な話に進む前に、1 点で確かめておこう。$g = 1$、$d = 1$ とし、$p_t = \mathcal{N}(0, t)$（0 から出発したブラウン運動の分布）をとる。$p = e^{-x^2/(2t)}/\sqrt{2\pi t}$ を直接微分すると、$\partial_t p = p\left(\frac{x^2}{2t^2} - \frac{1}{2t}\right)$ および $\partial_{xx} p = p\left(\frac{x^2}{t^2} - \frac{1}{t}\right)$ が得られる。$x = 0$、$t = 1$ では $p = 0.399$、$\partial_t p = -0.1995$、$\frac{1}{2}\partial_{xx} p = -0.1995$ となる。ピークはまさにラプラシアンが予測する速さで沈み込んでいく。

より一般に、ガウス分布 $\mathcal{N}(\mathbf{m}, v_t \mathbf{I})$ は、分散が $v_t = v_0 + g^2 t$（定数 g の場合）あるいは $v_t = v_0 + \int_0^t g(\tau)^2 \, d\tau$（$g$ が時間変化する場合）に従って増大するとき熱方程式の解となる。これはブラウン運動の $\mathrm{Var}(W_t) = t$ という法則にほかならない。ノイズのみの過程は 38.4.1.1 節の**分散爆発**（variance-exploding）族であり、拡散モデルでは増大する $g(t)$ とともに用いられる。

ここで Equation 38.52 の 2 つの項を読み解こう。第 1 項 $-\nabla \cdot (\mathbf{f} p)$ は**移流**（advection）であり、ドリフトが小さな箱へ運び込む質量より外へ運び出す質量の方が多い場所で密度を減少させる。第 2 項 $\frac{1}{2} g^2 \Delta p$ は**拡散**（diffusion）であり、密度が近傍の平均より低い場所では密度を引き上げ、高い場所では引き下げる。これは伊藤の $(dW)^2 = dt$ 補正を密度のレベルで見たものにほかならない。

38.5.2.4 OU 方程式のガウス解

OU 過程に対しては閉形式の答えがわかっているため、解の族全体にわたって方程式を検証できる。1 次元で $f(x) = -\theta x$ とすると、Equation 38.52 は次のようになる。

$$\partial_t p = \theta \, \partial_x (x \, p) + \tfrac{1}{2} g^2 \, \partial_{xx} p.$$

偏微分方程式（partial differential equation、PDE）の未知量は関数全体であり、手計算で求まることはめったにない。しかし、ここでは幸運に恵まれる。ガウス分布に対しては、この方程式が平均と分散それぞれに対する 2 本の常微分方程式へと帰着するのである。

命題（OU 方程式のガウス解） *$p(x,t) = \mathcal{N}(x; m(t), v(t))$ を、微分可能な平均 $m(t)$ と分散 $v(t) > 0$ を持つガウス密度とする。このとき、p が OU フォッカー・プランク方程式を満たすための必要十分条件は次が成り立つことである。*

$$\dot{m} = -\theta m, \qquad \dot{v} = g^2 - 2\theta v. \tag{38.54}$$

証明ガウス密度に対して $\partial_x p = -\frac{x-m}{v} \, p$ が成り立つ。直接微分すると、PDE の両辺はそれぞれ p と $(x - m)$ の二次多項式との積になり、まず左辺は次のようになる。

$$\partial_t p = p\left[\frac{\dot{m}}{v}(x - m) + \frac{\dot{v}}{2v^2}\big((x-m)^2 - v\big)\right],$$

一方、ドリフト項で $x = (x - m) + m$ と書けば、右辺は次のようになる。

$$\theta \, \partial_x (xp) + \tfrac{g^2}{2} \partial_{xx} p = p\left[-\frac{\theta m}{v}(x - m) + \left(\frac{g^2}{2v^2} - \frac{\theta}{v}\right)\big((x-m)^2 - v\big)\right].$$

関数 $1, (x-m), (x-m)^2$ は線形独立であるため、すべての x で両辺が一致するのは各係数が一致するときに限られる。1 次の項から $\dot m = -\theta m$ が、2 次の項から $\dot v = g^2 - 2\theta v$ が得られる（定数項からも後者と同じ式が出る）。■

平均は減衰し、分散は $g^2/(2\theta)$ へと緩和する。初期条件 $m(0)=m_0$、$v(0)=v_0$ からこの 2 本の ODE を解くと次を得る。

$$m(t) = m_0 e^{-\theta t}, \qquad v(t) = v_0 e^{-2\theta t} + \frac{g^2}{2\theta}\bigl(1 - e^{-2\theta t}\bigr).$$

1 点から出発する場合（$v_0 = 0$）、これはカーネル Equation 38.46 にすべての $t > 0$ で一致するため、38.4.3 節の遷移密度（transition density）はフォッカー・プランク方程式を満たす。

不動点 $\dot m = \dot v = 0$ は $m = 0$、$v = g^2/(2\theta)$ であり、定常ガウス分布が PDE の定常状態 $\partial_t p = 0$ に対応する（演習問題 3 ではガウス分布の形を仮定せずにこれを導出する）。また、$\theta = 0$ とおけば熱方程式の $\dot v = g^2$ が復元される。

この幸運には理由があり、限界もある。ドリフトが線形であるため、ガウス分布はガウス分布のまま保たれ、最初の 2 つのモーメントが閉じた系をなす。同じ演習問題で扱う二重井戸（double-well）ドリフト $f(x) = x - x^3$ では、$\mathbb{E}[X^2]$ の方程式に $\mathbb{E}[X^4]$ が入り、$\mathbb{E}[X^4]$ の方程式に $\mathbb{E}[X^6]$ が入って、この連鎖は決して閉じない。

38.5.2.5 線形性と混合分布

実データもまたガウス分布ではないため、より豊かな構造を持ちながらも閉形式で追跡できる密度が必要である。線形ドリフトのもとでは、混合ガウス分布（Gaussian mixture）がその条件を満たす。本節の通し例として用いる「データ」密度は次のとおりである。

$$p_0 = \tfrac{1}{2}\,\mathcal{N}(-2,\,0.25) + \tfrac{1}{2}\,\mathcal{N}(2,\,0.25),$$

これは標準偏差 0.5 のよく離れた 2 つのモードからなる分布である。p_0 からの抽出では、まず成分 $\mathcal{N}(\mu_k, v_k)$ を選び、次にそこから 1 点を抽出する。カーネル Equation 38.46 により、X_t は $\alpha_t X_0$ に独立な確率変数 $\mathcal{N}(0, \sigma_t^2)$ を足したものになる。したがって成分を固定すれば、それは 2 つの独立なガウス変数の和であり、平均 $\alpha_t \mu_k$、分散 $\alpha_t^2 v_k + \sigma_t^2$ を持つ。成分について平均をとれば周辺分布、すなわち時間発展した各成分の混合分布が得られる。$\theta = 1$、$g = \sqrt{2}$ のとき、これは $\mathcal{N}\bigl(\pm 2e^{-t},\, 1 - 0.75\,e^{-2t}\bigr)$ の混合分布となり、あらゆる時刻で閉形式で表される。

同じ事実は、p について線形であるフォッカー・プランク方程式からも直接読み取れる。解の混合 $w_1 p_t^{(1)} + w_2 p_t^{(2)}$ もまた解となり、Equation 38.52 の直後で述べた一意性により、それが混合初期分布に対する周辺分布となる。OU 過程のもとでは、各ガウス成分がそれぞれ単独でモーメント ODE Equation 38.54 に従う。

以降の実験はすべてこの周辺分布を用いるため、その累積分布関数（CDF）およびサンプラとあわせてここで一度だけ定義しておく。演習問題で変更できるよう、パラメータは明示的な引数としてある。

```
ws, mus, v0s = np.array([0.5, 0.5]), np.array([-2.0, 2.0]), np.array([0.25, 0.25])
erf = np.vectorize(math.erf)            # the error function, on arrays

def mixture_params(t, mus=mus, v0s=v0s, theta=theta, g=g):
    """Means and variances of the OU-evolved components (moment ODEs)."""
```

```
    alpha = np.exp(-theta * t)
    return alpha * mus, alpha**2 * v0s + g**2 / (2 * theta) * (1 - alpha**2)

def gauss(x, m, v):
    return np.exp(-(x - m)**2 / (2 * v)) / np.sqrt(2 * np.pi * v)

def p_t(x, t, ws=ws, mus=mus, v0s=v0s):
    """Exact marginal density of the mixture at time t."""
    m, v = mixture_params(t, mus, v0s)
    return sum(w * gauss(x, mk, vk) for w, mk, vk in zip(ws, m, v))

def mixture_cdf(x, t, ws=ws, mus=mus, v0s=v0s):
    m, v = mixture_params(t, mus, v0s)
    return sum(w * 0.5 * (1 + erf((x - mk) / np.sqrt(2 * vk)))
               for w, mk, vk in zip(ws, m, v))

def sample_p0(n, rng, ws=ws, mus=mus, v0s=v0s):
    k = rng.choice(len(ws), size=n, p=ws)   # component of each draw
    return mus[k] + np.sqrt(v0s[k]) * rng.standard_normal(n)
```

閉形式をコードに実装したので、PDE に直接代入して確かめられる。グリッド上で $t = 0.25$、1、3 において両辺を評価し、∂_t は時間に関する中心差分で、x の導関数は `np.gradient` で計算する。有限差分（finite difference）では厳密解に対しても残差（residual）が残るため、次のセルでは 2 つ目を 2 倍細かくした 2 種類のグリッドを用いる。これらの 2 次精度ステンシルの打ち切り誤差（truncation error）は 4 分の 1 に減少するはずである。

```
def fp_residual(t, points, eps=1e-5):
    """Largest Fokker--Planck residual on a grid of `points`, and max|dp/dt|."""
    xs = np.linspace(-5.0, 5.0, points)
    dx = xs[1] - xs[0]
    dpdt = (p_t(xs, t + eps) - p_t(xs, t - eps)) / (2 * eps)
    p = p_t(xs, t)
    rhs = (np.gradient(theta * xs * p, dx)                # -d/dx(f p)
           + 0.5 * g**2 * np.gradient(np.gradient(p, dx), dx))
    res = (dpdt - rhs)[5:-5]            # trim the one-sided edge stencils
    return np.abs(res).max(), np.abs(dpdt).max()

for t in (0.25, 1.0, 3.0):
    (r1, dpdt), (r2, _) = fp_residual(t, 4001), fp_residual(t, 8001)
    print(f't={t}: max|dp/dt| = {dpdt:.1e}; max|residual| at 4001, 8001 points: '
          f'{r1:.1e}, {r2:.1e} (ratio {r1 / r2:.1f})')
```

```
t=0.25: max|dp/dt| = 4.4e-01; max|residual| at 4001, 8001 points: 4.4e-06, 1.1e-06 (ratio 4.0)
t=1.0: max|dp/dt| = 1.7e-01; max|residual| at 4001, 8001 points: 6.3e-07, 1.6e-07 (ratio 4.0)
t=3.0: max|dp/dt| = 3.2e-03; max|residual| at 4001, 8001 points: 1.2e-06, 3.0e-07 (ratio 4.0)
```

残差は各時刻で $\partial_t p$ より 3〜5 桁小さく（4×10^{-6} 対 0.44 が $t = 0.25$ での値であり、$t = 3$ では密度がほとんど動かないものの 10^{-6} 対 3×10^{-3} となる）、3 つの時刻すべてにおいて細かいグリッドで収束次数どおり 4 分の 1 に減少している。残っているのは打ち切り誤差にほかならない。理論上の議論にはこの計算は必要なかったが、以降の全実験が依拠するコードに誤りがないことを保証する検算となっている。

38.5.3 スコア関数

フォッカー・プランク方程式の平滑化項は、実は姿を変えた輸送であることが後にわかる。その輸送の速度はただ 1 つの場、すなわち $\log p_t$ の勾配から作られる。恒等式 $\nabla p = p\,\nabla \log p$ によって、この勾配がラプラシアンの内部に現れる（38.5.4 節）。そこから先のすべての構成にこの場が必要となるため、混合分布を主な例としてまずこの場を調べよう。

38.5.3.1 定義と幾何

問いは、点 $\mathbf{x}$ からどの方向へ進めば、より確率の高い点へ最も速く向かえるかということである。密度 p_t の**スコア**（score）がその答えを与える。

$$\mathbf{s}_t(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}). \tag{38.55}$$

スコアはデータ空間上のベクトル場であり、対数密度の上り方向を指す。この勾配は入力 $\mathbf{x}$ に関するものであり、モデルのパラメータに関するものではない。37.5.1 節ではこのデータに関するスコアを導入し、最尤法（maximum likelihood）におけるパラメータに関するスコアと対比した。

$\mathbf{s}_t$ の長さは $\log p_t$ の増加率 $\|\nabla p_t\|/p_t$ である。したがって、スコアが大きくなるのは密度が大きい場所ではなく、裾やモード間の谷のように、密度が小さいにもかかわらず変化している場所である。スコアは p_t の臨界点（critical point）、すなわちモード、極小点、鞍点（saddle point）においてゼロになり、その導関数によってこれらを見分けられる。

まずガウス分布を考えよう。$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu, v\mathbf{I})$ に対して対数密度は $-\|\mathbf{x}-\mu\|^2/(2v) + \text{const}$ であるから、次が成り立つ。

$$\nabla \log p(\mathbf{x}) = -\frac{\mathbf{x} - \mu}{v}. \tag{38.56}$$

これは 37.5.1 節で見た線形バネであり、平均へ向かって引き戻し、分散が小さいほど硬くなる。$\mu = 0$、$v = 0.25$、$x = 1$ のときその値は -4 となり、単位分散のときの 4 倍の引き戻し力になる。一般の共分散（covariance）に対しては、このバネは $-\Sigma^{-1}(\mathbf{x}-\mu)$ となる。Figure 38.18 に、2 次元における 2 モード密度のスコア場を示す。

38.5.3.2 混合ガウス分布のスコア

ここで扱うデータ密度は混合分布であるため、数値評価も解釈もできる形でそのスコアを求めておく必要がある。$p(x) = \sum_k w_k\, N_k(x)$（重み w_k、成分 $N_k(x) = \mathcal{N}(x; \mu_k, v_k)$）に対し、$\log p$ を直接微分すると次を得る。

$$\nabla \log p(x) = \frac{\sum_k w_k \nabla N_k(x)}{\sum_j w_j N_j(x)} = \sum_k r_k(x) \left(-\frac{x - \mu_k}{v_k} \right), \qquad r_k(x) = \frac{w_k N_k(x)}{\sum_j w_j N_j(x)}. \tag{38.57}$$

混合分布のスコアは、各成分のバネの重み付き平均である。重み $r_k(x)$ は成分 k の**負担率**（responsibility）、すなわち位置 x の点がその成分から生じた事後確率（posterior probability）であり、37.5.1 節と同様である。等しい重み、平均 $\pm\mu$、共通の分散 v_0 を持つ 2 成分の場合、負担率は x のロジスティック関数に帰着し、スコアは次の閉形式を持つ。

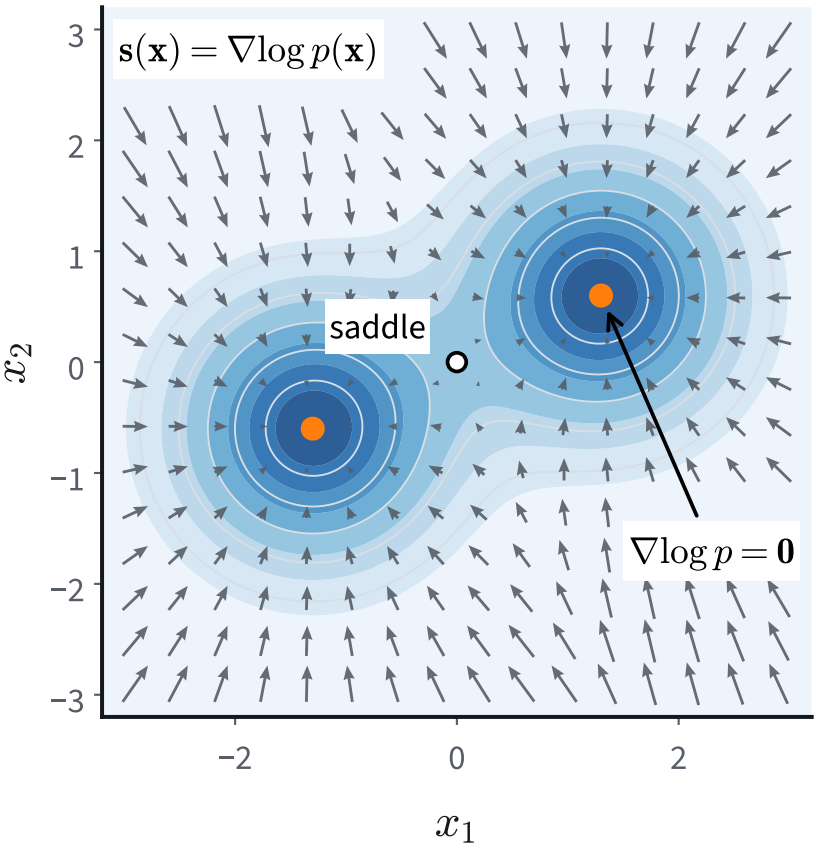


Figure 38.18: 2 成分混合ガウス分布のスコア $\mathbf{s}(\mathbf{x}) = \nabla \log p(\mathbf{x})$ を密度の等高線上に重ねて描いた図。矢印は上り方向を指し、臨界点でゼロになる。オレンジ色の点は 2 つのモードを表し、両者のあいだの対称点も曲率の異なるもう 1 つの零点である。矢印は真の長さ $\|\nabla p\|/p$ を持つが、限りなく増大する遠い裾では上限を 4 で打ち切っており、モードや鞍点の近くでは短くなる。

$$s_0(x) = \frac{\mu \tanh(\mu x / v_0) - x}{v_0}. \tag{38.58}$$

各モードの近くでは $\tanh \to \pm 1$ となり、スコアは局所的なバネ $-(x \mp \mu)/v_0$ に一致する。通し例の混合分布（$\mu = 2$、$v_0 = 0.25$）では $s_0(x) = 8\tanh(8x) - 4x$ となる。$x = 0$ ではスコアがゼロになり、これが 3 つ目の臨界点であって、ゼロから次の傾きで立ち上がる。

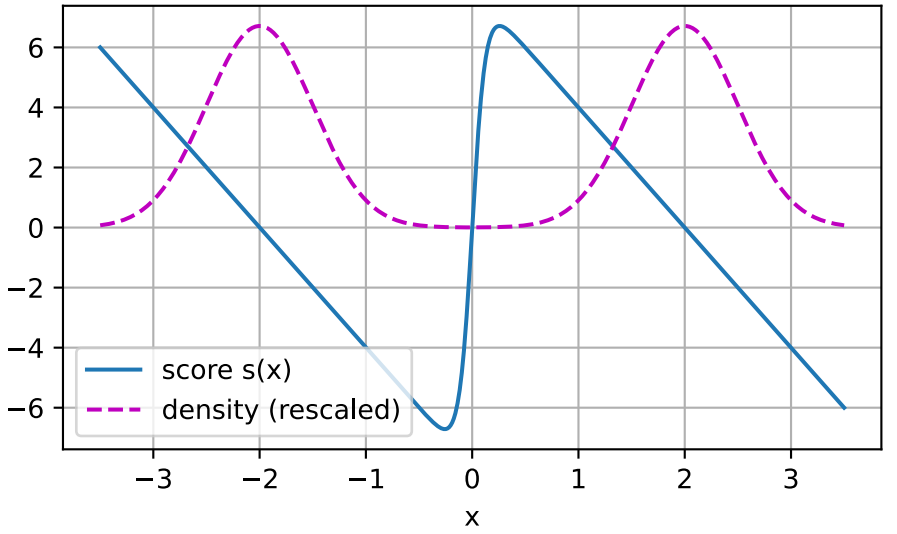
$$s_0'(0) = \frac{\mu^2 - v_0}{v_0^2} = 60.$$

傾きが正であることは、$x = 0$ が密度の極小点であり、上り方向のフロー $\dot{x} = s_0(x)$ の反発点（repeller）であることを意味する。混合分布が 2 つのモードを持つのは $\mu^2 > v_0$ のとき、かつそのときに限り、$\mu^2 \le v_0$ では 2 つのモードは 1 つに合流する（演習問題 4）。

コードでは一般式 Equation 38.57 を用い、負担率の計算には数値的に安定なソフトマックスを使う。後の実験であらゆる時刻における p_t のスコアが必要になるからである。tanh の式、零点、および傾き 60 は実装の検算として役立つ。

```
def score_t(x, t, ws=ws, mus=mus, v0s=v0s):
    """Exact score of p_t: responsibility-weighted component springs."""
    m, v = mixture_params(t, mus, v0s)
    logw = np.stack([np.log(w) - 0.5 * np.log(2 * np.pi * vk)
                     - (x - mk)**2 / (2 * vk) for w, mk, vk in zip(ws, m, v)])
    r = np.exp(logw - logw.max(axis=0))
    r /= r.sum(axis=0)                  # responsibilities, a stable softmax
    return sum(rk * -(x - mk) / vk for rk, mk, vk in zip(r, m, v))

gs = np.linspace(-3.5, 3.5, 1400)
mu, v0 = mus[1], v0s[1]
s = score_t(gs, 0.0)
s_tanh = (mu * np.tanh(mu * gs / v0) - gs) / v0
print(f'max|responsibility form - tanh form| = {np.abs(s - s_tanh).max():.1e}')
mid = (gs[:-1] + gs[1:]) / 2
print('score changes sign near', np.round(mid[s[:-1] * s[1:] < 0], 2) + 0.0)
delta = 1e-5                            # centered difference at x = 0
print(f'slope at 0: {(score_t(delta, 0.0) - score_t(-delta, 0.0)) / (2 * delta):.2f}, '
      f'formula (mu^2 - v0) / v0^2 = {(mu**2 - v0) / v0**2:.2f}')
dens = p_t(gs, 0.0)
d2l.plot(gs, [s, dens * np.abs(s).max() / dens.max()], 'x', '',
         legend=['score s(x)', 'density (rescaled)'], figsize=(5.5, 3))
```

```
max|responsibility form - tanh form| = 7.3e-15
score changes sign near [-2.  0.  2.]
slope at 0: 60.00, formula (mu^2 - v0) / v0^2 = 60.00
```



検算はすべて通る。2 つの式は機械精度で一致し、零点と傾きも計算どおりの位置にある。図から新たに読み取れるのは、スコアが大きくなる場所である。すなわち、密度がほぼゼロでありながらその対数がどちらのモードへ向かっても急勾配で登っていくモード間の谷と、分布の裾である。

38.5.4 連続の式と確率フロー ODE

フォッカー・プランク方程式はランダムに動く粒子を記述する。では、決定論的に動く粒子を支配するのはどの法則だろうか。ノイズを速度として書き直せば、これら 2 つの法則は一致することがわかる。

38.5.4.1 確率の保存

いったんノイズを忘れよう。すべての粒子が速度場 $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}, t)$ に従うとする。このとき粒子群の密度 q_t はどの PDE に従うだろうか。確率は生成も消滅もしないため、固定された領域 Ω の内部にある質量は、その境界を横切って出入りすることによってのみ変化しうる。粒子が境界の微小面を横切る速さは、密度と速度の積になる。この流束密度を**確率流**（probability current）と呼び、$\mathbf{j} = q\,\mathbf{u}$ と書く。これは各点に置いた面を単位面積あたりに横切る確率の流量を表す。領域における質量収支は次のようになる。

$$\frac{d}{dt}\int_{\Omega} q_t \, d\mathbf{x} = -\oint_{\partial\Omega} \mathbf{j}\cdot\mathbf{n}\, dS = -\int_{\Omega} \nabla\cdot(q_t\mathbf{u})\, d\mathbf{x}, \tag{38.59}$$

ここで、発散定理 Equation 38.48 によって境界を通る流束を体積積分へと書き換えた。領域内部の質量は、正味の流出量にマイナスをつけた速さで変化する。Ω は任意であるから被積分関数が各点で一致し、**連続の式**（continuity equation）が得られる。

$$\partial_t q_t + \nabla\cdot(q_t\,\mathbf{u}) = 0. \tag{38.60}$$

ある点における密度は、その点で確率流が発散する速さで減少する。1 次元では、この収支は $\partial_t \int_a^b q\,dx = j(a) - j(b)$、すなわち a から流入する量から b へ流出する量を引いたものになる。この様子を Figure 38.19 に描く。

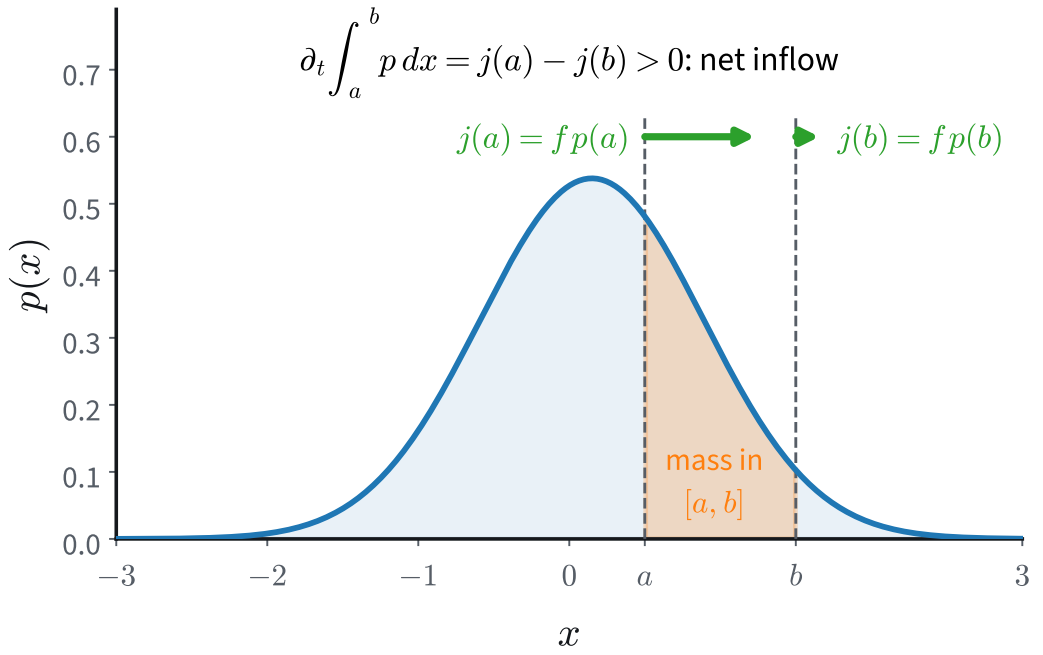


Figure 38.19: 流束の収支としての連続の式。ここでは速度を定数ドリフト f とし、密度を p と書いている。密度 $p(x)$（青）が f によって運ばれ、確率流 $j(x) = f\,p(x)$ を生じる。山の右側斜面にある a から b までの区間では、内部の質量（オレンジ）が正味の流入量 $j(a) - j(b)$ の速さで増加する。a からの流入が b からの流出を上回るのは、a における密度の方が高いからである。

この収支から、定義域の端で何が起こるかもわかる。$\mathbb{R}^d$ 全体で考える場合、確率流は無限遠で減衰すると仮定する。フォッカー・プランク方程式においては、これは $\mathbf{f}p_t$ と ∇p_t が減衰することを意味し、OU 過程のドリフトが線形に増大してもガウス分布の裾があればこの条件は保証される。このとき Equation 38.60 を $\mathbb{R}^d$ 全体で積分すると $\frac{d}{dt}\int q_t = 0$ となり、全確率が保存される。有界領域上では、**反射**（reflecting）境界が $\mathbf{j}\cdot\mathbf{n} = 0$ を課すためやはり質量が保存される一方、**吸収**（absorbing）境界は $q = 0$ を課すため質量が外へ流出しうる。

1 次元の 2 つの簡単な例で、この方程式の働きを見てみよう。平行移動 $u = c$ の場合、方程式は $\partial_t q + c\,\partial_x q = 0$ となり、解は $q_t(x) = q_0(x - ct)$ である。密度の形状は変わらずにそのまま横滑りする。縮小 $u = -x$ の場合、すべての粒子は $x(t) = x(0)e^{-t}$ に従って動き、粒子群は倍率 e^{-t} で縮み、各区間の質量が保たれるため各軌跡（trajectory）に沿って運ばれる密度は e^{t} 倍に上昇する。

この縮小の例は、運動する粒子から見た密度に対する一般的な規則を示唆している。積の微分法則を用いて発散を展開すると、次のようになる。

$$\nabla\cdot(q\,\mathbf{u}) = \mathbf{u}\cdot\nabla q + q\,\nabla\cdot\mathbf{u},$$

すると連続の式は $\partial_t q + \mathbf{u}\cdot\nabla q = -q\,\nabla\cdot\mathbf{u}$ となる。連鎖律により、左辺は q の、軌跡 $\mathbf{x}(t)$（$\dot{\mathbf{x}} = \mathbf{u}$ の解）に沿った時間変化率である。両辺を q で割れば次を得る。

$$\frac{d}{dt}\log q_t(\mathbf{x}(t)) = -\nabla\cdot\mathbf{u}(\mathbf{x}(t), t). \tag{38.61}$$

軌跡に沿って、対数密度は局所的な膨張率に等しい速さで減少する。先ほどの縮小の例では $\nabla\cdot u = -1$ であり、上で計算したとおり対数密度は速さ 1 で増加する。$\dot{\mathbf{x}} = \mathbf{u}$ の軌跡は、連続の式の**特性曲線**（characteristic）と呼ばれる。この規則は 38.3.3 節で軌跡に沿って証明した瞬間的な変数変換（change of variables）の公式にほかならない。

38.5.4.2 輸送としての拡散

フォッカー・プランク方程式 Equation 38.52 はこのままでは連続の式ではない。輸送項 $-\nabla\cdot(\mathbf{f}p)$ の隣にラプラシアンがあるからである。しかし連鎖律を使えば、このラプラシアンが輸送項へと変わる。わずか 1 行の計算だが、本節で最も重要な帰結をもたらす式変形である。$p > 0$ であるすべての点で、次が成り立つ。

$$\nabla p = p\,\nabla\log p, \tag{38.62}$$

これは $\log p$ の導関数を逆向きに読んだ式にほかならない。したがって、次が得られる。

$$\tfrac{1}{2}g^2\,\Delta p = \tfrac{1}{2}g^2\,\nabla\cdot(\nabla p) = \nabla\cdot\big(p\cdot\tfrac{1}{2}g^2\nabla\log p\big). \tag{38.63}$$

拡散項は、密度と $+\frac{1}{2}g^2\nabla\log p$ の積である流束の発散となっている。これを連続の式の左辺へ移項すればマイナス符号がつき、速度 $-\frac{1}{2}g^2\nabla\log p$ として現れる。

命題（フォッカー・プランク方程式は連続の式である） $p_t > 0$ **であるすべての点で、フォッカー・プランク方程式 Equation 38.52 は次と同値である。**

$$\partial_t p_t + \nabla\cdot(p_t\,\mathbf{u}_t) = 0, \qquad \mathbf{u}_t(\mathbf{x}) = \mathbf{f}(\mathbf{x}, t) - \tfrac{1}{2}g(t)^2\,\nabla\log p_t(\mathbf{x}). \tag{38.64}$$

証明 Equation 38.63 を Equation 38.52 に代入し、2 つの発散を 1 つにまとめる。

$$\begin{aligned}\partial_t p &= -\nabla\cdot(\mathbf{f}p) + \nabla\cdot\big(p\cdot\tfrac{1}{2}g^2\nabla\log p\big)\\ &= -\nabla\cdot\big(p\,[\mathbf{f} - \tfrac{1}{2}g^2\nabla\log p]\big). \qquad \blacksquare\end{aligned}$$

密度のレベルで見ると、SDE はあたかも各粒子が速度 $\mathbf{u}_t$、すなわちドリフトに対数密度勾配を下る向きの押し出しを加えた速度に従っているかのように確率を動かす。対応する確率流は次のとおりである。

$$\mathbf{j} = \mathbf{f}\,p_t - \tfrac{1}{2}g^2\,\nabla p_t = p_t\,\mathbf{u}_t. \tag{38.65}$$

この速度がどのような姿をしているかは、熱方程式を見ればよくわかる。1 次元で $\mathbf{f}=0$、$p_t = \mathcal{N}(0, v_t)$（ただし $v_t = v_0 + g^2 t$）としよう。スコアは $-x/v_t$ であるから、速度は次のようになる。

$$u_t(x) = -\tfrac{1}{2}g^2\Big(-\frac{x}{v_t}\Big) = \frac{g^2}{2v_t}\,x.$$

速度は中心から外向きを指し、中心からの距離に比例し、密度が広がるにつれて弱まる。ODE $\dot{x} = g^2 x/(2v_t)$ は変数分離でき、$\frac{d}{dt}\log x = \frac{g^2}{2v_t} = \frac{1}{2}\frac{d}{dt}\log v_t$ より次を得る。

$$x(t) = x(0)\sqrt{v_t/v_0}. \tag{38.66}$$

すべての粒子が同じ倍率で引き伸ばされ、ガウス分布を $\sqrt{v_t/v_0}$ 倍に引き伸ばせば分散は v_t となって、ブラウン運動が生み出す分散と一致する。$v_0 = g = 1$、$t = 3$ のとき、すべての粒子は中心からの距離が 2 倍になる。一方、$x = 1$ から出発したブラウン粒子は、1 に標準偏差 $\sqrt{3} \approx 1.73$ の乱数を足した位置に着地する。2 つの過程はあらゆる時刻でヒストグラムが一致するが、個々の粒子の動かし方は異なる（Figure 38.20）。

この構成には限界もある。$v_0 \to 0$ とすると、小さい時刻における速度は $g^2 x/(2g^2 t) = x/(2t)$ となり、$t \to 0$ で非有界となる。リプシッツ連続なフローは 1 点を 1 点にしか写せないため、決定論的なフローが点質量をガウス分布へと広げられるのは、出発時点で爆発する場を介する場合に限られる。

次のセルでこの例を実行する。20,000 個の点を $\mathcal{N}(0,1)$ から 1 回抽出し、そこから $t=3$ におけるブラウン粒子群（独立な増分 $\mathcal{N}(0,3)$ を加算）と、$\dot{x} = x/(2(1+t))$ をホイン法（Heun's method、38.2 節）で積分したフロー粒子群を作る。そして両者の標準偏差と、フロー粒子群が初期サンプルを 2 倍したものからどれだけずれているかを表示する。

```
def heat_velocity(x, t, v0=1.0, g=1.0):  # u_t = -(1/2) g^2 score = g^2 x / (2 v_t)
    return g**2 * x / (2 * (v0 + g**2 * t))

n, T, steps = 20_000, 3.0, 600
dt = T / steps
z = rng.standard_normal(n)              # p_0 = N(0, 1)
x_bm = z + np.sqrt(T) * rng.standard_normal(n)   # Brownian motion to time T
x_ode = z.copy()
for k in range(steps):                  # Heun steps for the probability flow
    k1 = heat_velocity(x_ode, k * dt)
```

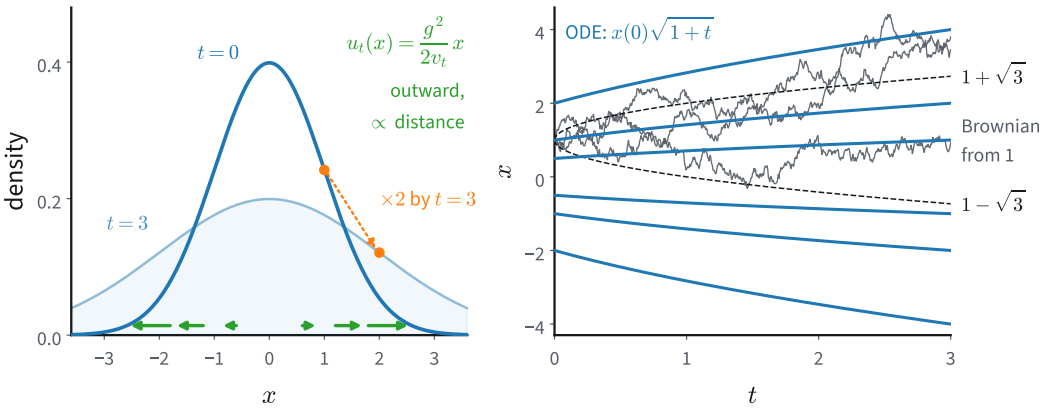


Figure 38.20: 熱方程式 $\partial_t p = \frac{1}{2}\partial_{xx} p$ を初期分布 $p_0 = \mathcal{N}(0,1)$ からフローとして解いた図。(a) $t=0$ と $t=3$ における密度。軸上の矢印は速度 $u_0(x) = x/2$ を表し、外向きで距離に比例する。$x=1$ にある粒子は $x=2$ へと $t=3$ までに移動し、そこでの密度は半分になる。(b) 滑らかで互いに交差しないフローの軌跡 $x(0)\sqrt{1+t}$ と、$x=1$ から出発する 3 本のブラウン経路およびその 1 標準偏差の包絡線 $1 \pm \sqrt{t}$ の比較。

```
    k2 = heat_velocity(x_ode + dt * k1, (k + 1) * dt)
    x_ode += 0.5 * dt * (k1 + k2)
print(f'std at t={T}: Brownian {x_bm.std():.3f}, flow {x_ode.std():.3f}, '
      f'exact {np.sqrt(1 + T):.3f}')
print(f'max|x_ode - 2 z| = {np.abs(x_ode - 2 * z).max():.1e}')
```

```
std at t=3.0: Brownian 1.995, flow 1.994, exact 2.000
max|x_ode - 2 z| = 5.8e-06
```

どちらの粒子群も標準偏差は 2.00（オーダー 0.01 の標本誤差の範囲内）となり、フロー写像（flow map）は $x \mapsto 2x$ に 10^{-5} 未満のホイン法の誤差の範囲で一致する。ランダムな揺さぶりと決定論的な引き伸ばしが、同じヒストグラムを生み出したのである。

38.5.4.3 確率フロー ODE

熱方程式は 1 つの例にすぎない。上の命題によれば、本節で扱う形のあらゆる SDE に対して同じ構成が成り立つ。すなわち、速度 $\mathbf{u}_t$ に従うことで次の ODE が得られる。

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}_t(\mathbf{x}) = \mathbf{f}(\mathbf{x},t) - \tfrac{1}{2} g(t)^2 \nabla \log p_t(\mathbf{x}), \tag{38.67}$$

これが**確率フロー ODE**（probability-flow ODE）である (Y. Song, Sohl-Dickstein, et al. 2021)。次の命題は、生成モデルが本節から受け取る 2 つの主要結果のうちの 1 つ目である。

命題（決定論的な双子） p_t を $d\mathbf{X} = \mathbf{f}\,dt + g\,d\mathbf{W}$（$\mathbf{X}_0 \sim p_0$）の周辺密度とし、滑らかかつ正であってフォッカー・プランク方程式を満たすとする。また、$\mathbf{u}_t$ は t について連続、$\mathbf{x}$ について連続微分可能であり、かつ $\mathbf{x}$ について t に依存しない定数でリプシッツ連続であるとする。$\mathbf{x}(t)$ を Equation 38.67 の $\mathbf{x}(0) \sim p_0$ に対する解とする。このとき、$\mathbf{x}(t) \sim p_t$ がすべての t で成り立つ。すなわち、この ODE と SDE はすべての時間周辺分布を共有する。

証明 38.1.1.3 節の存在と一意性の定理により、t に依存しない定数でリプシッツ連続な場は、順方向にも逆方向にも各点を通る軌跡をただ 1 つ持つため、その時刻 t の写像は $\mathbb{R}^d$ の全単射となる。ODE の状態の密度を q_t とする。ここに循環論法はない。場 $\mathbf{u}_t$ は p から作られており q から作られたものではないため既知の速度場であり、q_t は連続の式 $\partial_t q + \nabla \cdot (q\mathbf{u}_t) = 0$ に従う。前の命題により p_t も同じ方程式を満たし、両者はともに初期分布 $q_0 = p_0$ から出発する。対数密度の規則 Equation 38.61 の導出はこの PDE しか使っていなかったため、各特性曲線に沿って $\log q_t$ と $\log p_t$ は同じ変化率 $-\nabla \cdot \mathbf{u}_t$ で変化し、出発時点では等しい。特性曲線はすべての点に到達するため、いたるところで $q_t = p_t$ が成り立つ。■

このリプシッツ連続の仮定は、OU 過程のドリフトや、正の成分分散を持つガウス分布および混合ガウス分布のスコア（その傾きは t が $[0, T]$ にある限り有界である）に対して成り立つ。

周辺分布は一致するが、経路は一致しない。SDE の経路はギザギザで他の経路と交差する。一方、ODE の軌跡は滑らかであり、解の一意性によって 2 本の軌跡が同じ時刻に同じ点で出会うことは決してない（Figure 38.20 の (b)）。

この ODE からは尤度も得られる。対数密度の規則 Equation 38.61 で $q_t = p_t$ とおき、軌跡に沿って 0 から T まで積分すると次を得る。

$$\log p_0(\mathbf{x}(0)) = \log p_T(\mathbf{x}(T)) + \int_0^T \nabla \cdot \mathbf{u}_t(\mathbf{x}(t))\, dt. \tag{38.68}$$

データ点の対数密度は、ほぼガウス分布に従い容易に評価できる時刻 T での像の対数密度に、途中で累積した膨張率を足したものになる。厳密なスコアの代わりに学習されたスコアを用いれば、これはソルバの誤差を除いて、その場が定義する**モデル**の対数密度となる (Y. Song, Sohl-Dickstein, et al. 2021)。拡散モデルの尤度を報告する際に用いられる 2 つの数値のうちの 1 つがこれであり、もう 1 つは変分境界（variational bound、17.6 節）である。

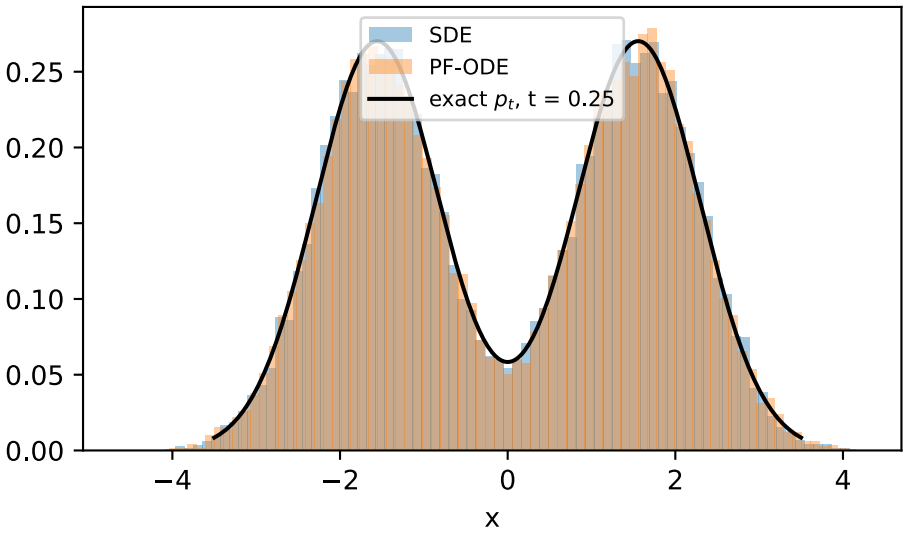
38.5.4.4 SDE と ODE の周辺分布の比較

熱フローは単一のガウス分布の引き伸ばしにすぎなかった。混合分布の上では、この命題にはさらに大きな仕事がある。密度が 2 モードから 1 モードへと形状を変えるなかでも、ギザギザな過程と滑らかな過程はあらゆる時刻で同じ密度を与えなければならない。p_0 からの 1 組のサンプルから、20,000 個の粒子からなる 2 つの粒子群を出発させる。一方はオイラー・丸山法のステップで SDE に従い、もう一方は厳密なスコアを用いたホイン法のステップで確率フロー ODE に従う。物差しとして、各粒子群と p_t の厳密な CDF との 1 標本コルモゴロフ・スミルノフ距離（Kolmogorov–Smirnov distance）、すなわち経験 CDF と厳密な CDF の最大乖離幅を計算する。サイズ n の厳密なサンプルでは、この距離が $1.36/\sqrt{n}$ を超える確率はわずか 5% である。出力を見る前に、スナップショット間でどちらの粒子群の距離がより大きく変動するか予想してみてほしい。

```
def pf_velocity(x, t, score=score_t, theta=theta, g=g):
    return -theta * x - 0.5 * g**2 * score(x, t)   # u_t = f - (1/2) g^2 s_t

def ks_exact(sample, t, **mixture):
    """One-sample Kolmogorov--Smirnov distance to the exact p_t."""
    xs_sorted, n = np.sort(sample), len(sample)
    F = mixture_cdf(xs_sorted, t, **mixture)
    return max((np.arange(1, n + 1) / n - F).max(), (F - np.arange(n) / n).max())

n, T, steps = 20_000, 3.0, 600
```

```
    dt = T / steps
    x_sde = sample_p0(n, rng)
    x_ode = x_sde.copy()                    # both clouds start from one sample
    snaps = {}
    for k in range(steps):
        t = k * dt
        x_sde += -theta * x_sde * dt + g * np.sqrt(dt) * rng.standard_normal(n)
        k1 = pf_velocity(x_ode, t)          # Heun steps for the ODE
        k2 = pf_velocity(x_ode + dt * k1, t + dt)
        x_ode += 0.5 * dt * (k1 + k2)
        if k + 1 in (50, 200, 600):         # t = 0.25, 1, 3
            snaps[(k + 1) * dt] = (x_sde.copy(), x_ode.copy())
    print(f'95% band for n = {n}: {1.36 / np.sqrt(n):.4f}')
    for t, (a, b) in snaps.items():
        print(f't={t:.2f}: KS(SDE) = {ks_exact(a, t):.4f}, KS(ODE) = {ks_exact(b, t):.4f}')
    gx = np.linspace(-3.5, 3.5, 400)
    d2l.set_figsize((5.5, 3))
    d2l.plt.hist(snaps[0.25][0], bins=80, density=True, alpha=0.4, label='SDE')
    d2l.plt.hist(snaps[0.25][1], bins=80, density=True, alpha=0.4, label='PF-ODE')
    d2l.plt.plot(gx, p_t(gx, 0.25), 'k', label='exact $p_t$, t = 0.25')
    d2l.plt.xlabel('x'); d2l.plt.legend(loc='upper center', fontsize=8);
```

```
95% band for n = 20000: 0.0096
t=0.25: KS(SDE) = 0.0051, KS(ODE) = 0.0074
t=1.00: KS(SDE) = 0.0088, KS(ODE) = 0.0074
t=3.00: KS(SDE) = 0.0044, KS(ODE) = 0.0074
```



ほとんどの実行で、両方の粒子群とも 3 つの時刻すべてにおいて 0.0096 の帯域内に収まり、距離は約 0.003 から 0.01 のあいだになる。帯域をわずかに超える値が出ても異常ではない。この帯域は 5% のしきい値であり、厳密なサンプルでも 20 回に 1 回はこれを超えるからである。$t = 0.25$ では周辺分布はまだ二峰性（bimodal）であり、両方のヒストグラムがその 2 つのモードを再現している。$t = 1$ までには単峰になり、モードが合流する時刻は 0.78 である（演習問題 4）。

さて、予想の答え合わせをしよう。ODE の距離は、表示されているすべての桁にわたって 3 つの

時刻すべてで同じ値になっている。p_0 を p_t へと運ぶ狭義単調増加写像はサンプルの経験 CDF もそのまま運ぶため、ODE の粒子群は初期サンプルが持っていた距離をそのまま保ち、そこに加わるのはソルバの誤差だけである。一方、SDE の粒子群はステップごとにノイズによって引き直されるため、距離が変動する。

38.5.5 時間反転

順方向の SDE はデータをノイズへと変える。生成に必要なのは、ノイズからデータへとこの過程を逆再生することである。これは不可能に聞こえるかもしれない。平滑化によって失われた細部はどんな操作でも復元できないため、任意の終端密度から熱方程式を逆向きに解くことはできないからである。しかし、ここで熱方程式を逆向きに解くわけではない。順方向の分布族 p_t はすでに与えられており、それを逆順にたどる過程を求めているだけである（Figure 38.21）。そのドリフトには、失われた細部を保持している $\nabla \log p_t$ が含まれることになる。確率フロー ODE を逆向きに積分したものは、まさにそのような過程の 1 つである。SDE の場合の答えはそれほど自明ではない。逆向きの過程も再び拡散過程となり、そのドリフトは ODE の 2 倍のスコア補正を持つ。

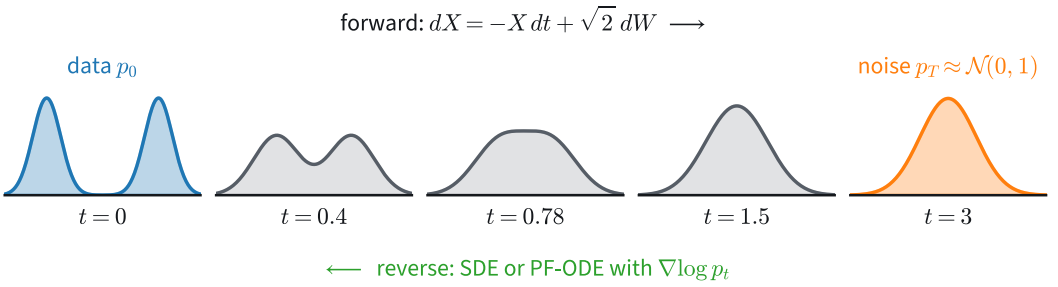


Figure 38.21: 通し例の混合分布 $p_0 = \frac{1}{2}\mathcal{N}(-2, 0.25) + \frac{1}{2}\mathcal{N}(2, 0.25)$ の、$\theta = 1$、$g = \sqrt{2}$ の OU 過程のもとでの順方向の周辺分布（各時刻で厳密）。本節の逆過程は、同じ周辺分布を右から左へとたどる。$t \approx 0.78$ における平坦な頂上は、2 つのモードが合流する地点である。

38.5.5.1 無限小ステップにおけるベイズの定理

微小な 1 ステップを逆転させるため、ベイズの定理を用いて粒子がどこから来たかを問う。短い時間幅 τ にわたる順方向のステップは、オイラー・丸山法のガウスカーネルである（38.4.4.1 節）。

$$p(\mathbf{x}_{t+\tau} \mid \mathbf{x}_t) \approx \mathcal{N}\big(\mathbf{x}_{t+\tau};\ \mathbf{x}_t + \mathbf{f}(\mathbf{x}_t, t)\,\tau,\ g^2 \tau\, \mathbf{I}\big).$$

ここで終点 $\mathbf{x}_{t+\tau} = \mathbf{y}$ で条件付けよう。ベイズの定理により、確からしい出発点の分布は、順方向カーネルを時刻 t の周辺分布で重み付けしたものとして与えられる。

$$p(\mathbf{x}_t = \mathbf{x} \mid \mathbf{x}_{t+\tau} = \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{x})\ p_t(\mathbf{x}).$$

ある出発点が確からしいのは、そこから $\mathbf{y}$ に到達しえた場合、かつそもそもその出発点自体が確からしかった場合である。対数をとり、終点のまわりで $\log p_t(\mathbf{x})$ を展開すると $\log p_t(\mathbf{x}) \approx \log p_t(\mathbf{y}) + (\mathbf{x} - \mathbf{y}) \cdot \nabla \log p_t(\mathbf{y})$ となる。ステップ $\mathbf{x} - \mathbf{y}$ の大きさは $\sqrt{\tau}$ であるため、無視した 2 次の項は τ のオーダーであり、事後分布（posterior）にはより高次の項としてしか影響しない。同じ次数までで考えれば、$\mathbf{f}$ は $\mathbf{y}$ で評価してよい（$\mathbf{x}$ で評価する場合との差 $\mathbf{f}(\mathbf{x}, t) - \mathbf{f}(\mathbf{y}, t)$

は大きさが $\sqrt{\tau}$ であり、そこに τ が掛かるからである）。ここで、ガウス分布の指数部に 1 次の項を加えると、平方完成により分散は変わらないまま平均だけが「分散 × 傾き」の分だけシフトする。いま指数部は $-\|\mathbf{x}-(\mathbf{y}-\mathbf{f}(\mathbf{y},t)\,\tau)\|^2/(2g^2\tau)$、分散は $g^2\tau$、傾きは $\nabla \log p_t(\mathbf{y})$ であるから、次を得る。

$$p(\mathbf{x}\mid\mathbf{y}) \approx \mathcal{N}\big(\mathbf{x};\ \mathbf{y}-[\mathbf{f}(\mathbf{y},t)-g^2\nabla\log p_t(\mathbf{y})]\,\tau,\ g^2\tau\,\mathbf{I}\big). \tag{38.69}$$

逆向きのステップは同じ分散 $g^2\tau$ を持つガウス分布に従う揺さぶりであり、そのドリフトは順方向のドリフトに g^2 とスコアの積を丸ごと加えた補正を受けたものになる。周辺分布 p_t は事前分布（prior）として働き、逆向きのステップをデータ密度があった場所へと引き寄せる。

38.5.5.2 逆時間 SDE

この 1 ステップの計算は発見的な導出である。1 次までで展開し、近似カーネルから出発しているからである。しかし Anderson (1982) は、これが過程全体に対して正しい答えを与えることを証明した。この定理を証明なしに述べよう。拡散過程 $d\mathbf{X}=\mathbf{f}\,dt+g\,d\mathbf{W}$ の時間反転は、次の**逆時間 SDE**（reverse-time SDE）である。

$$d\mathbf{X} = \big[\mathbf{f}(\mathbf{X},t)-g(t)^2\,\nabla_{\mathbf{x}}\log p_t(\mathbf{X})\big]\,dt + g(t)\,d\bar{\mathbf{W}}, \tag{38.70}$$

これを $t=T$ から $t=0$ まで逆向きに積分する。ここで $\bar{\mathbf{W}}$ は時間を逆向きに進むブラウン運動であり、シミュレーションにおいては逆向きの各ステップで独立な新しいガウス増分を加えることを意味する。$\mathbf{X}_T\sim p_T$ から出発すると、この過程はすべての中間時刻で周辺分布 p_t を持ち、$t=0$ においてデータ密度を与える。この定理は逆転された過程全体を特定するものだが、以下ではサンプリングに必要な周辺分布に関する主張のみを証明する。

ステップとして読めば、この式は前項の計算そのものである。$t+\tau$ から t へ戻ることは $dt=-\tau$ のステップをとることにあたり、粒子は $-[\mathbf{f}-g^2\nabla\log p_t]\,\tau$ に分散 $g^2\tau$ の揺さぶりを加えた分だけ動く。これはガウス分布 Equation 38.69 にほかならない。実際の計算には、順方向に進む時計の方が扱いやすい。**逆向きの時計**（reversed clock）$\bar{t}=T-t$ を用いれば、同じ過程はドリフト $-\mathbf{f}+g^2\,\nabla\log p_t$ とノイズ g を持つ通常の SDE になる。時間の向きが変わるとドリフトは符号を変えるが、ノイズは向きを問わない。

最小の例は手計算で確かめられる。定常状態にある OU 過程、すなわち $p_t=\mathcal{N}(0,1)$ がすべての t で成り立つ場合（$\theta=1$、$g=\sqrt{2}$、スコア $-x$）を考えよう。確率フロー速度は $-x-\tfrac{1}{2}\cdot 2\cdot(-x)=0$ となり、ODE の粒子は動かず、粒子群は $\mathcal{N}(0,1)$ のまま保たれる。逆向きの時計では、逆 SDE はドリフト $-f+g^2 s_t = x-2x=-x$ とノイズ $\sqrt{2}$ を持ち、元の OU 過程と同じになる。定常な OU 過程はそれ自身の時間反転なのである。

もし係数を $\tfrac{1}{2}g^2$ にして g^2 の代わりに用いれば、逆ドリフトは $x-x=0$ となり、粒子群は定常分布にとどまるどころかブラウン運動のように広がってしまう。したがって逆 SDE には係数 g^2 丸ごと、すなわち ODE の 2 倍の補正が必要になる。この係数の理由は、サンプラの族を考えることで説明がつく。

命題（周辺分布を共有する 1 パラメータ族） **p_t を、決定論的な双子の命題の仮定および $\mathbf{f}$ が $\mathbf{x}$ についてリプシッツ連続であるという仮定のもとでの順方向 SDE の周辺密度とする。任意の $\lambda\ge 0$ に対し、ノイズ振幅 λg と次のドリフトを持つ逆時間 SDE は、**

$$\mathbf{b}_\lambda = \mathbf{f} - \tfrac{1+\lambda^2}{2}\,g^2\,\nabla\log p_t \tag{38.71}$$

周辺分布 p_t を、初期分布 p_T から出発したときに持つ。$\lambda = 1$ と選べば Anderson の逆 SDE Equation 38.70 になり、$\lambda = 0$ と選べば確率フロー ODE Equation 38.67 になる。

サンプラが注入するノイズが大きいほど、必要なスコア補正も大きくなる。

証明逆向きの時計 $\bar{t} = T - t$ で考え、密度の時間発展を逆再生したものを $q_{\bar{t}} = p_{T-\bar{t}}$ とおくと、$\partial_{\bar{t}} q = -\partial_t p = \nabla \cdot (\mathbf{f} q) - \tfrac{1}{2} g^2 \Delta q$ となる（ただし $\mathbf{f}$ と g は順方向の時刻 $T - \bar{t}$ で評価する）。一方、逆向きの時計における過程 $d\mathbf{Y} = \mathbf{b}'\, d\bar{t} + \lambda g\, d\mathbf{W}$ は、それ自身のフォッカー・プランク方程式 $\partial_{\bar{t}} q = -\nabla \cdot (q\, \mathbf{b}') + \tfrac{\lambda^2}{2} g^2 \Delta q$ に従う。両者が一致するためには次が必要である。

$$-\nabla \cdot (q\, \mathbf{b}') = \nabla \cdot (\mathbf{f} q) - \tfrac{1+\lambda^2}{2} g^2 \Delta q = \nabla \cdot \Big(q \Big[\mathbf{f} - \tfrac{1+\lambda^2}{2} g^2 \nabla \log q \Big] \Big),$$

ここで**両方**のラプラシアンを Equation 38.63 によって輸送形へと書き換えた。この式は $\mathbf{b}' = -\mathbf{f} + \tfrac{1+\lambda^2}{2} g^2 \nabla \log q$ に対して成り立つ。逆向きの時計におけるドリフト $\mathbf{b}'$ は順方向の時刻ではドリフト $-\mathbf{b}'$ となるため、Equation 38.71 が得られる。残るは、発展方程式の一致と初期分布の一致 $q_0 = p_T$ から周辺分布の一致を導く段階であり、ここで一意性が必要になる。$\lambda = 0$ の場合は、決定論的な双子の命題における特性曲線に沿った議論そのものである。$\lambda > 0$ の場合は、Equation 38.52 の直後で証明なしに述べた一意性による。実際、$\mathbf{b}'$ は $\mathbf{f}$ と $\mathbf{u}_t$ がリプシッツ連続であるときリプシッツ連続になるため、この一意性が適用できる。■

この証明から係数 2 の意味も明らかになる。ノイズ λg を持つ逆過程はそれ自身の密度を平滑化してしまうため、そのドリフトは 2 つ分の平滑化を打ち消さなければならない。まず 1 つ目の $\tfrac{1}{2} g^2 \nabla \log p_t$ が順方向の平滑化を打ち消す。ノイズを一切注入しない確率フロー ODE の補正はこれで全部である。さらに $\tfrac{\lambda^2}{2} g^2 \nabla \log p_t$ が、逆過程自身のノイズによる平滑化を補償する。Anderson の $\lambda = 1$ はこの半分ずつを両方とも含んでいる。もっとも、周辺分布だけではドリフトは一意に定まらない。同じ周辺分布を持つ他の任意のドリフトは $\mathbf{b}'$ と場 $\mathbf{w}$（ただし $\nabla \cdot (q\mathbf{w}) = 0$）の分だけ異なり、演習問題 7 ではそのような場を構成する。

この自由度は何の役に立つのだろうか。ノイズはサンプラに「忘却」をもたらす。初期分布が p_T からわずかにずれていたり、前のステップで誤差が生じたりして状態の分布に誤差が入ると、ODE はそれを $t = 0$ までそのまま運び切る。写像が全単射であり、何も失わないからである。これに対し SDE のノイズはその誤差の一部を洗い流してくれる（演習問題 10 でこれら両方の主張を厳密に示す）。その代償としてスコア項は大きくなり、ノイズを含む長いステップはそれ自身の誤差を新たに付け加える。拡散モデルのサンプラはこのトレードオフを踏まえて λ を選ぶ（17.5 節）。どの値を選んでもサンプラが必要とする場は同じ $\nabla \log p_t$ であるため、スコアの推定が 1 つあれば族全体を動かせる。

Figure 38.22 に、通し例の混合分布における 3 つの値の λ を示す。決定論的な経路は $\bar{t} \approx 2.2$ 付近、すなわち周辺分布が二峰性でなくなる順方向の時刻 0.78 付近で 2 つのモードへ向かって分岐する。

38.5.5.3 逆時間サンプリング

訓練されたネットワークの代わりに厳密なスコアを使って、実際にデータを生成してみよう。この実験に限って、データの重みを非対称にする。モードの位置と幅はそのままにし、質量の 5 分の 4 を左側へ置く。すなわち $p_0 = 0.8\, \mathcal{N}(-2, 0.25) + 0.2\, \mathcal{N}(2, 0.25)$ とする。対称なノイズから出発するサンプラは、モードの位置だけでなくその比率も正しく再現しなければならず、これは失敗しうる厳しいテストになる。

20,000 個の点を $\mathcal{N}(0, 1)$（p_T の $T = 3$ における代用）からサンプリングし、逆向きの時計でオイラー・丸山法のステップを用いて Anderson の逆 SDE を $t = 0$ まで積分する。次のセルで

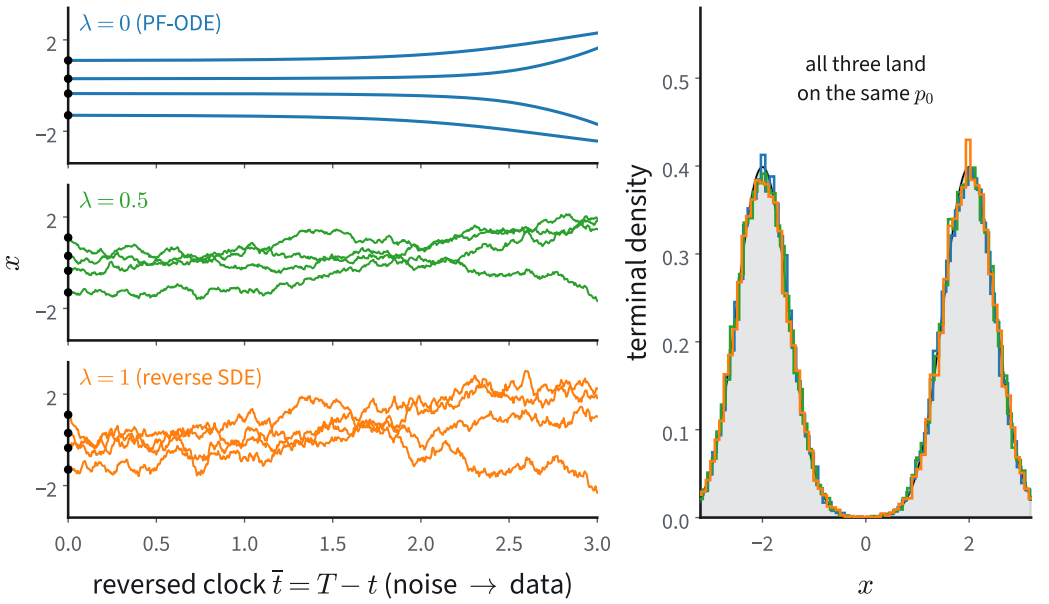
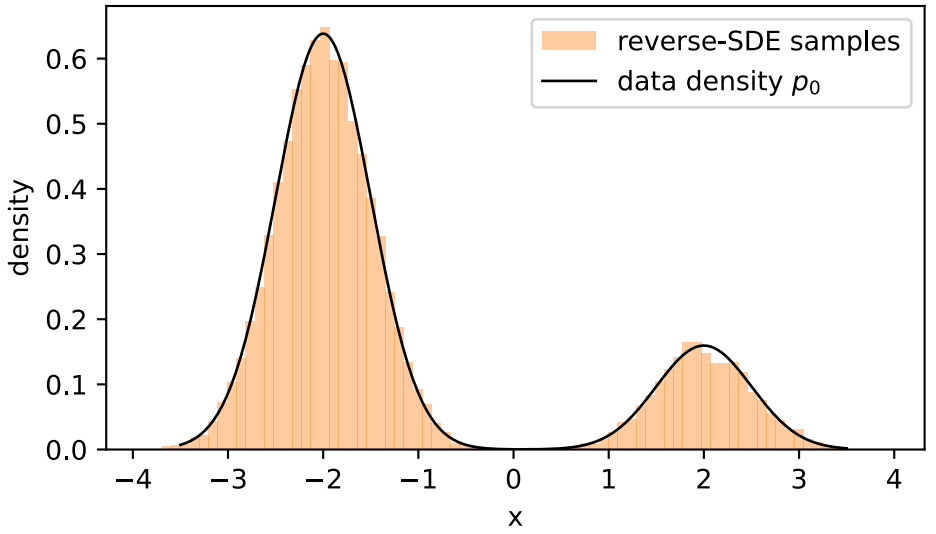


Figure 38.22: $t = T$ における同じ 4 つの出発点からの逆時間軌跡。通し例の混合分布の厳密なスコアを用い、ドリフトを $\mathbf{f} - \frac{1+\lambda^2}{2} g^2 \nabla \log p_t$、ノイズを λg としている。確率フロー ODE（$\lambda = 0$）は決定論的であり、$\lambda = 0.5$ と Anderson の逆 SDE（$\lambda = 1$）はノイズとより大きなスコア補正を加える。$t = 0$ における各 20,000 粒子の 3 つのヒストグラム（右）はいずれもデータ密度 p_0 と一致する。

は、まず p_T が $\mathcal{N}(0,1)$ からどれだけ離れているかを測り、続いて生成サンプルをデータと比較する。すなわち、原点より左側の質量、各モードの平均と幅、および p_0 に対する KS 距離（同じサイズの厳密なサンプルの KS 距離と並べて表示）を確認する。

```
uneven = dict(ws=np.array([0.8, 0.2]))  # four fifths of the data on the left
pT = p_t(gs, 3.0, **uneven)
print(f'max|p_T - N(0,1)| at T=3: {np.abs(pT - gauss(gs, 0.0, 1.0)).max():.4f} '
      f'(peak {pT.max():.2f})')
x = rng.standard_normal(n)              # N(0, 1) stands in for p_T
for k in range(steps):                  # reversed clock: t_bar = k dt
    t = T - k * dt                      # forward time
    x += (theta * x + g**2 * score_t(x, t, **uneven)) * dt \
         + g * np.sqrt(dt) * rng.standard_normal(n)
left, right = x[x < 0], x[x >= 0]
print(f'mass on the left: {len(left) / n:.3f} (data 0.800)')
print(f'means {left.mean():+.3f}, {right.mean():+.3f} (data -2, +2)')
print(f'widths {left.std():.3f}, {right.std():.3f} (data 0.5)')
x_ref = sample_p0(n, rng, **uneven)     # an exact sample of the same size
print(f'KS to the exact p_0: sampler {ks_exact(x, 0.0, **uneven):.4f}, '
      f'exact sample {ks_exact(x_ref, 0.0, **uneven):.4f}')
d2l.set_figsize((5.5, 3))
d2l.plt.hist(x, bins=80, density=True, alpha=0.4, color='C1',
             label='reverse-SDE samples')
d2l.plt.plot(gs, p_t(gs, 0.0, **uneven), 'k', lw=1, label='data density $p_0$')
d2l.plt.xlabel('x'); d2l.plt.ylabel('density'); d2l.plt.legend();
```

```
max|p_T - N(0,1)| at T=3: 0.0145 (peak 0.40)
mass on the left: 0.800 (data 0.800)
means -1.993, +1.998 (data -2, +2)
widths 0.504, 0.509 (data 0.5)
KS to the exact p_0: sampler 0.0068, exact sample 0.0054
```



サンプラはデータを復元する。モードは約 0.02 以内の誤差で ± 2 に位置し、その幅は 0.5 に近く、粒子の 5 分の 4 が 100 分の 1 以内の精度で左側に落ち着く。注目すべきはこの最後の数値で

ある。粒子の半分は原点より右側から出発しており、サンプラの中で重みに言及している箇所はスコア以外に存在しない。スコアはどこでその重みを保持しているのだろうか。$t = 0$ の近くではない。そこでは密度が無視できないあらゆる場所で Equation 38.57 の負担率が 0 か 1 になり、重みが何であれスコアは 2 つのバネ $-(x \pm 2)/0.25$ のいずれかに一致する。これは 37.5.1 節で見た、離れたモードの重みに対するスコアの不感性である。$t = 3$ でもない。そこでは 2 つの成分がほぼ同じガウス分布になっており、粒子がどこにあっても左側の成分の負担率は重み 0.8 に近い。しかし 2 つのバネもほぼ重なっている。両者の差は $2m_t/v_t$（各成分の平均を $\mp m_t$、分散を v_t としたとき）であり、その大きさは 0.2（$t = 3$）と、16（$t = 0$）に比べてごく小さい。ほぼ等しい 2 つのバネを重み 0.8 と 0.2 で平均しても、左向きのごく小さな一様の押し出しが生じるだけで、どの粒子がどちらのモードへ行くかは決まらない。

重みが働くのはその中間、すなわち 2 つのモードが分離していく最中である。2 つの成分の負担率が等しくなるのは点 $x_t^* = v_t \ln(0.8/0.2)/(2m_t)$ である。この点より左にある粒子は左の成分から来た確率の方が高く、逆過程のもとでは左の成分に行き着く確率の方が高くなる。$t = 3$ ではこの分水嶺（watershed）は $x^* \approx 7$ にあり、どの粒子よりもはるか右方に位置する。$t = 1$ になると 0.85 まで内側へ寄ってくるため、$x = +0.5$ にある粒子でさえ右より左に行き着く可能性の方が高い。$t = 0$ では谷の底の 0.09 に位置するが、その頃には谷に粒子は残っていない。高ノイズでは重みの情報はいたるところにあるが何も決めず、低ノイズでは決められるはずなのにそこに粒子がいないのである。

出発点の分布は厳密には正しくなかったが、ほとんど影響しなかった。重みが不均等な場合、p_T はデータを完全には忘れていない。その平均は $-1.2\,e^{-3} \approx -0.06$ であり、出力の 1 行目は最大 0.0145 の差が $\mathcal{N}(0,1)$ とのあいだにあることを示している（ピーク値は 0.40）。逆 SDE のノイズはこの不一致の大半を洗い流し、左モードの質量の減少は約 0.002 と、表示された比率の二項分布ノイズ 0.003 よりも小さい。何も忘れない確率フロー ODE では、その数倍の質量を取りこぼすことになる（演習問題 10）。

このサンプラは厳密ではなく、KS 距離の行はそのわずかな兆候を示しているにすぎない。どちらの距離も 0.0096 の帯域と同程度であるが、サンプラの距離の方がたいてい大きく、厳密なサンプルが帯域を超えるのが 20 回に 1 回であるのに対し、サンプラではおよそ 6 回に 1 回超える。このサンプルサイズの 1 回の実行だけでは、このようなサンプラを厳密なサンプラと区別することはできない。初期分布のずれに加えて、もう 1 つの誤差要因である離散化（discretization）がある。オイラー・丸山法の弱収束次数は 1 であるため（38.4.4.1 節）、その出力にはステップ幅に比例するバイアスが生じる。このバイアスはモードの幅に現れるが、1 回の実行ではそこでも測定できない。M 個のサンプルから推定した幅のモンテカルロ誤差は幅を $\sqrt{2M}$ で割った値であり、約 0.003（16,000 サンプルの左モード）および 0.006（4,000 サンプルの右モード）になるからである。

このバイアスはサンプリングを行わずに切り分けられる。右モード単独のガウスデータ $\mathcal{N}(2, 0.25)$ に対しては、周辺分布もガウス分布 $\mathcal{N}(m_t, v_t)$ となりスコアは $-(x - m_t)/v_t$ であるから、逆向きの各ステップは x のアフィン変換となり、サンプラの状態の平均 m と分散 V は厳密な漸化式に従う。逆ドリフトの傾きを $\kappa_t = \theta - g^2/v_t$ とおけば、漸化式は $m \leftarrow m + \Delta t\,(\kappa_t m + g^2 m_t/v_t)$ および $V \leftarrow (1 + \kappa_t \Delta t)^2 V + g^2 \Delta t$ となる。厳密な p_T から出発すれば、この漸化式には離散化以外の誤差が一切入らない。次のセルでは、300 から 2,400 ステップの一様グリッド上でこれを評価し、さらに 1,500 ステップを 2 通りの方法、すなわち一様配分と、そのうち 1,000 ステップを $t \le 0.5$ に集中させる配分で比較する。

```
def em_reverse_width(ts, mu=2.0, v0=0.25):
    """Exact output width of reverse Euler--Maruyama on data N(mu, v0),
    stepping down the time grid ts = (0, t_1, ..., T) from T to 0."""
    mu, v0 = np.array([mu]), np.array([v0])
```

```
        (m,), (V,) = mixture_params(ts[-1], mu, v0)     # start from the exact p_T
        for t, dt in zip(ts[:0:-1], np.diff(ts)[::-1]):
            (mt,), (vt,) = mixture_params(t, mu, v0)
            kappa = theta - g**2 / vt       # slope of the reverse drift in x
            m, V = (m + dt * (kappa * m + g**2 * mt / vt),
                    (1 + kappa * dt)**2 * V + g**2 * dt)
        return np.sqrt(V)

    grids = {f'{N} uniform steps': np.linspace(0, T, N + 1)
             for N in (300, 600, 1200, 2400, 1500)}
    grids['1500 steps, 1000 on t <= 0.5'] = np.concatenate(
        [np.linspace(0, 0.5, 1001), np.linspace(0.5, T, 501)[1:]])
    for name, ts in grids.items():
        print(f'{name:>28}: width excess {em_reverse_width(ts) - 0.5:.4f}')
```

```
          300 uniform steps: width excess 0.0096
          600 uniform steps: width excess 0.0047
         1200 uniform steps: width excess 0.0024
         2400 uniform steps: width excess 0.0012
         1500 uniform steps: width excess 0.0019
1500 steps, 1000 on t <= 0.5: width excess 0.0006
```

最初の 4 つのグリッドでは、真の幅 0.5 に対する幅の超過分は順に 0.0096、0.0047、0.0024、0.0012 となり、1 次のバイアスがそうあるべきとおり、ステップ幅を半分にすると半分になる。先ほどのサンプリング実行のグリッドである 600 ステップでは 0.005 である。表示された幅に見込まれる超過分はまさにこの程度であり、モンテカルロ誤差と同じ大きさであるため、1 回のサンプリング実行では分解できないが、漸化式なら明確に捉えられる。

最後の 2 行は、ステップをどこに配分すべきかを問うている。1,500 ステップを一様に使えば、600 ステップ時のバイアスの 10 分の 6 が取り除かれる。一方、$t \le 0.5$ の区間だけステップを 10 倍細かくすると（合計は同じく 1,500 ステップ）、10 分の 9 近くが取り除かれる。ステップを 10 倍細かくすると細分化した区間由来のバイアスは 10 分の 1 になるため、0.0047 から 0.0006 への減少は、出力のバイアスのほぼすべてが $t \le 0.5$ の区間、すなわち逆時間積分の最後の 6 分の 1 のステップから生じていることを物語っている。

これには 2 つの要因が重なっている。第 1 に、この区間では逆ドリフトが最も急になり、傾きは $\theta - g^2/v_0 = 1 - 8 = -7$（$t = 0$）に達する一方、$-1$ 程度にとどまるのが $t = T$ 付近であるため、1 ステップあたりの誤差が大きくなる。第 2 に、逆時間積分の序盤で生じた誤差は後続のすべてのステップで漸化式の係数 $(1 + \kappa_t \Delta t)^2 < 1$ によって縮小されるのに対し、終盤のステップで生じた誤差はそのまま出力に到達する。データの幅がさらに狭くなると傾き $\theta - g^2/v_0$ は限りなく大きくなり、ステップ幅は 38.2.3 節の安定性限界に突き当たるが、今の設定ではそこから十分に離れている。演習問題 9 ではこれら両方の効果を測定し、最良のステップ配置を探る。

ここまでの議論はすべて、明示的に書き下せるスコアに支えられていた。データは 2 成分混合分布であり、残る誤差はオイラー・丸山法のバイアスと、p_T と $\mathcal{N}(0, 1)$ のわずかな差だけであった。しかし実データにはそのような公式はなく、フォッカー・プランク方程式から p_t を数値計算することも論外である。各軸に 100 個のセルを切ったグリッドには 100^d 個の未知数が並び、しかも d はピクセルの数だからである。スコアは学習するほかない（38.6 節）。画像に対して手元にあるのは p_0 からのサンプルだけであり、それ以外には何もない。

38.5.6 まとめ

SDE $d\mathbf{X} = \mathbf{f}\,dt + g\,d\mathbf{W}$ の密度はフォッカー・プランク方程式 $\partial_t p = -\nabla\cdot(\mathbf{f}p) + \tfrac{1}{2}g^2\Delta p$ に従い、ドリフトが密度を輸送し、ノイズが密度を平滑化する。$\nabla p = p\nabla\log p$ であるから、この平滑化自体が速度 $-\tfrac{1}{2}g^2\nabla\log p$ による輸送となっており、したがって同じ密度が常微分方程式、すなわち確率フロー ODE $\dot{\mathbf{x}} = \mathbf{f} - \tfrac{1}{2}g^2\nabla\log p_t$ によって動かされる。この ODE はあらゆる時刻で SDE と同じ周辺分布を持ち、決して交差しない滑らかな軌跡と尤度を与える。同じ密度族は、ノイズ λg とドリフト $\mathbf{f} - \tfrac{1+\lambda^2}{2}g^2\nabla\log p_t$ を持つ過程の族によって逆順にたどることもできる。スコア補正のうち半分相当が順方向の平滑化を打ち消し、残りがサンプラ自身のノイズを補償する。Anderson の逆 SDE（$\lambda=1$）が ODE（$\lambda=0$）の 2 倍の補正を持つ理由はここにある。本節ではフォッカー・プランク方程式の解の一意性を認めたうえで周辺分布についてこれを証明し、過程全体に関する Anderson の定理は引用にとどめた。厳密なスコアで駆動した逆 SDE は、対称なノイズから重み 0.8 と 0.2 の混合分布をモードの位置から比率に至るまで復元した。その比率が決まるのは、モードが分離していく中間のノイズ領域である。これらすべてのサンプラにおいて、唯一の未知量はスコア $\nabla\log p_t$ である。

38.5.7 演習問題

1. **平均化としてのラプラシアン。** 滑らかな関数 ψ を、半径 r、中心 $\mathbf{x}$ の球面上で平均すると $\psi(\mathbf{x}) + \tfrac{r^2}{2d}\Delta\psi(\mathbf{x}) + O(r^4)$ になることは 38.5.2.1 節で述べたとおりである。

   1. $\mathbf{e}$ が $\mathbb{R}^d$ の単位球面上で一様分布に従うとする。対称性により、期待値が $\mathbb{E}[e_i]=0$、$\mathbb{E}[e_ie_j]=\delta_{ij}/d$、および $\mathbb{E}[e_ie_je_k]=0$ となることを示せ。これらのモーメントを $\psi(\mathbf{x}+r\mathbf{e})$ のテイラー展開に用いて、上の平均化公式を証明せよ。r^3 の項が現れないのはなぜか。
   2. 1 次元では、球面は 2 点 $x\pm\delta$ の組である。熱方程式 $\partial_t p = D\,\partial_{xx}p$ において $\partial_{xx}p$ を中心差分 $(p_{i+1}-2p_i+p_{i-1})/\delta^2$ で置き換えた前進オイラー法の 1 ステップは、新しい各 p_i を p_{i-1}、p_i、p_{i+1} の重みの和が 1 となる重み付き結合に設定することを示せ。3 つの重みすべてが非負となる最大のステップ幅 Δt を求めよ。このとき、反復によって新たな最大値が決して生じないのはなぜか。

2. **弱形式からのモーメント方程式。** 時間発展則 Equation 38.51 は、有界集合の外でゼロになるテスト関数に対して導出された。本節で扱う密度が持つガウス分布の裾によって正当化されるものとして、この法則が多項式の ϕ に対しても成り立つと仮定せよ。

   1. 1 次元 OU 過程に対して $\phi(x)=x$ と $\phi(x)=x^2$ を代入し、平均 $m(t)$ と分散 $v(t)$（いずれも X_t のモーメント）に対する ODE を導け。ガウス分布であるか否かにかかわらず、これらがモーメント ODE Equation 38.54 に任意の初期分布 p_0 で一致することを示せ。`mixture_params` と照らし合わせ、時刻 t における 2 モード混合分布の平均と分散を計算してこれを確かめよ。
   2. 線形 SDE $d\mathbf{X} = \mathbf{A}\mathbf{X}\,dt + g\,d\mathbf{W}$（$\mathbb{R}^d$ 上、$\mathbf{A}$ は定数行列）に対し、$\phi(\mathbf{x})=x_i$ と $\phi(\mathbf{x})=x_ix_j$ を用いて、平均と共分散が $\dot{\mathbf{m}}=\mathbf{A}\mathbf{m}$ および $\dot{\Sigma}=\mathbf{A}\Sigma+\Sigma\mathbf{A}^\top+g^2\mathbf{I}$ を満たすことを示せ。定常共分散はどの行列方程式を満たさなければならないか。

3. **確率流ゼロから導く定常密度。**

   1. OU フォッカー・プランク方程式で $\partial_t p=0$ とおいた式 $0=\theta\,\partial_x(xp)+\tfrac{1}{2}g^2\partial_{xx}p$ を考えよ。x で 1 回積分し、無限遠での減衰から積分定数（確率流 Equation 38.65 にマイナスをつけたもの）がゼロになることを論じよ。得られた 1 階 ODE を解き、ガウス分布の形を仮定することなく $p_\infty=\mathcal{N}(0,g^2/(2\theta))$ を示せ。
   2. 一般のドリフト $f(x)$ と定数 g に対して同様の議論を行い、p_∞ をポテンシャル U（ただし $f=-U'$）を用いて表せ。逆に、密度 $p\propto e^{-E(x)}$ が与えられたとき、p を定常

にするドリフトを求め、それを p のスコアを用いて書け。

3. 二重井戸ドリフト $f(x) = x - x^3$（$g = \sqrt{2}$）にこの結果を適用し、p_∞ のモードを求めよ。不安定なドリフト $f(x) = \theta x$（$\theta > 0$）では何が破綻するか。

4. **モードの合流。**スコア Equation 38.58 を持つ対称な 2 成分混合分布を考えよ。

   1. Equation 38.58 を負担率の公式 Equation 38.57 から導け。
   2. $\mu^2 > v_0$ のときスコアは 3 つの零点を持ち、$x = 0$ のみが $\mu^2 \le v_0$ のときの零点であることを示せ。各零点を密度の極大値（極大点）または極小値（極小点）に分類せよ。（ヒント：0 における傾きは $(\mu^2 - v_0)/v_0^2$ である。）
   3. OU 過程のもとでは混合分布は対称なまま保たれ、各成分の平均は $\pm \mu e^{-\theta t}$、各成分の分散は `mixture_params` が計算する値となる。本節のパラメータである時刻 t^* の条件、すなわち p_t が二峰性でなくなる時刻を、$\mu = 2$、$v_0 = 0.25$、$\theta = 1$、$g = \sqrt{2}$ に対して求めよ。
   4. t^* におけるこの変化は、ノイズから出発する逆時間サンプラにとって何を意味するか。（Biroli et al. (2024) はこの対称性の破れの高次元版を解析し、種分化と呼んでいる。）

5. [code] **状態依存ノイズ。**●行列ノイズを持つ SDE $d\mathbf{X} = \mathbf{f}\,dt + \mathbf{G}(\mathbf{X}, t)\,d\mathbf{W}$ に対し、$\mathbf{a} = \mathbf{G}\mathbf{G}^\top$ とおく。このとき伊藤の補正項は $\tfrac{1}{2}\sum_{ij} a_{ij}\,\partial_i\partial_j \phi$ となり、2 回の部分積分 $\int p\, a_{ij}\,\partial_i\partial_j\phi = \int \phi\,\partial_i\partial_j(a_{ij} p)$ によって $\partial_t p = -\nabla\cdot\mathbf{j}$ が得られる。ここで確率流は

$$j_i = f_i\, p - \tfrac{1}{2}\sum_k \partial_k (a_{ik}\, p)$$

であり、p で割ることで確率フロー速度は

$$\mathbf{u} = \mathbf{f} - \tfrac{1}{2}\,\mathbf{a}\,\nabla \log p - \tfrac{1}{2}\,\nabla\cdot\mathbf{a}, \qquad (\nabla\cdot\mathbf{a})_i = \sum_k \partial_k a_{ik}.$$

となる。定数行列の場合はスコア項の g^2 が $\mathbf{a}$ に置き換わるだけだが、状態に依存する $\mathbf{a}$ では最後の項が加わる。ここでドリフトがなく $dX = g(X)\,dW$（g は滑らかで正）で表される、周期境界条件付きの円周 $[0, 1)$ 上の過程を考えよ。

   1. 上の行列に対する確率流と一般の速度をこの SDE に特殊化せよ。定常密度では確率流が定数になり、周期性からその定数がゼロになることを示し、p_∞ を求めよ。
   2. 定数 g^2 を Equation 38.52 の中で単純に $g(x)^2$ に置き換えると、PDE $\partial_t p = \tfrac{1}{2} g(x)^2\,\partial_{xx} p$ が得られる。この PDE はどのような定常密度を予測するか。
   3. 50,000 個の粒子を一様分布から出発させ、オイラー・丸山法を用いて $g(x) = 1 + \tfrac{1}{2}\sin(2\pi x)$、$\Delta t = 10^{-3}$ で $t = 2$ までシミュレーションせよ（位置は 1 を法として折り返す）。ヒストグラムを両方の予測と比較せよ。粒子はどこに集積するか、またそれはなぜか。

6. [code] **グリッド上のフォッカー・プランク方程式。**●区間 $[-6, 6]$ を N 個のセル（幅 $\delta = 12/N$、中心 x_i）に分割し、$p_i(t)$ でセル i 内の密度を近似する。OU 方程式の保存型離散化では、各内部境界 $x_{i+1/2}$ に確率流

$$j_{i+1/2} = f(x_{i+1/2})\,\frac{p_i + p_{i+1}}{2} - \frac{g^2}{2}\,\frac{p_{i+1} - p_i}{\delta}, \tag{38.72}$$

を割り当て、両端の外側境界を通る確率流をゼロとおいて、$\dot{p}_i = -(j_{i+1/2} - j_{i-1/2})/\delta$ により時間発展させる。これは線形系 $\dot{\mathbf{p}} = \mathbf{F}\mathbf{p}$ である。

1. $\sum_i p_i\,\delta$ が任意の $\mathbf{p}$ に対して保存されること（発散定理 Equation 38.48 の離散版）を示し、外側境界の確率流をゼロとおくことが 38.5.4 節のどの境界条件に対応するか答えよ。
2. $\mathbf{F}$ を $\theta = 1$、$g = \sqrt{2}$、$\delta = 0.05$ に対して構築せよ。セル中心上の 2 モード混合分布 `p_t(x, 0.0)` から出発し、$t = 0.5$ までステップ幅 $\Delta t = 10^{-3}$ の前進オイラー法およびステップ幅 $\Delta t = 10^{-2}$ の後退オイラー法（各ステップで 1 回の `np.linalg.solve`）を用いて積分せよ。両方の結果を `p_t(x, 0.5)` と比較せよ。
3. 前進オイラー法のステップ幅を 10^{-3} から大きくしていき、丸め誤差を種とする不安定モードが成長するのに十分長い $t = 5$ までの実行で安定に保たれる最大のステップ幅を求めよ。このステップ幅を、最も負の固有値 $-\nu$（$\mathbf{F}$ の固有値）と上界 $\Delta t < 2/\nu$（38.2.3 節）を通じて関連付けよ。δ を半分にするとこの上限はどう変わるか。演習問題 1b の条件と比較せよ。
4. 各軸に 100 個のセルを持つ d 次元グリッドには 100^d 個の未知数がある。$d = 2$ および $d = 784$（28×28 画像のピクセル数）に対してこの個数を計算せよ。この結果は、拡散モデルにおけるフォッカー・プランク方程式の使われ方について何を意味するか。

Särkkä and Solin (2019) の演習問題 5.3 を改変。

7. **発散ゼロの速度場。●**$\mathbf{R}$ を定数の反対称 $d \times d$ 行列とする。

   1. 任意の滑らかな正の密度に対して $\nabla \cdot (p\,\mathbf{R}\nabla \log p) = 0$（密度を p とする）が成り立つことを示せ。ここから、任意の $\beta \in \mathbb{R}$ に対して ODE $\dot{\mathbf{x}} = \mathbf{u}_t(\mathbf{x}) + \beta\mathbf{R}\nabla \log p_t(\mathbf{x})$（ただし $\mathbf{u}_t$ は Equation 38.64 の速度）が確率フロー ODE と同じ周辺分布を持つことを導け。
   2. 定常状態にある 2 次元 OU 過程、すなわちすべての時刻で $p_t = \mathcal{N}(\mathbf{0}, \frac{g^2}{2\theta}\mathbf{I})$（すべての t）および $\mathbf{R} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ を考えよ。確率フロー速度がゼロになることを示し、修正した ODE の軌跡がどのようなものになるか述べよ。
   3. 今度は回転を SDE 自体に組み込み、$d\mathbf{X} = (-\theta\mathbf{I} + \beta\mathbf{R})\mathbf{X}\,dt + g\,d\mathbf{W}$（$\mathbb{R}^2$ 上）とする。演習問題 2b の共分散方程式を用いて、$\frac{g^2}{2\theta}\mathbf{I}$ が任意の β に対して定常共分散となり、この SDE が小問 (b) と同じ定常密度を持つことを示せ。その定常確率流、すなわちこの SDE の確率流 Equation 38.65 を計算せよ。定常確率流をゼロに強制した演習問題 3a の議論が、2 次元では成り立たないのはなぜか。
   4. $\mathbf{f} = -\nabla U$ と仮定し、$\mathbf{u}_t$ が勾配場であるとする。速度場 $\mathbf{w}$ のうち、その連続の式 Equation 38.60 の解が p_t となるすべての場の中で、$\mathbf{u}_t$ が運動エネルギー（期待値 $\mathbb{E}_{p_t}\|\mathbf{w}\|^2$）を最小にすることを示せ。なお、ベナムー・ブレニエの定式化 (Benamou and Brenier 2000) は経路 p_t 自体についても同じエネルギーを最小化する。

8. **厳密なガウス逆ステップ。●**$p_t = \mathcal{N}(m_t, v_t)$ をガウス分布である OU 過程の周辺分布とする。ステップ幅 τ にわたる厳密な順遷移は $X_{t+\tau} \mid X_t = x \sim \mathcal{N}(\alpha_\tau x, \sigma_\tau^2)$（ただし $\alpha_\tau = e^{-\theta\tau}$、$\sigma_\tau^2 = \frac{g^2}{2\theta}(1 - e^{-2\theta\tau})$）で与えられる（カーネル Equation 38.46 による）。

   1. ガウス分布の条件付けを用いて、次を示せ。

$$X_t \mid X_{t+\tau} = y \;\sim\; \mathcal{N}\!\left(\frac{\sigma_\tau^2\, m_t + \alpha_\tau v_t\, y}{v_{t+\tau}},\; \frac{\sigma_\tau^2\, v_t}{v_{t+\tau}}\right). \tag{38.73}$$

   ここで $v_{t+\tau} = \alpha_\tau^2 v_t + \sigma_\tau^2$ は時刻 $t + \tau$ における周辺分散である。

2. 平均と分散を τ の 1 次まで展開し、逆ステップ Equation 38.69 を、スコアの前の係数 g^2 まで含めて導出せよ。

9. [code] **ステップの配置。**●関数 `em_reverse_width` は、任意の時間グリッドに対してデータ $\mathcal{N}(2, 0.25)$ 上の逆時間オイラー・丸山法の厳密な出力幅を返す。本文では 1,500 ステップの 2 つのグリッドを比較した。

   1. 600 ステップの一様グリッドから始め、$t \le c$ の区間のみステップを 10 倍細かくせよ（ただし $c \in \{0.1, 0.25, 0.5, 1, 2\}$）。それぞれのステップ数と幅の超過分を記録せよ。ステップを 10 倍細かくすると細分化した区間由来のバイアスは 10 分の 1 になる。これを用いて、600 ステップのバイアスのうち $t \le c$ のステップから生じる割合を計算せよ。$t \le 0.1$ に何個の追加ステップを置けば、本文で一様グリッドに追加した 900 ステップ分の働きをするか。本文ではこれらのステップが支配的になる理由を 2 つ挙げた。一様グリッド上でそれぞれを測定せよ。すなわち、$t \in \{0.1, 0.5, 3\}$ から出発する単一の 1 ステップについて、厳密な分散から始めたときに生じる V の誤差（初期値を v_t としたとき）と、後続のステップが $t = 0$ に至るまでにその誤差に掛ける倍率を計算せよ。
   2. 総ステップ数を $N = 1{,}500$ に固定し、グリッド $t_i = T\,(i/N)^\rho$ を $\rho \in \{1, 1.5, 2, 3, 4, 6\}$ に対して試せ。どの ρ が最小の超過分を与えるか。ρ を大きくしすぎると再び悪化するのはなぜか。
   3. 二乗グリッド（$\rho = 2$）上のバイアスは、依然として $1/N$ に関して 1 次であるか。$N = 750$、1,500、3,000 で確かめよ。グリッドの変更が変えたのは次数か、それとも定数係数か。
   4. より幅の狭いデータ $v_0 = 0.01$（$t = 0$ における逆ドリフトの傾きは $1 - 2/0.01 = -199$ となる）に対して小問 (b) を繰り返し、各超過分を目標幅 0.1 と比較せよ。最良の ρ はどう移動するか。この結果は、一部の方向で分散が小さいデータに対して何を示唆するか。（17.5 節では、ノイズレベルに関するべき乗グリッドを用いて画像サンプラのステップを低ノイズ側へと密に集めている。）

10. [extended] **事前分布の不一致：ODE 対 SDE。**●●サンプラは厳密な終端分布ではなく基準の事前分布 $q_T = \mathcal{N}(0, 1)$（p_T の代用）から出発する。本課題では、逆 SDE と逆時間確率フロー ODE がその不一致をどのように $t = 0$ まで運ぶかを比較する。小さい終端時刻に対して不均等な重み $w = (0.8, 0.2)$（38.5.5 節の逆サンプラで用いた設定であり、p_T が q_T から目に見えてずれる（小さい T の場合））を用いよ。具体的には、`**uneven` を `p_t`、`score_t`、`mixture_cdf`、`sample_p0` に渡し、`score=lambda x, t: score_t(x, t, **uneven)` を `pf_velocity` に渡す。

    1. 本文と同様の Anderson の逆 SDE と、ホイン法のステップによる逆時間確率フロー ODE を、いずれも q_T から $T \in \{0.25, 0.5, 1, 2, 3\}$、ステップ幅 $\Delta t = 0.005$ で実行せよ。$x < 0$ となるサンプルの割合（目標値は 0.8）を記録せよ。なお、$T = 3$ における SDE は本文の実行と同じである。
    2. Anderson の定理は過程全体に対して成り立つため、$X_T = y$ から出発した逆 SDE は事後分布に従う $X_0 \sim p(x_0 \mid x_T = y)$ を生成する。これを用いて、SDE の左モード質量を、$y \sim q_T$ に関する p_T の負担率（Equation 38.57）の期待値として予測せよ。ODE については、x_T から x_0 への写像が単調増加であり p_T を p_0 へ写すことを用いて、標準正規分布の CDF および p_T と p_0 の CDF から左モード質量を予測せよ。両方の予測を数値積分で評価し、実測値と比較せよ。
    3. サンプラの出力の分布を q_0 とする。37.3.3 節のデータ処理不等式を用いて、ODE の出力が $D_{\mathrm{KL}}(q_0 \| p_0) = D_{\mathrm{KL}}(q_T \| p_T)$ を満たす一方、SDE の出力は $D_{\mathrm{KL}}(q_0 \| p_0) \le D_{\mathrm{KL}}(q_T \| p_T)$ を満たすことを証明せよ。
    4. $D_{\mathrm{KL}}(q_T \| p_T)$ と SDE の $D_{\mathrm{KL}}(q_0 \| p_0)$ を数値積分で各 T に対して計算せよ。事後分布 $p(x_0 \mid x_T = y)$ は 2 成分混合ガウス分布であり、その各成分は演習問題 8a から従う

ため、q_0 は y に関する 1 次元積分で表される。$T = 1$ において、SDE は不一致を何分の 1 に縮小するか。

5. $T = 1$ において、族 Equation 38.71 のサンプラ（$\lambda \in \{0, 0.5, 1, 2\}$）に対して小問 (a) の測定を繰り返せ。順方向の終端時刻 T を短くせざるをえない場合、どのサンプラを使うべきか。また、ステップサイズを固定したとき λ をどこまで大きくできるかを制限する要因は何か。

38.6 スコアマッチングと拡散モデル

ガウスノイズをデータに変換する方法はすでにわかっている。逆時間 SDE（reverse-time SDE）または確率フロー ODE（probability-flow ODE）を積分すればよい。その代わりに、どちらも訪れるすべての点におけるノイズ付加データのスコア関数 $\nabla \log p_t$ という 1 つの量を必要とする。前節の 2 モード混合分布では、数式がこれを与えてくれた。しかし画像では、手元にあるのはサンプルだけであり、数式はない。本節では画像の訓練セットに見立てて、その混合分布から抽出した 4,096 個のサンプルを用いる。たった 1 点においてすら値を直接評価できないベクトル場を、学習しなければならないのである。

最尤法（maximum likelihood）で密度を当てはめてから微分しても解決にはならない。ネットワーク出力の指数をとる柔軟な密度には、全空間にわたる積分である正規化定数（normalizing constant）が必要になるからである。Aapo Hyvärinen は 2005 年、スコアには正規化定数が不要であり、部分積分（integration by parts）によって未知のデータスコアを目的関数から消去できることを見いだした (Hyvärinen 2005)。Pascal Vincent はさらに単純な事実に気づいた (Vincent 2011)。ノイズ付加データのスコアは、あるノイズ付きの点を生成し得たクリーンな点すべてにわたって、「元のクリーンな点へ戻る方向」を平均したものに等しい。この方向はすべての訓練ペアに対して既知であるため、通常の最小二乗法（least squares）によってサンプルのみからスコアを学習できる。

拡散モデル（diffusion model）はこのようにして訓練される。「加えたノイズを予測せよ」という標準的な損失は、特定の重み付けを伴うこの回帰にほかならない。最もよく知られた 2 つのサンプラである DDPM と DDIM も新しい方程式を持ち込むわけではなく、それぞれ既存の方程式を離散化したものである。混合分布で検証する前に、1 次元のガウス分布を用いれば、各主張を手計算で確かめられる数値へと落とし込める。

本節ではニューラルネットワークの訓練はしない。まず 1 つのノイズレベルにおけるスコアを、厳密に解けるほど小さなモデルを用いて最小二乗法で学習する。続いて混合分布の厳密なスコアの上でサンプラを解析し、測定される各誤差がサンプラ自身に由来するようにする。最初は単一のノイズレベルで試すが、それだけでは不十分であることがわかり、次にすべてのレベルを同時に扱う。そして終盤で 1 度だけ、すべてのレベルで最小二乗法により当てはめたスコアの上でサンプラを動かす。ネットワークとその誤差は第 17 章の役割であり、そこでは本節の 3 つの結果、すなわち Vincent の定理、DDPM と 38.4 節の分散保持（variance-preserving）SDE との同一視、および DDIM とオイラー法との同一視を土台とする。Hyvärinen の恒等式の証明とガイダンスの項は、初読時には飛ばしてもよい。

38.6.1 スコアの学習

データのスコアを近似するベクトル場 $\mathbf{s}_\theta$（実際には d 入力 d 出力のネットワーク）を求めたいが、できるのはサンプルの抽出だけである。最初の課題は、モデルと真のベクトル場を比較し、かつサンプルから評価できる目的関数を構成することである。

38.6.1.1 正規化定数を含まない目的関数

ニューラルネットワークが最も柔軟に密度を定義するのはエネルギーを介する場合である。**エネルギーベースモデル**（energy-based model）とは $p_\theta(\mathbf{x}) = e^{-E_\theta(\mathbf{x})}/Z_\theta$ のことであり、ここで E_θ はネットワーク、$Z_\theta = \int e^{-E_\theta(\mathbf{x})}\, d\mathbf{x}$ は正規化定数である (LeCun et al. 2006)。最尤法には $\nabla_\theta \log Z_\theta$ が必要となるが、これは d 次元積分であり、θ を更新するたびに変化する。一方、$\mathbf{x}$ に関するスコアにはこれが現れない。Z_θ は $\mathbf{x}$ に関して定数であるため、$\nabla_{\mathbf{x}} \log p_\theta = -\nabla_{\mathbf{x}} E_\theta$ となるからである。

そこで、密度ではなくベクトル場どうしを比較する。データのスコアを $\nabla \log p$ とすると、**明示的スコアマッチング**（explicit score matching）の目的関数は、データが存在する場所で重み付けした 2 つのベクトル場間の平均二乗距離である。

$$J_{\text{ESM}}(\theta) = \tfrac{1}{2}\, \mathbb{E}_{\mathbf{x}\sim p}\left[\left\| \mathbf{s}_\theta(\mathbf{x}) - \nabla \log p(\mathbf{x}) \right\|^2 \right]. \tag{38.74}$$

この目的関数は、データが確率質量を持つすべての場所でモデルのベクトル場がデータのスコアに一致するとき、かつそのときに限り 0 となる。$\mathbf{s}_\theta$ が密度 q のスコアである場合、これは 37.5.1 節のフィッシャーダイバージェンス（Fisher divergence）に一致する（同節では、このようなダイバージェンスが何を識別でき、何を識別できないかも論じている）。ここでは $\mathbf{s}_\theta$ は任意のベクトル場でよい。d 個の出力を持つネットワークは、何らかの関数の勾配である必要がないからである。この目的関数から正規化定数は消えたが、唯一未知であるベクトル場 $\nabla \log p$ がまだ含まれている。Hyvärinen の部分積分がこれを消去する。

命題（Hyvärinen の恒等式） *p を滑らかな正の密度、$\mathbf{s}_\theta$ を滑らかなベクトル場とし、$p(\mathbf{x})\, \mathbf{s}_\theta(\mathbf{x}) \to \mathbf{0}$（$\|\mathbf{x}\| \to \infty$ のとき）が成り立ち、以下の期待値がすべて有限であるとする。このとき、次が成り立つ。*

$$J_{\text{ESM}}(\theta) = \mathbb{E}_{\mathbf{x}\sim p}\left[\tfrac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|^2 + \nabla \cdot \mathbf{s}_\theta(\mathbf{x}) \right] + C, \tag{38.75}$$

ここで、$\nabla \cdot \mathbf{s}_\theta = \sum_i \partial s_i / \partial x_i$ は発散であり、$C = \tfrac{1}{2}\mathbb{E}_p \|\nabla \log p\|^2$ は θ に依存しない。

証明 Equation 38.74 の二乗を展開する。項 $\tfrac{1}{2}\mathbb{E}\|\nabla \log p\|^2$ は定数 C であり、項 $\tfrac{1}{2}\mathbb{E}\|\mathbf{s}_\theta\|^2$ はそのまま Equation 38.75 に現れるため、残るは交差項の変形である。1 次元では次のようになる。

$$-\mathbb{E}_{x\sim p}\left[s_\theta(x)\, (\log p)'(x) \right] = -\int s_\theta(x)\, \frac{p'(x)}{p(x)}\, p(x)\, dx = -\int s_\theta(x)\, p'(x)\, dx,$$

さらに境界項 $[s_\theta\, p]_{-\infty}^{\infty} = 0$ のもとで部分積分すると、$+\int s_\theta'(x)\, p(x)\, dx = \mathbb{E}_p[s_\theta']$ が残る。これはスタインの恒等式 $\mathbb{E}_p[s_\theta' + s_\theta\, (\log p)'] = 0$（37.5.2 節）にほかならず、導関数が未知の密度からモデル側へと移っている。d 次元では、各座標 i に対して s_i を s_θ の代わりに、$\partial_i p$ を p' の代わりに用いて同じ手順を適用し、まず x_i について積分する（仮定した可積分性のもとで、フビニの定理によりこれが正当化される）。これを i について足し合わせると、交差項は $\mathbb{E}_p[\sum_i \partial_i s_i] = \mathbb{E}_p[\nabla \cdot \mathbf{s}_\theta]$ となる。■

Equation 38.75 の右辺の各項はいずれも、与えられたモデルに対して評価できる量の p のもとでの期待値であるため、Z_θ も $\nabla \log p$ も使わずに標本平均で推定できる。この右辺を**陰的スコアマッチング**（implicit score matching）と呼ぶ。

ガウス分布を使えば、この恒等式が機能することが確かめられる。データを $x \sim \mathcal{N}(0, v)$（スコアは $-x/v$）とし、1 パラメータモデル $s_w(x) = -x/w$ をとる。このとき $\tfrac{1}{2}s_w^2 + s_w' = x^2/(2w^2) - 1/w$

となるため、陰的目的関数は $J(w) = v/(2w^2) - 1/w$ である。その導関数 $-v/w^3 + 1/w^2$ は $w = v$ で 0 になる。すなわち、目的関数はデータのスコアを一度も参照せずに分散を返す。定数項も一致する。最小点では $J(v) = -1/(2v)$ であり、また $C = \frac{1}{2}\mathbb{E}[x^2/v^2] = 1/(2v)$ であるため、そこでの明示的目的関数は $J(v) + C = 0$ となり、モデルがデータに一致するときに成り立つべき値と合致する。

実際には、期待値は標本平均になる。v（目的関数 J 中の分散）を、x^2 の 10,000 回の抽出（$\mathcal{N}(0, 2)$ からのサンプル）にわたる平均で置き換え、w のグリッド上で最小化してみよう。このとき最小点はその標本平均となり、その標準誤差は $v\sqrt{2/10^4} \approx 0.03$ であるため、2 の前後（小数第 2 位で数単位の誤差範囲内）の値が得られるはずである。

```
rng = np.random.default_rng()
v = 2.0
x = rng.normal(0.0, np.sqrt(v), 10000)        # data x ~ N(0, v)
w = np.linspace(0.5, 4.0, 701)                # model s_w(x) = -x / w
J_ism = (x**2).mean() / (2 * w**2) - 1 / w    # E[s_w^2 / 2 + s_w']
print(f'implicit score matching picks w = {w[J_ism.argmin()]:.2f}  (v = {v})')
```

```
implicit score matching picks w = 1.98  (v = 2.0)
```

実際にその通りになる。Equation 38.75 の 2 つの項は互いに逆向きに引き合っており、ガウス分布の例はその仕組みをよく示している。発散の項 $-1/w$ は、サンプルの位置で内側を向くベクトル場、すなわちデータがある場所の吸い込み口（シンク）を優遇し、その傾きが急であるほど目的関数を大きく引き下げる。一方、二乗ノルム $x^2/(2w^2)$ はベクトル場が際限なく大きくなるのを食い止める。両者が釣り合う点が真の分散に一致する。

陰的スコアマッチングには 2 つの限界がある。第 1 に、証明では境界項が消える必要がある。有界区間上のデータでは境界項が 0 にならず、陰的目的関数は誤ったベクトル場を選んでしまう。第 2 の限界はコストである。発散はヤコビ行列（Jacobian matrix）のトレース $\nabla \cdot \mathbf{s}_\theta = \operatorname{tr}\left(\partial \mathbf{s}_\theta / \partial \mathbf{x}\right)$ である。これを厳密に計算するにはヤコビ行列の 1 行ごとに 1 回のリバースモードの逆伝播が必要となり、合計で d 回のパスを要するが、画像では d が数百万にのぼる。Hutchinson の推定量（38.3.3 節）はランダムな方向ごとに 1 回のベクトル・ヤコビ行列積でトレースを置き換えるものの、分散が増える。ノイズ除去スコアマッチングは、このトレースを完全に回避する。

38.6.1.2 ノイズ除去スコアマッチング

トレースを含まない目的関数が必要であり、同時にサンプラが実際に必要とする場所（データ上だけでなくその近傍の点を含む領域）でのスコアも必要である。**ノイズを加えた**データのスコアを一致させることで、この両方の要件が満たされる。各サンプルにスケール σ のガウスノイズを加えて摂動させる。

$$\tilde{\mathbf{x}} = \mathbf{x} + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$
$$p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}};\, \mathbf{x},\, \sigma^2 \mathbf{I}).$$

ノイズ付加周辺密度 $p_\sigma(\tilde{\mathbf{x}}) = \int p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}$ はデータの密度をガウス分布でぼかしたものであり、p へと σ の減少に伴って近づく。そのスコアはデータのスコアと同様に未知である。しかし、**条件付き分布**のスコアはガウス密度の対数から直ちに得られる。

$$\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \frac{\mathbf{x} - \tilde{\mathbf{x}}}{\sigma^2} = -\frac{\epsilon}{\sigma}. \tag{38.76}$$

このベクトルは、ノイズ付き観測からそれを生み出したクリーンサンプル（clean sample）へと戻る方向を指しており、生成したすべてのペアに対して既知である。**ノイズ除去スコアマッチング**（denoising score matching、DSM）は、これを回帰目標（regression target）として用いる。

$$J_{\mathrm{DSM}}(\theta) = \tfrac{1}{2}\,\mathbb{E}_{\mathbf{x}\sim p,\ \tilde{\mathbf{x}}\sim p_\sigma(\cdot\mid\mathbf{x})} \Big[\, \big\| \mathbf{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \big\|^2 \Big]. \tag{38.77}$$

この損失は、直接見ることはできないクリーンな点へ戻る方向を、ノイズ付きの点のみから予測するようネットワークに求める。目標はクリーンな点に依存する一方、求めたいスコアはノイズ付きの点のみに依存する。最小二乗法は目標の条件付き平均を当てはめるため、両者を結びつける。次の補題はその事実を述べたものである。初等的な内容であるが、本節と次節はいずれもこれに依拠しており、次節ではこれを速度に適用する。

補題（条件付き平均への回帰） (X, Y) **が同時分布に従い** $\mathbb{E}\|Y\|^2 < \infty$ **を満たすとし、** $\mathbf{m}(X) = \mathbb{E}[Y \mid X]$ **とおく。このとき、任意の可測関数 v（ただし** $\mathbb{E}\|\mathbf{v}(X)\|^2 < \infty$**）に対して次が成り立つ。**

$$\mathbb{E}\,\|\mathbf{v}(X) - Y\|^2 = \mathbb{E}\,\|\mathbf{v}(X) - \mathbf{m}(X)\|^2 + \mathbb{E}\,\|Y - \mathbf{m}(X)\|^2 . \tag{38.78}$$

第 2 項は v を含まない。すなわち、ノイズを含む目標 Y **に対する最小二乗損失を最小化することは、定数項の差を除いて、条件付き平均** $\mathbf{m}(X)$ **に対する最小二乗損失を最小化することと同じ問題である。**

証明 $\pm\mathbf{m}(X)$ を挿入して展開する。交差項は $2\,\mathbb{E}\big[(\mathbf{v}(X) - \mathbf{m}(X))^\top (\mathbf{m}(X) - Y)\big]$ である。塔性質 $\mathbb{E}[Z] = \mathbb{E}[\mathbb{E}[Z \mid X]]$（36.2 節）により、まず Y について（X を固定したもとで）平均をとることができ、$\mathbb{E}[\mathbf{m}(X) - Y \mid X] = \mathbf{0}$ から交差項は 0 になる。■

命題（Vincent の定理） **ノイズ付加カーネル** $p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$ **が** $\tilde{\mathbf{x}}$ **に関して微分可能であり、勾配と x に関する積分が交換可能（ガウスカーネルでは両方成り立つ）であるとし、さらに DSM の目標と** $\mathbf{s}_\theta(\tilde{\mathbf{x}})$ **が有限の二次モーメントを持つとする。このとき、次が成り立つ。**

$$J_{\mathrm{DSM}}(\theta) = \tfrac{1}{2}\,\mathbb{E}_{\tilde{\mathbf{x}}\sim p_\sigma} \Big[\, \big\| \mathbf{s}_\theta(\tilde{\mathbf{x}}) - \nabla \log p_\sigma(\tilde{\mathbf{x}}) \big\|^2 \Big] + C,$$

ここで、 C **は** θ **に依存しない。右辺は** p_σ **に対する明示的目的関数 Equation 38.74 であるため、ノイズ付加データ上でのノイズ除去スコアマッチングと明示的スコアマッチングは、同じ最小点および同じ勾配を持つ。**

証明 $X = \tilde{\mathbf{x}}$、$Y = \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$ として補題を適用する。残る課題は Y の条件付き平均を求めることであり、その同定こそが本定理の実質的な内容である。すなわち、**周辺スコアは条件付きスコアの事後平均（posterior mean）に等しい**。事後分布 $p(\mathbf{x} \mid \tilde{\mathbf{x}})$ をベイズの定理で書き下し、続いて勾配と積分を交換すると、次のようになる。

$$\begin{aligned} \mathbb{E}\,[Y \mid \tilde{\mathbf{x}}] &= \int \frac{\nabla_{\tilde{\mathbf{x}}}\, p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}{p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}\, \frac{p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x})}{p_\sigma(\tilde{\mathbf{x}})}\, d\mathbf{x} \\ &= \frac{\nabla_{\tilde{\mathbf{x}}} \int p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}}{p_\sigma(\tilde{\mathbf{x}})} = \nabla \log p_\sigma(\tilde{\mathbf{x}}). \end{aligned}$$

■

この証明では、ガウス分布の性質としては微分可能性しか使っておらず、データ分布についてはいかなる性質も仮定していない。有限の訓練セットに対しては積分は和になり、定理はぼかされた訓練セットのスコアを返す。ここから示唆に富む重大な帰結が導かれる。あらゆるノイズレベルで最小値に到達したネットワークと、そのスコアを誤差なく追跡するサンプラがあれば、返されるのは訓練点だけであり、それ以外の点は一切生成されない（演習問題 3 で極限 $\sigma \to 0$ を調べる）。したがって、モデルが新しいサンプルを生み出せるのは、最小値の手前で止まる回帰のおかげであって、目的関数のおかげではない。いつそのような汎化が起きるかについて知られている知見は 17.10 節にまとめられている。ガウスノイズが選ばれる理由は、その条件付きスコア Equation 38.76 がノイズそのものを $-1/\sigma$ 倍したものになるからである。ゆえにスコアの予測とノイズ ϵ の予測は同じ課題であり、ノイズを予測するネットワーク ϵ_θ はスコアモデル $\mathbf{s}_\theta = -\epsilon_\theta/\sigma$ を定める。拡散モデルが用いるのは、この**ノイズ予測**（noise prediction）形式である。

例：ガウス分布再びデータを $x \sim \mathcal{N}(0, v)$ とし、レベル σ でノイズを加える。ノイズ付加周辺分布は $p_\sigma = \mathcal{N}(0, v+\sigma^2)$ であり、そのスコアは $-\tilde{x}/(v+\sigma^2)$ である。2 つのガウス分布に対するベイズの定理から、クリーンな点の事後分布は次のようになる。

$$x \mid \tilde{x} \;\sim\; \mathcal{N}\!\left(\frac{v}{v+\sigma^2}\,\tilde{x},\; \frac{v\sigma^2}{v+\sigma^2}\right),$$

これはノイズ付きの点をデータの平均へ向けて縮小（shrinkage）させた分布であり、その広がりは $\tilde{x}$ に依存しない。本項のすべての主張が具体的な数式になる。目標 $Y = (x-\tilde{x})/\sigma^2$ の条件付き平均は $\left(\frac{v}{v+\sigma^2} - 1\right)\tilde{x}/\sigma^2 = -\tilde{x}/(v+\sigma^2)$ となり、Vincent の定理の通り周辺スコアに一致する。達成可能な最小の平均二乗誤差、すなわち Equation 38.78 の第 2 項は、目標の事後分散（posterior variance）である $\mathrm{Var}(x \mid \tilde{x})/\sigma^4 = \frac{1}{\sigma^2} - \frac{1}{v+\sigma^2}$ となる。$v = 1$、$\sigma = 0.5$ のとき、この下限は $4 - 0.8 = 3.2$ であり、クリーンな点の事後平均は $0.8\,\tilde{x}$ である。100,000 組のペアを用いたモンテカルロ実行によって両方の数値を確かめよう。

```
x, eps = rng.standard_normal((2, 100000))             # v = 1
xt, y = x + 0.5 * eps, -eps / 0.5                      # sigma = 0.5; DSM target y
print(f'mean squared error of the conditional mean -xt/1.25: {((y + xt /
↪  1.25)**2).mean():.3f}'
      '  (floor 3.2)')
print(f'least-squares slope of x on xt: {(xt * x).mean() / (xt**2).mean():.3f}  (0.8)')
```

```
mean squared error of the conditional mean -xt/1.25: 3.186  (floor 3.2)
least-squares slope of x on xt: 0.800  (0.8)
```

最良の回帰関数であっても、目標に対する平均二乗誤差は 0 ではなく 3.2 になる。1 つのノイズ付きの点を説明し得るクリーンな点は多数存在し、回帰はその平均しか返せないからである。ここでは 3.2 の損失が完璧な当てはめを意味し、ゼロの損失は不可能である。傾き 0.80 は縮小係数 $v/(v+\sigma^2)$ である。この 2 つの観測事実は、いずれもガウス分布に限ったものではない。

この傾きは、Robbins が Tweedie に帰属させて発表した **Tweedie の公式**（Tweedie's formula）の一例である (Robbins 1956; Efron 2011)。Vincent の恒等式 $\mathbb{E}[(\mathbf{x}-\tilde{\mathbf{x}})/\sigma^2 \mid \tilde{\mathbf{x}}] = \nabla \log p_\sigma(\tilde{\mathbf{x}})$ を整理すると、次の式が得られる。

$$\mathbb{E}[\mathbf{x} \mid \tilde{\mathbf{x}}] = \tilde{\mathbf{x}} + \sigma^2\, \nabla \log p_\sigma(\tilde{\mathbf{x}}). \tag{38.79}$$

二乗誤差のもとで最良のノイズ除去器（denoiser）は、ノイズ付きの点をノイズ付加データのスコアの方向にステップサイズ σ^2 で 1 ステップ進める（Figure 38.23）。ガウス分布の場合は

$\tilde{x} - \sigma^2 \tilde{x}/(v+\sigma^2) = 0.8\,\tilde{x}$ となり、先ほど測定した傾きに一致する。

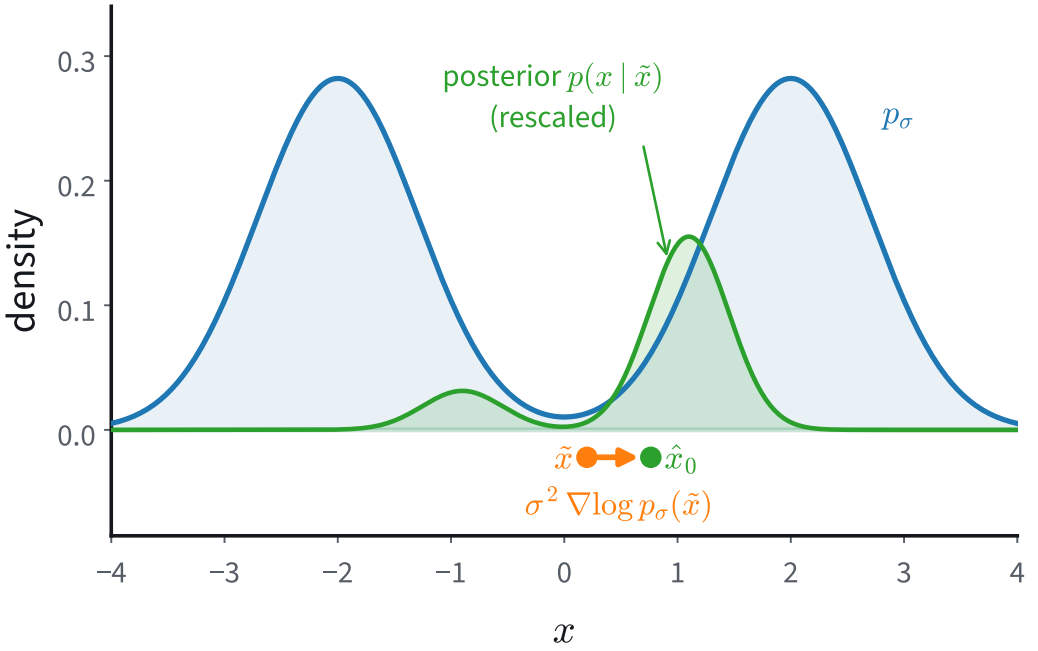


Figure 38.23: 混合分布 $\frac{1}{2}\mathcal{N}(-2, 0.5^2) + \frac{1}{2}\mathcal{N}(2, 0.5^2)$ に $\sigma = 0.5$ でノイズを加えた場合の Tweedie の公式。ノイズ付き観測 $\tilde{x}$ は、ノイズ付加密度 p_σ の低密度の谷に位置している。クリーンな点の事後分布は二峰性だが非対称であり、ステップ $\tilde{x} + \sigma^2 \nabla \log p_\sigma(\tilde{x})$ はその平均 $\hat{x}_0 = \mathbb{E}[x \mid \tilde{x}]$、すなわち最適なノイズ除去器に着地する。

損失の下限も同様に一般的である。最適点において、平均二乗誤差 $2J_{\text{DSM}}$ は $\mathbb{E}\|Y - \mathbf{m}(X)\|^2$、すなわち目標の平均事後分散であり回帰の**ベイズリスク**（Bayes risk）でもある量に等しい。これを計算するため、補題 Equation 38.78 をベクトル場 $\mathbf{v} = \mathbf{0}$ に適用すると、$\mathbb{E}\|Y\|^2 = \mathbb{E}\|\mathbf{m}(X)\|^2 + \mathbb{E}\|Y - \mathbf{m}(X)\|^2$ が得られる。目標 $Y = -\epsilon/\sigma$ は $\mathbb{E}\|Y\|^2 = d/\sigma^2$（$d$ 次元の場合）を満たし、その条件付き平均はスコアであるため、次が成り立つ。

$$\mathbb{E}\,\|Y - \nabla \log p_\sigma(\tilde{\mathbf{x}})\|^2 = \frac{d}{\sigma^2} - \mathbb{E}_{\tilde{\mathbf{x}} \sim p_\sigma}\,\|\nabla \log p_\sigma(\tilde{\mathbf{x}})\|^2 . \tag{38.80}$$

ガウス分布に対しては、これは再び $\frac{1}{\sigma^2} - \frac{1}{v+\sigma^2}$ となる。実際に観測される損失には当てはめ自体の誤差も含まれるため、ゼロではなくこの下限と比較すべきである。

またこの下限から、ノイズレベルが非常に小さいときに何が問題になるかもわかる。滑らかな密度を持つデータでは、下限は d/σ^2 のオーダーで $\sigma \to 0$ に伴って増大する。目標 $-\epsilon/\sigma$ が巨大になる一方で、その平均先であるスコアの平均二乗は有限にとどまる。したがって、単一の小さなノイズレベルでは目標の大部分がノイズで占められ、悪条件な回帰問題になってしまう。

Tweedie の公式は、最良のノイズ除去器がノイズ付きの点をどこへ送るかを表す。これをもう 1 回微分すると、ノイズ付きの点の変化に対してその出力がどれだけ速く変化するかがわかり、それがノイズ付加密度の曲がり具合（曲率）の鋭さを制限することが明らかになる。この上界が必要になるのは最後の項（サンプラを特定のクラスへ誘導する箇所）の 1 度だけであるが、事後分布が目の前にある今のうちに数行で証明しておこう。

命題（ノイズ付加密度の曲率） 任意のデータ分布と任意の $\sigma > 0$ に対して、次が成り立つ。

$$\nabla^2 \log p_\sigma(\tilde{\mathbf{x}}) = -\frac{\mathbf{I}}{\sigma^2} + \frac{1}{\sigma^4}\,\mathrm{Cov}\left[\mathbf{x} \mid \tilde{\mathbf{x}}\right], \tag{38.81}$$

したがって、$-\nabla^2 \log p_\sigma$ のいかなる固有値も $1/\sigma^2$ を超えない。

証明 Tweedie の公式（Equation 38.79）より $\nabla \log p_\sigma(\tilde{\mathbf{x}}) = (\mathbb{E}[\mathbf{x} \mid \tilde{\mathbf{x}}] - \tilde{\mathbf{x}})/\sigma^2$ であるから、事後平均 $\mathbb{E}[\mathbf{x} \mid \tilde{\mathbf{x}}] = \int \mathbf{x}\, p(\mathbf{x} \mid \tilde{\mathbf{x}})\, d\mathbf{x}$ を微分すればよい。事後分布はベイズの定理を通じて $\tilde{\mathbf{x}}$ に依存しており、$\nabla_{\tilde{\mathbf{x}}} \log p(\mathbf{x} \mid \tilde{\mathbf{x}}) = \nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) - \nabla \log p_\sigma(\tilde{\mathbf{x}}) = (\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \tilde{\mathbf{x}}])/\sigma^2$ が成り立つ。$\nabla p = p\, \nabla \log p$ を用いて積分記号下で微分すると、事後平均のヤコビ行列は次のようになる。

$$\int \mathbf{x}\, \frac{(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \tilde{\mathbf{x}}])^\top}{\sigma^2}\, p(\mathbf{x} \mid \tilde{\mathbf{x}})\, d\mathbf{x} = \frac{1}{\sigma^2}\,\mathrm{Cov}\left[\mathbf{x} \mid \tilde{\mathbf{x}}\right].$$

ここからヤコビ行列 $\mathbf{I}$（$\tilde{\mathbf{x}}$ のヤコビ行列）を引いて σ^2 で割れば、Equation 38.81 が得られる。共分散は半正定値であるため、求める上界が従う。■

ノイズ付加密度は、加えたノイズ自身よりも鋭く曲がることはできない。すなわち、どのピークもそれをぼかしたガウス分布より細くはならない。ガウスデータの場合、Equation 38.81 の右辺は $-\frac{1}{\sigma^2} + \frac{v}{\sigma^2(v+\sigma^2)} = -\frac{1}{v+\sigma^2}$ となり、確かに $\log \mathcal{N}(0, v+\sigma^2)$ の曲率に一致する。

38.6.1.3 1 次元における最小二乗スコア

ガウス分布のスコアは線形であり、1 パラメータモデルで表現できた。では、ノイズ付きペアに対する最小二乗法は、曲がったスコアも復元できるだろうか。データとして前節の混合分布 $p = \frac{1}{2}\mathcal{N}(-2, 0.5^2) + \frac{1}{2}\mathcal{N}(2, 0.5^2)$ をとり、$\sigma = 0.5$ でノイズを加える。ノイズ付加により各成分の分散に σ^2 が加わるため、p_σ も再び混合分布となり、その成分分散は $0.25 + 0.25 = 0.5$ である。38.5.3 節で導出した通り、平均 $\pm\mu$ と共通の成分分散 v を持つ対称な混合分布のスコアは $(\mu \tanh(\mu x/v) - x)/v$ で与えられる。$\mu = 2$、$v = 0.5$ とすると、目標となる曲線は次の通りである。

$$\nabla \log p_\sigma(\tilde{x}) = 4 \tanh(4\tilde{x}) - 2\tilde{x}.$$

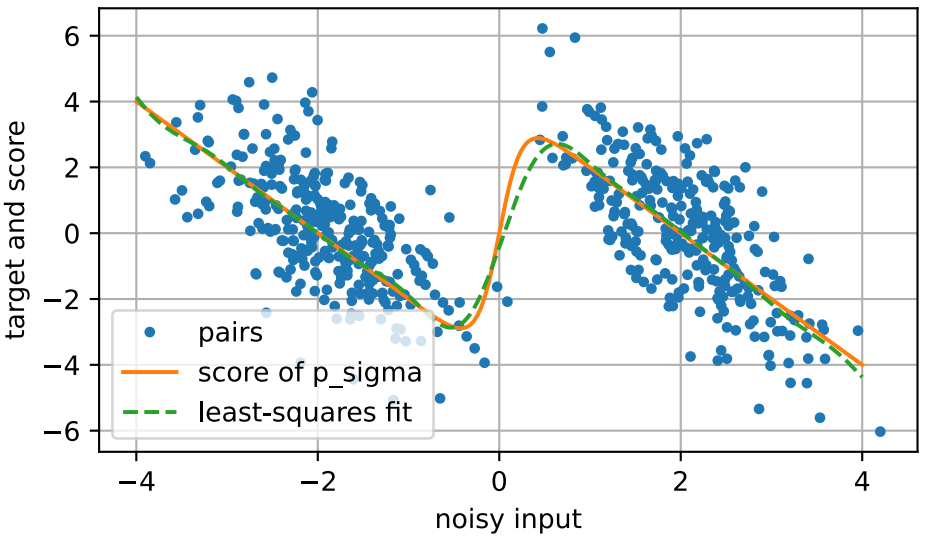
各モードの近傍では、このスコアは単一成分の線形スコア $-(\tilde{x} \mp 2)/0.5$ に近い。モード間では、-2.9 付近（$\tilde{x} = -0.45$）から $+2.9$（$\tilde{x} = 0.45$）へと立ち上がり、谷から外へと点を押し出す。

ここで用いるモデルはネットワークではない。17 個の固定された特徴量、すなわち幅 0.6 で $[-4.5, 4.5]$ 上に中心を並べた 16 個のガウスバンプと恒等関数 x の線形結合である。これを選ぶのは、DSM 損失 Equation 38.77 の当てはめがペア $(\tilde{x}, -\epsilon/\sigma)$ に対する通常の最小二乗法になり、`np.linalg.lstsq` で厳密に解けるからである。当てはめと真のスコアとのずれはすべて標本誤差（sampling error）であり、オプティマイザのせいにはできない。ノイズの抽出にはコストがかからないため、4,096 個の各クリーンな点に 4 回ずつノイズを加える。得られた 16,384 組のペアにモデルを当てはめて解析的なスコアと比較し、さらにデータから遠く離れた $x = 8$ でも両者を評価する。出力される平均二乗誤差は、回帰補題が分解する量 $2J_{\mathrm{DSM}}$ である。

```
n, sigma = 4096, 0.5
def mixture_score(q, var, means=(-2.0, 2.0), ws=(0.5, 0.5)):   # analytic score, any level
    logw = np.stack([np.log(w) - (q - mu)**2 / (2 * var) for mu, w in zip(means, ws)])
```

```
    r = np.exp(logw - logw.max(0))                                  # responsibilities, stable
↪   softmax
    return (r * np.stack([(mu - q) / var for mu in means])).sum(0) / r.sum(0)
def noisy_pairs(n, rng):                                            # n points of p, four noisings
↪   each
    x = np.repeat(rng.normal(4.0 * rng.integers(0, 2, n) - 2.0, 0.5), 4)   # x ~ p, the
↪   mixture
    eps = rng.standard_normal(x.size)
    return x + sigma * eps, -eps / sigma                            # the (input, target) pairs
centers = np.linspace(-4.5, 4.5, 16)
def features(q):                                                    # identity + 16 bumps, width 0.6
    return np.column_stack([q, np.exp(-(q[:, None] - centers)**2 / 0.72)])
def fit(xt, y):                                                     # least squares = the regression
    return np.linalg.lstsq(features(xt), y, rcond=None)[0]
xt, y = noisy_pairs(n, rng)
w_ls = fit(xt, y)
s_hat = lambda q: features(q) @ w_ls
g = np.linspace(-8, 8, 16001); dg = g[1] - g[0]
p_sigma = 0.5 * (np.exp(-(g + 2)**2) + np.exp(-(g - 2)**2)) / np.sqrt(np.pi)  # var 0.5 each
floor = 1 / sigma**2 - (p_sigma * mixture_score(g, 0.5)**2).sum() * dg
excess = (p_sigma * (s_hat(g) - mixture_score(g, 0.5))**2).sum() * dg
print(f'mean squared error: fit {((s_hat(xt) - y)**2).mean():.2f}, exact score '
      f'{((mixture_score(xt, 0.5) - y)**2).mean():.2f}, population floor {floor:.2f}')
print(f'excess risk {excess:.3f}; fit at x = 8: {s_hat(np.array([8.0]))[0]:+.1f}, '
      f'exact score {4 * np.tanh(32) - 16:.0f} (no pairs there)')
grid = np.linspace(-4, 4, 201)
d2l.plot([xt[:600], grid, grid], [y[:600], 4 * np.tanh(4 * grid) - 2 * grid, s_hat(grid)],
         'noisy input', 'target and score', fmts=('.', '-', '--'),
         legend=['pairs', 'score of p_sigma', 'least-squares fit'], figsize=(5.5, 3))
```

```
mean squared error: fit 2.10, exact score 2.11, population floor 2.11
excess risk 0.012; fit at x = 8: +1.6, exact score -12 (no pairs there)
```



プロットには回帰が目にする情報のすべてが現れている。すなわち、目標 $-\epsilon/\sigma$ の点群があらゆ

る $\tilde{x}$ において数単位にわたり広がっている。その点群の真ん中を条件付き平均である p_σ のスコアが貫いており、最小二乗による当てはめはその上に重なっている。出力される数値も回帰補題の通りである。母集団の下限 Equation 38.80、すなわち数値積分で計算した $\frac{1}{\sigma^2} - \mathbb{E}_{p_\sigma}(\nabla \log p_\sigma)^2$ は 2.11 である。これは曲線の周りにおける点群の縦方向の平均二乗の広がりを表す。当てはめと厳密なスコアはいずれもこの下限から 0.1 以内の平均二乗誤差を示し、両者どうしは 0.01 程度の精度で一致する。どちらも 2.11 ぴったりになる必要はない。各値は 1 組のペア標本にわたる平均であり、標本を取り直すごとに変動するからである。当てはめと真のスコアを隔てるのは補題の第 1 項である**過剰リスク**（excess risk）であり、約 0.01 と下限の 1% 未満に収まっている。

最後の数値はこの手法の限界を示している。$|\tilde{x}| \approx 5$ を超える領域にはノイズ付きペアが 1 つも落ちないため、$x = 8$ における当てはめは基底関数がたまたま外挿した値にすぎない。出力値が厳密値 -12 の近くに落ちることもあれば大きく外れることもあり、1 回の当てはめだけではどちらになるか判断できない。そこで、毎回新たなデータと新たなノイズを用いて実験を 20 回繰り返してみよう。1 回の標本抽出で値がどれだけ動くかを見るために厳密なスコアの平均二乗誤差を記録するとともに、2 つの点での当てはめ値を記録する。ペアが豊富に存在する右モードの裾 $x = 1$ と、ペアが皆無である $x = 8$ である。

```
runs = [noisy_pairs(n, rng) for _ in range(20)]                # 20 fresh draws of data and
↪ noise
errs = np.array([((mixture_score(xt_r, 0.5) - y_r)**2).mean() for xt_r, y_r in runs])
W = np.array([fit(xt_r, y_r) for xt_r, y_r in runs])           # (20, 17) fitted weights
print(f'exact score over 20 draws: mean squared error {errs.mean():.2f}, sd
↪ {errs.std():.3f}')
for q in (1.0, 8.0):
    s_q = W @ features(np.array([q]))[0]
    print(f'fit at x = {q:.0f} over 20 draws: from {s_q.min():+.1f} to {s_q.max():+.1f} '
          f'(exact score {4 * np.tanh(4 * q) - 2 * q:+.1f})')
```

```
exact score over 20 draws: mean squared error 2.12, sd 0.047
fit at x = 1 over 20 draws: from +1.9 to +2.1 (exact score +2.0)
fit at x = 8 over 20 draws: from -12.6 to +27.0 (exact score -12.0)
```

20 回の抽出にわたって、厳密なスコアの平均二乗誤差の標準偏差は 0.03〜0.05 となる。したがって、1 組のペア標本から下限を見積もれる精度は小数第 2 位で数単位までが限界である（演習問題 3 では単一の標本から同じ数値を導く）。$x = 1$ では、20 回の当てはめはすべて厳密なスコア $+2.0$ から約 0.2 以内に収まる。一方、$x = 8$ では数十単位にわたってばらつき、正負両方の符号をとる。抽出によっては当てはめたスコアが正となり、データから遠ざかる向きを指すことすらある。回帰はペアを 1 つも観測しなかった場所について何の根拠も持たないにもかかわらず、そこでも平然と数値を返す。最小二乗法はペアが落ちる場所でのみ当てはめを制約するため、データから遠く離れた位置から出発するサンプラは、そこで恣意的なベクトル場に出くわすことになる。

38.6.2 スコアを用いたサンプリング

前節のサンプラは、順過程（forward process）のあらゆるノイズレベルにおけるスコアを必要とした。しかし 1 つのレベルのスコアしかなくても、あるいは p 自身の厳密なスコアしかなくても、順過程を一切必要としない手法によってサンプルを抽出できる。この手法には、1908 年に流体中の粒子を力とランダムな衝撃の和として記述した Paul Langevin の名が冠されている。

38.6.2.1 ランジュバン動力学

対数密度（log-density）の坂を上向きにドリフトしながら、ノイズに揺さぶられる粒子を想像してほしい。ドリフトは粒子を高密度領域へと引き寄せ、ノイズは粒子が単一のモードへ潰れてしまうのを防ぐ。**ランジュバン動力学**（Langevin dynamics）とは、次の確率微分方程式（SDE）である。

$$d\mathbf{X} = \tfrac{1}{2}\nabla \log p(\mathbf{X})\, dt + d\mathbf{W}, \tag{38.82}$$

ここで、ドリフトはスコアの半分であり、ノイズは標準ブラウン運動 $\mathbf{W}$ である。主張したいのは、p がその定常法則（定常分布）になるということである。

ガウス分布 $p = \mathcal{N}(0, v)$（スコアは $-x/v$）については、この事実はすでに知っている。このとき Equation 38.82 はオルンシュタイン・ウーレンベック過程 $dX = -\frac{X}{2v}\,dt + dW$ となる。OU 過程 $dX = -\theta X\,dt + g\,dW$ の定常法則は $\mathcal{N}(0, g^2/(2\theta))$ であり（38.4.3 節）、ここでは $\theta = 1/(2v)$ と $g = 1$ から分散 v が得られる。次の命題は、スコアが高々線形にしか増大しない滑らかな目標分布へとこれを拡張する。

命題（定常性） *p を滑らかな正の密度とし、そのスコア $\nabla \log p$ が局所的リプシッツ連続かつ高々線形にしか増大しないことで、Equation 38.82 が爆発しない一意解を持ち、対応するフォッカー・プランク方程式も各初期密度に対して一意解を持つとする。このとき $\mathbf{X}_0 \sim p$ ならば、$\mathbf{X}_t \sim p$ がすべての $t \geq 0$ に対して成り立つ。*

証明単位拡散を持つ SDE $d\mathbf{X} = \mathbf{b}\,dt + d\mathbf{W}$ に対して、密度 q_t（$\mathbf{X}_t$ の密度）はフォッカー・プランク方程式 $\partial_t q = -\nabla \cdot (q\,\mathbf{b}) + \frac{1}{2}\Delta q$ を満たす（38.5.2 節）。ここに $\mathbf{b} = \frac{1}{2}\nabla \log p$ と $q = p$ を代入し、$p\,\nabla \log p = \nabla p$ を用いると、次のようになる。

$$-\nabla \cdot \big(\tfrac{1}{2}\, p\, \nabla \log p\big) + \tfrac{1}{2}\Delta p = -\tfrac{1}{2}\nabla \cdot (\nabla p) + \tfrac{1}{2}\Delta p = 0.$$

ゆえに $q_t \equiv p$ は初期密度 p に対する方程式の解である。解の一意性により、「p が方程式を満たす」ことから「$\mathbf{X}_t$ の法則は p である」ことが従う。$\blacksquare$

ドリフトが対数密度の坂を上向きに確率質量を運ぶ速度は、ノイズがそれを広げる速度とちょうど釣り合っている。ただし定常性だけでは、別の分布から出発した連鎖についてはまだ何も言えず、実際のサンプラは常に別の場所から出発する。上の命題のような目標分布（滑らかで正であり、スコアが高々線形に増大する分布）に対しては、$\mathbf{X}_t$ の法則は任意の出発点から確かに p へと収束する (Roberts and Tweedie 1996)。ここではその証明は省略する。収束にどれだけの時間がかかるかはまた別の問題であり、以下の実験でそれを測定する。

サンプリングするには離散化する。Equation 38.82 に対するステップサイズ h のオイラー・丸山ステップが**ランジュバンサンプラ**（Langevin sampler）である。

$$\mathbf{x} \leftarrow \mathbf{x} + \tfrac{h}{2}\,\mathbf{s}(\mathbf{x}) + \sqrt{h}\,\xi, \qquad \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

これに必要なのは、厳密または学習されたスコア $\mathbf{s}$ だけである。ただし離散化された連鎖は、元の SDE の定常性をそのまま受け継ぐわけではない。ガウス分布の場合、1 ステップは線形漸化式 $x \leftarrow (1 - \frac{h}{2v})\,x + \sqrt{h}\,\xi$ となり、その定常分散 V は $V = (1 - \frac{h}{2v})^2 V + h$ を満たす。したがって $V = v/(1 - h/(4v))$ となり、サンプラは分散を約 $h/4$ だけ過大評価する（演習問題 1 でこれを測定する）。

ガウス分布は単峰性であり、モードが 1 つしかないと本質的な困難が隠れてしまう。そこで、38.5.5 節の非対称な混合分布 $p = 0.8\,\mathcal{N}(-2, 0.5^2) + 0.2\,\mathcal{N}(2, 0.5^2)$ をサンプラに与

え、さらに当てはめではなく p の厳密なスコアを渡して、問題が起きればすべてサンプラ自身のせいであるようにしよう。p の確率質量の 5 分の 1 はゼロより右側にあり、重みによらず $\mathbb{E}X^2 = 4 + 0.25 = 4.25$ である。10,000 本の連鎖をステップサイズ $h = 0.01$ で 2,000 ステップ走らせる実験を、広い初期分布 $\mathcal{N}(0, 3^2)$ から、左の平均から、そして右の平均からの 3 通りで行う。出力を読む前に、広い初期分布から出発した連鎖のうちどれだけの割合がゼロより右側に落ち着くか、また一方のモードから出発した連鎖が何本もう一方のモードへ移るかを予想してみてほしい。

```
def langevin(q, h, steps, var=0.25, ws=(0.8, 0.2)):   # var: component variance of p
    for _ in range(steps):
        xi = rng.standard_normal(q.shape)
        q = q + 0.5 * h * mixture_score(q, var, ws=ws) + np.sqrt(h) * xi
    return q
warm = langevin(rng.normal(0.0, 3.0, 10000), 0.01, 2000)
print(f'broad start: P(X > 0) = {(warm > 0).mean():.3f}, '
      f'E[X^2] = {(warm**2).mean():.2f}  (truth 0.2, 4.25)')
for start, w in ((-2.0, 0.8), (2.0, 0.2)):              # every chain in one mode
    q = langevin(np.full(10000, start), 0.01, 2000)
    print(f'all chains at {start:+.0f} (weight {w}): '
          f'{(np.sign(q) != np.sign(start)).mean():.3f} end on the other side')
```

```
broad start: P(X > 0) = 0.476, E[X^2] = 4.23  (truth 0.2, 4.25)
all chains at -2 (weight 0.8): 0.005 end on the other side
all chains at +2 (weight 0.2): 0.020 end on the other side
```

10,000 本のうち重いモードから出発した連鎖で反対側に到達したのは、およそ 200 本に 1 本にすぎない。軽いモードから出発した連鎖でも、およそ 40〜50 本に 1 本である。谷では密度がごく小さく、ドリフトは連鎖がもともといたモードの側へと引き戻す向きを指す。谷を越えるには一方向にノイズが連続するというまれな事象が必要であり、その待ち時間は $-\log p$ の障壁の高さに対して指数関数的に増大する。

2 つの割合が異なるのは当然である。平衡状態では左から右へ流れる質量と右から左へ流れる質量が等しくなるため、重いモードの 4 分の 1 の質量しか持たない軽いモードは、4 倍の頻度で連鎖を手放さなければならない。サンプラの遷移率の比はおおむね正しい。問題は、その絶対値があまりにも小さいことである。2 つの状態を持つ系が初期比率を忘れる速さは 2 つの遷移率の和で決まるが、出力された 2 つの割合を足しても 1 回の実行あたり約 0.03 にしかならない。したがって、比率の誤差が e 分の 1 に縮むには $2{,}000/0.03 \approx 67{,}000$ ステップ、すなわち今回の実行長の 30 倍以上を要する。

これを踏まえると、1 行目の出力の見方が変わる。二次モーメントは 4.25 に小数第 2 位で数単位の精度で一致しており、これは 10,000 本の連鎖の標本誤差の範囲内である（ステップサイズによる統計的バイアス $h/4 \approx 0.0025$ はそれよりはるかに小さい）。各連鎖は、自分が滑り込んだモードの内部では平衡に達している。しかし、ほぼ半数の連鎖がゼロより右側に落ち着いており、そこには本来 p の質量の 5 分の 1 しかない。初期分布が対称であるため、各連鎖は谷に対して自分がいた側のモードへと滑り込み、そこからほとんど出ていかない。結果の比率は初期分布の比率であって、p の比率ではないのである（17.1 節では、3 つのモード上で学習されたスコアを用いて同じ不感性を測定する）。

連鎖がどこから出発したかを忘れるまでに要する時間を**混合時間**（mixing time）と呼ぶ。混合の遅さは SDE 自身の性質であるため、ステップサイズを小さくしても解決しない。ステップサイズを小さくして取り除けるのは、離散化による統計的バイアスだけである。

38.6.2.2 単一ノイズレベルの限界

単一ノイズレベルのスコアは、ここまでで 3 つの失敗を見せた。第 1 に、ノイズ付きペアが落ちなかった領域では、学習されたスコアは恣意的になる。第 2 に、正しいスコアを与えても、ランジュバン動力学はモード間の谷をほとんど越えられない。第 3 に、得られるサンプルは初期分布の比率を引き継ぎ、5 分の 1 しか質量がないはずのモードにほぼ半分の質量が集まってしまう。3 つ目の失敗はサンプラだけの責任ではない。よく離れたモードからなる混合分布が質量を持つ領域では、重みが何であれ、そのスコアは最も近いモード単独のスコアとほぼ変わらない（37.5.1 節）。すなわち低ノイズでは、スコア自体に比率の情報がほとんど刻まれていないのである。

38.5.5 節の逆時間サンプラは、同じ非対称な混合分布に対して同様に対称な初期分布から出発しながら、質量の 5 分の 4 を正しく左側へ戻した。このサンプラは、ランジュバン動力学に欠けているもの、すなわちあらゆるノイズレベルにおけるスコアを持っていたのである。比率が決着したのは中程度のノイズ領域であった。そこでは 2 つの成分がまだ重なり合いつつ引き寄せる力にはすでに差が生じているため、混合重みが粒子の行き先を決定する。

減少するノイズレベルの系列（sequence）を用いれば、3 つの限界すべてに対処できる。大きな σ ではノイズ付加密度に谷がなく、ノイズ付きペアはサンプラが訪れる領域全体を覆う。対称な混合分布の場合、p_σ の成分分散は $0.25 + \sigma^2$ であり、成分の標準偏差が中心から各平均までの距離 2 に達した時点（前節の演習問題 4）、すなわち $\sigma \geq 1.94$ で単峰性になる。重みが不均等であれば、谷はさらに早く消える。そこから σ が縮むにつれて細部の構造が戻り、すでに両モードへ質量を配分し終えたサンプラはその分布を精緻化するだけでよい。

アニーリング付きランジュバン動力学（annealed Langevin dynamics）は、各レベルに専用のスコアを用意してこのような系列に沿ってサンプラを走らせる手法である (Song and Ermon 2019)。演習問題 2 ではこの方法で単一モードに捕らわれた連鎖を解放し、17.2 節ではこのような系列が不均等な重みを復元することを示す。しかしノイズレベルを 1 つずつ選ぶやり方には、レベル間隔をどう決めるか、あるレベルから次のレベルへどう移るかという 2 つの問いが残る。順過程はこの両方を一挙に解決する。連続的なノイズレベルを供給し、その逆過程がそのままサンプラになるからである。

38.6.3 複数ノイズレベルにわたるスコア

ここから先、モデルはノイズレベルを第 2 引数として受け取る単一のベクトル場 $\mathbf{s}_\theta(\mathbf{x}, t)$ となり、次項で導入する損失によってすべてのレベルで同時に訓練される。サンプラは冒頭で挙げた 2 つ、すなわち 38.5.5 節の逆時間 SDE と 38.5.4.3 節の確率フロー ODE である。

38.6.3.1 順過程とノイズ条件付き損失

DSM にはクリーンな点が与えられたもとでのノイズ付きの点の条件付き法則が必要であり、目標が陽に書けるためにはその法則がガウス分布でなければならない。ドリフトが $\mathbf{x}$ に関して線形であり、ノイズの振幅が $\mathbf{x}$ に依存しないすべての SDE は、38.4.3 節で OU 過程について示した通りガウス遷移カーネルを持つ。38.4 節の 2 つの標準的な順過程はいずれもこの種のものであり、そのカーネルは両者共通の形で次のように書ける。

$$p_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\alpha_t \mathbf{x}_0,\, \sigma_t^2 \mathbf{I}), \qquad \text{i.e.} \qquad \mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon.$$

係数 α_t は信号を縮小し、σ_t はノイズのスケールを定める。2 つの過程の違いは、この 2 つの数値だけである。

- **分散爆発**（variance-exploding、VE）SDE $d\mathbf{X} = \sqrt{\frac{d}{dt}\sigma^2(t)}\, d\mathbf{W}$（ただし $\sigma^2(t)$ は単調増加かつ $\sigma(0) = 0$）は、データを縮小せずにノイズを加える過程であり、$\alpha_t = 1$、$\sigma_t = \sigma(t)$

となる。これはアニーリング付きランジュバン動力学が降りていくノイズレベルの系列を連続化したものである。

- **分散保持**（VP）SDE は時間依存のレートを持つオルンシュタイン・ウーレンベック過程

$$d\mathbf{X} = -\tfrac{1}{2}\beta(t)\,\mathbf{X}\,dt + \sqrt{\beta(t)}\,d\mathbf{W},$$

であり、$\alpha_t = e^{-\frac{1}{2}\int_0^t \beta(r)\,dr}$ かつ $\sigma_t^2 = 1 - \alpha_t^2$ となるため、単位分散のデータは任意の時刻 t で単位分散を保つ（38.4 節）。

本節の残りの部分では VP 過程を用いる。VE 過程は、DDIM が単純な形になる座標系としてあとで 1 度だけ再登場する。

任意の t でカーネルがガウス分布であるため、条件付きスコアは $-\epsilon/\sigma_t$ となり、各レベルで DSM がそのまま適用できる。重み $w(t) > 0$ をかけてレベル全体にわたり平均をとると、**ノイズ条件付き DSM 損失**（noise-conditional DSM loss）が得られる。

$$\mathcal{L}(\theta) = \tfrac{1}{2}\,\mathbb{E}_t\, w(t)\, \mathbb{E}_{\mathbf{x}_0,\,\epsilon}\left[\left\| \mathbf{s}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \epsilon, t) + \frac{\epsilon}{\sigma_t} \right\|^2\right], \tag{38.83}$$

ここで t は訓練範囲から一様に抽出する。$(t, \mathbf{x}_0, \epsilon)$ の 1 回の抽出が 1 組の回帰ペアになる。本節の以降ではこの損失を用いる。以下で述べる重みの選び方は、初読時には細部として読み飛ばして差し支えない。

重みが決めるのは、ネットワークが何を学習するかではなく、どのレベルに最も計算資源を割くかである。任意の正の重み w に対し、ネットワークに十分な表現力があれば、各レベルでの Vincent の定理により母集団における最小点は $\mathbf{s}(\cdot, t) = \nabla \log p_t$（すべての t に対して）となる。重みが変えるのは、各レベルの寄与の大きさである。Equation 38.80 より、レベル t における下限は $d/\sigma_t^2 - \mathbb{E}\|\nabla \log p_t\|^2$ であり、これは d/σ_t^2 のオーダーで小さな σ_t において増大する。したがって、重みなしの損失は最もノイズの小さいレベルに支配されてしまう。

標準的な重み付けは 2 つある。次項の離散時間モデルは $w(t) = \sigma_t^2$ を用い、この増大を打ち消すことで、すべてのレベルが同じオーダーの損失を寄与するようにする。

もう 1 つの**尤度重み付け**（likelihood weighting）$w = g^2$（順方向 SDE の拡散係数の二乗）は、スコアの誤差が尤度を制御するようになる重みである。この重みのもとでのスコアの二乗誤差を全レベルにわたって積分すると、逆時間 SDE が定めるモデルに対するデータからの KL ダイバージェンスを上から抑えられる。ただし、p_T とサンプラの出発点であるガウス分布との不一致を表す項が別途加わる（演習問題 7）。実際に計算できる損失にはこの誤差に加えて下限が含まれるが、尤度重み付けのもとではこの下限が厄介になる。VP 過程では $g^2 = \beta$ であり、重み付き下限は d/t のオーダーで $t \to 0$ に伴って増大するため、$t = 0$ までの積分が無限大に発散する。そのため、訓練は小さな時刻 $\varepsilon > 0$ から開始する（演習問題 4）。

生成には新たな仕組みは何もいらない。厳密なスコアの代わりに学習されたベクトル場を用いて、2 つのサンプラをそのまま動かせばよい。順方向 SDE $d\mathbf{X} = \mathbf{f}(\mathbf{X}, t)\,dt + g(t)\,d\mathbf{W}$（区間 $[0, T]$ 上）に対し、p_T を近似するガウス分布から出発して、逆時間 SDE（38.5.5 節）または確率フロー ODE（38.5.4.3 節）を $t = T$ から 0 まで逆向きに積分する。

$$d\mathbf{X} = \big[\mathbf{f} - g^2\,\mathbf{s}_\theta(\mathbf{X}, t)\big]\,dt + g\,d\bar{\mathbf{W}}, \qquad \frac{d\mathbf{x}}{dt} = \mathbf{f} - \tfrac{1}{2} g^2\,\mathbf{s}_\theta(\mathbf{x}, t), \tag{38.84}$$

ここで、$\bar{\mathbf{W}}$ は時間を逆向きに走るブラウン運動である。VP 過程では $\mathbf{f} = -\frac{1}{2}\beta\mathbf{x}$ および $g^2 = \beta$ となる。

こうして生成モデルは、順過程・学習されたスコア・サンプラの 3 つ組として定式化される。Figure 38.24 は 2 次元でこの 3 要素が働く様子を示したものであり、Figure 38.21 の順方向・逆方向密度と対をなす図である。

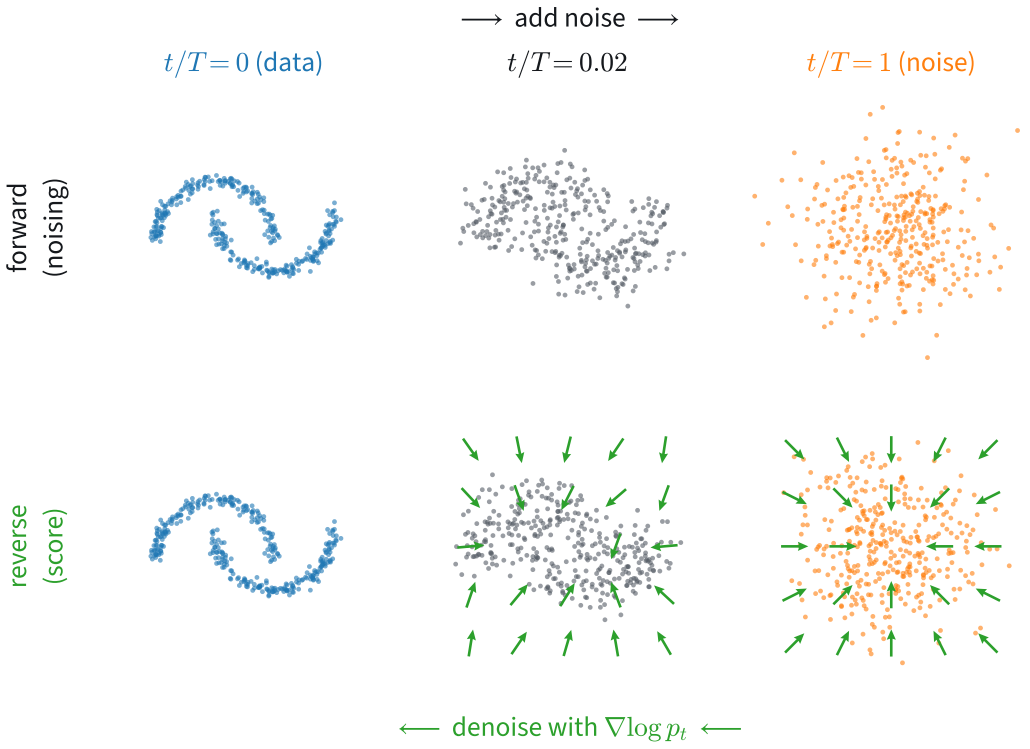


Figure 38.24: VP 順過程のもとでの 2 つの三日月（two-moons）分布（上段左から右へ $t/T = 0$、0.02、1）は、等方的なガウス分布へと近づいていく。下段は同じ周辺分布を右から左へ逆向きにたどり、短い矢印はノイズ付加データの厳密なスコア $\nabla \log p_t$ の向きを表す。高ノイズではスコアは原点への引力となり、低ノイズではデータへ向かう向きを指す。

38.6.3.2 DDPM と分散保持 SDE

ノイズ除去拡散確率モデル（denoising diffusion probabilistic model、DDPM）は離散時間で記述されており、一見すると別の構成法のように見える (Ho et al. 2020)。このモデルはノイズ付加連鎖・損失・サンプラの 3 要素からなるが、そのどれもがすでに手元にある概念に帰着する。ノイズ付加連鎖は $N = 1000$ ステップからなるマルコフ連鎖であり、スケジュール $\beta_1, \ldots, \beta_N \in (0,1)$ を用いて次のように定義される。

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ i.i.d.}, \tag{38.85}$$

各ステップで状態をわずかに縮小し、少量のノイズを加える。本項と次項では、t はレベル $t = 0, 1, \ldots, N$ を数える添字とし、レベル t は SDE の時刻 $t\,\Delta$（ただし $\Delta = T/N$）を表すものとする。

この連鎖は VP-SDE とどう関係しているだろうか。$\beta_t = \beta((t-1)\Delta)\,\Delta$ とおくと、1 ステップは β_t の 1 次の範囲で VP-SDE のオイラー・丸山ステップ $(1 - \tfrac{1}{2}\beta_t)\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\epsilon_t$ に一致する。実はそれ以上のことが成り立ち、定数レートの場合については 38.4.4.1 節で示した通りである。すなわち、このステップはほぼ同じ長さの区間における VP 過程の**厳密な**遷移になっている。任意の 2 時刻間における VP カーネルは、時刻 0 からの式で β の積分範囲をその 2 時刻間に置き換えた形をとる。そこで $\int \beta(r)\,dr = -\log(1-\beta_t)$ となる区間をとろう。そのカーネルの平均係数は $e^{-\frac{1}{2}\int \beta} = \sqrt{1-\beta_t}$、分散は $1-(1-\beta_t) = \beta_t$ となり、Equation 38.85 の $\mathbf{x}_{t-1}$ が与えられたもとでの法則に一致する。標準スケジュールの最大ステップ $\beta_t = 0.02$ に対しても、この区間の積分値は $\int \beta = 0.0202$ であり、0.02 とほとんど変わらない。したがって DDPM の連鎖は離散時刻で観測された VP 過程にほかならず、$\mathbf{x}_t$ の（$\mathbf{x}_0$ が与えられたもとでの）法則は時刻 0 からの VP カーネルとなる。これこそが本項で記憶すべき結果であり、DDPM の訓練損失はここから 2 行で導かれる。なお、SDE を経由せずとも、DDPM 自身の記法のまま連鎖上で直接この周辺分布を確かめることもできる。

命題（$\bar\alpha$ 周辺分布） $\bar\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$ **とおく。このとき、$\mathbf{x}_0$ が与えられた条件のもとで次が成り立つ。**

$$\mathbf{x}_t = \sqrt{\bar\alpha_t}\,\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{38.86}$$

証明 t に関する数学的帰納法による。$t = 0$ の場合は $\bar\alpha_0 = 1$ として成り立つ。$t-1$ で Equation 38.86 が成り立つと仮定し、Equation 38.85 に代入すると次のようになる。

$$\mathbf{x}_t = \sqrt{(1-\beta_t)\bar\alpha_{t-1}}\,\mathbf{x}_0 + \sqrt{(1-\beta_t)(1-\bar\alpha_{t-1})}\,\epsilon' + \sqrt{\beta_t}\,\epsilon_t,$$

ここで ϵ' と ϵ_t は独立な標準ガウス確率変数である。独立なガウス分布の和は分散の和を持つガウス分布になり、$(1-\beta_t)(1-\bar\alpha_{t-1}) + \beta_t = 1-(1-\beta_t)\bar\alpha_{t-1} = 1-\bar\alpha_t$ かつ $(1-\beta_t)\bar\alpha_{t-1} = \bar\alpha_t$ が成り立つ。■

前項のカーネルの記法で言えば、DDPM は $\alpha_t = \sqrt{\bar\alpha_t}$、$\sigma_t = \sqrt{1-\bar\alpha_t}$ に対応する。したがって訓練時に連鎖を逐次シミュレーションすることは一切ない。Equation 38.86 によって任意のレベルへ 1 ステップで飛び移ることができ、カーネルがガウス分布であるため DSM がそのまま適用でき、条件付きスコアは $-\epsilon/\sigma_t$ となる。ノイズ予測ネットワークを用いてスコアモデルを $\mathbf{s}_\theta(\mathbf{x}_t, t) = -\epsilon_\theta(\mathbf{x}_t, t)/\sigma_t$ とパラメータ化すると、次が成り立つ。

$$\left\| \mathbf{s}_\theta(\mathbf{x}_t, t) + \frac{\epsilon}{\sigma_t} \right\|^2 = \frac{1}{\sigma_t^2}\,\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2,$$

ゆえに DDPM の**単純損失**（simple loss）$\tfrac{1}{2}\mathbb{E}_{t,\mathbf{x}_0,\epsilon}\|\epsilon-\epsilon_\theta(\mathbf{x}_t,t)\|^2$ は、重みを $w(t) = \sigma_t^2 = 1-\bar\alpha_t$ としたノイズ条件付き DSM 損失 Equation 38.83 に一致する。「加えたノイズを予測せよ」とは、下限の $1/\sigma_t^2$ の増大を打ち消す重みを付けたノイズ除去スコアマッチングのことであり、どのレベルもオーダー d の損失を寄与する（(Ho et al. 2020) を読む際の注意として、原論文では α_t を $1-\beta_t$ の意味で書いており、本節の α_t が信号係数 $\sqrt{\bar\alpha_t}$ を表すのと異なるほか、原論文の単純損失には本節の係数 $\tfrac{1}{2}$ が付かない）。

第 3 の要素はサンプラである。**祖先サンプリング**（ancestral sampling）は、レベル N から 1 まで順にステップを進める。

$$\mathbf{x}_{t-1} = \frac{\mathbf{x}_t + \beta_t\,\mathbf{s}_\theta(\mathbf{x}_t, t)}{\sqrt{1-\beta_t}} + \sqrt{\beta_t}\,\xi, \qquad \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

ただし最後のステップではノイズを加えない。β_t の 1 次の範囲では、これは $\mathbf{x}_t + \frac{1}{2}\beta_t \mathbf{x}_t + \beta_t \mathbf{s}_\theta + \sqrt{\beta_t}\,\xi$ となり、VP 過程に対する Equation 38.84 の逆 SDE のオイラー・丸山ステップに一致する。なお、このモデルはもともと別の経路、すなわち潜在変数モデルの対数尤度に対する下界として導出された (Sohl-Dickstein et al. 2015; Ho et al. 2020)。17.6 節でたどるのがその経路であり、各レベルの重みが異なるだけで同じ回帰が得られる。

以下のサンプラでは、β_t を 10^{-4} から 0.02 まで $N = 1000$ レベルにわたり線形に変化させる標準スケジュールを用いる。VP 過程は単位分散のデータ向けに設計されているため、ここから先は混合分布を**標準化**する。すなわち $\sqrt{4.25}$ で割り、モードを $\pm 2/\sqrt{4.25} = \pm 0.970$、成分の標準偏差を 0.243 とする。サンプリングを行う前に、まずスケジュール自体を確認しておこう。3 つのレベルにおける $\bar{\alpha}_t$ と、最初および最後のレベルにおける信号に対するノイズの比 $\sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$（次項で時間変数として用いる量）を出力する。

```
N = 1000
beta = np.linspace(1e-4, 0.02, N)                       # the DDPM schedule
alpha_bar = np.cumprod(1.0 - beta)
sigma_bar = np.sqrt((1 - alpha_bar) / alpha_bar)        # noise scale of x_t / sqrt(alpha_bar)
print(f'alpha_bar at t = 10, 100, 1000: {alpha_bar[9]:.3f}, {alpha_bar[99]:.3f}, '
      f'{alpha_bar[-1]:.0e}; sigma_bar from {sigma_bar[0]:.2f} to {sigma_bar[-1]:.0f}')
```

```
alpha_bar at t = 10, 100, 1000: 0.998, 0.897, 4e-05; sigma_bar from 0.01 to 157
```

信号は最初はゆっくりと薄れ、最後には完全に消え去る。$t = 100$ では $\bar{\alpha} = 0.897$ であり、ノイズ付加後の法則は依然として二峰性を保ち、モードは $\pm\sqrt{0.897}\cdot 0.970 = \pm 0.92$、成分の標準偏差は $\sqrt{0.897 \cdot 0.243^2 + 0.103} = 0.39$ である。$t = N$ では $\bar{\alpha}_N \approx 4 \times 10^{-5}$ となり、元のデータが何であったかにかかわらずサンプルは標準ガウス分布に従う。最初のレベルから最後のレベルまでの間に、信号に対するノイズの比は 0.01 から 157 へと 4 桁増大する。

38.6.4 決定論的サンプリングとガイダンス

祖先サンプリングは 1 レベルにつき 1 回、1 サンプルあたり $N = 1000$ 回ネットワークを呼び出す。同じネットワークを使って、はるかに少ない呼び出し回数でサンプルを生成できないだろうか。DDIM はそれを実現しており、その理由は ODE ソルバに関するある事実にある。まず厳密なスコアの上でその事実を証明し、次に当てはめたスコアの上で DDIM を 1 度走らせ、最後にサンプラを特定のクラスへ誘導したとき何を追跡しているのかを調べる。

38.6.4.1 ノイズ除去拡散陰的モデル

確率フロー ODE は決定論的かつ滑らかであるため、粗いレベルのグリッドでも十分に積分できるはずである。ノイズ除去拡散陰的モデル（denoising diffusion implicit model、DDIM）は、訓練済みのノイズ予測ネットワークをそのまま再利用しつつレベルを飛び越える更新則によってこれを実現する (J. Song et al. 2021)。レベル t において、ネットワークのノイズ推定値から周辺分布 Equation 38.86 を逆算することで、クリーンサンプルの推定値が次のように得られる。

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}.$$

より前の任意のレベル $t' < t$ へ移るため、DDIM はこの推定値と**同じ**ノイズ推定値から、レベル t' の係数を用いてサンプルを再構成する。

$$\mathbf{x}_{t'} = \sqrt{\bar{\alpha}_{t'}}\,\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t'}}\,\epsilon_\theta(\mathbf{x}_t, t). \tag{38.87}$$

この更新は決定論的であり、新たなノイズは一切入らない。文字通りに読めば、この式は順方向の曲線 $\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$（ペア $(\hat{\mathbf{x}}_0, \epsilon_\theta)$ が定める曲線）に沿ってサンプルを動かす。しかし最小二乗法で訓練されたネットワークが返すのは条件付き平均 $\mathbb{E}[\epsilon \mid \mathbf{x}_t]$ であって、そのサンプルを生成した実際のノイズではなく、しかもその推定値はレベルごとに変化する。では DDIM は一体何を積分しており、長いストライドをとるとどれほどの誤差が生じるのだろうか。

座標変換によって両方の問いに答えが出る。全体を $\sqrt{\bar{\alpha}_t}$ で割ってみよう。再スケーリングされた点 $\bar{\mathbf{x}}_t = \mathbf{x}_t/\sqrt{\bar{\alpha}_t}$ は $\mathbf{x}_0 + \bar{\sigma}_t\epsilon$ に等しく、ここで $\bar{\sigma}_t = \sqrt{(1-\bar{\alpha}_t)/\bar{\alpha}_t}$ は先ほどスケジュールとともに出力した信号に対するノイズの比である。この座標系では VP 過程は VE 過程、すなわちクリーンなデータにスケール $\bar{\sigma}$ のノイズを加えたものになり、そのスケールは最初のレベルの 0.01 から最後のレベルの 157 まで増大する。スケジュールが決めるのは、グリッドがどの $\bar{\sigma}$ の値を訪れるかだけである。なお $\bar{\mathbf{x}}_t$ と $\bar{\sigma}_t$ のバーは再スケーリングされた点とそのノイズレベルを表す記号であり、$\bar{\alpha}_t$ のバーは DDPM 由来の記法であって別の意味を持つ。

命題（DDIM はオイラー法である） 座標 $(\bar{\sigma}, \bar{\mathbf{x}})$ において、DDIM の更新（Equation 38.87）は次の形になる。

$$\bar{\mathbf{x}}_{t'} = \bar{\mathbf{x}}_t + (\bar{\sigma}_{t'} - \bar{\sigma}_t)\,\epsilon_\theta(\mathbf{x}_t, t), \tag{38.88}$$

これは常微分方程式（ODE）$d\bar{\mathbf{x}}/d\bar{\sigma} = \epsilon_\theta$ の 1 回のオイラーステップである。もし ϵ_θ が厳密なノイズ予測 $\hat{\epsilon}(\mathbf{x}_t, t) = \mathbb{E}[\epsilon \mid \mathbf{x}_t]$ であれば、この ODE は座標 $(\bar{\sigma}, \bar{\mathbf{x}})$ で書かれた VP 過程の確率フロー ODE に一致する。

証明ステップ (i) は単なる代数変形であり、実質的な内容は確率フロー ODE が新しい座標でこの形をとることを示す (ii) と (iii) にある。(ii) において、t は SDE の連続時間を表す。

(i) Equation 38.87 を $\sqrt{\bar{\alpha}_{t'}}$ で割り、$\hat{\mathbf{x}}_0 = \bar{\mathbf{x}}_t - \bar{\sigma}_t\epsilon_\theta$ を代入すると、$\bar{\mathbf{x}}_{t'} = \hat{\mathbf{x}}_0 + \bar{\sigma}_{t'}\epsilon_\theta = \bar{\mathbf{x}}_t + (\bar{\sigma}_{t'} - \bar{\sigma}_t)\epsilon_\theta$ となる。

(ii) Equation 38.84 の VP 確率フロー ODE $d\mathbf{x}/dt = -\frac{1}{2}\beta\mathbf{x} - \frac{1}{2}\beta\nabla_{\mathbf{x}}\log p_t$ において、$\alpha_t = \sqrt{\bar{\alpha}(t)}$、$\bar{\alpha}(t) = e^{-\int_0^t \beta}$ として変数変換する。$\bar{p}_t$ を $\bar{\mathbf{x}}_t$ の密度とすると、$\nabla_{\mathbf{x}}\log p_t = \nabla_{\bar{\mathbf{x}}}\log \bar{p}_t/\sqrt{\bar{\alpha}}$ である。$d\sqrt{\bar{\alpha}}/dt = -\frac{1}{2}\beta\sqrt{\bar{\alpha}}$ より縮小項どうしが打ち消し合い、$d\bar{\mathbf{x}}/dt = -\frac{\beta}{2\bar{\alpha}}\nabla\log\bar{p}_t$ となる。また $\bar{\sigma}^2 = 1/\bar{\alpha} - 1$ より $d\bar{\sigma}/dt = \beta/(2\bar{\alpha}\bar{\sigma})$ であり、両式の比をとれば $d\bar{\mathbf{x}}/d\bar{\sigma} = -\bar{\sigma}\,\nabla\log\bar{p}(\bar{\mathbf{x}})$ が得られる。

(iii) カーネル $\mathcal{N}(\mathbf{x}_0, \bar{\sigma}^2\mathbf{I})$ に対する Tweedie の公式は $\mathbb{E}[\mathbf{x}_0 \mid \bar{\mathbf{x}}] = \bar{\mathbf{x}} + \bar{\sigma}^2\nabla\log\bar{p}$ であるため、$\mathbb{E}[\epsilon \mid \bar{\mathbf{x}}] = (\bar{\mathbf{x}} - \mathbb{E}[\mathbf{x}_0 \mid \bar{\mathbf{x}}])/\bar{\sigma} = -\bar{\sigma}\nabla\log\bar{p}$ となり、(ii) の右辺に一致する。$\bar{\mathbf{x}}_t$ による条件付けは $\mathbf{x}_t$ による条件付けと同じである。■

したがって DDIM は、ノイズレベルを時間とする時計のもとでの、確率フロー ODE に対する 1 次ソルバである。その誤差はオイラー法の誤差にほかならない。すなわち、1 回のストライドは ϵ_θ を一定に固定する一方、真の $\hat{\epsilon}$ は軌跡に沿って変化する。次の 2 つの系は、この誤差の両極端を示す。

系（点質量は誤差なく積分される） データが単一の点 $\mathbf{x}_0$ であるならば、厳密な $\hat{\epsilon}$ を用いた DDIM のストライドは、その長さにかかわらず確率フローの軌跡上に厳密に着地する。この場合 $\hat{\epsilon} = (\bar{\mathbf{x}} - \mathbf{x}_0)/\bar{\sigma}$ であり、$d\bar{\mathbf{x}}/d\bar{\sigma} = (\bar{\mathbf{x}} - \mathbf{x}_0)/\bar{\sigma}$ の解に沿って比 $(\bar{\mathbf{x}} - \mathbf{x}_0)/\bar{\sigma}$ は一定に保たれる。右辺が定数である微分方程式に対するオイラーステップは厳密である。

系（ガウスデータは縮小する） データ $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ に対して厳密なフローは恒等写像であるが、厳密なノイズ予測を用いた 1 回の DDIM ストライドはすべてのサンプルに次の因子を掛ける。

$$\sqrt{\bar{\alpha}_t \bar{\alpha}_{t'}} + \sqrt{(1-\bar{\alpha}_t)(1-\bar{\alpha}_{t'})} = \cos(\phi_t - \phi_{t'}), \qquad \bar{\alpha}_t = \cos^2 \phi_t,$$

正の長さを持つあらゆるストライドに対して、この値は 1 より小さい。VP 過程はこのようなデータをそのまま保つ。どのレベルでも周辺分布は $\mathcal{N}(\mathbf{0}, \mathbf{I})$、スコアは $-\mathbf{x}$ であり、確率フローのドリフト $-\frac{1}{2}\beta\mathbf{x} - \frac{1}{2}\beta(-\mathbf{x})$ は 0 になるため、厳密なフローのサンプルはまったく動かない。Tweedie の公式より $\hat{\epsilon}(\mathbf{x}_t, t) = \sqrt{1-\bar{\alpha}_t}\,\mathbf{x}_t$ であり、これを Equation 38.87 に代入すると上の因子が得られる。$\bar{\alpha}_t = \cos^2 \phi_t$ とおけば、これは進んだ角度の余弦になる。バー付きの座標では厳密解が $\bar{\mathbf{x}} \propto \sqrt{1+\bar{\sigma}^2}$ となるため、解に沿って $\hat{\epsilon} = \bar{\sigma}\bar{\mathbf{x}}/(1+\bar{\sigma}^2)$ は変化する。ストライドの間これを固定してしまうことで、上の因子の分だけ信号を取り除きすぎてしまうのである。

上の系は 1 ストライドあたりの縮小率を与えている。サンプリング実行全体ではこの因子が掛け合わされるため、DDPM スケジュールを等間隔にとった 10、50、999 ストライドについてその積を計算してみよう（999 ストライドは 1,000 個のレベルを 1 つずつ進むことに対応する）。ガウス分布ではないデータでは、この誤差がどれほど残るだろうか。これを確かめるため、標準化した混合分布に対して厳密なノイズ予測 $\hat{\epsilon} = -\sqrt{1-\bar{\alpha}_t}\,\nabla \log p_t$ を用いて DDIM を走らせる。ここでレベル t におけるノイズ付加分布のモードは $\pm\sqrt{\bar{\alpha}_t}\,m$、成分分散は $\bar{\alpha}_t v_0 + 1 - \bar{\alpha}_t$ であり、クリーンな値は $m = 0.970$、$v_0 = 0.243^2$ である。同じ初期ノイズから出発した 10 ストライドおよび 50 ストライドの結果を 999 ストライドの結果と 2 通りの指標で比較する。対応する終点間の平均距離と、2 標本の経験分布関数（empirical distribution function）間の最大乖離であるコルモゴロフ・スミルノフ（KS）統計量である。

```
m, v0 = 2 / np.sqrt(4.25), 0.25 / 4.25                # standardized: modes +-m, variance v0
def ks(a, b):                                        # two-sample Kolmogorov-Smirnov statistic
    q = np.sort(np.concatenate([a, b]))
    return np.abs(np.searchsorted(np.sort(a), q, 'right') / len(a)
                  - np.searchsorted(np.sort(b), q, 'right') / len(b)).max()
def strides(K):                                       # K equal strides from level N to level 1
    return np.linspace(N - 1, 0, K + 1).round().astype(int)
for K in (10, 50, 999):                                 # the Gaussian corollary
    a = alpha_bar[strides(K)]
    print(f'N(0,1) data, {K:4d} strides: every sample times '
          f'{np.prod(np.sqrt(a[:-1] * a[1:]) + np.sqrt((1 - a[:-1]) * (1 - a[1:]))):.3f}')
def eps_hat(xt, a, ws=(0.5, 0.5)):                    # exact noise prediction of the mixture
    means = (-np.sqrt(a) * m, np.sqrt(a) * m)
    return -np.sqrt(1 - a) * mixture_score(xt, a * v0 + 1 - a, means, ws)
def ddim(z, K, eps_fn):                                 # K DDIM strides from noise z
    xt, idx = z.copy(), strides(K)
    for t1, t2 in zip(idx[:-1], idx[1:]):
        a1, a2 = alpha_bar[t1], alpha_bar[t2]
        e = eps_fn(xt, a1)
        xt = np.sqrt(a2) * (xt - np.sqrt(1 - a1) * e) / np.sqrt(a1) + np.sqrt(1 - a2) * e
    return xt
z = rng.standard_normal(8000)                         # one draw, shared by every stride count
x_ref = ddim(z, 999, eps_hat)
for K in (10, 50):
    x_K = ddim(z, K, eps_hat)
    print(f'mixture, {K:3d} strides vs 999: mean |gap| {np.abs(x_K - x_ref).mean():.3f}, '
```

```
            f'KS {ks(x_K, x_ref):.3f}')
```

```
N(0,1) data,   10 strides: every sample times 0.832
N(0,1) data,   50 strides: every sample times 0.964
N(0,1) data,  999 strides: every sample times 0.998
mixture,  10 strides vs 999: mean |gap| 0.084, KS 0.078
mixture,  50 strides vs 999: mean |gap| 0.015, KS 0.018
```

単位分散のガウスデータは、10 ストライドで 0.832 倍、50 ストライドで 0.964 倍、999 ストライドで 0.998 倍に縮小する。ノイズ予測は厳密であるため、これらの倍率は決定論的であり、純粋にオイラー法の誤差のみを測っている。ストライド数を 5 倍にすると不足分は 0.17 から 0.04 へと減り、1 次手法の理論通りの挙動を示す。混合分布でも同様である。10 ストライドの終点は 999 ストライドの結果から平均 0.08 離れており（モードが ± 0.97 にあるスケール上で）、KS 統計量も同じく 0.08 となる。50 ストライドに増やすと、両方の数値とも約 0.02 まで下がる。Figure 38.25 に両方の実験結果を示す。

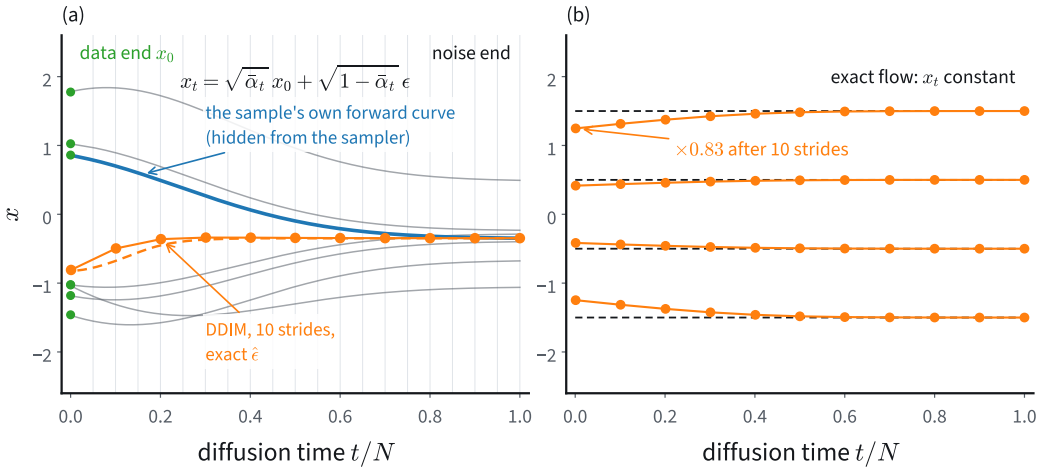


Figure 38.25: (a) 標準化した混合分布の各順方向ペア (x_0, ϵ) はそれぞれ 1 本の曲線を持ち、青いペアは谷を横切っている。同じノイズ側の終点から出発しても、厳密なノイズ予測を用いた DDIM は確率フロー ODE（10 ストライドが点、レベルごとに 1 歩ずつ進む 999 ストライドが破線）に沿って反対側のモードへ進み、そのサンプル固有の曲線を見ることは決してない。(b) 単位分散のガウスデータでは厳密なフローは恒等写像（破線）であるが、厳密なノイズ予測を用いた 10 回の DDIM ストライドはガウス分布の系が示すオイラー誤差により、すべてのサンプルを 0.83 倍に縮小する。

パネル (a) は、更新式を文字通りに読んだときの疑問に答えている。サンプルがたどるのは、そのノイズ終点を通る確率フローの軌跡であり、そのサンプルを生み出した順方向の曲線ではない。DDIM は逆過程（reverse process、38.5.5 節）の族における決定論的な極限にあたる。この族の各手法は選んだ量のノイズを再注入しつつ順過程と同じ単一時刻周辺分布を共有するため、1 つの訓練済みネットワークですべての手法に対応できる。それらの使い分けは 17.5 節で扱う。

38.6.4.2 当てはめたスコア上での DDIM

ここまでのサンプラにはすべて厳密なスコアを渡してきた。いよいよスコアを学習してサンプリングを行おう。対象はランジュバン動力学が敗れたデータ、すなわち $-m$ と $+m$ にある 2 つの

標準化モードに重みを付け直し、左が質量の 0.8、右が 0.2 を占めるようにした分布である。

モデルには単一レベルで機能した 17 個の固定特徴量をそのまま用い、今度は 50 ストライドの DDIM が訪れる 50 個の各レベルで個別に当てはめる。各レベルにおいて 4,096 個のクリーンな抽出に 4 回ずつノイズを加え、ノイズ ϵ をノイズ付きの点の特徴量に回帰する。その際、バンプが元のデータと同じように標準化データを覆うよう、特徴量は $\sqrt{4.25}\,x_t$ で評価する。あるレベルでペアが 1 つも届かないバンプには、小さな重み減衰（weight decay、2.7 節）で対処する。通常の最小二乗法では $x = 8$ での当てはめと同様にそうしたバンプの重みが不定になるが、ペナルティ項がそれをゼロに抑える。続いて前のセルの初期ノイズから DDIM を 2 回、すなわち非対称な混合分布の厳密なノイズ予測と当てはめたノイズ予測のそれぞれで走らせる。単一レベルでは学習できなかった情報を全レベルの回帰が捉えていれば、どちらの行も軽いモードの質量として 0.2 を示すはずである。

```
ws = (0.8, 0.2)                                          # the right mode is the light one
def draw(size):                                          # standardized modes +-m, weights ws
    return rng.normal(np.where(rng.random(size) < ws[0], -m, m), np.sqrt(v0))
x_pairs = np.repeat(draw(4096), 4)                       # four noisy copies per clean point
def fit_eps(a, x_pairs):                                 # ridge least squares for eps at one
↪  level
    e = rng.standard_normal(x_pairs.size)
    F = features(np.sqrt(4.25) * (np.sqrt(a) * x_pairs + np.sqrt(1 - a) * e))
    return np.linalg.solve(F.T @ F + 1e-6 * e.size * np.eye(17), F.T @ e)
coef = {a: fit_eps(a, x_pairs) for a in alpha_bar[strides(50)[:-1]]}   # the levels DDIM
↪  visits
x_exact = ddim(z, 50, lambda q, a: eps_hat(q, a, ws))
x_fit = ddim(z, 50, lambda q, a: features(np.sqrt(4.25) * q) @ coef[a])
x_true = draw(8000)
for name, xs in (('exact', x_exact), ('fitted', x_fit)):
    L, R = xs[xs < 0], xs[xs > 0]
    print(f'{name:6s} scores: light mode {(xs > 0).mean():.3f}, means {L.mean():+.2f} '
          f'{R.mean():+.2f}, widths {L.std():.3f} {R.std():.3f}, KS to p0 {ks(xs,
           ↪  x_true):.2f}')
print(f'truth: light mode 0.2, means {-m:+.2f} {m:+.2f}, widths {np.sqrt(v0):.3f}; '
      f'KS between the clouds {ks(x_exact, x_fit):.2f}')
```

```
exact  scores: light mode 0.190, means -0.96 +0.94, widths 0.229 0.234, KS to p0 0.03
fitted scores: light mode 0.199, means -0.95 +0.91, widths 0.232 0.263, KS to p0 0.05
truth: light mode 0.2, means -0.97 +0.97, widths 0.243; KS between the clouds 0.03
```

結果はまさにその通りになる。単一レベルの厳密なスコアを用いたランジュバン動力学がほぼ半分を残してしまった軽いモードに、どちらの行も 0.01〜0.02 程度の誤差でサンプルの 5 分の 1 を配分している。当てはめを用いたサンプラは、密度も、スコアも、0.2 という数値も一度も目にしていない。目にしたのは、ノイズ付きの点とそれを作ったノイズだけである。

2 つの行は同じ初期ノイズから出発しているため、共通するずれはサンプラの誤差であり、両者を隔てる差は当てはめの誤差である。両者に共通しているのは、軽いモードの平均が 0.97 より小数第 2 位で数単位小さく、重いモードの幅が約 0.23 と真の値 0.243 よりわずかに狭い点である。厳密なノイズ予測では誤差はオイラー法のものに限られるため、これが 50 ストライドの離散化誤差である。50 ストライドでガウスデータが 0.964 倍になったガウス分布の系から予想される通り、数パーセントの縮小が生じている。一方、当てはめは独自の目に見える誤差を 1 つ付け加えており、軽いモードの幅が 0.25〜0.28（厳密な行では約 0.24）に広がっている。2 つのモードの違いはデータに占める割合だけであるから、この余分な広がりはそこに起因する。最小二乗法は 17 個の重みをペアが集まる場所へ優先的に振り向けるが、ペアの 5 分の 1 しか軽いモードには落

ちない。KS 統計量もこの見方を裏づけており、どちらの行もデータの新規抽出から小数第 2 位で数単位の距離にあり、2 つの行どうしの距離もそれ以下である。

実際の拡散モデルは各レベルを別々に当てはめたりはしない。ネットワークがレベルを入力として受け取り、全レベルで重みを共有する（演習問題 8）。それでも回帰、サンプラ、そして 2 種類の誤差の構造は変わらない。

38.6.4.3 ベイズの定理によるガイダンス

p に対するサンプラは、p に含まれるあらゆる種類のサンプルを返す。しかし実際には、特定のクラスの画像やキャプションに合致する画像など、1 種類だけを求めたいことが多い。**ガイダンス**（guidance）は、サンプラが追跡するスコアを書き換えることで生成をクラス y へと誘導する。17.4 節では画像向けにガイダンス付きサンプラを構築して評価するが、ここでは混合分布で手計算により確かめられる 2 つの事実に絞る。すなわち、ガイダンスとはスコアに対するベイズの定理であり、強度を 1 より大きくすると、いかなるノイズ付加データも持ち得ない密度をサンプラが追跡するようになるという事実である。

等しい重みを持つ標準化混合分布において、y を「右のモード」とし、クラス $\mathcal{N}(m, v_0)$（ただし $m = 0.970$、$v_0 = 0.243^2$）のみからサンプルを得たいとしよう。ノイズレベル t におけるベイズの定理 $p_t(\mathbf{x} \mid y) \propto p_t(\mathbf{x})\, p_t(y \mid \mathbf{x})$ は、エビデンス $p(y)$ が $\mathbf{x}$ に依存しないため、スコアに対しては加法的な関係になる。

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid y) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(y \mid \mathbf{x}). \tag{38.89}$$

条件付きスコアは、無条件スコアに分類器の対数確率（log-probability）の勾配を足したものであり、本節のどのサンプラもこれをそのまま入れて動かせる。**分類器ガイダンス**（classifier guidance、(Dhariwal and Nichol 2021)）は、ノイズ付き入力に対する分類器 $p_\phi(y \mid \mathbf{x}, t)$ から第 2 項をとり、それを**ガイダンス強度**（guidance scale）γ でスケール倍する。

$$\begin{aligned} \tilde{\mathbf{s}}(\mathbf{x}, t) &= \nabla \log p_t(\mathbf{x}) + \gamma \nabla \log p_\phi(y \mid \mathbf{x}, t), \\ \tilde{\mathbf{s}} &= \nabla \log \tilde{p}_t, \qquad \tilde{p}_t(\mathbf{x}) \propto p_t(\mathbf{x})\, p_\phi(y \mid \mathbf{x}, t)^\gamma. \end{aligned}$$

$\gamma = 1$ のとき、これはベイズの定理そのものである。$\gamma > 1$ のとき、誘導されたベクトル場は分類器の確率をべき乗した**傾斜**密度のスコアになる。**分類器なしガイダンス**（classifier-free guidance、CFG、(Ho and Salimans 2022)）は、Equation 38.89 を右から左へ $\nabla \log p_t(y \mid \mathbf{x}) = \nabla \log p_t(\mathbf{x} \mid y) - \nabla \log p_t(\mathbf{x})$ と読むことで分類器を不要にし、条件付きスコアモデル $\mathbf{s}_\theta(\mathbf{x}, t, y)$ と無条件スコアモデル $\mathbf{s}_\theta(\mathbf{x}, t, \emptyset)$ だけを用いる（これらは 1 つのネットワークでまかなえる。17.4.4.2 節）。

$$\tilde{\mathbf{s}}(\mathbf{x}, t) = \mathbf{s}_\theta(\mathbf{x}, t, \emptyset) + \gamma \left[\mathbf{s}_\theta(\mathbf{x}, t, y) - \mathbf{s}_\theta(\mathbf{x}, t, \emptyset) \right], \tag{38.90}$$

これは $(1-\gamma)\, \mathbf{s}_\emptyset + \gamma\, \mathbf{s}_y$ と同値であり、$\gamma > 1$ では条件付きスコアのさらに先へと外挿する。

ある 1 つの事実が $\gamma > 1$ の解釈を制約する。傾斜密度 $\tilde{p}_t$ は一般に、**いかなる**クリーン分布のノイズ付加周辺分布にもなっていないのである。周辺分布 p_t は $\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0$ にスケール σ_t のノイズを加えた法則であるため、曲率上界 Equation 38.81 により、データが何であれ $-\nabla^2 \log p_t$ の固有値が $1/\sigma_t^2$ を超えることはない。ところが分類器を γ 乗すると、特にクラス間の中間付近で曲率が上乗せされる。この混合分布では、$\gamma > 1 + v_0 v_t / (\sigma_t^2 m^2)$ となった時点で中間付近において上界が破綻する。ここで $v_t = \bar{\alpha}_t v_0 + \sigma_t^2$ はノイズ付加されたクラスの分散である（演習問題 6）。この閾値は最もノイズの大きいレベルで 1.0625 であり、1.09（$t = 100$ のとき）にとど

まる。$\gamma = 3$ では 1,000 レベルのうち最もノイズの小さい約 10 レベルでしか上界が成り立たず、$\gamma = 10$ ではさらに少なくなる。

なぜこれが重要なのか。もし傾斜密度の列が単一のクリーン分布のノイズ付加周辺分布であったなら、そのスコア上で逆方向連鎖を走らせれば、p のときと同様にそのクリーン分布へと行き着くはずである。しかし実際にはそうではないため、ガイダンス付きの連鎖がどこに行き着くかは事前には保証されない。実際に確かめてみよう。38.6.3.2 節の祖先連鎖を厳密なスコア、すなわち混合分布の無条件スコアと単一ガウス分布 $\mathcal{N}(\sqrt{\bar{\alpha}_t}\, m,\ v_t)$ のクラス条件付きスコアを Equation 38.90 により $\gamma \in \{1, 3, 10\}$ で組み合わせたスコアの上で走らせる。

```
    def cfg_sample(gamma, n=8000):                       # ancestral chain with the guided score
        xt = rng.standard_normal(n)
        for t in range(N - 1, -1, -1):
            a, b = alpha_bar[t], beta[t]
            v = a * v0 + 1 - a                             # noised component variance
            s_uncond = mixture_score(xt, v, (-np.sqrt(a) * m, np.sqrt(a) * m))
            s_cond = (np.sqrt(a) * m - xt) / v              # class y = the right mode
            xt = (xt + b * ((1 - gamma) * s_uncond + gamma * s_cond)) / np.sqrt(1 - b)
            if t > 0:
                xt = xt + np.sqrt(b) * rng.standard_normal(n)
        return xt
    for gamma in (1.0, 3.0, 10.0):
        xs = cfg_sample(gamma)
        print(f'gamma = {gamma:4.1f}: mass right of 0 {(xs > 0).mean():.3f}, '
              f'mean {xs.mean():.3f}, std {xs.std():.3f}')
    print(f'exact conditional: mean {m:.3f}, std {np.sqrt(v0):.3f}')
```

```
gamma =  1.0: mass right of 0 1.000, mean 0.973, std 0.243
gamma =  3.0: mass right of 0 1.000, mean 1.027, std 0.234
gamma = 10.0: mass right of 0 1.000, mean 1.070, std 0.236
exact conditional: mean 0.970, std 0.243
```

$\gamma = 1$ では、連鎖は目的のクラスを忠実に再現する。無条件の混合分布では半分しか置かれないゼロより右側の領域にすべての質量が集まり、平均と標準偏差は約 0.01 以内の差で厳密値 0.970 および 0.243 に一致する。$\gamma = 3$ および 10 では、もはやモード間で移すべき質量が残っていないため、ガイダンスはモード自体をクラス境界から遠ざかる方向へ押し動かし、平均は約 1.03、さらには 1.07〜1.08 へと移動する。一方、幅はほとんど変化せず、約 5% の範囲で 0.24 付近にとどまる。

ここに曲率上界の破綻が現れている。クリーンなレベルでは 2 つのクラスは標準偏差の 8 倍離れていて分類器は確信を持っているため、そこでの傾斜密度は平均 0.970（任意の γ に対して）のクラスそのものである。単一の順過程の周辺分布をたどる連鎖であれば、そこへ着地するはずである。しかし $\gamma = 3$ と 10 で走らせた連鎖は、そこから約 0.06 および 0.1 だけ右にずれた場所へ着地する。連鎖が追跡するスコアの背後にある密度たちが、単一の過程に属していないからである。このずれは、ノイズ付加されたクラスどうしが重なり合う中間のレベルで生み出される（演習問題 5）。このように $\gamma > 1$ のガイダンスは、サンプルを他のクラスから遠ざけ、本来のクラスの位置を超えた先へと押しやる。単一のガウスモードしか持たないクラスには失うべき多様性がほとんどないが、画像ではその多様性の喪失がはっきりと測定される（17.4.4.2 節）。

ガイダンスの有無にかかわらず、本節のサンプラがいずれもスコアを何度も評価するのは、その経路が曲がっているからである。ガウス分布の系が示した通り、最も単純なデータでさえ DDIM の時計のもとでは経路が曲がってしまう。ここまで順過程をガウスカーネルのために選んできたが、その経路が追跡しやすいかどうかは一度も問わなかった。ノイズからデータへと向かう経路

のうち、まっすぐなものはどれだろうか。

38.6.5 まとめ

スコアには正規化定数が不要であり、ノイズ付加データに対してはすべての訓練ペアで既知の目標 $-\epsilon/\sigma$ の条件付き平均に等しい。したがってノイズ付きペアに対する最小二乗法によってサンプルのみからスコアを学習でき、その損失はゼロではなく事後分散の下限に落ち着く。ただし単一のノイズレベルでは不十分である。データから離れた場所では当てはめが恣意的になり、ランジュバン動力学は初期分布の比率を保ったまま、本来の質量が 5 分の 1 しかないモードにほぼ半分のサンプルを残してしまう。順過程のあらゆるレベルにわたるスコアがあれば、最もよく知られた 2 つのサンプラは既存の方程式の離散化として理解できる。DDPM は離散時刻で観測された分散保持 SDE であり、重み σ_t^2 を付けたノイズ除去スコアマッチングを損失とする。DDIM は座標 $(\bar{\sigma}, \bar{\mathbf{x}})$ における確率フロー ODE に対するオイラー法であり、点質量に対しては厳密である一方、ガウスデータに対しては縮小しすぎる。全レベルで最小二乗法により当てはめたスコアの上で DDIM を走らせると、単一レベルでは復元できなかった重み 0.8 と 0.2 が正しく得られた。ガイダンスはスコアに対するベイズの定理であり、$\gamma > 1$ ではいかなるノイズ付加密度よりも鋭く曲がる密度を追跡する。

38.6.6 演習問題

1. [code] **ランジュバン離散化による統計的バイアス。** ガウス目標分布 $p = \mathcal{N}(0, v)$ に対し、Equation 38.82 のステップサイズ h を用いたランジュバンサンプラは線形漸化式 $x \leftarrow (1 - \frac{h}{2v})\, x + \sqrt{h}\,\xi$ となる。
   1. この漸化式が定常分布を持つステップサイズの範囲を求め、その分散を計算し、統計的バイアスを h の 1 次まで展開せよ。
   2. `langevin` を定義しているセルの混合分布の各モードを $v = 0.25$ の独立したガウス分布とみなし、定常状態の $\mathbb{E}[X^2]$（$h \in \{0.01, 0.1, 0.25, 0.5\}$ における値）を予測せよ。予測した各バイアスを、10,000 本の連鎖からの推定量のモンテカルロ標準誤差と比較せよ。続いて `langevin` をこれら各ステップサイズで 2,000 ステップ、初期点 `rng.normal(0.0, 3.0, 10000)`（`warm` と同じ初期点）から実行せよ。どのステップサイズから実験でバイアスを識別できるか。
   3. メトロポリス調整付きランジュバンアルゴリズム (Roberts and Tweedie 1996) は、密度比 $p(\mathbf{x}')/p(\mathbf{x})$ を必要とする採択・棄却判定によってこのバイアスを取り除く。訓練済みのスコアネットワーク $\mathbf{s}_\theta$ しか持たないサンプラがこの判定を直接適用できないのはなぜか。

2. [code] **アニーリングによる単一モードからの脱出。** `langevin` を定義しているセルでは、$0.8\,\mathcal{N}(-2, 0.5^2) + 0.2\,\mathcal{N}(2, 0.5^2)$ の一方のモードから出発した連鎖はそこにとどまり、広い初期分布から出発するとほぼ半分の質量が右側に残った。アニーリング付きランジュバン動力学は、まず平滑化された密度 p_σ（ここでは成分分散 $0.25 + \sigma^2$ の混合分布）の上で、減少するノイズレベルの系列 $\sigma = 2, 1, 0.5, 0.25, 0.1, 0$ に沿ってサンプラを走らせる。
   1. 本文で求めた通り、等しい重みのもとでは p_σ は $\sigma \geq 1.94$ で谷を持たない。重みが 0.8 と 0.2 の場合、谷はさらに早く閉じる。細かいグリッド上で `mixture_score(ws=(0.8, 0.2))` の符号変化の回数を数え、谷が消える最小の σ を求めよ。上の系列のうち、谷を持たないレベルはどれか。
   2. 系列の各レベルにおいて 300 ステップずつ `langevin` を、成分分散 $0.25 + \sigma^2$ を `var` に渡して実行せよ。これを 10,000 本の連鎖を -2 から出発させた場合と、同じく 10,000 本の連鎖を $+2$ から出発させた場合のそれぞれについて行え。またステップサイズについては、固定ステップ $h = 0.01$ の場合と、$h = 0.04\,(0.25 + \sigma^2)$（値は 0.01 に

$\sigma = 0$ で一致する）の場合の 2 通りで試せ。4 回の実行それぞれの終了時点における $X > 0$ の連鎖の割合を報告せよ。出発点を忘れるのはどちらのステップ規則か。その割合は目標値 0.2 に近づくが完全には一致しない。各レベルにおけるゼロより右側の p_σ の質量を計算し、どのレベルで比率が固定されたかを述べよ。

3. Song and Ermon (2019) は各レベルのステップサイズを σ^2 に比例してスケーリングしている。演習問題 1 の線形漸化式の縮小率を用いて、大きなノイズレベルで固定ステップが失敗する理由を説明せよ。

3. [code] **ベイズリスクの下限。**●`w_ls` を当てはめるセルでは、同一の 16,384 組のノイズ付きペアにわたる平均として当てはめと厳密なスコアそれぞれの平均二乗誤差を出力し、さらに数値積分による母集団の下限を出力している。

   1. 本文では回帰のベイズリスクである下限 Equation 38.80 を、補題で $\mathbf{v} = \mathbf{0}$ とおくことで導いた。Tweedie の公式（Equation 38.79）を用いてこの下限が $\mathbb{E}\ \mathrm{tr}\,\mathrm{Cov}[\mathbf{x} \mid \tilde{\mathbf{x}}]/\sigma^4$ にも等しいことを示し、$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, v\mathbf{I})$ に対して 2 つの表式が一致することを確かめよ。$\sigma \to 0$ のとき、滑らかな密度を持つデータに対して下限はどう振る舞い、互いに異なる n 個の点からなる訓練セットに対してはどう振る舞うか。その極限において、訓練セットに対する最適なノイズ除去器は何を返すか。
   2. 出力された厳密なスコアの平均二乗誤差は、母集団の下限そのものではない。その標準誤差を 2 通りの方法で計算せよ。すなわち、16,384 個のペアごとの二乗誤差を互いに独立とみなした場合と、各クリーンな点に対する 4 個のノイズ付きコピーにわたる平均 4,096 個から計算した場合である。2 つが異なるのはなぜか。また、本文で出力した 20 回の新規抽出にわたる標準偏差 `errs.std()` と一致するのはどちらか。出力された値は母集団の下限と整合しているか。さらに、同じペア上で評価した当てはめと厳密なスコアとの差が、この標準誤差よりはるかに小さくなる理由を説明せよ。

4. **ノイズレベルにわたる重み付け。**●線形スケジュール $\beta(t) = \beta_{\min} + t\,(\beta_{\max} - \beta_{\min})$（区間 $[0, 1]$ 上、$\beta_{\min} = 0.1$、$\beta_{\max} = 20$）を持つ VP-SDE を考える。このとき $\bar{\alpha}(t) = \exp\big(-\beta_{\min} t - \tfrac{1}{2}(\beta_{\max} - \beta_{\min})t^2\big)$ および $\sigma_t^2 = 1 - \bar{\alpha}(t)$ となる。$\mathbb{R}^d$ 上のデータが滑らかな密度を持ち、$\mathbb{E}\|\nabla \log p_t\|^2$ は $t \to 0$ で有界にとどまると仮定する。$R(t)$ をノイズ条件付き DSM 損失 Equation 38.83 のレベル t におけるベイズリスク、すなわち下限 Equation 38.80 において σ_t を σ の代わりに置いたものとする。

   1. $R(t) \approx d/(\beta_{\min} t)$（$t \to 0$ のとき）を示せ。
   2. $w(t)\,R(t)$ の $t \to 0$ における漸近挙動を、$w = 1$、DDPM の重み付け $w = \sigma_t^2$、および尤度重み付け $w = g^2 = \beta(t)$ のそれぞれについて求めよ。$\int_0^1 w(t)\,R(t)\,dt$ が有限になるのはどの重み付けか。また、各重み付けはどのノイズレベルを重視するか。
   3. 訓練では t を $[\varepsilon, 1]$ から一様に抽出する。尤度重み付けにおいて、この抽出のもとでの $w(t)\,R(t)$ の二次モーメントが $1/\varepsilon$ のオーダーで増大することを示せ。Y. Song, Durkan, et al. (2021)（5.1 節）は代わりに、t を $\beta(t)/\sigma_t^2$ に比例する $[\varepsilon, 1]$ 上の密度から抽出し、抽出されたレベルの DSM 項に $c\,\sigma_t^2$（ただし定数 $c = \frac{1}{1-\varepsilon}\int_\varepsilon^1 \beta(r)/\sigma_r^2\,dr$）を掛けている。この推定量が尤度重み付け損失に対して不偏（統計的バイアスが 0）であること、および最適点におけるその値 $c\,\sigma_t^2\,R(t)$ が $t \to 0$ で有界にとどまることを示せ。

5. [code] **ノイズレベル別のガイダンス。**●`cfg_sample` を定義しているセルでは、γ を 1 から 10 へ引き上げると、誘導されたサンプルの平均が約 0.97 から 1.07〜1.08 へと移動した。

   1. 本文では、クリーンなレベルにおける傾斜密度はクラスそのものであると述べた。これを確かめよ。連鎖の最後のレベル、すなわち `cfg_sample` における $t = 0$ では、右モードのクラス事後確率は x のロジスティック関数になる。その式を書き下し、そのレベルにおいて $p_t(x)\,p_t(y \mid x)^\gamma$ に比例する傾斜密度の平均を $\gamma \in \{1, 3, 10\}$ に対し

て数値積分で計算せよ。各平均は m からどれだけ離れており、出力されたサンプル平均はどれだけ離れているか。

2. `cfg_sample` を変更し、$\gamma = 10$ をステップ $t \geq 500$ のみ、$200 \leq t < 500$ のみ、または $t < 200$ のみで用い、それ以外のステップでは $\gamma = 1$ を用いるようにせよ。シフトを生み出しているのはどのノイズレベル帯か。その答えを、ノイズ付加された 2 つのクラスの分離度 $\sqrt{\bar{\alpha}_t}\, m / \sqrt{v_t}$ と関連づけよ。ここで m と v_0 は変数 `m` と `v0` であり、$v_t = \bar{\alpha}_t\, v_0 + 1 - \bar{\alpha}_t$ はノイズ付加された成分分散（変数 `v`、`cfg_sample` 内の定義）である。
3. Kynkäänniemi et al. (2024) は画像モデルにおいて、ガイダンスは高ノイズレベルでは有害であり、低ノイズレベルではほぼ不要であり、その中間で有益であることを見いだしている。この 1 次元の例は、このパターンのどの部分を再現でき、どの部分を示すことができないか。

6. **誘導された密度とノイズ付加周辺分布。**●Equation 38.90 の誘導されたベクトル場（$\gamma > 1$）は、傾斜密度 $\tilde{p}_t(\mathbf{x}) \propto p_t(\mathbf{x})\, p_t(y \mid \mathbf{x})^{\gamma}$ のスコアである。ある密度がレベル t におけるノイズ付加周辺分布であるとは、$\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sigma_t \epsilon$（ただし $\sigma_t^2 = 1 - \bar{\alpha}_t$）の法則として、何らかの $\mathbf{x}_0$ のクリーン分布から得られることをいう。

   1. Equation 38.81 をクリーンな変数 $\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0$ に適用し、ノイズ付加周辺分布 p_t が次を満たすことを示せ。

$$\nabla^2 \log p_t(\mathbf{x}) = -\frac{\mathbf{I}}{\sigma_t^2} + \frac{\bar{\alpha}_t}{\sigma_t^4} \mathrm{Cov}\left[\mathbf{x}_0 \mid \mathbf{x}_t = \mathbf{x}\right].$$

      クリーン分布が何であれ、$-\nabla^2 \log p_t$ のいかなる固有値も $1/\sigma_t^2$ を超えないと結論せよ。

   2. `cfg_sample` において、ノイズ付加された 2 つのクラスは $\mathcal{N}(\pm\sqrt{\bar{\alpha}_t}\, m, v_t)$ である。ここで m と v_0 は変数 `m` と `v0` であり、$v_t = \bar{\alpha}_t\, v_0 + \sigma_t^2$ はノイズ付加された成分分散である。ラベル y は右のクラスを指す。$-(\log \tilde{p}_t)''(x) = 1/v_t + (\gamma - 1)\, a_t^2\, \varsigma(a_t x)\,(1 - \varsigma(a_t x))$ が成り立つことを示せ。ただし $a_t = 2\sqrt{\bar{\alpha}_t}\, m / v_t$ であり、$\varsigma(u) = 1/(1+e^{-u})$ はロジスティック関数である。

   3. 最小の γ（$\tilde{p}_t$ が (a) の上界にある x で違反するような値）を求めよ。それをレベル $t = 0$、100、500（`alpha_bar` の添字）で評価し、さらに $t \to N$ としたときの極限を求めよ。これらの値から、`cfg_sample` の連鎖が $\gamma = 3$ において追跡している密度について何が言えるか。

7. **スコア誤差と尤度。**●●尤度重み付け $w = g^2$ のもとでは、母集団スコア損失がデータからモデルへのカルバック・ライブラー（KL）ダイバージェンスを上から抑える（Y. Song, Durkan, et al. (2021) の定理 1）。同論文では経路測度を用いてこの上界を証明しているが、本問では 38.5.2 節のフォッカー・プランク方程式から導出する。これは SDE（命題 C.1）および確率フロー ODE（定理 3.1）に対してこの経路をとった Lu, Zheng, et al. (2022) に倣ったものである。順方向 SDE $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{W}$ が周辺分布 p_t（区間 $[0, T]$ 上）を持つとする。逆時間 $\bar{t} = T - t$ において、厳密な時間反転とモデルはそれぞれ次で与えられる。

$$\begin{aligned} d\mathbf{x} &= \left[-\mathbf{f} + g^2\, \nabla \log p_t\right] d\bar{t} + g\, d\mathbf{B}_{\bar{t}}, \\ d\mathbf{x} &= \left[-\mathbf{f} + g^2\, \mathbf{s}_\theta(\mathbf{x}, t)\right] d\bar{t} + g\, d\mathbf{B}_{\bar{t}}, \end{aligned} \tag{38.91}$$

   ただし右辺は $t = T - \bar{t}$ で評価する。第 1 式は周辺分布 p_t を持ち、事前分布 π（$\bar{t} = 0$ の

とき）から出発した第 2 式は周辺分布 q_t を持つ。すべての密度は滑らかかつ正であり、必要なすべての部分積分が成り立つほど速く減衰すると仮定する。

1. $\bar{t}$ に関する両方のフォッカー・プランク方程式を $\partial_{\bar{t}}\rho = -\nabla\cdot(\rho\,\mathbf{u})$ の形で書き、速度場 $\mathbf{u}_p$ と $\mathbf{u}_q$ を求めよ。
2. $\frac{d}{dt}D_{\mathrm{KL}}(p_t\,\|\,q_t) = \mathbb{E}_{p_t}\left[(\mathbf{u}_p-\mathbf{u}_q)^{\top}\nabla\log(p_t/q_t)\right]$ を示せ。
3. 右辺を $\frac{1}{2}g^2\,\mathbb{E}_{p_t}\|\mathbf{s}_\theta-\nabla\log p_t\|^2$ で上から評価し、$\bar{t}$ について積分することで次を得よ。

$$D_{\mathrm{KL}}(p_0\,\|\,q_0) \le D_{\mathrm{KL}}(p_T\,\|\,\pi) + \frac{1}{2}\int_0^T g(t)^2\,\mathbb{E}_{p_t}\left\|\mathbf{s}_\theta(\mathbf{x},t)-\nabla\log p_t(\mathbf{x})\right\|^2 dt. \tag{38.92}$$

   q_t がどのような条件を満たすとき、この導関数の上界は等号を達成するか。
4. スコアの代わりに $\mathbf{s}_\theta$ を用いた確率フロー ODE に対して (a) と (b) を繰り返せ。(c) を可能にしたどの項が欠けており、なぜスコア損失だけではもはや $D_{\mathrm{KL}}(p_0\,\|\,q_0)$ を上から抑えられないのかを説明せよ。

8. [extended] **学習されたノイズ予測。**●●`fit_eps` を定義しているセルでは、17 個の固定特徴量を用い、50 個の各レベルで個別にノイズ予測を学習した。本来の拡散モデルは全レベルに対して 1 つのネットワークを持つ。同じデータ（非対称な標準化混合分布からの抽出 `draw(size)`）に対し、DDPM スケジュール `beta` と `alpha_bar` のもとでそれを構築せよ。
   1. 使用しているタブのフレームワークにおいて、ノイズ予測ネットワーク $\epsilon_\theta(x_t,t)$（入力 $(x_t, t/1000)$、64 個の tanh ユニットからなる 2 つの隠れ層）を、学習率 3×10^{-3} の Adam を用いて毎回 256 個の新規抽出からなるバッチ上で 4,000 ステップ訓練せよ。損失には t を 1,000 個のレベルから一様に抽出する単純損失を用いること。厳密なノイズ予測 `eps_hat(xt, a, ws)` に対する二乗平均平方根誤差をいくつかのレベルにおける p_t のサンプル（複数の t）上で計算し、本文で当てはめた特徴量モデルについても当てはめた各レベルで同様に計算せよ。それぞれの誤差はどこで最大になるか。
   2. `ddim` を用いてノイズ `z` から $K=10$ および $K=50$ でサンプリングし、さらに 38.6.3.2 節の祖先ステップ（誘導スコアをネットワークのスコアに置き換えた `cfg_sample` のループ）でもサンプリングせよ。得られた各サンプル集合を `x_exact`、`x_fit`、`x_true` と、軽いモードの質量・2 つのモードの幅・`ks` によって比較せよ。(a) の誤差に最も敏感なサンプラはどれであり、そのサンプラのどのステップで誤差が入り込むか。
   3. $w=1$ および尤度重み付け $w=\beta$ に対応するレベル別の重み、すなわちレベル t におけるノイズ損失に $w(t)/(1-\bar{\alpha}_t)$ を掛けた損失で再訓練し、(a) と (b) を繰り返せ。演習問題 4 は各重み付けがどのレベルを重視するかを予測しているが、実際の誤差はその予測と一致するか。

   提出物は、各重み付けと各サンプラに対する (b) の統計量をまとめた表、および (a) の誤差を t に対してプロットした図である。

38.7 フローマッチングと最適輸送

前節のサンプラは 1 ステップにつき 1 回スコアを呼び出し、逆方向の連鎖では 1,000 回、その決定論的なストライドでも数十回呼び出す。これほど多くのステップを要するのは、ノイズからデータへ至る経路が曲がっており、ソルバは曲がった経路を短い区間に刻まなければ追跡できない

からである。そうした経路は誰かが設計したものではなく、ノイズ付加過程に付随して生じたものにすぎない。ノイズからデータへの経路を自ら選び、ソルバが大きなステップを踏めるような経路にすることはできないだろうか。

-2 と $+2$ にある 2 つのデータ点と、最終的にそこへ到達しなければならないガウスノイズの雲を思い浮かべよう。もしすべての粒子がそれぞれの目的地へ向かって一定速度で直線上を移動するなら、長さ 1 のオイラーステップ 1 回で正確にそこへ運べるはずである。しかし、速度場は位置と時刻に対する 1 つの関数にすぎない。異なる点へ向かう粒子が同じ場所を通過しても、そこでの速度場は 1 つの向きしか指せないのである。

フローマッチング (Lipman et al. 2023) と Rectified Flow (Liu et al. 2023) は実際に経路を選ぶ。すなわち、経路をあらかじめ指定し、それに沿って質量を運ぶ速度だけを学習する。どの経路が最も低コストかという問いははるかに古く、**最適輸送**（optimal transport）の問題として知られる。1781 年に Gaspard Monge が土の山を穴へ運ぶ問題として提起したものであり、二乗距離のもとで最も低コストな運動は、まさに先ほど望んだとおり、すべての粒子が一定速度で直線上を進む運動となることがわかっている (McCann 1997; Benamou and Brenier 2000)。

単純な要素を組み合わせて作った経路であれば、どのような経路でも 1 回の回帰によってその速度を訓練できる。速度は粒子にわたる平均であり、最小二乗法（least squares）は平均を学習するからである（38.7.1 節）。それでも、直線の要素を組み合わせただけではフローは曲がってしまう。ノイズとデータのペアを組み直せばフローは真っ直ぐになり（38.7.2 節）、その再ペアリングが何を目指しているかは最適輸送が明らかにする（38.7.3 節）。なお、2 つの箇所は再読時に回してもよい。1 つは 38.7.1.3 節でガウス経路（Gaussian path）の定義に続いて述べるスコア・ノイズ・速度の相互変換であり、もう 1 つは 38.7.4 節におけるソルバの比較である。

> **!** 1 つの時計
>
> 時間は前節と同じく、$t = 0$ のデータから、今度は $t = 1$ に置かれたノイズへと進む。これはスコアでも速度でも同様である。訓練では時間を順方向に進めてデータにノイズを加え、どのサンプラも $t = 1$ から $t = 0$ へと逆向きに積分する。
> 本節で引用するフローマッチングの諸論文およびそのコードでは、時間を逆向きに、すなわち 0 のノイズから 1 のデータへと進めている（Figure 38.26）。本書の式をそれらの文献と比較するには、$1 - t$ を論文の t に代入し、論文のノイズ $\mathbf{x}_0$ を本書の ϵ、論文のデータ $\mathbf{x}_1$ を本書の $\mathbf{x}_0$ と読み替え、すべての速度の符号を反転させればよい。

データは $\mathbf{x}_0 \sim p_0$ とし、p_t は p_0 からノイズ $p_1 = \mathcal{N}(\mathbf{0}, \mathbf{I})$ へ至る密度の経路とする。ノイズ側からのサンプルには 2 つの記号を使い分ける。前節のようにガウスノイズを表すときは ϵ と書き、任意の 2 つの端点分布に対して成り立つペアリングに関する命題では $\mathbf{x}_1$ と書く。

38.7.1 フローマッチング

38.7.1.1 確率経路と速度場

拡散モデル（diffusion model）はノイズ付加過程から経路を得ていたため、その過程を逆向きに走らせるにはスコアを学習しなければならなかった。ここでは経路自体を自由に選べる。**確率経路**（probability path）とは、密度の族 $(p_t)_{t\in[0,1]}$ であって、p_0 がデータ、p_1 がノイズとなるものである。速度場（velocity field）$\mathbf{u}_t$ がこの経路を**実現する**とは、両者の組が 38.5.4 節の連続の式（continuity equation）を満たすことをいう。連続の式は、速度場によって運ばれる密度が質量の正味の流入分だけ変化することを表す次の式である。

$$\partial_t p_t + \nabla \cdot (p_t \, \mathbf{u}_t) = 0.$$

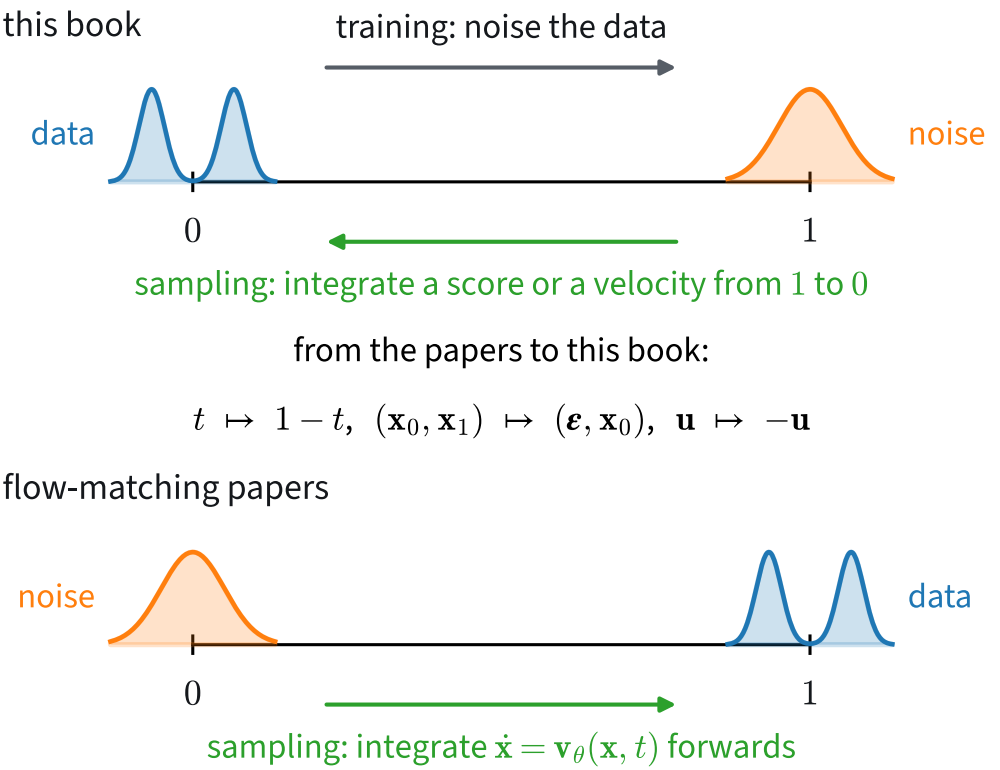


Figure 38.26: 1 つの時計と論文の時計。本書（上）はデータを $t = 0$、ノイズを $t = 1$ に置き、時間に関して順方向にデータへノイズを加え、スコアまたは速度を 1 から 0 へ積分してサンプリングする。フローマッチングの諸論文（下）はノイズを 0、データを 1 に置き、順方向にサンプリングする。両端の密度は同じであり、変数変換 $t \mapsto 1 - t$ によって一方の時計は他方の時計へと変換され、すべての速度の符号が反転する。

このとき、速度場がリプシッツ連続であれば、$\dot{\mathbf{x}} = \mathbf{u}_t(\mathbf{x})$ を p_1 のサンプルから $t=1$ より $t=0$ まで積分することで、各時刻で p_t のサンプルが、そして $t=0$ でデータのサンプルが得られる。向きに注意しよう。速度場はデータからノイズへ向かうノイズ付加の方向を指しており、サンプラはそれに逆らって進む。ネットワーク $\mathbf{u}_\theta(\mathbf{x}, t)$ が $\mathbf{u}_t$ を近似するなら、生成はノイズから常微分方程式（ODE）を解くことになる。このようなモデルは時間を逆向きに走らせた連続正規化フロー（continuous normalizing flow、38.3.3 節）であり、$t=1$ から $t=0$ へのフロー写像（flow map）が生成器（generator）となる。フローマッチングは、この ODE を 1 度も解くことなくモデルを訓練する。自然な目的関数は、次の**フローマッチング損失**（flow-matching loss）である。

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1],\, \mathbf{x} \sim p_t}\left[\left\| \mathbf{u}_\theta(\mathbf{x}, t) - \mathbf{u}_t(\mathbf{x}) \right\|^2 \right]. \tag{38.93}$$

これは真の速度に対するネットワークの平均二乗誤差を、時間と雲全体にわたって平均したものである。しかし、このままでは役に立たない。$\mathbf{u}_t$ はデータ分布（data distribution）全体に依存するため、評価できない場を回帰のターゲットにしているからである。

そこで、具体的に書き下せる要素から経路を組み立てる。条件付け変数 $\mathbf{z}$ を分布 q から引くものとし、それをデータ点、あるいはノイズ点とデータ点のペアとする。$\mathbf{z}$ の各値に対し、移動する小さな分布の塊（ブロブ）$p_t(\mathbf{x} \mid \mathbf{z})$ として、$t=0$ でデータ点の上にあり、$t=1$ でノイズに行き着くものを選ぶ（$\mathbf{z}$ がデータ点単独ならノイズ全体に広がり、$\mathbf{z}$ がペアなら対応するノイズ点の上に集まる）。各ブロブには閉じた形でわかる速度 $\mathbf{u}_t(\mathbf{x} \mid \mathbf{z})$ が備わっている。経路全体はこれらブロブの混合分布 $p_t(\mathbf{x}) = \int p_t(\mathbf{x} \mid \mathbf{z})\, q(\mathbf{z})\, d\mathbf{z}$ であり、その粒子を $\mathbf{x}_t$ と書く。すなわち、$\mathbf{z}$ を q から引き、続いて時刻 t におけるそのブロブから点を引くのである。では、この混合分布を動かす速度は何だろうか。本節のすべてはその答えの上に成り立っている。

命題（周辺速度は事後平均である）。 各 $\mathbf{z}$ に対して密度 $p_t(\cdot \mid \mathbf{z})$ が正かつ $(\mathbf{x}, t)$ について連続微分可能であり、連続微分可能な速度のもとで連続の式 $\partial_t p_t(\mathbf{x} \mid \mathbf{z}) = -\nabla \cdot \left(p_t(\mathbf{x} \mid \mathbf{z})\, \mathbf{u}_t(\mathbf{x} \mid \mathbf{z}) \right)$ を満たし、さらに以下の積分で積分記号下の微分が可能であるとする。$\{p_t > 0\}$ 上で次式により周辺速度（marginal velocity）を定義する。

$$\mathbf{u}_t(\mathbf{x}) = \mathbb{E}\left[\mathbf{u}_t(\mathbf{x} \mid \mathbf{z}) \mid \mathbf{x}_t = \mathbf{x} \right] = \int \mathbf{u}_t(\mathbf{x} \mid \mathbf{z})\, \frac{p_t(\mathbf{x} \mid \mathbf{z})\, q(\mathbf{z})}{p_t(\mathbf{x})}\, d\mathbf{z}. \tag{38.94}$$

このとき $(p_t, \mathbf{u}_t)$ は連続の式を満たす。

位置 $\mathbf{x}$ における雲の速度は、$\mathbf{x}$ に粒子を送り込み得た各ブロブの速度を、実際にそうした尤もらしさを重みとして平均したものである。

証明。 混合分布を積分記号下で微分し、条件付きの連続の式を代入して、発散を積分の外へ出す。

$$\begin{aligned} \partial_t p_t(\mathbf{x}) &= \int \partial_t p_t(\mathbf{x} \mid \mathbf{z})\, q(\mathbf{z})\, d\mathbf{z} \\ &= -\nabla \cdot \int p_t(\mathbf{x} \mid \mathbf{z})\, \mathbf{u}_t(\mathbf{x} \mid \mathbf{z})\, q(\mathbf{z})\, d\mathbf{z} = -\nabla \cdot \left(p_t(\mathbf{x})\, \mathbf{u}_t(\mathbf{x}) \right). \end{aligned}$$

最後の変形では $p_t(\mathbf{x})$ を掛けて割り、Equation 38.94 の形を用いている。$\blacksquare$

1 つのブロブしか通過しない場所では、周辺速度はそのブロブ自身の速度に一致する。複数のブロブが通過する場所ではそれらの平均となり、この平均化こそがフローを曲げる原因となる。

38.7.1.2 条件付きフローマッチング定理

Equation 38.93 のターゲットは、サンプルごとに計算できる量の事後平均（posterior mean）Equation 38.94 である。回帰補題（38.6.1.2 節）により、サンプルごとのターゲットに対する最小二乗法はその条件付き平均に適合する。したがってサンプルごとの量に対して直接回帰すればよく、それを行うのが次の**条件付きフローマッチング**（conditional flow matching、CFM）損失である。

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\,\mathbf{z}\sim q,\,\mathbf{x}\sim p_t(\cdot\mid\mathbf{z})}\left[\,\left\|\mathbf{u}_\theta(\mathbf{x},t) - \mathbf{u}_t(\mathbf{x}\mid\mathbf{z})\right\|^2\right]. \tag{38.95}$$

この損失に必要なのは、$(t,\mathbf{z},\mathbf{x})$ のサンプルと閉じた形の条件付き速度（conditional velocity）だけである。

定理（CFM は周辺速度場を訓練する）。$\mathbf{u}_t$ を周辺速度 Equation 38.94 とし、$\mathbb{E}\|\mathbf{u}_t(\mathbf{x}\mid\mathbf{z})\|^2 < \infty$ と仮定する。このとき次が成り立つ。

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathcal{L}_{\mathrm{FM}}(\theta) + C$$

ここで C は θ に依存しない定数である。特に、これら 2 つの目的関数は同じ勾配と同じ最小点（minimizer）を持つ (Lipman et al. 2023; Tong et al. 2024)。

証明。 $X = (\mathbf{x},t)$（ただし $\mathbf{x}\sim p_t(\cdot\mid\mathbf{z})$）と、仮定より二乗可積分な $Y = \mathbf{u}_t(\mathbf{x}\mid\mathbf{z})$ に対して回帰補題を適用する。X の分布は $t\sim\mathcal{U}[0,1]$ かつ $\mathbf{x}\sim p_t$ であり、Y の X が与えられたもとでの条件付き平均は、定義 Equation 38.94 より周辺速度である。補題により $\mathcal{L}_{\mathrm{CFM}}$ は $\mathbb{E}\|\mathbf{u}_\theta(X)-\mathbf{u}_t(\mathbf{x})\|^2 = \mathcal{L}_{\mathrm{FM}}$ と $C = \mathbb{E}\|Y-\mathbf{u}_t(\mathbf{x})\|^2$（$\theta$ を含まない）とに分解される。■

補題が本質的な役割を果たすため証明は 1 行で済み、この定理自体は二乗可積分なターゲットという以外に経路への仮定を必要としない。定理が述べているのは、回帰が正しい最小点を持つということだけである。この最小点が p_1 を p_0 へ（$t=1$ から $t=0$ まで積分したときに）運ぶことは、直前の命題が保証している。定数 C はターゲットの条件付き分散（conditional variance）である。これは 1 つの点を通過するサンプルごとの速度がどれだけ食い違っているかを測る量であり、したがって完璧な速度場であっても CFM 損失は 0 ではなく C に等しくなる。この C を損失の**下限**（floor）と呼ぶ。ノイズ除去スコアマッチング（denoising score matching）も同じ構造をしており、ノイズ付加カーネルのスコアがサンプルごとの量、周辺スコアがその事後平均にあたる。

この損失にはソルバが現れず、訓練中に軌跡（trajectory）を積分することもない。38.3.1 節で 10 回のオイラーステップを通して訓練したモデルは、10 ステップではターゲットに一致したものの、より細かく積分すると外れてしまった。10 ステップのプログラムそのものに適合していたからである。これに対し、今回の回帰が適合させる対象は速度場そのものであり、ステップ数はあとから自由に選べる。

38.7.1.3 スコア、ノイズ、速度

周辺速度の命題を使うには速度が既知のブロブが必要だが、ここまではまだ具体例を挙げていなかった。最初の族はすでに手元にある。前節のガウス経路はこの種の確率経路であり、その上では速度は新しい対象ではない。拡散モデルのように、ノイズを予測するよう訓練されたネットワークがあるとする。速度を得るために別の訓練をやり直す必要があるだろうか。その必要はない。スコア、ノイズ、クリーンな点、そして速度はいずれも同一の事後平均のアフィン関数であり、ノイズを学習したネットワークは残り 3 つもすでに学習しているからである。

データ点 $\mathbf{z}=\mathbf{x}_0$ で条件付けを行い、次の経路をとる。

$$\mathbf{x}_t = \alpha_t\,\mathbf{x}_0 + \sigma_t\,\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{38.96}$$

ここで (α_t, σ_t) は滑らかな係数であり、$(\alpha_0, \sigma_0) = (1, 0)$ かつ $(\alpha_1, \sigma_1) = (0, 1)$ を満たす。$\alpha_t \mathbf{x}_0$ の周りのブロブは幅 σ_t を持ち、$t = 0$ ではデータ点そのもの、$t = 1$ ではノイズとなる。これは前節のノイズ付加カーネルにおいて、ノイズ側の端点を $t = 1$ に置いたものにほかならない。

最小の例として、1 次元のガウスデータ $x_0 \sim \mathcal{N}(0, s_0^2)$ とコサイン経路 $\alpha_t = \cos\frac{\pi t}{2}$、$\sigma_t = \sin\frac{\pi t}{2}$ から始めよう。周辺分布は $\mathcal{N}(0, s_t^2)$（ただし $s_t^2 = \alpha_t^2 s_0^2 + \sigma_t^2$）となり、スコアは $-x/s_t^2$、速度は $u_t(x) = (\dot{s}_t/s_t)\,x$ となる。つまりガウス分布の雲は引き伸ばされるか縮むかしか起こらない（演習問題 3）。$s_0 = 2$ とすると $t = \frac{1}{2}$ で $s_t^2 = 2.5$ となり、位置 x にある粒子の速度は $-0.94\,x$ である。速度場が内側を向いているのは、ノイズ付加によってこの雲が幅 2（$t = 0$）から幅 1（$t = 1$）へと縮むためである。サンプラはこれに逆らって進み、雲を再び幅 2 へと引き伸ばす。$s_0 = 1$ のときはすべての p_t が標準ガウス分布となり、雲はまったく動かない（$u \equiv 0$）。

命題（スコア・速度恒等式）。ガウス経路 Equation 38.96 に対し、任意の時刻 t（ただし $\alpha_t, \sigma_t > 0$）および $\{p_t > 0\}$ 上で、周辺速度は周辺スコアのアフィン関数となる。

$$\mathbf{u}_t(\mathbf{x}) = \frac{\dot{\alpha}_t}{\alpha_t}\,\mathbf{x} - \left(\sigma_t\dot{\sigma}_t - \sigma_t^2\,\frac{\dot{\alpha}_t}{\alpha_t}\right)\nabla \log p_t(\mathbf{x}). \tag{38.97}$$

速度は位置の定数倍とスコアの定数倍の和であり、その両方の係数はスケジュールだけで決まる。データはスコアを通してのみ現れる。スコアの前の括弧内は $\sigma_t^2\,\frac{d}{dt}\log(\sigma_t/\alpha_t)$ に等しく、本書で用いるすべての経路のように信号に対するノイズの比が増加している限り正となる。したがってこの恒等式をスコアについて解くこともでき、一方の場が他方の場を一意に定める。

証明。 両辺が同一の事後平均 $\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t = \mathbf{x}]$ のアフィン関数であることを示す。経路を微分すると条件付き速度 $\dot{\mathbf{x}}_t = \dot{\alpha}_t\mathbf{x}_0 + \dot{\sigma}_t\epsilon$ が得られ、また $\epsilon = (\mathbf{x} - \alpha_t\mathbf{x}_0)/\sigma_t$ である。Equation 38.94 に従って事後平均をとると、次式を得る。

$$\mathbf{u}_t(\mathbf{x}) = \frac{\dot{\sigma}_t}{\sigma_t}\,\mathbf{x} + \left(\dot{\alpha}_t - \alpha_t\,\frac{\dot{\sigma}_t}{\sigma_t}\right)\hat{\mathbf{x}}_0.$$

周辺スコアは各カーネルのスコアの事後平均であり、これは前節で扱った Tweedie の公式にほかならない。カーネル $\mathcal{N}(\alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$ のスコアは $(\alpha_t\mathbf{x}_0 - \mathbf{x})/\sigma_t^2$ であるから、次が成り立つ。

$$\nabla \log p_t(\mathbf{x}) = \mathbb{E}\left[\frac{\alpha_t\mathbf{x}_0 - \mathbf{x}}{\sigma_t^2}\,\middle|\,\mathbf{x}_t = \mathbf{x}\right] = \frac{\alpha_t\hat{\mathbf{x}}_0 - \mathbf{x}}{\sigma_t^2},$$
$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x} + \sigma_t^2\,\nabla \log p_t(\mathbf{x})}{\alpha_t}.$$

この $\hat{\mathbf{x}}_0$ を速度の式に代入し、$\mathbf{x}$ と $\nabla \log p_t$ の項を整理すれば Equation 38.97 が得られる。■

2 つの特別な場合を具体的に書き下しておこう。分散保持（variance-preserving）経路では $\alpha_t^2 + \sigma_t^2 = 1$ であるため $\sigma_t\dot{\sigma}_t = -\alpha_t\dot{\alpha}_t$ となり、スコアの係数は $\dot{\alpha}_t/\alpha_t$ に簡約されて $\mathbf{u}_t(\mathbf{x}) = \frac{\dot{\alpha}_t}{\alpha_t}(\mathbf{x} + \nabla \log p_t(\mathbf{x}))$ となる。前節の分散保持型の確率微分方程式（SDE）では $\alpha_t = \exp\big(-\frac{1}{2}\int_0^t \beta(s)\,ds\big)$ であるから $\dot{\alpha}_t/\alpha_t = -\frac{1}{2}\beta(t)$ となり、速度は $-\frac{1}{2}\beta(t)(\mathbf{x} + \nabla \log p_t(\mathbf{x}))$ となる。これは確率フロー速度 $\mathbf{f} - \frac{1}{2}g^2\nabla \log p_t$（38.5.4 節）において $\mathbf{f} = -\frac{1}{2}\beta\,\mathbf{x}$ かつ $g^2 = \beta$ と

したものに一致する。この経路上では、フローマッチングの速度と確率フロー ODE（probability-flow ODE）は同じ時計の上でまったく同じ場になる。先のガウス分布の例で用いたコサイン経路も分散保持であるため、この式を確かめられる。実際、$\nabla \log p_t = -x/s_t^2$ を代入すると $\frac{\dot{\alpha}_t}{\alpha_t}(1 - 1/s_t^2)\, x = \alpha_t \dot{\alpha}_t (s_0^2 - 1)\, x/s_t^2$ となり、上で求めた伸縮率 $\dot{s}_t/s_t$ が得られる。

本節の残りで用いる直線経路 $\alpha_t = 1 - t$、$\sigma_t = t$ では、この恒等式は $\mathbf{u}_t(\mathbf{x}) = -\big(\mathbf{x} + t\,\nabla \log p_t(\mathbf{x})\big)/(1-t)$ となる。

実際の訓練でネットワークのターゲットとして使われる諸量も、すべてこの同じ事後平均に基づいている（Table 38.2）。

Table 38.2: ガウス経路上におけるネットワークのターゲット。各行は、ネットワークが予測する量、それが推定する事後平均、およびスコアからの変換式を示す。**v** 予測という名称は既存文献に由来する。コサイン経路上では、そのターゲット $\alpha_t \epsilon - \sigma_t \mathbf{x}_0$ は速度ターゲットの $2/\pi$ 倍である。

ネットワークの予測対象	事後平均	スコア $\mathbf{s} = \nabla \log p_t$ からの変換
スコア $\mathbf{s}_\theta$	$\nabla \log p_t(\mathbf{x})$	$\mathbf{s}$
ノイズ $\hat{\epsilon}$	$\mathbb{E}[\epsilon \mid \mathbf{x}_t = \mathbf{x}]$	$-\sigma_t\, \mathbf{s}$
クリーンな点 $\hat{\mathbf{x}}_0$	$\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t = \mathbf{x}]$	$(\mathbf{x} + \sigma_t^2\, \mathbf{s})/\alpha_t$
速度 $\mathbf{u}_\theta$	$\mathbb{E}[\dot{\mathbf{x}}_t \mid \mathbf{x}_t = \mathbf{x}]$	Equation 38.97 の恒等式
v 予測 $\hat{\mathbf{v}}$	$\mathbb{E}[\alpha_t \epsilon - \sigma_t \mathbf{x}_0 \mid \mathbf{x}_t = \mathbf{x}]$	$-\frac{\sigma_t}{\alpha_t}\, \mathbf{x} - \sigma_t\, \frac{\alpha_t^2 + \sigma_t^2}{\alpha_t}\, \mathbf{s}$

どの行も同一の事後平均に関する t 依存のアフィン関数であるため、いずれか 1 つの行で訓練されたネットワークは他のすべてを定める。どの行が最も学習しやすいかは別の問題であり、経路の両端における訓練ターゲットの分散に関わる（演習問題 6）。したがってノイズ予測器もまた 1 つの速度場であり、Table 38.2 の変換を施せば本節のサンプラをそのまま適用できる。

38.7.1.4 例：1 次元上の 2 点

経路の選び方を議論する前に、1 つの例を最後まで計算してみよう。冒頭で挙げた 2 点のデータセットでは、ここまで登場したすべての対象が閉じた形で求まり、38.7.2 節が修復することになる失敗の原因もその式にはっきり現れる。データは $x_0 = \pm a$（それぞれ確率 $\frac{1}{2}$、$a = 2$）とする。これは本章を通じて用いている混合分布の 2 つのモードから幅を取り除いたものである。ノイズは $\epsilon \sim \mathcal{N}(0, 1)$ とし、x_0 とは独立に引く。ここで $\alpha_t = 1 - t$、$\sigma_t = t$ のガウス経路をとると、各データ点はそのノイズ点と一定速度の直線で結ばれる。すなわち $x_t = (1-t)\, x_0 + t\, \epsilon$ であり、速度は $\dot{x}_t = \epsilon - x_0$ となる。

データ点を条件付け変数 $\mathbf{z} = x_0$ としよう。x_0 が与えられたとき、位置 x_t は $\mathcal{N}((1-t)\, x_0, t^2)$ に従い、$t > 0$ で正の密度を持つため周辺速度の命題が適用できる。$\epsilon = (x - (1-t)\, x_0)/t$ を消去すると、速度は現在位置の関数として $u_t(x \mid x_0) = (x - x_0)/t$ と表される。時間を逆向きに走らせると、粒子は $t = 0$ でちょうどデータ点に到着する速さで目的地へ向かう。

この速度を平均するには、x_0 の位置 x における事後分布（posterior distribution）が必要である。$+a$ 対 $-a$ の位置 x における事後オッズは、2 つのガウス密度の比になる。指数部では x^2 と $(1-t)^2 a^2$ の項が打ち消し合って交差項だけが残るため、オッズは $\exp\big(2(1-t)\, ax/t^2\big)$ となり、事後平均は $\mathbb{E}[x_0 \mid x_t = x] = a \tanh\big((1-t)\, ax/t^2\big)$ となる。したがって周辺速度は次式で与えられる。

$$u_t(x) = \frac{x - a \tanh\!\big(\frac{(1-t)\, ax}{t^2}\big)}{t}. \tag{38.98}$$

時間を逆向きに走らせると、位置 x にある雲は、x から見てそれぞれのターゲットがどれだけ尤もらしいかで重み付けした平均点へ向かって、時間どおりに到着する速さで進む。$(t, x) = (0.5, 0.5)$ では速度は -2.86 であり、$(0.7, -1)$ では 0.97 である。Equation 38.98 からは 3 つの性質が読み取れ、Figure 38.27 はそのうちの最初の 2 つを描いている。

- **時刻 $t = 1$ では場は $u_1(x) = x$ となる。** ノイズサンプルはターゲットに関する情報をまったく持たないため、事後平均はデータの平均である 0 になる。したがって $t = 1$ から $t = 0$ への 1 回のオイラーステップ $x \mapsto x - u_1(x)$ は、**すべての**サンプルを 0 へ送ってしまう。これは独立ペアリングのもとでは任意のデータに対して成り立つ。すなわち $\mathbf{u}_1(\mathbf{x}) = \mathbb{E}[\epsilon - \mathbf{x}_0 \mid \epsilon = \mathbf{x}] = \mathbf{x} - \mathbb{E}[\mathbf{x}_0]$ であるため、厳密な速度場による 1 ステップは必ずデータ平均の上に着地する。
- **原点は最初引き寄せ、のちに反発する。** t が減少するときサンプラは u_t に逆らって動くため、原点での正の傾きは粒子を原点へ引き寄せ、負の傾きは原点から押し出す。傾きは $\partial_x u_t(0) = \big(1 - (1-t)\,a^2/t^2\big)/t$ であり、$+0.56$ $(t = 0.9)$、0 $(t^* = 2\sqrt{2} - 2 \approx 0.83)$、すなわち $(1-t)\,a^2 = t^2$ の根)、-0.31 $(t = 0.8)$、-14 $(t = 0.5)$ と変化する。0 の近くにある粒子は、ドリフトによって 0 へ向かい続け、t が t^* まで下がってからようやくいずれかのモードへ向けて離れていくため、フローがモードを決める時期は遅い。原点から離れた場所では tanh が飽和し、$u_t \approx (x \mp a)/t$ となって条件付き速度そのものに近づく。
- **損失の下限はノイズの近くで生じる。** ターゲット $\epsilon - x_0$ の二乗平均は $a^2 + 1 = 5$ である。このうち $C = \mathbb{E}_t \mathbb{E}_x \mathrm{Var}(x_0 \mid x_t)/t^2 \approx 2.0$ が損失の下限であり（ここで $\mathrm{Var}(x_0 \mid x_t = x) = a^2 \,\mathrm{sech}^2\big((1-t)\,ax/t^2\big)$)、残りの 3.0 が周辺速度の二乗平均である。$C$ のほとんどすべては $t > 0.5$ の領域から生じる。そこでは 2 つのブロブが重なり合っており、位置を見ただけでは粒子がどちらのデータ点へ向かっているのか判別できないからである。$t = 0.4$ を下回ると行き先が判別できるようになり、条件付き速度 $(x - x_0)/t$ が限りなく大きくなるにもかかわらず、そこで生じる下限への寄与は全体の約 1,000 分の 1 にすぎない（演習問題 2)。

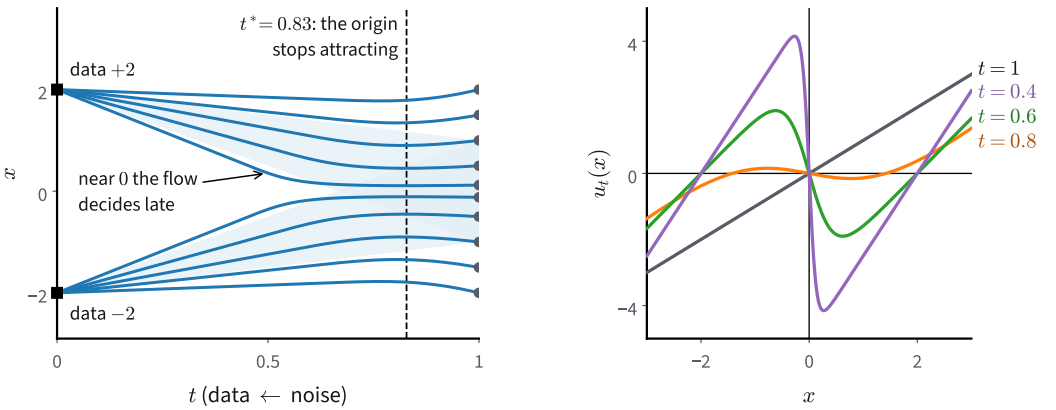


Figure 38.27: ± 2 にある 2 つのデータ点と、独立ペアリングのもとでの標準ガウスノイズ。サンプリングは $t = 1$ のノイズ（右）から $t = 0$ のデータ（左）へと進む。(a) 2 つの条件付きブロブの 1 標準偏差帯の上に重ねた厳密な周辺フローの軌跡。t が $t^* = 0.83$ まで下がる間はすべての軌跡が 0 に向かって動き、0 の近くから出発した軌跡は $t \approx 0.6$ を下回ってようやくモードへ向けて離れる。(b) 速度 $u_t(x)$ は x（$t = 1$ のとき）であり、t が下がるにつれて原点での傾きは負かつ急峻になり、原点から離れると $(x \mp 2)/t$ に漸近する。

次のコードセルでは、2 番目の性質で挙げた 3 つの時刻において原点での傾きを有限差分（finite difference）で求める。続いて 1 番目の性質を検証し、さらにステップ数を増やすとどうなるか

を調べる。ノイズ分布全体を出発点として、厳密な速度場を $t = 1$ から $t = 0$ まで 1、2、4、8 回の等間隔オイラーステップで積分し、データ点から 0.1 以内に着地する質量の割合と終端での $|x|$ の平均を表示する。ノイズ分布はガウス密度で重み付けした細かい開始点グリッドとして表しているため、出力にランダム性は一切ない。

```
a = 2.0                                    # data at -a and +a, noise N(0, 1)
def u_two_point(x, t):                     # closed form of the marginal velocity
    return (x - a * np.tanh((1 - t) * a * x / t**2)) / t

ts = np.array([0.9, 0.8, 0.5])
print('slope at the origin at t = 0.9, 0.8, 0.5:',
      ((u_two_point(1e-6, ts) - u_two_point(-1e-6, ts)) / 2e-6).round(2),
      '   t* =', round(2 * np.sqrt(2) - 2, 4))
x1 = np.linspace(-6, 6, 4001)              # the noise law N(0, 1), as a weighted grid
w = np.exp(-x1**2 / 2) / np.exp(-x1**2 / 2).sum()
for K in (1, 2, 4, 8):                     # K equal Euler steps from t = 1 to t = 0
    x, ts = x1, np.linspace(1, 0, K + 1)
    for t, t_next in zip(ts[:-1], ts[1:]):
        x = x - (t - t_next) * u_two_point(x, t)
    print(f'{K} Euler step(s): mass within 0.1 of +-a {w @ (abs(abs(x) - a) < 0.1):.3f}, '
          f'mean |x| {w @ abs(x):.2f}')
```

```
slope at the origin at t = 0.9, 0.8, 0.5: [  0.56  -0.31 -14.  ]    t* = 0.8284
1 Euler step(s): mass within 0.1 of +-a 0.000, mean |x| 0.00
2 Euler step(s): mass within 0.1 of +-a 0.359, mean |x| 1.48
4 Euler step(s): mass within 0.1 of +-a 0.987, mean |x| 1.99
8 Euler step(s): mass within 0.1 of +-a 0.999, mean |x| 2.00
```

傾きは式どおりの値となる。1 回のオイラーステップのあとではデータ点の近くに到達する質量は皆無であり、$|x|$ の平均は 0 となる。すなわち、すべてのサンプルがデータ平均の上に集まってしまう。こうして冒頭の願いは最初のテストで潰える。個々の条件付き経路はすべて直線であるにもかかわらず、周辺フローは曲率を持ち、1 回の大きなステップで済ませようとすると全サンプルが平均へと潰れてしまうのである。

ステップ数を増やせば、この破綻は急速に修復される。2 ステップで質量の 3 分の 1 強がデータ点に届き、4 ステップで約 1% を除くほぼすべてが、8 ステップで事実上すべてが到達する。これほど少ないステップ数で足りるのは、どのようなグリッドでも最後の 1 ステップが事後平均 $a\tanh(\cdot)$ の上に着地し、それ以前のステップで粒子が原点から十分遠くへ動いてさえいればその値が $\pm a$ になるからである（演習問題 2）。しかし、そもそも節約したかったのはそのステップ数である。フローそのものが真っ直ぐになり、1 ステップで済むようにノイズとデータをペアリングする方法が 38.7.2 節の主題である。

38.7.2 Rectified Flow

38.7.2.1 直線経路とその交差

2 点の例では直線の条件付き経路を用いた。d 次元でも同じ構成により、データサンプル $\mathbf{x}_0$ と独立に引いたノイズサンプル ϵ をペアにし、一定速度で進む直線で両者を結ぶ。

$$\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\epsilon, \qquad \mathbf{u}_t(\mathbf{x}_t \mid \mathbf{z}) = \dot{\mathbf{x}}_t = \epsilon - \mathbf{x}_0, \qquad \mathbf{z} = (\mathbf{x}_0, \epsilon). \tag{38.99}$$

条件付き速度は t に依存せず、CFM 損失 Equation 38.95 は次の **Rectified Flow** の目的関数とな

る (Liu et al. 2023; Lipman et al. 2023)。

$$\mathcal{L}_{\mathrm{RF}}(\theta) = \mathbb{E}_{t,\,\mathbf{x}_0\sim p_0,\,\epsilon\sim p_1} \Big[\big\| \mathbf{u}_\theta\big((1-t)\mathbf{x}_0 + t\epsilon,\, t\big) - (\epsilon - \mathbf{x}_0) \big\|^2 \Big]. \tag{38.100}$$

訓練ではデータとノイズを引き、両者を補間して、その差に対して回帰する。アルゴリズムはこれだけである。この名称はその先を見据えている。**rectified** は「真っ直ぐにされた（整流された）」という意味であり、ペアリングが修正されたあとのフローはまさにそのとおり真っ直ぐになる。しかし先ほど見たように、独立ペアからなるフローは真っ直ぐではない。

技術的な点を 1 つ片付けておく必要がある。ガウスノイズであれば、2 点の例のように $\mathbf{x}_0$ だけで条件付けでき、周辺速度の命題がそのまま適用できる。あとでペアリングを変更できるようにするにはペアで条件付けする必要があるが、ペアを固定すると直線上を滑る点は密度を持たないため、命題はそのままの形では適用できない。それでも結論自体は、期待値 $\mathbb{E}\,\varphi(\mathbf{x}_t)$（ただし φ は滑らかなテスト関数）に対して同じ計算を行うことで成り立つ（演習問題 7）。

なぜフローは曲がるのだろうか。$(\mathbf{x}, t)$ で交差する 2 本の線分は、事後平均 Equation 38.94 に 2 つの異なる向きを与える（Figure 38.28）。関数は各点で 1 つの値しか持てないため、速度場はそれらを平均する。一方、速度場が $\mathbf{x}$ についてリプシッツ連続である常微分方程式は、各点を通る軌跡をただ 1 つしか持たないため（38.1.1.3 節）、学習されたフローの 2 本の軌跡が決して交わることはない。線分が交差する場所では、軌跡は交わる代わりに互いを避けるように曲がるのである。

目指すのは、すべての軌跡が一定速度で進む直線となるような速度場である。そのような場は各軌跡に沿って一定となるため、任意の長さのオイラーステップ 1 回で厳密になる。独立ペアリングはこの理想からほど遠い。損失は正しい速度場によって最小化されており、個々の要素もこれ以上ないほど真っ直ぐであるから、変更できる残された自由度は、どのノイズ点をどのデータ点と結ぶかという点だけである。

38.7.2.2 例：Two Moons

2 点の例には解析解があり、1 ステップでは失敗することがその式からわかった。では解析解を持たないデータに対して、訓練されたネットワークは何ステップを必要とするのだろうか。平面上で 2 つの三日月が噛み合う two-moons データを取り上げ、小さなニューラルネットワークを訓練してステップ数を数えてみよう。数えるにはサンプルの良さを測るスコアが必要であり、ここでは有限の一次モーメント（first moment）を持つ分布 P と Q の間の**二乗エネルギー距離**（squared energy distance）(Székely and Rizzo 2013) を用いる。

$$\mathcal{E}^2(P, Q) = 2\,\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|,$$

ここで $X, X' \sim P$ と $Y, Y' \sim Q$ は互いに独立である。この量は、P の点と Q の点の間の平均距離を、それぞれの分布内部での平均距離と比較する。Székely と Rizzo はこの式そのものをエネルギー距離と呼んでいるが、距離の公理を満たすのはその平方根であるため、本書では二乗を付けて呼ぶ。これは二乗 MMD（37.4.1 節）の一種であり、$P = Q$ の場合に限り 0 となる。2 組のサンプルから不偏 U 統計量によって推定し、以下で示すスコアはすべてこの二乗距離である。

次のコードセルでは、座標ごとに平均 0・分散 1 に標準化したデータ、2,048 点からなるホールドアウトデータ（held-out data）のサンプル、および以降のすべての比較で再利用する 1 つのノイズ配列 `z_eval` を作成する。続いてホールドアウトサンプルに対する 2 つの基準スコアを計算する。1 つはデータから新たに引いたサンプルであり、もう 1 つは厳密な速度場のオイラーステップ 1 回が生成するデータ平均上の点質量である。

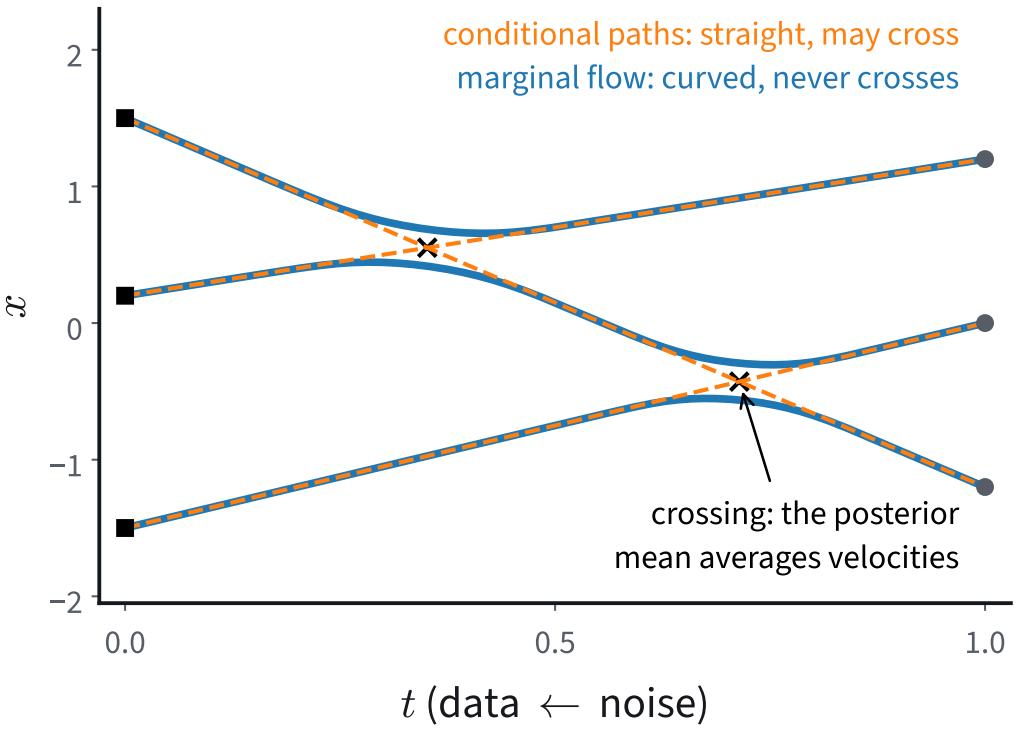


Figure 38.28: 直線分（破線）で結ばれた 1 次元の 3 組のデータ・ノイズペア。うち 2 組が交差している（×）。線分が交差する場所では周辺速度がそれらの向きを平均するため、$t = 1$ のノイズから $t = 0$ のデータへと走らせたフローの軌跡（実線）は曲がり、決して交わらない。各軌跡は元の線分とは異なるデータ点に到達し、出発した順序を保ったまま到着する。

```
def two_moons(n, rng, noise=0.07):
    t = rng.uniform(0.0, np.pi, n)
    m = rng.integers(0, 2, n)                       # which moon
    X = np.stack([np.cos(t) * (1 - 2 * m) + m,
                  np.sin(t) * (1 - 2 * m) + 0.5 * m], axis=1)
    return X + noise * rng.standard_normal((n, 2))

rng = np.random.default_rng()
raw = two_moons(8192, rng)
mu, sd = raw.mean(0), raw.std(0)
moons = (raw - mu) / sd                             # standardized training data
held_out = (two_moons(2048, rng) - mu) / sd
z_eval = rng.standard_normal((2048, 2))             # noise for every comparison

def energy_distance(a, b):                          # squared energy distance, U-statistic
    d = lambda u, v: np.linalg.norm(u[:, None, :] - v[None, :, :], axis=-1)
    n, m = len(a), len(b)
    return (2 * d(a, b).mean() - d(a, a).sum() / (n * (n - 1))
            - d(b, b).sum() / (m * (m - 1)))

print(f'fresh data sample vs held-out: {energy_distance(moons[:2048], held_out):+.4f}')
print(f'point mass at the data mean vs held-out: '
      f'{energy_distance(np.zeros((2048, 2)), held_out):.3f}')
```

```
fresh data sample vs held-out: -0.0000
point mass at the data mean vs held-out: 0.904
```

同じ分布からの 2 つのサンプルのスコアは 0 から 0.001 の数倍以内に収まり、これがこの標本サイズにおけるこの指標の分解能である。点質量のスコアは約 0.9 となる。

モデルは $(\mathbf{x}, t)$ を受け取って速度を返す小さな tanh ネットワークであり、`field` がそれを評価する。あとでペアリングを変更できるよう、関数 `train_rf` はペアのサンプラを引数にとる。単一のバッチでは値がばらつくため最後の 500 バッチにわたり、さらに 2 つの座標にわたって平均した訓練損失をネットワークとともに返す。したがって表示される値は $\mathcal{L}_{\mathrm{RF}}/2$ である。

```
def make_net():                         # 3 -> 256 -> 256 -> 2, Glorot weights, zero biases
    net = nn.Sequential(nn.Linear(3, 256), nn.Tanh(), nn.Linear(256, 256), nn.Tanh(),
                        nn.Linear(256, 2))
    for layer in net[::2]:
        nn.init.xavier_uniform_(layer.weight)
        nn.init.zeros_(layer.bias)
    return net

def field(net, x, t):                   # the time enters centered, as 2t - 1
    return net(torch.cat([x, 2 * t - 1], 1))

def train_rf(sample_pairs, steps=10000, batch=2048, tail=500):
    """Regress v(x_t, t) on eps - x0 (papers: -(x1 - x0) at time 1 - t).
    Return the net and the mean tail loss."""
    net = make_net()
    opt = torch.optim.Adam(net.parameters(), lr=3e-3)
    sched = torch.optim.lr_scheduler.CosineAnnealingLR(opt, steps)   # 3e-3 -> 0
```

```
    tail_sum = torch.zeros(())
    for step in range(steps):
        x0, eps = sample_pairs(batch)              # data, noise
        t = torch.rand(batch, 1)
        xt = (1 - t) * x0 + t * eps                # a point on each segment
        loss = ((field(net, xt, t) - (eps - x0))**2).mean()
        opt.zero_grad()
        loss.backward()
        opt.step()
        sched.step()
        if step >= steps - tail:
            tail_sum += loss.detach()              # stays on the device
    return net, (tail_sum / tail).item()
```

最初の実行ではノイズとデータを独立にペアリングする。CFM 定理により損失は 0 に到達できず、損失が下げ止まる位置は線分がどれだけ交差しているかを測る指標となる。下限 C は、$\epsilon - \mathbf{x}_0$ の $\mathbf{x}_t$ が与えられたもとでの条件付き分散を t にわたって平均したものである。訓練を始める前にこの値を知ることはできるだろうか。有限の訓練セットであれば、ネットワークを使わずに計算できる。$\mathbf{x}_t$ が与えられたとき、$\mathbf{x}_0$ の事後分布は各訓練点 $\mathbf{x}_0'$ に $\exp\left(-\|\mathbf{x}_t - (1-t)\,\mathbf{x}_0'\|^2/2t^2\right)$ に比例する重みを割り当てる。ターゲットは $\epsilon - \mathbf{x}_0 = (\mathbf{x}_t - \mathbf{x}_0)/t$ であるから、その条件付き分散は $\mathbf{x}_0$ の事後分散（posterior variance）を t^2 で割ったものになる。関数 `pairing_floor` は $4{,}000$ 回抽出した $(t, \mathbf{x}_0, \epsilon)$ にわたってこの量を平均し、`train_rf` が座標あたりの損失を報告するのに合わせて値を半分にする。

```
def pairing_floor(data, rng, m=4000, chunk=500):
    """Floor C of independent pairing, per coordinate: E Var(eps - x0 | x_t) / 2."""
    t = rng.uniform(0, 1, (m, 1))
    xt = (1 - t) * data[rng.integers(0, len(data), m)] + t * rng.standard_normal((m, 2))
    var = []
    for i in range(0, m, chunk):              # posterior of x0 over all training points
        ti, xi = t[i:i + chunk], xt[i:i + chunk]
        d2 = ((xi**2).sum(1, keepdims=True) - 2 * (1 - ti) * xi @ data.T
              + (1 - ti)**2 * (data**2).sum(1))          # |x_t - (1 - t) x0'|^2
        logw = -d2 / (2 * ti**2)
        r = np.exp(logw - logw.max(1, keepdims=True))
        r /= r.sum(1, keepdims=True)
        v = r @ (data**2).sum(1) - ((r @ data)**2).sum(1)  # Var(x0 | x_t), summed
        var.append(v / ti[:, 0]**2)           # eps - x0 = (x_t - x0) / t
    return np.concatenate(var).mean() / 2

print(f'floor of the independent pairing, per coordinate: {pairing_floor(moons, rng):.2f}')
```

```
floor of the independent pairing, per coordinate: 1.39
```

下限は座標あたり約 1.37 であり、推定値は抽出ごとに小数第 2 位が変動する。完璧なネットワークであればここで下げ止まる。今回用いるネットワークは、隠れ層 2 層（各 256 ユニット）を持ち、$10{,}000$ ステップ（各ステップ $2{,}048$ ペアのミニバッチ）の訓練によってこの下限に近づく。

```
data = torch.tensor(moons, dtype=torch.float32)
def independent_pairs(n):                      # random data rows, fresh noise
    return data[torch.randint(0, len(data), (n,))], torch.randn(n, 2)
```

```
net, plateau = train_rf(independent_pairs)
print(f'CFM loss, mean of the last 500 batches: {plateau:.2f}')
```

```
CFM loss, mean of the last 500 batches: 1.40
```

損失は座標あたり約 1.4 で落ち着き、下限との差は 0.01 の数倍以内に収まる。したがって、損失が 1.4 であることは当てはまりが悪いことを意味しない。そのほとんどすべては交差する線分間の食い違いであり、いかなるネットワークも取り除くことはできない。CFM 定理により、速度場の誤差にあたるのは下限を超える過剰分だけである。

もっとも、その 0.01 の数倍という差は見かけほど小さくはない。ターゲットの座標あたりの二乗平均（二次モーメント）は 2（ノイズと標準化データから 1 ずつ）であり、そのうち 1.37 が下限である。したがって速度場自体の二乗平均は約 0.6 にすぎず、誤差はその数パーセントに相当する。この誤差には、あとで生成サンプルの中で再び出会うことになる。

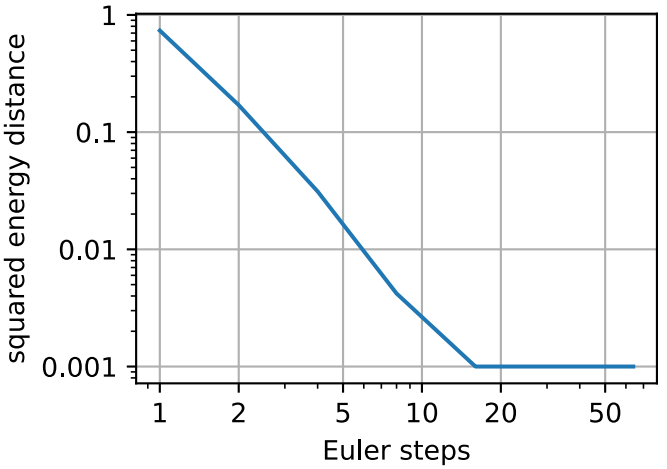
サンプリングでは、$\dot{\mathbf{x}} = \mathbf{u}_\theta(\mathbf{x}, t)$ を $t = 1$ から 0 までオイラー法（38.2 節）で積分する。すなわち K 回のステップ（ステップ幅 $1/K$）を行い、各ステップでは $\mathbf{u}_\theta(\mathbf{x}, t)/K$ を $t = 1, 1 - 1/K, \dots, 1/K$ において引く。関数 `euler_path` は形状 `(steps + 1, n, 2)` の軌跡全体を返し、その最後の行が生成サンプルとなる。

```
@torch.inference_mode()
def euler_path(net, z, steps):
    q = torch.tensor(z, dtype=torch.float32)
    path = [q]
    for k in range(steps):                    # t = 1, 1 - 1/steps, ..., 1/steps
        t = torch.full((len(z), 1), 1 - k / steps)
        q = q - field(net, q, t) / steps
        path.append(q)
    return torch.stack(path).numpy()
```

厳密な速度場の 1 ステップは、スコアが 0.9 であった点質量を与えるはずである。訓練された速度場について、次のコードセルはノイズ配列 `z_eval` を $K = 1, 2, 4, \dots, 64$ ステップで積分し、ホールドアウトデータとの二乗エネルギー距離を両対数軸にプロットする。推定量は不偏であるため指標の分解能付近では負になり得るが、プロットでは 10^{-3} でクリップしている。

```
steps_list = [1, 2, 4, 8, 16, 32, 64]
eds = [energy_distance(euler_path(net, z_eval, K)[-1], held_out) for K in steps_list]
print('  '.join(f'{K}: {e:.3f}' for K, e in zip(steps_list, eds)))
d2l.plot(steps_list, np.maximum(eds, 1e-3), 'Euler steps',
         'squared energy distance', xscale='log', yscale='log')
```

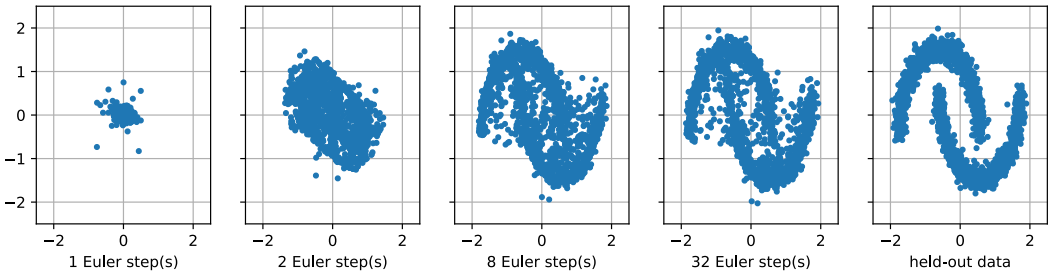
```
1: 0.736  2: 0.170  4: 0.031  8: 0.004  16: -0.000  32: -0.001  64: -0.001
```



1 ステップのスコアは 0.7〜0.75 となり、平均上の点質量（0.9）と同程度である。つまりネットワークは、厳密な速度場が $t = 1$ で引き起こす崩壊を不完全ながら学習している。2 ステップでは 0.16〜0.19、4 ステップでは 0.03〜0.04、8 ステップでは約 0.01 以下となり、16 ステップ以降ではスコアは 0 から 0.001 の数倍以内に留まる。この減少分はオイラー法の離散化誤差である。あとに残るのは訓練された速度場自体の誤差であり、2,048 点でこの指標が分解できる限界まで小さくなっている。ソルバを介さずに適合させた速度場であるから、ステップ数を増やすほど精度が向上するのは当然である。対照的に、38.3.1 節で 10 回のオイラーステップを通して訓練した場は、ステップを細かくするとかえって悪化したのであった。

スコアは 1 つの数値にすぎないため、それが何を要約しているのかを実際に見ておく価値がある。次のコードセルでは、1、2、8、32 ステップの生成サンプルと、その横にホールドアウトデータを描画する。

```
panels = [(f'{K} Euler step(s)', euler_path(net, z_eval, K)[-1]) for K in (1, 2, 8, 32)]
fig, axes = d2l.plt.subplots(1, 5, figsize=(12, 2.6), sharex=True, sharey=True)
for ax, (title, s) in zip(axes, panels + [('held-out data', held_out)]):
    d2l.plot(s[:, 0], s[:, 1], title, fmts=('.',),
             xlim=(-2.5, 2.5), ylim=(-2.5, 2.5), axes=ax)
```



1 ステップでは平均周りのブロブとなり、2 ステップでは斜めの帯になる。8 ステップで雲は 2 つの三日月形へと広がるが、その間にはまだ多くの点が散らばっている。32 ステップになるとそれらの大半が消え、最後のパネルのデータのように 2 つの分離した三日月が描かれる。もっとも、データと並べて見比べると三日月はわずかに太く、隙間にも少数の点が残っている。ステップ数をこれ以上増やしても消えないため、これらの欠陥はソルバではなく速度場自体に属するもので

ある。損失とその下限との間にあった 0.01 の数倍という差は、サンプル上ではこのような姿となって現れる。

スコアはこうした細部の欠陥をほとんど捉えない。点間の平均距離を比較する指標であるため、輪郭がどれほど鋭いかよりも先に質量がどこにあるかを反映し、指標の分解能に達したスコアであってもデータと同等の鋭さまで保証するわけではないからである。ステップ数を増やすことで確実に取り除かれるのはソルバの誤差であり、スコアで見る限りそれにはサンプルあたり 16〜32 回のネットワーク呼び出しを要する。これが、独立ペアリングによってフローに組み込まれた曲率のコストである。

38.7.2.3 Reflow

フローを曲げている原因は独立ペアリングにあり、訓練済みのモデルはより良いペアリングを提供してくれる。新たなノイズからモデルの ODE を走らせ、各ノイズ点 ϵ をその到達端点 $\hat{\mathbf{x}}_0(\epsilon)$ とペアにするのである。単一のフローの軌跡は決して交差しないため、こうして作ったペアは独立ペアよりもはるかに交差しにくい。この新しいペアの上で同じアーキテクチャを再訓練する手続きを **Reflow** と呼ぶ (Liu et al. 2023)。

ペアを組み直すことで何か悪影響は生じないだろうか。端点の分布が崩れたり、以前より遠い点どうしがペアになったりする懸念があるが、実際にはそのどちらも起こらない。ペアリングが点をどれだけ遠くへ動かすかは、**輸送コスト**（transport cost）$\mathbb{E}\,c(\mathbf{x}_1-\mathbf{x}_0)$、すなわち変位の凸関数 c（二乗長さなど）をペア全体で平均した値で測られる。以下の命題では大文字でランダムな経路を表す。$\mathbf{X}_t$ はペアを結ぶ直線分であり、$\mathbf{Z}_t$ は同じノイズ点から出発するフローの軌跡である。

命題（Reflow は周辺分布を保ち輸送コストを下げる）。 データ点とノイズ点のペア $(\mathbf{X}_0,\mathbf{X}_1)$ が有限の二次モーメントを持つとし、$\mathbf{X}_t=(1-t)\mathbf{X}_0+t\mathbf{X}_1$ とおき、$\mathbf{u}_t$ をこの直線経路の周辺速度とし、$\mathbf{Z}_t$ をノイズ側の端点から時間を逆向きに $\dot{\mathbf{Z}}_t=\mathbf{u}_t(\mathbf{Z}_t)$ を解いた解（ただし $\mathbf{Z}_1=\mathbf{X}_1$）とする。このフローが存在するよう、$\mathbf{X}_t$ は任意の $t>0$ で正の密度を持ち、$\mathbf{u}_t$ は連続微分可能かつ $\mathbf{x}$ について（t に依らない定数で）リプシッツ連続であると仮定する。このとき、(i) $\mathbf{Z}_t$ は $\mathbf{X}_t$ と同じ分布に任意の t で従い、特に $\mathbf{Z}_0\sim p_0$ となる。(ii) 任意の凸関数 c に対して次が成り立つ。

$$\mathbb{E}\,c(\mathbf{Z}_1-\mathbf{Z}_0)\;\le\;\mathbb{E}\,c(\mathbf{X}_1-\mathbf{X}_0). \tag{38.101}$$

証明。 (i) $\mathbf{X}_t$ の分布は $\mathbf{u}_t$ とともに連続の式を満たす。これは $\mathbf{z}=(\mathbf{X}_0,\mathbf{X}_1)$ として点質量も扱える弱形式の周辺速度命題（演習問題 7）を適用した結果である。$\mathbf{Z}_t$ の分布もまた、$\mathbf{Z}_t$ が速度場 $\mathbf{u}_t$ に従って動くため同じ連続の式を満たす。両者はともに p_1 に $t=1$ で一致し、このような速度場を持つ連続の式の 2 つの解はある時刻で一致すればすべての時刻で一致する。この一意性だけは完全には証明しないステップである。38.5.4.3 節では滑らかな正の密度に対して各軌跡上で両方の対数密度が変化率 $-\nabla\cdot\mathbf{u}_t$ で変化することからこれを示したが、ここでは弱形式での再証明は省いてそのまま用いる。(ii) 変位を時間積分として書き、イェンセンの不等式（Jensen's inequality）を 2 回適用する。

$$\begin{aligned}\mathbb{E}\,c(\mathbf{Z}_1-\mathbf{Z}_0)&=\mathbb{E}\,c\Big(\int_0^1\mathbf{u}_t(\mathbf{Z}_t)\,dt\Big)\le\int_0^1\mathbb{E}\,c(\mathbf{u}_t(\mathbf{Z}_t))\,dt\\&=\int_0^1\mathbb{E}\,c\big(\mathbb{E}[\mathbf{X}_1-\mathbf{X}_0\mid\mathbf{X}_t]\big)\,dt\le\mathbb{E}\,c(\mathbf{X}_1-\mathbf{X}_0).\end{aligned}$$

1 つ目の不等号は時間平均に対するイェンセンの不等式である。等号では (i) を用いて $\mathbf{Z}_t$ を $\mathbf{X}_t$ に置き換え、続いて周辺速度の定義を用いている。最後の不等号は条件付きイェンセンの不等式である。■

$c = \|\cdot\|^2$ とすると、最後のステップでちょうど条件付き分散の分だけ減少するため、コストは少なくとも元のペアリングの下限の分だけ下がる。すなわち $\mathbb{E}\|\mathbf{Z}_1 - \mathbf{Z}_0\|^2 \le \mathbb{E}\|\mathbf{X}_1 - \mathbf{X}_0\|^2 - C$ である。訓練時には厄介者であった下限 C が、1 ラウンドの Reflow によって保証されるコスト削減量となる。線分がまったく交差しない場合には下限は 0 であり、再ペアリングで得られる改善はない。

この命題の仮定は、ここでの具体例が満たす条件よりも強い。2 点の例や、ガウスノイズと独立にペアリングした任意の有限訓練セットでは、周辺速度場は任意の $[\delta, 1]$（各 $\delta > 0$）上ではリプシッツ連続だが、$t = 0$ まではそうならない。実際、2 点の例の速度場における傾き -14（$t = 0.5$）は、$t \to 0$ で限りなく大きくなる。そのため上の証明がそのままカバーするのは時刻 δ までのフローに限られ、(i) は $t \ge \delta$ に対して、(ii) は $\mathbb{E}\, c\big((\mathbf{Z}_1 - \mathbf{Z}_\delta)/(1-\delta)\big) \le \mathbb{E}\, c(\mathbf{X}_1 - \mathbf{X}_0)$ の形で得られる。極限 $\delta \to 0$ の移行は証明しないため、その意味で以下の例は命題の適用範囲を一歩越えている。

2 点の例では結論を直接確かめられる。1 次元のフローは順序を保つため、1 ラウンドの Reflow は各ノイズ点 ϵ を $a\,\mathrm{sign}(\epsilon)$ とペアにする（演習問題 4）。このペアリングのコストは $a^2 - 2a\sqrt{2/\pi} + 1 = 1.81$ と、独立ペアリングの $a^2 + 1 = 5$ より小さく、しかも線分が決して交差しないため 2 ラウンド目では $C = 0$ となり、その速度場は 1 回のオイラーステップで厳密に積分される。

two-moons データに対しては、次のコードセルで新たに 8,192 個のノイズ点を引き、訓練済みモデルを 32 ステップ積分してその端点を求め、この固定されたペアの上で `train_rf` により新しいネットワークを訓練する。

```
eps_pairs = rng.standard_normal((8192, 2))            # noise endpoints, kept
x0_pairs = euler_path(net, eps_pairs, 32)[-1]         # the model's own endpoints
X0, Eps = (torch.tensor(p, dtype=torch.float32) for p in (x0_pairs, eps_pairs))
def model_pairs(n):                                   # the same pairs, never re-paired
    i = torch.randint(0, len(X0), (n,))
    return X0[i], Eps[i]

net2, plateau2 = train_rf(model_pairs)
print(f'reflow loss, mean of the last 500 batches: {plateau2:.4f}')
```

```
reflow loss, mean of the last 500 batches: 0.0007
```

最初の損失が 1.4 で下げ止まっていたのに対し、Reflow の損失は 1,000 分の 1 未満にまで下がっており、新しい線分はほとんど交差していない。交差のない速度場は軌跡に沿ってほぼ一定となるため、少数のオイラーステップで十分なはずである。次のコードセルでは、同じノイズ配列から両方のモデルを 1、2、32 ステップでサンプリングする。

あわせて、命題が上界を与えるコストである旧ペアと新ペアの平均二乗長さも表示する。旧ペアのコストは 4 である。ノイズと標準化データはそれぞれ 2 つの座標の各々で分散 1 を持ち、独立な抽出では分散がそのまま足し合わされるからである。独立ペアリングの下限を 2 座標分合計すると、先に表示した値の 2 倍、すなわち約 2.75 となる。したがって厳密な周辺フローであれば、命題は新しいペアのコストが高々約 1.25 になることを保証する。

```
for K in (1, 2, 32):
    e1 = energy_distance(euler_path(net, z_eval, K)[-1], held_out)
    e2 = energy_distance(euler_path(net2, z_eval, K)[-1], held_out)
    print(f'{K:2d} Euler step(s): squared energy distance  original {e1:.3f}'
          f'  reflowed {e2:.3f}')
```

```
x0_indep = moons[rng.integers(0, len(moons), len(eps_pairs))]
print(f'mean squared pair length: independent {np.mean(np.sum((eps_pairs - x0_indep)**2,
↪  1)):.2f}'
      f', reflowed {np.mean(np.sum((eps_pairs - x0_pairs)**2, 1)):.2f}')
```

```
 1 Euler step(s): squared energy distance  original 0.736  reflowed -0.001
 2 Euler step(s): squared energy distance  original 0.170  reflowed -0.001
32 Euler step(s): squared energy distance  original -0.001  reflowed -0.001
mean squared pair length: independent 4.04, reflowed 0.29
```

Reflow 後のモデルのオイラーステップ 1 回は、元のモデルの 32 ステップと同じスコア、すなわち高々 1,000 分の 1〜2 という指標の分解能限界を達成する（元のモデルの 1 ステップは約 0.7 であった）。これ以上良くなりようはない。Reflow 後のネットワークは元のモデルの 32 ステップ端点を再現するよう訓練されているため、最初のモデルのサンプルと一緒にその誤差も受け継ぐからである。

ペアの平均二乗長さは 4 から約 0.3 へと低下し、上界が保証する 1.25 を大きく下回る。この余裕は証明の最初のステップから生じている。最初のフローの曲がった軌跡上では、時間平均に対するイェンセンの不等式が等号からほど遠いのである。

各 32 ステップ軌跡と、その両端を結ぶ弦との平均距離を計算すれば、軌跡の直線性も測ることができる。

```
def chord_gap(P):                                   # P: (steps + 1, n, 2) trajectory
    chord = P[0] + (P[-1] - P[0]) * np.linspace(0, 1, len(P))[:, None, None]
    return np.linalg.norm(P - chord, axis=-1).mean()

print(f'mean distance from the chord: original {chord_gap(euler_path(net, z_eval, 32)):.3f}'
      f', reflowed {chord_gap(euler_path(net2, z_eval, 32)):.3f}')
```

```
mean distance from the chord: original 0.249, reflowed 0.005
```

元の軌跡は弦から平均して約 4 分の 1 単位ほど外れる。最初はオイラーステップ 1 回が着地した平均の方向へ向かって出発し、あとになってようやく三日月の方へと向きを変えるためである。一方、Reflow 後の軌跡は弦から 0.005 以内に収まっており、約 50 倍も直線に近い。1 ステップで十分な理由はここにある。Reflow はフローを真っ直ぐにするが、最適にするとは限らない。2 次元以上では、フローが真っ直ぐで交差が一切なくても、必要以上に遠くまで質量を運んでしまうことがあり得るからである。

38.7.3 最適輸送と直線性

38.7.3.1 カップリングとワッサースタイン距離

Reflow は輸送コストを下げた。コストはどこまで下げられるのか、そして最も低コストなペアリングのフローは真っ直ぐなのだろうか。37.4.2 節の用語を改めて確認しておこう。**カップリング** π（p_0 と p_1 のカップリング）とは、これらを周辺分布に持つペアの同時分布であり、各始点から各終点へどれだけの質量を運ぶかを定める輸送計画である。**2 次ワッサースタイン距離**（2-Wasserstein distance）は、二乗距離のもとで最も低コストな計画のコストを表す次の量である。

$$W_2^2(p_0, p_1) = \min_{\pi \in \Pi(p_0, p_1)} \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_1) \sim \pi} \left[\|\mathbf{x}_1 - \mathbf{x}_0\|^2 \right], \tag{38.102}$$

ここで $\Pi(p_0, p_1)$ はカップリング全体の集合である。すなわち W_2^2 は、一方の分布を他方の分布へ変えるのに必要な最小の平均二乗変位である。

最も低コストな計画はどのような姿をしているのだろうか。2 つの分布の一方が密度を持ち、両方が有限の二次モーメントを持つとき、その計画は一意に定まり、質量を分割しない。すなわち、密度を持つ側の分布からの写像となる。ここでの設定ではその分布はノイズ p_1 であり、ノイズからデータへの写像は凸関数の勾配、すなわち **Brenier 写像**（Brenier map）となる。これが Brenier の定理 (Brenier 1991) であり、本書では証明なしに引用する。データ側は密度を持つ必要はない。2 点の例も密度を持たないし、先ほどの 8,192 点の two-moons のような有限の訓練セットも密度を持たない。直線（1 次元）上では、この定理は視覚的に理解しやすい。1 変数凸関数の導関数は単調非減少であるため、最適写像は点の順序を保つのである。

最小の例は 1 次元のガウス分布であり、データを $\mathcal{N}(m, s^2)$（$s > 0$）、ノイズを $\mathcal{N}(0,1)$ とする。ソート済みペアリングは各分位点（quantile）を同じ分位点へと送り（$x_0 = m + s\epsilon$）、コストは $\mathbb{E}(m + s\epsilon - \epsilon)^2 = m^2 + (s-1)^2$ となる。一方、独立ペアリングのコストは $\mathbb{E}(\epsilon - x_0)^2 = m^2 + 1 + s^2$ である。1 次元では任意の凸コストに対してソート済みペアリングが最適となる。2 組のペアが交差しているとき、互いの行き先を入れ替えてもコストは増えないからである（演習問題 2d、37.4 節）。したがって次式が成り立つ。

$$W_2^2\big(\mathcal{N}(0,1), \mathcal{N}(m, s^2)\big) = m^2 + (s-1)^2. \tag{38.103}$$

この距離は、平均を平行移動させるコストと幅を伸縮させるコストとに分離される。$\mathcal{N}(0,4)$ と $\mathcal{N}(0,1)$ の間ではその値は 1 であるのに対し、独立ペアでは 5 となる（Figure 38.29 のパネル (a) と (b)）。独立ペアリングは、コストの 5 分の 4 を他の質量とすれ違わせる無駄な移動に費やしている。

これにより、Figure 38.28 の軌跡がどこに行き着いたかが説明できる。1 次元ではフローの軌跡は交差できないためフローは順序を保ち、順序を保つペアリングこそが最適なペアリングとなる。したがって直線上では、1 ラウンドの Reflow で最適輸送に到達する（演習問題 4）。

2 次元では、交差がないというだけでは不十分である。標準ガウス分布の雲を原点の周りに単位時間で 4 分の 1 回転させる場、すなわち点 (ξ_1, ξ_2) に速度 $\frac{\pi}{2}(-\xi_2, \xi_1)$ を割り当てる速度場を考えよう。どの 2 本の軌跡も交差せず、雲は出発した場所と同じ分布に行き着くため $W_2^2(p_0, p_1) = 0$ である。それにもかかわらず、すべての粒子が速さ $\frac{\pi}{2}\|\mathbf{x}\|$ で動く。これはいわばメリーゴーランドであり、全員が移動しているのに群衆全体としてはどこへも進まないのである（Figure 38.29 のパネル (c)）。

円弧を直線に伸ばしてもこれは解決しない。各 $\mathbf{x}_0$ をその 4 分の 1 回転 $\mathbf{R}\mathbf{x}_0$ とペアにし、両者を線分で結んでみよう。線分 $\big((1-t)\mathbf{I} + t\mathbf{R}\big)\mathbf{x}_0$ は決して交差しない（この行列が任意の t で正則であるため）うえ、各線分は一定速度でたどられる。したがって $C = 0$ であり Reflow はこのペアリングを変化させないが、それでもコストは $\mathbb{E}\|\mathbf{R}\mathbf{x}_0 - \mathbf{x}_0\|^2 = 4$ かかり、$W_2^2 = 0$ とは一致しない。直線上では運動を無駄にする原因は交差だけだが、平面上ではペアリングが直線かつ非交差であっても最適でないことが起こり得る。このペアリングに何が欠けているかは Brenier の定理が教えてくれる。最適写像は凸関数の勾配であり、4 分の 1 回転はいかなる関数の勾配でもない。勾配の導関数は対称行列（symmetric matrix）になるが、$\mathbf{R}$ は対称ではないからである。

ペアリングが記録するのは各粒子が最終的にどこへ着くかだけである。途中の運動にはどのようにコストを課すべきだろうか。1 個の粒子が $\mathbf{x}_1$ から $\mathbf{x}_0$ まで単位時間で何らかの経路 $\mathbf{x}_t$ に沿って移動するとしよう。時間平均に対するイェンセンの不等式より、次が成り立つ。

$$\int_0^1 \|\dot{\mathbf{x}}_t\|^2\, dt \;\geq\; \Big\| \int_0^1 \dot{\mathbf{x}}_t\, dt \Big\|^2 = \|\mathbf{x}_1 - \mathbf{x}_0\|^2,$$

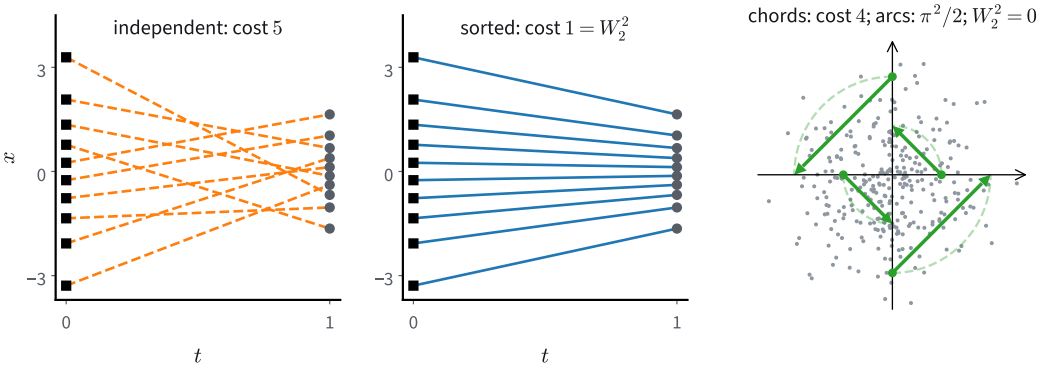


Figure 38.29: ガウス分布を動かす 3 つの方法。(a) データ $\mathcal{N}(0, 4)$（$t = 0$）とノイズ $\mathcal{N}(0, 1)$（$t = 1$）の独立ペアリングを、等間隔な 10 個の分位点水準（$0.05, 0.15, \dots, 0.95$）について描いたもの。線分は交差し、平均二乗変位は 5 となる。(b) ソート済みペアリング $x_0 = 2\epsilon$ は決して交差せず、コストは二乗ワッサースタイン距離である 1 となる。(c) 平面内の標準ガウス分布の雲を 4 分の 1 回転させたもの。回転の円弧（破線）の運動エネルギーは $\pi^2/2$ である。各点をその 4 分の 1 回転先と結ぶ弦（矢印）は直線で決して交差せず、コストは 4 である。雲は出発した場所に戻るため $W_2^2 = 0$ であり、2 次元では直線かつ非交差なペアリングであっても最適とは限らない。

等号が成り立つのは速度が一定の場合に限られる。速さの二乗の積分は等速直線運動のときに最小となり、あらゆる迂回や速度変化に対してコストを課す。これを粒子全体にわたって平均したものがフローの**運動エネルギー**（kinetic energy）$\int_0^1 \mathbb{E}\|\mathbf{u}_t(\mathbf{x}_t)\|^2\, dt$ である。先ほどのメリーゴーランドの運動エネルギーは $(\pi/2)^2\, \mathbb{E}\|\mathbf{x}\|^2 = \pi^2/2 \approx 4.9$ であり、そのすべてが無駄に費やされている。

運動エネルギーは、同じ端点を持つフローどうしの順序付けにも使える。データ $\mathcal{N}(0, 4)$ とノイズ $\mathcal{N}(0, 1)$ の間の 3 つのフロー、すなわちソート済みペアリングの直線フロー、独立ペアリングのもとでの直線分の周辺フロー、および 38.7.1.3 節のコサイン経路を比べよう。どれが最も低コストであり、直線分から組み立てたフローはそうでないフローに勝るのだろうか。3 つともガウス周辺分布 $\mathcal{N}(0, s_t^2)$ を持つため、速度は $(\dot{s}_t/s_t)\, x$、運動エネルギーは $\int_0^1 \dot{s}_t^2\, dt$ となる（演習問題 3）。次のコードセルではまずサンプルをソートすることで、2 つのペアリングコスト 5 と 1 を確かめる。

```
n = 200_000
x0, eps = 2 * rng.standard_normal(n), rng.standard_normal(n)
print(f'data N(0,4), noise N(0,1): independent pairing {np.mean((eps - x0)**2):.2f}, '
      f'sorted pairing {np.mean((np.sort(eps) - np.sort(x0))**2):.3f}')

def kinetic_energy(s, t):                    # marginals N(0, s_t^2): u_t(x) = (s'/s) x
    return np.sum(np.diff(s)**2 / np.diff(t))  # the integral of (ds/dt)^2

t = np.linspace(0, 1, 100_001)
for name, s in [('sorted, straight', 2 - t),
                ('independent, straight', np.sqrt(4 * (1 - t)**2 + t**2)),
                ('cosine path', np.sqrt(1 + 3 * np.cos(np.pi * t / 2)**2))]:
    print(f'{name:22s} kinetic energy {kinetic_energy(s, t):.3f}')
```

```
data N(0,4), noise N(0,1): independent pairing 4.99, sorted pairing 0.985
```

```
sorted, straight       kinetic energy 1.000
independent, straight  kinetic energy 1.858
cosine path            kinetic energy 1.234
```

ソート済みの直線フローが最も低コストであり、その運動エネルギーは $W_2^2 = 1$ となる。驚くべきは残り 2 つの順序である。独立ペアリングの直線分から生じる周辺フローのコストは 1.86 であり、コサイン経路の 1.23 よりも大きい。その分散 $s_t^2 = 4(1-t)^2 + t^2$ は、t が 1 から 0.8 へ下がる間にいったん縮小してから増大に転じるため、粒子は一度内側へ動いてから再び外側へ押し出される。誤ったペアリングのもとで条件付き線分を直線にしても、直線フローにはならないのである。

38.7.3.2 Benamou–Brenier の定理

3 つのフローのうち最も低コストなもののコストは、ちょうど最も低コストなペアリングのコスト W_2^2 に一致した。これはガウス分布に特有の偶然ではない。2 つの分布間の最も低コストなフローは、常に最も低コストなペアリングと同じコストを持ち、しかもそのペアリングの各ペアを一定速度で線分に沿って運ぶフローとなる。これが本節における最も深い事実である。端点をペアにする静的な問題 Equation 38.102 と、フローマッチングが訓練する対象であるフローとが、これによって結びつく。

定理（Benamou–Brenier）。 **p_0 と p_1 が有限の二次モーメントを持ち、その一方（ここではノイズ p_1）が密度を持つとする。組 $(p_t, \mathbf{u}_t)$ について、$\mathbf{u}_t$ が連続微分可能かつ $\mathbf{x}$ に関して（t に依らない定数で）リプシッツ連続であり、この組が端点 p_0 と p_1 を持つ連続の式を満たすとき、この組を許容的（admissible）と呼ぶ。(i) すべての許容的な組は少なくとも $W_2^2(p_0, p_1)$ の運動エネルギーを持つ。(ii) 最適カップリングのペアを直線分で結ぶ。この経路の周辺速度が許容的であれば、その運動エネルギーは $W_2^2(p_0, p_1)$ に等しく、したがって許容的な組全体にわたる最小値として次式が成り立つ** (Benamou and Brenier 2000)。

$$W_2^2(p_0, p_1) = \min_{(p_t, \mathbf{u}_t)} \int_0^1 \int \|\mathbf{u}_t(\mathbf{x})\|^2 \, p_t(\mathbf{x}) \, d\mathbf{x} \, dt. \tag{38.104}$$

(ii) の条件は、たとえば Brenier 写像が滑らかな微分同相写像であり、その導関数と逆写像の導関数がともに有界であるときに満たされる。この条件がない場合でも、二乗可積分な速度を持つ連続の式の弱解全体にわたって最小値をとれば同じ等式が成り立つ。この一般形については証明を省くが、ここでの例にはそれが必要となる。2 点の例における最適速度場は傾き $1/t$ を持ち、上の意味では許容的ではないからである。

証明。 最小運動エネルギーを W_2^2 によって下と上から評価する。ステップ II が下界 (i) であり、ステップ III が (ii) を与える。

I. すべてのフローはカップリングを定める。 $\mathbf{Z}_t$ を、$\dot{\mathbf{Z}}_t = \mathbf{u}_t(\mathbf{Z}_t)$ を時間を逆向きに $\mathbf{Z}_1 \sim p_1$ から解いた解とする。Reflow の命題で用いた一意性により、$\mathbf{Z}_t \sim p_t$ が任意の t に対して成り立つため、$(\mathbf{Z}_0, \mathbf{Z}_1)$ は p_0 と p_1 のカップリングである。

II. 下界。 このカップリングのコストは少なくとも W_2^2 であり、時間平均に対するイェンセンの不等式によって運動エネルギーで上から抑えられる。

$$\begin{aligned} W_2^2(p_0, p_1) &\le \mathbb{E}\,\|\mathbf{Z}_1 - \mathbf{Z}_0\|^2 = \mathbb{E}\left\| \int_0^1 \mathbf{u}_t(\mathbf{Z}_t)\, dt \right\|^2 \\ &\le \int_0^1 \mathbb{E}\,\|\mathbf{u}_t(\mathbf{Z}_t)\|^2 \, dt = \int_0^1 \int \|\mathbf{u}_t\|^2 \, p_t \, d\mathbf{x}\, dt. \end{aligned}$$

III. **上界**。最適カップリング $\pi^\star$ とその線形補間 $\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{X}_1$ をとり、周辺速度を $\mathbf{u}_t(\mathbf{x}) = \mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t = \mathbf{x}]$ とする。弱形式の周辺速度命題により、この組は正しい端点を持つ連続の式を満たし、条件付きイェンセンの不等式によりその運動エネルギーは次のようになる。

$$\int_0^1 \mathbb{E}\big\|\mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_t]\big\|^2 \, dt \le \mathbb{E}\|\mathbf{X}_1 - \mathbf{X}_0\|^2 = W_2^2(p_0, p_1).$$

上下の評価が一致した。■

この定理はモデルについて何を語っているだろうか。最適なペアリングの上で Rectified Flow を訓練すれば、冒頭の願いが叶う。すなわち損失の下限は 0 となり、厳密な速度場は各軌跡に沿って一定となり、長さ 1 のオイラーステップ 1 回で誤差なく積分できる。これら 2 つの性質はいずれも、Reflow の命題にも登場した証明中の 2 つの不等式における等号成立条件にあたる。最適カップリングのフローが許容的であれば下界を達成するため、III の不等式は等号となり、条件付き分散は 0 になる。つまり最適カップリングの線分は交差しない。また II のイェンセンの不等式が等号となるのは各粒子が一定速度で動く場合に限られるため、あらゆる最小点は粒子を一定速度で直線上を動かす。その逆は成り立たない。4 分の 1 回転の例が示したとおりである。フローが真っ直ぐで交差がなく、1 ステップで厳密であっても最も低コストであるとは限らず、W_2^2 を達成するのはそのペアリング自体も最適である場合に限られる（演習問題 7）。

結局のところ目標は最適カップリングであり、実務ではそれに対する 2 つの近似手法が用いられる。1 つはモデルを使って再ペアリングする Reflow である。もう 1 つの**ミニバッチ最適輸送**（minibatch optimal-transport）ペアリング (Tong et al. 2024; Pooladian et al. 2023) は訓練済みモデルを必要とせず、各訓練ステップの前にバッチ内のノイズ点とデータ点を二乗距離の総和が最小となる割り当てによって組み直す。演習問題 5 ではこのペアリングで two-moons を訓練し、損失のプラトー、1 ステップのスコア、およびペアの長さへの効果を測定する。いずれの手法も 2 つの分布間の真の最適カップリングを見つけるわけではない。バッチごとの計画はバッチの選び方に依存する。また Reflow は、4 分の 1 回転の例が示したように最適であるか否かにかかわらず交差のない任意のペアリングで停止するうえ、最初のモデルの誤差も受け継ぐ。

38.7.4 学習された速度場によるサンプリング

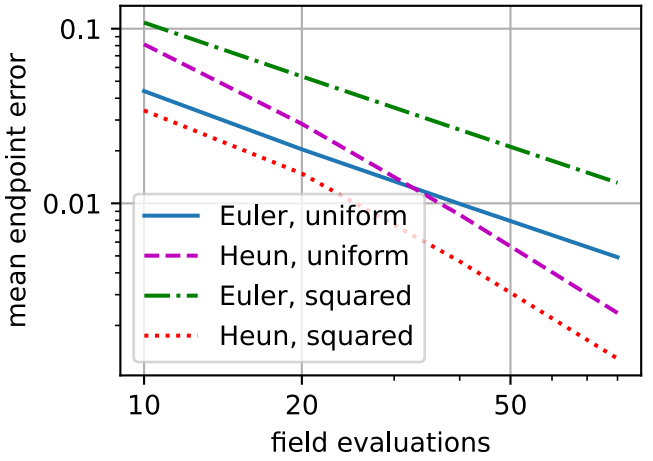
フローを真っ直ぐにすることは、ネットワーク呼び出しを節約する 1 つの方法である。もう 1 つの方法は、高次のソルバを用いたり経路が大きく動く場所にステップを配置したりして、呼び出しをより有効に使うことである。この工夫は学習された速度だけでなく学習されたスコアにも使える。38.6.3 節の確率フロー ODE はスコアを速度場へと変換し、とりわけ分散保持経路上ではフローマッチングが学習するのとまさに同じ場へと変換するからである。サンプルの生成にはソルバの 1 段あたり 1 回の速度場評価というコストがかかり、問題は限られた評価予算をどう配分するかである。ここでは学習誤差のない厳密な速度場の上で 1 つ実験を行い、現れる誤差がすべてソルバに由来するようにして調べる。

ホイン法（Heun's method）は 2 次精度だが、1 ステップあたり速度場を 2 回評価する。2 回分の評価を費やす価値はあるのだろうか。本章で繰り返し用いている混合分布（± 2 に分散 0.25 の成分を持つ分布）の**厳密な**スコアを用いた確率フロー ODE を、$\beta(t)$ が 0.1 から 20 まで線形に増加する分散保持スケジュールのもとで考える。次のコードセルでは、オイラー法とホイン法（38.2 節）を用いて $t=1$ から $t=0$ まで積分し、8,000 個の出発点の到達端点を 800 ステップのホイン法の解と比較する。両手法には 10〜80 回の速度場評価（出力される表の NFE の行、すなわち関数評価回数「number of function evaluations」）という同一の予算を与え、2 種類の時間グリッド上で比較する。1 つは t に関する等間隔ステップであり、もう 1 つは二**乗**グリッド $t_k = (1 - k/K)^2$（$k = 0, \dots, K$、データ側の端点 $t=0$ に向かってステップ幅が縮むグリッ

ド）である。ホイン法の誤差はステップ幅の二乗に比例して減り、オイラー法の誤差はステップ幅に比例して減るため、評価回数が十分多ければ必ずホイン法が勝つ。では 10 回で十分だろうか。

```
bmin, bmax = 0.1, 20.0
beta_fn = lambda t: bmin + t * (bmax - bmin)            # VP noise schedule
abar_fn = lambda t: np.exp(-(bmin * t + 0.5 * (bmax - bmin) * t**2))
def score_vp(q, t, means=(-2.0, 2.0), var=0.25):        # exact mixture score
    a = abar_fn(t)
    v = a * var + (1 - a)
    w = np.stack([np.exp(-(q - np.sqrt(a) * m)**2 / (2 * v)) for m in means])
    return (w * np.stack([(np.sqrt(a) * m - q) / v
                          for m in means])).sum(0) / w.sum(0)
def pf_ode(q, t):                                       # probability-flow ODE
    return -0.5 * beta_fn(t) * (q + score_vp(q, t))
def solve(q, ts, heun=False):                           # steps along grid ts
    for t1, t2 in zip(ts[:-1], ts[1:]):
        h, d1 = t2 - t1, pf_ode(q, t1)
        q = q + 0.5 * h * (d1 + pf_ode(q + h * d1, t2)) if heun else q + h * d1
    return q
grids = {'uniform': lambda K: np.linspace(1, 0, K + 1),     # equal steps in t
         'squared': lambda K: np.linspace(1, 0, K + 1)**2}  # dense near t = 0
z = rng.standard_normal(8000)
ref = solve(z, grids['uniform'](800), heun=True)        # fine reference
nfes = [10, 20, 40, 80]                                 # Heun: n // 2 steps
errs = {f'{m}, {g}': [np.abs(solve(z, grid(n // k), k == 2) - ref).mean()
                      for n in nfes]
        for g, grid in grids.items() for m, k in (('Euler', 1), ('Heun', 2))}
print(f'{"NFE":16s}' + ''.join(f'{n:8d}' for n in nfes))
for name, e in errs.items():
    print(f'{name:16s}' + ''.join(f'{v:8.4f}' for v in e))
d2l.plot(nfes, list(errs.values()), 'field evaluations', 'mean endpoint error',
         legend=list(errs), xscale='log', yscale='log')
```

```
NFE                10      20      40      80
Euler, uniform   0.0439  0.0204  0.0099  0.0049
Heun, uniform    0.0814  0.0285  0.0086  0.0024
Euler, squared   0.1084  0.0533  0.0264  0.0131
Heun, squared    0.0340  0.0148  0.0047  0.0013
```



一様グリッド上では、各手法の次数が減少率にそのまま現れる。予算を 2 倍にするごとに、オイラー法の誤差は約 2 分の 1 になり、ホイン法の誤差は約 3 分の 1 から 4 分の 1 へと近づく倍率で縮む。ところが同じコストで比べると、ホイン法は評価回数 10（約 0.08 に対しオイラー法は 0.04）および 20（約 0.03 に対し 0.02）で敗れ、40 以上でようやく優位に立つ。一方、二乗グリッド上ではすべての予算でホイン法が勝ち、評価回数 10 での誤差（約 0.03）はいずれのグリッド上のオイラー法よりも小さい。オイラー法は逆の動きを示し、二乗グリッドにすると誤差が 2 倍以上に悪化する。この速度場では、予算が小さいときどちらのソルバが勝つかを決めるのはグリッドなのである。

なぜ 1 次法が 2 次法に勝てるのだろうか。終端誤差は符号付きのステップ誤差の和であり、符号付きの誤差どうしは打ち消し合うことがある。次のコードセルではその度合いを測定する。厳密な経路上の点から出発した k 番目の粗いステップの誤差をとり、その時刻での変位に対する終端の感度（厳密なフローの近傍軌跡から読み取った値）を掛ける。その積 c_k が、終端誤差に対するそのステップの 1 次近似での寄与である。コードセルは 2 つの表を表示する。和をとったあとに残る割合 $|\sum_k c_k| / \sum_k |c_k|$ と、打ち消し合いが起こる前の総量 $\sum_k |c_k|$ を、それぞれ出発点にわたって平均したものである。最後の行には、各経路の長さのうち $t = 1$ から $t = 0.5$ の間に含まれる割合を表示する。

```
def exact_path(q, ts):                  # fine Heun path, and the gain dx(0)/dx(t)
    qs = [np.stack([q, q + 1e-6])]      # the path and a close neighbor
    for t1, t2 in zip(ts[:-1], ts[1:]):
        qs.append(solve(qs[-1], [t1, t2], heun=True))
    qs = np.array(qs)
    return qs[:, 0], (qs[-1, 1] - qs[-1, 0]) / (qs[:, 1] - qs[:, 0])

def step_errors(ts, qs, gain, steps, heun):   # c_k, shape (steps, starting points)
    j = np.arange(0, len(ts), (len(ts) - 1) // steps)    # coarse nodes
    return np.array([gain[b] * (solve(qs[a], ts[[a, b]], heun) - qs[b])
                     for a, b in zip(j[:-1], j[1:])])   # step error, carried to 0

paths = {g: exact_path(z, grid(400)) for g, grid in grids.items()}   # 400 fine steps
cs = {f'{m}, {g}': [step_errors(grids[g](400), qs, gain, n // k, k == 2) for n in nfes]
      for g, (qs, gain) in paths.items() for m, k in (('Euler', 1), ('Heun', 2))}
```

```
    stats = {'surviving share': (lambda c: np.abs(c.sum(0)).mean() / np.abs(c).sum(0).mean(),
    ↪  '.2f'),
             'total of |c_k|': (lambda c: np.abs(c).sum(0).mean(), '.4f')}   # sum_k |c_k|
    for title, (stat, fmt) in stats.items():
        print(f'{title:16s}' + ''.join(f'{n:8d}' for n in nfes))
        for name, c in cs.items():
            print(f'{name:16s}' + ''.join(f'{stat(c_n):8{fmt}}' for c_n in c))
    d = np.abs(np.diff(paths['uniform'][0], axis=0))    # |fine step| along each path
    share = (d[:200].sum(0) / d.sum(0)).mean()           # first 200 steps: t from 1 to 0.5
    print(f'share of the path length covered by t = 0.5: {share:.2f}')
```

```
surviving share       10      20      40      80
Euler, uniform      0.11    0.11    0.12    0.12
Heun, uniform       0.59    0.66    0.71    0.73
Euler, squared      0.31    0.32    0.34    0.35
Heun, squared       0.25    0.36    0.41    0.42
total of |c_k|        10      20      40      80
Euler, uniform    0.2867  0.1502  0.0769  0.0389
Heun, uniform     0.1317  0.0414  0.0119  0.0032
Euler, squared    0.2472  0.1327  0.0691  0.0353
Heun, squared     0.1258  0.0397  0.0113  0.0030
share of the path length covered by t = 0.5: 0.09
```

$t = 0.5$ までに経路が進む距離は全長の 10 分の 1 未満である。その時間帯の大半ではノイズを加えた分布が $\mathcal{N}(0, 1)$ に近く、そのスコアがドリフトを打ち消すからである。t に関する等間隔ステップはその 10 分の 1 の移動に予算の半分を費やしてしまうため、そもそもステップを偏らせたグリッドを試す動機がそこにある。

残存割合を見ればオイラー法が優位に立った理由がわかる。等間隔ステップでは、どの予算でもオイラー法のステップ誤差のうち終端まで残るのは約 8 分の 1 にすぎないのに対し、ホイン法では半分以上が残る。打ち消し合う前のステップ誤差自体はオイラー法の方が大きく、2 つ目の表の総量はどの予算でもホイン法の 2 倍を超えている。もしホイン法と同じ割合だけ誤差が残っていたなら、オイラー法はすべてで敗れていたはずである。オイラー法の優位は誤差の相殺によるものであり、それは手法自体の性質ではなく、このグリッド上でのこの速度場に固有の性質にすぎない。二乗グリッドはこの相殺を奪い去り、オイラー法のステップ誤差は約 3 分の 1 が残るようになる一方、ホイン法では 4 分の 1〜5 分の 2 となる。

驚くべきことに、二乗グリッドが変えたのはほぼこの点だけである。絶対誤差の総量はグリッドの違いをほとんど受けない。ホイン法の評価回数 10 における総量は等間隔ステップで 0.13、二乗グリッドで 0.125 であり、どの予算でも両者の差は数パーセント以内に収まる。オイラー法の総量に至っては二乗グリッド上の方がむしろ小さく、評価回数 10 で約 7 分の 1、80 で約 10 分の 1 減っているにもかかわらず、終端誤差は 2 倍以上に増えている。つまりステップをデータ側に密集させても、ステップ誤差の総量はほんのわずかしか減らなかったのである。変わったのはどれだけ誤差が打ち消し合うかであり、それがホイン法には有利に、オイラー法には不利に働いた。二乗グリッドは見かけよりも粗く、時刻 t におけるステップ幅は等間隔ステップの約 $2\sqrt{t}$ 倍であるため、$t = 1/4$ 未満では短くなるがそれより上では長くなる。

画像モデルのサンプラも同様にデータ側へステップを密集させており、その度合いは実測によって調整される (Karras et al. 2022)（詳しくは 17.5 節で述べる）。なぜ実測せざるを得ないのかは上の表が物語っている。ある 1 つのグリッド上でのソルバの比較は、そのグリッドについての主張でしかないのである。

2 つの問いが未解決のまま残っている。高次元において最適なペアリングを低コストに見つける

方法は知られておらず、Reflow やバッチ内の再ペアリングはそれに近づくことしかできない。また、本章の冒頭で掲げた「モデルはどこまで少ないステップ数で済ませられるか」という問いも、依然として新しい速度場ごとに試行錯誤で答えが出されているのが実情であり、ステップ数とその配置は導出されるのではなく調整される。第 17 章では、画像を対象にこれら両方の問いを取り上げる。

38.7.5 まとめ

ノイズからデータへの経路は自由に選ぶことができる。その経路を実現する速度はサンプルごとの経路の速度の事後平均であるため、条件付きフローマッチングは最小二乗法によってそれを訓練し、経路どうしの食い違いの大きさを測る損失の下限 C を伴う。本節と前節のすべてのモデルはこの方式で訓練される。すなわち、閉じた形で既知であるサンプルごとの量の条件付き平均に適合させ、ソルバを用いてサンプリングする。ガウス経路上ではスコア、ノイズ、速度がすべて同一の事後平均のアフィン関数となるため、1 つのネットワークをそのいずれとしても解釈できる。条件付き経路を直線にしても線分が交差する場所ではフローが曲がるが、Reflow はノイズをフロー自身の到達端点と組み直すことで輸送コストを増やさずに交差を減らし、two-moons ではオイラーステップ 1 回で 32 ステップに匹敵する結果をもたらす。輸送コストは両端を結ぶあらゆるフローの最小運動エネルギーである二乗ワッサースタイン距離を下回ることはできず、等速直線運動がこの最小値を達成するのは最適なペアリングのもとでのみである。速度場が必要とするステップ数はステップの配置場所にも依存する。グリッドがステップ誤差の相殺量を左右するからである。本章の混合分布の厳密な速度場では、ホイン法は一様グリッド上では評価回数約 40 未満でオイラー法に敗れるが、データ側にステップを密集させたグリッド上ではすべての予算で勝利する。

38.7.6 演習問題

1. **確率的補間。**●Albergo et al. (2023) は、本節と前節の経路を $\mathbf{x}_t = a_t\mathbf{x}_0 + b_t\mathbf{x}_1 + c_t\mathbf{w}$ と表している。ここで $(\mathbf{x}_0, \mathbf{x}_1)$ は p_0 と p_1 のカップリングから引かれ、$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ はその両方と独立であり、滑らかなスケジュールは $a_0 = b_1 = 1$ および $a_1 = b_0 = c_0 = c_1 = 0$ を満たす。ガウス分布の p_1 に対しては、拡散経路や Rectified Flow の経路は片側の場合 $c \equiv 0$ にあたり、潜在項 $c_t\mathbf{w}$ は任意の p_0 と p_1 を結ぶ両側経路の役割を果たす。
   1. $\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{w})$ を周辺速度 Equation 38.94 にとれ。それが $\mathbb{E}[\dot a_t\mathbf{x}_0 + \dot b_t\mathbf{x}_1 + \dot c_t\mathbf{w} \mid \mathbf{x}_t = \mathbf{x}]$ となることを示し、対応する CFM 損失 Equation 38.95 を書き下せ。
   2. Rectified Flow Equation 38.99 と、データ係数およびノイズ係数が $\cos\frac{\pi t}{2}$ と $\sin\frac{\pi t}{2}$ であるコサイン経路（いずれも独立な端点かつ $p_1 = \mathcal{N}(\mathbf{0}, \mathbf{I})$ とする）を復元するスケジュールを挙げよ。
   3. $p_1 = \mathcal{N}(\mathbf{0}, \mathbf{I})$、独立な端点、および $c_t > 0$ のもとで、この補間式が $c \equiv 0$ かつ $\sqrt{b_t^2 + c_t^2}$ で b_t を置き換えた補間式と同じ周辺速度を持つことを示せ。
   4. $c_t > 0$ のとき、$\nabla \log p_t(\mathbf{x}) = -\mathbb{E}[\mathbf{w} \mid \mathbf{x}_t = \mathbf{x}]/c_t$ を示せ。p_1 がガウス分布ではなく任意の分布であるとき、途中の時刻で加わるノイズは何をもたらすか。
2. **遅い意思決定。**データ $\pm a$ を持つ 2 点の例を考える。
   1. 点 $+a$ に重み w、点 $-a$ に重み $1-w$ を与える。Equation 38.98 の導出を繰り返し、事後平均が $a\tanh\left((1-t)\,ax/t^2 + \frac{1}{2}\log\frac{w}{1-w}\right)$ となることを示せ。このとき $t = 1$ から $t = 0$ への 1 回のオイラーステップはどこに着地するか。
   2. 重みが等しいとき、$\partial_x u_t(0) = \left(1 - (1-t)\,a^2/t^2\right)/t$ となること、および t が減少するにつれて原点が $t^* = \frac{1}{2}\left(\sqrt{a^4 + 4a^2} - a^2\right)$（値は $2\sqrt{2} - 2 \approx 0.83$、ただし $a = 2$ のとき）で吸引から反発へと切り替わることを示せ。t^* は $a \to \infty$ および $a \to 0$ のときどのように振る舞うか。両方の極限を解釈せよ。
   3. 原点から離れた場所では傾き $\partial_x u_t$ が $1/t$ に漸近すること、および $t = h$ から $t = 0$

へのオイラーステップが厳密に $\pm a$ へ（tanh が飽和している任意の点から出発したとき）着地することを示せ。それでは、固定ステップ幅ソルバの律速となるのはフローのどの部分であり、どの時間帯か。

4. 損失の下限は $C = \mathbb{E}_t \mathbb{E}_x \mathrm{Var}(x_0 \mid x_t)/t^2$ であり、ここで $\mathrm{Var}(x_0 \mid x_t = x) = a^2 \operatorname{sech}^2\left((1-t)\,ax/t^2\right)$ である。$C \le a^2 + 1$ を示せ。さらに、$t = 0$ 付近の時間帯が C にほとんど寄与しない理由を（そこでは条件付き速度 $(x - x_0)/t$ が非有界であるにもかかわらず）説明せよ。ただし x_t の 0 付近における密度のオーダーが $e^{-(1-t)^2 a^2/2t^2}/t$ である事実を用いよ。

3. **ガウス経路の運動エネルギー**。
   1. 1 次元において $p_t = \mathcal{N}(0, s_t^2)$ であるとき、速度 $u_t(x) = (\dot{s}_t/s_t)\,x$ が連続の式を満たし、その運動エネルギーが $\int_0^1 \dot{s}_t^2\,dt$ となることを示せ。ガウス分布の雲に他の速度が不要なのはなぜか。
   2. データ $\mathcal{N}(0, 4)$ とノイズ $\mathcal{N}(0, 1)$ に対し、本文のコードセルで扱った 3 つのフロー（ソート済みペアリングの直線フロー、独立ペアリングのもとでの直線分の周辺フロー、およびコサイン経路）について s_t を書き下せ。最初の 2 つの運動エネルギーが厳密に 1 と $5 - \pi$ になることを示せ。
   3. データ $\mathcal{N}(0, 1)$ に対して独立ペアリングを同様に計算せよ。その周辺フローが運動エネルギー $2 - \pi/2 \approx 0.43$ を持つ（$W_2^2 = 0$ であるにもかかわらず）ことを示せ。コストが 0 になるのはどのペアリングであり、そのフローはどのようなものか。
4. [code] **2 点に対する 1 ラウンドの Reflow**。データ $\pm a$、ノイズ $\mathcal{N}(0, 1)$、独立ペアリング、および 2 点の例の周辺フロー Equation 38.98 を考える。
   1. 38.3.1.2 節の順序保存の命題と対称性を用いて、$t = 1$ から $t = 0$ へのフロー写像が $\epsilon \mapsto a\,\mathrm{sign}(\epsilon)$ であり、したがって 1 ラウンドの Reflow がソート済みペアリングを生成することを示せ。
   2. そのコストが $a^2 - 2a\sqrt{2/\pi} + 1$（約 1.81、ただし $a = 2$ のとき。これに対し独立ペアリングでは $a^2 + 1 = 5$）であることを示し、なぜこのコストが W_2^2 に等しいのかを説明せよ。
   3. Reflow 後の線分 $(1-t)\,a\,\mathrm{sign}(\epsilon) + t\,\epsilon$ が決して交差せず、したがって 2 ラウンド目の下限が $C = 0$ となることを示し、その速度場を書き下せ。その速度場に対する 1 回のオイラーステップは厳密か。
   4. 閉じた形の速度場を、20,000 個のガウスサンプルから $t = 1$ より $t = 10^{-3}$ まで RK4 で積分し（速度場の傾きが $1/t$ のオーダーで増大するため 0 の手前で止める）、到達端点とコストを確認せよ。
   5. 独立ペアリングの下限は $C \approx 2.03$（$a = 2$ のとき）である。演習問題 2 の (d) の式から数値積分によってこれを計算し、その 99% が集まる境目となる時刻を求めよ。Reflow の命題の上界 $\mathbb{E}\|\mathbf{Z}_1 - \mathbf{Z}_0\|^2 \le \mathbb{E}\|\mathbf{X}_1 - \mathbf{X}_0\|^2 - C$ を確認せよ。ここではどれだけの余裕（スラック）が残っているか。
5. [code] **ミニバッチ最適輸送ペアリング**。`independent_pairs` を、256 個のノイズ点と 256 個のデータ点からなるブロック単位でバッチを引き、各ブロック内を二乗距離の総和が最小となる割り当て（`scipy.optimize.linear_sum_assignment` を 256×256 のコスト行列に適用）によって組み直すサンプラに置き換え、`train_rf` で訓練せよ。
   1. 損失のプラトー、1、2、8、32 ステップにおける二乗エネルギー距離、および再ペアリング前後のペアの平均二乗長さを報告せよ。独立ペアリングおよび Reflow と比較せよ。
   2. 再ペアリングされたノイズ点が依然として標準ガウス分布からのサンプルであり、再ペアリングされたデータ点も依然としてデータからのサンプルであるため、周辺速度場が引き続き p_1 を p_0 へ運ぶことを示せ。
   3. バッチ内のペアリングが 2 つの分布の最適カップリングにはならない理由と、ブロックサイズを大きくするとどうなるかを説明せよ。2 つの再ペアリング手法のうち、訓

練済みモデルを必要とするのはどちらであり、その誤差を受け継ぐのはどちらか。

6. **予測ターゲットの条件性。**ガウス経路 Equation 38.96 において、1 次元データ $x_0 \sim \mathcal{N}(0,1)$ とコサイン経路をとる。
   1. t の関数として、Table 38.2 に挙げたサンプリングされた各訓練ターゲットの分散を計算せよ。対象は条件付きスコア $-\epsilon/\sigma_t$、ノイズ ϵ、クリーンな点 x_0、速度 $\dot\alpha_t x_0 + \dot\sigma_t \epsilon$、および $\mathbf{v}$ ターゲット $\alpha_t\epsilon - \sigma_t x_0$ である。発散するのはどれであり、経路のどちらの端で発散するか。
   2. ノイズに対して訓練されたネットワークが、誤差を伴う出力 $\hat\epsilon$（誤差 δ）を返すとする。これから導かれるクリーンな点 $(x - \sigma_t\hat\epsilon)/\alpha_t$ が誤差 $\sigma_t\delta/\alpha_t$ を伴い、逆に誤差 δ を持つクリーンな点の予測から導かれるノイズが誤差 $\alpha_t\delta/\sigma_t$ を伴うことを示せ。それぞれ経路のどちらの端で増幅されるか。
   3. コサイン経路上では速度ターゲットが $\pi/2$ 倍の $\mathbf{v}$ ターゲットに等しく、したがって $\mathbf{v}$ 予測が速度予測になること、さらにノイズ予測から導かれる速度が誤差 $\pi\delta/(2\alpha_t)$ を伴うことを示せ。両端で誤差を有界に保つパラメータ化はどれか。
   4. Table 38.2 の変換式は任意のデータに対して成り立つ。混合分布 $\frac{1}{2}\mathcal{N}(-2, 0.25) + \frac{1}{2}\mathcal{N}(2, 0.25)$ とコサイン経路の $t = 0.4$ において、Equation 38.97 を数値的に確認せよ。点 x のグリッド上で、周辺速度を一度は各成分にその事後確率で重みを付けた $\dot\alpha_t x_0 + \dot\sigma_t\epsilon$ の事後平均として計算し、もう一度はノイズを加えた混合分布のスコアから計算せよ。両者は丸め誤差の範囲で一致するはずである。混合比の重みを 0.8 と 0.2 に変えたとき、計算中のどの数値が変化し、どの数値が変化しないか。
7. **Reflow は周辺分布を保つが停滞し得る。**
   1. 条件付き分布が点質量となる直線経路に対して、Reflow の命題の (i) を詳細に証明せよ。コンパクトな台を持つ滑らかなテスト関数 φ に対し、弱形式の連続の式である $\frac{d}{dt}\mathbb{E}\,\varphi(\mathbf{X}_t) = \mathbb{E}\big[\nabla\varphi(\mathbf{X}_t)\cdot\mathbf{u}_t(\mathbf{X}_t)\big]$ を示せ。条件付き期待値の塔の性質はどこで使われ、リプシッツ条件の仮定はどこで使われるか。
   2. π_k を k ラウンド後のペアリングとする。$\mathbb{E}_{\pi_k}\|\mathbf{x}_1 - \mathbf{x}_0\|^2$ が単調非増加かつ W_2^2 で下から有界であり、したがって収束することを示せ。
   3. 1 ラウンドで何も変化しない（$\pi_{k+1} = \pi_k$）ならば、$C = 0$ が π_k に対して成り立つことを示せ。1 次元における不動点を特定せよ。
   4. 2 次元において、各 $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ を $\mathbf{R}\mathbf{x}_0$（ただし $\mathbf{R}$ は 4 分の 1 回転行列）とペアにする。線分 $\big((1-t)\mathbf{I} + t\mathbf{R}\big)\mathbf{x}_0$ が決して交差せず（この行列は任意の t で正則である。固有値が $(1-t) \pm \mathrm{i}\,t$ となるため）、したがって $C = 0$ となって Reflow が何も変化させない一方、コストが $\mathbb{E}\|\mathbf{R}\mathbf{x}_0 - \mathbf{x}_0\|^2 = 4$（これに対し $W_2^2 = 0$）となることを示せ。どのような行列 $\mathbf{A}$ に対して線形変換 $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ は凸関数の勾配となり、したがって $\mathcal{N}(\mathbf{0}, \mathbf{I})$ とその像との最適ペアリングを与えるか。
8. [code] **フローマッチングモデルに対する確率的サンプラ。●**
   1. 直線経路 $\alpha_t = 1 - t$、$\sigma_t = t$ 上で Equation 38.97 を逆に解き、$\nabla \log p_t(\mathbf{x}) = -\big(\mathbf{x} + (1-t)\,\mathbf{u}_t(\mathbf{x})\big)/t$ を導け。
   2. 任意の $g \geq 0$ に対し、確率微分方程式 $d\mathbf{X} = \big[\mathbf{u}_t(\mathbf{X}) - \frac{1}{2}g^2\nabla\log p_t(\mathbf{X})\big]dt + g\,d\bar{\mathbf{W}}$ を $t = 1$ から $t = 0$ まで（時間を逆向きに走るブラウン運動 $\bar{\mathbf{W}}$ を用いて）積分した解が周辺分布 p_t を持つことを示せ。ただし $\tau = 1 - t$ と置換し、τ に関するフォッカー・プランク方程式を書いて $\nabla\cdot(p\nabla\log p) = \Delta p$ を用いよ。ここでスコア項が負号を伴って現れるのはなぜか。以上により、訓練済みの速度場は確率的サンプラの族を定める。
   3. 訓練済みの two-moons 速度場に対し、$g \in \{0, \frac{1}{2}, 1\}$ および $K \in \{8, 32, 128\}$ ステップでオイラー・丸山法を用いてこのサンプラを実装せよ。スコアは $t = 1, 1 - 1/K, \ldots, 1/K$ のみで評価し（$t = 0$ で特異になるため）、各実行の二乗エネルギー距離を報告せよ。与えられた品質を達成するのに最も少ないステップ数で済むのはどの g であり、$K = 8$ においてノイズは有効か。

4. Reflow 後の速度場に対して同じ実験を繰り返せ。その軌跡の直線性から、1 ステップの確率的サンプルが決して 1 ステップの決定論的サンプルに匹敵し得ない理由を説明せよ。

38.8 関連文献

これまでの 7 つの節では、1 つの点の軌跡から出発してノイズをデータへと変換するフローに至るまでをたどってきたが、その途上では数多くの問いに立ち止まることなく通り過ぎてきた。物理系におけるノイズを記述するのはどの確率積分（stochastic integral）か。モデル自体が SDE であるとき、随伴系はどうなるのか。以下の各項目では、そうした問いを 1 つずつ取り上げ、その答えがどこにあるかを示す。

拡散モデル（diffusion model）の実践は第 17 章とその関連文献（17.10 節）の守備範囲であるため、そこで扱われる手法については参照先を示すにとどめる。残る話題は数学である。まずは次の 3 つの項目から読むことを勧める。拡散サンプラが少数のステップで済む理由の 1 つである指数積分器（exponential integrator）、拡散する密度が何を下っていくのかを述べるワッサースタイン勾配フロー（Wasserstein gradient flow）、そして最適なペアリングの確率版にあたるシュレーディンガーブリッジ（Schrödinger bridge）である。残りの項目は、読者によって抱くかもしれない疑問に答えるものである。38.7 節で読み終えたとしても、本書の残りの部分で用いる内容を何ひとつ見落とすことはない。

38.8.1 ソルバ

38.8.1.1 指数積分器

学習されたスコアを用いた VP 過程の確率フロー ODE（38.6.3 節）である $\dot{\mathbf{x}} = -\frac{1}{2}\beta\,\mathbf{x} - \frac{1}{2}\beta\,\mathbf{s}_\theta(\mathbf{x}, t)$ は、閉じた形で既知の線形項と学習された部分の和からなる半線形（semilinear）方程式である。ルンゲ・クッタ法のステップは両方を離散化するため、線形項も誤差に寄与するうえ、学習された部分がどれほど滑らかであっても 38.2.3 節の安定性限界によってステップ幅が制約される。線形項は厳密に解ける以上、これは無駄である。指数積分器は、学習された部分を駆動線形方程式（38.1.2.3 節）への入力として扱う。すなわち、定数変化法（variation of constants）の公式によって線形項による状態の時間発展を厳密に運び、学習された部分の積分だけを近似するのである。

ステップの間、学習された部分を一定に保てば、ネットワークの出力を入力とする同項のゼロ次ホールドになる。その出力が変化しなければこのステップは厳密になるが、実際には変化するため 1 次精度となる。実は本章でも、この名前こそ出さなかったものの同じステップを一度用いている。$\sqrt{\bar{\alpha}_t}$ で割る操作（38.6.4 節）によって線形項が取り除かれ、DDIM は 1 ストライドにわたってノイズ予測を一定に保つ。つまり DDIM は 1 次の指数積分器である。

この系統のサンプラは、どの出力（スコア、ノイズ予測、またはクリーンな点の予測）を一定に保つか、そしてどの時計を用いるかという点で異なる。より高次の手法では、定数の代わりに複数の予測値を通る多項式を用いる。その理論は Hochbruck and Ostermann (2010) が概観している。これに基づく拡散サンプラには DEIS (Zhang and Chen 2023) や DPM-Solver (Lu, Zhou, et al. 2022) がある。後者は演習問題 3（17.5 節）の題材であり、その後継手法は 17.10 節に挙げられている。SDE に対しても同じ考え方から、演習問題 7（38.4 節）の指数オイラー・丸山法のステップが得られる。

38.8.1.2 幾何学的数値積分とリープフロッグ法

摩擦のない振動子に対して前進オイラー法はステップごとにエネルギーを増加させ、後退オイラー法はエネルギーを減少させるため、長時間実行すると前者は発散し、後者は静止してしまう。よ

り高次の手法を用いても、これを先延ばしにできるにすぎない。**幾何学的数値積分**（geometric integration）が手法に求めるのは別の性質、すなわち厳密なフローを精度よく追跡するかどうかとは無関係に、厳密なフローが持つ構造を保存することである。シンプレクティックオイラー法やリープフロッグ法のような**シンプレクティック**手法は、厳密なフローと同様に、位置と運動量の平面における面積を保存する（座標が増えれば面積は体積になる）。

さらに後退誤差解析（backward error analysis）によれば、このような手法は $1/h$ に関して指数関数的に小さい誤差を除いて、わずかに異なるエネルギーに対する厳密なフローを追跡する。したがって真のエネルギーにおける誤差は、ホライズンとともに増大するのではなく、非常に長い時間にわたって有界にとどまる。演習問題 4（38.2 節）では、振動子に対するシンプレクティックオイラー法についてこの異なるエネルギーを求める。その等位集合は、真のエネルギーが描く円に近い楕円になる。勾配降下法にもエネルギーを損失に置き換えた同じ解析が成り立ち、損失に加わる項は Barrett and Dherin (2021) の暗黙的勾配正則化項（implicit gradient regularizer）である（同節の演習問題 5）。この理論は Hairer et al. (2006) が展開している。

リープフロッグ法は、ハミルトニアンモンテカルロ法 (Duane et al. 1987; Neal 2011) の内部で用いられる数値積分器でもある。このサンプラは、目標分布の負の対数密度をポテンシャルエネルギーとして持つ摩擦のない粒子をシミュレーションすることで遠くへの移動を提案し、メトロポリス則（36.8.3 節）によって採択または棄却する。リープフロッグ法のエネルギー誤差が有界にとどまるため、遠くへの移動であっても採択され得る。さらに、リープフロッグ法のステップは可逆で体積を保存するため、この判定則は 2 つのエネルギーを比較するだけでよく、ヤコビ行列を必要としない。

38.8.2 連続時間モデル

38.8.2.1 行列式を安価に計算できるフロー

離散的な正規化フロー（normalizing flow）は、ヤコビ行列の行列式を安価に計算できるよう各層に制約を課す。38.3.3 節では通常の制約として三角ヤコビ行列を挙げたが、層がどのようにしてそれを得るかは述べなかった。カップリング層 (Dinh et al. 2014; Dinh, Sohl-Dickstein, et al. 2017) は、座標の半分をそのまま残し、もう半分を前半の座標の関数によってシフトおよびスケーリングする。そのヤコビ行列は三角行列となり、逆変換は減算と除算だけで計算できる。1 つの層では状態の半分しか変化しないため、多数の層を積み重ね、層ごとに役割を入れ替える。

可逆残差ネットワーク（invertible residual network）(Behrmann et al. 2019) は別の対象に制約を課す。残差ブロックはオイラー法の 1 ステップであり（38.3.1 節）、分岐のリプシッツ定数を 1 未満に抑えれば、リプシッツ連続なベクトル場のフローと同様に（38.1.1.4 節）、そのステップは可逆になる。ヤコビ行列は一般の形状を保ち、その対数行列式は演習問題 4（38.3 節）のべき級数によって推定される。Papamakarios et al. (2021) はこれらのモデル族と、それぞれが何を犠牲にしているかを概観している。

38.8.2.2 可逆ソルバと正則化フロー

38.3.2 節の随伴法（adjoint method）は、逆方向の計算中に順方向の状態を再構成することでメモリを節約するが、縮小するフローではその再構成に伴う丸め誤差が増幅される。再構成には縮小がなくても生じる第 2 の誤差がある。ソルバの逆方向ステップは順方向ステップの逆写像に近いものの、厳密な逆写像ではない。そのため逆向きにたどって得られる状態は順方向に送り出した状態と一致せず、得られる勾配も実際に実行した計算の勾配ではなくなる。可逆ソルバ（reversible solver）は各ステップを代数的に元へ戻せるように構成されており、この第 2 の誤差を取り除く。ただし丸め誤差は依然として増幅されるため、硬い（stiff）ベクトル場に対してはチェックポイント保存（checkpointing）の方が引き続き安全な選択肢である。両者の比較は Kidger (2022)（第 5 章）にある。

尤度によって訓練されたフローには、数値積分が安価になるべき理由がない。尤度は各点が最終的にどこへ到達するかだけで決まり、たどった経路には依存しないからである。そこで Finlay et al. (2020) は、訓練中に運動エネルギー $\int \mathbb{E}\|\mathbf{f}_\theta\|^2\, dt$ にペナルティを課す。固定された 2 端点間では等速の直線経路が最小の運動エネルギーを持つため、このペナルティによって経路は直線に近づき、ソルバは直線経路を少数のステップで走破できる。フローマッチングは、直線的な回帰目標とノイズ・データのペアリング（38.7.3 節）を通じて同じ目標を追求する。

38.8.2.3 潜在微分方程式と制御微分方程式

本章のモデルは初期状態を最終状態へと写像する。しかし患者の診療記録のような時系列データは不規則な時刻に到着するため、そのモデルには各観測が得られた時点でそれを順次取り込むことが求められる。潜在 ODE（latent ODE）(Rubanova et al. 2019) はその半分を実現する。すなわち、観測列を 1 つの初期状態へエンコードし、任意の時刻集合における軌跡をデコードするため、後から到着した観測が進路を変えることはできない。

ニューラル制御微分方程式（neural controlled differential equation）(Kidger et al. 2020) では、すべての観測が状態に作用できる。この手法は補間されたデータ経路 $\mathbf{X}_t$ そのものによって状態を駆動し、$d\mathbf{z} = \mathbf{f}_\theta(\mathbf{z})\, d\mathbf{X}_t$（ただし $\mathbf{f}_\theta$ は行列値関数）と表される。この方程式はブラウン経路をデータに置き換えた SDE の形をしているが、補間された経路は滑らかであるため、積分は通常の積分になる。そのオイラーステップ $\mathbf{z}_{n+1} = \mathbf{z}_n + \mathbf{f}_\theta(\mathbf{z}_n)(\mathbf{X}_{n+1} - \mathbf{X}_n)$ は、データの増分を読み取るリカレントネットワークにほかならない。

38.8.2.4 相互作用する粒子としての Transformer

残差ネットワークはオイラーステップによって 1 つの状態を動かす（38.3.1 節）。一方、Transformer は系列内のすべてのトークンを同時に動かし、アテンションによって各トークンの速度が他のトークンの位置に依存するようになる。つまりトークンは相互作用する粒子である。Geshkovski et al. (2023) は、層正規化によってすべてのトークンが単位球面上に保たれる理想化されたモデルでこの系を解析し、深さが増すにつれてトークンがクラスタへと集まることを示している。本章で扱った系との唯一の違いは、ベクトル場が自ら動かす密度に依存するため、連続の式（continuity equation）が密度に関してもはや線形ではなくなる点である。

38.8.3 ノイズ

38.8.3.1 ワッサースタイン勾配フロー

勾配フロー上の点は損失を下っていく（38.2.4 節）。拡散過程の密度もまた、フォッカー・プランク方程式に従って決定論的に時間発展する。では、密度も何かを下っているのだろうか。ドリフトが勾配 $\mathbf{f} = -\nabla V$ であり、かつ g が定数であるときには、確かに下っている。このとき密度は自由エネルギー（free energy）$\mathcal{F}(p) = \int V p\, d\mathbf{x} + \frac{g^2}{2}\int p \log p\, d\mathbf{x}$ を下る。第 1 項は平均ポテンシャルエネルギーであり、すべての質量が V の底に集まるとき最小になる。第 2 項は負のエントロピーであり、質量が広がっているとき最小になる。ドリフトは第 1 項に、ノイズは第 2 項に働き、両者は e^{-2V/g^2} に比例する定常密度において釣り合う。

最急降下方向は距離の測り方に依存し、ここで 2 つの密度間の距離として用いなければならないのは、一方を他方へ輸送するコストを測る 2 次ワッサースタイン距離である。距離がどこに現れるかは 38.2.4 節の後退オイラーステップを見ればわかる。凸な損失に対して、このステップは $\mathcal{L}(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_k\|^2$ を最小化する。JKO スキーム (Jordan et al. 1998) は、このステップの点を密度に、二乗距離を輸送コストに置き換えた $p_{k+1} = \arg\min_p \ \mathcal{F}(p) + \frac{1}{2\eta} W_2^2(p, p_k)$ である。$\eta \to 0$ のとき、そのステップ列はフォッカー・プランク方程式の解へと収束する。

とりわけ記憶にとどめておくべき例がランジュバン動力学である。そのドリフトは目標分布のスコアの半分であり、ノイズは $g = 1$ を持つ（38.6.2 節）。目標分布を p_∞ と書き、粒子群の密度を表す p と区別しよう。すると $V = -\frac{1}{2}\log p_\infty$ となり、自由エネルギーは $\frac{1}{2}D_{\mathrm{KL}}(p \,\|\, p_\infty)$ となる。すなわちランジュバン粒子の群れは、輸送によって距離を測ったもとで、目標分布に対する KL ダイバージェンス上の勾配降下法を実行しているのである。連続の式からベナムー・ブレニエ (Benamou and Brenier 2000) に至るまで、本章で輸送が繰り返し姿を現す理由はここにある。

38.8.3.2 物理ノイズとウォン・ザカイの定理

38.4.2.2 節では左端点を用いて伊藤積分を定義し、ストラトノビッチ積分を定義する中点則では通常の連鎖律が保たれることを述べた。では、物理系を記述するのはこの 2 つのうちどちらか。物理系におけるノイズはブラウン経路ではない。短い記憶と滑らかな経路を持っており、滑らかな経路に対しては通常の積分ただ 1 つしか存在しない。Wong and Zakai (1965) は、そのような滑らかな経路がブラウン経路に近づくとき、通常の微分方程式の解がストラトノビッチ解へと収束することを証明した。1 次元において、$dX = f\,dt + g\,dW$ をストラトノビッチの意味で解釈した方程式は、ドリフトを $f + \frac{1}{2}g\,\partial_x g$ とした伊藤方程式に等しい。

拡散モデルの加法的ノイズ（additive noise）に対しては、この 2 つの解釈は一致する。乗法的ノイズ（multiplicative noise）に対しては両者は異なり、どちらを選ぶべきかをこの定理が教えてくれる。滑らかな物理ノイズを理想化した方程式はストラトノビッチとして解釈する。一方、離散時間で構築され、係数が自らの掛ける増分を先回りして見ることのできないモデルは伊藤として解釈する。確率的勾配降下法のステップはこの後者の種類に属する。

38.8.3.3 SDE としての確率的勾配降下法

勾配降下法は勾配フローに対する前進オイラー法である（38.2.4 節）。では、確率的勾配降下法は何の離散化なのだろうか。ミニバッチ勾配は全勾配にノイズを加えたものであり、その共分散 Σ は $1/B$ に比例してバッチサイズ B とともに縮小する。Q. Li et al. (2017) は SDE として $d\theta = -\nabla\mathcal{L}\,dt + \sqrt{\eta}\,\Sigma^{1/2}d\mathbf{W}$ を提案している。長さ η のオイラー・丸山法のステップをとってみよう。ノイズ係数にさらに $\sqrt{\eta}$ が掛かり、結果として学習率 η の確率的勾配降下法の 1 ステップが得られる。

したがって学習率は、ステップ幅として、そしてノイズの内部に、計 2 回現れる。Σ がいたるところ単位行列の同じ定数倍 $c\,\mathbf{I}$ であるとき、先ほどの自由エネルギーが $V = \mathcal{L}$ および $g^2 = \eta c$ としてそのまま当てはまり、損失に対してエントロピーの重みを定める。この意味において、学習率は温度である。二次損失の上では、反復列は OU 過程（38.4.3 節）の定常ガウス分布へと落ち着き、その分散は比 η/B とともに増大する。Mandt et al. (2017) はこの性質を利用し、確率的勾配降下法を近似的な事後分布サンプラとして動かしている。ただしこのモデル化には小さな学習率とほぼガウス分布に従う勾配ノイズが必要であり、実用上はその両方の仮定が破れることがある。

38.8.3.4 ニューラル SDE

学習されるモデル自体が、学習されるドリフトと学習されるノイズ振幅 g を持つ SDE であるとき、随伴系はどうなるのか。その場合、随伴系は同じブラウン経路で駆動される後向き SDE（backward SDE）となるため、そのブラウン経路を保存しておくか、乱数シードから再生成しなければならない (X. Li et al. 2020)。損失もまた変化する。このようなモデルの出力は確率的であるため、1 つの端点を 1 つの目標値と直接比較することはできない。訓練信号は、経路全体の法則（確率分布）間のダイバージェンス、あるいは経路同士を比較する識別器になる (Kidger 2022)。

ギルサノフの定理（Girsanov's theorem）(Øksendal 2003) は、同じ g とドリフト $\mathbf{f}$ および $\mathbf{f}'$ を持つ 2 つの過程に対して、このダイバージェンスを閉じた形で与える。両者の経路法則間の KL

ダイバージェンスは、初期法則間の KL ダイバージェンスに $\frac{1}{2}\int \mathbb{E}\|\mathbf{f}-\mathbf{f}'\|^2/g^2\,dt$（期待値は第 1 の過程の経路に関するもの）を加えた値に等しい。ここで g が一致するという条件は外せない。異なる g を持つ 2 つの過程は、たった 1 本の経路からでもその二次変分（38.4.2.1 節）によって見分けられるため、経路法則間の KL ダイバージェンスは無限大になる。

37.3.3.3 節のデータ処理不等式（data-processing inequality）により、端点の法則間の差は経路の法則間の差を超えられない。したがって、拡散モデルの厳密な逆ドリフトと学習された逆ドリフトにギルサノフの公式を適用すれば、演習問題 7（38.6 節）で経路法則を用いずに導出した尤度境界が得られる。

38.8.4 学習される生成器

この方向で本章の先に続く内容の大半は第 17 章で扱う。予測対象、損失の重み付け、およびスケジュールについては 17.6 節で解説しており、そこでは信号対雑音比（signal-to-noise ratio）の対数である $-2\log\bar{\sigma}_t$（38.6.4 節の記法）によって各選択肢が整理される (Kingma et al. 2021)。1 ステップまたは少数ステップでの生成は 17.8 節で扱う。以下では、両章の範囲を超える 4 つのトピックを取り上げる。

38.8.4.1 スライススコアマッチング

38.6.1 節ではトレースの計算コストを理由に陰的スコアマッチング（implicit score matching）を脇に置き、そこでは採用しなかった抜け道として、ランダムな方向ごとに 1 回のベクトル・ヤコビ行列積で済む Hutchinson の推定値に言及した。この推定値を用いて訓練する手法はスライススコアマッチング（sliced score matching）と呼ばれる (Song et al. 2019)。この手法はノイズレベルを選ぶ必要がなく、データそのもののスコアを学習する。もっとも、拡散モデルにはこの利点の使い道がない。拡散モデルが必要とするのはあらゆるノイズレベルにおけるノイズ付きデータのスコアであり、ノイズ除去スコアマッチングはトレースなしでそれを提供するため、ノイズ除去が標準となったのである。

38.8.4.2 スコア誤差とサンプリング誤差

本章のほぼすべてのサンプラは厳密なスコアを用いて動かしたが、訓練されたスコアはどこにおいても厳密ではない。では、どれほど正確であればよいのだろうか。平均的に良ければ十分であり、それ以上は必要ない。Sitan Chen et al. (2023) は、データ分布と離散化された逆 SDE の出力との間の全変動（total variation）を 3 つの項の和で上から抑えており、その各項はすでに本章で個別に登場している。p_T とサンプラの出発点であるガウス分布との距離は e^{-T} のオーダーで減衰し、離散化誤差は次元とともに増大しステップ幅とともに減少する。この 2 つは 38.5.5 節で測定した。第 3 の項はノイズ付きデータのもとでのスコアの二乗平均平方根誤差であり、これはノイズ除去損失がその下限からの超過分として測っている量にほかならない。この定理はそれ以外の項が一切入らないことを示しており、仮定もわずかである。すなわち、データの有限な二次モーメントと、ノイズ付き密度のスコアがリプシッツ連続であることだけを要求する。単峰性は不要であるため、38.6.2 節でランジュバン動力学を阻んだ 2 つのモードも対象に含まれる。

38.8.4.3 シュレーディンガーブリッジとエントロピー輸送

1931 年、Schrödinger は粒子に関するある問いを立てた。自由に拡散する粒子群が、現在ある形状で観測され、後の時刻に別の起こりそうにない形状で観測されたとき、そこへ至る最も確からしい道筋はどのようなものか、という問いである。独立な多数の粒子からなる群れでは、自由拡散とは異なる経路法則に従って群れが動くのを目にする確率は指数関数的に減少し、その減衰率は自由拡散に対するその経路法則の KL ダイバージェンスで与えられる。したがって答えは、与

えられた 2 つの周辺分布を持つ経路法則のうち、KL ダイバージェンスの意味で自由拡散に最も近いものとなる。これを**シュレーディンガーブリッジ**と呼ぶ。

このブリッジを 38.7.3 節のミニバッチペアリングと並べてみよう。ミニバッチペアリングは各バッチ上での厳密な割り当てであり、訓練される生成器は ODE である。これに対しブリッジは、37.4.2.4 節のシンクホーン反復によって計算されるエントロピー正則化最適輸送の計画 (Cuturi 2013; Peyré and Cuturi 2019) に従って両端を対応付け、その生成器は SDE となる。自由拡散のノイズが小さくなるにつれて、この輸送計画は最適輸送計画へと近づく。

拡散モデルと並べると、ブリッジの方が難しい問題である。拡散モデルは順過程を固定して逆過程だけを学習し、事前分布には近似的にしか到達しない。一方ブリッジは与えられた 2 つの法則を有限時間で結ぶため、両方向を学習しなければならない。De Bortoli et al. (2021) は順方向と逆方向の拡散を交互に適合させており、その交互反復の最初の 1 巡は拡散モデルの訓練に一致する。Albergo et al. (2023) はブリッジ問題を解くことなく、ODE と SDE の両方の生成器に到達している。任意の 2 つの分布からのサンプル $\mathbf{x}_0$、$\mathbf{x}_1$ と、独立なガウス変数 $\mathbf{w}$ をとろう。経路 $a_t\mathbf{x}_0 + b_t\mathbf{x}_1 + c_t\mathbf{w}$（ただし両端で c_t はゼロ）が持つ周辺分布は、1 つの ODE または SDE の族によって生成できる（演習問題 1、38.7 節）。

38.8.4.4 直線的カップリングと最適カップリング

Reflow によって変化しないペアリングは交差しない直線経路を持つが、2 次元以上ではそれが最適であるとは限らない。38.7.3 節の例では、ガウス分布に従う点群の各点を、それを 4 分の 1 回転させた点と対応付けている。線分は決して交差しないが、コストが 4 かかるのに対し、本来は $W_2^2 = 0$ である。このペアリングが持ち、最適なペアリングが持たないものは回転である。最適なフローの速度場は勾配であるが (Benamou and Brenier 2000)、点群を原点のまわりに回転させるベクトル場は勾配ではない。

そこで Liu (2022) は速度場を勾配 $\nabla\phi_t$ に制限し、この制限付き Reflow によって変化しないペアリングが最適なペアリングになると論じた。しかし、この主張が成り立つにはそこで置かれた仮定以上の条件が必要である。Hertrich et al. (2025) は、移動中の点群があらゆる時刻で空間全体を覆っている場合にこれを証明する一方、点群が複数の離れた塊に分かれている場合の反例を与えている。したがってこの制限だけでは、38.7.3 節が残した問い、すなわち訓練のみによって最適なペアリングをいかにして見つけるかという問いは決着しない。

38.8.5 ダイナミクスと生成フロー全般の参考資料

以下の文献を開く前に 1 点注意しておく。MIT の講義ノートと講義、Flow Matching Guide、およびフローマッチングに関する 2 編の論文では、時刻を 0 のノイズから 1 のデータへと進めるのに対し、本章ではその逆向きに進めている（Figure 38.26）。

書籍と講義ノート

- 『Nonlinear Dynamics and Chaos』（Steven Strogatz 著）：フロー、不動点とその安定性に関する幾何学的な入門書であり、38.1 節の内容を扱っている。
- 『On Neural Differential Equations』（Patrick Kidger 著、2022 年）：ニューラル ODE、制御微分方程式、確率微分方程式を書籍規模で体系的に論じた文献であり、38.3 節の随伴法の仕組みが余すところなく展開されているほか、38.2 節のソルバの選び方に関する助言も収められている。
- 『Applied Stochastic Differential Equations』（Särkkä、Solin 著）：38.4 節および 38.5 節に対応する伊藤解析、SDE、フォッカー・プランク方程式を応用重視の視点から解説した書籍であり、著者らによって全文の PDF が公開されている。

- 『Stochastic Differential Equations: An Introduction with Applications』（Bernt Øksendal 著）：38.4 節の背景にあるブラウン運動、伊藤積分、および解の存在理論に関する厳密な基本文献である。
- 『Numerical Solution of Stochastic Differential Equations』（Kloeden、Platen 著）：38.4 節のシミュレーション部分に対応する、オイラー・丸山法、ミルスタイン法、より高次のスキーム、およびそれらの強収束次数・弱収束次数に関する基本文献である。
- 「An Introduction to Flow Matching and Diffusion Models (MIT 6.S184 lecture notes)」（Holderrieth、Erives 著）：SDE と連続の式から拡散モデルとフローマッチングを導出する自己完結した講義ノートであり、38.6 節と 38.7 節の副読本となる。

講義とチュートリアル

- 「MIT 6.S184: Generative AI with Stochastic Differential Equations」：38.5 節から 38.7 節への道筋に沿って第一原理から拡散モデルを構築する講義、スライド、および演習ラボであり、オープンライセンスで公開されている。
- 「Understanding Diffusion Models: A Unified Perspective」（Calvin Luo 著、2022 年）：38.6 節に関連して、拡散モデルの変分的な見方とスコアベースの見方を並べて導出している。
- 「Flow Matching Guide and Code」（Lipman ら著、2024 年）：38.7 節向けのフローマッチングに関する詳細な解説文献であり、PyTorch ライブラリ `facebookresearch/flow_matching` と対になっている。

基礎論文

- 「Neural Ordinary Differential Equations」（Chen ら著、NeurIPS 2018）：38.3 節と同様に、ネットワークのダイナミクスを ODE としてパラメータ化し、随伴法を用いて訓練する。
- 「Generative Modeling by Estimating Gradients of the Data Distribution」（Song、Ermon 著、NeurIPS 2019）：38.6 節に対応する、ノイズ除去スコアマッチングとアニーリング付きランジュバンサンプリングによるスコアベース生成を提案している。
- 「Denoising Diffusion Probabilistic Models」（Ho、Jain、Abbeel 著、NeurIPS 2020）：38.6 節で離散化された SDE として解釈した、離散時間の DDPM 定式化を与えている。
- 「Score-Based Generative Modeling through SDEs」（Song ら著、ICLR 2021）：スコアマッチングと拡散モデルを順方向・逆方向の SDE として統一し、38.5 節の確率フロー ODE を導入している。
- 「Denoising Diffusion Implicit Models」（Song、Meng、Ermon 著、ICLR 2021）：訓練済みの DDPM を再利用してより少ないステップ数で生成を行う、38.6 節の決定論的サンプラである。
- 「Flow Matching for Generative Modeling」（Lipman ら著、ICLR 2023）：条件付きベクトル場の回帰によって連続フローをシミュレーションなしで訓練する、38.7 節の手法である。
- 「Flow Straight and Fast: Rectified Flow」（Liu、Gong、Liu 著、ICLR 2023）：38.7 節に対応する、少数ステップサンプリングのための直線補間経路と Reflow を提案している。

ブログと視覚的解説

- 「Generative Modeling by Estimating Gradients of the Data Distribution」（Yang Song 著）：スコアマッチング、ランジュバン動力学、および 38.6 節の SDE による描像を図とともに結びつける著者自身の解説記事である。
- 「What are Diffusion Models?」（Lilian Weng 著）：DDPM からスコア SDE やガイダンスに至る拡散モデルのサーベイ記事であり、2024 年まで更新されている（38.6 節に対応）。
- 「Perspectives on Diffusion」（Sander Dieleman 著）：拡散モデルがいかにしてノイズ除去器、スコア推定器、逆 SDE ソルバ、そしてフローという顔を併せ持つかを示す記事であり、38.7.1.3 節の対応辞書にあたる。

- 「The Annotated Diffusion Model」（Hugging Face）：38.6 節の数学を実行可能なコードへと結びつける、PyTorch による DDPM の 1 行ごとの実装解説である。

参考文献

Albergo, Michael S., Nicholas M. Boffi, and Eric Vanden-Eijnden. 2023. "Stochastic Interpolants: A Unifying Framework for Flows and Diffusions." **arXiv Preprint arXiv:2303.08797**.

Alemi, Alexander A., Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. "Deep Variational Information Bottleneck." **International Conference on Learning Representations**.

Amari, Shun-ichi. 1998. "Natural Gradient Works Efficiently in Learning." **Neural Computation** 10 (2): 251–76.

Amari, Shun-ichi. 2016. **Information Geometry and Its Applications**. Springer.

Anderson, Brian D. O. 1982. "Reverse-Time Diffusion Equation Models." **Stochastic Processes and Their Applications** 12 (3): 313–26.

Angelopoulos, Anastasios N., and Stephen Bates. 2023. "Conformal Prediction: A Gentle Introduction." **Foundations and Trends in Machine Learning** 16 (4): 494–591. https://doi.org/10.1561/2200000101.

Arıkan, Erdal. 2009. "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels." **IEEE Transactions on Information Theory** 55 (7): 3051–73. https://doi.org/10.1109/TIT.2009.2021379.

Arjevani, Yossi, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake Woodworth. 2023. "Lower Bounds for Non-Convex Stochastic Optimization." **Mathematical Programming** 199: 165–214.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. "Wasserstein Generative Adversarial Networks." **Proceedings of the 34th International Conference on Machine Learning**, 214–23.

Armijo, Larry. 1966. "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives." **Pacific Journal of Mathematics** 16 (1): 1–3.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. "Layer Normalization." **ArXiv:1607.06450**. https://arxiv.org/abs/1607.06450.

Bach, Francis, and Eric Moulines. 2013. “Non-Strongly-Convex Smooth Stochastic Approximation with Convergence Rate $O(1/n)$.” **Advances in Neural Information Processing Systems** 26.

Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. 2019. “Deep Equilibrium Models.” **Advances in Neural Information Processing Systems** 32.

Baik, Jinho, and Jack W. Silverstein. 2006. “Eigenvalues of Large Sample Covariance Matrices of Spiked Population Models.” **Journal of Multivariate Analysis** 97 (6): 1382–408. https://doi.org/10.1016/j.jmva.2005.08.003.

Baldi, Pierre, and Kurt Hornik. 1989. “Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima.” **Neural Networks** 2 (1): 53–58.

Ballé, Johannes, Valero Laparra, and Eero P. Simoncelli. 2017. “End-to-End Optimized Image Compression.” **International Conference on Learning Representations**.

Ballé, Johannes, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. 2018. “Variational Image Compression with a Scale Hyperprior.” **International Conference on Learning Representations**.

Barber, David, and Felix Agakov. 2003. “The IM Algorithm: A Variational Approach to Information Maximization.” **Advances in Neural Information Processing Systems** 16.

Barrett, David G. T., and Benoit Dherin. 2021. “Implicit Gradient Regularization.” **International Conference on Learning Representations**.

Bartlett, Peter L., Philip M. Long, Gábor Lugosi, and Alexander Tsigler. 2020. “Benign Overfitting in Linear Regression.” **Proceedings of the National Academy of Sciences** 117 (48): 30063–70.

Bartlett, Peter L., and Shahar Mendelson. 2002. “Rademacher and Gaussian Complexities: Risk Bounds and Structural Results.” **Journal of Machine Learning Research** 3: 463–82.

Baur, Walter, and Volker Strassen. 1983. “The Complexity of Partial Derivatives.” **Theoretical Computer Science** 22 (3): 317–30.

Baydin, Atılım Güneş, Barak A. Pearlmutter, Don Syme, Frank Wood, and Philip Torr. 2022. “Gradients Without Backpropagation.” **arXiv Preprint arXiv:2202.08587**.

Beck, Amir, and Marc Teboulle. 2003. “Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization.” **Operations Research Letters** 31 (3): 167–75. https://doi.org/10.1016/S0167-6377(02)00231-6.

Beck, Amir, and Marc Teboulle. 2009. “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.” **SIAM Journal on Imaging Sciences** 2 (1): 183–202.

Beck, Amir, and Luba Tetruashvili. 2013. “On the Convergence of Block Coordinate Descent Type Methods.” **SIAM Journal on Optimization** 23 (4): 2037–60.

Behrmann, Jens, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. “Invertible Residual Networks.” **International Conference on Machine Learning**, 573–82.

Belghazi, Mohamed Ishmael, Aristide Baratin, Sai Rajeshwar, et al. 2018. “Mutual Information Neural Estimation.” **Proceedings of the 35th International Conference on Machine Learning**, 531–40.

Benamou, Jean-David, and Yann Brenier. 2000. “A Computational Fluid Mechanics Solution to the Monge–Kantorovich Mass Transfer Problem.” **Numerische Mathematik** 84 (3): 375–93.

Benaych-Georges, Florent, and Raj Rao Nadakuditi. 2012. “The Singular Values and Vectors of Low Rank Perturbations of Large Rectangular Random Matrices.” **Journal of Multivariate Analysis** 111: 120–35. https://doi.org/10.1016/j.jmva.2012.04.019.

Bengio, Yoshua, Nicholas Léonard, and Aaron Courville. 2013. “Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation.” **arXiv Preprint arXiv:1308.3432**.

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. 1994. “Learning Long-Term Dependencies with Gradient Descent Is Difficult.” **IEEE Transactions on Neural Networks** 5 (2): 157–66. https://doi.org/10.1109/72.279181.

Benjamini, Yoav, and Yosef Hochberg. 1995. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.” **Journal of the Royal Statistical Society: Series B (Methodological)** 57 (1): 289–300.

Bernstein, Jeremy, and Laker Newhouse. 2024. “Old Optimizer, New Norm: An Anatomy.” **arXiv Preprint arXiv:2409.20325**.

Bertoin, David, Jérôme Bolte, Sébastien Gerchinovitz, and Edouard Pauwels. 2021. “Numerical Influence of ReLU’(0) on Backpropagation.” **Advances in Neural Information Processing Systems** 34: 468–79. https://arxiv.org/abs/2106.12915.

Billingsley, Patrick. 1995. **Probability and Measure**. 3rd ed. Wiley.

Biroli, Giulio, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. 2024. “Dynamical Regimes of Diffusion Models.” **Nature Communications** 15: 9957. https://arxiv.org/abs/2402.18491.

Bishop, Christopher M. 2006. **Pattern Recognition and Machine Learning**. Springer. https://doi.org/10.1007/978-0-387-45528-0.

Black, Fischer, and Myron Scholes. 1973. “The Pricing of Options and Corporate Liabilities.” **Journal of Political Economy** 81: 637–54. https://doi.org/10.1086/260062.

Blier, Léonard, and Yann Ollivier. 2018. “The Description Length of Deep Learning Models.” **Advances in Neural Information Processing Systems** 31. https://arxiv.org/abs/1802.07044.

Blum, Avrim, and Moritz Hardt. 2015. “The Ladder: A Reliable Leaderboard for Machine Learning Competitions.” **Proceedings of the 32nd International Conference on Machine Learning**, Proceedings of machine learning research, vol. 37: 1006–14.

Bogachev, Vladimir I., Nicolai V. Krylov, Michael Röckner, and Stanislav V. Shaposhnikov. 2015. **Fokker–Planck–Kolmogorov Equations**. Vol. 207. Mathematical Surveys and Monographs. American Mathematical Society.

Bolte, Jérôme, and Edouard Pauwels. 2021. “Conservative Set Valued Fields, Automatic Differentiation, Stochastic Gradient Methods and Deep Learning.” **Mathematical Programming** 188 (1): 19–51.

Bonferroni, Carlo E. 1936. “Teoria Statistica Delle Classi e Calcolo Delle Probabilità.” **Pubblicazioni Del R. Istituto Superiore Di Scienze Economiche e Commerciali Di Firenze** 8: 3–62.

Bonneel, Nicolas, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. 2015. “Sliced and Radon Wasserstein Barycenters of Measures.” **Journal of Mathematical Imaging and Vision** 51 (1): 22–45. https://doi.org/10.1007/s10851-014-0506-3.

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. “A Training Algorithm for Optimal Margin Classifiers.” **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**, 144–52.

Bottou, Léon, Frank E. Curtis, and Jorge Nocedal. 2018. “Optimization Methods for Large-Scale Machine Learning.” **SIAM Review** 60 (2): 223–311.

Boucheron, Stéphane, Olivier Bousquet, and Gábor Lugosi. 2005. “Theory of Classification: A Survey of Some Recent Advances.” **ESAIM: Probability and Statistics** 9: 323–75. https://doi.org/10.1214/154957805100000046.

Boucheron, Stéphane, Gábor Lugosi, and Pascal Massart. 2013. **Concentration Inequalities: A Nonasymptotic Theory of Independence**. Oxford University Press.

Bousquet, Olivier, and André Elisseeff. 2002. “Stability and Generalization.” **Journal of Machine Learning Research** 2: 499–526. https://www.jmlr.org/papers/v2/bousquet02a.html.

Boyd, Stephen, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.” **Foundations and Trends in Machine Learning** 3 (1): 1–122. https://doi.org/10.1561/2200000016.

Boyd, Stephen, and Lieven Vandenberghe. 2004. **Convex Optimization**. Cambridge University Press. https://web.stanford.edu/~boyd/cvxbook/.

Bray, Alan J., and David S. Dean. 2007. “Statistics of Critical Points of Gaussian Fields on Large-Dimensional Spaces.” **Physical Review Letters** 98 (15): 150201.

Brenier, Yann. 1991. “Polar Factorization and Monotone Rearrangement of Vector-Valued

Functions." **Communications on Pure and Applied Mathematics** 44 (4): 375–417.

Bretagnolle, Jean, and Catherine Huber. 1978. "Estimation Des Densités: Risque Minimax." In **Séminaire de Probabilités XII**, vol. 649. Lecture Notes in Mathematics. Springer.

Brin, Sergey, and Lawrence Page. 1998. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." **Computer Networks and ISDN Systems** 30 (1–7): 107–17.

Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. 2016. "Importance Weighted Autoencoders." **International Conference on Learning Representations**. https://arxiv.org/abs/1509.00519.

Candès, Emmanuel J., and Benjamin Recht. 2009. "Exact Matrix Completion via Convex Optimization." **Foundations of Computational Mathematics** 9 (6): 717–72.

Canonne, Clément L. 2022. "A Short Note on an Inequality Between KL and TV." **arXiv Preprint arXiv:2202.07198**. https://arxiv.org/abs/2202.07198.

Carson, Erin, and Nicholas J. Higham. 2018. "Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions." **SIAM Journal on Scientific Computing** 40 (2): A817–47.

Čencov, N. N. 1982. **Statistical Decision Rules and Optimal Inference**. Vol. 53. Translations of Mathematical Monographs. American Mathematical Society.

Chebyshev, Pafnuty L. 1867. "Des Valeurs Moyennes." **Journal de Mathématiques Pures Et Appliquées** 12 (2): 177–84.

Chechik, Gal, Amir Globerson, Naftali Tishby, and Yair Weiss. 2005. "Information Bottleneck for Gaussian Variables." **Journal of Machine Learning Research** 6: 165–88.

Chen, Lei, and Joan Bruna. 2023. "Beyond the Edge of Stability via Two-Step Gradient Updates." **International Conference on Machine Learning**, 4330–91.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. "Neural Ordinary Differential Equations." **Advances in Neural Information Processing Systems** 31.

Chen, Sitan, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. 2023. "Sampling Is as Easy as Learning the Score: Theory for Diffusion Models with Minimal Data Assumptions." **International Conference on Learning Representations**.

Chen, Tianqi, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. "Training Deep Nets with Sublinear Memory Cost." **arXiv Preprint arXiv:1604.06174**.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. "A Simple Framework for Contrastive Learning of Visual Representations." **Proceedings of the 37th International Conference on Machine Learning**, 1597–607.

Cheng, Pengyu, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin.

2020. “CLUB: A Contrastive Log-Ratio Upper Bound of Mutual Information.” **Proceedings of the 37th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 119: 1779–88.

Church, Kenneth Ward, and Patrick Hanks. 1990. “Word Association Norms, Mutual Information, and Lexicography.” **Computational Linguistics** 16 (1): 22–29.

Church, Kenneth W., and William A. Gale. 1995. “Poisson Mixtures.” **Natural Language Engineering** 1 (2): 163–90. https://doi.org/10.1017/S1351324900000139.

Chwialkowski, Kacper, Heiko Strathmann, and Arthur Gretton. 2016. “A Kernel Test of Goodness of Fit.” **Proceedings of the 33rd International Conference on Machine Learning**, 2606–15.

Coddington, Earl A., and Norman Levinson. 1955. **Theory of Ordinary Differential Equations**. McGraw-Hill.

Cohen, Jeremy M., Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. 2021. “Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability.” **International Conference on Learning Representations**.

Cohen, Joel E., and Charles M. Newman. 1984. “The Stability of Large Random Matrices and Their Products.” **The Annals of Probability** 12 (2): 283–310. https://doi.org/10.1214/aop/1176993291.

Cover, T, and JM Thomas. 1999. **Elements of Information Theory**. John Wiley & Sons. https://doi.org/10.1002/047174882X.

Cramér, H. 1946. **Mathematical Methods of Statistics.** Princeton University Press.

Csiszár, Imre. 1967. “Information-Type Measures of Difference of Probability Distributions and Indirect Observations.” **Studia Scientiarum Mathematicarum Hungarica** 2: 299–318.

Csiszár, Imre. 2008. “Axiomatic Characterizations of Information Measures.” **Entropy** 10 (3): 261–73. https://doi.org/10.3390/e10030261.

Cuturi, Marco. 2013. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport.” **Advances in Neural Information Processing Systems** 26.

Dahlquist, Germund G. 1963. “A Special Stability Problem for Linear Multistep Methods.” **BIT Numerical Mathematics** 3: 27–43.

Damian, Alex, Eshaan Nichani, and Jason D. Lee. 2023. “Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability.” **International Conference on Learning Representations**.

Darmois, Georges. 1935. “Sur Les Lois de Probabilité à Estimation Exhaustive.” **Comptes Rendus de l’Académie Des Sciences** 200: 1265–66.

Dasgupta, Sanjoy, and Anupam Gupta. 2003. “An Elementary Proof of a Theorem of Johnson and Lindenstrauss.” **Random Structures & Algorithms** 22 (1): 60–65.

Daskalakis, Constantinos, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. 2018. “Training GANs with Optimism.” **International Conference on Learning Representations**.

Dauphin, Yann N., Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. 2014. “Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization.” **Advances in Neural Information Processing Systems** 27: 2933–41.

Daxberger, Erik, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. 2021. “Laplace Redux – Effortless Bayesian Deep Learning.” **Advances in Neural Information Processing Systems** 34. https://arxiv.org/abs/2106.14806.

De Bortoli, Valentin, James Thornton, Jeremy Heng, and Arnaud Doucet. 2021. “Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling.” **Advances in Neural Information Processing Systems** 34: 17695–709.

Dean, David S., and Satya N. Majumdar. 2006. “Large Deviations of Extreme Eigenvalues of Random Matrices.” **Physical Review Letters** 97 (16): 160201.

DeepSeek-AI. 2024. “DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model.” **arXiv Preprint arXiv:2405.04434**.

Defazio, Aaron, Francis Bach, and Simon Lacoste-Julien. 2014. “SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives.” **Advances in Neural Information Processing Systems** 27.

Defazio, Aaron, Ashok Cutkosky, Harsh Mehta, and Konstantin Mishchenko. 2023. “Optimal Linear Decay Learning Rate Schedules and Further Refinements.” **arXiv Preprint arXiv:2310.07831**.

Delétang, Grégoire, Anian Ruoss, Paul-Ambroise Duquenne, et al. 2023. “Language Modeling Is Compression.” **ArXiv:2309.10668**. https://arxiv.org/abs/2309.10668.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin. 1977. “Maximum Likelihood from Incomplete Data via the EM Algorithm.” **Journal of the Royal Statistical Society: Series B** 39 (1): 1–38.

Deshpande, Ishan, Ziyu Zhang, and Alexander G. Schwing. 2018. “Generative Modeling Using the Sliced Wasserstein Distance.” **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**.

Dhariwal, Prafulla, and Alex Nichol. 2021. “Diffusion Models Beat GANs on Image Synthesis.” **Advances in Neural Information Processing Systems** 34.

DiCiccio, Thomas J., and Bradley Efron. 1996. “Bootstrap Confidence Intervals.” **Statistical Science** 11 (3): 189–228. https://doi.org/10.1214/ss/1032280214.

Dinh, Laurent, David Krueger, and Yoshua Bengio. 2014. “NICE: Non-Linear Independent Components Estimation.” **ArXiv:1410.8516**. https://arxiv.org/abs/1410.8516.

Dinh, Laurent, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. 2017. “Sharp Minima Can Generalize for Deep Nets.” **Proceedings of the 34th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 70: 1019–28. https://proceedings.mlr.press/v70/dinh17b.html.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio. 2017. “Density Estimation Using Real NVP.” **International Conference on Learning Representations**. https://openreview.net/forum?id=HkpbnH9lx.

Domingos, Pedro, and Michael Pazzani. 1997. “On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss.” **Machine Learning** 29: 103–30.

Donsker, Monroe D., and S. R. Srinivasa Varadhan. 1983. “Asymptotic Evaluation of Certain Markov Process Expectations for Large Time. IV.” **Communications on Pure and Applied Mathematics** 36 (2): 183–212.

Dormand, John R., and Peter J. Prince. 1980. “A Family of Embedded Runge–Kutta Formulae.” **Journal of Computational and Applied Mathematics** 6 (1): 19–26.

Dowson, D. C., and B. V. Landau. 1982. “The Fréchet Distance Between Multivariate Normal Distributions.” **Journal of Multivariate Analysis** 12 (3): 450–55.

Drori, Yoel, and Marc Teboulle. 2014. “Performance of First-Order Methods for Smooth Convex Minimization: A Novel Approach.” **Mathematical Programming** 145 (1–2): 451–82. https://doi.org/10.1007/s10107-013-0653-0.

Duane, Simon, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. 1987. “Hybrid Monte Carlo.” **Physics Letters B** 195 (2): 216–22.

Duchi, John, Elad Hazan, and Yoram Singer. 2011. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.” **Journal of Machine Learning Research** 12: 2121–59. https://jmlr.org/papers/v12/duchi11a.html.

Duchi, John, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. 2008. “Efficient Projections onto the ℓ_1-Ball for Learning in High Dimensions.” **Proceedings of the 25th International Conference on Machine Learning (ICML)**, 272–79.

Duda, Jarek. 2013. **Asymmetric Numeral Systems: Entropy Coding Combining Speed of Huffman Coding with Compression Rate of Arithmetic Coding**. arXiv:1311.2540. https://arxiv.org/abs/1311.2540.

Dupont, Emilien, Arnaud Doucet, and Yee Whye Teh. 2019. “Augmented Neural ODEs.” **Advances in Neural Information Processing Systems** 32.

Dwork, Cynthia, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. 2015. “Preserving Statistical Validity in Adaptive Data Analysis.” **ACM Symposium on Theory of Computing**, 117–26.

Dziugaite, Gintare Karolina, and Daniel M. Roy. 2017. “Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters Than Training Data.” **Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)**. https://arxiv.org/abs/1703.11008.

Eckart, Carl, and Gale Young. 1936. “The Approximation of One Matrix by Another of Lower Rank.” **Psychometrika** 1 (3): 211–18.

Efron, Bradley. 1979. “Bootstrap Methods: Another Look at the Jackknife.” **Annals of Statistics** 7 (1): 1–26.

Efron, Bradley. 2011. “Tweedie’s Formula and Selection Bias.” **Journal of the American Statistical Association** 106 (496): 1602–14.

Elhage, Nelson, Tristan Hume, Catherine Olsson, et al. 2022. “Toy Models of Superposition.” **Transformer Circuits Thread**.

Endres, Dominik M., and Johannes E. Schindelin. 2003. “A New Metric for Probability Distributions.” **IEEE Transactions on Information Theory** 49 (7): 1858–60.

Erven, Tim van, and Peter Harremoës. 2014. “Rényi Divergence and Kullback–Leibler Divergence.” **IEEE Transactions on Information Theory** 60 (7): 3797–820.

Feydy, Jean, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. 2019. “Interpolating Between Optimal Transport and MMD Using Sinkhorn Divergences.” **Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics**, 2681–90.

Finlay, Chris, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M. Oberman. 2020. “How to Train Your Neural ODE: The World of Jacobian and Kinetic Regularization.” **International Conference on Machine Learning**.

Fisher, Ronald A. 1925. “Theory of Statistical Estimation.” **Mathematical Proceedings of the Cambridge Philosophical Society** 22 (5): 700–725.

Fisher, Ronald A. 1935. **The Design of Experiments**. Oliver; Boyd.

Fisher, Ronald A. 1936. “The Use of Multiple Measurements in Taxonomic Problems.” **Annals of Eugenics** 7 (2): 179–88. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Fokker, Adriaan D. 1914. “Die Mittlere Energie Rotierender Elektrischer Dipole Im Strahlungsfeld.” **Annalen Der Physik** 348 (5): 810–20.

Folland, Gerald B. 1999. **Real Analysis: Modern Techniques and Their Applications**. 2nd ed. Wiley.

Frank, Marguerite, and Philip Wolfe. 1956. “An Algorithm for Quadratic Programming.” **Naval Research Logistics Quarterly** 3 (1–2): 95–110. https://doi.org/10.1002/nav.3800030109.

Frey, Brendan J., and Geoffrey E. Hinton. 1997. “Efficient Stochastic Source Coding and an Application to a Bayesian Network Source Model.” **The Computer Journal** 40 (2/3): 157–65.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2010. “Regularization Paths for Generalized Linear Models via Coordinate Descent.” **Journal of Statistical Software** 33 (1): 1–22.

Furstenberg, Harry, and Harry Kesten. 1960. “Products of Random Matrices.” **The Annals of Mathematical Statistics** 31 (2): 457–69.

Gavish, Matan, and David L. Donoho. 2014. “The Optimal Hard Threshold for Singular Values Is $4/\sqrt{3}$.” **IEEE Transactions on Information Theory** 60 (8): 5040–53.

Geman, Stuart. 1986. “The Spectral Radius of Large Random Matrices.” **The Annals of Probability** 14 (4): 1318–28.

Geman, Stuart, Elie Bienenstock, and René Doursat. 1992. “Neural Networks and the Bias/Variance Dilemma.” **Neural Computation** 4 (1): 1–58.

Geman, Stuart, and Donald Geman. 1984. “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images.” **IEEE Transactions on Pattern Analysis and Machine Intelligence** 6 (6): 721–41.

Genevay, Aude, Gabriel Peyré, and Marco Cuturi. 2018. “Learning Generative Models with Sinkhorn Divergences.” **Proceedings of the 21st International Conference on Artificial Intelligence and Statistics**.

Geng, Zhengyang, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. 2021. “On Training Implicit Models.” **Advances in Neural Information Processing Systems** 34. https://arxiv.org/abs/2111.05177.

Gershgorin, Semyon A. 1931. “Über Die Abgrenzung Der Eigenwerte Einer Matrix.” **Izvestiya Akademii Nauk SSSR, Seriya Matematicheskaya**, no. 6: 749–54.

Geshkovski, Borjan, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. 2023. “A Mathematical Perspective on Transformers.” **arXiv Preprint arXiv:2312.10794**.

Geweke, John. 1989. “Bayesian Inference in Econometric Models Using Monte Carlo Integration.” **Econometrica** 57 (6): 1317–39. https://doi.org/10.2307/1913710.

Ghadimi, Saeed, and Guanghui Lan. 2013. “Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming.” **SIAM Journal on Optimization** 23 (4): 2341–68. https://arxiv.org/abs/1309.5549.

Gibbs, Josiah Willard. 1902. **Elementary Principles of Statistical Mechanics**. Scribner’s.

Giles, Michael B. 2008. “Multilevel Monte Carlo Path Simulation.” **Operations Research** 56 (3): 607–17.

Giles, Mike B. 2008. **An Extended Collection of Matrix Derivative Results for Forward and Reverse Mode Algorithmic Differentiation**. NA-08/01. Oxford University Computing Laboratory.

Ginibre, Jean. 1965. "Statistical Ensembles of Complex, Quaternion, and Real Matrices." **Journal of Mathematical Physics** 6 (3): 440–49. https://doi.org/10.1063/1.1704292.

Gnedenko, B. V., and A. N. Kolmogorov. 1954. **Limit Distributions for Sums of Independent Random Variables**. Addison-Wesley.

Gneiting, Tilmann, Fadoua Balabdaoui, and Adrian E. Raftery. 2007. "Probabilistic Forecasts, Calibration and Sharpness." **Journal of the Royal Statistical Society Series B: Statistical Methodology** 69 (2): 243–68. https://doi.org/10.1111/j.1467-9868.2007.00587.x.

Gneiting, Tilmann, and Adrian E. Raftery. 2007. "Strictly Proper Scoring Rules, Prediction, and Estimation." **Journal of the American Statistical Association** 102 (477): 359–78. https://doi.org/10.1198/016214506000001437.

Goldberg, David. 1991. "What Every Computer Scientist Should Know about Floating-Point Arithmetic." **ACM Computing Surveys** 23 (1): 5–48.

Goldfeld, Ziv, Ewout van den Berg, Kristjan Greenewald, et al. 2019. "Estimating Information Flow in Deep Neural Networks." **Proceedings of the 36th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 97: 2299–308.

Golub, Gene H, and Charles F Van Loan. 1996. **Matrix Computations**. Johns Hopkins University Press. https://jhupbooks.press.jhu.edu/title/matrix-computations.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, et al. 2014. "Generative Adversarial Nets." **Advances in Neural Information Processing Systems**, 2672–80. https://doi.org/10.5555/2969033.2969125.

Goyal, Priya, Piotr Dollár, Ross Girshick, et al. 2017. "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour." **ArXiv:1706.02677**. https://arxiv.org/abs/1706.02677.

Grathwohl, Will, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. 2019. "FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models." **International Conference on Learning Representations**.

Gretton, Arthur, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. "A Kernel Two-Sample Test." **Journal of Machine Learning Research** 13: 723–73.

Griewank, Andreas. 1992. "Achieving Logarithmic Growth of Temporal and Spatial Complexity in Reverse Automatic Differentiation." **Optimization Methods and Software** 1 (1): 35–54.

Griewank, Andreas, and Andrea Walther. 2008. **Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation**. 2nd ed. Society for Industrial; Applied Mathematics (SIAM). https://doi.org/10.1137/1.9780898717761.

Grobman, David M. 1959. “Homeomorphism of Systems of Differential Equations.” **Doklady Akademii Nauk SSSR** 128: 880–81.

Grünwald, Peter D. 2007. **The Minimum Description Length Principle**. MIT Press. https://doi.org/10.7551/mitpress/4643.001.0001.

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. “Improved Training of Wasserstein GANs.” **Advances in Neural Information Processing Systems** 30.

Gumbel, Emil J. 1954. **Statistical Theory of Extreme Values and Some Practical Applications**. U.S. National Bureau of Standards.

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. “On Calibration of Modern Neural Networks.” **International Conference on Machine Learning**, 1321–30. https://arxiv.org/abs/1706.04599.

Gupta, Suyog, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. “Deep Learning with Limited Numerical Precision.” **International Conference on Machine Learning**, 1737–46.

Gupta, Vineet, Tomer Koren, and Yoram Singer. 2018. “Shampoo: Preconditioned Stochastic Tensor Optimization.” **Proceedings of the 35th International Conference on Machine Learning (ICML)**, 1842–50.

Hairer, Ernst, Christian Lubich, and Gerhard Wanner. 2006. **Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations**. 2nd ed. Vol. 31. Springer Series in Computational Mathematics. Springer.

Hairer, Ernst, Syvert P. Nørsett, and Gerhard Wanner. 1993. **Solving Ordinary Differential Equations I: Nonstiff Problems**. 2nd ed. Springer.

Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp. 2011. “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions.” **SIAM Review** 53 (2): 217–88.

Hall, Peter. 1992. **The Bootstrap and Edgeworth Expansion**. Springer Series in Statistics. Springer. https://doi.org/10.1007/978-1-4612-4384-7.

Hanin, Boris, and Mihai Nica. 2020. “Products of Many Large Random Matrices and Gradients in Deep Neural Networks.” **Communications in Mathematical Physics** 376 (1): 287–322.

Hardt, Moritz, Ben Recht, and Yoram Singer. 2016. “Train Faster, Generalize Better: Stability of Stochastic Gradient Descent.” **Proceedings of the 33rd International Conference on Machine Learning**, Proceedings of machine learning research, vol. 48: 1225–34. https://proceedings.mlr.press/v48/hardt16.html.

Hartman, Philip. 1960. “A Lemma in the Theory of Structural Stability of Differential Equations.” **Proceedings of the American Mathematical Society** 11 (4): 610–20.

Hastings, W. Keith. 1970. “Monte Carlo Sampling Methods Using Markov Chains and Their Applications.” **Biometrika** 57 (1): 97–109.

Haveliwala, Taher H., and Sepandar D. Kamvar. 2003. **The Second Eigenvalue of the Google Matrix**. Stanford University. http://ilpubs.stanford.edu:8090/582/.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.” **Proceedings of the IEEE International Conference on Computer Vision**, 1026–34. https://doi.org/10.1109/iccv.2015.123.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. “Deep Residual Learning for Image Recognition.” **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, 770–78. https://doi.org/10.1109/cvpr.2016.90.

Held, Michael, Philip Wolfe, and Harlan P. Crowder. 1974. “Validation of Subgradient Optimization.” **Mathematical Programming** 6: 62–88.

Hertrich, Johannes, Antonin Chambolle, and Julie Delon. 2025. “On the Relation Between Rectified Flows and Optimal Transport.” **arXiv Preprint arXiv:2505.19712**.

Hestenes, Magnus R., and Eduard Stiefel. 1952. “Methods of Conjugate Gradients for Solving Linear Systems.” **Journal of Research of the National Bureau of Standards** 49 (6): 409–36.

Higham, Nicholas J. 2002. **Accuracy and Stability of Numerical Algorithms**. 2nd ed. SIAM.

Higham, Nicholas J., and Theo Mary. 2019. “A New Approach to Probabilistic Rounding Error Analysis.” **SIAM Journal on Scientific Computing** 41 (5): A2815–35. https://doi.org/10.1137/18M1226312.

Hill, Bruce M. 1975. “A Simple General Approach to Inference about the Tail of a Distribution.” **The Annals of Statistics** 3 (5): 1163–74. https://doi.org/10.1214/aos/1176343247.

Hinton, Geoffrey E., and Drew van Camp. 1993. “Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights.” **Proceedings of the Sixth Annual Conference on Computational Learning Theory**, 5–13.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. “Distilling the Knowledge in a Neural Network.” **arXiv Preprint arXiv:1503.02531**.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. “Denoising Diffusion Probabilistic Models.” **Advances in Neural Information Processing Systems** 33: 6840–51. https://arxiv.org/abs/2006.11239.

Ho, Jonathan, and Tim Salimans. 2022. “Classifier-Free Diffusion Guidance.” **arXiv Preprint arXiv:2207.12598**.

Hochbruck, Marlis, and Alexander Ostermann. 2010. “Exponential Integrators.” **Acta Numerica** 19: 209–86.

Hoeffding, Wassily. 1963. “Probability Inequalities for Sums of Bounded Random Variables.” **Journal of the American Statistical Association** 58 (301): 13–30.

Hoffman, Matthew D., and Andrew Gelman. 2014. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.” **Journal of Machine Learning Research** 15: 1593–623. https://jmlr.org/papers/v15/hoffman14a.html.

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, et al. 2022. “Training Compute-Optimal Large Language Models.” **Advances in Neural Information Processing Systems** 35. https://arxiv.org/abs/2203.15556.

Horn, Roger A., and Charles R. Johnson. 2012. **Matrix Analysis**. 2nd ed. Cambridge University Press.

Howard, Steven R., Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. 2021. “Time-Uniform, Nonparametric, Nonasymptotic Confidence Sequences.” **The Annals of Statistics** 49 (2): 1055–80. https://doi.org/10.1214/20-AOS1991.

Hu, Edward J., Yelong Shen, Phillip Wallis, et al. 2021. “LoRA: Low-Rank Adaptation of Large Language Models.” **arXiv Preprint arXiv:2106.09685**.

Hu, Shengding, Yuge Tu, Xu Han, et al. 2024. “MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies.” **ArXiv:2404.06395**. https://arxiv.org/abs/2404.06395.

Huffman, David A. 1952. “A Method for the Construction of Minimum-Redundancy Codes.” **Proceedings of the IRE** 40 (9): 1098–101. https://doi.org/10.1109/JRPROC.1952.273898.

Hutchinson, Michael F. 1989. “A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines.” **Communications in Statistics—Simulation and Computation** 18 (3): 1059–76.

Hyvärinen, Aapo. 2005. “Estimation of Non-Normalized Statistical Models by Score Matching.” **Journal of Machine Learning Research** 6: 695–709.

IEEE. 2019. **IEEE Standard for Floating-Point Arithmetic**. IEEE Std 754-2019.

Itô, Kiyosi. 1944. “Stochastic Integral.” **Proceedings of the Imperial Academy** 20 (8): 519–24.

Jaggi, Martin. 2013. “Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.” **Proceedings of the 30th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 28: 427–35.

James, W., and Charles Stein. 1961. “Estimation with Quadratic Loss.” **Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability** 1: 361–79.

Jang, Eric, Shixiang Gu, and Ben Poole. 2017. “Categorical Reparameterization with Gumbel-Softmax.” **International Conference on Learning Representations**.

Jaynes, Edwin T. 1957. “Information Theory and Statistical Mechanics.” **Physical Review** 106 (4): 620–30. https://doi.org/10.1103/PhysRev.106.620.

Jaynes, Edwin T. 1968. “Prior Probabilities.” **IEEE Transactions on Systems Science and Cybernetics** 4 (3): 227–41. https://doi.org/10.1109/TSSC.1968.300117.

Jelinek, Frederick, Robert L. Mercer, Lalit R. Bahl, and James K. Baker. 1977. “Perplexity—a Measure of the Difficulty of Speech Recognition Tasks.” **Journal of the Acoustical Society of America** 62 (S1): S63.

Johari, Ramesh, Pete Koomen, Leonid Pekelis, and David Walsh. 2017. “Peeking at A/B Tests: Why It Matters, and What to Do about It.” **Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, 1517–25. https://doi.org/10.1145/3097983.3097992.

Johnson, Rie, and Tong Zhang. 2013. “Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction.” **Advances in Neural Information Processing Systems** 26: 315–23.

Johnson, William B., and Joram Lindenstrauss. 1984. “Extensions of Lipschitz Mappings into a Hilbert Space.” **Contemporary Mathematics** 26: 189–206.

Jordan, Keller, Yuchen Jin, Vlado Boza, et al. 2024. **Muon: An Optimizer for Hidden Layers in Neural Networks**. Https://kellerjordan.github.io/posts/muon/.

Jordan, Richard, David Kinderlehrer, and Felix Otto. 1998. “The Variational Formulation of the Fokker–Planck Equation.” **SIAM Journal on Mathematical Analysis** 29 (1): 1–17.

Kahan, William. 1965. “Further Remarks on Reducing Truncation Errors.” **Communications of the ACM** 8 (1): 40.

Kaplan, Jared, Sam McCandlish, Tom Henighan, et al. 2020. “Scaling Laws for Neural Language Models.” **ArXiv:2001.08361**. https://arxiv.org/abs/2001.08361.

Karimi, Hamed, Julie Nutini, and Mark Schmidt. 2016. “Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak–Łojasiewicz Condition.” **Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)**, 795–811.

Karras, Tero, Miika Aittala, Timo Aila, and Samuli Laine. 2022. “Elucidating the Design Space of Diffusion-Based Generative Models.” **Advances in Neural Information Processing Systems** 35.

Karush, William. 1939. “Minima of Functions of Several Variables with Inequalities as Side Constraints.” Master’s thesis, Department of Mathematics, University of Chicago.

Kidger, Patrick. 2022. “On Neural Differential Equations.” **arXiv Preprint arXiv:2202.02435**.

Kidger, Patrick, James Morrill, James Foster, and Terry Lyons. 2020. “Neural Controlled

Differential Equations for Irregular Time Series." **Advances in Neural Information Processing Systems** 33: 6696–707.

Kingma, Diederik P, and Jimmy Ba. 2015. "Adam: A Method for Stochastic Optimization." **International Conference on Learning Representations**. https://arxiv.org/abs/1412.6980.

Kingma, Diederik P., Tim Salimans, Ben Poole, and Jonathan Ho. 2021. "Variational Diffusion Models." **Advances in Neural Information Processing Systems** 34.

Kingma, Diederik P., and Max Welling. 2014. "Auto-Encoding Variational Bayes." **International Conference on Learning Representations (ICLR)**. https://arxiv.org/abs/1312.6114.

Kloeden, Peter E., and Eckhard Platen. 1992. **Numerical Solution of Stochastic Differential Equations**. Springer.

Kolda, Tamara G., and Brett W. Bader. 2009. "Tensor Decompositions and Applications." **SIAM Review** 51 (3): 455–500.

Koller, Daphne, and Nir Friedman. 2009. **Probabilistic Graphical Models: Principles and Techniques**. MIT Press. https://doi.org/10.7551/mitpress/7432.001.0001.

Kolmogorov, Andrey N. 1931. "Über Die Analytischen Methoden in Der Wahrscheinlichkeitsrechnung." **Mathematische Annalen** 104: 415–58.

Koopman, Bernard O. 1936. "On Distributions Admitting a Sufficient Statistic." **Transactions of the American Mathematical Society** 39 (3): 399–409.

Kraskov, Alexander, Harald Stögbauer, and Peter Grassberger. 2004. "Estimating Mutual Information." **Physical Review E** 69 (6): 066138.

Kuhn, Harold W., and Albert W. Tucker. 1951. "Nonlinear Programming." **Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability**, 481–92.

Kullback, Solomon, and Richard A. Leibler. 1951. "On Information and Sufficiency." **Annals of Mathematical Statistics** 22 (1): 79–86.

Kynkäänniemi, Tuomas, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. 2024. "Applying Guidance in a Limited Interval Improves Sample and Distribution Quality in Diffusion Models." **Advances in Neural Information Processing Systems** 37. https://arxiv.org/abs/2404.07724.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles." **Advances in Neural Information Processing Systems** 30.

Laplace, Pierre-Simon. 1814. **Essai Philosophique Sur Les Probabilités**. Courcier.

Läuchli, Peter. 1961. "Jordan-Elimination Und Ausgleichung Nach Kleinsten Quadraten."

Numerische Mathematik 3 (1): 226–40.

LeCun, Yann, Leon Bottou, G Orr, and Klaus-Robert Muller. 1998. "Efficient Backprop." In **Neural Networks: Tricks of the Trade**. Springer. https://doi.org/10.1007/3-540-49430-8_2.

LeCun, Yann, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. 2006. "A Tutorial on Energy-Based Learning." In **Predicting Structured Data**. MIT Press.

Lessard, Laurent, Benjamin Recht, and Andrew Packard. 2016. "Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints." **SIAM Journal on Optimization** 26 (1): 57–95.

Levin, David A., and Yuval Peres. 2017. **Markov Chains and Mixing Times**. 2nd ed. American Mathematical Society.

Levy, Omer, and Yoav Goldberg. 2014. "Neural Word Embedding as Implicit Matrix Factorization." **Advances in Neural Information Processing Systems** 27.

Li, Ming, and Paul Vitányi. 2019. **An Introduction to Kolmogorov Complexity and Its Applications**. 4th ed. Texts in Computer Science. Springer. https://doi.org/10.1007/978-3-030-11298-1.

Li, Qianxiao, Cheng Tai, and Weinan E. 2017. "Stochastic Modified Equations and Adaptive Stochastic Gradient Algorithms." **Proceedings of the 34th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 70: 2101–10.

Li, Xuechen, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. 2020. "Scalable Gradients for Stochastic Differential Equations." **Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics**, Proceedings of machine learning research, vol. 108: 3870–82.

Li, Yujia, Kevin Swersky, and Richard Zemel. 2015. "Generative Moment Matching Networks." **Proceedings of the 32nd International Conference on Machine Learning**, 1718–27.

Lin, Jianhua. 1991. "Divergence Measures Based on the Shannon Entropy." **IEEE Transactions on Information Theory** 37 (1): 145–51.

Linnainmaa, Seppo. 1970. "The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors." Master's thesis, University of Helsinki.

Lipman, Yaron, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. "Flow Matching for Generative Modeling." **International Conference on Learning Representations**.

Liu, Chaoyue, Libin Zhu, and Mikhail Belkin. 2022. "Loss Landscapes and Optimization in over-Parameterized Non-Linear Systems and Neural Networks." **Applied and Computational Harmonic Analysis** 59: 85–116.

Liu, Dong C, and Jorge Nocedal. 1989. “On the Limited Memory BFGS Method for Large Scale Optimization.” **Mathematical Programming** 45 (1): 503–28. https://doi.org/10.1007/bf01589116.

Liu, Qiang. 2022. “Rectified Flow: A Marginal Preserving Approach to Optimal Transport.” **arXiv Preprint arXiv:2209.14577**.

Liu, Qiang, Jason D. Lee, and Michael I. Jordan. 2016. “A Kernelized Stein Discrepancy for Goodness-of-Fit Tests.” **Proceedings of the 33rd International Conference on Machine Learning**, 276–84.

Liu, Qiang, and Dilin Wang. 2016. “Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm.” **Advances in Neural Information Processing Systems** 29.

Liu, Shih-Yang, Chien-Yi Wang, Hongxu Yin, et al. 2024. “DoRA: Weight-Decomposed Low-Rank Adaptation.” **International Conference on Machine Learning**.

Liu, Xingchao, Chengyue Gong, and Qiang Liu. 2023. “Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow.” **International Conference on Learning Representations**.

Łojasiewicz, Stanisław. 1963. “Une Propriété Topologique Des Sous-Ensembles Analytiques réels.” In **Les Équations Aux dérivées Partielles**. Éditions du CNRS.

Loshchilov, Ilya, and Frank Hutter. 2016. “SGDR: Stochastic Gradient Descent with Warm Restarts.” **ArXiv:1608.03983**. https://arxiv.org/abs/1608.03983.

Loshchilov, Ilya, and Frank Hutter. 2019. “Decoupled Weight Decay Regularization.” **International Conference on Learning Representations**. https://arxiv.org/abs/1711.05101.

Lu, Cheng, Kaiwen Zheng, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. “Maximum Likelihood Training for Score-Based Diffusion ODEs by High-Order Denoising Score Matching.” **International Conference on Machine Learning**.

Lu, Cheng, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. “DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps.” **Advances in Neural Information Processing Systems** 35.

Ma, Siyuan, Raef Bassily, and Mikhail Belkin. 2018. “The Power of Interpolation: Understanding the Effectiveness of SGD in Modern over-Parametrized Learning.” **Proceedings of the 35th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 80: 3325–34.

MacKay, David J. C. 1992. “The Evidence Framework Applied to Classification Networks.” **Neural Computation** 4 (5): 720–36. https://doi.org/10.1162/neco.1992.4.5.720.

Maddison, Chris J., Andriy Mnih, and Yee Whye Teh. 2017. “The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables.” **International Conference on Learning Representations**.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. “Towards Deep Learning Models Resistant to Adversarial Attacks.” **International Conference on Learning Representations**. https://arxiv.org/abs/1706.06083.

Mahoney, Michael W., and Petros Drineas. 2009. “CUR Matrix Decompositions for Improved Data Analysis.” **Proceedings of the National Academy of Sciences** 106 (3): 697–702.

Mandt, Stephan, Matthew D. Hoffman, and David M. Blei. 2017. “Stochastic Gradient Descent as Approximate Bayesian Inference.” **Journal of Machine Learning Research** 18 (134): 1–35.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. **Introduction to Information Retrieval**. Cambridge University Press. https://doi.org/10.1017/CBO9780511809071.

Marchenko, Vladimir A., and Leonid A. Pastur. 1967. “Distribution of Eigenvalues for Some Sets of Random Matrices.” **Mathematics of the USSR-Sbornik** 1 (4): 457–83.

Maron, Melvin E. 1961. “Automatic Indexing: An Experimental Inquiry.” **Journal of the ACM** 8 (3): 404–17.

Martens, James. 2010. “Deep Learning via Hessian-Free Optimization.” **Proceedings of the 27th International Conference on Machine Learning**, 735–42.

Martens, James. 2020. “New Insights and Perspectives on the Natural Gradient Method.” **Journal of Machine Learning Research** 21 (146).

Martens, James, and Roger Grosse. 2015. “Optimizing Neural Networks with Kronecker-Factored Approximate Curvature.” **Proceedings of the 32nd International Conference on Machine Learning (ICML)**, 2408–17.

Martin, Charles H., and Michael W. Mahoney. 2021. “Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning.” **Journal of Machine Learning Research** 22 (165): 1–73.

Martins, André F. T., and Ramón F. Astudillo. 2016. “From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification.” **Proceedings of the 33rd International Conference on Machine Learning (ICML)**, 1614–23.

Maruyama, Gisiro. 1955. “Continuous Markov Processes and Stochastic Equations.” **Rendiconti Del Circolo Matematico Di Palermo** 4: 48–90.

Mazumder, Rahul, Trevor Hastie, and Robert Tibshirani. 2010. “Spectral Regularization Algorithms for Learning Large Incomplete Matrices.” **Journal of Machine Learning Research** 11 (80): 2287–322.

McAllester, David A. 1999. “Some PAC-Bayesian Theorems.” **Machine Learning** 37 (3): 355–63. https://doi.org/10.1023/A:1007618624809.

McAllester, David, and Karl Stratos. 2020. “Formal Limitations on the Measurement of Mutual Information.” **International Conference on Artificial Intelligence and Statistics**, 875–84.

McCandlish, Sam, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. 2018. “An Empirical Model of Large-Batch Training.” **arXiv Preprint arXiv:1812.06162**.

McCann, Robert J. 1997. “A Convexity Principle for Interacting Gases.” **Advances in Mathematics** 128 (1): 153–79.

McDiarmid, Colin. 1989. “On the Method of Bounded Differences.” In **Surveys in Combinatorics**. Cambridge University Press.

McFadden, Daniel. 1974. “Conditional Logit Analysis of Qualitative Choice Behavior.” In **Frontiers in Econometrics**, edited by Paul Zarembka. Academic Press.

McMillan, Brockway. 1956. “Two Inequalities Implied by Unique Decipherability.” **IRE Transactions on Information Theory** 2 (4): 115–16.

Meng, Fanxu, Zhaohui Wang, and Muhan Zhang. 2024. “PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models.” **Advances in Neural Information Processing Systems**.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. “Equation of State Calculations by Fast Computing Machines.” **The Journal of Chemical Physics** 21 (6): 1087–92.

Metropolis, Nicholas, and Stanislaw Ulam. 1949. “The Monte Carlo Method.” **Journal of the American Statistical Association** 44 (247): 335–41.

Miller, Evan. 2024. “Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations.” **arXiv Preprint arXiv:2411.00640**.

Milstein, Grigori N. 1975. “Approximate Integration of Stochastic Differential Equations.” **Theory of Probability and Its Applications** 19 (3): 557–62.

Mironov, Ilya. 2017. “Rényi Differential Privacy.” **2017 IEEE 30th Computer Security Foundations Symposium (CSF)**, 263–75.

Mirsky, Leon. 1960. “Symmetric Gauge Functions and Unitarily Invariant Norms.” **The Quarterly Journal of Mathematics** 11 (1): 50–59.

Miyato, Takeru, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. “Spectral Normalization for Generative Adversarial Networks.” **International Conference on Learning Representations**.

Mohamed, Shakir, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. 2020. “Monte Carlo Gradient Estimation in Machine Learning.” **Journal of Machine Learning Research** 21 (132): 1–62.

Moler, Cleve B. 1967. “Iterative Refinement in Floating Point.” **Journal of the ACM** 14 (2): 316–21.

Müller, Alfred. 1997. “Integral Probability Metrics and Their Generating Classes of Functions.” **Advances in Applied Probability** 29 (2): 429–43.

Müller, Rafael, Simon Kornblith, and Geoffrey Hinton. 2019. “When Does Label Smoothing Help?” **Advances in Neural Information Processing Systems** 32. https://arxiv.org/abs/1906.02629.

Naumann, Uwe. 2008. “Optimal Jacobian Accumulation Is NP-Complete.” **Mathematical Programming** 112 (2): 427–41.

Neal, Radford M. 2011. “MCMC Using Hamiltonian Dynamics.” In **Handbook of Markov Chain Monte Carlo**. Chapman; Hall/CRC. https://doi.org/10.1201/b10905-6.

Nemirovski, Arkadi. 2004. “Prox-Method with Rate of Convergence $O(1/t)$ for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems.” **SIAM Journal on Optimization** 15 (1): 229–51. https://doi.org/10.1137/S1052623403425629.

Nemirovski, Arkadi, and David Yudin. 1983. **Problem Complexity and Method Efficiency in Optimization**. Wiley.

Nesterov, Yu. 2018. **Lectures on Convex Optimization**. Springer. https://doi.org/10.1007/978-3-319-91578-4.

Nesterov, Yurii. 1983. “A Method of Solving a Convex Programming Problem with Convergence Rate $O(1/k^2)$.” **Soviet Mathematics Doklady** 27 (2): 372–76.

Nesterov, Yurii. 2005. “Smooth Minimization of Non-Smooth Functions.” **Mathematical Programming** 103 (1): 127–52. https://doi.org/10.1007/s10107-004-0552-5.

Nesterov, Yurii. 2012. “Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems.” **SIAM Journal on Optimization** 22 (2): 341–62.

Neyman, Jerzy. 1937. “Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability.” **Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences** 236 (767): 333–80. https://doi.org/10.2307/jj.8501421.24.

Ng, Andrew Y., and Michael I. Jordan. 2002. “On Discriminative Vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes.” **Advances in Neural Information Processing Systems** 14: 841–48.

Nguyen, XuanLong, Martin J. Wainwright, and Michael I. Jordan. 2010. “Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization.” **IEEE Transactions on Information Theory** 56 (11): 5847–61.

Nigam, Kamal, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000.

"Text Classification from Labeled and Unlabeled Documents Using EM." **Machine Learning** 39 (2–3): 103–34. https://doi.org/10.1023/A:1007692713085.

Nocedal, Jorge, and Stephen J. Wright. 2006. **Numerical Optimization**. 2nd ed. Springer.

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka. 2016. "F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization." **Advances in Neural Information Processing Systems** 29.

Øksendal, Bernt. 2003. **Stochastic Differential Equations: An Introduction with Applications**. 6th ed. Springer.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. 2018. "Representation Learning with Contrastive Predictive Coding." **arXiv Preprint arXiv:1807.03748**.

Oseledets, Ivan V. 2011. "Tensor-Train Decomposition." **SIAM Journal on Scientific Computing** 33 (5): 2295–317.

Owen, Art B. 2013. **Monte Carlo Theory, Methods and Examples**.

Paley, Raymond E. A. C., Norbert Wiener, and Antoni Zygmund. 1933. "Notes on Random Functions." **Mathematische Zeitschrift** 37: 647–68.

Papamakarios, George, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. "Normalizing Flows for Probabilistic Modeling and Inference." **Journal of Machine Learning Research** 22 (57): 1–64.

Parikh, Neal, and Stephen Boyd. 2014. "Proximal Algorithms." **Foundations and Trends in Optimization** 1 (3): 127–239. https://doi.org/10.1561/2400000003.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. 2013. "On the Difficulty of Training Recurrent Neural Networks." **International Conference on Machine Learning**, 1310–18.

Pearlmutter, Barak A. 1994. "Fast Exact Multiplication by the Hessian." **Neural Computation** 6 (1): 147–60. https://doi.org/10.1162/neco.1994.6.1.147.

Pennington, Jeffrey, Samuel Schoenholz, and Surya Ganguli. 2017. "Resurrecting the Sigmoid in Deep Learning Through Dynamical Isometry: Theory and Practice." **Advances in Neural Information Processing Systems**, 4785–95. https://proceedings.neurips.cc/paper/2017/hash/a9fc2d3b0c721b5b3f0b3f11bb24a75c-Abstract.html.

Peyré, Gabriel, and Marco Cuturi. 2019. "Computational Optimal Transport." **Foundations and Trends in Machine Learning** 11 (5–6): 355–607.

Pinsker, Mark S. 1964. **Information and Information Stability of Random Variables and Processes**. Holden-Day.

Pitman, Edwin J. G. 1936. "Sufficient Statistics and Intrinsic Accuracy." **Mathematical Proceedings of the Cambridge Philosophical Society** 32 (4): 567–79.

Planck, Max. 1917. “Über Einen Satz Der Statistischen Dynamik Und Seine Erweiterung in Der Quantentheorie.” **Sitzungsberichte Der Preussischen Akademie Der Wissenschaften**, 324–41.

Platt, John C. 1999. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.” In **Advances in Large Margin Classifiers**, edited by Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans. MIT Press.

Polyak, Boris T. 1964. “Some Methods of Speeding up the Convergence of Iteration Methods.” **USSR Computational Mathematics and Mathematical Physics** 4 (5): 1–17. https://doi.org/10.1016/0041-5553(64)90137-5.

Polyak, Boris T. 1963. “Gradient Methods for the Minimisation of Functionals.” **USSR Computational Mathematics and Mathematical Physics** 3 (4): 864–78.

Polyak, Boris T., and Anatoli B. Juditsky. 1992. “Acceleration of Stochastic Approximation by Averaging.” **SIAM Journal on Control and Optimization** 30 (4): 838–55.

Pontryagin, Lev S., Vladimir G. Boltyanskii, Revaz V. Gamkrelidze, and Evgenii F. Mishchenko. 1962. **The Mathematical Theory of Optimal Processes**. Interscience.

Pooladian, Aram-Alexandre, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. 2023. “Multisample Flow Matching: Straightening Flows with Minibatch Couplings.” **Proceedings of the 40th International Conference on Machine Learning**.

Poole, Ben, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. 2019. “On Variational Bounds of Mutual Information.” **Proceedings of the 36th International Conference on Machine Learning**, 5171–80.

Rademacher, Hans. 1919. “Über Partielle Und Totale Differenzierbarkeit von Funktionen Mehrerer Variabeln Und über Die Transformation Der Doppelintegrale.” **Mathematische Annalen** 79 (4): 340–59.

Radford, Alec, Jong Wook Kim, Chris Hallacy, et al. 2021. “Learning Transferable Visual Models from Natural Language Supervision.” **International Conference on Machine Learning**, 8748–63. https://proceedings.mlr.press/v139/radford21a.html.

Radhakrishna Rao, C. 1945. “Information and Accuracy Attainable in the Estimation of Statistical Parameters.” **Bulletin of the Calcutta Mathematical Society** 37 (3): 81–91. https://doi.org/10.1007/BF02908227.

Rahimi, Ali, and Benjamin Recht. 2007. “Random Features for Large-Scale Kernel Machines.” **Advances in Neural Information Processing Systems** 20.

Ramdas, Aaditya, Sashank J. Reddi, Barnabás Póczos, Aarti Singh, and Larry Wasserman. 2015. “On the Decreasing Power of Kernel and Distance Based Nonparametric Hypothesis Tests in High Dimensions.” **Proceedings of the AAAI Conference on Artificial Intelligence** 29.

Reddi, Sashank J, Satyen Kale, and Sanjiv Kumar. 2018. “On the Convergence of Adam and Beyond.” **International Conference on Learning Representations**. https://arxiv.org/abs/1904.09237.

Rényi, Alfréd. 1961. “On Measures of Entropy and Information.” **Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics**, 547–61.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. 2014. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models.” **International Conference on Machine Learning**, 1278–86. https://proceedings.mlr.press/v32/rezende14.html.

Rezende, Danilo, and Shakir Mohamed. 2015. “Variational Inference with Normalizing Flows.” **International Conference on Machine Learning**, 1530–38. https://proceedings.mlr.press/v37/rezende15.html.

Richardson, Tom, and Rüdiger Urbanke. 2008. **Modern Coding Theory**. Cambridge University Press. https://doi.org/10.1017/CBO9780511791338.

Risken, Hannes. 1996. **The Fokker–Planck Equation: Methods of Solution and Applications**. 2nd ed. Springer.

Rissanen, Jorma J. 1976. “Generalized Kraft Inequality and Arithmetic Coding.” **IBM Journal of Research and Development** 20 (3): 198–203.

Robbins, Herbert. 1956. “An Empirical Bayes Approach to Statistics.” **Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1**, 157–63.

Robbins, Herbert, and Sutton Monro. 1951. “A Stochastic Approximation Method.” **The Annals of Mathematical Statistics** 22 (3): 400–407.

Roberts, Gareth O., Andrew Gelman, and Walter R. Gilks. 1997. “Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms.” **The Annals of Applied Probability** 7 (1): 110–20. https://doi.org/10.1214/aoap/1034625254.

Roberts, Gareth O., and Richard L. Tweedie. 1996. “Exponential Convergence of Langevin Distributions and Their Discrete Approximations.” **Bernoulli** 2 (4): 341–63.

Rockafellar, R. T. 1970. **Convex Analysis**. Princeton University Press. https://doi.org/10.1515/9781400873173.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. “High-Resolution Image Synthesis with Latent Diffusion Models.” **IEEE/CVF Conference on Computer Vision and Pattern Recognition**, 10684–95.

Rubanova, Yulia, Ricky T. Q. Chen, and David Duvenaud. 2019. “Latent Ordinary Differential Equations for Irregularly-Sampled Time Series.” **Advances in Neural Information Processing Systems** 32.

Rudin, Walter. 1976. **Principles of Mathematical Analysis**. 3rd ed. McGraw-Hill.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. 1986. "Learning Representations by Back-Propagating Errors." **Nature** 323 (6088): 533–36. https://doi.org/10.1038/323533a0.

Sahami, Mehran, Susan Dumais, David Heckerman, and Eric Horvitz. 1998. "A Bayesian Approach to Filtering Junk e-Mail." **AAAI Workshop on Learning for Text Categorization**.

Samuelson, Paul A. 1965. "Rational Theory of Warrant Pricing." **Industrial Management Review** 6 (2): 13–31.

Särkkä, Simo, and Arno Solin. 2019. **Applied Stochastic Differential Equations**. Cambridge University Press.

Saxe, Andrew M., Yamini Bansal, Joel Dapello, et al. 2018. "On the Information Bottleneck Theory of Deep Learning." **International Conference on Learning Representations**.

Saxe, Andrew M., James L. McClelland, and Surya Ganguli. 2014. "Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks." **International Conference on Learning Representations**. https://arxiv.org/abs/1312.6120.

Schölkopf, Bernhard, and Alexander J Smola. 2002. **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**. MIT Press. https://doi.org/10.7551/mitpress/4175.001.0001.

Seitzer, Maximilian, Arash Tavakoli, Dimitrije Antic, and Georg Martius. 2022. "On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks." **International Conference on Learning Representations**. https://arxiv.org/abs/2203.09168.

Shannon, Claude E. 1959. "Coding Theorems for a Discrete Source with a Fidelity Criterion." **IRE National Convention Record** 7 (4): 142–63.

Shannon, Claude Elwood. 1948. "A Mathematical Theory of Communication." **The Bell System Technical Journal** 27 (3): 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Shannon, Claude Elwood. 1951. "Prediction and Entropy of Printed English." **The Bell System Technical Journal** 30 (1): 50–64. https://doi.org/10.1002/j.1538-7305.1951.tb01366.x.

Shwartz-Ziv, Ravid, and Naftali Tishby. 2017. "Opening the Black Box of Deep Neural Networks via Information." **arXiv Preprint arXiv:1703.00810**.

Simsekli, Umut, Levent Sagun, and Mert Gurbuzbalaban. 2019. "A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural Networks." **Proceedings of the 36th International Conference on Machine Learning**, Proceedings of machine learning research, vol. 97: 5827–37. https://proceedings.mlr.press/v97/simsekli19a.html.

Slater, Morton. 1950. **Lagrange Multipliers Revisited**. Discussion Paper Mathematics 403. Cowles Commission for Research in Economics.

Smith, Daniel G. A., and Johnnie Gray. 2018. "Opt_einsum: A Python Package for Optimizing Contraction Order for Einsum-Like Expressions." **Journal of Open Source Software** 3 (26): 753.

Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. "Deep Unsupervised Learning Using Nonequilibrium Thermodynamics." **International Conference on Machine Learning**, 2256–65. https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Song, Jiaming, and Stefano Ermon. 2020. "Understanding the Limitations of Variational Mutual Information Estimators." **International Conference on Learning Representations**.

Song, Jiaming, Chenlin Meng, and Stefano Ermon. 2021. "Denoising Diffusion Implicit Models." **International Conference on Learning Representations**.

Song, Yang, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. "Maximum Likelihood Training of Score-Based Diffusion Models." **Advances in Neural Information Processing Systems** 34.

Song, Yang, and Stefano Ermon. 2019. "Generative Modeling by Estimating Gradients of the Data Distribution." **Advances in Neural Information Processing Systems** 32. https://arxiv.org/abs/1907.05600.

Song, Yang, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. 2019. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation." **Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence**, Proceedings of machine learning research, vol. 115: 574–84. https://arxiv.org/abs/1905.07088.

Song, Yang, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. "Score-Based Generative Modeling Through Stochastic Differential Equations." **International Conference on Learning Representations**. https://arxiv.org/abs/2011.13456.

Soudry, Daniel, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. 2018. "The Implicit Bias of Gradient Descent on Separable Data." **Journal of Machine Learning Research** 19 (70): 1–57.

Squire, William, and George Trapp. 1998. "Using Complex Variables to Estimate Derivatives of Real Functions." **SIAM Review** 40 (1): 110–12. https://doi.org/10.1137/S0036144596301241X.

Stein, Charles. 1986. **Approximate Computation of Expectations**. Vol. 7. Institute of Mathematical Statistics Lecture Notes–Monograph Series. Institute of Mathematical Statistics.

Stein, Charles M. 1981. "Estimation of the Mean of a Multivariate Normal Distribution."

Annals of Statistics 9 (6): 1135–51.

Sterbenz, Pat H. 1974. **Floating-Point Computation**. Prentice-Hall.

Stratonovich, Ruslan L. 1966. "A New Representation for Stochastic Integrals and Equations." **SIAM Journal on Control** 4 (2): 362–71.

Student (Gosset, William S.). 1908. "The Probable Error of a Mean." **Biometrika** 6 (1): 1–25.

Sutskever, Ilya, James Martens, George Dahl, and Geoffrey Hinton. 2013. "On the Importance of Initialization and Momentum in Deep Learning." **International Conference on Machine Learning**, 1139–47. https://proceedings.mlr.press/v28/sutskever13.html.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. "Rethinking the Inception Architecture for Computer Vision." **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, 2818–26. https://doi.org/10.1109/cvpr.2016.308.

Székely, Gábor J., and Maria L. Rizzo. 2013. "Energy Statistics: A Class of Statistics Based on Distances." **Journal of Statistical Planning and Inference** 143 (8): 1249–72.

Tao, Terence, and Van Vu. 2010. "Random Matrices: Universality of ESDs and the Circular Law." **Annals of Probability** 38 (5): 2023–65.

Taylor, Adrien B., Julien M. Hendrickx, and François Glineur. 2018. "Exact Worst-Case Convergence Rates of the Proximal Gradient Method for Composite Convex Minimization." **Journal of Optimization Theory and Applications** 178 (2): 455–76.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." **Journal of the Royal Statistical Society: Series B** 58 (1): 267–88.

Tieleman, Tijmen, and Geoffrey Hinton. 2012. "Divide the Gradient by a Running Average of Its Recent Magnitude." In **COURSERA: Neural Networks for Machine Learning, Lecture 6.5-Rmsprop**. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

Tishby, Naftali, Fernando C. Pereira, and William Bialek. 1999. "The Information Bottleneck Method." **Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing**, 368–77.

Tishby, Naftali, and Noga Zaslavsky. 2015. "Deep Learning and the Information Bottleneck Principle." **IEEE Information Theory Workshop (ITW)**, 1–5. https://doi.org/10.1109/ITW.2015.7133169.

Tong, Alexander, Kilian Fatras, Nikolay Malkin, et al. 2024. "Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport." **Transactions on Machine Learning Research**.

Townsend, James, Thomas Bird, and David Barber. 2019. "Practical Lossless Compression with Latent Variables Using Bits Back Coding." **International Conference on Learning**

Representations. https://arxiv.org/abs/1901.04866.

Trefethen, Lloyd N., and David Bau. 1997. **Numerical Linear Algebra**. SIAM.

Trefethen, Lloyd N., and Mark Embree. 2005. **Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators**. Princeton University Press.

Tschannen, Michael, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. 2020. "On Mutual Information Maximization for Representation Learning." **International Conference on Learning Representations**.

Tseng, Paul. 2001. "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization." **Journal of Optimization Theory and Applications** 109 (3): 475–94.

Tucker, George, Andriy Mnih, Chris J. Maddison, John Lawson, and Jascha Sohl-Dickstein. 2017. "REBAR: Low-Variance, Unbiased Gradient Estimates for Discrete Latent Variable Models." **Advances in Neural Information Processing Systems** 30.

Uhlenbeck, George E., and Leonard S. Ornstein. 1930. "On the Theory of the Brownian Motion." **Physical Review** 36 (5): 823–41.

Vapnik, V., and A. Chervonenkis. 1971. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities." **Theory Probab. Appl.** 16 (2): 264–81. https://doi.org/10.1137/1116025.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, et al. 2017. "Attention Is All You Need." **Advances in Neural Information Processing Systems**, 5998–6008. https://doi.org/10.5555/3295222.3295349.

Vaswani, Sharan, Francis Bach, and Mark Schmidt. 2019. "Fast and Faster Convergence of SGD for over-Parameterized Models and an Accelerated Perceptron." **Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics**, Proceedings of machine learning research, vol. 89: 1195–204.

Veit, Andreas, Michael J Wilber, and Serge Belongie. 2016. "Residual Networks Behave Like Ensembles of Relatively Shallow Networks." **Advances in Neural Information Processing Systems**. https://arxiv.org/abs/1605.06431.

Vershynin, Roman. 2018. **High-Dimensional Probability: An Introduction with Applications in Data Science**. Cambridge University Press.

Vincent, Pascal. 2011. "A Connection Between Score Matching and Denoising Autoencoders." **Neural Computation** 23 (7): 1661–74.

Vovk, Vladimir, Alexander Gammerman, and Glenn Shafer. 2022. **Algorithmic Learning in a Random World**. 2nd ed. Springer. https://doi.org/10.1007/978-3-031-06649-8.

Wallace, Chris S. 1990. "Classification by Minimum-Message-Length Inference." **Advances in Computing and Information – ICCI '90**, Lecture notes in computer science, vol. 468: 72–81.

Wang, Feng, and Huaping Liu. 2021. “Understanding the Behaviour of Contrastive Loss.” **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**, 2495–504.

Wang, Sinong, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. “Linformer: Self-Attention with Linear Complexity.” **arXiv Preprint arXiv:2006.04768**.

Wasserman, Larry. 2013. **All of Statistics: A Concise Course in Statistical Inference**. Springer. https://link.springer.com/book/10.1007/978-0-387-21736-9.

Wasserstein, Ronald L, and Nicole A Lazar. 2016. “The ASA Statement on p-Values: Context, Process, and Purpose.” **The American Statistician** 70 (2): 129–33. https://doi.org/10.1080/00031305.2016.1154108.

Weed, Jonathan, and Francis Bach. 2019. “Sharp Asymptotic and Finite-Sample Rates of Convergence of Empirical Measures in Wasserstein Distance.” **Bernoulli** 25 (4A): 2620–48.

Welch, Lloyd R. 1974. “Lower Bounds on the Maximum Cross Correlation of Signals.” **IEEE Transactions on Information Theory** 20 (3): 397–99. https://doi.org/10.1109/TIT.1974.1055219.

Welford, B. P. 1962. “Note on a Method for Calculating Corrected Sums of Squares and Products.” **Technometrics** 4 (3): 419–20.

Welling, Max, and Yee W Teh. 2011. “Bayesian Learning via Stochastic Gradient Langevin Dynamics.” **Proceedings of the 28th International Conference on Machine Learning (ICML-11)**, 681–88. https://dl.acm.org/doi/10.5555/3104482.3104568.

Wengert, Robert Edwin. 1964. “A Simple Automatic Derivative Evaluation Program.” **Communications of the ACM** 7 (8): 463–64. https://doi.org/10.1145/355588.365726.

Wenliang, Li K., and Heishiro Kanagawa. 2020. “Blindness of Score-Based Methods to Isolated Components and Mixing Proportions.” **arXiv Preprint arXiv:2008.10087**. https://arxiv.org/abs/2008.10087.

White, Halbert. 1982. “Maximum Likelihood Estimation of Misspecified Models.” **Econometrica** 50 (1): 1–25.

White, Tom. 2016. “Sampling Generative Networks.” **arXiv Preprint arXiv:1609.04468**.

Wiener, Norbert. 1923. “Differential-Space.” **Journal of Mathematics and Physics** 2: 131–74.

Williams, Ronald J. 1992. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning.” **Machine Learning** 8: 229–56.

Witten, Ian H., Radford M. Neal, and John G. Cleary. 1987. “Arithmetic Coding for Data Compression.” **Communications of the ACM** 30 (6): 520–40.

Wong, Eugene, and Moshe Zakai. 1965. “On the Convergence of Ordinary Integrals to Stochastic Integrals.” **The Annals of Mathematical Statistics** 36 (5): 1560–64.

Woodruff, David P. 2014. “Sketching as a Tool for Numerical Linear Algebra.” **Foundations and Trends in Theoretical Computer Science** 10 (1–2): 1–157.

Wu, C. F. Jeff. 1983. “On the Convergence Properties of the EM Algorithm.” **Annals of Statistics** 11 (1): 95–103.

Wu Fung, Samy, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. 2022. “JFB: Jacobian-Free Backpropagation for Implicit Networks.” **Proceedings of the AAAI Conference on Artificial Intelligence** 36: 6648–56. https://doi.org/10.1609/aaai.v36i6.20619.

Xiao, Han, Kashif Rasul, and Roland Vollgraf. 2017. “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms.” **ArXiv:1708.07747**. https://arxiv.org/abs/1708.07747.

Xie, Shuo, and Zhiyuan Li. 2024. “Implicit Bias of AdamW: ℓ_∞ Norm Constrained Optimization.” **arXiv Preprint arXiv:2404.04454**.

Xu, Yilun, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. “A Theory of Usable Information Under Computational Constraints.” **International Conference on Learning Representations**.

Yin, Y. Q., Z. D. Bai, and P. R. Krishnaiah. 1988. “On the Limit of the Largest Eigenvalue of the Large Dimensional Sample Covariance Matrix.” **Probability Theory and Related Fields** 78 (4): 509–21.

Zadrozny, Bianca, and Charles Elkan. 2002. “Transforming Classifier Scores into Accurate Multiclass Probability Estimates.” **Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, 694–99. https://doi.org/10.1145/775047.775151.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. “Understanding Deep Learning (Still) Requires Rethinking Generalization.” **Communications of the ACM** 64 (3): 107–15. https://doi.org/10.1145/3446776.

Zhang, Jingzhao, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2020. “Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity.” **International Conference on Learning Representations**.

Zhang, Qinsheng, and Yongxin Chen. 2023. “Fast Sampling of Diffusion Models with Exponential Integrator.” **International Conference on Learning Representations**. https://arxiv.org/abs/2204.13902.

Zinkevich, Martin. 2003. “Online Convex Programming and Generalized Infinitesimal Gradient Ascent.” **Proceedings of the 20th International Conference on Machine Learning**, 928–36.

Ziv, Jacob, and Abraham Lempel. 1977. “A Universal Algorithm for Sequential Data Compression.” **IEEE Transactions on Information Theory** 23 (3): 337–43. https://doi.org/10.1109/TIT.1977.1055714.

Ziv, Jacob, and Abraham Lempel. 1978. “Compression of Individual Sequences via Variable-Rate Coding.” **IEEE Transactions on Information Theory** 24 (5): 530–36. https://doi.org/10.1109/TIT.1978.1055934.