# Learning Theory from First Principles

DRAFT

December 25, 2023

**Francis Bach**

francis.bach@inria.fr

# Contents

**Preface** ix

**I Preliminaries** 1

**1 Mathematical preliminaries** 3

1.1 Linear algebra and differentiable calculus . . . 3

1.1.1 Minimization of quadratic forms . . . 3

1.1.2 Inverting a $2 \times 2$ matrix . . . 4

1.1.3 Inverting matrices defined by blocks, matrix inversion lemma . . . 4

1.1.4 Eigenvalue and singular value decomposition . . . 6

1.1.5 Differential calculus . . . 7

1.2 Concentration inequalities . . . 7

1.2.1 Hoeffding's inequality . . . 9

1.2.2 McDiarmid's inequality . . . 12

1.2.3 Bernstein's inequality (♦) . . . 14

1.2.4 Expectation of the maximum . . . 16

1.2.5 Estimation of expectations through quadrature (♦) . . . 17

1.2.6 Concentration inequalities for matrices (♦♦) . . . 18

**2 Introduction to supervised learning** 21

2.1 From training data to predictions . . . 22

2.2 Decision theory . . . 25

2.2.1 Loss functions . . . 25

2.2.2 Risks . . . 26

2.2.3 Bayes risk and Bayes predictor . . . 27

2.3 Learning from data . . . 30

2.3.1 Local averaging . . . 30

2.3.2 Empirical risk minimization . . . 31

2.4 Statistical learning theory . . . 33

2.4.1 Measures of performance . . . 35

2.4.2 Notions of consistency over classes of problems . . . 35

2.5 No free lunch theorems (♦) . . . 36

2.6 Quest for adaptivity . . . . . 38
2.7 Beyond supervised learning . . . . . 39
2.8 Summary - book outline . . . . . 40

**3 Linear least-squares regression 43**
3.1 Introduction . . . . . 43
3.2 Least-squares framework . . . . . 44
3.3 Ordinary least-squares (OLS) estimator . . . . . 45
3.3.1 Closed-form solution . . . . . 45
3.3.2 Geometric interpretation . . . . . 46
3.3.3 Numerical resolution . . . . . 47
3.4 Statistical analysis of OLS . . . . . 47
3.5 Fixed design setting . . . . . 48
3.5.1 Statistical properties of the OLS estimator . . . . . 50
3.5.2 Experiments . . . . . 52
3.6 Ridge least-squares regression . . . . . 53
3.7 Lower-bound (♦) . . . . . 57
3.8 Random design analysis . . . . . 59
3.8.1 Gaussian designs . . . . . 61
3.8.2 General designs (♦♦) . . . . . 61
3.9 Principal component analysis (♦) . . . . . 63
3.10 Conclusion . . . . . 65

**II Generalization bounds for learning algorithms 67**

**4 Empirical risk minimization 69**
4.1 Convexification of the risk . . . . . 70
4.1.1 Convex surrogates . . . . . 71
4.1.2 Geometric interpretation of the support vector machine (♦) . . . . 72
4.1.3 Conditional $\Phi$-risk and classification calibration (♦) . . . . . 74
4.1.4 Relationship between risk and $\Phi$-risk (♦♦) . . . . . 76
4.2 Risk minimization decomposition . . . . . 80
4.3 Approximation error . . . . . 80
4.4 Estimation error . . . . . 81
4.4.1 Application of McDiarmid's inequality . . . . . 82
4.4.2 Easy case I: quadratic functions . . . . . 83
4.4.3 Easy case II: Finite number of models . . . . . 84
4.4.4 Beyond finitely many models through covering numbers (♦) . . . 85
4.5 Rademacher complexity . . . . . 86
4.5.1 Symmetrization . . . . . 87
4.5.2 Lipschitz-continuous losses . . . . . 89
4.5.3 Ball-constrained linear predictions . . . . . 91
4.5.4 Putting things together (linear predictions) . . . . . 92
4.5.5 From constrained to regularized estimation (♦) . . . . . 93

4.5.6 Extensions and improvements . . . 96
4.6 Model selection (♦) . . . 97
4.6.1 Structural risk minimization . . . 98
4.6.2 Selection based on validation set . . . 99
4.7 Relationship with asymptotic statistics (♦) . . . 99
4.8 Summary . . . 101

**5 Optimization for machine learning 103**
5.1 Optimization in machine learning . . . 103
5.2 Gradient descent . . . 105
5.2.1 Simplest analysis: ordinary least-squares . . . 106
5.2.2 Convex functions and their properties . . . 110
5.2.3 Analysis of GD for strongly convex and smooth functions . . . 112
5.2.4 Analysis of GD for convex and smooth functions (♦) . . . 117
5.2.5 Beyond gradient descent (♦) . . . 120
5.2.6 Non-convex objective functions (♦) . . . 122
5.3 Gradient methods on non-smooth problems . . . 123
5.4 Convergence rate of stochastic gradient descent (SGD) . . . 127
5.4.1 Strongly convex problems (♦) . . . 132
5.4.2 Adaptive methods (♦) . . . 134
5.4.3 Bias-variance trade-offs for least-squares (♦) . . . 135
5.4.4 Variance reduction (♦) . . . 138
5.5 Conclusion . . . 143

**6 Local averaging methods 145**
6.1 Introduction . . . 145
6.2 Local averaging methods . . . 147
6.2.1 Linear estimators . . . 147
6.2.2 Partition estimators . . . 148
6.2.3 Nearest-neighbors . . . 150
6.2.4 Nadaraya-Watson estimator a.k.a. kernel regression (♦) . . . 151
6.3 Generic "simplest" consistency analysis . . . 153
6.3.1 Fixed partition . . . 155
6.3.2 $k$-nearest neighbor . . . 158
6.3.3 Kernel regression (Nadaraya-Watson) (♦) . . . 160
6.4 Universal consistency (♦) . . . 163
6.5 Adaptivity (♦♦) . . . 166
6.6 Conclusion . . . 167

**7 Kernel methods 169**
7.1 Introduction . . . 170
7.2 Representer theorem . . . 170
7.3 Kernels . . . 173
7.3.1 Linear and polynomial kernels . . . 175
7.3.2 Translation-invariant kernels on $[0, 1]$ . . . 177

7.3.3 Translation-invariant kernels on $\mathbb{R}^d$ . . . 179
7.3.4 Beyond vectorial input spaces (♦) . . . 183
7.4 Algorithms . . . 184
7.4.1 Representer theorem . . . 184
7.4.2 Column sampling . . . 185
7.4.3 Random features . . . 186
7.4.4 Dual algorithms (♦) . . . 187
7.4.5 Stochastic gradient descent (♦) . . . 188
7.4.6 "Kernelization" of linear algorithms . . . 188
7.5 Generalization guarantees - Lipschitz-continuous losses . . . 189
7.5.1 Risk decomposition . . . 190
7.5.2 Approximation error for translation-invariant kernels on $\mathbb{R}^d$ . . . 192
7.6 Theoretical analysis of ridge regression (♦) . . . 195
7.6.1 Kernel ridge regression as a "linear" estimator . . . 195
7.6.2 Bias and variance decomposition (♦) . . . 196
7.6.3 Relating empirical and population covariance operators . . . 198
7.6.4 Analysis for well-specified problems (♦) . . . 200
7.6.5 Analysis beyond well-specified problems (♦) . . . 202
7.6.6 Balancing bias and variance (♦) . . . 202
7.7 Experiments . . . 204
7.8 Conclusion . . . 204

**8 Sparse methods 207**
8.1 Introduction . . . 207
8.1.1 Dedicated proof technique for constrained least-squares . . . 209
8.1.2 Probabilistic and combinatorial lemmas . . . 210
8.2 Variable selection by the $\ell_0$-penalty . . . 212
8.2.1 Assuming $k$ is known . . . 212
8.2.2 Estimating $k$ (♦) . . . 214
8.3 Variable selection by $\ell_1$-regularization . . . 216
8.3.1 Intuition and algorithms . . . 217
8.3.2 Slow rates - random design . . . 221
8.3.3 Slow rates - fixed design (square loss) . . . 221
8.3.4 Fast rates (♦) . . . 224
8.3.5 Zoo of conditions (♦♦) . . . 225
8.3.6 Random design (♦) . . . 227
8.4 Experiments . . . 228
8.5 Extensions . . . 229
8.6 Conclusion . . . 230

**9 Neural networks 233**
9.1 Introduction . . . 233
9.2 Single hidden layer neural network . . . 235
9.2.1 Optimization . . . 236
9.2.2 Rectified linear units and homogeneity . . . 237

9.2.3 Estimation error . . . . . 239
9.3 Approximation properties . . . . . 241
9.3.1 Universal approximation property in one dimension . . . . . 242
9.3.2 Infinitely many neurons and variation norm . . . . . 243
9.3.3 Variation norm in one dimension . . . . . 244
9.3.4 Variation norm in arbitrary dimension . . . . . 248
9.3.5 Precise approximation properties . . . . . 249
9.3.6 From the variation norm to a finite number of neurons (♦) . . . . . 250
9.4 Generalization performance for neural networks . . . . . 253
9.5 Relationship with kernel methods (♦) . . . . . 255
9.5.1 From a Banach space $\mathcal{F}_1$ to a Hilbert space $\mathcal{F}_2$ (♦) . . . . . 255
9.5.2 Kernel function (♦♦) . . . . . 257
9.5.3 Upper-bound on RKHS norm (♦♦) . . . . . 258
9.6 Experiments . . . . . 259
9.7 Extensions . . . . . 260
9.8 Conclusion . . . . . 261

## III Special topics 263

**10 Ensemble learning 265**
10.1 Averaging / bagging . . . . . 266
10.1.1 Independent datasets . . . . . 266
10.1.2 Bagging . . . . . 268
10.2 Random projections and averaging . . . . . 270
10.2.1 Gaussian sketching . . . . . 271
10.2.2 Random projections . . . . . 273
10.3 Boosting . . . . . 278
10.3.1 Problem set-up . . . . . 278
10.3.2 Incremental learning . . . . . 281
10.3.3 Matching pursuit . . . . . 282
10.3.4 Adaboost . . . . . 283
10.3.5 Greedy algorithm based on gradient boosting . . . . . 284
10.3.6 Convergence of expected risk . . . . . 287
10.3.7 Experiments . . . . . 290
10.4 Conclusion . . . . . 290

**11 From online learning to bandits 291**
11.1 First-order online convex optimization . . . . . 292
11.1.1 Convex case . . . . . 293
11.1.2 Strongly-convex case (♦) . . . . . 295
11.1.3 Online mirror descent (♦) . . . . . 295
11.1.4 Lower bounds (♦♦) . . . . . 297
11.2 Zero-th order convex optimization . . . . . 299
11.2.1 Smooth stochastic gradient descent . . . . . 301

11.2.2 Stochastic smoothing (♦) . . . 303
11.2.3 Extensions . . . 307
11.3 Multi-armed bandits . . . 307
11.3.1 Need for an exploration-exploitation trade-off . . . 308
11.3.2 “Explore-then-commit” . . . 309
11.3.3 Optimism in the face of uncertainty (♦) . . . 310
11.3.4 Adversarial bandits (♦) . . . 313
11.4 Conclusion . . . 315

**12 Over-parameterized models 317**
12.1 Implicit bias of gradient descent . . . 318
12.1.1 Least-squares . . . 318
12.1.2 Separable classification . . . 320
12.1.3 Beyond convex problems (♦) . . . 325
12.2 Double descent . . . 327
12.2.1 The double descent phenomenon . . . 327
12.2.2 Empirical evidence . . . 328
12.2.3 Linear regression with Gaussian projections (♦) . . . 329
12.3 Global convergence of gradient descent . . . 334
12.3.1 Mean field limits . . . 335
12.3.2 From linear networks to positive definite matrices . . . 339
12.3.3 Global convergence for positive definite matrices . . . 340
12.3.4 Special case of Oja flow . . . 342
12.4 Lazy regime and neural tangent kernels (♦) . . . 343
12.5 Conclusion . . . 345

**13 Structured prediction 347**
13.1 Multi-category classification . . . 348
13.1.1 Extension of classical convex surrogates . . . 348
13.1.2 Generalization bound I: stochastic gradient descent . . . 350
13.1.3 Generalization bound II: Rademacher complexities (♦) . . . 352
13.2 General set-up and examples . . . 354
13.2.1 Examples . . . 355
13.2.2 Structure encoding loss functions . . . 357
13.3 Surrogate methods . . . 358
13.3.1 Score functions and decoding step . . . 359
13.3.2 Fisher consistency and calibration functions . . . 359
13.3.3 Main surrogate frameworks . . . 360
13.4 Smooth/quadratic surrogates . . . 360
13.4.1 Quadratic surrogate . . . 360
13.4.2 Theoretical guarantees . . . 361
13.4.3 Linear estimators and decoding steps . . . 362
13.4.4 Smooth surrogates (♦) . . . 362
13.5 Max-margin formulations . . . 364
13.5.1 Structured SVM . . . 365

13.5.2 Max-min formulations (♦♦) . . . 365
13.6 Generalization bounds (♦) . . . 367
13.7 Experiments . . . 368
13.7.1 Robust regression . . . 368
13.7.2 Ranking . . . 369
13.8 Conclusion . . . 372

**14 Probabilistic methods** **373**
14.1 From empirical risks to log-likelihoods . . . 373
14.1.1 Conditional likelihoods . . . 375
14.1.2 Classical priors . . . 375
14.1.3 Sparse priors . . . 376
14.1.4 On the relationship between MAP and MMSE (♦) . . . 377
14.2 Discriminative vs. generative models . . . 380
14.2.1 Linear discriminant analysis and softmax regression . . . 381
14.2.2 Naive Bayes . . . 381
14.2.3 Maximum likelihood estimations . . . 382
14.3 Bayesian inference . . . 383
14.3.1 Computational handling of posterior distributions . . . 384
14.3.2 Model selection through marginal likelihood . . . 385
14.4 PAC-Bayesian analysis . . . 386
14.4.1 Set-up . . . 386
14.4.2 Uniformly bounded loss functions . . . 387
14.5 Conclusion . . . 389

**15 Lower bounds on performance** **391**
15.1 Statistical lower bounds . . . 392
15.1.1 Minimax lower bounds . . . 392
15.1.2 Reduction to a hypothesis test . . . 393
15.1.3 Review of information theory . . . 395
15.1.4 Lower-bound on hypothesis testing based on information theory . 397
15.1.5 Examples . . . 400
15.1.6 Minimax lower bounds through Bayesian analysis . . . 401
15.2 Optimization lower bounds . . . 404
15.2.1 Convex optimization . . . 404
15.2.2 Non-convex optimization (♦) . . . 406
15.3 Lower bounds for stochastic gradient descent (♦) . . . 409
15.4 Conclusion . . . 411

# Preface

**Why study learning theory?** Data have become ubiquitous in science, engineering, industry, and personal life, leading to the need for automated processing. Machine learning is concerned with making predictions from training examples and is used in all of these areas, in small and large problems, with a variety of learning models, from simple linear models to deep neural networks. It has now become an important part of the algorithmic toolbox.

How can we make sense of these practical successes? Can we extract a few principles to understand current learning methods and guide the design of new techniques for new applications or to adapt to new computational environments? This is precisely the goal of learning theory. Beyond being already mathematically rich and interesting (as it imports from many mathematical fields), most behaviors seen in practice can, in principle, be understood with sufficient effort and idealizations. In return, once understood, appropriate modifications can be made to obtain even stronger success.

**Why read this book?** The goal of this textbook is to present old and recent results in learning theory for the most widely used learning architectures. Doing so, a few principles are laid out to understand the overfitting and underfitting phenomena, as well as a systematic exposition of the three types of components in their analysis, estimation, approximation *and* optimization errors. Moreover, the goal is not only to show that learning methods can learn given sufficient amounts of data but to understand how fast (or slow) they learn, with a particular eye towards adaptivity to specific structures that make learning faster (such as smoothness of the prediction functions, or dependence on low-dimensional subspace).

This book is geared towards theory-oriented students as well as students who want to acquire a basic mathematical understanding of algorithms used throughout machine learning and associated fields that are significant users of learning methods such as computer vision or natural language processing. Moreover, it is well suited to students and researchers coming from other areas of applied mathematics or computer science who want to learn about the theory behind machine learning. Finally, since many simple proofs have been put together, it can serve as a reference for researchers in theoretical machine learning.

A particular effort will be made to prove **many results from first principles** while keeping the exposition as simple as possible. This will naturally lead to a choice of key results showcasing the essential concepts in learning theory in simple but relevant instances. A few general results will also be presented without proof. Of course, the concept of first principles is subjective, and I will assume a good knowledge of linear algebra, probability theory, and differential calculus.

Moreover, I will focus on the part of learning theory that deals with algorithms that can be run in practice, and thus, all algorithmic frameworks described in this book are routinely used. Since many modern learning methods are based on optimization, a chapter is dedicated to them. For most learning methods, some simple **illustrative experiments** are presented, with accompanying code (Matlab for the moment, Python underway, Julia in the future) so that students can see for themselves that the algorithms are simple and effective in synthetic experiments. Exercises currently come with no solutions.

Finally, the third part of the book provides an in-depth discussion of **moderne special topics** such as online learning, ensemble learning, structured prediction, and overparameterized models.

Note that this is *not* an introductory textbook on machine learning. There are already several good ones in several languages (see, e.g., Alpaydin, 2020; Lindholm et al., 2022; Azencott, 2019; Alpaydin, 2022). This textbook focuses on learning theory, that is, deriving mathematical guarantees for the most widely used learning algorithms, and characterizing what makes a particular algorithmic framework successful. In particular, given that many modern methods are based on optimization algorithms, we put a significant emphasis on gradient-based methods.

A key goal of the book is to look at the simplest results to make them most accessible to understand, rather than focusing on material that is more advanced but potentially too hard at first and that provides marginally better understanding. Throughout the book, we propose references to more modern work that goes deeper.

**Book organization.** The book comprises three main parts: introduction, core part, and special topics. Readers are encouraged to read the first two parts to understand the main concepts fully and can pick from the special topic chapters in a second reading or if used in a two-semester class.

All chapters start with a summary of the main concepts and results that will be covered. All simulations experiments are available at `https://www.di.ens.fr/~fbach/ltfp/` as Matlab code. Python code is currently being finalized.

Sections or more advanced exercises are denoted by ♦, ♦♦, or ♦♦♦. Comments or suggestions are most welcome and should be sent to `francis.bach@inria.fr`.

Many topics are not covered, and many more are not covered in depth. There are many good textbooks on learning theory that go deeper or wider (Christmann and Steinwart, 2008; Koltchinskii, 2011; Mohri et al., 2018; Shalev-Shwartz and Ben-David, 2014). See also the nice notes from Alexander Rakhlin and Karthik Sridharan,[1] as well as from Michael Wolf.[2]

In particular, the book focuses primarily on real-valued prediction functions, as it has become the de-facto standard for modern machine learning techniques, even when predicting discrete-valued outputs. Thus, although its historical importance and influence are crucial, I choose not to present the Vapnik-Chervonenkis dimension (see, e.g., Vapnik and Chervonenkis, 2015), and instead based our generic bounds on Rademacher complexities. This focus on real-valued prediction functions makes least-squares regression a central part of the theory, which is well appreciated by students. Moreover, this allows drawing links with the related statistical literature.

Some areas, such as online learning or probabilistic methods, are described in a single chapter to draw links with the classical theory and encourage readers to learn more about them through dedicated books. I have also included a chapter on over-parameterized models, which presents modern topics in machine learning. More generally, the goal in the third part of the book (special topics) was to cover most parts, which are a "few steps" away from the core material.

A book is always a work in progress. In particular, there are still typos and almost surely places where more details are needed; readers are most welcome to report them to me (and then get credit for it). I am convinced that more straightforward mathematical arguments are possible in many places in the book. Please let me know if you know of elegant and simple ideas I have overlooked.

**Mathematical notations.** Throughout the textbook, I will try to provide unified notations:

- Random variables: given a set $\mathcal{X}$, we will use the lower-case notation for a random variable with values in $\mathcal{X}$, as well as for its observations. Probability distributions will be denoted $\mu$ or $p$ and expectations as $\mathbb{E}[f(x)] = \int_{\mathcal{X}} f(x)dp(x)$. This is slightly ambiguous but will not cause major problems (and is standard in research papers). In this book, following most of the learning theory literature, we will gloss over measurability issues to avoid over-formalizations. For a detailed treatment, see Devroye et al. (1996); Christmann and Steinwart (2008).
- Norms on $\mathbb{R}^d$: we will consider the usual $\ell_p$-norms on $\mathbb{R}^d$, defined through $\|x\|_p^p = \sum_{i=1}^d |x_i|^p$ for $p \in [1, \infty)$, with $\|x\|_\infty = \max_{i\in\{1,\dots,d\}} |x_i|$.
- For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, $A \succcurlyeq 0$ means that $A$ is positive semi-definite (that is, all of its eigenvalues are non-negative), and for two symmetric matrices $A$ and $B$, $A \succcurlyeq B$ means that $A - B \succcurlyeq 0$.
- For a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, its gradient at $x$ is denoted $f'(x) \in \mathbb{R}^d$, and if it is twice differentiable, its Hessian is denoted $f''(x) \in \mathbb{R}^{d\times d}$.

[1] http://www.mit.edu/~rakhlin/courses/stat928/stat928_notes.pdf

[2] https://mediatum.ub.tum.de/doc/1723378/1723378.pdf

**How to use this book?** The first 9 chapters (in sequence, without the diamond parts) are adapted for a one-semester upper-undergraduate or graduate class, if possible, after an introductory course on machine learning. The following 6 chapters can be read mostly in any order and are here to deepen the understanding of some special topics; they can be read as homework assignments (using the exercises) or taught within a longer (e.g., 2-semester) class. The book is intended to be adapted to self-study, with the first 9 chapters being read in a sequence and the last 6 in a random order. In all situations, the first chapter on mathematical preliminaries can be read quickly and studied more in detail when relevant notions are needed in subsequent chapters.

**Acknowledgements.** This textbook is extracted from lecture notes from a class that I have taught (unfortunately online, but this gave me an opportunity to write more detailed notes) during the Fall 2020 semester, with an extra pass during the class I taught in the Spring 2021, Fall 2021, Fall 2022, and Fall 2023 semester.

These class notes have been adapted from the notes of many colleagues I had the pleasure to work with, in particular Lénaïc Chizat, Pierre Gaillard, Alessandro Rudi, and Simon Lacoste-Julien. Special thanks to Lénaïc Chizat for his help for the chapter on neural networks and for proof-reading many of the chapters, to Jaouad Mourtada for his help on lower bounds and random design analysis for least-squares regression, to Alex Nowak-Vila for his help on calibration functions, to Vivien Cabannes for the help on consistency proofs for local averaging techniques, to Alessandro Rudi for his help on kernel methods, to Adrien Taylor for his help on the optimization chapter, to Marc Lelarge for his help on over-parameterized models, and Olivier Cappé for his help on multi-arm bandits. The notes from Philippe Rigollet have also been a very precious help for the model selection chapter. The careful readings by Bertille Follain and Gabriel Stoltz have also been very helpful.

Typos have been found by Ritobrata Ghosh, Thanh Nguyen-Tang, Ishaan Gulrajani, Johannes Oswald, Seijin Kobayashi, Mathieu Dagreou, Dimitri Meunier, Antoine Moulin, Laurent Condat, Quentin Duchemin, Quentin Berthet, Mathieu Bloch, Fabien Pesquerel, Guillaume Bied, Uladzimir Yahorau, Pierre Dognin, Vihari Piratla, Tim Tsz-Kit Lau, Samy Clementz, Mohammad Alkousa, Eloïse Berthier, Pierre Marion, Vincent Liu, Atsushi Nitanda, Cheik Traoré, Ruiyuan Huang, Naoyuki Terashita, Jiangrui Kang, Moritz Haas, Mastane Achab, Berné Nortier, Cassidy Laidlaw, Jing Wang, Motonobu Kanagawa, Shane Hoeberichts, Dishank Jain, Aymeric Dieuleveut, Steffen Grünewälder, Claire Boyer, Bernhard Schölkopf, Piyushi Manupriya, Qingyue Zhao, Thomas Pock, Eliot Beyler, Yves Leconte, Jean Pichon, Brieuc Antoine Dit Urban, Théo Voldoire, Guénolé Joubioux, Adéchola Kouande, Zhu Wang (⚠ Add your name to the list!).

# Part I

# Preliminaries

# Chapter 1

# Mathematical preliminaries

**Chapter summary**

- Linear algebra: a bag of tricks to avoid lengthy and faulty computations.
- Concentration inequalities: for $n$ independent random variables, the deviation between the empirical average and the expectation is of order $O(1/\sqrt{n})$. What is in the big $O$, and how does it depend explicitly on problem parameters?

The mathematical analysis and design of machine learning algorithms require specialized tools beyond classic linear algebra, differential calculus, and probability. In this chapter, I will review these non-elementary mathematical tools used throughout the book: first, linear algebra tricks, then concentration inequalities. The chapter can be safely skipped since relevant results will be referenced when needed.

## 1.1 Linear algebra and differentiable calculus

This section reviews basic linear algebra and differential calculus results that will be used throughout the book. Using these may usually greatly simplify computations. Matrix notations will be used as much as possible.

### 1.1.1 Minimization of quadratic forms

Given a positive definite (and hence invertible) symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$, the minimization of quadratic forms with linear terms can be done in closed form as:

$$\inf_{x\in\mathbb{R}^n} \ \frac{1}{2}x^\top Ax - b^\top x = -\frac{1}{2}b^\top A^{-1}b,$$

with minimizer $x_* = A^{-1}b$ obtained by zeroing the gradient $f'(x) = Ax - b$ of the function $f(x) = \frac{1}{2}x^\top Ax - b^\top x$. Moreover, we have:

$$\frac{1}{2}x^\top Ax - b^\top x = \frac{1}{2}(x - x_*)^\top A(x - x_*) - \frac{1}{2}b^\top A^{-1}b.$$

If $A$ was not invertible (simply positive semi-definite) and $b$ was not in the column space of $A$, then the infimum would be $-\infty$.

Note that this result is often used in various forms, such as

$$b^\top x \leqslant \frac{1}{2}b^\top A^{-1}b + \frac{1}{2}x^\top Ax \text{ with equality if and only if } b = Ax.$$

This form is exactly the Fenchel-Young inequality[1] for quadratic forms (see Chapter 5), and is often used in one dimension in the form $ab \leqslant \frac{a^2}{2\eta} + \frac{\eta b^2}{2}$, for any $\eta \geqslant 0$ (and equality if and only if $\eta = a/b$).

### 1.1.2 Inverting a $2 \times 2$ matrix

Solving small systems happens frequently, as well as inverting small matrices. This can be easily done in two dimensions. Let $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ be a $2 \times 2$ matrix. If $ad - bc \neq 0$, then we may invert it as follows

$$M^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

This can be checked by multiplying the two matrices or using Cramer's rule,[2] and can be generalized to matrices defined by blocks, as we present next.

### 1.1.3 Inverting matrices defined by blocks, matrix inversion lemma

The example above may be generalized to matrices of the form $M = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$, with blocks of consistent sizes (note that $A$ and $D$ have to be square matrices). The inverse of $M$ may be obtained by applying directly Gaussian elimination[3] done in block form. Given the two matrices $M = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ and $N = \left(\begin{smallmatrix} I & 0 \\ 0 & I \end{smallmatrix}\right)$, we may linearly combine lines (with the same coefficients for the two matrices). Once $M$ has been transformed into the identity matrix, $N$ has been transformed to the inverse of $M$.

We make the simplifying assumption that $A$ is invertible; we use the notation $(M/A) = D - CA^{-1}B$ for the Schur complement of the block $A$ and also assume that $(M/A)$ is

[1] See https://en.wikipedia.org/wiki/Convex_conjugate.

[2] See https://en.wikipedia.org/wiki/Cramer's_rule.

[3] See https://en.wikipedia.org/wiki/Gaussian_elimination.

invertible. We thus get by Gaussian elimination, referring to $L_i$, $i = 1, 2$, as the two lines of blocks, so that for the first matrix $M = \left(\begin{smallmatrix} L_1 \\ L_2 \end{smallmatrix}\right)$:

$$\begin{array}{lll}
\text{Original matrices:} & \begin{pmatrix} A & B \\ C & D \end{pmatrix} & \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \\
L_2 \leftarrow L_2 - CA^{-1}L_1 : & \begin{pmatrix} A & B \\ 0 & (M/A) \end{pmatrix} & \begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix} \\
L_2 \leftarrow (M/A)^{-1}L_2 : & \begin{pmatrix} A & B \\ 0 & I \end{pmatrix} & \begin{pmatrix} I & 0 \\ -(M/A)^{-1}CA^{-1} & (M/A)^{-1} \end{pmatrix} \\
L_1 \leftarrow L_1 - BL_2 : & \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} & \begin{pmatrix} I + B(M/A)^{-1}CA^{-1} & -B(M/A)^{-1} \\ -(M/A)^{-1}CA^{-1} & (M/A)^{-1} \end{pmatrix} \\
L_1 \leftarrow A^{-1}L_1 : & \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} & \begin{pmatrix} A^{-1} + A^{-1}B(M/A)^{-1}CA^{-1} & -A^{-1}B(M/A)^{-1} \\ -(M/A)^{-1}CA^{-1} & (M/A)^{-1} \end{pmatrix}.
\end{array}$$

This shows that

$$M^{-1} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B(M/A)^{-1}CA^{-1} & -A^{-1}B(M/A)^{-1} \\ -(M/A)^{-1}CA^{-1} & (M/A)^{-1} \end{pmatrix}. \tag{1.1}$$

Moreover, by doing the same operations but by putting to zero first the upper-right block, and assuming $D$ and $(M/D) = A - BD^{-1}C$ are invertible, we obtain:

$$M^{-1} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} (M/D)^{-1} & -(M/D)^{-1}BD^{-1} \\ -D^{-1}C(M/D)^{-1} & D^{-1} + D^{-1}C(M/D)^{-1}BD^{-1} \end{pmatrix}. \tag{1.2}$$

By identifying the upper-left and lower-right blocks in Eq. (1.1) and Eq. (1.2), we obtain the identities (sometimes referred to as Woodbury matrix identities, or the *matrix inversion lemma*):

$$\begin{aligned}
\big(A - BD^{-1}C\big)^{-1} &= A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \\
\big(D - CA^{-1}B\big)^{-1} &= D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1}.
\end{aligned}$$

Another classical formulation is:

$$\big(A - BD^{-1}C\big)^{-1}B = A^{-1}B(D - CA^{-1}B)^{-1}D.$$

These are particularly interesting when the blocks $A$ and $D$ have very different sizes, as the inverse of a large matrix may be obtained from the inverse of a small matrix.

The lemma is often applied when $C = B^\top$, $A = I$ and $D = -I$, which leads to

$$(I + BB^\top)^{-1} = I - B(I + B^\top B)^{-1}B^\top,$$

and, once right-multiplied by $B$, this leads to the compact formula (which is easier to rederive and remember):

$$(I + BB^\top)^{-1}B = B(I + B^\top B)^{-1}.$$

These equalities are commonly used both for theoretical and algorithmic purposes.

**Exercise 1.1** *(♦) Show that we can "diagonalize" by blocks the matrices $M$ and $M^{-1}$ as:*

$$\begin{aligned} M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} &= \begin{pmatrix} I & 0 \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & (M/A) \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix} \\ M^{-1} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} &= \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & (M/A)^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix}. \end{aligned}$$

**Conditional covariance matrices for Gaussian vectors (♦).** The identities above can be used to compute conditional mean vectors and covariance matrices for Gaussian vectors (in this book, we will use the denominations "normal" and "Gaussian" interchangeably). If we have a Gaussian vector $\big(\begin{smallmatrix} x \\ y \end{smallmatrix}\big)$ with $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, with mean vector defined by block as $\mu = \big(\begin{smallmatrix} \mu_x \\ \mu_y \end{smallmatrix}\big)$, and covariance matrix $\Sigma = \big(\begin{smallmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{smallmatrix}\big) \succcurlyeq 0$ (defined with blocks of appropriate sizes), then the joint density $p(x, y)$ of $(x, y)$ is proportional to

$$\exp\bigg( -\frac{1}{2} \begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}^\top \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}^{-1} \begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix} \bigg).$$

By writing it as the product of a function of $x$ and of a function of $(x, y)$, we can get that $x$ is Gaussian with mean $\mu_x$ and covariance matrix $\Sigma_x$, and that given $x$, $y$ is Gaussian with mean $\mu_{y|x} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x - \mu_x)$ and covariance matrix $\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$.

**Exercise 1.2** *(♦) Prove the identities $\mu_{y|x} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x - \mu_x)$ and covariance matrix $\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$.*

### 1.1.4 Eigenvalue and singular value decomposition

In this book, we will often use eigenvalue decompositions of symmetric matrices. If $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix, there exists an orthogonal matrix $U \in \mathbb{R}^{n\times n}$ (that is, such that $U^\top U = UU^\top = I$), and a vector $\lambda \in \mathbb{R}^n$ of eigenvalues, such that $A = U \operatorname{Diag}(\lambda) U^\top$. If $u_i \in \mathbb{R}^n$ denotes the $i$-th column of $U$, then we have $A = \sum_{i=1}^n \lambda_i u_i u_i^\top$, and $Au_i = \lambda_i u_i$. A symmetric matrix is said to be positive semi-definite if and only if all its eigenvalues are non-negative.

Given a rectangular matrix $X \in \mathbb{R}^{n\times d}$, such that $n \geqslant d$, there exists an orthogonal matrix $V \in \mathbb{R}^{d\times d}$ (that is, such that $V^\top V = VV^\top = I$), a matrix $U \in \mathbb{R}^{n\times d}$ with orthonormal columns (that is, such that $U^\top U = I$), a vector $s \in \mathbb{R}_+^d$ of singular values, such that $X = U \operatorname{Diag}(s) V^\top$; this is often called the "economy-size" singular value decomposition (SVD) of the matrix $X$. If $u_i \in \mathbb{R}^n$ and $v_i \in \mathbb{R}^d$ denote the $i$-th columns of $U$ and $V$, then we have $X = \sum_{i=1}^d s_i u_i v_i^\top$, and $Xv_i = s_i u_i$, $X^\top u_i = s_i v_i$.

There are several ways of relating eigenvalues and singular values. For example, if $s_i$ is a singular value of $X$, then $s_i^2$ is an eigenvalue of $XX^\top$ and $X^\top X$. Moreover, the eigenvalues of the matrix $\big(\begin{smallmatrix} 0 & X \\ X^\top & 0 \end{smallmatrix}\big)$ are zero, the singular values of $X$, and their opposites.

For further properties of eigenvalues and singular values, see Golub and Loan (1996), Stewart and Sun (1990) and Bhatia (2013).

**Exercise 1.3** *Express the eigenvectors of $XX^\top$ and $X^\top X$ using the singular vectors of $X$.*

**Exercise 1.4** *Express the eigenvectors of $\left(\begin{smallmatrix} 0 & X \\ X^\top & 0 \end{smallmatrix}\right)$ using the singular vectors of $X$.*

### 1.1.5 Differential calculus

Throughout the book, we will compute gradients and Hessians of functions in almost all cases in matrix notations. Here are some classic examples:

- Quadratic forms: assuming $A = A^\top$, with $F(\theta) = \frac{1}{2}\theta^\top A\theta - b^\top\theta$, $F'(\theta) = A\theta - b$, $F''(\theta) = A$. If $A$ is not symmetric, then we have $F'(\theta) = \frac{1}{2}(A + A^\top)\theta$ and $F''(\theta) = \frac{1}{2}(A + A^\top)$.
- Least-squares with $X \in \mathbb{R}^{n\times d}$ and $y \in \mathbb{R}^n$: $F(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$. Then $F'(\theta) = \frac{1}{n}X^\top(X\theta - y)$ and $F''(\theta) = \frac{1}{n}X^\top X$.

**Exercise 1.5** *Show that for the logistic regression objective function defined as $F(\theta) = \frac{1}{n}\sum_{i=1}^n \log(1 + \exp(-y_i(X\theta)_i)$ with $X \in \mathbb{R}^{n\times d}$ and $y \in \{-1, 1\}^n$, then $F'(\theta) = \frac{1}{n}X^\top g$, where $g \in \mathbb{R}^n$ is defined as $g_i = -y_i\sigma(-y_i(X\theta)_i)$, with $\sigma(u) = (1 + e^{-u})^{-1}$ is the sigmoid function. Show that the Hessian is $\frac{1}{n}X^\top \mathrm{Diag}(h)X$, with $h \in \mathbb{R}^n$ defined as $h_i = \sigma(y_i(X\theta)_i)\sigma(-y_i(X\theta)_i)$.*

## 1.2 Concentration inequalities

All results presented in this textbook rely on the simple probabilistic assumption that data are independently and identically distributed (i.i.d.). The primary tool is then to relate empirical averages to expectations.

The key (very classical) insight behind probabilistic inequalities used in machine learning is that when you have $n$ *independent zero-mean* random variables, the natural "magnitude" of their average is $1/\sqrt{n}$ times smaller than their average magnitude. The simplest instance of this phenomenon is that if $Z_1, \dots, Z_n \in \mathbb{R}$ are independent and identically distributed with variance $\sigma^2 = \mathbb{E}(Z - \mathbb{E}[Z])^2$, then the variance of the sum is the sum of variances, and

$$\mathrm{var}\Big(\frac{1}{n}\sum_{i=1}^n Z_i\Big) = \frac{1}{n^2}\sum_{i=1}^n \mathrm{var}(Z_i) = \frac{\sigma^2}{n}.$$

Be careful with error measures or magnitudes: some are squared, some are not. Therefore, the $1/\sqrt{n}$ becomes $1/n$ after taking the square (this is trivial but typically leads to confusion).

The equality above can be interpreted as

$$\mathbb{E}\Big[\Big(\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbb{E}[Z]\Big)^2\Big] = \frac{\sigma^2}{n}, \tag{1.3}$$

which provides the simplest proof of the law of large numbers when variances exist and also highlights the convergence in squared mean of the random variable $\frac{1}{n}\sum_{i=1}^{n} Z_i$ to the constant $\mathbb{E}[Z]$.

**From moments to deviation bounds.** Given an (in)equality on the moments of a random variable, deviation bounds can be derived. Markov's inequality (see proof in Exercise 1.6 below) states that

$$\mathbb{P}(Y \geqslant \varepsilon) \leqslant \frac{1}{\varepsilon}\mathbb{E}[Y], \tag{1.4}$$

for all non-negative random variable $Y$ with finite expectation and any scalar $\varepsilon > 0$. Chebyshev's inequality is obtained by applying Markov's inequality to the random variable $Y = (X - \mathbb{E}[X])^2$ for a random variable $X$ with finite mean $\mathbb{E}[X]$ and variance $\mathrm{var}[X]$, leading to

$$\mathbb{P}(|X - \mathbb{E}[X]| \geqslant \varepsilon) = \mathbb{P}(|X - \mathbb{E}[X]|^2 \geqslant \varepsilon^2) \leqslant \frac{1}{\varepsilon^2}\,\mathrm{var}[X].$$

Thus, from the mean $\mathbb{E}[Z]$ and variance $\frac{\sigma^2}{n}$ of the random variable $\frac{1}{n}\sum_{i=1}^{n} Z_i$, computed in Eq. (1.3), we obtain the deviation bounds

$$\mathbb{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbb{E}[Z]\Big| \geqslant \varepsilon\Big) \leqslant \frac{1}{\varepsilon^2}\mathbb{E}\Big[\Big(\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbb{E}[Z]\Big)^2\Big] = \frac{\sigma^2}{n\varepsilon^2},$$

which implies convergence in probability.[4]

To characterize the deviations more finely, there are two classical tools: the *central limit theorem*, which states that $\frac{1}{n}\sum_{i=1}^{n} Z_i$ is approximately Gaussian with mean $\mathbb{E}[Z]$ and variance $\sigma^2/n$. This is an asymptotic statement: formally $\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbb{E}[Z])$ converges in distribution to a Gaussian distribution with mean zero and variance $\sigma^2$. Although it gives the correct scaling in $n$, in this textbook, we will look primarily at

[4] See https://en.wikipedia.org/wiki/Convergence_of_random_variables for convergences of random variables.

non-asymptotic results that quantify the deviation for any $n$.

⚠ In what follows, we will always provide versions of inequalities for *averages* of random variables (some authors equivalently consider sums).

Before describing various concentration inequalities, let us recall the classical *union bound*: given events indexed by $f \in \mathcal{F}$ (which can have a countably infinite number of elements), we have:

$$\mathbb{P}\Big( \bigcup_{f \in \mathcal{F}} A_f \Big) \leqslant \sum_{f \in \mathcal{F}} \mathbb{P}(A_f).$$

It has (among many other uses in machine learning) a direct application in upper-bounding the tail probability of the supremum of random variables:

$$\mathbb{P}\Big( \sup_{f \in \mathcal{F}} Z_f > t \Big) = \mathbb{P}\Big( \bigcup_{f \in \mathcal{F}} \{Z_f > t\} \Big) \leqslant \sum_{f \in \mathcal{F}} \mathbb{P}(Z_f > t).$$

We will only cover the most useful inequalities for machine learning. For more advanced inequalities, see, e.g., Boucheron et al. (2013); Vershynin (2018).

**Homogeneity.** ⚠ Random variables or vectors typically have a unit, and it is always helpful to perform some basic dimensional analysis[5] to spot mistakes. For example, when performing linear predictions of the form $y = x^\top \theta$, the unit of $y$ is the one of $x$ times that of $\theta$. Typically, these units are encapsulated in the constants describing the problem (such as the noise standard deviation for $y$ or bounds for $x$ and $\theta$).

**Exercise 1.6** *Let $Y$ be a non-negative random variable with finite expectation, and $\varepsilon > 0$. Show that $\varepsilon 1_{Y \geqslant \varepsilon} \leqslant Y$ almost surely and proof Markov's inequality in Eq. (1.4).*

**Exercise 1.7** *Let $Y$ be a non-negative random variable with finite expectation. Show that $\mathbb{E}[Y] = \int_0^\infty \mathbb{P}(Y \geqslant t)dt$.*

### 1.2.1 Hoeffding's inequality

The simplest concentration inequality considers bounded real-valued random variables.

**Proposition 1.1 (Hoeffding's inequality)** *If $Z_1, \dots, Z_n$ are independent random variables such that $Z_i \in [0,1]$ almost surely, then, for any $t \geqslant 0$,*

$$\mathbb{P}\Big( \frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Z_i] \geqslant t \Big) \leqslant \exp(-2nt^2). \qquad (1.5)$$

**Proof** The usual proof uses standard convexity arguments and is divided into two parts.

[5] https://en.wikipedia.org/wiki/Dimensional_analysis

(1) Lemma: If $Z \in [0,1]$ almost surely, then $\mathbb{E}\big[\exp(s(Z-\mathbb{E}[Z]))\big] \leqslant \exp(s^2/8)$ for any $s \geqslant 0$.

Proof: we can compute the first two derivatives of $\varphi : s \mapsto \log(\mathbb{E}\big[\exp(s(Z-\mathbb{E}[Z]))\big])$, which is a "log-sum-exp" function, often referred to as the "cumulant generating function", so that the second derivative is related to a certain variance. We can compute the derivatives of $\varphi$ as:

$$\begin{aligned}\varphi'(s) &= \frac{\mathbb{E}\big[(Z-\mathbb{E}[Z])e^{s(Z-\mathbb{E}[Z])}\big]}{\mathbb{E}\big[e^{s(Z-\mathbb{E}[Z])}\big]}\\ \varphi''(s) &= \frac{\mathbb{E}\big[(Z-\mathbb{E}[Z])^2e^{s(Z-\mathbb{E}[Z])}\big]}{\mathbb{E}\big[e^{s(Z-\mathbb{E}[Z])}\big]} - \bigg[\frac{\mathbb{E}\big[(Z-\mathbb{E}[Z])e^{s(Z-\mathbb{E}[Z])}\big]}{\mathbb{E}\big[e^{s(Z-\mathbb{E}[Z])}\big]}\bigg]^2.\end{aligned}$$

We thus get $\varphi(0) = \varphi'(0) = 0$, and $\varphi''(s)$ is the variance of some random variable $\tilde{Z} \in [0,1]$, with distribution with density $z \mapsto e^{s(z-\mathbb{E}[Z])}$ with respect to $\mu$, where $\mu$ is the distribution of $Z$. We recall that the variance of $\tilde{Z}$ is the minimum squared deviation to a constant and can thus bound this variance as

$$\operatorname{var}(\tilde{Z}) = \inf_{\nu\in[0,1]} \mathbb{E}[(\tilde{Z}-\nu)^2] \leqslant \mathbb{E}[(\tilde{Z}-1/2)^2] = \frac{1}{4}\mathbb{E}[(2\tilde{Z}-1)^2] \leqslant \frac{1}{4},$$

since $2\tilde{Z}-1 \in [-1,1]$ almost surely. Thus, for all $s \geqslant 0$, $\varphi''(s) \leqslant 1/4$, and by Taylor's formula, $\varphi(s) \leqslant \frac{s^2}{8}$.

(2) We recall Markov's inequality for any non-negative random variable $X$ and $a > 0$, which states $\mathbb{P}(X \geqslant a) \leqslant \frac{1}{a}\mathbb{E}[X]$. For any $t \geqslant 0$, and denoting $\bar{Z} = \frac{1}{n}\sum_{i=1}^n Z_i$:

$$\begin{aligned} & \mathbb{P}\big(\bar{Z}-\mathbb{E}[\bar{Z}] \geqslant t\big)\\ = \ & \mathbb{P}\big(\exp(s(\bar{Z}-\mathbb{E}[\bar{Z}])) \geqslant \exp(st)\big) \text{ by monotonicity of the exponential,}\\ \leqslant \ & \exp(-st)\mathbb{E}\big[\exp(s(\bar{Z}-\mathbb{E}[\bar{Z}]))\big] \text{ using Markov's inequality,}\\ \leqslant \ & \exp(-st)\prod_{i=1}^n \mathbb{E}\Big[\exp\Big(\frac{s}{n}(Z_i-\mathbb{E}[Z_i])\Big)\Big] \text{ by independence,}\\ \leqslant \ & \exp(-st)\prod_{i=1}^n \exp\Big(\frac{s^2}{8n^2}\Big) = \exp\Big(-st+\frac{s^2}{8n}\Big), \text{ using the lemma above,}\end{aligned}$$

which is minimized for $s = 4nt$. We then get the result.

■

Note the difference with the central limit theorem, which states that when $n$ goes to infinity, the probability in Eq. (1.5) is asymptotically equivalent to

$$\frac{1}{\sqrt{2\pi\sigma^2/n}}\int_t^\infty \exp\Big(-\frac{nz^2}{2\sigma^2}\Big)dz \text{ which can be shown to be less than } \exp\Big(-\frac{nt^2}{2\sigma^2}\Big),$$

where $\sigma^2 = \lim_{n\to+\infty}\frac{1}{n}\sum_{i=1}^n \operatorname{var}(Z_i)$. The central limit theorem is more precise (as it involves the variance of $Z_i$'s and not an almost sure bound) but is asymptotic. Bernstein inequalities (see Section 1.2.3) will be in between as they use both the variance and an almost sure bound.

**Extensions.** We get the following corollary by just applying the inequality to $Z_i$'s and $1 - Z_i$'s and using the union bound.

**Corollary 1.1 (Two-sided Hoeffding's inequality)** *If $Z_1, \dots, Z_n$ are independent random variables such that $Z_i \in [0,1]$ almost surely, then, for any $t \geqslant 0$,*

$$P\Big(\Big|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\Big| \geqslant t\Big) \leqslant 2\exp(-2nt^2). \tag{1.6}$$

We can make the following observations:

- Hoeffding's inequality can be extended to the assumption that $Z_i \in [a,b]$ almost surely, leading to

$$\mathbb{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\Big| \geqslant t\Big) \leqslant 2\exp(-2nt^2/(a-b)^2).$$

- Such an inequality is often used "in the other direction", starting from the probability and deriving $t$ from it as follows. For any $\delta \in (0,1)$, with probability greater than $1-\delta$, we have:

$$\Big|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\Big| < \frac{|a-b|}{\sqrt{2n}}\sqrt{\log\Big(\frac{2}{\delta}\Big)}.$$

  Note the dependence in $n$ as $1/\sqrt{n}$ and the logarithmic dependence in $\delta$ (corresponding to the exponential tail bound in $t$).

  **Exercise 1.8** *Show the one-sided inequality: with probability greater than $1-\delta$,* $\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i] < \frac{|a-b|}{\sqrt{2n}}\sqrt{\log\Big(\frac{1}{\delta}\Big)}$.

- When $Z_i \in [a_i, b_i]$ almost surely, with potentially different $a_i$'s and $b_i$'s, the probability upper-bound can be replaced by $2\exp(-2nt^2/c^2)$, where $c^2 = \frac{1}{n}\sum_{i=1}^n (b_i - a_i)^2$.

- The result extends to martingales with essentially the same proof, leading to Azuma's inequality (see the exercise below).

  **Exercise 1.9 (Azuma's inequality (♦))** *Assume that the sequence of random variables $(Z_i)_{i\geqslant 0}$, satisfies $\mathbb{E}(Z_i|\mathcal{F}_{i-1}) = 0$ for an increasing sequence of increasing "σ-fields" $(\mathcal{F}_i)_{i\geqslant 0}$.[6], and $|Z_i| \leqslant c_i$ almost surely, for $i \geqslant 1$. Then*

$$\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^n Z_i \geqslant t\Big) \leqslant \exp\Big(\frac{-n^2t^2}{2(c_1^2 + \cdots + c_n^2)}\Big).$$

[6] See more details in https://en.wikipedia.org/wiki/Azuma's_inequality.

- Hoeffding's inequality is often applied to so-called "sub-Gaussian" random variables, that is, random variables $X$ for which there exists $\tau \in \mathbb{R}_+$ such that the following bound on the Laplace transform of $X$ holds:

  $$\forall s \in \mathbb{R}, \ \mathbb{E}\big[\exp(s[X - \mathbb{E}[X])\big] \leqslant \exp(\frac{\tau^2 s^2}{2}),$$

  which is exactly what we used in the proof. In other words, a random variable with values in $[a, b]$ is sub-Gaussian with constant $\tau^2 = (b-a)^2/4$. For these sub-Gaussian variables, we have similar concentration inequalities (see next exercise). Moreover, for such sub-Gaussian random variables, we have the usual two versions of the tail bound:

  $$\forall t \geqslant 0, \ \mathbb{P}(|Z - \mathbb{E}[Z]| \geqslant t) \leqslant 2\exp\big(-\frac{t^2}{2\tau^2}\big) \tag{1.7}$$

  $$\Leftrightarrow \quad \forall \delta \in (0,1], \ |Z - \mathbb{E}[Z]| \leqslant \tau\sqrt{2\log\big(\frac{2}{\delta}\big)} \text{ with probability } 1-\delta. \tag{1.8}$$

  **Exercise 1.10** *Show that a Gaussian random variable with variance $\sigma^2$ is sub-Gaussian with constant $\sigma^2$.*

  **Exercise 1.11** *If $Z_1, \dots, Z_n$ are independent random variables which are sub-Gaussian with constant $\tau^2$, then, $P\big(\big|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\big| \geqslant t\big) \leqslant 2\exp(-\frac{nt^2}{2\tau^2})$ for any $t \geqslant 0$.*

- Sub-Gaussian random variables can be defined in several other ways, equivalent to constants with the bound on the Laplace transform. See the exercises below.

  **Exercise 1.12** *(♦) Let $Z$ be a random variable which is sub-Gaussian with constant $\tau^2$. Then, by using the tail bound $\mathbb{P}(|Z - \mathbb{E}[Z]| \geqslant t) \leqslant 2\exp(-\frac{t^2}{2\tau^2})$ in Eq. (1.7), show that for any positive integer $q$, $\mathbb{E}[(Z - \mathbb{E}[Z])^{2q}] \leqslant (2q)q!(2\tau^2)^q$.*

  **Exercise 1.13** *(♦♦) Let $Z$ be a random variable such that for any positive integer $q$, $\mathbb{E}[(Z - \mathbb{E}[Z])^{2q}] \leqslant (2q)q!(2\tau^2)^q$. Then show that $Z$ is sub-Gaussian with parameter $24\tau^2$.*

**Exercise 1.14** *Assume the random variable $Z$ has almost surely non-negative values and finite second-order moment. Show that for any $s \geqslant 0$, $\log\big(\mathbb{E}[e^{-sZ}]\big) \leqslant -s\mathbb{E}[Z] + \frac{s^2}{2}\mathbb{E}[Z^2]$.*

### 1.2.2 McDiarmid's inequality

Given $n$ independent random variables, it may be useful to concentrate other quantities than their average. What is needed is that the function of these random variables has "bounded variation".

**Proposition 1.2 (McDiarmid's inequality)** *Let $Z_1, \dots, Z_n$ be independent random variables (in any measurable space $\mathcal{Z}$), and $f : \mathcal{Z}^n \to \mathbb{R}$ a function of "bounded variation",* *that is, such that for all $i \in \{1, \ldots, n\}$, and all $z_1, \ldots, z_n, z_i' \in \mathcal{Z}$, we have*

$$|f(z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_n) - f(z_1, \ldots, z_{i-1}, z_i', z_{i+1}, \ldots, z_n)| \leqslant c.$$

*Then*

$$\mathbb{P}\big(\big|f(Z_1, \ldots, Z_n) - \mathbb{E}[f(Z_1, \ldots, Z_n)]\big| \geqslant t\big) \leqslant 2\exp(-2t^2/(nc^2)).$$

**Proof** (♦) The proof generalizes the formulation of Hoeffding's inequality in Eq. (1.6), which corresponds to $f(z) = \frac{1}{n}\sum_{i=1}^n z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]$ and $c = \frac{1}{n}$. We will only consider the one-sided inequality

$$\mathbb{P}\big(f(Z_1, \ldots, Z_n) - \mathbb{E}[f(Z_1, \ldots, Z_n)] \ \geqslant t\big) \leqslant \exp(-2t^2/(nc^2)),$$

which is sufficient to get the two-sided inequality using the union bound.

We introduce the random variables, for $i \in \{1, \ldots, n\}$:

$$V_i = \mathbb{E}[f(Z_1, \ldots, Z_n)|Z_1, \ldots, Z_i] - \mathbb{E}[f(Z_1, \ldots, Z_n)|Z_1, \ldots, Z_{i-1}],$$

with $V_1 = \mathbb{E}[f(Z_1, \ldots, Z_n)|Z_1] - \mathbb{E}[f(Z_1, \ldots, Z_n)]$. We have $\mathbb{E}[V_i|Z_1, \ldots, Z_{i-1}] = 0$. Moreover, given $Z_1, \ldots, Z_n$, the maximal value of $V_i$ minus the minimal value of $V_i$ is almost surely less than $c$ as a consequence of the bounded variation assumption since it is the difference of two terms that are conditional expectations of values of $f$ taken at arguments that only differ in the $i$-th variable. Moreover, through a telescoping sum, we have $f(Z_1, \ldots, Z_n) - \mathbb{E}[f(Z_1, \ldots, Z_n)] = \sum_{i=1}^n V_i$. Using the same argument as in part (1) of the proof of Hoeffding's inequality, we get for any $s > 0$, $\mathbb{E}\big(e^{sV_i}|Z_1, \ldots, Z_{i-1}\big) \leqslant e^{s^2c^2/8}$, and we can obtain a proof with the same steps as part (2) of Hoeffding's inequality by being careful with conditioning, for any $s \geqslant 0$:

$$\begin{aligned}
\mathbb{P}\Big(\sum_{i=1}^n V_i \geqslant t\Big) &\leqslant \exp(-st) \cdot \mathbb{E}\Big[\exp\Big(s\sum_{i=1}^n V_i\Big)\Big] \text{ using Markov's inequality,} \\
&= \exp(-st) \cdot \mathbb{E}\Big[\exp\Big(s\sum_{i=1}^{n-1} V_i\Big)\mathbb{E}\big[\exp(sV_n)\big|Z_1, \ldots, Z_{n-1}\big]\Big], \\
&\quad \text{since } V_1, \ldots, V_{n-1} \text{ are in the } \sigma\text{-algebra generated by } Z_1, \ldots, Z_{n-1}, \\
&\leqslant \exp(-st + s^2c^2/8) \cdot \mathbb{E}\Big[\exp\Big(s\sum_{i=1}^{n-1} V_i\Big)\Big],
\end{aligned}$$

using the bound above on $\mathbb{E}\big(e^{sV_n}|Z_1, \ldots, Z_{n-1}\big)$. Applying the same reasoning $n$ times, we get a probability less than $\exp(-st + ns^2c^2/8)$ and the desired result by minimizing with respect to $s$ (leading to $s = 4t/(nc^2)$). ■

This inequality will be used to provide high-probability bounds on the estimation error in empirical risk minimization in Section 4.4.1.

**Exercise 1.15** *(♦) Use McDiarmid's inequality to prove a Hoeffding-type bound for vectors, that is, if $Z_1, \dots, Z_n \in \mathbb{R}^d$ are independent centered vectors such that $\|Z_i\|_2 \leqslant c$ almost surely, then with probability greater than $1-\delta$, we have*

$$\Big\| \frac{1}{n}\sum_{i=1}^n Z_i \Big\|_2 \leqslant \frac{c}{\sqrt{n}}\Big(1+\sqrt{2\log\frac{1}{\delta}}\Big).$$

### 1.2.3 Bernstein's inequality (♦)

As mentioned earlier, Hoeffding's inequality only uses an almost sure bound, but not explicitly the variance, as the central limit theorem is using (but only with an asymptotic result). Bernstein's inequality allows the use of variance to get a finer non-asymptotic result.

**Proposition 1.3 (Bernstein's inequality)** *Let $Z_1, \dots, Z_n$ be $n$ independent random variables such that $|Z_i| \leqslant c$ almost surely and $\mathbb{E}[Z_i] = 0$. Then, for $t \geqslant 0$,*

$$\mathbb{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^n Z_i\Big| \geqslant t\Big) \leqslant 2\exp\Big(-\frac{nt^2}{2\sigma^2 + 2ct/3}\Big), \qquad (1.9)$$

*where $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \mathrm{var}(Z_i)$. Moreover, for $\delta \in (0,1)$, with probability greater than $1-\delta$, we have:*

$$\Big|\frac{1}{n}\sum_{i=1}^n Z_i\Big| \leqslant \sqrt{\frac{2\sigma^2\log(2/\delta)}{n}} + \frac{2c\log(2/\delta)}{3n}. \qquad (1.10)$$

**Proof** The proof is also divided into two parts, with first a lemma on the Laplace transform.

(a) Lemma: if $|Z| \leqslant c$ almost surely, $\mathbb{E}[Z] = 0$, and $\mathbb{E}[Z^2] = \sigma^2$, then for any $s > 0$, we have $\mathbb{E}[e^{sZ}] \leqslant \exp\big(\frac{\sigma^2}{c^2}(e^{sc} - 1 - sc)\big)$.

Proof: using the power series expansion of the exponential, we get:

$$\begin{aligned}\mathbb{E}[e^{sZ}] &= 1 + \mathbb{E}[sZ] + \sum_{k=2}^\infty \frac{s^k}{k!}\mathbb{E}[Z^k] = 1 + \sum_{k=2}^\infty \frac{s^k}{k!}\mathbb{E}[Z^k] \text{ because } Z \text{ has zero mean,}\\ &\leqslant 1 + \sum_{k=2}^\infty \frac{s^k}{k!}\mathbb{E}[|Z|^{k-2}|Z|^2] \leqslant 1 + \sum_{k=2}^\infty \frac{s^k}{k!}c^{k-2}\sigma^2 = 1 + \frac{\sigma^2}{c^2}\big(e^{sc} - 1 - sc\big).\end{aligned}$$

Using the bound $1 + \alpha \leqslant e^\alpha$ valid for all $\alpha \in \mathbb{R}$ leads to the desired result.

(b) With $\sigma_i^2 = \mathrm{var}(Z_i)$, we have the one-sided inequality:

$$\begin{aligned}
\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^n Z_i \geqslant t\Big) &= \mathbb{P}\Big(\exp\Big(s\sum_{i=1}^n Z_i\Big) \geqslant \exp(nst)\Big) \\
&\qquad\qquad\text{by monotonicity of the exponential,} \\
&\leqslant \mathbb{E}\Big[\exp\Big(s\sum_{i=1}^n Z_i\Big)\Big]e^{-nst} \text{ using Markov's inequality,} \\
&\leqslant e^{-nst}\prod_{i=1}^n \exp\Big(\frac{\sigma_i^2}{c^2}(e^{sc}-1-sc)\Big) = e^{-nst}\exp\Big(\frac{n\sigma^2}{c^2}(e^{sc}-1-sc)\Big),
\end{aligned}$$

using the lemma above. We now need to find an upper-bound on the minimal value (with respect to $s$) of $-nst + \frac{n\sigma^2}{c^2}(e^{sc}-1-sc) = \frac{n\sigma^2}{c^2}\big(e^{sc}-1-sc-\alpha sc\big)$, with $\alpha = ct/\sigma^2$. We first bound for $u = sc$, $e^u - 1 - u = \sum_{k=0}^\infty \frac{u^{k+2}}{(k+2)!} \leqslant \sum_{k=0}^\infty \frac{u^{k+2}}{2\cdot 3^k}$, since $(k+2)! = 2\cdot 3\cdots(k+2) \geqslant 2\cdot 3^k$. Thus, for $u \in (0,3)$, we get

$$e^u - 1 - u \leqslant \frac{u^2}{2}\sum_{k=0}^\infty (u/3)^k = \frac{u^2}{2}\frac{1}{1-u/3}.$$

Using the candidate $u = \frac{\alpha}{1+\alpha/3}$, we get $1 - u/3 = \frac{3}{\alpha+3}$, and thus:

$$e^u - 1 - u - \alpha u \leqslant \frac{u^2}{2}\frac{1}{1-u/3} - \alpha u = \frac{\alpha^2}{2(1+\alpha/3)^2}\frac{\alpha+3}{3} - \frac{\alpha^2}{1+\alpha/3} = -\frac{\alpha^2}{2(1+\alpha/3)}.$$

This exactly leads to the one-sided version of Eq. (1.9).

To get Eq. (1.10) from the two sided-version of Eq. (1.9), we solve in $t$ the equation $2\exp\big(\frac{-nt^2}{2\sigma^2+2ct/3}\big) = \delta \Leftrightarrow \log\frac{2}{\delta} = \frac{nt^2}{2\sigma^2+2ct/3}$. Solving the quadratic equation in $t$ leads to (using $(a+b)^{1/2} \leqslant a^{1/2} + b^{1/2}$):

$$t = \frac{1}{2}\Big[\frac{2c}{3n}\log\frac{2}{\delta} + \Big(\Big(\frac{2c}{3n}\log\frac{2}{\delta}\Big)^2 + \frac{8\sigma^2}{n}\log\frac{2}{\delta}\Big)^{1/2}\Big] \leqslant \frac{2c}{3n}\log\frac{2}{\delta} + \frac{1}{2}\Big(\frac{8\sigma^2}{n}\log\frac{2}{\delta}\Big)^{1/2},$$

which leads to Eq. (1.10).

■

Note here that we get the same dependence as for the central limit theorem for small deviations $t$ (and a strict improvement on Hoeffding because the variance is essentially bounded by the squared diameter of the support). In contrast, for large $t$, the dependence in $t$ is worse than Hoeffding's inequality.

**Beyond zero mean random variables.** Bernstein's inequality can also be applied when the random variables $Z_i$ do not have zero means. Then Eq. (1.9) is replaced by

$$\mathbb{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\Big| \geqslant t\Big) \leqslant 2\exp\Big(-\frac{nt^2}{2\sigma^2 + 2ct/3}\Big).$$

.

**Exercise 1.16** *(♦) Prove the inequality above.*

### 1.2.4 Expectation of the maximum

Concentration inequalities bound the deviation from the expectation. Often, computing the expectation is tricky, particularly for maxima of random variables. In a nutshell, taking the maximum of $n$ bounded random variables leads to an extra factor of $\sqrt{\log n}$. Note here that we do not impose independence. We will consider other tools such as Rademacher complexities in Section 4.5. See Figure 1.1 for an illustration.

⚠ This logarithmic factor appears many times in this textbook and can often be traced back to the expectation of a maximum and to the Gaussian decay of tail bounds.

⚠ The variables do not need to be independent.

**Proposition 1.4 (Expectation of the maximum)** *If $Z_1, \dots, Z_n$ are (potentially dependent) zero-mean real random variables which are sub-Gaussian with constant $\tau^2$, then*

$$\mathbb{E}[\max\{Z_1, \dots, Z_n\}] \leqslant \sqrt{2\tau^2 \log n}.$$

**Proof** We have:

$$\begin{aligned}
\mathbb{E}[\max\{Z_1, \dots, Z_n\}] &\leqslant \frac{1}{t} \log \mathbb{E}[e^{t \max\{Z_1, \dots, Z_n\}}] \text{ by Jensen's inequality,} \\
&= \frac{1}{t} \log \mathbb{E}[\max\{e^{tZ_1}, \dots, e^{tZ_n}\}] \\
&\leqslant \frac{1}{t} \log \mathbb{E}[e^{tZ_1} + \cdots + e^{tZ_n}] \text{ bounding the max by the sum,} \\
&\leqslant \frac{1}{t} \log(n e^{\tau^2 t^2/2}) = \frac{\log n}{t} + \tau^2 \frac{t}{2} = \sqrt{2\tau^2 \log n} \text{ with } t = \tau^{-1}\sqrt{2 \log n},
\end{aligned}$$

using the definition of sub-Gaussianity in Section 1.2.1 (and the fact that the variables have zero means). ■

While we consider a direct proof using Laplace transforms above, we can prove a similar result using Gaussian tail bounds together with the union bound

$$\mathbb{P}(\max\{U_1, \dots, U_n\} \geqslant t) \leqslant \mathbb{P}(U_1 \geqslant t) + \cdots + \mathbb{P}(U_n \geqslant t),$$

for well-chosen random variables $U_1, \dots, U_n$. In other words, the dependence in the probability $\delta$ as $\sqrt{\log(\frac{2}{\delta})}$ in Eq. (1.8) is directly related to the term $\sqrt{\log n}$ above (see exercise below). We will see a different dependence in $n$ in Section 8.1.2 for the maximum of squared norms of Gaussians.

**Exercise 1.17** *Assume $Z_1, \dots, Z_n$ are random variables that are sub-Gaussian with constant $\tau^2$ and have zero means. Show that $\mathbb{E}[\max\{|Z_1|, \dots, |Z_n|\}] \leqslant \sqrt{2\tau^2 \log(2n)}$. Prove*

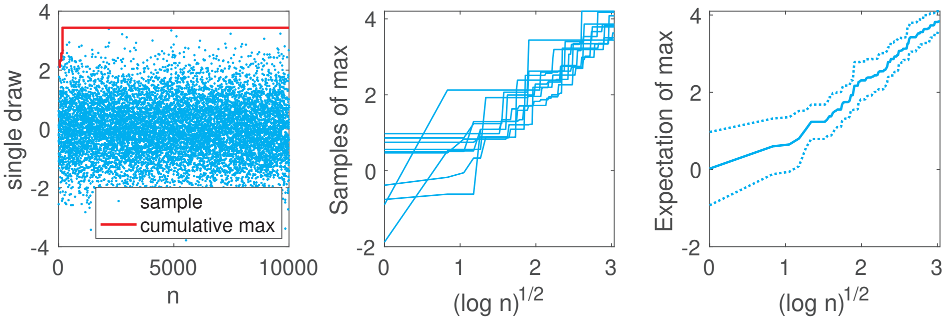


Figure 1.1: Expectation of the maximum of $n$ independent standard Gaussian randoms variables. Left: illustration of the cumulative maximum $\max\{Z_1, \dots, Z_n\}$. Middle: 10 samples of the cumulative maximum as a function of $\sqrt{\log n}$. Right: mean and standard deviations from 1000 replications. Notice the linear growth in $\sqrt{\log n}$ compatible with our bounds.

*the same result up to a universal constant using the tail bounds* $\mathbb{P}(|Z_i| \geqslant t) \leqslant 2\exp(-\frac{t^2}{2\tau^2})$ *together with the union bound, and the property* $\mathbb{E}[|Y|] = \int_0^{+\infty} \mathbb{P}(|Y| \geqslant t)dt$ *for any random variable* $Y$ *such that* $\mathbb{E}[|Y|]$ *exists.*

**Exercise 1.18** *(♦♦) Assume* $Z_1, \dots, Z_n$ *are independent Gaussian random variables with mean zero and variance* $\sigma^2$*. Provide a lower bound for* $\mathbb{E}[\max\{Z_1, \dots, Z_n\}]$ *of the form* $c\sqrt{\log n}$ *for* $c > 0$.

**Exercise 1.19** *(♦♦) We consider a convex function* $f : \mathbb{R}^d \to \mathbb{R}$ *such that* $f(0) = 0$ *and* $f$ *is* $L$*-smooth with respect to the norm* $\Omega$*, that is,* $f$ *is continuously differentiable and for all* $\theta, \eta \in \mathbb{R}^d$, $f(\theta) \leqslant f(\eta) + f'(\eta)^\top(\theta - \eta) + \frac{L}{2}\Omega(\theta - \eta)^2$*. Let* $Z_i \in \mathbb{R}^d$ *be independent zero-mean random vectors with* $\mathbb{E}[\Omega(Z_i)^2] \leqslant \sigma^2$*, for* $i = 1, \dots, n$*. Show by induction in* $n$ *that* $\mathbb{E}[f(Z_1 + \cdots + Z_n)] \leqslant nL\frac{\sigma^2}{2}$.

**Exercise 1.20** *Assume* $Z_1, \dots, Z_n$ *are sub-Gaussian random variables with common sub-Gaussianity parameter* $\tau$*, and potentially different means* $\mu_1, \dots, \mu_n$*. For a fixed set of non-negative weights* $\pi_1, \dots, \pi_n$ *that sum to one, and* $\delta \in (0, 1)$*, show that with probability greater than* $1 - \delta$*, for all* $i \in \{1, \dots, n\}$, $|z_i - \mu_i| \leqslant \tau\sqrt{2\log(1/\pi_i)} + \tau\sqrt{2\log(2/\delta)}$*. If* $\hat{i} \in \arg\min_{i \in \{1,\dots,n\}} \big\{z_i + \tau\sqrt{2\log(1/\pi_i)}\big\}$*, show that with probability greater than* $1 - \delta$, $\mu_{\hat{i}} \leqslant \min_{i \in \{1,\dots,n\}} \big\{\mu_i + \tau\sqrt{2\log(1/\pi_i)}\big\} + 2\tau\sqrt{2\log(2/\delta)}$.

### 1.2.5 Estimation of expectations through quadrature (♦)

In machine learning, the generalization error is an expectation of a function (the loss associated with a specific prediction function) of a random variable (the pair input/output). This generalization error is naturally approximated by an empirical average given some independent and identically distributed (i.i.d.) samples, with a convergence rate of $O(1/\sqrt{n})$ from $n$ samples (as shown, for example, from Hoeffding's inequality).

In this section, we briefly present *quadrature* methods whose aim is to estimate the same expectation but with potentially non-random observations. For simplicity, we consider a random variable $X$ uniformly distributed in $[0,1]$, and the task of computing the expectation of a function $f : [0,1] \to \mathbb{R}$, that is, $I = \mathbb{E}[f(X)] = \int_0^1 f(x)dx$, noting that there are many variants of such methods (see, e.g., Davis and Rabinowitz, 1984; Brass and Petras, 2011), and that these techniques extend to higher dimensions (Holtz, 2010). Moreover, while we focus on equally spaced data in the interval, "quasi-random" methods lead to better convergence rates (Niederreiter, 1992).

We consider uniformly spaced grid points on $[0,1]$, as it can serve as an idealization of random sampling when studying regression models, in particular in Chapter 6 and Chapter 7. That is, we consider $x_i = \frac{i}{n}$ for $i \in \{0,\dots,n\}$ (with $n+1$ points). The classical trapezoidal rule considers the approximation

$$\hat{I} = \frac{1}{n}\Big[\frac{1}{2}f(x_0) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}f(x_n)\Big].$$

The error $|I - \hat{I}|$ then depends on the regularity of $f$. We have a decomposition of the error as the integral between $f$ and its piecewise affine interpolant:

$$\begin{aligned} I - \hat{I} &= \sum_{i=1}^{n}\Big(\int_{x_{i-1}}^{x_i} f(x)dx - \frac{x_i - x_{i-1}}{2}\big[f(x_i) + f(x_{i-1})\big]\Big) \\ &= \sum_{i=1}^{n}\Big(\int_{x_{i-1}}^{x_i} f(x)dx - \int_{x_{i-1}}^{x_i}\Big\{\frac{x_i - x}{x_i - x_{i-1}} f(x_{i-1}) + \frac{x - x_{i-1}}{x_i - x_{i-1}} f(x_i)\Big\}dx\Big). \end{aligned}$$

If $f$ is twice differentiable and has a second-derivative bounded by $L$ uniformly in absolute value, then we have the bound (which can be obtained by Taylor's formula):

$$|I - \hat{I}| \leqslant \sum_{i=1}^{n} \frac{L}{2}\int_{x_{i-1}}^{x_i} (x_i - x)(x - x_{i-1})dx = \sum_{i=1}^{n} \frac{L}{12}(x_i - x_{i-1})^3 dx = \frac{L}{12n^2}.$$

We thus have an error bound in $O(1/n^2)$ if we assume two bounded derivatives. We typically get an error of $O(1/n^s)$ for such numerical integration methods if we assume $s$ bounded derivatives (with the appropriate rule, such as Simpson's rule, which makes a piecewise quadratic interpolation). See the exercises below.

**Exercise 1.21** *Show that the trapezoidal rule leads to an error in $O(1/n)$ if we only assume one bounded derivative.*

**Exercise 1.22** *(♦) Show that for 1-periodic functions, the trapezoidal rule leads to an error in $O(1/n^s)$ if we assume $s$ bounded derivatives.*

### 1.2.6 Concentration inequalities for matrices (♦♦)

It turns out the concentration inequalities that have been presented in this chapter apply equally well to matrices with the positive semi-definite order. The following bounds are adapted from Tropp (2012) and presented without proofs, with the following notations: $\lambda_{\max}(M)$ denote the largest eigenvalue of the symmetric matrix $M$. In contrast, $\|M\|_{\text{op}}$ denotes the largest singular value of a potentially rectangular matrix $M$, and $A \preccurlyeq B$ if and only if $B - A$ is positive semi-definite.

**Proposition 1.5 (Matrix Hoeffding bound) (Tropp, 2012, Theorem 1.3)** *Given $n$ independent symmetric matrices $M_i \in \mathbb{R}^{d\times d}$, such that for all $i \in \{1,\dots,n\}$, $\mathbb{E}[M_i] = 0$, $M_i^2 \preccurlyeq C_i^2$ almost surely. Then for all $t \geqslant 0$,*

$$\mathbb{P}\Big(\lambda_{\max}\Big(\frac{1}{n}\sum_{i=1}^n M_i\Big) \geqslant t\Big) \leqslant d \cdot \exp\Big(-\frac{nt^2}{8\sigma^2}\Big),$$

*for $\sigma^2 = \lambda_{\max}\big(\frac{1}{n}\sum_{i=1}^n C_i^2\big)$.*

**Proposition 1.6 (Matrix Bernstein bound) (Tropp, 2012, Theorem 1.4)** *Given $n$ independent symmetric matrices $M_i \in \mathbb{R}^{d\times d}$, such that for all $i \in \{1,\dots,n\}$, $\mathbb{E}[M_i] = 0$, $\lambda_{\max}(M_i) \leqslant c$ almost surely. Then for all $t \geqslant 0$,*

$$\mathbb{P}\Big(\lambda_{\max}\Big(\frac{1}{n}\sum_{i=1}^n M_i\Big) \geqslant t\Big) \leqslant d \cdot \exp\Big(-\frac{nt^2/2}{\sigma^2 + ct/3}\Big),$$

*for $\sigma^2 = \lambda_{\max}\big(\frac{1}{n}\sum_{i=1}^n \mathbb{E}[M_i^2]\big)$.*

We can make the following observations:

- Note the similarity with the corresponding bounds for scalar random variables when $d = 1$. McDiarmid's inequality can also be extended (Tropp, 2012, Corollary 7.5).
- These bounds apply as well to rectangular matrices $M_i \in \mathbb{R}^{d_1\times d_2}$ by considering the symmetric matrices $\widetilde{M_i} = \begin{pmatrix} 0 & M_i \\ M_i^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$, whose eigenvalues are plus and minus the singular values of $M_i$; see Section 1.1.4 and Stewart and Sun (1990, Theorem 4.2).

**Exercise 1.23** *Assume the matrices $M_i \in \mathbb{R}^{d_1\times d_2}$ are independent, have zero mean, and such that $\|M_i\|_{\text{op}} \leqslant c$ almost surely for all $i \in \{1,\dots,n\}$. Show that*

$$\mathbb{P}\Big(\Big\|\frac{1}{n}\sum_{i=1}^n M_i\Big\|_{\text{op}} \geqslant t\Big) \leqslant (d_1 + d_2) \cdot \exp\Big(-\frac{nt^2}{8c^2}\Big).$$

*Moreover, with $\sigma^2 = \max\big\{\lambda_{\max}\big(\frac{1}{n}\sum_{i=1}^n M_i^\top M_i\big), \lambda_{\max}\big(\frac{1}{n}\sum_{i=1}^n M_i M_i^\top\big)\big\}$, show that*

$$\mathbb{P}\Big(\Big\|\frac{1}{n}\sum_{i=1}^n M_i\Big\|_{\text{op}} \geqslant t\Big) \leqslant (d_1 + d_2) \cdot \exp\Big(-\frac{nt^2/2}{\sigma^2 + ct/3}\Big).$$

**Infinite-dimensional covariance operators (♦♦).** As used within Chapter 7, we will need to extend the results above, which depend on the underlying dimension, to the notion of "intrinsic dimension", which can still be finite if the underlying dimension is infinite. That is, we have this bound from Minsker (2017, Eq. (3.9)):

**Proposition 1.7 (Matrix Bernstein bound - intrinsic dimension)** *Given $n$ independent random bounded self-adjoint operators $M_i$ on a Hilbert space, such that for all $i \in \{1, \ldots, n\}$, $\mathbb{E}[M_i] = 0$, $\lambda_{\max}(M_i) \leqslant c$ almost surely, and $\frac{1}{n}\sum_{i=1}^n \mathbb{E}[M_i^2] \preccurlyeq V$. Then for all $t \geqslant 0$,*

$$\mathbb{P}\Big(\lambda_{\max}\Big(\frac{1}{n}\sum_{i=1}^n M_i\Big) \geqslant t\Big) \leqslant d \cdot \Big(1 + \frac{6}{n^2 t^4}(\sigma^2 + ct/3)^2\Big) \exp\Big(-\frac{nt^2/2}{\sigma^2 + ct/3}\Big),$$

*for $\sigma^2 \geqslant \lambda_{\max}(V)$ and $d = \frac{\operatorname{tr}(V)}{\sigma^2}$. When $t \geqslant \frac{c}{3n} + \frac{\sigma}{\sqrt{n}}$, then we get the upper-bound $7d\exp\big(-\frac{nt^2/2}{\sigma^2+ct/3}\big)$.*

# Chapter 2

# Introduction to supervised learning

**Chapter summary**

- Decision theory (loss, risk, optimal predictors): what is the optimal prediction and performance given infinite data and infinite computational resources?
- Statistical learning theory: when is an algorithm "consistent"?
- No free lunch theorems: learning is impossible without making assumptions.

| | |
|---|---|
| $\mathcal{X}$ | input space |
| $\mathcal{Y}$ | output space |
| $p$ | joint distribution on $\mathcal{X} \times \mathcal{Y}$ |
| $(x_1, y_1, \dots, x_n, y_n)$ | training data |
| $f : \mathcal{X} \to \mathcal{Y}$ | prediction function |
| $\ell(y, z)$ | loss function between output $y$ and prediction $z$ |
| $\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))]$ | expected risk of prediction function $f$ |
| $\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$ | empirical risk of prediction function $f$ |
| $f^*(x') = \arg\min_{z \in \mathcal{Y}} \mathbb{E}[\ell(y, z) \vert x = x']$ | Bayes prediction at $x'$ |
| $\mathcal{R}^* = \mathbb{E}_{x' \sim p} \inf_{z \in \mathcal{Y}} \mathbb{E}[\ell(y, z) \vert x = x']$ | Bayes risk |

Table 2.1: Summary of notions and notations presented in this chapter and used throughout the book.

## 2.1 From training data to predictions

**Main goal.** Given some observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of inputs/outputs, features/labels, covariates/responses (which are referred to as the training data), the main goal of supervised learning is to predict a new $y \in \mathcal{Y}$ given a new previously unseen $x \in \mathcal{X}$. The unobserved data are usually referred to as the testing data.

⚠ There are few fundamental differences between machine learning and the branch of statistics dealing with regression and its various extensions, particularly when providing theoretical guarantees. The focus on algorithms and computational scalability is arguably stronger within machine learning (but also present in statistics). At the same time, the emphasis on models and their interpretability beyond their predictive performance is more prominent within statistics (but also present in machine learning).

**Examples.** Supervised learning is used in many areas of science, engineering, and industry. There are thus many examples where $\mathcal{X}$ and $\mathcal{Y}$ can be very diverse:

- **Inputs $x \in \mathcal{X}$:** they can be images, sounds, videos, text, proteins, sequences of DNA bases, web pages, social network activities, sensors from industry, financial time series, etc. The set $\mathcal{X}$ may thus have a variety of structures that can be leveraged. All learning methods that we present in this textbook will use at one point a vector space representation of inputs, either by building an explicit mapping from $\mathcal{X}$ to a vector space (such as $\mathbb{R}^d$) or implicitly by using a notion of pairwise dissimilarity or similarity between pairs of inputs. The choice of these representations is highly domain-dependent. However, we note that (a) common topologies are encountered in many diverse areas (such as sequences, two-dimensional or three-dimensional objects), and thus common tools are used, and (b) learning these representations is an active area of research (see discussions in Chapter 7 and Chapter 9).

  In this textbook, we will primarily consider that inputs are $d$-dimensional vectors, with $d$ potentially large (up to $10^6$ or $10^9$).

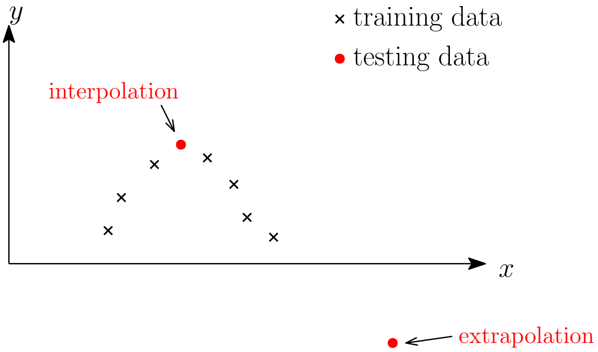
- **Outputs $y \in \mathcal{Y}$:** the most classical examples are binary labels $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$, multicategory classification problems with $\mathcal{Y} = \{1, \dots, k\}$, and classical regression with real responses/outputs $\mathcal{Y} = \mathbb{R}$. These will be the main examples we treat in most of the book. Note, however, that most of the concepts extend to the more general *structured prediction* set-up, where more general structured outputs (e.g., graph prediction, visual scene analysis, source separation) can be considered (see Chapter 13).

**Why is it difficult?** Supervised learning is difficult (and thus interesting) for a variety of reasons:

- The label $y$ may not be a deterministic function of $x$: given $x \in \mathcal{X}$, the outputs are noisy, that is, $y$ is not a deterministic function of $x$. When $y \in \mathbb{R}$, we will often make the simplifying "additive noise" assumption that $y = f(x) + \varepsilon$ with some zero-mean noise $\varepsilon$, but in general, we only assume that there is a conditional distribution of $y$

given $x$. This stochasticity is typically due to diverging views between labelers or dependence on random external unobserved quantities (that is, $y = f(x, z)$, with $z$ random and not observed).

- The prediction function $f$ may be quite complex, highly non-linear when $\mathcal{X}$ is a vector space, and even hard to define when $\mathcal{X}$ is not a vector space.
- Only a few $x$'s are observed: we thus need interpolation and potentially extrapolation (see below for an illustration for $\mathcal{X} = \mathcal{Y} = \mathbb{R}$), and therefore overfitting (predicting well on the training data but not as well on the testing data) is always a possibility.



Moreover, the training observations may not be uniformly distributed in $\mathcal{X}$. In this book, they will be assumed to be random, but some analyses will rely on deterministically located inputs to simplify some theoretical arguments.

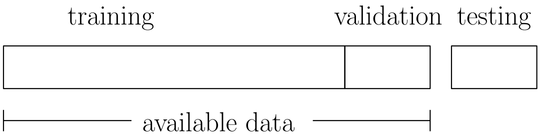
- The input space $\mathcal{X}$ may be very large, that is, with high dimension when this is a vector space. This leads to both computational issues (scalability) and statistical issues (generalization to unseen data). One usually refers to this problem as the *curse of dimensionality*.
- There may be a weak link between training and testing distributions. In other words, the data at training time can have different characteristics than the data at testing time.
- The criterion for performance is not always well defined.

**Main formalization.** Most modern theoretical analyses of supervised learning rely on a probabilistic formulation, that is, we see $(x_i, y_i)$ as a realization of random variables. The criterion is to maximize the expectation of some "performance" measure with respect to the distribution of the test data (in this book, *maximizing* the performance will be obtained by *minimizing* a loss function). The main assumption is that the random variables $(x_i, y_i)$ are independent and identically distributed (i.i.d.) with the same distribution as the testing distribution. In this course, we will ignore the potential mismatch between train and test distributions (although this is an important research topic, as in most applications, training data are not i.i.d. from the same distribution as the test data).

A machine learning algorithm $\mathcal{A}$ is a function that goes from a dataset, i.e., an element of $(\mathcal{X} \times \mathcal{Y})^n$, to a function from $\mathcal{X}$ to $\mathcal{Y}$. In other words, the output of a machine learning algorithm is itself an algorithm!

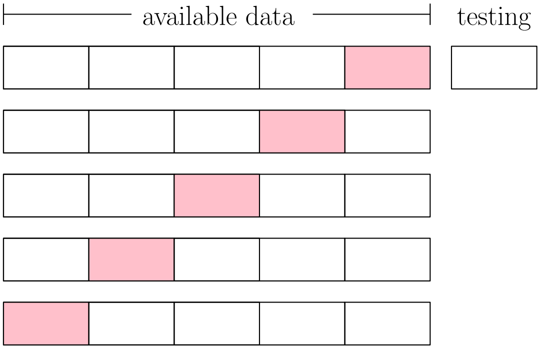
**Practical performance evaluation.** In practice, we do not have access to the test distribution but samples from it. In most cases, the data given to the machine learning user are split into three parts:

- the *training set*, on which learning models will be estimated,
- the *validation set*, to estimate hyperparameters (all learning techniques have some) to optimize the performance measure,
- the *testing set*, to evaluate the performance of the final chosen model.



⚠ In theory, the test set can only be used once! In practice, this is unfortunately only sometimes the case. If the test data are seen multiple times, the estimation of the performance on unseen data is overestimated.

Cross-validation is often preferred to use a maximal amount of training data and reduce the variability of the validation procedure: the available data are divided in $k$ folds (typically $k = 5$ or 10), and all models are estimated $k$ times, each time choosing a different fold as validation data (pink data below), and averaging the $k$ obtained error measures. Cross-validation can be applied to any learning method, and its detailed theoretical analysis is an active area of research (see, Arlot and Celisse, 2010, and the many references therein).



"Debugging" a machine learning implementation is often an art: on top of commonly found bugs, the learning method may not predict well enough on testing data. This is where theory can be useful to understand when a method is supposed to work or not. This is the primary goal of this book.

**Model selection.** Most machine learning models have hyper-parameters (e.g., regularization weight, size of the model, number of parameters). To estimate them from data, the common practical approach is to use validation approaches like those highlighted above. It is also possible to use penalization techniques based on generalization bounds. These two approaches are analyzed in Section 4.6.

**Random design vs. fixed design.** What we have described is often referred to as the "random design" set-up in statistics, where both $x$ and $y$ are assumed random and sampled i.i.d. It is common to simplify the analysis by considering that the input data $x_1, \dots, x_n$ are deterministic, either because they are actually deterministic (e.g., equally spaced in the input space $\mathcal{X}$), or by conditioning on them if they are actually random. This will be referred to as the "fixed design" setting and studied precisely in the context of least-squares regression in Chapter 3.

In the context of fixed design analysis, the error is evaluated "within-sample" (that is, for the same input points $x_1, \dots, x_n$, but over new associated outputs). This explicitly removes the difficulty of extrapolating to new inputs, hence a simplification in the mathematical analysis.

## 2.2 Decision theory

**Main question.** In this section, we tackle the following question: What is the optimal performance, regardless of the finiteness of the training data? In other words, what should be done if we have a perfect knowledge of the underlying probability distribution of the data? We will thus introduce the concept of *loss function*, *risk*, and *"Bayes" predictor*.

We consider a fixed (testing) distribution $p_{(x,y)}$ on $\mathcal{X} \times \mathcal{Y}$, with marginal distribution $p_{(x)}$ on $\mathcal{X}$. Note that we make no assumptions at this point on the input space $\mathcal{X}$.

⚠ We will almost always use the overloaded notation $p$, to denote $p_{(x,y)}$ and $p_{(x)}$, where the context can always make the definition unambiguous. For example, when $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we have $\mathbb{E}[f(x)] = \int_{\mathcal{X}} f(x) dp(x)$ and $\mathbb{E}[g(x,y)] = \int_{\mathcal{X} \times \mathcal{Y}} g(x,y) dp(x,y)$.

⚠ We ignore on-purpose measurability issues. The interested reader can look at the book by Christmann and Steinwart (2008) for a more formal presentation.

### 2.2.1 Loss functions

We consider a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (often $\mathbb{R}_+$), where $\ell(y,z)$ is the loss of predicting $z$ while the true label is $y$.

⚠ Some authors swap $y$ and $z$ in the definition above.

⚠ Some related research communities (e.g., economics) use the concept of "utility", which is then maximized.

The loss function only concerns the output space $\mathcal{Y}$ independently of the input space $\mathcal{X}$. The main examples are:

- **Binary classification:** $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$, or, less often, when seen as a subcase of the situation below, $\mathcal{Y} = \{1, 2\}$), and $\ell(y, z) = 1_{y \neq z}$ ("0-1" loss), that is, 0 if $y$ is equal to $z$ (no mistake), and 1 otherwise (mistake).

⚠ It is very common to mix the two conventions $\mathcal{Y} = \{0, 1\}$ and $\mathcal{Y} = \{-1, 1\}$.

- **Multicategory classification:** $\mathcal{Y} = \{1, \dots, k\}$, and $\ell(y, z) = 1_{y \neq z}$ ("0-1" loss).
- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). The absolute loss $\ell(y, z) = |y - z|$ is often used for "robust" estimation (since the penalty for large errors is smaller).
- **Structured prediction**: while this textbook focuses primarily on the examples above, there are many practical problems where $\mathcal{Y}$ is more complicated, with associated algorithms and theoretical results. For examples, when $\mathcal{Y} = \{0, 1\}^k$ (leading to multi-label classification), the Hamming loss $\ell(y, z) = \sum_{j=1}^k 1_{y_j \neq z_j}$ is commonly used; also, ranking problems involve losses on permutations. See Chapter 13.

Throughout the textbook, we will assume that the loss function is given to us. Note that in practice, the final user imposes the loss function, as this is how models will be evaluated. Clearly, a single real number may not be enough to characterize the entire prediction behavior. For example, in binary classification, there are two types of errors, false positives and false negatives, which can be considered simultaneously. Since we now have two performance measures, we typically need a curve to characterize the performance of a prediction function. This is precisely what "receiver operating characteristic" (ROC) curves are achieving (see, e.g., Bach et al., 2006, and references therein). For simplicity, we stick to a single loss function $\ell$ in this book.

⚠ While the loss function $\ell$ will be used to define the generalization performance below, for computational reasons, learning algorithms may explicitly minimize a different (but related) loss function, with better computational properties. This loss function used in training is often called a "surrogate". This will be studied in the context of binary classification in Section 4.1, and more generally for structured prediction in Chapter 13.

### 2.2.2 Risks

Given the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we can define the *expected risk* (also referred to as *generalization performance*, or *testing error*) of a function $f : \mathcal{X} \to \mathcal{Y}$, as the expectation of the loss function between the output $y$ and the prediction $f(x)$.

**Definition 2.1 (Expected risk)** *Given a prediction function $f : \mathcal{X} \to \mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and a probability distribution $p$ on $\mathcal{X} \times \mathcal{Y}$, the* expected risk *of $f$ is defined as:*

$$\mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big] = \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, f(x)) dp(x, y).$$

The risk depends on the distribution $p$ on $(x, y)$. We sometimes use the notation $\mathcal{R}_p(f)$

to make it explicit. The expected risk is our main performance criterion in this textbook.

⚠ Be careful with the randomness, or lack thereof, of $f$: when performing learning from data, $f$ will depend on the random training data and not on the testing data, and thus $\mathcal{R}(f)$ is typically random because of the dependence on the training data. However, as a function on functions, the expected risk $\mathcal{R}$ is deterministic.

Note that sometimes, we consider random predictions, that is, for any $x$, we output a distribution on $y$, and then the risk is taken as the expectation over the randomness of the outputs.

Averaging the loss on the training data defines the *empirical risk*, or *training error*.

**Definition 2.2 (Empirical risk)** *Given a prediction function $f : \mathcal{X} \to \mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, the* empirical risk *of $f$ is defined as:*

$$\widehat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

Note that $\widehat{\mathcal{R}}$ is a random function on functions (and is often applied to random functions, with dependent randomness as both will depend on the training data).

**Special cases.** For the classical losses defined earlier, the risks have specific formulations:

- **Binary classification:** $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$), and $\ell(y, z) = 1_{y \neq z}$ ("0-1" loss). We can express the risk as $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$. This is simply the probability of making a mistake on the testing data, while the empirical risk is the proportion of mistakes on the training data.

  ⚠ In practice, the *accuracy*, which is one minus the error rate, is often reported.

- **Multi-category classification:** $\mathcal{Y} = \{1, \dots, k\}$, and $\ell(y, z) = 1_{y \neq z}$ ("0-1" loss). We can also express the risk as $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$. This is also the probability of making a mistake.

- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). The risk is then equal to $\mathcal{R}(f) = \mathbb{E}\big[(y - f(x))^2\big]$, often referred to as mean squared error.

### 2.2.3 Bayes risk and Bayes predictor

Now that we have defined the performance criterion for supervised learning (the expected risk), the main question we tackle here is: what is the best prediction function $f$ (regardless of the training data)?

Using the conditional expectation and its associated law of total expectation, we have

$$\mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big] = \mathbb{E}\big[\mathbb{E}[\ell(y, f(x)) | x]\big],$$

which we can rewrite, for a fixed $x' \in \mathcal{X}$:

$$\mathcal{R}(f) = \mathbb{E}_{x' \sim p}\Big[\mathbb{E}\big[\ell(y, f(x'))|x = x'\big]\Big] = \int_{\mathcal{X}} \mathbb{E}\big[\ell(y, f(x))|x = x'\big] dp(x').$$

⚠ To distinguish between the random variable $x$ and a value it may take, we use the notation $x'$.

Given the conditional distribution given any $x' \in \mathcal{X}$, that is $y|x = x'$, we can define the *conditional risk* for any $z \in \mathcal{Y}$ (it is a deterministic function):

$$r(z|x') = \mathbb{E}\big[\ell(y, z)|x = x'\big],$$

which leads to

$$\mathcal{R}(f) = \int_{\mathcal{X}} r(f(x')|x') dp(x').$$

To find a minimizing function $f : \mathcal{X} \to \mathbb{R}$, let us first assume that the set $\mathcal{X}$ is finite: in this situation, the risk can be expressed as a sum of functions that depends on a *single* value of $f$, that is, $\mathcal{R}(f) = \sum_{x' \in \mathcal{X}} r(f(x')|x')\mathbb{P}(x = x')$. Therefore, we can minimize with respect to each $f(x')$ *independently*. Therefore, a minimizer of $\mathcal{R}(f)$ can be obtained by considering for any $x' \in \mathcal{X}$, the function value $f(x')$ to be equal to a minimizer $z \in \mathcal{Y}$ of $r(z|x') = \mathbb{E}\big[\ell(y, z)|x = x'\big]$. This extends beyond finite sets, as shown below.

⚠ Minimizing the expected risk with respect to a function $f$ in a restricted set does not lead to such decoupling.

**Proposition 2.1 (Bayes predictor and Bayes risk)** *The expected risk is minimized at a* Bayes predictor $f^* : \mathcal{X} \to \mathcal{Y}$ *satisfying for all* $x' \in \mathcal{X}$,

$$f^*(x') \in \arg\min_{z \in \mathcal{Y}} \mathbb{E}\big[\ell(y, z)|x = x'\big] = \arg\min_{z \in \mathcal{Y}} r(z|x'). \tag{2.1}$$

*The Bayes risk* $\mathcal{R}^*$ *is the risk of all Bayes predictors and is equal to*

$$\mathcal{R}^* = \mathbb{E}_{x' \sim p}\Big[\inf_{z \in \mathcal{Y}} \mathbb{E}\big[\ell(y, z)|x = x'\big]\Big].$$

**Proof** We have $\mathcal{R}(f) - \mathcal{R}^* = \mathcal{R}(f) - \mathcal{R}(f^*) = \int_{\mathcal{X}} \big[r(f(x')|x') - \min_{z \in \mathcal{Y}} r(z|x')\big] dp(x')$, which shows the proposition. ■

Note that (a) the Bayes predictor is not always unique, but that all lead to the same Bayes risk (for example, in binary classification when $\mathbb{P}(y = 1|x) = 1/2$), and (b) that the Bayes risk is usually non zero (unless the dependence between $x$ and $y$ is deterministic). Given a supervised learning problem, the Bayes risk is the optimal performance; we define the excess risk as the deviation with respect to the optimal risk.

**Definition 2.3 (Excess risk)** *The excess risk of a function* $f : \mathcal{X} \to \mathcal{Y}$ *is equal to* $\mathcal{R}(f) - \mathcal{R}^*$ *(it is always non-negative).*

Therefore, machine learning is "trivial": *given* the distribution $y|x$ for any $x$, the optimal predictor is known and given by Eq. (2.1). The difficulty will be that this distribution is unknown.

**Special cases.** For our usual set of losses, we can compute the Bayes predictors in closed form:

- **Binary classification**: the Bayes predictor for $\mathcal{Y} = \{0, 1\}$ and $\ell(y, z) = 1_{y \neq z}$ is such that

$$\begin{aligned} f^*(x') \in \arg \min_{z \in \{0,1\}} \mathbb{P}(y \neq z | x = x') &= \arg \min_{z \in \{0,1\}} 1 - \mathbb{P}(y = z | x = x') \\ &= \arg \max_{z \in \{0,1\}} \mathbb{P}(y = z | x = x'). \end{aligned}$$

  The optimal classifier will select the most likely class given $x'$. Using the notation $\eta(x') = \mathbb{P}(y = 1 | x = x')$, then, if $\eta(x') > 1/2$, $f^*(x') = 1$, while if $\eta(x') < 1/2$, $f^*(x') = 0$. What happens for $\eta(x') = 1/2$ is irrelevant.

  The Bayes risk is then equal to $\mathcal{R}^* = \mathbb{E}\big[\min\{\eta(x), 1 - \eta(x)\}\big]$, which in general strictly positive (unless $\eta(x) \in \{0, 1\}$ almost surely, that is, $y$ is a deterministic function of $x$).

  This extends directly to multiple categories $\mathcal{Y} = \{1, \dots, k\}$, for $k \geqslant 2$, where we have $f^*(x') \in \arg \max_{i \in \{1,\dots,k\}} \mathbb{P}(y = i | x = x')$.

  ⚠ These Bayes predictors and risks are only valid for the 0-1 loss. Less symmetric losses are common in applications (e.g., for spam detection) and would lead to different formulas (see exercise below).

- **Regression**: the Bayes predictor for $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ is such that[1]

$$\begin{aligned} f^*(x') &\in \arg\min_{z \in \mathbb{R}} \mathbb{E}\big[(y - z)^2 | x = x'\big] \\ &= \arg\min_{z \in \mathbb{R}} \Big\{ \mathbb{E}\big[(y - \mathbb{E}[y|x = x'])^2 | x = x'\big] + (z - \mathbb{E}[y|x = x'])^2 \Big\}. \end{aligned}$$

  This leads to the conditional expectation $f^*(x') = \mathbb{E}[y|x = x']$.

**Exercise 2.1** *We consider binary classification with $\mathcal{Y} = \{-1, 1\}$ with the loss function $\ell(-1, -1) = \ell(1, 1) = 0$ and $\ell(-1, 1) = c_- > 0$ (cost of a false positive), $\ell(1, -1) = c_+ > 0$ (cost of a false negative). Compute a Bayes predictor at $x$ as a function of $\mathbb{E}[y|x]$.*

**Exercise 2.2** *What is a Bayes predictor for regression with the absolute loss defined as $\ell(y, z) = |y - z|$?*

**Exercise 2.3** *What is a Bayes predictor for regression with the "$\varepsilon$-insentive" loss defined as $\ell(y, z) = \max\{0, |y - z| - \varepsilon\}$?*

**Exercise 2.4** *(inverting predictions) We consider the binary classification problem with $\mathcal{Y} = \{-1, 1\}$ and the 0-1 loss. Relate the risk of a prediction $f$ and its opposite $-f$.*

[1] We use the law of total variance: $\mathbb{E}[(y - a)^2] = \mathrm{var}(y) + (\mathbb{E}[y] - a)^2$ for any random variable $y$ and constant $a \in \mathbb{R}$, which can be shown by expanding the square.

**Exercise 2.5** *("chance" predictions) We consider binary classification problems with the 0-1 loss, what is the risk of a random prediction rule where we predict the two classes with equal probabilities independently of the input $x$? Same question with multiple categories.*

**Exercise 2.6** *(♦) We consider a random prediction rule where we predict from the probability distribution of $y$ given $x'$. When is this achieving the Bayes risk?*

## 2.3 Learning from data

The decision theory framework outlined in the previous section gives a test performance criterion and optimal predictors, but it depends on the full knowledge of the test distribution $p$. We now briefly review how we can obtain good prediction functions from training data, that is, data sampled i.i.d. from the same distribution.

Two main classes of prediction algorithms will be studied in this textbook:

(1) Local averaging (Chapter 6).

(2) Empirical risk minimization (Chapters 3, 4, 7, 8, 9, 11, 12, 13).

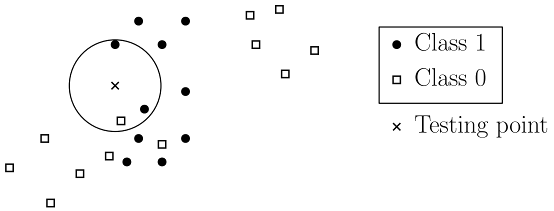
Note that there are prediction algorithms that do not fit precisely into one of these two categories, such as boosting or ensemble classifiers (see Chapter 10). Moreover, some situations do not fit the classical i.i.d. framework, such as in online learning (see Chapter 11). We consider probabilistic methods in Chapter 14, which rely on a different principle.

### 2.3.1 Local averaging

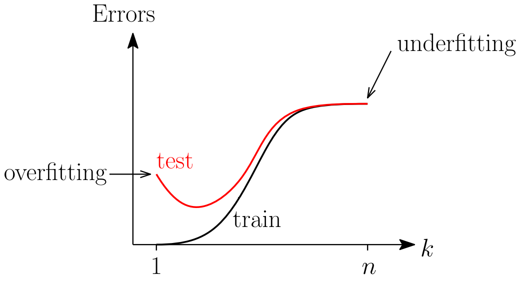
The goal here is to try to approximate/emulate the Bayes predictor, e.g., $f^*(x') = \mathbb{E}(y|x = x')$ for least-squares regression, from empirical data. This is often done by explicit/implicit estimation of the conditional distribution by *local averaging* ($k$-nearest neighbors, which is used as the primary example for this chapter, Nadaraya Watson, or decision trees). We briefly outline here the main properties for one instance of these algorithms; see Chapter 6 for details.

**$k$-nearest-neighbor classification.** Given $n$ observations $(x_1, y_1), \dots, (x_n, y_n)$ where $\mathcal{X}$ is a metric space and $\mathcal{Y} \in \{0, 1\}$, a new point $x^{\text{test}}$ is classified by a majority vote among the $k$-nearest neighbors of $x^{\text{test}}$.

Below, we consider the 3-nearest-neighbor classifier on a particular testing point (which will be predicted as 1).



- Pros: (a) no optimization or training, (b) often easy to implement, (c) can get very good performance in low dimensions (in particular for non-linear dependences between $x$ and $y$).
- Cons: (a) slow at query time: must pass through all training data at each testing point (there are algorithmic tools to reduce complexity, see Chapter 6), (b) bad for high-dimensional data (because of the curse of dimensionality, more on this in Chapter 6), (c) the choice of local distance function is crucial, (d) the choice of "width" hyperparameters (or $k$) has to be performed.
- Plot of training errors and testing errors as functions of $k$ for a typical problem. When $k$ is too large, there is *underfitting* (the learned function is too close to a constant, which is too simple), while for $k$ too small, there is *overfitting* (there is a strong discrepancy between the testing and training errors).



**Exercise 2.7** *How would the curve move when $n$ increases (assuming the same balance between classes)?*

### 2.3.2 Empirical risk minimization

Consider a parameterized family of prediction functions, often referred to as *models*, $f_\theta : \mathcal{X} \to \mathcal{Y}$ for $\theta \in \Theta$ (typically a subset of a vector space), and minimize the empirical risk with respect to $\theta \in \Theta$:

$$\hat{\mathcal{R}}(f_\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)).$$

This defines an estimator $\hat{\theta} \in \arg\min_{\theta \in \Theta} \hat{\mathcal{R}}(f_\theta)$, and thus a function $f_{\hat{\theta}} : \mathcal{X} \to \mathcal{Y}$.

The most classic example is linear least-squares regression (studied at length in Chapter 3), where we minimize $\frac{1}{n} \sum_{i=1}^{n} (y_i - \theta^\top \varphi(x_i))^2$, where $f$ is linear in some feature vector

$\varphi(x) \in \mathbb{R}^d$ (there is no need for $\mathcal{X}$ to be a vector space). The vector $\varphi(x)$ can be quite large (or even implicit, like in kernel methods, see Chapter 7). Other examples include neural networks (Chapter 9).

- Pros: (a) can be relatively easy to optimize (e.g., least-squares with simple derivation and numerical algebra, see Chapter 3), many algorithms available (primarily based on gradient descent, see Chapter 5), (b) can be applied in any dimension (if a suitable feature vector is available).
- Cons: (a) can be relatively hard to optimize when the optimization formulation is not convex (e.g., neural networks), (b) need a suitable feature vector for linear methods, (c) the dependence on parameters can be complex (e.g., neural networks), (d) need some capacity control to avoid overfitting, (e) how to parameterize functions with values in $\{0,1\}$ (see Chapter 4 for the use of convex surrogates)?

**Risk decomposition.** The material in this section will be studied further in more detail in Chapter 4.

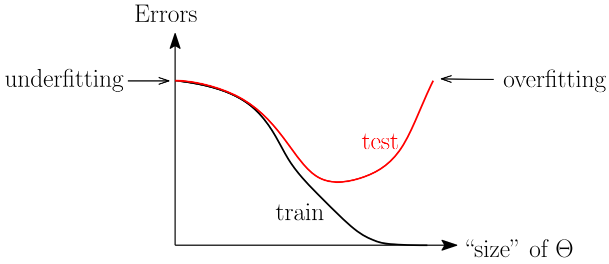
- Risk decomposition in estimation error + approximation error: given any $\hat{\theta} \in \Theta$, we can write the excess risk of $f_{\hat{\theta}}$ as:

$$\begin{aligned}\mathcal{R}(f_{\hat{\theta}}) - \mathcal{R}^* &= \Big\{\mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'})\Big\} + \Big\{\inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - \mathcal{R}^*\Big\} \\ &= \text{estimation error} \qquad + \ \text{approximation error.}\end{aligned}$$

The approximation error $\inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - \mathcal{R}^*$ is always non-negative, does not depend on the chosen $f_{\hat{\theta}}$ and depends only on the class of functions parameterized by $\theta \in \Theta$. It is thus always a deterministic quantity, which characterizes the modeling assumptions made by the chosen class of functions. When $\Theta$ grows, the approximation error goes down to zero if arbitrary functions can be approximated arbitrarily well by the functions $f_\theta$. It is also independent of $n$.

The estimation error $\big\{\mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'})\big\}$ is also always non-negative and is typically random because the function $f_{\hat{\theta}}$ is random. It typically decreases in $n$ and increases when $\Theta$ grows.

Overall, the typical error curves look like this:



- Typically, we will see in later chapters that the estimation error is often decomposed as follows, for $\theta'$ a minimizer on $\Theta$ of the expected risk $\mathcal{R}(f_{\theta'})$:

$$\begin{aligned}\mathcal{R}(f_{\hat\theta}) - \mathcal{R}(f_{\theta'}) &= \big\{\mathcal{R}(f_{\hat\theta}) - \hat{\mathcal{R}}(f_{\hat\theta})\big\} + \big\{\hat{\mathcal{R}}(f_{\hat\theta}) - \hat{\mathcal{R}}(f_{\theta'})\big\} + \big\{\hat{\mathcal{R}}(f_{\theta'}) - \mathcal{R}(f_{\theta'})\big\} \\ &\leqslant 2\sup_{\theta\in\Theta}\big|\hat{\mathcal{R}}(f_\theta) - \mathcal{R}(f_\theta)\big| + \text{ empirical optimization error,}\end{aligned}$$

where the "empirical optimization error" is $\sup_{\theta\in\Theta}\big\{\hat{\mathcal{R}}(f_{\hat\theta}) - \hat{\mathcal{R}}(f_\theta)\big\}$ (it is equal to zero for the exact empirical risk minimizer, but in practice when using optimization algorithms from Chapter 5, it is not). The uniform deviation $\sup_{\theta\in\Theta}\big|\hat{\mathcal{R}}(f_\theta) - \mathcal{R}(f_\theta)\big|$ grows with the "size" of $\Theta$, and usually decays with $n$. See more details in Chapter 4.

**Capacity control.** To avoid overfitting, we need to make sure that the set of allowed functions is not too large by typically reducing the number of parameters or by restricting the norm of predictors (thus by lowering the "size" of $\Theta$): this typically leads to constrained optimization, and allows for risk decompositions as done above.

Capacity control can also be done by regularization, that is, by minimizing

$$\hat{\mathcal{R}}(f_\theta) + \lambda\Omega(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f_\theta(x_i)) + \lambda\Omega(\theta),$$

where $\Omega(\theta)$ controls the complexity of $f_\theta$. The main example is ridge regression:

$$\min_{\theta\in\mathbb{R}^d} \ \frac{1}{n}\sum_{i=1}^n (y_i - \theta^\top\varphi(x_i))^2 + \lambda\|\theta\|_2^2.$$

This is often easier for optimization but harder to analyze (see Chapter 4 and Chapter 5).

⚠ There is a difference between parameters (e.g., $\theta$) learned on the training data and hyperparameters (e.g., $\lambda$) estimated on the validation data.

**Examples of approximations by polynomials in one-dimensional regression.** We consider $(x, y) \in \mathbb{R}\times\mathbb{R}$, with prediction functions which are polynomials of order $k$, from $k = 0$ (constant functions) to $k = 14$. For each $k$, the model has $k+1$ parameters. The training error (using square loss) is minimized with $n = 20$ observations. The data were generated with inputs uniformly distributed on $[-1, 1]$ and outputs as the quadratic function $f(x) = x^2 - \frac{1}{2}$ of the inputs plus some independent additive noise (Gaussian with standard deviation $1/4$). As shown in Figure 2.1 and Figure 2.2, the training error monotonically decreases in $k$ while the testing error goes down and then up.

## 2.4 Statistical learning theory

The goal of learning theory is to provide some guarantees of performance on unseen data. A common assumption is that the data $\mathcal{D}_n(p) = \{(x_1, y_1), \dots, (x_n, y_n)\}$ is obtained as

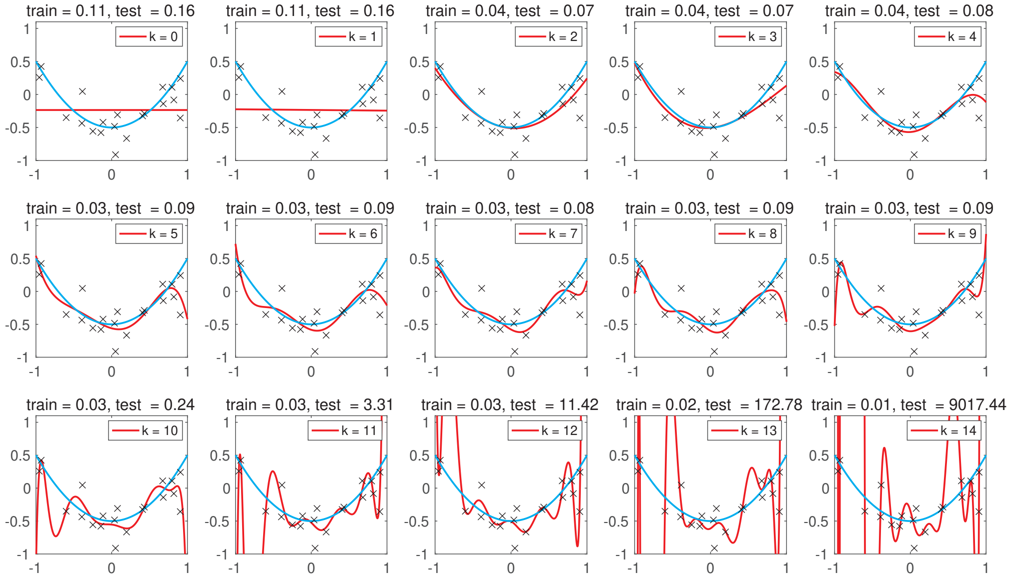


Figure 2.1: Polynomial regression with increasing orders $k$. Plots of estimated functions, with training and testing errors. The Bayes prediction function $f^*(x) = \mathbb{E}[y|x]$ is plotted in blue.

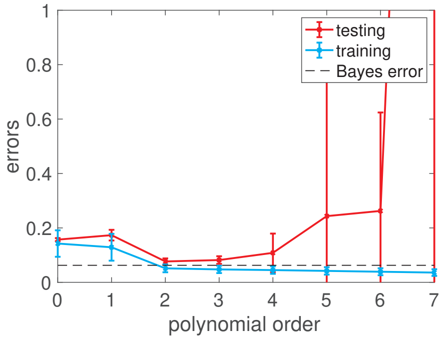


Figure 2.2: Polynomial regression with increasing orders. Plots of training and testing errors with error bars (computed as standard deviations obtained from 32 replications), together with the Bayes error.

independent and identically distributed (i.i.d.) observations from some unknown distribution $p$ from a family $\mathcal{P}$.

An algorithm $\mathcal{A}$ is a mapping from $\mathcal{D}_n(p)$ (for any $n$) to a function from $\mathcal{X}$ to $\mathcal{Y}$. The expected risk depends on the probability distribution $p \in \mathcal{P}$, as $\mathcal{R}_p(f)$. The goal is to find $\mathcal{A}$ such that the (expected) risk

$$\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p))) - \mathcal{R}_p^*$$

is small, where $\mathcal{R}_p^*$ is the Bayes risk (which depends on the joint distribution $p$), assuming $\mathcal{D}_n(p)$ is sampled from $p$, but without knowing which $p \in \mathcal{P}$ is considered. Moreover, the risk is random because $\mathcal{D}_n(p)$ is random.

### 2.4.1 Measures of performance

There are several ways of dealing with randomness to obtain a criterion.

- *Expected error*: we measure performance as

  $$\mathbb{E}\big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\big],$$

  where the expectation is with respect to the training data. An algorithm $\mathcal{A}$ is called *consistent in expectation* for the distribution $p$, if

  $$\mathbb{E}\big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\big] - \mathcal{R}_p^*$$

  goes to zero when $n$ tends to infinity. In this course, we will primarily use this notion of consistency.

- *"Probably approximately correct" (PAC) learning*: for a given $\delta \in (0,1)$ and $\varepsilon > 0$:

  $$\mathbb{P}\Big(\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p))) - \mathcal{R}_p^* \leqslant \varepsilon\Big) \geqslant 1 - \delta.$$

  The crux is to find $\varepsilon$, which is as small as possible (typically as a function of $\delta$). The notion of PAC consistency corresponds, for any $\varepsilon > 0$, to have such an inequality for each $n$ and a sequence $\delta_n$ that tends to zero.

### 2.4.2 Notions of consistency over classes of problems

An algorithm is called *universally consistent* (in expectation) if for all probability distributions $p = p_{(x,y)}$ on $(x, y)$ the algorithm $\mathcal{A}$ is consistent in expectation for the distribution $p$.

⚠ Be careful with the order of quantifiers: convergence speed will depend on $p$. See the no-free lunch theorem section below to highlight that having a uniform rate over all distributions is hopeless.

Most often, we want to study uniform consistency within a class $\mathcal{P}$ of distributions satisfying some regularity properties (e.g., the inputs live in a compact space, or the dependence between $y$ and $x$ is at most of some complexity). We thus aim at finding an algorithm $\mathcal{A}$ such that

$$\sup_{p\in\mathcal{P}} \ \mathbb{E}\big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\big] - \mathcal{R}_p^*$$

is as small as possible. The so-called "minimax risk" is equal to

$$\inf_{\mathcal{A}} \sup_{p\in\mathcal{P}} \ \mathbb{E}\big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\big] - \mathcal{R}_p^*.$$

This is typically a function of the sample size $n$ and properties of $\mathcal{X}$, $\mathcal{Y}$ and the allowed set of problems $\mathcal{P}$ (e.g., dimension of $\mathcal{X}$, number of parameters). To compute estimates of the minimax risk, several techniques exist:

- Upper-bounding the optimal performance: one given algorithm with a convergence proof provides an upper bound. This is the main focus of this book.
- Lower-bounding the optimal performance: in some setups, it is possible to show that the infimum over all algorithms is greater than a certain quantity. See Chapter 15 for a description of techniques to obtain such lower bounds. Machine learners are happy when upper-bounds and lower-bounds match (up to constant factors).

**Non-asymptotic vs. asymptotic analysis.** The analysis can be "non-asymptotic", with an upper bound with explicit dependence on all quantities; the bound is then valid for all $n$, even if sometimes vacuous (e.g., a bound greater than 1 for a loss uniformly bounded by 1).

The analysis can also be "asymptotic", where, for example, $n$ goes to infinity and limits are taken (alternatively, several quantities can be made to grow simultaneously).

⚠ What (arguably) matters most here is the dependence of these rates on the problem, not the choice of "in expectation" vs. "in high probability", or "asymptotic" vs. "non-asymptotic", as long as the problem parameters explicitly appear.

## 2.5 No free lunch theorems (♦)

Although it may be tempting to define the optimal learning algorithm that works optimally for all distributions, this is impossible. In other words, learning is only possible with assumptions. See Devroye et al. (1996, Chapter 7) for more details.

The following theorem shows that for any algorithm, for a fixed $n$, there is a data distribution that makes the algorithm useless (with a risk that is the same as the chance level).

**Proposition 2.2 (no free lunch - fixed $n$)** *Consider the binary classification with 0-1 loss, with $\mathcal{X}$ infinite. Let $\mathcal{P}$ denote the set of all probability distributions on $\mathcal{X} \times \{0, 1\}$.*

*For any $n > 0$ and any learning algorithm $\mathcal{A}$,*

$$\sup_{p\in\mathcal{P}} \mathbb{E}\big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\big] - \mathcal{R}_p^* \geqslant 1/2.$$

**Proof** (♦♦) Let $k$ be a positive integer. Without loss of generality, we can assume that $\mathbb{N} \subset \mathcal{X}$. The main ideas of the proof are (a) to construct a probability distribution supported on $k$ elements in $\mathbb{N}$, where $k$ is large compared to $n$ (which is fixed), and to show that the knowledge of $n$ labels does not imply doing well on all $k$ elements, and (b) to choose parameters of this distribution (the vector $r$ below) by comparing to a performance obtained by random parameters.

Given $r \in \{0,1\}^k$, we define the joint distribution $p$ on $(x, y)$ such that we have $\mathbb{P}(x = j, y = r_j) = 1/k$ for $j \in \{1, \dots, k\}$; that is, for $x$, we choose one of the first $k$ elements uniformly at random, and then $y$ is selected deterministically as $y = r_x$. Thus, the Bayes risk is zero (because there is a deterministic relationship): $\mathcal{R}_p^* = 0$.

Denoting $\hat{f}_{\mathcal{D}_n} = \mathcal{A}(\mathcal{D}_n(p))$ the classifier, and $S(r) = \mathbb{E}\big[\mathcal{R}_p(\hat{f}_{\mathcal{D}_n})\big]$ the expectation of the expected risk, we want to maximize $S(r)$ with respect to $r \in \{0,1\}^k$; the maximum is greater than the expectation of $S(r)$ for any probability distribution $q$ on $r$, in particular the uniform distribution (each $r_j$ being an independent unbiased Bernoulli variable). Then

$$\begin{aligned}\max_{r\in\{0,1\}^k} S(r) &\geqslant \mathbb{E}_{r\sim q} S(r)\\ &= \mathbb{P}(\hat{f}_{\mathcal{D}_n}(x) \neq y) = \mathbb{P}(\hat{f}_{\mathcal{D}_n}(x) \neq r_x),\end{aligned}$$

because $x$ is almost surely in $\{1, \dots, k\}$ and $y = r_x$ almost surely. Note that we take expectations and probabilities with respect to $x_1, \dots, x_n, x$, *and* $r$ (all being independent of each other).

Then, we get, using that $\mathcal{D}_n(p) = \{x_1, r_{x_1}, \dots, x_n, r_{x_n}\}$:

$$\begin{aligned}\mathbb{E}_{r\sim q} S(r) &= \mathbb{E}\Big[\mathbb{P}\big(\hat{f}_{\mathcal{D}_n}(x) \neq r_x \big| x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n}\big)\Big] \text{ by the law of total expectation,}\\ &\geqslant \mathbb{E}\Big[\mathbb{P}\big(\hat{f}_{\mathcal{D}_n}(x) \neq r_x \ \&\ x \notin \{x_1, \dots, x_n\} \big| x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n}\big)\Big]\\ &\qquad\qquad \text{by monotonicity of probabilities,}\\ &= \mathbb{E}\Big[\frac{1}{2}\mathbb{P}\big(x \notin \{x_1, \dots, x_n\} \big| x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n}\big)\Big],\end{aligned}$$

because $\mathbb{P}\big(\hat{f}_{\mathcal{D}_n}(x) \neq r_x \big| x \notin \{x_1, \dots, x_n\}, x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n}\big) = 1/2$ (the label $x = r_x$ has the same probability of being 0 or 1, given that it was not observed). Thus,

$$\mathbb{E}_{r\sim q} S(r) \geqslant \frac{1}{2}\mathbb{P}\big(x \notin \{x_1, \dots, x_n\}\big) = \frac{1}{2}\mathbb{E}\Big[\prod_{i=1}^n \mathbb{P}(x_i \neq x | x)\Big] = \frac{1}{2}\big(1 - 1/k\big)^n.$$

Given $n$, we can let $k$ tend to infinity to conclude. ■

A caveat is that the hard distribution may depend on $n$ (from the proof, it takes $k$ values, with $k$ tending to infinity fast enough compared with $n$). The following theorem is given without proof and is much "stronger" (Devroye et al., 1996, Theorem 7.2), as it more convincingly shows that learning can be arbitrarily slow without assumption (note that the earlier one is not a corollary of the later one).

**Proposition 2.3 (no free lunch - sequence of errors)** *Consider a binary classification problem with the 0-1 loss, with $\mathcal{X}$ infinite. Let $\mathcal{P}$ denote the set of all probability distributions on $\mathcal{X} \times \{0,1\}$. For any decreasing sequence $a_n$ tending to zero and such that $a_1 \leqslant 1/16$, for any learning algorithm $\mathcal{A}$, there exists $p \in \mathcal{P}$, such that for all $n \geqslant 1$:*

$$\mathbb{E}\big[\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)))\big] - \mathcal{R}_p^* \geqslant a_n.$$

## 2.6 Quest for adaptivity

As seen in the previous section, no method can be universal and achieve a good convergence rate on all problems. However, such negative results consider classes of problems that are arbitrarily large. In this textbook, we will consider reduced sets of learning problems by considering $\mathcal{X} = \mathbb{R}^d$ and putting restrictions on the target function $f^*$ based on smoothness and/or dependence on an unknown low-dimensional projection. That is, the most general set of functions will be the set of Lipschitz-continuous functions, for which the optimal rate will be essentially proportional to $O(n^{-1/d})$, typical of the curse of dimensionality. No method can beat this, not $k$-nearest-neighbors, not kernel methods, not even neural networks.

When the target function is smoother, that is, with all derivatives up to order $m$ bounded, then we will see that kernel methods (Chapter 7) and neural networks (Chapter 9), with the proper choice of the regularization parameter, will lead to the optimal rate of $O(n^{-m/d})$.

When the target function moreover depends only on a $k$-dimensional linear projection, neural networks (if the optimization problem is solved correctly) will have the extra ability to lead to rates of the form $O(n^{-m/k})$ instead of $O(n^{-m/d})$, which is not the case for kernel methods (see Chapter 9)

Note that another form of adaptivity, which is often considered, is in situations where the input data lie on a submanifold of $\mathbb{R}^d$ (e.g., an affine subspace), where for most methods presented in this textbook, adaptivity is obtained. In the convergence rate, $d$ can be replaced by the dimension of the subspace (or submanifold) where the data live. See studies by Kpotufe (2011) for $k$-nearest neighbors, and Hamm and Steinwart (2021) for kernel methods.

See more details in `https://francisbach.com/quest-for-adaptivity/` as well as Chapter 7 and Chapter 9 for detailed results.

## 2.7 Beyond supervised learning

This textbook focuses primarily on the traditional supervised learning paradigm, with independent and identically distributed data and where the training and testing distributions match. Many applications require extensions to this basic framework, which also lead to many interesting theoretical developments that are out of scope. We present briefly some of these extensions below with references for further reading.

**Unsupervised learning.** While in supervised learning, both inputs and outputs (e.g., labels) are observed, and the main goal is to model how the output depends on the input, in unsupervised learning, only inputs are given. The goal is then to "find some structure" within the data, for example, an affine subspace around which the data live (for principal component analysis), the separation of the data in several groups (for clustering), or the identification of an explicit latent variable model (such as with matrix factorization). The new representation of the data is typically either used for visualization (then, with two or three dimensions), or for reducing dimension before applying a supervised learning algorithm.

While supervised learning relied on an explicit decision-theoretic framework, it is not always clear how to characterize performance and perform evaluation; each method typically has an ad-hoc empirical criterion, such as reconstruction of the data, full or partial (like in self-supervised learning), log-likelihood when probabilistic models are used (see Chapter 14), in particular graphical models (Bishop, 2006; Murphy, 2012). Often, intermediate representations are used for subsequent processing (see, e.g., Goodfellow et al., 2016).

Theory can be applied to sampling behavior and recovery of specific structures when assumed (e.g., for clustering or dimension reduction), with a variety of theoretical results in manifold learning, matrix factorization methods such as K-means, principal component analysis or sparse dictionary learning (Mairal et al., 2014), outlier/novelty detection, or independent component analysis (Hyvärinen et al., 2001).

**Semi-supervised learning.** This is the intermediate situation between supervised and unsupervised, with some labeled (typically few) examples and some unlabeled (typically many) examples. Several frameworks exist based on various assumptions (Chapelle et al., 2010; Van Engelen and Hoos, 2020).

**Active learning.** This is a similar setting as semi-supervised learning, but the user can choose which unlabelled point to label to maximize performance once new labels are obtained. The selection of samples to label is often done by computing some form of uncertainty estimation on the unlabelled data points (see, e.g. Settles, 2009).

**Online learning.** Mostly in a supervised setting, this framework allows us to go beyond the training/testing splits, where data are acquired, and predictions are made on the fly, with a criterion that takes into account the sequential nature of learning. See Cesa-Bianchi and Lugosi (2006); Hazan (2022) and Chapter 11.

**Reinforcement learning.** On top of the sequential nature of learning already present in online learning, predictions may influence the future sampling distributions, for example, in situations where some agents interact with an environment (Sutton and Barto, 2018), with algorithms relying on similar concepts than optimal control (Liberzon, 2011).

## 2.8 Summary - book outline

Now that the main concepts are introduced, we can give an outline of the book chapters, which we have separated into three parts.

**Part I: Preliminaries.** The first part contains this introductory chapter and Chapter 3 on linear least-squares regression. We start with such a chapter as it allows for the introduction of the main concepts of the book, such as underfitting, overfitting, regularization, using simple linear algebra, and without the need for more advanced analytic or probabilistic tools.

**Part II: Generalization bounds for learning algorithms.** The second part is dedicated to the core concepts in learning theory and should be studied sequentially.

- **Empirical risk minimization**: Chapter 4 is dedicated to methods based on the minimization of the potentially regularized or constrained regularized risk, with the introduction of the key concept of Rademacher complexity that analyzes estimation errors efficiently. Convex surrogates for binary classification are also introduced to allow the use of only real-valued prediction functions.
- **Optimization**: Chapter 5 shows how gradient-based techniques can be used to minimize approximately the empirical risk and, through stochastic gradient descent, obtain generalization bounds for finitely-parameterized linear models (which are linear in their parameters) leading to convex objective functions.
- **Local averaging methods**: Chapter 6 is the first chapter dealing with so-called "non-parametric" methods that can potentially adapt to complex prediction functions. This class of methods explicitly builds a prediction function mimicking the Bayes predictor (without any optimization algorithm), such as $k$-nearest-neighbor methods. These methods are classically subject to the curse of dimensionality.
- **Kernel methods**: Chapter 7 presents the most general class of linear models that can be infinite-dimensional and adapt to complex prediction functions. They are made computationally feasible using the "kernel trick", and they still rely on convex optimization, so they lead to strong theoretical guarantees, particularly by being adaptive to the smoothness of the target prediction function.
- **Sparse methods**: While the previous chapter focused on Euclidean or Hilbertian regularization techniques for linear models, Chapter 8 considers regularization

by sparsity-inducing penalties such as the $\ell_1$-norm or the $\ell_0$-penalty, leading to the high-dimensional phenomenon that learning is possible even with potentially exponentially many irrelevant variables.

- **Neural networks**: Chapter 9 presents a class of prediction functions that are not linearly parameterized, therefore leading to non-convex optimization problems, where obtaining a global optimum is not certain. The chapter studies approximation and estimation errors, showing the adaptivity of neural networks to smoothness and linear latent variables (in particular for non-linear variable selection).

**Part III: Special topics.** The third part presents a series of special topic chapters that can be read in essentially any order.

- **Ensemble learning**: Chapter 10 presents a class of techniques aiming at combining several predictors obtained from the same model class but learned on slightly modified datasets. This can be done in parallel, such as in bagging techniques, or sequentially, like boosting methods.
- **From online learning to bandits**: Chapter 11 consider sequential decision problems within the regret framework, focusing first on online convex optimization, then on zeroth order optimization (without access to gradients), and then multi-armed bandits.
- **Over-parameterized models**: Chapter 12 presents a series of results related to models with a large number of parameters (enough to fit the training data perfectly) and trained with gradient descent. We present the implicit bias of gradient descent in linear models towards minimum Euclidean norm solutions and then the double descent phenomenon, before looking at implicit biases and global convergence for non-convex optimization problems.
- **Structured prediction**: Chapter 13 goes beyond the traditional regression and binary classification frameworks by first considering multi-category classification and then the general framework of structured prediction, where output spaces can be arbitrarily complex.
- **Probabilistic methods**: Chapter 14 presents a collection of results related to probabilistic modeling, highlighting that probabilistic interpretations can sometimes be misleading but also naturally lead to model selection frameworks through Bayesian inference and PAC-Bayes analysis.
- **Lower bounds on performance**: While most of the book is dedicated to obtaining upper bounds on the generalization or optimization performance of our algorithms, Chapter 15 considers lower-bounds on such performance, showing how many algorithms presented in this book are, in fact, “optimal” for a specific class of learning or optimization problems.

# Chapter 3

# Linear least-squares regression

**Chapter summary**

– Ordinary least-squares estimator: least-squares regression with linearly parameterized predictors leads to a linear system of size $d$ (the number of predictors).

– Guarantees in the fixed design setting with no regularization: when the inputs are assumed deterministic and $d < n$, the excess risk is equal to $\sigma^2 d/n$.

– Ridge regression: with $\ell_2$-regularization, excess risk bounds become dimension independent and allow high-dimensional feature vectors where $d > n$.

– Guarantees in the random design setting: although they are harder to show, they have a similar form.

– Lower bound of performance: under well-specification, the rate $\sigma^2 d/n$ is unimprovable.

## 3.1 Introduction

In this chapter, we introduce and analyze linear least-squares regression, a tool that can be traced back to Legendre (1805) and Gauss (1809).[1]

Why should we study linear least-squares regression? Has there not been any progress since 1805? A few reasons:

- It already captures many of the concepts in learning theory, such as the bias-variance trade-off, as well as the dependence of generalization performance on the underlying dimension of the problem with no regularization or on dimension-less quantities when regularization is added.
- Because of its simplicity, many results can be easily derived without the need for

[1] see https://en.wikipedia.org/wiki/Least_squares for an interesting discussion and the claim that Gauss knew about it already in 1795.

complicated mathematics, both in terms of algorithms and statistical analysis (simple linear algebra for the simplest results in the fixed design setting).

- Using non-linear features, it can be extended to arbitrary non-linear predictions (see kernel methods in Chapter 7).

In subsequent chapters, we will extend many of these results beyond least-squares with the proper additional mathematical tools.

## 3.2 Least-squares framework

We recall the goal of supervised machine learning from Chapter 2: given some observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of inputs/outputs, features/variables (training data), given a new $x \in \mathcal{X}$, predict $y \in \mathcal{Y}$ (testing data) with a *regression* function $f$ such that $y \approx f(x)$. We assume that $\mathcal{Y} = \mathbb{R}$ and we use the square loss $\ell(y, z) = (y - z)^2$, for which we know from the previous chapter that the optimal predictor is $f^*(x) = \mathbb{E}[y|x]$.

In this chapter, we consider empirical risk minimization. We choose a parameterized family of prediction functions (often referred to as "models") $f_\theta : \mathcal{X} \to \mathcal{Y} = \mathbb{R}$ for some parameter $\theta \in \Theta$ and minimize the empirical risk

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2,$$

leading to the estimator $\hat{\theta} \in \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2$. Note that in most cases, the Bayes predictor $f^*$ does not belong to the class of functions $\{f_\theta,\ \theta \in \Theta\}$, that is, the model is said *misspecified*.

Least-squares regression can be carried out with parameterizations of the function $f_\theta$, which may be non-linear in the parameter $\theta$ (such as for neural networks in Chapter 9). In this chapter, we will consider only situations where $f_\theta(x)$ is linear in $\theta$, which is thus assumed to live in a vector space, and which we take to be $\mathbb{R}^d$ for simplicity.

⚠ Being linear in $x$ or linear in $\theta$ is different!

While we assume linearity in the parameter $\theta$, nothing forces $f_\theta(x)$ to be linear in the input $x$. In fact, even the concept of linearity may be meaningless if $\mathcal{X}$ is not a vector space. If $f_\theta(x)$ is linear in $\theta \in \mathbb{R}^d$, then it has to be a linear combination of the form $f_\theta(x) = \sum_{i=1}^{d} \alpha_i(x)\theta_i$, where $\alpha_i : \mathcal{X} \to \mathbb{R}$, $i = 1, \dots, d$, are $d$ functions. By concatenating them in a vector $\varphi(x) \in \mathbb{R}^d$ where $\varphi(x)_i = \alpha_i(x)$, we get the representation

$$f_\theta(x) = \varphi(x)^\top \theta.$$

The vector $\varphi(x) \in \mathbb{R}^d$ is typically called the *feature vector*, which we assume to be known (in other words, it is given to us and can be computed explicitly when needed). We thus consider minimizing the empirical risk

$$\hat{\mathcal{R}}(\theta) := \frac{1}{n}\sum_{i=1}^{n}(y_i - \varphi(x_i)^\top\theta)^2. \tag{3.1}$$

When $\mathcal{X} \subset \mathbb{R}^d$, we can make the extra assumptions that $f_\theta$ is an affine function, which could be obtained through $\varphi(x) = \binom{x}{1} = (x^\top, 1)^\top \in \mathbb{R}^{d+1}$. Other classical assumptions are $\varphi(x)$ composed of monomials (so that prediction functions are polynomials). We will see in Chapter 7 (kernel methods) that we can consider infinite-dimensional features.

**Matrix notation.** The cost function above in Eq. (3.1) can be rewritten in matrix notations. Let $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ be the vector of outputs (sometimes called the *response vector*), and $\Phi \in \mathbb{R}^{n\times d}$ the matrix of inputs, whose rows are $\varphi(x_i)^\top$. It is called the *design* matrix or *data* matrix. In these notations, the empirical risk is

$$\hat{\mathcal{R}}(\theta) = \frac{1}{n}\|y - \Phi\theta\|_2^2, \tag{3.2}$$

where $\|\alpha\|_2^2 = \sum_{j=1}^d \alpha_j^2$ is the squared $\ell_2$-norm of $\alpha$.

⚠ It is sometimes tempting at first to avoid matrix notations. We strongly advise against it as it leads to lengthy and error-prone formulas.

## 3.3 Ordinary least-squares (OLS) estimator

We assume that the matrix $\Phi \in \mathbb{R}^{n\times d}$ has full column rank (i.e., the rank of $\Phi$ is $d$). In particular, the problem is said to be "over-determined," and we must have $d \leqslant n$, that is, more observations than feature dimension. Equivalently, we assume that $\Phi^\top\Phi \in \mathbb{R}^{d\times d}$ is invertible.

**Definition 3.1 (OLS)** *When $\Phi$ has full column rank, the minimizer of Eq. (3.2) is unique and called the* ordinary least-squares *(OLS) estimator.*

### 3.3.1 Closed-form solution

Since the objective function is quadratic, the gradient will be linear, and zeroing it will lead to a closed-form solution.

**Proposition 3.1** *When $\Phi$ has full column rank, the OLS estimator exists and is unique. It is given by*

$$\hat{\theta} = (\Phi^\top\Phi)^{-1}\Phi^\top y.$$

*Denote the (non-centered[2]) empirical* covariance *matrix by $\widehat{\Sigma} := \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d\times d}$; we have $\hat{\theta} = \frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top y$.*

[2] The "centered" covariance matrix would be $\frac{1}{n}\sum_{i=1}^n[\varphi(x_i)-\mu][\varphi(x_i)-\mu]^\top$ where $\mu = \frac{1}{n}\sum_{i=1}^n \varphi(x_i) \in \mathbb{R}^d$ is the empirical mean, while we consider $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top$.

**Proof** Since the function $\hat{\mathcal{R}}$ is coercive (i.e., going to infinity at infinity) and continuous, it admits at least a minimizer. Moreover, it is differentiable, so a minimizer $\hat{\theta}$ must satisfy $\hat{\mathcal{R}}'(\hat{\theta}) = 0$ where $\hat{\mathcal{R}}'(\theta) \in \mathbb{R}^d$ is the gradient of $\hat{\mathcal{R}}$ at $\theta$. For all $\theta \in \mathbb{R}^d$, we have, by expanding the square and computing the gradient:

$$\hat{\mathcal{R}}(\theta) = \frac{1}{n}\left(\|y\|_2^2 - 2\theta^\top\Phi^\top y + \theta^\top\Phi^\top\Phi\theta\right) \quad \text{and} \quad \hat{\mathcal{R}}'(\theta) = \frac{2}{n}\left(\Phi^\top\Phi\theta - \Phi^\top y\right).$$

The condition $\hat{\mathcal{R}}'(\hat{\theta}) = 0$ gives the so-called *normal equations*:

$$\Phi^\top\Phi\hat{\theta} = \Phi^\top y.$$

The normal equations have a unique solution $\hat{\theta} = (\Phi^\top\Phi)^{-1}\Phi^\top y$. This shows the uniqueness of the minimizer of $\hat{\mathcal{R}}$ as well as its closed-form expression. ■

Another way to show the uniqueness of the minimizer is by showing that $\hat{\mathcal{R}}$ is strongly convex since $\hat{\mathcal{R}}''(\theta) = 2\widehat{\Sigma}$ is invertible for all $\theta \in \mathbb{R}^d$ (convexity will be studied in Chapter 5).

⚠ For readers worried about carrying a factor of two in the gradients, we will use an additional factor 1/2 in chapters on optimization (e.g., Chapter 5).

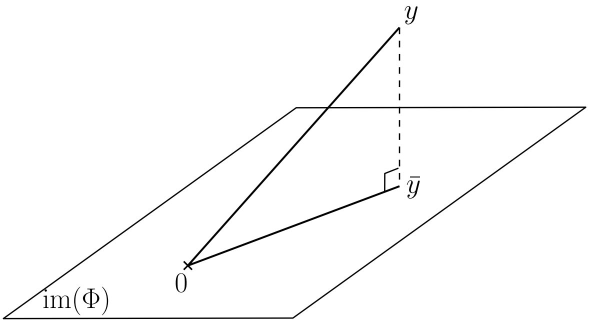
### 3.3.2 Geometric interpretation

**Proposition 3.2** *The vector of predictions $\Phi\hat{\theta} = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top y$ is the orthogonal projection of $y \in \mathbb{R}^n$ onto* $\mathrm{im}(\Phi) \subset \mathbb{R}^n$, *the column space of* $\Phi$.

**Proof** Let us show that $P = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top \in \mathbb{R}^{n\times n}$ is the orthogonal projection on $\mathrm{im}(\Phi)$. For any $a \in \mathbb{R}^d$, it holds $P\Phi a = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top\Phi a = \Phi a$, so $Pu = u$ for all $u \in \mathrm{im}(\Phi)$. Also, since $\mathrm{im}(\Phi)^\perp = \mathrm{null}(\Phi^\top)$, $Pu' = 0$ for all $u' \in \mathrm{im}(\Phi)^\perp$. These properties characterize the orthogonal projection on $\mathrm{im}(\Phi)$. ■

Thus, we can interpret the OLS estimation as doing the following (see below for an illustration):

1. compute $\bar{y}$ the projection of $y$ on the image of $\Phi$,
2. solve the linear system $\Phi\theta = \bar{y}$ which has a unique solution.



### 3.3.3 Numerical resolution

While the closed-form $\hat{\theta} = (\Phi^\top \Phi)^{-1}\Phi^\top y$ is convenient for analysis, inverting $\Phi^\top \Phi$ is sometimes unstable and has a large computational cost when $d$ is large. The following methods are usually preferred.

**$QR$ factorization.** The $QR$ decomposition factorizes the matrix $\Phi$ as $\Phi = QR$ where $Q \in \mathbb{R}^{n\times d}$ has orthonormal columns, that is, $Q^\top Q = I$, and $R \in \mathbb{R}^{d\times d}$ is upper triangular (see Golub and Loan, 1996). Computing a $QR$ decomposition is faster and more stable than inverting a matrix. We then have $\Phi^\top \Phi = R^\top Q^\top QR = R^\top R$, and $R$ is thus the Cholesky factor of $\Phi^\top \Phi$. One then has

$$(\Phi^\top \Phi)\hat{\theta} = \Phi^\top y \Leftrightarrow R^\top Q^\top QR\hat{\theta} = R^\top Q^\top y \Leftrightarrow R^\top R\hat{\theta} = R^\top Q^\top y \Leftrightarrow R\hat{\theta} = Q^\top y.$$

It only remains to solve a triangular linear system, which is easy. The overall running time complexity remains $O(d^3)$. The conjugate gradient algorithm can also be used (see Golub and Loan, 1996, for details).

**Gradient descent.** We can bypass the need for matrix inversion or factorization using gradient descent. It consists in approximately minimizing $\hat{\mathcal{R}}$ by taking an initial point $\theta_0 \in \mathbb{R}^d$ and iteratively going towards the minimizer by following the opposite of the gradient

$$\theta_t = \theta_{t-1} - \gamma \hat{\mathcal{R}}'(\theta_{t-1}) \quad \text{for } t \geqslant 1,$$

where $\gamma > 0$ is the step-size. When these iterates converge, it is towards the OLS estimator since a fixed-point $\theta$ satisfies $\hat{\mathcal{R}}'(\theta) = 0$. We will study such algorithms in Chapter 5, with running-time complexities going down to linear in $d$.

## 3.4 Statistical analysis of OLS

We now provide guarantees on the performance of the OLS estimator. There are two classical settings of analysis for least-squares:

- *Random design.* In this setting, both the inputs and the outputs are random. This is the classical setting of supervised machine learning, where the goal is *generalization* to unseen data (as in the last chapter). Since obtaining guarantees is mathematically more complicated, it will be done after the fixed design setting.
- *Fixed design.* In this setting, we assume that the input data $(x_1, \dots, x_n)$ are *not* random, and we are interested in obtaining a small prediction error *on those input points only.* Alternatively, this can be seen as a prediction problem where the input distribution is the empirical distribution of $(x_1, \dots, x_n)$.

  Our goal is thus to minimize the fixed design risk (where thus $\Phi$ is deterministic):

$$\mathcal{R}(\theta) = \mathbb{E}_y\Big[\frac{1}{n}\sum_{i=1}^n (y_i - \varphi(x_i)^\top\theta)^2\Big] = \mathbb{E}_y\Big[\frac{1}{n}\|y - \Phi\theta\|_2^2\Big]. \quad (3.3)$$

This assumption allows a complete analysis with basic linear algebra. It is justified in some settings, e.g., when the inputs are equally spaced along a fixed grid, but is otherwise just a simplifying assumption. It can also be understood as learning the optimal vector $\Phi\theta_* \in \mathbb{R}^n$ of best predictions instead of a function from $\mathcal{X}$ to $\mathbb{R}$.

In the fixed design setting, no attempts are made to generalize to unseen input points $x \in \mathcal{X}$, and we want to estimate well a label vector $y$ resampled from the same distribution as the observed $y$. The risk in Eq. (3.3) is often called the *in-sample prediction error*, and the task can be seen as "denoising" the labels.

We will first consider below the fixed design setting, where the celebrated rate $\sigma^2 d/n$ will appear naturally.

**Relationship to maximum likelihood estimation.** If, in the fixed design setting, we make the stronger assumption that the noise is Gaussian with mean zero and variance $\sigma^2$, i.e., $\varepsilon_i = y_i - \varphi(x_i)^\top \theta_* \sim \mathcal{N}(0, \sigma^2)$, then the least mean-squares estimator of $\theta_*$ coincides with the maximum likelihood estimator (where $\Phi$ is assumed fixed). Indeed, the density/likelihood of $y$ is, using independence and the density of the normal distribution:

$$p(y|\theta, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\big( - (y_i - \varphi(x_i)^\top \theta)^2 / (2\sigma^2)\big).$$

Taking the logarithm and removing constants, the maximum likelihood estimator $(\tilde{\theta}, \tilde{\sigma}^2)$ minimizes

$$\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \varphi(x_i)^\top \theta)^2 + \frac{n}{2} \log(\sigma^2).$$

We immediately see that $\tilde{\theta} = \hat{\theta}$, that is, OLS corresponds to maximum likelihood.

⚠ While maximum likelihood under a Gaussian model provides an interesting interpretation, the Gaussian assumption is not needed for the forthcoming analysis.

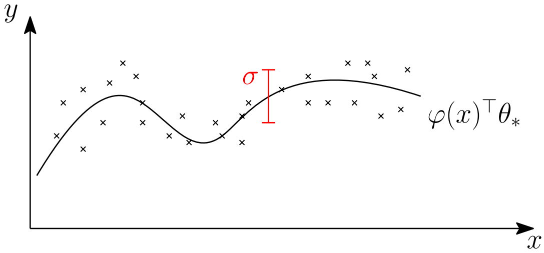
**Exercise 3.1** *In the Gaussian model above, compute $\tilde{\sigma}^2$ the maximum likelihood estimator of $\sigma^2$.*

## 3.5 Fixed design setting

We now assume that $\Phi$ is deterministic, and as before, we assume that $\widehat{\Sigma} = \frac{1}{n}\Phi^\top \Phi$ is invertible. Any guarantee requires assumptions about how the data are generated. We assume that:

- There exists a vector $\theta_* \in \mathbb{R}^d$ such that the relationship between input and output is for $i \in \{1, \dots, n\}$
$$y_i = \varphi(x_i)^\top \theta_* + \varepsilon_i. \tag{3.4}$$
- All noise variables $\varepsilon_i$, $i \in \{1, \dots, n\}$, are independent with expectation $\mathbb{E}[\varepsilon_i] = 0$ and variance $\mathbb{E}[\varepsilon_i^2] = \sigma^2$.

The vector $\varepsilon \in \mathbb{R}^n$ accounts for variabilities in the output due to unobserved factors or noise. The "homoscedasticity" assumption above, where the noise variances are uniform, is made for simplicity (and allows for the later bound $\sigma^2 d/n$ to be an equality). Note that to prove upper-bounds in performance, we could also only assume that $\mathbb{E}[\varepsilon_i^2] \leqslant \sigma^2$ for each $i \in \{1, \dots, n\}$. The noise variance $\sigma^2$ is the expected squared error between the observations $y_i$ and the model $\varphi(x_i)^\top \theta_*$.



⚠ In Eq. (3.4), we assume the model is *well-specified*, that is, the target function is a linear function of $\varphi(x)$. In general, an additional approximation error is incurred because of a misspecified model (see Chapter 4).

Denoting by $\mathcal{R}^*$ the minimum value of $\mathcal{R}(\theta) = \mathbb{E}_y \left[\frac{1}{n}\|y - \Phi\theta\|_2^2\right]$ over $\mathbb{R}^d$, the following proposition shows that it is attained at $\theta_*$, and that it is equal to $\sigma^2$.

**Proposition 3.3 (Risk decomposition for OLS - fixed design)** *Under the linear model and fixed design assumptions above, for any $\theta \in \mathbb{R}^d$, we have $\mathcal{R}^* = \sigma^2$ and*

$$\mathcal{R}(\theta) - \mathcal{R}^* = \|\theta - \theta_*\|_{\widehat{\Sigma}}^2,$$

*where $\widehat{\Sigma} := \frac{1}{n}\Phi^\top\Phi$ is the input covariance matrix and $\|\theta\|_{\widehat{\Sigma}}^2 := \theta^\top \widehat{\Sigma}\theta$. If $\hat{\theta}$ is now a random variable (such as an estimator of $\theta_*$), then*

$$\mathbb{E}[\mathcal{R}(\hat{\theta})] - \mathcal{R}^* = \underbrace{\|\mathbb{E}[\hat{\theta}] - \theta_*\|_{\widehat{\Sigma}}^2}_{\text{Bias}} + \underbrace{\mathbb{E}\left[\|\hat{\theta} - \mathbb{E}[\hat{\theta}]\|_{\widehat{\Sigma}}^2\right]}_{\text{Variance}}.$$

**Proof** We have, using $y = \Phi\theta_* + \varepsilon$, with $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}[\|\varepsilon\|_2^2] = n\sigma^2$:

$$\begin{aligned}
\mathcal{R}(\theta) &= \mathbb{E}_y\left[\frac{1}{n}\|y - \Phi\theta\|_2^2\right] = \mathbb{E}_\varepsilon\left[\frac{1}{n}\|\Phi\theta_* + \varepsilon - \Phi\theta\|_2^2\right] \\
&= \frac{1}{n}\mathbb{E}_\varepsilon\left[\|\Phi(\theta_* - \theta)\|_2^2 + \|\varepsilon\|_2^2 + 2\big[\Phi(\theta_* - \theta)\big]^\top \varepsilon\right] \\
&= \sigma^2 + \frac{1}{n}(\theta - \theta_*)^\top \Phi^\top \Phi(\theta - \theta_*).
\end{aligned}$$

Since $\widehat{\Sigma} = \frac{1}{n}\Phi^\top\Phi$ is invertible, this shows that $\theta_*$ is the unique global minimizer of $\mathcal{R}(\theta)$, and that the minimum value $\mathcal{R}^*$ is equal to $\sigma^2$. This shows the first claim.

Now if $\theta$ is random, we perform the usual *bias/variance decomposition*:

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\hat\theta)] - \mathcal{R}^* &= \mathbb{E}\Big[\|\hat\theta - \mathbb{E}[\hat\theta] + \mathbb{E}[\hat\theta] - \theta_*\|^2_{\widehat\Sigma}\Big] \\ &= \mathbb{E}\Big[\|\hat\theta - \mathbb{E}[\hat\theta]\|^2_{\widehat\Sigma}\Big] + 2\mathbb{E}\Big[(\hat\theta - \mathbb{E}[\hat\theta])^\top \widehat\Sigma(\mathbb{E}[\hat\theta] - \theta_*)\Big] + \mathbb{E}\Big[\|\mathbb{E}[\hat\theta] - \theta_*\|^2_{\widehat\Sigma}\Big] \\ &= \mathbb{E}\Big[\|\hat\theta - \mathbb{E}[\hat\theta]\|^2_{\widehat\Sigma}\Big] + 0 + \|\mathbb{E}[\hat\theta] - \theta_*\|^2_{\widehat\Sigma}\end{aligned}$$

(note that this is also a simple application of the law of total variance for vectors, that is, $\mathbb{E}\big[\|z-a\|_M^2\big] = \|\mathbb{E}[z]-a\|_M^2 + \mathbb{E}\big[\|z-\mathbb{E}[z]\|_M^2\big]$ to $a=\theta_*$, $M=\widehat\Sigma$ and $z=\hat\theta$). ■

Note that the quantity $\|\cdot\|_{\widehat\Sigma}$ is called the Mahalanobis distance norm (it is a "true" norm whenever $\widehat\Sigma$ is positive definite). It is the norm on the parameter space induced by the input data.

### 3.5.1 Statistical properties of the OLS estimator

We can now analyze the properties of the OLS estimator, which has a closed form $\hat\theta = (\Phi^\top\Phi)^{-1}\Phi^\top y = \widehat\Sigma^{-1}(\frac{1}{n}\Phi^\top y)$, with the model $y = \Phi\theta_* + \varepsilon$. The only randomness comes from $\varepsilon$, and we thus need to compute the expectation of linear and quadratic forms in $\varepsilon$.

**Proposition 3.4 (Estimation properties of OLS)** *The OLS estimator $\hat\theta$ has the following properties:*

1. *it is unbiased, that is, $\mathbb{E}[\hat\theta] = \theta_*$,*
2. *its variance is $\mathrm{var}(\hat\theta) = \mathbb{E}\big[(\hat\theta-\theta_*)(\hat\theta-\theta_*)^\top\big] = \frac{\sigma^2}{n}\widehat\Sigma^{-1}$, where $\widehat\Sigma^{-1}$ is often called the* precision *matrix.*

**Proof**

1. Since $\mathbb{E}[y] = \Phi\theta_*$, we have directly $\mathbb{E}[\hat\theta] = (\Phi^\top\Phi)^{-1}\Phi^\top\Phi\theta_* = \theta_*$.
2. It follows that $\hat\theta - \theta_* = (\Phi^\top\Phi)^{-1}\Phi^\top(\Phi\theta_* + \varepsilon) - \theta_* = (\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon$. Thus, using that $\mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I$, we get

$$\mathrm{var}(\hat\theta) = \mathbb{E}\big[(\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon\varepsilon^\top\Phi(\Phi^\top\Phi)^{-1}\big] = \sigma^2(\Phi^\top\Phi)^{-1}(\Phi^\top\Phi)(\Phi^\top\Phi)^{-1} = \sigma^2(\Phi^\top\Phi)^{-1},$$

which leads to the desired result $\frac{\sigma^2}{n}\widehat\Sigma^{-1}$.

■

We can now put back the expression of the variance in the risk.

**Proposition 3.5 (Risk of OLS)** *The excess risk of the OLS estimator is equal to*

$$\mathbb{E}\big[\mathcal{R}(\hat\theta)\big] - \mathcal{R}^* = \frac{\sigma^2 d}{n}. \tag{3.5}$$

**Proof** Note here that the expectation is over $\varepsilon$ only as we are in the fixed design setting. Using the risk decomposition of Proposition 3.3 and the fact that $\mathbb{E}[\hat\theta] = \theta_*$, we have

$$\mathbb{E}\big[\mathcal{R}(\hat\theta)\big] - \mathcal{R}^* = \mathbb{E}\big[\|\hat\theta - \theta_*\|^2_{\widehat\Sigma}\big].$$

We have: $\mathbb{E}\big[\mathcal{R}(\hat{\theta})\big] - \mathcal{R}^* = \mathrm{tr}[\mathrm{var}(\hat{\theta})\widehat{\Sigma}] = \mathrm{tr}[\frac{\sigma^2}{n}\widehat{\Sigma}^{-1}\widehat{\Sigma}] = \frac{\sigma^2}{n}\,\mathrm{tr}(I) = \frac{\sigma^2 d}{n}$.

We can also give a direct proof. Using the identity $\hat{\theta} - \theta_* = (\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon$, we get

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\hat{\theta})] - \mathcal{R}^* &= \mathbb{E}\|(\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon\|^2_{\widehat{\Sigma}} \\ &= \frac{1}{n}\mathbb{E}\big[\varepsilon^\top\Phi(\Phi^\top\Phi)^{-1}\Phi^\top\Phi(\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon\big] = \frac{1}{n}\mathbb{E}\big[\varepsilon^\top\Phi(\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon\big] \\ &= \frac{1}{n}\mathbb{E}\big[\varepsilon^\top P\varepsilon\big] = \frac{1}{n}\mathbb{E}\big[\mathrm{tr}(P\varepsilon\varepsilon^\top)\big] = \frac{\sigma^2}{n}\mathrm{tr}(P) = \frac{\sigma^2 d}{n},\end{aligned}$$

where we used that $P = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top$ is the orthogonal projection on $\mathrm{im}(\Phi)$, which is $d$-dimensional. ■

We can make the following observations:

- ⚠ In the fixed design setting, the expectation over $\varepsilon$ appears twice: (1) in the definition of the risk of some $\theta$ in Eq. (3.3), and when taking an expectation over the data in Eq. (3.5).

  **Exercise 3.2** *Show that the expected empirical risk* $\mathbb{E}[\hat{\mathcal{R}}(\hat{\theta})]$ *is equal to* $\mathbb{E}[\hat{\mathcal{R}}(\hat{\theta})] = \frac{n-d}{n}\sigma^2$. *In particular, when* $n > d$, *deduce that an unbiased estimator of the noise variance* $\sigma^2$ *is given by* $\frac{\|Y-\Phi\hat{\theta}\|_2^2}{n-d}$.

- In the exercise above, we have an expression of the expected training error, which is equal to $\frac{n-d}{n}\sigma^2 = \sigma^2 - \frac{d}{n}\sigma^2$, while the expected testing error is $\sigma^2 + \frac{d}{n}\sigma^2$. We thus see that in the context of least-squares, the training error underestimates (in expectation) the testing error by a factor of $2\sigma^2 d/n$, which characterizes the amount of overfitting. This difference can be used to perform model selection.[3]

- In the fixed design setting, OLS thus leads to unbiased estimation, with an excess risk of $\sigma^2 d/n$.

- On the positive side, the math is elementary, and as we will show in Section 3.7, the obtained convergence rate is optimal.

- On the negative side, for the excess risk being small compared to $\sigma^2$, we need $d/n$ to be small, which seems to exclude high-dimensional problems where $d$ is close to $n$ (let alone problems where $d > n$ or $d$ much larger than $n$). Regularization (ridge in this chapter or with the $\ell_1$-norm in Chapter 8) will come to the rescue.

- This is only for the fixed design setting. We consider the random design setting below, which is a bit more involved mathematically, primarily because of the presence of $\widehat{\Sigma}^{-1}$ which does not cancel anymore, leading to the term $\widehat{\Sigma}^{-1}\Sigma$, where $\Sigma$ is the population covariance matrix.

**Exercise 3.3** *(general noise) We consider the fixed design regression model* $y = \Phi\theta_* + \varepsilon$

[3] See https://en.wikipedia.org/wiki/Mallows's_Cp.

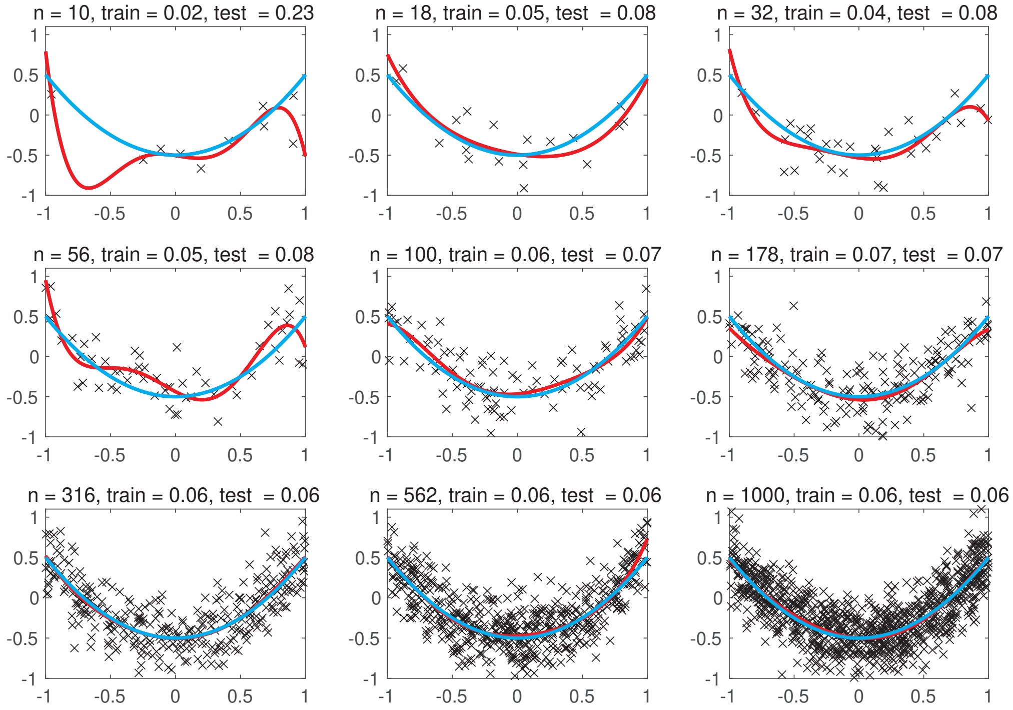


Figure 3.1: Polynomial regression with a varying number of observations. Blue: Optimal prediction, red: estimated prediction by ordinary least-squares with degree 5 polynomials.

*with $\varepsilon$ with zero mean and covariance matrix equal to $C$ (not anymore $\sigma^2 I$). Show that the expected excess risk of the OLS estimator is equal to $\frac{1}{n}\operatorname{tr}\big[\Phi(\Phi^\top\Phi)^{-1}\Phi^\top C\big]$.*

**Exercise 3.4** *(♦) (multivariate regression) We consider $\mathcal{Y} = \mathbb{R}^k$ and the multivariate regression model $y = \theta_*^\top\varphi(x) + \varepsilon \in \mathbb{R}^k$, where $\theta_* \in \mathbb{R}^{d\times k}$, and $\varepsilon$ has zero-mean with covariance matrix $C \in \mathbb{R}^{k\times k}$. In the fixed regression setting with design matrix $\Phi \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^{n\times k}$* the matrix *of responses, derive the OLS estimator and its excess risk.*

### 3.5.2 Experiments

To illustrate the bound $\sigma^2 d/n$, we consider polynomial regression in one dimension, with $x \in \mathbb{R}$, $\varphi(x) = (1, x, x^2, \dots, x^k)^\top \in \mathbb{R}^{k+1}$, so $d = k + 1$. The inputs are sampled from the uniform distribution in $[-1, 1]$, while the optimal regression function is a degree 2 polynomial $f(x) = x^2 - \frac{1}{2}$ (blue curve in Figure 3.1). Gaussian noise with standard deviation $\frac{1}{4}$ is added to generate the outputs (black crosses). The ordinary least-squares estimator is plotted in red for various values of $n$, from $n = 10$ to $n = 1000$, for $k = 5$.

We can now plot in Figure 3.2 the expected excess risk as a function of $n$, estimated by 32 replications of the experiment, together with the bound. In the right plot, we consider

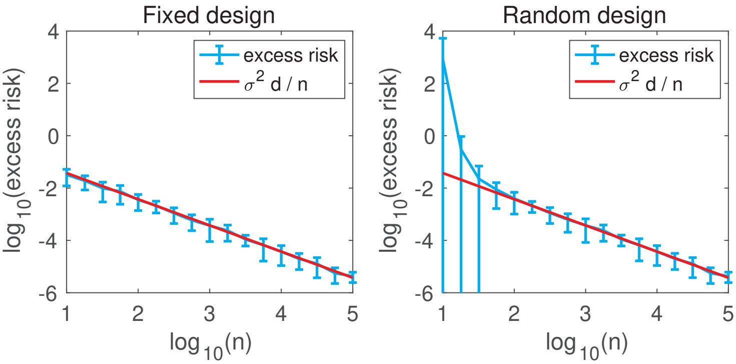


Figure 3.2: Convergence rate for polynomial regression with error bars (obtained from 32 replications by adding/subtracting standard deviations), plotted in logarithmic scale, with fixed design (left plot) and random design (right plot). The large error bars for small $n$ in the right plot are due to the lower error bar being negative before taking the logarithm.

the random design setting (generalization error, considered in Section 3.8), while in the left plot, we consider the fixed design setting (in-sample error). Notice the closeness of the bound for all $n$ for the fixed design (as predicted by our bounds), while this is only valid for $n$ large enough in the random design setting.

## 3.6 Ridge least-squares regression

**Least-squares in high dimensions.** When $d/n$ approaches 1, we are essentially memorizing the observations $y_i$ (that is, for example, when $d = n$ and $\Phi$ is a square invertible matrix, $\theta = \Phi^{-1}y$ leads to $y = \Phi\theta$, that is, ordinary least-squares will lead to a perfect fit, which is typically not good for generalization to unseen data, see more details in Chapter 12). Also, when $d > n$, $\Phi^\top\Phi$ is not invertible, and the normal equations admit a linear subspace of solutions. These behaviors of OLS in high dimensions ($d$ large) are often undesirable.

Two main classes of solutions exist to fix these issues: dimension reduction and regularization. Dimension reduction aims to replace the feature vector $\varphi(x)$ with another feature of lower dimension, with a classical method being principal component analysis, presented in Section 3.9. Regularization adds a term to the least-squares objective, typically either an $\ell_1$-penalty $\|\theta\|_1$ (leading to *"Lasso"* regression, see Chapter 8) or $\|\theta\|_2^2$ (leading to *ridge* regression, as done in this chapter and also in Chapter 7).

**Definition 3.2 (Ridge least-squares regression)** *For a regularization parameter $\lambda > 0$, we define the ridge least-squares estimator $\hat{\theta}_\lambda$ as the minimizer of*

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n}\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_2^2.$$

The ridge regression estimator can be obtained in closed form, and we do not require anymore $\Phi^\top\Phi$ to be invertible.

**Proposition 3.6** *We recall that* $\widehat{\Sigma} = \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d\times d}$. *We have* $\hat{\theta}_\lambda = \dfrac{1}{n}(\widehat{\Sigma}+\lambda I)^{-1}\Phi^\top y$.

**Proof** As for the proof of Proposition 3.1, we can compute the gradient of the objective function, which is equal to $\frac{2}{n}\big(\Phi^\top\Phi\theta - \Phi^\top y\big) + 2\lambda\theta$. Setting it to zero leads to the estimator. Note that when $\lambda > 0$, the linear system always has a unique solution regardless of the invertibility of $\widehat{\Sigma}$. ■

**Exercise 3.5** *Using the matrix inversion lemma (Section 1.1.3), show that the estimator above can be written* $\hat{\theta}_\lambda = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top y = \Phi^\top(\Phi\Phi^\top + n\lambda I)^{-1}y$. *What could be the computational benefits?*

As for the OLS estimator, we can analyze its statistical properties under the linear model and fixed design assumptions. See Chapter 7 for an analysis of random design and potentially infinite-dimensional features.

**Proposition 3.7** *Under the linear model assumption (and for the fixed design setting), the ridge least-squares estimator* $\hat{\theta}_\lambda = \frac{1}{n}(\widehat{\Sigma}+\lambda I)^{-1}\Phi^\top y$ *has the following excess risk*

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_\lambda)\big] - \mathcal{R}^* = \lambda^2\theta_*^\top(\widehat{\Sigma}+\lambda I)^{-2}\widehat{\Sigma}\theta_* + \frac{\sigma^2}{n}\operatorname{tr}\big[\widehat{\Sigma}^2(\widehat{\Sigma}+\lambda I)^{-2}\big].$$

**Proof** We use the risk decomposition of Proposition 3.3 into a bias term $B$ and a variance term $V$. Since we have $\mathbb{E}[\hat{\theta}_\lambda] = \frac{1}{n}(\widehat{\Sigma}+\lambda I)^{-1}\Phi^\top\Phi\theta_* = (\widehat{\Sigma}+\lambda I)^{-1}\widehat{\Sigma}\theta_* = \theta_* - \lambda(\widehat{\Sigma}+\lambda I)^{-1}\theta_*$, it follows

$$\begin{aligned} B &= \|\mathbb{E}[\hat{\theta}_\lambda] - \theta_*\|^2_{\widehat{\Sigma}} \\ &= \lambda^2\theta_*^\top(\widehat{\Sigma}+\lambda I)^{-2}\widehat{\Sigma}\theta_*. \end{aligned}$$

For the variance term, using the fact that $\mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I$, we have

$$\begin{aligned} V &= \mathbb{E}\Big[\|\hat{\theta}_\lambda - \mathbb{E}[\hat{\theta}_\lambda]\|^2_{\widehat{\Sigma}}\Big] = \mathbb{E}\Big[\Big\|\frac{1}{n}(\widehat{\Sigma}+\lambda I)^{-1}\Phi^\top\varepsilon\Big\|^2_{\widehat{\Sigma}}\Big] \\ &= \mathbb{E}\Big[\frac{1}{n^2}\operatorname{tr}\Big(\varepsilon^\top\Phi(\widehat{\Sigma}+\lambda I)^{-1}\widehat{\Sigma}(\widehat{\Sigma}+\lambda I)^{-1}\Phi^\top\varepsilon\Big)\Big] \\ &= \mathbb{E}\Big[\frac{1}{n^2}\operatorname{tr}\Big(\Phi^\top\varepsilon\varepsilon^\top\Phi(\widehat{\Sigma}+\lambda I)^{-1}\widehat{\Sigma}(\widehat{\Sigma}+\lambda I)^{-1}\Big)\Big] = \frac{\sigma^2}{n}\operatorname{tr}\Big(\widehat{\Sigma}(\widehat{\Sigma}+\lambda I)^{-1}\widehat{\Sigma}(\widehat{\Sigma}+\lambda I)^{-1}\Big). \end{aligned}$$

The proposition follows by summing the bias and variance terms. ■

We can make the following observations:

- The result above is also a bias/variance decomposition with the bias term equal to $B = \lambda^2\theta_*^\top(\widehat{\Sigma}+\lambda I)^{-2}\widehat{\Sigma}\theta_*$, and the variance term equal to $V = \frac{\sigma^2}{n}\operatorname{tr}\big[\widehat{\Sigma}^2(\widehat{\Sigma}+\lambda I)^{-2}\big]$. They are plotted in Figure 3.3.

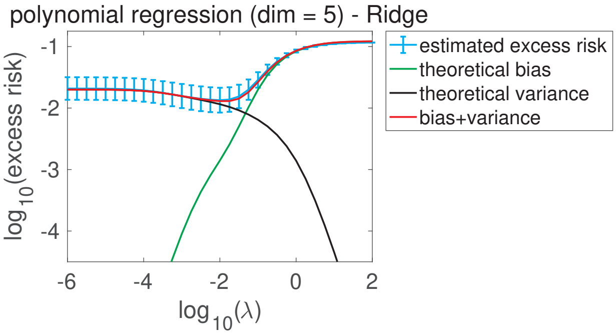


Figure 3.3: Polynomial regression (same set-up as Figure 3.2, with $n = 300$), with $k = 5$: bias/variance trade-offs for ridge regression as a function of $\lambda$. We can see the monotonicity of bias and variance with respect to $\lambda$ and the presence of an optimal choice of $\lambda$.

The bias/variance decomposition can be related to the decomposition in approximation error and estimation error presented in Section 2.3.2 and further developed in Chapter 4. The bias term is the part of the excess risk due to the regularization term constraining the proper estimation of the model. It plays the role of the approximation error, while the variance term characterizes the effect of the noise and plays the role of the estimation error.

- The bias term is increasing in $\lambda$ and equal to zero for $\lambda = 0$ if $\widehat{\Sigma}$ is invertible, while when $\lambda$ goes to infinity, the bias goes to $\theta_*^\top \widehat{\Sigma} \theta_*$. It is independent of $n$ and plays the role of the approximation error in the risk decomposition.
- The variance term is decreasing in $\lambda$, and equal to $\sigma^2 d/n$ for $\lambda = 0$ if $\widehat{\Sigma}$ is invertible, and converging to zero when $\lambda$ goes to infinity. It depends on $n$ and plays the role of the estimation error in the risk decomposition.
- The quantity $\operatorname{tr}\big[\widehat{\Sigma}^2(\widehat{\Sigma} + \lambda I)^{-2}\big]$ is called the "degrees of freedom", and is often considered as an implicit number of parameters. It can be expressed as $\sum_{j=1}^d \frac{\lambda_j^2}{(\lambda_j+\lambda)^2}$, where $(\lambda_j)_{j\in\{1,\dots,d\}}$ are the eigenvalues of $\widehat{\Sigma}$. This quantity will be very important in analyzing kernel methods in Chapter 7. Since the function $\mu \mapsto \mu^2/(\mu + \lambda^2)$ in increasing from 0 to 1, close to zero if $\mu \ll \lambda$, close to one if $\mu \gg \lambda$, the degrees of freedom provide a soft count of the number of eigenvalues that are larger than $\lambda$.
- Observe how this converges to the OLS estimator (when defined) as $\lambda \to 0$.
- In most cases, $\lambda = 0$ is not the optimal choice; that is, biased estimation (with controlled bias) is preferable to unbiased estimation. In other words, the mean square error is minimized for a biased estimator.

**Choice of $\lambda$.** Based on the expression for the risk, we can tune the regularization parameter $\lambda$ to obtain a potentially better bound than with the OLS (which corresponds to $\lambda = 0$ and the excess risk $\sigma^2 d/n$).

**Proposition 3.8 (choice of regularization parameter)** *With the choice of regularization parameter* $\lambda^* = \frac{\sigma \operatorname{tr}(\widehat{\Sigma})^{1/2}}{\|\theta_*\|_2\sqrt{n}}$, *we have*

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_{\lambda^*})\big] - \mathcal{R}^* \leqslant \frac{\sigma \operatorname{tr}(\widehat{\Sigma})^{1/2}\|\theta_*\|_2}{\sqrt{n}}.$$

**Proof** We have, using the fact that the eigenvalues of $(\widehat{\Sigma}+\lambda I)^{-2}\lambda\widehat{\Sigma}$ are less than $1/2$ (which is a simple consequence of $(\mu+\lambda)^{-2}\mu\lambda \leqslant 1/2 \Leftrightarrow (\mu+\lambda)^2 \geqslant 2\lambda\mu$ for all eigenvalues $\mu$ of $\widehat{\Sigma}$):

$$B = \lambda^2\theta_*^\top(\widehat{\Sigma}+\lambda I)^{-2}\widehat{\Sigma}\theta_* = \lambda\theta_*^\top(\widehat{\Sigma}+\lambda I)^{-2}\lambda\widehat{\Sigma}\theta_* \leqslant \frac{\lambda}{2}\|\theta_*\|_2^2.$$

Similarly, we have $V = \frac{\sigma^2}{n}\operatorname{tr}\big[\widehat{\Sigma}^2(\widehat{\Sigma}+\lambda I)^{-2}\big] = \frac{\sigma^2}{\lambda n}\operatorname{tr}\big[\widehat{\Sigma}\lambda\widehat{\Sigma}(\widehat{\Sigma}+\lambda I)^{-2}\big] \leqslant \frac{\sigma^2 \operatorname{tr}\widehat{\Sigma}}{2\lambda n}$. This leads to

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_{\lambda^*})\big] - \mathcal{R}^* \leqslant \frac{\lambda}{2}\|\theta_*\|_2^2 + \frac{\sigma^2 \operatorname{tr}\widehat{\Sigma}}{2\lambda n}. \tag{3.6}$$

Plugging in $\lambda^*$ (which was chosen to minimize the upper bound on $B+V$) gives the result.[4] ■

We can make the following observations:

- If we write $R = \max_{i\in\{1,\dots,n\}}\|\varphi(x_i)\|_2$, then we have

$$\operatorname{tr}(\widehat{\Sigma}) = \sum_{j=1}^{d}\widehat{\Sigma}_{jj} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\varphi(x_i)_j^2 = \frac{1}{n}\sum_{i=1}^{n}\|\varphi(x_i)\|_2^2 \leqslant R^2.$$

  Thus, the dimension $d$ plays no explicit role in the excess risk bound and could even be infinite (given that $R$ and $\|\theta_*\|_2$ remain finite). This type of bounds is called *dimension-free* bounds (see more details in Chapter 7).

  The number of parameters is not the only way to measure the generalization capabilities of a learning method, hence the need for explicit constants that depend on problem parameters.

- Comparing this bound with that of the OLS estimator, we see that it converges slower to 0 as a function of $n$ (from $n^{-1}$ to $n^{-1/2}$), but it has a milder dependence on the noise (from $\sigma^2$ to $\sigma$). The presence of a "fast" rate in $O(n^{-1})$ with a potentially large constant and of a "slow" rate $O(n^{-1/2})$ with a smaller constant will appear several times in this book.

  Depending on $n$ and the constants, the "fast" rate result is not always the best.

[4] We have used the property that for any vector $u$, any symmetric matrix $M$, and any symmetric positive semi-definite matrix $A$, $u^\top Mu \leqslant \|u\|_2^2 \cdot \lambda_{\max}(M)$ and $\operatorname{tr}(AM) \leqslant \operatorname{tr}(A)\cdot\lambda_{\max}(M)$.

- The value of $\lambda^*$ involves quantities that we typically do not know in practice (such as $\sigma$ and $\|\theta_*\|_2$). This is still useful to highlight the existence of some $\lambda$ with good predictions (which can be found by cross-validation, as presented in Section 2.1).
- Note here that the choice of $\lambda^* = \frac{\sigma\sqrt{\operatorname{tr}(\widehat{\Sigma})}}{\|\theta_*\|_2\sqrt{n}}$ is optimizing the *upper-bound* $\frac{\lambda}{2}\|\theta_*\|_2^2 + \frac{\sigma^2 \operatorname{tr}\widehat{\Sigma}}{2\lambda n}$, and is thus typically not optimal for the true expected risk.
- We can check the homogeneity of the various formulas by a basic dimensional analysis. We use the bracket notation to denote the unit. Then $[\lambda] \times [\theta]^2 = [y^2] = [\sigma^2]$ since $\lambda\|\theta\|_2^2$ appears in the same objective function as $y^2$ (or $\sigma^2$). Moreover, we have $[y] = [\sigma] = [\varphi][\theta]$, leading to $[\lambda] = [\varphi]^2$. The value of $\lambda$ suggested above has the dimension $\frac{[\varphi]\times[\sigma]}{[\theta]}$, which is indeed equal to $[\varphi]^2$. Similarly, we can check that the bias and variance terms have the correct dimensions.

**Choosing $\lambda$ in practice.** The regularization $\lambda$ is an example of a *hyper-parameter*. This term broadly refers to any quantity that influences the behavior of a machine learning algorithm and that is left to choose by the practitioner. While theory often offers guidelines and qualitative understanding on best choosing the hyper-parameters, their precise numerical value depends on quantities that are often difficult to know or even guess. In practice, we typically resort to validation and cross-validation.

**Exercise 3.6** *Compute the expected risk of the estimators obtained by regularizing by $\theta^\top \Lambda \theta$ instead of $\lambda\|\theta\|_2^2$, where $\Lambda \in \mathbb{R}^{d\times d}$ is a positive definite matrix.*

**Exercise 3.7** *(♦) We consider the leave-one-out estimator $\theta_\lambda^{-i} \in \mathbb{R}^d$ obtained, for each $i \in \{1,\dots,n\}$, by minimizing $\frac{1}{n}\sum_{j\neq i}(y_j - \theta^\top\varphi(x_j))^2 + \lambda\|\theta\|_2^2$. Given the matrix $H = \Phi(\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top \in \mathbb{R}^{n\times n}$, and its diagonal $h = \operatorname{diag}(H) \in \mathbb{R}^n$, show that*

$$\frac{1}{n}\sum_{i=1}^n (y_i - \varphi(x_i)^\top\theta_\lambda^{-i})^2 = \frac{1}{n}\|(I - \operatorname{Diag}(h))^{-1}(I-H)^\top y\|_2^2.$$

## 3.7 Lower-bound (♦)

To show a lower bound in the fixed design setting, we will consider only Gaussian noise, that is, $\varepsilon$ has a joint Gaussian distribution with mean zero and covariance matrix $\sigma^2 I$ (adding an extra assumption can only make the lower bound smaller). We follow the elegant and simple proof technique outlined by Mourtada (2019).

The only unknown in the model is the location of $\theta_*$. To make the dependence on $\theta_*$ explicit, we denote by $\mathcal{R}_{\theta_*}(\theta) - \mathcal{R}^*$ the excess risk (in the previous chapter, we were using the notation $\mathcal{R}_p$ to make the dependence on the distribution $p$ explicit), which is equal to

$$\mathcal{R}_{\theta_*}(\theta) - \mathcal{R}^* = \|\theta - \theta_*\|_{\widehat{\Sigma}}^2.$$

Our goal is to lower bound

$$\sup_{\theta_* \in \mathbb{R}^d} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)} \big[\mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon))\big] - \mathcal{R}^*,$$

over all functions $\mathcal{A}$ from $\mathbb{R}^n$ to $\mathbb{R}^d$ (these functions are allowed to depend on the observed deterministic quantities such as $\Phi$). Indeed, algorithms take $y = \Phi\theta_* + \varepsilon \in \mathbb{R}^n$ as an input and output a vector of parameters in $\mathbb{R}^d$.

The main idea, which is classical in the Bayesian analysis of learning algorithms, is to lower bound the supremum by the expectation with respect to some probability on $\theta_*$, called the prior distribution in Bayesian statistics. That is, we have, for any algorithm/estimator $\mathcal{A}$:

$$\sup_{\theta_* \in \mathbb{R}^d} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)} \mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon)) \ \geqslant \ \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)} \mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon)). \quad (3.7)$$

Here, we choose the normal distribution with mean 0 and covariance matrix $\frac{\sigma^2}{\lambda n} I$ as a prior distribution since this will lead to closed-form computations.

Using the expression of the excess risk (and ignoring the additive constant $\sigma^2 = \mathcal{R}^*$), we thus get the lower bound

$$\mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)} \big[\|\mathcal{A}(\Phi\theta_* + \varepsilon) - \theta_*\|^2_{\widehat{\Sigma}}\big],$$

which we need to minimize with respect to $\mathcal{A}$. By making $\theta_*$ random, we now have a joint Gaussian distribution for $(\theta_*, \varepsilon)$. The joint distribution of $(\theta_*, y) = (\theta_*, \Phi\theta_* + \varepsilon)$ is also Gaussian with mean zero and covariance matrix

$$\begin{pmatrix} \frac{\sigma^2}{\lambda n} I & \frac{\sigma^2}{\lambda n} \Phi^\top \\ \frac{\sigma^2}{\lambda n} \Phi & \frac{\sigma^2}{\lambda n} \Phi\Phi^\top + \sigma^2 I \end{pmatrix} = \frac{\sigma^2}{\lambda n} \begin{pmatrix} I & \Phi^\top \\ \Phi & \Phi\Phi^\top + n\lambda I \end{pmatrix}.$$

We need to perform an operation similar to computing the Bayes predictor in Chapter 2. This will be done by conditioning on $y$ by writing

$$\begin{aligned} \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)} \big[\|\mathcal{A}(\Phi\theta_* + \varepsilon) - \theta_*\|^2_{\widehat{\Sigma}}\big] &= \mathbb{E}_{(\theta_*, y)} \big[\|\mathcal{A}(y) - \theta_*\|^2_{\widehat{\Sigma}}\big] \\ &= \int_{\mathbb{R}^n} \Big( \int_{\mathbb{R}^d} \|\mathcal{A}(y) - \theta_*\|^2_{\widehat{\Sigma}} dp(\theta_* | y) \Big) dp(y). \end{aligned}$$

Thus, for each $y$, the optimal $\mathcal{A}(y)$ has to minimize $\int_{\mathbb{R}^d} \|\mathcal{A}(y) - \theta_*\|^2_{\widehat{\Sigma}} dp(\theta_* | y)$, which is exactly the posterior mean of $\theta_*$ given $y$. Indeed, the vector that minimizes the expected squared deviation is the expectation (exactly like when we computed the Bayes predictor for regression), here applied to the distribution $dp(\theta_* | y)$.

Since the joint distribution of $(\theta_*, y)$ is Gaussian with known parameters, we could use classical results about conditioning for Gaussian vectors (see Section 1.1.3). Still, we can also use the property that for Gaussian variables, the posterior mean given $y$ is equal to the posterior mode given $y$, that is, it can be obtained by maximizing the log-likelihood

$\log p(\theta_*, y)$ with respect to $\theta_*$. Up to constants and using independence of $\varepsilon$ and $\theta_*$, this log-likelihood is

$$-\frac{1}{2\sigma^2}\|\varepsilon\|^2 - \frac{\lambda n}{2\sigma^2}\|\theta_*\|_2^2 = -\frac{1}{2\sigma^2}\|y - \Phi\theta_*\|^2 - \frac{\lambda n}{2\sigma^2}\|\theta_*\|_2^2,$$

which is exactly (up to a sign and a constant) the ridge regression cost function. Thus, we have $\mathcal{A}^*(y) = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top y$, which is exactly the ridge regression estimator $\hat\theta_\lambda$, and we can compute the corresponding optimal risk, to get:

$$\begin{aligned}
& \inf_{\mathcal{A}} \sup_{\theta_* \in \mathbb{R}^d} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon))\big] - \mathcal{R}^* \\
\geqslant\ & \inf_{\mathcal{A}} \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon))\big] - \mathcal{R}^* \text{ using Eq. (3.7)}, \\
=\ & \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)}\ \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\mathcal{R}_{\theta_*}(\mathcal{A}^*(\Phi\theta_* + \varepsilon))\big] - \mathcal{R}^* \text{ using the reasoning above}, \\
=\ & \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\|\mathcal{A}^*(\Phi\theta_* + \varepsilon) - \theta_*\|_{\widehat\Sigma}^2\big] \text{ using the expression of the risk}, \\
=\ & \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\|(\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top(\Phi\theta_* + \varepsilon) - \theta_*\|_{\widehat\Sigma}^2\big] \\
& \qquad\qquad\qquad \text{using the closed-form expression of the OLS estimator}, \\
=\ & \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\|(\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top\varepsilon - n\lambda(\Phi^\top\Phi + n\lambda I)^{-1}\theta_*\|_{\widehat\Sigma}^2\big] \\
=\ & \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)}\big[\|-n\lambda(\Phi^\top\Phi + n\lambda I)^{-1}\theta_*\|_{\widehat\Sigma}^2\big] + \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\|(\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top\varepsilon\|_{\widehat\Sigma}^2\big] \\
& \qquad\qquad\qquad \text{by independence}, \\
=\ & \frac{\sigma^2}{n\lambda}(n\lambda)^2\frac{1}{n^2}\operatorname{tr}\big[(\widehat\Sigma + \lambda I)^{-2}\widehat\Sigma\big] + \frac{\sigma^2}{n}\operatorname{tr}\big[(\widehat\Sigma + \lambda I)^{-2}\widehat\Sigma^2\big] \\
=\ & \frac{\sigma^2}{n}\operatorname{tr}\big[(\widehat\Sigma + \lambda I)^{-1}\widehat\Sigma\big].
\end{aligned}$$

When $\Phi$ (and thus $\widehat\Sigma$) has full rank, this last expression tends to $\frac{\sigma^2}{n}\operatorname{tr}(I) = \frac{\sigma^2 d}{n}$ when $\lambda$ tends to zero (otherwise, it tends to $\frac{\sigma^2}{n}\operatorname{rank}(\Phi)$). This such shows that

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \mathbb{R}^d} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon))\big] - \mathcal{R}^* \geqslant \frac{\sigma^2 d}{n}.$$

This gives us a lower bound on performance, which exactly matches the upper bound obtained by OLS. In the general non-least-squares case, such results are significantly harder to show. See more general lower bounds in Chapter 15.

## 3.8 Random design analysis

In this section, we consider the regular random design setting, that is, both $x$ and $y$ are considered random, and each pair $(x_i, y_i)$ is assumed independent and identically distributed from a probability distribution $p$ on $\mathcal{X} \times \mathbb{R}$. We aim to show that the bound on the excess risk we have shown for the fixed design setting, namely $\sigma^2 d/n$, is still valid.

We will make the following assumptions regarding the joint distribution $p$, transposed from the fixed design setting to the random design setting:

- There exists a vector $\theta_* \in \mathbb{R}^d$ such that the relationship between input and output is for all $i$,
$$y_i = \varphi(x_i)^\top \theta_* + \varepsilon_i.$$
- The noise distribution of $\varepsilon_i \in \mathbb{R}$ is independent from $x_i$, and $\mathbb{E}[\varepsilon_i] = 0$ and with variance $\mathbb{E}[\varepsilon_i^2] = \sigma^2$ (and is the same for all $i$, as observations are i.i.d.).

With the assumption above, $\mathbb{E}[y_i|x_i] = \varphi(x_i)^\top \theta_*$, and thus, we perform empirical risk minimization where our class of functions includes the Bayes predictor. This situation is often referred to as the *well-specified* setting. The risk also has a simple expression:

**Proposition 3.9 (Excess risk for random design least-squares regression)** *Under the linear model above, for any $\theta \in \mathbb{R}^d$, the excess risk is equal to:*

$$\mathcal{R}(\theta) - \mathcal{R}^* = \|\theta - \theta_*\|_\Sigma^2,$$

*where $\Sigma := \mathbb{E}[\varphi(x)\varphi(x)^\top]$ is the (non-centered) covariance matrix, and $\mathcal{R}^* = \sigma^2$.*

**Proof** We have, for a pair $(x_0, y_0)$ sampled from the same distribution as all $(x_i, y_i)$, $i = 1, \dots, n$, with $\varepsilon_0$ the corresponding noise variable:

$$\begin{aligned}\mathcal{R}(\theta) &= \mathbb{E}\big[(y_0 - \theta^\top \varphi(x_0))^2\big] = \mathbb{E}\big[(\varphi(x_0)^\top \theta_* + \varepsilon_0 - \theta^\top \varphi(x_0))^2\big] \\ &= \mathbb{E}\big[(\varphi(x_0)^\top \theta_* - \theta^\top \varphi(x_0))^2\big] + \mathbb{E}\big[\varepsilon_0^2\big] = (\theta - \theta_*)^\top \Sigma (\theta - \theta_*) + \sigma^2,\end{aligned}$$

which leads to the desired result. ■

Note that the only difference with the fixed design setting is the replacement of $\widehat{\Sigma}$ by $\Sigma$. We can now express the risk of the OLS estimator.

**Proposition 3.10** *Under the linear model above, assuming $\widehat{\Sigma}$ is invertible, the expected excess risk of the OLS estimator is equal to*

$$\frac{\sigma^2}{n} \mathbb{E}\big[\operatorname{tr}(\Sigma \widehat{\Sigma}^{-1})\big].$$

**Proof** Since the OLS estimator is equal to $\hat{\theta} = \frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top y = \frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top(\Phi\theta_* + \varepsilon) = \theta_* + \frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top \varepsilon$, we have:

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\hat{\theta})] - \mathcal{R}^* &= \mathbb{E}\big[\big(\frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top \varepsilon\big)^\top \Sigma \big(\frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top \varepsilon\big)\big] \\ &= \mathbb{E}\big[\operatorname{tr}\big(\Sigma \big(\frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top \varepsilon\big)\big(\frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top \varepsilon\big)^\top\big)\big] = \frac{1}{n^2}\mathbb{E}\big[\operatorname{tr}\big(\Sigma \widehat{\Sigma}^{-1}\Phi^\top \varepsilon \varepsilon^\top \Phi \widehat{\Sigma}^{-1}\big)\big] \\ &= \frac{1}{n^2}\mathbb{E}\big[\operatorname{tr}\big(\Sigma \widehat{\Sigma}^{-1}\Phi^\top \mathbb{E}[\varepsilon\varepsilon^\top]\Phi \widehat{\Sigma}^{-1}\big)\big] = \mathbb{E}\Big[\frac{\sigma^2}{n^2}\operatorname{tr}\big(\Sigma \widehat{\Sigma}^{-1}\Phi^\top \Phi \widehat{\Sigma}^{-1}\big)\Big] \\ &= \mathbb{E}\Big[\frac{\sigma^2}{n}\operatorname{tr}(\Sigma \widehat{\Sigma}^{-1})\Big].\end{aligned}$$

■

Thus, to compute the expected risk of the OLS estimator, we need to compute $\mathbb{E}\big[\operatorname{tr}(\Sigma\widehat{\Sigma}^{-1})\big]$. One difficulty here is the potential non-invertibility of $\widehat{\Sigma}$. Under simple assumptions (e.g., $\varphi(x)$ has a density on $\mathbb{R}^d$), as soon as $n > d$, $\widehat{\Sigma}$ is almost surely invertible. However, its smallest eigenvalue can be very small. Additional assumptions are then needed to control it (see, e.g., Mourtada, 2019, Section 3).

**Exercise 3.8** *Show that for the random design setting with the same assumptions as Prop. 3.10, the expected risk of the ridge regression estimator is*

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_\lambda) - \mathcal{R}^*\big] = \lambda^2 \mathbb{E}\Big[\theta_*^\top(\widehat{\Sigma}+\lambda I)^{-1}\Sigma(\widehat{\Sigma}+\lambda I)^{-1}\theta_*\Big] + \frac{\sigma^2}{n}\mathbb{E}\Big[\operatorname{tr}\big[(\widehat{\Sigma}+\lambda I)^{-2}\widehat{\Sigma}\Sigma\big]\Big].$$

### 3.8.1 Gaussian designs

Suppose we assume that $\varphi(x)$ is normally distributed with mean 0 and covariance matrix $\Sigma$. In that case, we can directly compute the desired expectation by first considering $z = \Sigma^{-1/2}\varphi(x)$, which has a standard normal distribution (that is, with mean zero and identity covariance matrix), with the corresponding normalized design matrix $Z \in \mathbb{R}^{n\times d}$, and compute $\mathbb{E}\big[\operatorname{tr}(\Sigma\widehat{\Sigma}^{-1})\big] = n\mathbb{E}\big[\operatorname{tr}(Z^\top Z)^{-1}\big]$.

Note that $\mathbb{E}[Z^\top Z] = nI$, and by convexity of the function $M \mapsto \operatorname{tr}(M^{-1})$ on the cone of positive definite matrices, and using Jensen's inequality, we see that $\mathbb{E}\big[\operatorname{tr}(Z^\top Z)^{-1}\big] \geqslant \frac{d}{n}$ (here we have not used the Gaussian assumption). However, this bound is in the wrong direction (this often happens with Jensen's inequality).

It turns out that for Gaussians, the matrix $(Z^\top Z)^{-1}$ has a specific distribution, called the inverse Wishart distribution[5], with an expectation that can be computed exactly as $\mathbb{E}[(Z^\top Z)^{-1}] = \frac{1}{n-d-1}I$. Thus, we have: $\mathbb{E}\big[\operatorname{tr}(Z^\top Z)^{-1}\big] = \frac{d}{n-d-1}$ if $n > d+1$, thus leading to the expected excess risk of

$$\frac{\sigma^2 d}{n-d-1} = \frac{\sigma^2 d}{n}\frac{1}{1-(d+1)/n}.$$

See Breiman and Freedman (1983) for further details. Note here that for Gaussian designs, the expected risk is precisely equal to the expression above and that later in this book, we will only consider upper bounds. See also a further analysis in Section 12.2.3.

Overall, in the Gaussian case, we have an explicit non-asymptotic bound on the risk, which is equivalent to $\sigma^2 d/n$ when $n$ goes to infinity.

### 3.8.2 General designs (♦♦)

This last more technical section highlights how the Gaussian assumption can be avoided. The main idea is to show that with high probability, the lowest eigenvalue of $\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2}$

[5] See `https://en.wikipedia.org/wiki/Inverse-Wishart_distribution`.

is larger than some $1-t$, for some $t \in (0,1)$. Since the excess risk is $\frac{\sigma^2}{n}\operatorname{tr}(\Sigma\widehat{\Sigma}^{-1})$, this immediately shows that with high probability, the excess risk is less than $\frac{\sigma^2 d}{n}\frac{1}{1-t}$.

To obtain such results, more refined concentration inequalities are needed, such as described by Tropp (2012), Hsu et al. (2012), Oliveira (2013), and Lecué and Mendelson (2016). See complementary results by Mourtada (2019).

**Matrix concentration inequality.** We will use the matrix Bernstein bound, adapted from (Tropp, 2012, Theorem 1.4), already discussed in Section 1.2.6 and recalled here.

**Proposition 3.11 (Matrix Bernstein bound)** *Given $n$ independent symmetric matrices $M_i \in \mathbb{R}^{d\times d}$, such that for all $i \in \{1,\dots,n\}$, $\mathbb{E}[M_i]=0$, $\lambda_{\max}(M_i) \leqslant b$ almost surely, for all $t \geqslant 0$, we have:*

$$\mathbb{P}\Big(\lambda_{\max}\Big(\frac{1}{n}\sum_{i=1}^n M_i\Big) \geqslant t\Big)\Big) \leqslant d\cdot\exp\Big(-\frac{nt^2/2}{\tau^2+bt/3}\Big),$$

*for $\tau^2 = \lambda_{\max}\Big(\frac{1}{n}\sum_{i=1}^n \mathbb{E}[M_i^2]\Big)$.*

**Application to re-scaled covariance matrices.** We can now prove the following proposition that will give the desired high-probability bound for the excess risk with one extra assumption. Below, we will use the order between symmetric matrices, defined as $A \succcurlyeq B \Leftrightarrow B \preccurlyeq A \Leftrightarrow A-B$ positive semi-definite.

**Proposition 3.12** *Given $\Sigma = \mathbb{E}[\varphi(x)\varphi(x)^\top] \in \mathbb{R}^{d\times d}$, and i.i.d. observations $\varphi(x_1),\dots,\varphi(x_n) \in \mathbb{R}^d$, assume that, for some $\rho > 0$,*

$$\mathbb{E}\Big[\varphi(x)^\top\Sigma^{-1}\varphi(x)\varphi(x)\varphi(x)^\top\Big] \preccurlyeq \rho d\Sigma. \tag{3.8}$$

*For $\delta \in (0,1)$, if $n \geqslant 5\rho d\log\frac{d}{\delta}$, then with probability greater than $1-\delta$,*

$$\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \succcurlyeq \frac{1}{4}I. \tag{3.9}$$

Before giving the proof, note that from the discussion earlier, the bound in Eq. (3.9) leads to an excess risk less than $\frac{\sigma^2 d}{n}\frac{1}{1-t} = 4\frac{\sigma^2 d}{n}$ for $t = 3/4$. Moreover, without surprise, the bound is non-vacuous only for $n \geqslant d$ (and, in fact, because of the constraint on $n$, more than a constant times $d\log d$). The extra assumption in Eq. (3.8) can be interpreted as follows. We consider the random vector $z = \Sigma^{-1/2}\varphi(x) \in \mathbb{R}^d$, which is such that $\mathbb{E}[zz^\top] = I$ and $\mathbb{E}[\|z\|_2^2] = d$. The assumption in Eq. (3.8) is then equivalent to

$$\lambda_{\max}\Big(\mathbb{E}\Big[\|z\|^2 zz^\top\Big]\Big) \leqslant \rho d.$$

A sufficient condition is that almost surely $\|z\|_2^2 \leqslant \rho d$, that is, $\varphi(x)^\top\Sigma^{-1}\varphi(x) \leqslant \rho d$. Moreover, for a Gaussian distribution with zero mean for $z$, one can check as an exercise that $\rho = (1+2/d)$. Similar results will be obtained for ridge regression in Chapter 7.

**Proof** We consider the random symmetric matrix $M_i = I - z_i z_i^\top$, which is such that $\mathbb{E}[M_i] = 0$, $\lambda_{\max}(M_i) \leqslant 1$ almost surely, and $\mathbb{E}[M_i^2] = \mathbb{E}\big[\|z_i\|^2 z_i z_i^\top\big] - I$ with largest eigenvalue less than $\rho d$. We thus have for any $t \geqslant 0$, using Prop. 3.11:

$$\mathbb{P}\Big(\lambda_{\max}(I - \frac{1}{n} Z^\top Z) \geqslant t\Big) \leqslant d \cdot \exp\Big(-\frac{nt^2/2}{\rho d + t/3}\Big).$$

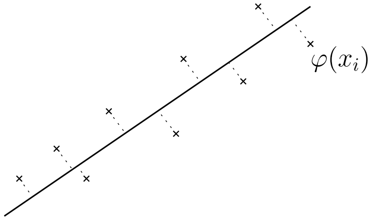
Thus, if $t$ is such that $\frac{nt^2}{2\rho d + 2t/3} \geqslant \log \frac{d}{\delta}$, then, with probability greater than $1-\delta$, we have $I - \Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \preccurlyeq tI$, that is, the desired result $\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \succcurlyeq (1-t)I$.

For $t = 3/4$, the condition becomes $n \geqslant (32\rho d/9 + 8/3) \log \frac{d}{\delta}$, which is implied by $n \geqslant 5\rho d \log \frac{d}{\delta}$ since we always $\rho \geqslant 1$. ■

## 3.9 Principal component analysis (♦)

Unsupervised dimension reduction is an effective way of reducing the number of features, either for computational efficiency (by storing and manipulating smaller feature vectors) or to avoid overfitting in a way complementary to ridge regularization. In this section, we present principal component analysis (PCA), which corresponds to looking for a low-dimensional subspace that contains approximately all feature vectors.

We consider $n$ feature vectors $\varphi(x_1), \dots, \varphi(x_n) \in \mathbb{R}^d$, with the corresponding design matrix $\Phi \in \mathbb{R}^{n \times d}$. PCA aims at finding a subspace of dimension $k$ such that all feature vectors are close to their orthogonal projections onto that subspace (see an illustration below for $d = 2$ and $k = 1$, where the goal is to minimize the sum of squares of all dotted segments).



In the formulation presented below, we consider a *linear* subspace (which contains 0), but it is common in practice to look for the optimal *affine* subspace (that may not contain 0), which can be done by first centering the data, that is, subtracting the mean from all feature vectors.

**Formulation as an eigenvalue problem.** We can parameterize the subspace (non-uniquely) by an orthonormal basis $V \in \mathbb{R}^{d \times k}$ such that $V^\top V = I$. Then each feature vector $\varphi(x_i)$, $i = 1, \dots, n$, has projection $VV^\top \varphi(x_i)$, and thus the design matrix of all projected vectors is $\Phi V V^\top$, and the optimal $V$ is found by minimizing:

$$\begin{aligned}
\|\Phi - \Phi V V^\top\|_{\mathrm{F}}^2 &= \operatorname{tr}\big[(\Phi - \Phi V V^\top)^\top (\Phi - \Phi V V^\top)\big] \\
&= \operatorname{tr}\big[\Phi^\top \Phi\big] + \operatorname{tr}\big[V V^\top \Phi^\top \Phi V V^\top\big] - 2 \operatorname{tr}\big[\Phi^\top \Phi V V^\top\big] \\
&= \operatorname{tr}\big[\Phi^\top \Phi\big] - \operatorname{tr}\big[V^\top \Phi^\top \Phi V\big].
\end{aligned}$$

Thus, minimizing $\|\Phi - \Phi V V^\top\|_{\mathrm{F}}^2$ is equivalent to maximizing $\operatorname{tr}\big[V^\top \Phi^\top \Phi V\big]$ with respect to an orthonormal matrix $V \in \mathbb{R}^{d\times k}$. Given an eigenvalue decomposition of the non-centered empirical covariance matrix $\widehat{\Sigma} = \frac{1}{n}\Phi^\top \Phi = U \operatorname{Diag}(\lambda) U^\top$, with $U \in \mathbb{R}^{d\times d}$ orthogonal and $\lambda$ a vector with non-increasing components, an optimal $V$ is obtained by taking the first $k$ columns of $U$, that is, a basis of the principal subspace of dimension $k$. Such a basis can be computed by various algorithms from numerical algebra (Golub and Loan, 1996). See Exercise 3.9 for a simple alternative optimization algorithm.

**Exercise 3.9** *Given $\Phi \in \mathbb{R}^{n\times d}$, we consider minimizing $\|\Phi - AD\|_{\mathrm{F}}^2$ with respect to $D \in \mathbb{R}^{k\times d}$ and $A \in \mathbb{R}^{n\times k}$. Show that the optimal solution is such that $AD$ is the data matrix after performing principal component analysis. Show that an alternating minimization algorithm that iteratively minimizes $\|\Phi - AD\|_{\mathrm{F}}^2$ with respect to $A$ and then $D$, converges to the global optimum for almost all initializations $D$, and compute the corresponding updates.*

**Exercise 3.10 (K-means clustering)** *Given $\Phi \in \mathbb{R}^{n\times d}$, we consider minimizing $\|\Phi - AD\|_{\mathrm{F}}^2$ with respect to $D \in \mathbb{R}^{k\times d}$ and $A \in \{0,1\}^{n\times k}$ such that each row of $A$ sums to one. Compute the updates of an alternative optimization algorithm that minimizes $\|\Phi - AD\|_{\mathrm{F}}^2$.*

**PCA and least-squares regression (♦♦).** While regularization is a common way to avoid overfitting for least-squares (as shown in Section 3.6), performing PCA and then (unregularized) ordinary least-squares provides an alternative with similar behavior. That is, we now consider the feature vector $\Phi V \in \mathbb{R}^{n\times k}$, and minimize $\|y - \Phi V \eta\|_2^2$ with respect to $\eta \in \mathbb{R}^k$, with solution $\eta = (V^\top \Phi^\top \Phi V)^{-1} V^\top \Phi^\top y$, leading to the prediction vector $\Phi V \eta = \Phi V (V^\top \Phi^\top \Phi V)^{-1} V^\top \Phi^\top y \in \mathbb{R}^n$.

If we assume the linear model $y = \Phi\theta_* + \varepsilon$ like in Section 3.6, we have:

$$\begin{aligned}
\frac{1}{n}\mathbb{E}_\varepsilon\big[\|\Phi V \eta - \Phi\theta_*\|_2^2\big] &= \frac{\sigma^2 k}{n} + \frac{1}{n}\big\|\Phi V (V^\top \Phi^\top \Phi V)^{-1} V^\top \Phi^\top \Phi \theta_* - \Phi\theta_*\big\|_2^2 \\
&= \frac{\sigma^2 k}{n} + \theta_*^\top \widehat{\Sigma}\theta_* - \theta_*^\top \widehat{\Sigma} V (V^\top \widehat{\Sigma} V)^{-1} \widehat{\Sigma}\theta_* \\
&= \frac{\sigma^2 k}{n} + \theta_*^\top (I - V V^\top)\widehat{\Sigma}(I - V V^\top)\theta_*,
\end{aligned}$$

using the fact that the columns of $V$ are eigenvector of $\widehat{\Sigma}$. We can now use the upper-bound

$$\leqslant \frac{\sigma^2 k}{n} + \|\theta_*\|_2^2 \lambda_{k+1},$$

where $\lambda_{k+1}$ is the $(k+1)$-th largest eigenvalue of $\widehat{\Sigma}$, which is less than $1/(k+1)$ times $\mathrm{tr}[\widehat{\Sigma}]$ (the sum of all eigenvalues). Thus, the excess risk of OLS after PCA is less than $\frac{\sigma^2 k}{n} + \|\theta_*\|_2^2 \frac{\mathrm{tr}[\widehat{\Sigma}]}{k}$, which is similar to Eq. (3.6). A good value of $k$ is then the closest integer to $\|\theta_*\|_2 \cdot \mathrm{tr}[\widehat{\Sigma}]\sqrt{n}/\sigma$, leading to, up to constants, the same excess risk than for ridge regression.

## 3.10 Conclusion

In this chapter, we have considered the simplest machine learning set-up, that is, square loss and prediction functions linearly parameterized by a finite-dimensional parameter. This simplest setup led to estimation algorithms based on numerical linear algebra (solving linear systems) and a statistical analysis based on simple probabilistic arguments (mostly variance computations). In particular, we highlighted the importance of regularization, which allows good predictive performance with high-dimensional features through dimension-free bounds.

Going beyond the square loss will require iterative algorithms based on optimization (presented in Chapter 5), and a more refined statistical analysis with deeper probabilistic tools (presented in Chapter 4).

# Part II

# Generalization bounds for learning algorithms

# Chapter 4

# Empirical risk minimization

**Chapter summary**

- Convexification of the risk: for binary classification, optimal predictions can be achieved with convex surrogates.
- Risk decomposition: the risk can be decomposed into the sum of the approximation and estimation errors.
- Rademacher complexity: To study estimation errors and compute expected uniform deviations of real-valued outputs, Rademacher complexities are a very flexible and powerful tool that allows obtaining dimension-independent concentration inequalities.
- Relationship with asymptotic statistics: classical asymptotic results provide a finer picture of the behavior of empirical risk minimization as they provide asymptotic limits of performance as a well-defined constant times $1/n$, but they may not, in general, characterize small-sample effects.

As outlined in Chapter 2, given a joint distribution $p$ on $\mathcal{X} \times \mathcal{Y}$, and $n$ independent and identically distributed observations from $p$, our goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ with minimum risk $\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))]$, or equivalently minimum expected excess risk:

$$\mathcal{R}(f) - \mathcal{R}^* = \mathcal{R}(f) - \inf_{g \text{ measurable}} \mathcal{R}(g).$$

In this chapter, we will consider methods based on empirical risk minimization. Before looking at the necessary probabilistic tools, we will first show how problems where the output space is not a vector space, such as binary classification with $\mathcal{Y} = \{-1, 1\}$, can be reformulated as real-valued outputs, with so-called "convex surrogates" of loss functions.

## 4.1 Convexification of the risk

In this section, for simplicity, we focus on binary classification where $\mathcal{Y} = \{-1, 1\}$ with the 0-1 loss, but many of the concepts extend to the more general structured prediction set-up (see Chapter 13).

As our goal is to estimate a binary-valued function, the first idea that comes into mind is to minimize the empirical risk over a hypothesis space of binary-valued functions $f$ (or equivalently, the subsets of $\mathcal{X}$ by considering the set $\{x \in \mathcal{X}, f(x) = 1\}$). However, this approach leads to a combinatorial problem that can be computationally intractable. Moreover, how to control the capacity (i.e., how to regularize) for these types of hypothesis spaces needs to be clarified. Learning a real-valued function instead through the framework of convex surrogates simplifies and overcomes this problem as it convexifies the problem. Classical penalty-based regularization techniques can then be used for theoretical analysis (this chapter) and algorithms (Chapter 5).

This choice of treating classification problems through real-valued prediction functions allows us to avoid introducing Vapnik-Chervonenkis dimensions (see Vapnik and Chervonenkis, 2015) to obtain general convergence results for empirical risk minimization in this chapter, where we will use the generic tool of Rademacher complexities in Section 4.5.

Instead of learning $f : \mathcal{X} \to \{-1, 1\}$, we will thus learn a function $g : \mathcal{X} \to \mathbb{R}$ and define $f(x) = \text{sign}(g(x))$ where

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a = 0 \\ -1 & \text{if } a < 0. \end{cases}$$

The convention $\text{sign}(0) = 0$ implies that the prediction can never be correct when $g(x) = 0$. Within our context, for maximally ambiguous observations, this corresponds to choosing none of the two labels (other conventions consider taking $+1$ or $-1$ uniformly at random).

The risk of the function $f = \text{sign} \circ g$, still denoted $\mathcal{R}(g)$ (⚠ note the slight overloading of notations $\mathcal{R}(g) = \mathcal{R}(\text{sign} \circ g)$), is then equal to:

$$\mathcal{R}(g) = \mathbb{P}(\text{sign}(g(x)) \neq y) = \mathbb{E}[1_{\text{sign}(g(x)) \neq y}] = \mathbb{E}[1_{yg(x) \leqslant 0}] = \mathbb{E}\big[\Phi_{0-1}(yg(x))\big],$$

where $\Phi_{0-1} : \mathbb{R} \to \mathbb{R}$, with $\Phi_{0-1}(u) = 1_{u \leqslant 0}$ is called the "margin-based" 0-1 loss function or simply the 0-1 loss function.

⚠ Note the slightly overloaded notation above where the 0-1 loss function is defined on $\mathbb{R}$, compared to the 0-1 loss function from Chapter 2, which is defined on $\{-1, 1\} \times \{-1, 1\}$.

In practice, for empirical risk minimization, we then minimize with respect to the function $g : \mathcal{X} \to \mathbb{R}$ the corresponding empirical risk $\frac{1}{n} \sum_{i=1}^{n} \Phi_{0-1}(y_i g(x_i))$. The function $\Phi_{0-1}$ is not continuous (and thus also non-convex) and leads to difficult optimization problems.

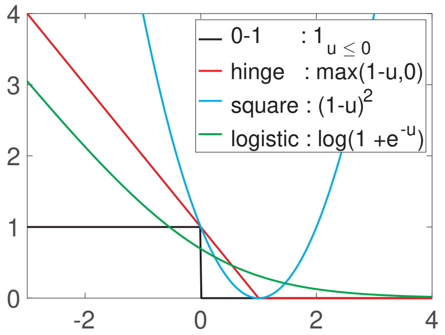


Figure 4.1: Classical convex surrogates for binary classification with the 0-1 loss.

### 4.1.1 Convex surrogates

A key concept in machine learning is the use of *convex surrogates*, where we replace $\Phi_{0-1}$ by another function $\Phi$ with better numerical properties (all will be convex). See classic examples below, plotted in Figure 4.1.

Instead of minimizing the classical risk $\mathcal{R}(g)$ or its empirical version, one then minimizes the "$\Phi$-risk" (and its empirical version) defined as

$$\mathcal{R}_\Phi(g) = \mathbb{E}[\Phi(yg(x))].$$

In this context, the function $g$ is sometimes called the *score function*.

The critical question we tackle in this section is: does it make sense to convexify the problem? In other words, does it lead to good predictions for the 0-1 loss?

**Classical examples.** We first review the primary examples used in practice:

- **Quadratic / square loss:** $\Phi(u) = (u-1)^2$, leading to, since we have $y^2 = 1$, $\Phi(yg(x)) = (y - g(x))^2 = (g(x) - y)^2$. We get back least-squares, ignore that the labels have to belong to $\{-1, 1\}$, and take the sign of $g(x)$ for the prediction. Note the overpenalization for a large positive value of $yg(x)$ that will not be present for the other losses below (which are non-increasing).
- **Logistic loss:** $\Phi(u) = \log(1 + e^{-u})$, leading to

$$\Phi(yg(x)) = \log(1 + e^{-yg(x)}) = -\log\left(\frac{1}{1 + e^{-yg(x)}}\right) = -\log(\sigma(yg(x))),$$

where: $\sigma(v) = \frac{1}{1+e^{-v}}$ is the sigmoid function. Note the link with maximum likelihood estimation, where we define the model through

$$\mathbb{P}(y = 1|x) = \sigma(g(x)) \text{ and } \mathbb{P}(y = -1|x) = \sigma(-g(x)) = 1 - \sigma(g(x)).$$

The risk is then the negative conditional log-likelihood $\mathbb{E}[-\log p(y|x)]$. It is also often called the cross-entropy loss.[1] See more details about probabilistic methods in Chapter 14.

[1] See https://en.wikipedia.org/wiki/Logistic_regression for details.

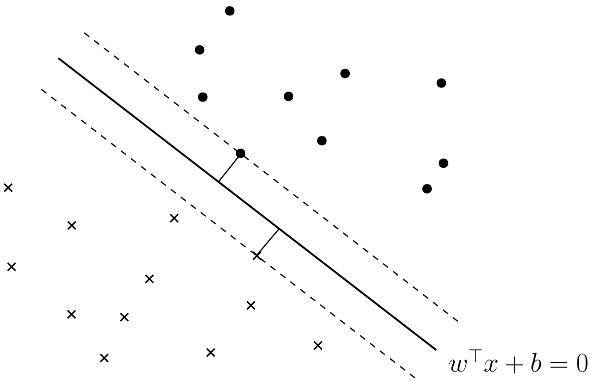
- **Hinge loss:** $\Phi(u) = \max(1-u, 0)$. With linear predictors, this leads to the support vector machine, and $yg(x)$ is often called the "margin" in this context. This loss has a geometric interpretation (see Section 4.1.2 below).[2]
- **Squared hinge loss:** $\Phi(u) = \max(1-u, 0)^2$. This is a smooth counterpart to the regular hinge loss.
- **Exponential loss:** $\Phi(u) = \exp(-u)$. This loss is often used within the boosting framework presented in Section 10.3, in particular through the Adaboost algorithm (Section 10.3.4).

This section analyzes precisely how replacing the 0-1 loss with convex surrogates still leads to optimal predictions. This allows us to only focus on *real-valued prediction functions* in the rest of this book. We will consider loss function $\ell(y, f(x))$ that will be the square loss $(y - f(x))^2$ for regression, and any of the ones above for binary classification, that is, $\Phi(yf(x))$. We will consider alternatives and extensions in Chapter 13.

### 4.1.2 Geometric interpretation of the support vector machine (♦)

Given its historical importance, this section provides a geometrical perspective on the hinge loss to highlight why it leads to a learning architecture called the "support vector machine" (SVM). We consider $n$ observations $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$, for $i = 1, \dots, n$.

**Separable data (Vapnik and Chervonenkis, 1964).** We first assume that the data are separable by an affine hyperplane, that is, there exist $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that for all $i \in \{1, \dots, n\}$, $y_i(w^\top x_i + b) > 0$. Among the infinitely many separating hyperplanes, we aim to select the one for which the closest points from the dataset are the farthest.



The distance from $x_i$ to the hyperplane $\{x \in \mathbb{R}^d,\ w^\top x + b = 0\}$ is equal to $\frac{|w^\top x_i + b|}{\|w\|_2}$, and thus, this minimal distance is

$$\min_{i \in \{1,\dots,n\}} \frac{y_i(w^\top x_i + b)}{\|w\|_2},$$

[2] See also https://en.wikipedia.org/wiki/Support_vector_machine for details.

and we thus aim at maximizing this quantity. Because of the invariance by rescaling (that is, we can multiply $w$ and $b$ by the same scalar constant without modifying the affine separator), this problem is equivalent to minimizing $\|w\|_2$ with the constraint that $\min_{i\in\{1,\dots,n\}} y_i(w^\top x_i + b) \geqslant 1$, and thus to the following problem:

$$\min_{w\in\mathbb{R}^d,\ b\in\mathbb{R}} \ \frac{1}{2}\|w\|_2^2 \ \text{ such that } \ \forall i \in \{1,\dots,n\},\ y_i(w^\top x_i + b) \geqslant 1. \tag{4.1}$$

**General data (Cortes and Vapnik, 1995).** When a hyperplane may not separate data, then we can introduce so-called "slack variables" $\xi_i \geqslant 0$, $i = 1,\dots,n$, allowing the constraint $y_i(w^\top x_i + b) \geqslant 1$ to be violated, by introducing the modified constraint $y_i(w^\top x_i + b) \geqslant 1 - \xi_i$ instead. The overall amount of slack is then minimized, leading to the following problem (with $C > 0$):

$$\min_{w\in\mathbb{R}^d,\ b\in\mathbb{R},\ \xi\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \xi_i \ \text{ such that } \ \forall i \in \{1,\dots,n\},\ y_i(w^\top x_i + b) \geqslant 1 - \xi_i,\ \xi_i \geqslant 0. \tag{4.2}$$

With $\lambda = \frac{1}{nC}$, the problem above is equivalent to

$$\min_{w\in\mathbb{R}^d,\ b\in\mathbb{R}} \ \frac{1}{n}\sum_{i=1}^n (1 - y_i(w^\top x_i + b))_+ + \frac{\lambda}{2}\|w\|_2^2,$$

which is exactly an $\ell_2$-regularized empirical risk minimization with the hinge loss for the prediction function $f(x) = w^\top x + b$.

**Lagrange dual and "support vectors" (♦).** The problem in Eq. (4.2) is a linearly constrained convex optimization problem and can be analyzed using Lagrangian duality (see, e.g., Boyd and Vandenberghe, 2004). We consider non-negative Lagrange multipliers $\alpha_i$ and $\beta_i$, $i \in \{1,\dots,n\}$, and the following Lagrangian:

$$\mathcal{L}(w,b,\xi,\alpha,\beta) \ = \ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i\big(y_i(w^\top x_i + b) - 1 + \xi_i\big) - \sum_{i=1}^n \beta_i \xi_i.$$

Minimizing with respect to $\xi \in \mathbb{R}^n$ leads to the equality constraints that for all $i \in \{1,\dots,n\}$, $\alpha_i + \beta_i = C$, while minimizing with respect to $b$ leads to the constraint $\sum_{i=1}^n y_i\alpha_i = 0$. Finally, minimizing with respect to $w$ can be done in closed form as $w = \sum_{i=1}^n \alpha_i y_i x_i$. Overall, this leads to the dual optimization problem:

$$\max_{\alpha\in\mathbb{R}^n} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_i^\top x_j \ \text{ such that } \ \sum_{i=1}^n y_i\alpha_i = 0 \text{ and } \forall i \in \{1,\dots,n\},\ \alpha_i \in [0, C]. \tag{4.3}$$

As we will show in Chapter 7 for all $\ell_2$-regularized learning problems with linear predictors, the optimization problem only depends on the dot-products $x_i^\top x_j$, $i,j = 1,\dots,n$. The optimal predictor can be written as a linear combination of input data points $x_i$,

$i = 1, \dots, n$. Moreover, for optimal primal and dual variables, the "complementary slackness" conditions for linear inequality constraints lead to $\alpha_i\big(y_i(w^\top x_i + b) - 1 + \xi_i\big) = 0$ and $(C - \alpha_i)\xi_i = 0$. This implies that $\alpha_i = 0$ as soon as $y_i(w^\top x_i + b) < 1$, and thus many of the $\alpha_i$'s are equal to zero, and the optimal predictor is a linear combination of only some of the data points $x_i$'s which are then called "support vectors". The sparsity of the $\alpha_i$'s can be leveraged computationally (Platt, 1998), but statistically, given that the number of support vectors is proportional to the number $n$ of observations (Steinwart, 2003), this sparsity alone cannot directly justify the potential superiority of the hinge loss over other convex surrogates.

### 4.1.3 Conditional Φ-risk and classification calibration (♦)

**From margin bounds to convergence to optimal predictions.** All of the convex surrogates presented in Section 4.1.1 are upper-bounds on the 0-1 loss or can be made so with rescaling. This simple fact allows us to get a variety of "margin bounds" where the 0-1 risk is upper-bounded by the Φ-risk. When the Φ-risk is equal to zero, which can only occur for problems with deterministic labels, this leads to a guarantee that the resulting classifier is the optimal one. In non-deterministic settings, however, the Φ-risk will be strictly positive, and while the margin bound shows that the error is controlled, it does not lead to guarantees to be close to the optimal predictions. In this section, we study the tools dedicated to obtaining such guarantees.

If we denote $\eta(x) = \mathbb{P}(y = 1|x) \in [0, 1]$, then we have, $\mathbb{E}[y|x] = 2\eta(x) - 1$, and, as seen in Chapter 2:

$$\mathcal{R}(g) = \mathbb{E}[\Phi_{0-1}(yg(x))] = \mathbb{E}[\mathbb{E}[1_{\text{sign}(g(x))\neq y}|x]] \geqslant \mathbb{E}[\min(\eta(x), 1 - \eta(x))] = \mathcal{R}^*,$$

and one best classifier is $f^*(x) = \text{sign}(2\eta(x) - 1) = \text{sign}(\mathbb{E}[y|x])$. Note that there are **many** potential other functions $g(x)$ than $2\eta(x) - 1$ so that $f^*(x) = \text{sign}(g(x))$ is optimal. The first (minor) reason is the arbitrary choice of prediction for $\eta(x) = 1/2$. The other reason is that $g(x)$ has to have the same sign as $2\eta(x) - 1$, which leads to many possibilities beyond $2\eta(x) - 1$.

This section aims to ensure that the minimizers of the expected Φ-risk lead to optimal predictions.

**Square loss.** Before moving on to general functions Φ, the square loss leads to simple arguments. Indeed, as seen in Chapter 2, the function minimizing the expected Φ-risk is then $g(x) = \mathbb{E}(y|x) = 2\eta(x) - 1$, and taking its sign leads to the optimal prediction. Thus, using the square loss for binary classification leads to the optimal prediction in the population case.

**General losses.** To study the impact of using the Φ-risk, we first look at the conditional risk for a given $x$ (as for the 0-1 loss, the function $g$ that will minimize the Φ-risk can be determined by looking at each $x$ separately).

**Definition 4.1 (conditional $\Phi$-risk)** *Let $g : \mathcal{X} \to \mathbb{R}$, we define the conditional $\Phi$-risk as*

$$\mathbb{E}[\Phi(yg(x))|x] = \eta(x)\Phi(g(x)) + (1-\eta(x))\Phi(-g(x)) \text{ which we denote } C_{\eta(x)}(g(x)),$$

*with $C_\eta(\alpha) = \eta\Phi(\alpha) + (1-\eta)\Phi(-\alpha)$.*

The least we can expect from a convex surrogate is that in the population case, where all $x$'s decouple, the optimal $g(x)$ obtained by minimizing the conditional $\Phi$-risk exactly leads to the same prediction as the Bayes predictor (at least when this prediction is unique). In other words, since the prediction is $\text{sign}(g(x))$, we want that for any $\eta \in [0,1]$ (below $\mathbb{R}_+^*$ is the set of strictly positive numbers, with a similar notation for $\mathbb{R}_-^*$):

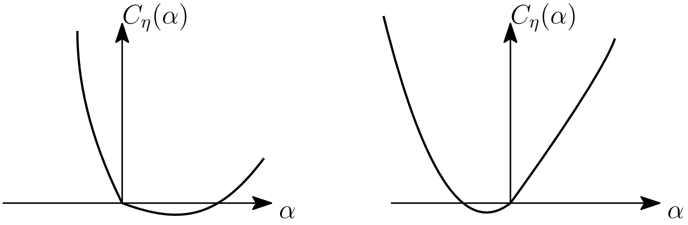
$$\text{(positive optimal prediction)} \qquad \eta > 1/2 \quad \Leftrightarrow \quad \arg\min_{\alpha\in\mathbb{R}} C_\eta(\alpha) \subset \mathbb{R}_+^* \tag{4.4}$$

$$\text{(negative optimal prediction)} \qquad \eta < 1/2 \quad \Leftrightarrow \quad \arg\min_{\alpha\in\mathbb{R}} C_\eta(\alpha) \subset \mathbb{R}_-^*. \tag{4.5}$$

A function $\Phi$ that satisfies these two statements is said *classification-calibrated*, or simply *calibrated*. It turns out that when $\Phi$ is convex, a simple sufficient and necessary condition is available:

**Proposition 4.1 (Bartlett et al., 2006)** *let $\Phi : \mathbb{R} \to \mathbb{R}$ convex. The surrogate function $\Phi$ is classification-calibrated if and only if $\Phi$ is differentiable at 0 and $\Phi'(0) < 0$.*

**Proof** Since $\Phi$ is convex, so is $C_\eta$ for any $\eta \in [0,1]$, and thus we simply consider left and right derivatives at zero to obtain conditions about the location of minimizers, with the two possibilities below (minimizer in $\mathbb{R}_+^*$ if and only if the right derivative at zero is strictly negative, and minimizer in $\mathbb{R}_-^*$ if and only if the left derivative at zero is strictly positive):



$$\arg\min_{\alpha\in\mathbb{R}} C_\eta(\alpha) \subset \mathbb{R}_+^* \quad \Leftrightarrow \quad (C_\eta)_+(0)' = \eta\Phi'_+(0) - (1-\eta)\Phi'_-(0) < 0 \tag{4.6}$$

$$\arg\min_{\alpha\in\mathbb{R}} C_\eta(\alpha) \subset \mathbb{R}_-^* \quad \Leftrightarrow \quad (C_\eta)_-(0)' = \eta\Phi'_-(0) - (1-\eta)\Phi'_+(0) > 0. \tag{4.7}$$

(a) Assume $\Phi$ is calibrated. By letting $\eta$ tend to $\frac{1}{2}+$ in Eq. (4.6), this leads to $(C_{1/2})_+(0)' = \frac{1}{2}\big[\Phi'_+(0) - \Phi'_-(0)\big] \leqslant 0$. Since $\Phi$ is convex, we always have the inequality $\Phi'_+(0) - \Phi'_-(0) \geqslant 0$. Thus, the left and right derivatives are equal, which implies that $\Phi$ is differentiable at 0. Then $C'_\eta(0) = (2\eta-1)\Phi'(0)$, and from Eq. (4.4) and Eq. (4.6), we need to have $\Phi'(0) < 0$.

(b) Assume $\Phi$ is differentiable at 0 and $\Phi'(0) < 0$, then $C'_\eta(0) = (2\eta-1)\Phi'(0)$; Eq. (4.4) and Eq. (4.5) are then direct consequences of Eq. (4.6) and Eq. (4.7).

■

Note that the proposition above excludes the convex surrogate $u \mapsto (-u)+ = \max\{-u, 0\}$, which is not differentiable at zero, but that all examples from Section 4.1.1 are calibrated.

We now assume that $\Phi$ is classification-calibrated and convex, that is, $\Phi$ is convex, $\Phi$ differentiable at 0, and $\Phi'(0) < 0$.

### 4.1.4 Relationship between risk and Φ-risk (♦♦)

Now that we know that for any $x \in \mathcal{X}$, minimizing $C_{\eta(x)}(g(x))$ with respect to $g(x)$ leads to the optimal prediction through $\mathrm{sign}(g(x))$, we would like to make sure that an explicit control of the excess $\Phi$-risk (which we aim to do with empirical risk minimization using tools from later sections) leads to an explicit control of the original excess risk. In other words, we are looking for an increasing function $H : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\mathcal{R}(g) - \mathcal{R}^* \leq H\big[\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*\big]$, where $\mathcal{R}_\Phi^*$ is the minimum possible $\Phi$-risk. The function $H$ is often called the *calibration function.* This section shows that this calibration is the identity for the hinge loss (support vector machine), while it can be the square root for smooth convex surrogates such as the square and logistic losses.

⚠ As opposed to the least-squares regression case, where the loss function used for testing is directly the one used within empirical risk minimization, there are two notions here: the testing *error* $\mathcal{R}(g)$, which is obtained after thresholding at zero the function $g$, and the quantity $\mathcal{R}_\Phi(g)$, which is sometimes called the testing *loss*.

We first start with a simple lemma expressing the excess risk, as well as an upper bound (adapted from Theorem 2.2 from Devroye et al., 1996), that we will need for comparison inequalities below:

**Lemma 4.1** *For any function $g : \mathcal{X} \to \mathbb{R}$, and for a Bayes predictor $g^* : \mathcal{X} \to \mathbb{R}$, we have:*

$$\mathcal{R}(g) - \mathcal{R}(g^*) = \mathbb{E}\big[1_{g(x)g^*(x)<0} \cdot |2\eta(x) - 1|\big].$$

*Moreover, we have* $\mathcal{R}(g) - \mathcal{R}(g^*) \leqslant \mathbb{E}\big[|2\eta(x) - 1 - g(x)|\big]$.

**Proof** We express the excess risk as:

$$\mathcal{R}(g) - \mathcal{R}(g^*) \quad = \mathbb{E}\big[\mathbb{E}\big[1_{\mathrm{sign}(g(x))\neq y} - 1_{\mathrm{sign}(g^*)(x)\neq y}\big|x\big]\big] \text{ by definition of the 0-1 loss.}$$

For any given $x \in \mathcal{X}$, we can look at the two possible cases for the signs of $\eta(x) - 1/2$ and $g(x)$ that lead to different predictions for $g$ and $g^*$, namely (a) $\eta(x) > 1/2$ and $g(x) < 0$, and (b) $\eta(x) < 1/2$ and $g(x) > 0$ (equality cases are irrelevant). For the first case the expectation with respect to $y$ is $\eta(x) - (1 - \eta(x)) = 2\eta(x) - 1$, while for the second case, we get $1 - 2\eta(x)$. By combining these two cases into the condition $g(x)g^*(x) < 0$ and the conditional expectation $|2\eta(x) - 1|$, we get the first result.

For the second result, we use the fact that if $g(x)g^*(x) < 0$, then, by splitting the cases in two (the first one being $\eta(x) > 1/2$ and $g(x) < 0$, the second one being $\eta(x) < 1/2$

and $g(x) > 0$), we get $|2\eta(x) - 1| \leqslant |2\eta(x) - 1 - g(x)|$, and thus the second result. ■

Note that for any function $b : \mathbb{R} \to \mathbb{R}$ that preserves the sign (that is $b(\mathbb{R}_+^*) \subset \mathbb{R}_+^*$ and $b(\mathbb{R}_-^*) \subset \mathbb{R}_-^*$), we have $\mathcal{R}(g) - \mathcal{R}(g^*) \leqslant \mathbb{E}[|2\eta(x) - 1 - b(g(x))|]$.

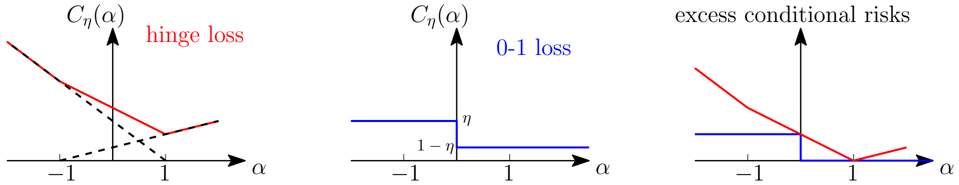
We see that the excess risk is the expectation of a quantity $|2\eta(x) - 1)| \cdot 1_{g(x)g^*(x)<0}$, which is equal to 0 if the classification through $g(x)$ is the same as the Bayes predictor and equal to $|2\eta(x) - 1|$ otherwise. The excess conditional $\Phi$-risk is the quantity

$$\eta(x)\Phi(g(x)) + (1 - \eta(x))\Phi(-g(x)) - \inf_{\alpha} \big\{\eta(x)\Phi(\alpha) + (1 - \eta(x))\Phi(-\alpha)\big\},$$

which, as a function of $g(x)$, is the deviation between a convex function (of $g(x)$) and its minimum value. We simply need to relate it to the quantity $|2\eta(x) - 1)| \cdot 1_{g(x)g^*(x)<0}$ above for any $x \in \mathcal{X}$ and take expectations.

Zhang (2004a) and Bartlett et al. (2006) propose a general framework. We will only consider the hinge loss and smooth losses for simplicity (they already cover all cases from Section 4.1.1).

- For the hinge loss $\Phi(\alpha) = (1 - \alpha)_+ = \max\{1 - \alpha, 0\}$, we can easily compute the minimizer of the conditional $\Phi$-risk (which leads to the minimizer of the $\Phi$-risk). Indeed, we need to minimize $\eta(x)(1 - \alpha)_+ + (1 - \eta(x))(1 + \alpha)_+$, which is a piecewise affine function with kinks at $-1$ and $1$, with a minimizer attained at $\alpha = 1$ for $\eta(x) > 1/2$ (see below), and symmetrically at $\alpha = -1$ for $\eta(x) < 1/2$, with a minimum conditional $\Phi$-risk equal to $2\min\{1 - \eta(x), \eta(x)\}$. The two excess risks are plotted below for the hinge loss and the 0-1 loss, for $\eta(x) > 1/2$, showing pictorially that the conditional excess $\Phi$-risk is greater than the excess risk.



  This leads to the calibration function $H(\sigma) = \sigma$ for the hinge loss.

  Note that when the Bayes risk is zero (but not in other cases), that is, $\eta(x) \in \{0, 1\}$ almost surely, then using the fact that the hinge loss is an upper-bound on the 0-1 loss is enough to show that the excess risk is less than the excess $\Phi$-risk (indeed, the two optimal risks $\mathcal{R}^*$ and $\mathcal{R}_\Phi^*$ are equal to zero).

- For the square loss $\Phi(v) = (v - 1)^2$, we can use directly Lemma 4.1, to get, using Jensen's inequality

$$\mathcal{R}(g) - \mathcal{R}(g^*) \leqslant \mathbb{E}\big[|2\eta(x) - 1 - g(x)|\big] \leqslant \big(\mathbb{E}\big[|2\eta(x) - 1 - g(x)|^2\big]\big)^{1/2} = \big(\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*\big)^{1/2},$$

  which is exactly a calibration result which we extend below to smooth losses.

- We consider smooth losses of the form (up to additive and multiplicative constants) $\Phi(v) = a(v) - v$, where $a(v) = \frac{1}{2}v^2$ for the quadratic loss, $a(v) = 2\log(e^{v/2} + e^{-v/2})$

for the logistic loss. We assume that $a$ is even, $a(0)=0$, $a$ is $\beta$-smooth (that is, as defined in Chapter 5, $a''(v) \leqslant \beta$ for all $v \in \mathbb{R}$). This implies[3] that for all $v \in \mathbb{R}$, $a(v) - \alpha v - \inf_{w \in \mathbb{R}} \big\{a(w) - \alpha w\big\} \geqslant \frac{1}{2\beta}|\alpha - a'(v)|^2$, leading to:

$$\begin{aligned}
\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^* &= \mathbb{E}\big[a(g(x)) - (2\eta(x)-1)g(x) - \inf_{w \in \mathbb{R}} \big\{a(w) - (2\eta(x)-1)w\big\}\big] \\
&\geqslant \frac{1}{2\beta}\mathbb{E}\big[\big|2\eta(x) - 1 - a'(g(x))\big|^2\big] \text{ by the property above,} \\
&\geqslant \frac{1}{2\beta}\big(\mathbb{E}\big[\big|2\eta(x) - 1 - a'(g(x))\big|\big]\big)^2 \text{ by Jensen's inequality,} \\
&\geqslant \frac{1}{2\beta}\big(\mathcal{R}(g) - \mathcal{R}^*\big)^2,
\end{aligned}$$

using Lemma 4.1, and the fact that $a'$ is sign-preserving (since $a'(0)=0$). This leads to the calibration function $H(\sigma) = \sqrt{2\sigma}$ for the square loss and $H(\sigma) = 2\sqrt{\sigma}$ for the logistic loss.

**Exercise 4.1** *(♦) Show that if $a^*$ is the Fenchel conjugate of $a$, then for any function $g : \mathcal{X} \to \mathbb{R}$, we have $a^*\big(\mathcal{R}(g) - \mathcal{R}^*\big) \leqslant \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*$.*

**Exercise 4.2** *(♦♦) We consider a convex function $\Phi : \mathbb{R} \to \mathbb{R}$ which is differentiable at zero with $\Phi'(0) < 0$. Define $G(z) = \Phi(0) - \inf_{\alpha \in \mathbb{R}} \big\{\frac{1+z}{2}\Phi(\alpha) + \frac{1-z}{2}\Phi(-\alpha)\big\}$. Show that $G$ is convex, $G(0) = 0$, and $G\big[\mathcal{R}(g) - \mathcal{R}^*\big] \leqslant \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*$ for any function $g : \mathcal{X} \to \mathbb{R}$. Compute $G$ for the exponential loss.*

We can make the following observations:

- For the (non-smooth) hinge loss, the calibration function is identity, so if the excess $\Phi$-risk goes to zero at a specific rate, the excess risk goes to zero at the same rate. In contrast, for smooth losses, the upper bound only ensures a (worse) rate with a square root. Therefore, when going from the excess $\Phi$-risk to the excess risk, that is, after thresholding the function $g$ at zero, the observed rates may be worse. However, as shown in Chapter 5, smooth losses can be easier to optimize. There is, thus, a trade-off between these two types of losses.
- Note that the noiseless case where $\eta(x) \in \{0,1\}$ (zero Bayes risk) leads to a stronger calibration function, as well as a series of intermediate "low-noise" conditions (see Bartlett et al., 2006, for details, as well as the exercise below).

**Exercise 4.3** *(♦) Assume that $|2\eta(x) - 1| > \varepsilon$ almost surely, for some $\varepsilon \in (0,1]$. Show that for any smooth convex classification calibrated function $\Phi : \mathbb{R} \to \mathbb{R}$ of the form $\Phi(v) = a(v) - v$ above, then we have $\mathcal{R}(g) - \mathcal{R}(g^*) \leqslant \frac{\varepsilon}{a^*(\varepsilon)}\big[\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*\big]$ for any function $g : \mathcal{X} \to \mathbb{R}$.*

[3] Using the Fenchel conjugate $a^* : \mathbb{R} \to \mathbb{R}$ which is $(1/\beta)$-strongly convex (see Chapter 5), we have: $a(v) - \alpha v - \inf_{w \in \mathbb{R}} \big\{a(w) - \alpha w\big\} = a(v) - \alpha v + a^*(\alpha) = a^*(\alpha) - a^*(a'(v)) - (\alpha - a'(v))(a^*)'(a'(v)) \geqslant \frac{1}{2\beta}|\alpha - a'(v)|^2$, where $a^*$ is the Fenchel conjugate of $a$ (Boyd and Vandenberghe, 2004).

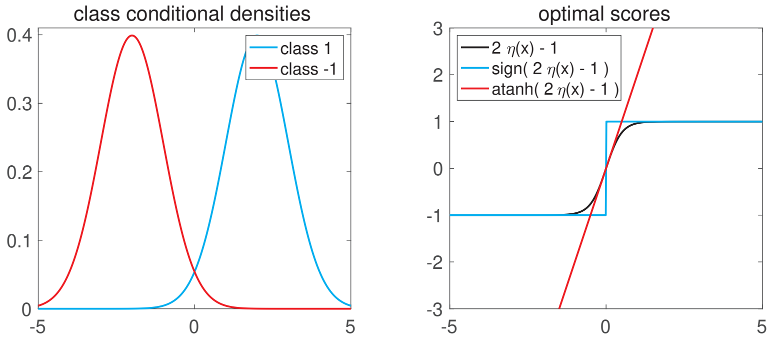


Figure 4.2: Optimal score functions for Gaussian class-conditional densities in one dimension. Left: conditional densities, right: optimal score functions for the square loss ($g^*(x) = 2\eta(x) - 1$), the hinge loss ($g^*(x) = \text{sign}(2\eta(x) - 1)$) and the logistic loss ($g^*(x) = \text{atanh}(2\eta(x) - 1)$).

**Impact on approximation errors (♦).** For the same classification problem, several convex surrogates can be used. While the Bayes classifier is always the same, that is, $f^*(x) = \text{sign}(2\eta(x)-1)$, the minimizer of the testing $\Phi$-risk will be different. For example, for the hinge loss, the minimizer $g(x)$ is exactly $\text{sign}(2\eta(x) - 1)$, while for losses of the form like above $\Phi(v) = a(v)-v$, we have $a'(g(x)) = 2\eta(x)-1$, and thus for the square loss $g(x) = 2\eta(x) - 1$, while for the logistic loss, one can check that $g(x) = \text{atanh}(2\eta(x) - 1)$ (hyperbolic arc tangent). See examples in Figure 4.2, with $\mathcal{X} = \mathbb{R}$ and Gaussian class conditional densities.

The choice of surrogates will have an impact since to attain the minimal $\Phi$-risk, different assumptions are needed on the class of functions used for empirical risk minimization, that is, $\text{sign}(2\eta(x) - 1)$ has to be in the class of functions we use (for the hinge loss), or $2\eta(x) - 1$ for the square loss, or $\text{atanh}(2\eta(x) - 1)$ for the logistic loss.

**Exercise 4.4** *For the logistic loss, show that for data generated with class-conditional densities of $x|y = 1$ and $x|y = -1$, which are Gaussians with the same covariance matrix, the function $g(x)$ minimizing the expected logistic loss is affine in $x$ (this model is often referred to as linear discriminant analysis). Provides an extension to the multi-class setting.*

**Beyond calibration and loss consistency.** The main property proved in this section is $\mathcal{R}(g) - \mathcal{R}^* \leq H\big[\mathcal{R}_\Phi(g) - \mathcal{R}^*_\Phi\big]$ for any prediction function $g : \mathcal{X} \to \mathbb{R}$, for a function $H$ which tends to zero at zero. When the space of functions chosen for $g$ is flexible enough to reach the minimizer of $\mathcal{R}_\Phi$, e.g., for kernel methods (Chapter 7) or neural networks with sufficiently many neurons (Chapter 9), then $g$ will reach the minimum risk $\mathcal{R}(g)$. Such properties will also be available for structured prediction in Chapter 13.

However, it is common in practice, in particular in high dimensions, to use a restricted class of models, in particular linear models, where reaching the minimum $\Phi$-risk is not possible anymore. In such setups, a more refined notion of consistency can be defined and studied, see, e.g., Long and Servedio (2013).

## 4.2 Risk minimization decomposition

We consider a family $\mathcal{F}$ of prediction functions $f : \mathcal{X} \to \mathbb{R}$. Empirical risk minimization aims to compute

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)),$$

with algorithms presented in Chapter 5. We consider loss functions that are defined for real-valued outputs even for binary classification problems through the use of surrogates presented in Section 4.1.1.

We can decompose the risk as follows into two terms:

$$\begin{aligned} \mathcal{R}(\hat{f}) - \mathcal{R}^* &= \Big\{ \mathcal{R}(\hat{f}) - \inf_{f' \in \mathcal{F}} \mathcal{R}(f') \Big\} + \Big\{ \inf_{f' \in \mathcal{F}} \mathcal{R}(f') - \mathcal{R}^* \Big\} \\ &= \quad \text{estimation error} \quad + \text{ approximation error.} \end{aligned}$$

A classic example is the situation where a subset of $\mathbb{R}^d$ parameterizes the family of functions, that is, $\mathcal{F} = \{f_\theta,\ \theta \in \Theta\}$, for $\Theta \subset \mathbb{R}^d$. This includes neural networks (Chapter 9) and the simplest case of linear models of the form $f_\theta(x) = \theta^\top \varphi(x)$, for a particular feature vector $\varphi(x)$ (such as in Chapter 3). We will use linear models with Lipschitz-continuous loss functions as a motivating example, most often with constraints or penalties on the $\ell_2$-norm $\|\theta\|_2$, but other norms can be considered as well (such as the $\ell_1$-norm in Chapter 8).

We now turn separately to the approximation and estimation errors.

## 4.3 Approximation error

The approximation error $\inf_{f \in \mathcal{F}} \mathcal{R}(f) - \mathcal{R}^*$ is deterministic and depends on the underlying distribution, and the class $\mathcal{F}$ of functions: the larger the class, the smaller the approximation error.

Bounding the approximation error requires assumptions on the Bayes predictor (sometimes also called the "target function") $f^*$, and hence on the testing distribution.

In this section, we will focus on $\mathcal{F} = \{f_\theta,\ \theta \in \Theta\}$, for $\Theta \subset \mathbb{R}^d$ (we will consider infinite-dimensions in Chapter 7), and convex Lipschitz-continuous losses, assuming that $\theta_*$ is the minimizer of $\mathcal{R}(f_\theta)$ over $\theta \in \mathbb{R}^d$, which is assumed to exist (typically, $\theta_*$ does not belong to $\Theta$). This implies that the approximation error decomposes into

$$\inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \mathcal{R}^* = \Big( \inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) \Big) + \Big( \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) - \mathcal{R}^* \Big).$$

- The second term $\inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) - \mathcal{R}^*$ is the incompressible approximation error coming from the chosen set of models $f_\theta$. For flexible models such as kernel methods

(Chapter 7) or neural networks (Chapter 9), this incompressible error can be made as small as desired.

- The function $\theta \mapsto \mathcal{R}(f_\theta) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta)$ is a positive function on $\mathbb{R}^d$, which can be typically upper bounded by a specific norm (or its square) $\Omega(\theta - \theta_*)$, and we can see the first term above $\inf\limits_{\theta \in \Theta} \mathcal{R}(f_\theta) - \inf\limits_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta)$ as a "distance" between $\theta_*$ and $\Theta$.

  For example, if the loss which is considered is $G$-Lipschitz-continuous with respect to the second variable (which is possible for regression or when using a convex surrogate for binary classification as presented in Section 4.1), we have,

  $$\mathcal{R}(f_\theta) - \mathcal{R}(f_{\theta'}) \quad = \quad \mathbb{E}\big[\ell(y, f_\theta(x)) - \ell(y, f_{\theta'}(x))\big] \leqslant G\mathbb{E}\big[|f_\theta(x) - f_{\theta'}(x)|\big],$$

  and thus, this second part of the approximation error is upper bounded by $G$ times the distance between $f_{\theta_*}$ and $\mathcal{F} = \{f_\theta, \ \theta \in \Theta\}$, for a particular pseudo-distance $d(\theta, \theta') = \mathbb{E}\big[|f_\theta(x) - f_{\theta'}(x)|\big]$.

  A classical example will be $f_\theta(x) = \theta^\top \varphi(x)$, and $\Theta = \{\theta \in \mathbb{R}^d, \ \|\theta\|_2 \leqslant D\}$, leading to the upper bound[4]

  $$\inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) \leqslant G \inf_{\|\theta\|_2 \leqslant D} \mathbb{E}\big[\|\varphi(x)\|_2\big] \cdot \|\theta - \theta_*\|_2 \leqslant G\, \mathbb{E}\big[\|\varphi(x)\|_2\big] (\|\theta_*\|_2 - D)_+,$$

  which is equal to zero if $\|\theta_*\|_2 \leqslant D$ (well-specified model).

  **Exercise 4.5** *Show that for $\Theta = \{\theta \in \mathbb{R}^d, \ \|\theta\|_1 \leqslant D\}$ ($\ell_1$-norm instead of the $\ell_2$-norm), we have*

  $$\inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) \leqslant G\, \mathbb{E}\big[\|\varphi(x)\|_\infty\big] (\|\theta_*\|_1 - D)_+.$$

  *Generalize to all norms.*

## 4.4 Estimation error

We will consider general techniques and apply them as illustrations to linear models with bounded $\ell_2$-norm by $D$ and $G$-Lipschitz-losses. See further applications in Chapter 7 and Chapter 9.

The estimation error is often decomposed using $g_{\mathcal{F}} \in \arg\min_{g \in \mathcal{F}} \mathcal{R}(g)$ the minimizer of the expected risk for our class of models and $\hat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f)$ the minimizer of

[4] The identity $\inf_{\|\theta\|_2 \leqslant D} \|\theta - \theta_*\|_2 = (\|\theta_*\|_2 - D)_+$ can be shown by looking for the optimal $\theta$ proportional to $\theta_*$ and optimizing with respect to the proportionality constant.

the empirical risk:

$$\begin{aligned}
\mathcal{R}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) &= \mathcal{R}(\hat{f}) - \mathcal{R}(g_{\mathcal{F}}) \\
&= \big\{\mathcal{R}(\hat{f}) - \widehat{\mathcal{R}}(\hat{f})\big\} + \big\{\widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(g_{\mathcal{F}})\big\} + \big\{\widehat{\mathcal{R}}(g_{\mathcal{F}}) - \mathcal{R}(g_{\mathcal{F}})\big\} \\
&\leqslant \sup_{f\in\mathcal{F}} \big\{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big\} + \big\{\widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(g_{\mathcal{F}})\big\} + \sup_{f\in\mathcal{F}} \big\{\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big\} \\
&\leqslant \sup_{f\in\mathcal{F}} \big\{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big\} + 0 + \sup_{f\in\mathcal{F}} \big\{\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big\} \text{ by definition of } \hat{f}. \quad (4.8)
\end{aligned}$$

This is often further upper-bounded by $2\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big|$. We can make the following observations:

- The key tool to remove the statistical dependence between $\widehat{\mathcal{R}}$ and $\hat{f}$ is to take a *uniform* bound.
- When $\hat{f}$ is not the global minimizer of $\widehat{\mathcal{R}}$ but satisfies $\widehat{\mathcal{R}}(\hat{f}) \leqslant \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + \varepsilon$, then the *optimization error* $\varepsilon$ has to be added to the bound above (see more details in Chapter 5).
- The uniform deviation grows with the "size" of $\mathcal{F}$, is a random quantity (because of its dependence on data), and usually decays with $n$. See the examples below.
- A key issue is that we need a *uniform control* for all $f \in \mathcal{F}$: with a single $f$, we could apply any concentration inequality to the random variable $\ell(y, f(x))$ to obtain a bound in $O(1/\sqrt{n})$; however, when controlling the maximal deviations over many functions $f$, there is always a small chance that one of these deviations get large. We thus need explicit control of this phenomenon, which we now tackle by first showing that we can focus on the expectation alone.

Since the estimation error is a random quantity, we need to bound it using probabilistic tools. This can be done either in high probability or in expectation. In the next section, we show how concentration inequalities allow us to focus on control in expectation.

### 4.4.1 Application of McDiarmid's inequality

Let $H(z_1, \dots, z_n) = \sup_{f\in\mathcal{F}} \big\{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big\}$, where the random variables $z_i = (x_i, y_i)$ are independent and identically distributed, and $\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i))$. We let $\ell_\infty$ denote the maximal absolute value of the loss functions for all $(x, y)$ in the support of the data generating distribution and $f \in \mathcal{F}$ (for most loss functions, this is a consequence of having bounded prediction functions).

For a single function $f \in \mathcal{F}$, we can control the deviation between $\widehat{\mathcal{R}}(f)$, which is an empirical average of bounded independent random variables, and its expectation $\mathcal{R}(f)$ through Hoeffding's inequality, presented in detail and proved in Section 1.2.1: for any $\delta \in (0, 1)$, with probability greater than $1 - \delta$,

$$\mathcal{R}(f) - \widehat{\mathcal{R}}(f) \leqslant \frac{\ell_\infty\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}.$$

Such a control can be extended beyond a single function $f$. When changing a single $z_i \in \mathcal{X} \times \mathcal{Y}$ into $z_i' \in \mathcal{X} \times \mathcal{Y}$, the deviation in $H$ is almost surely at most $\frac{2}{n}\ell_\infty$.[5] Thus, applying McDiarmid's inequality (see Section 1.2.2 in Chapter 1), with probability greater than $1-\delta$, we have:

$$H(z_1, \dots, z_n) - \mathbb{E}[H(z_1, \dots, z_n)] \leqslant \frac{\ell_\infty \sqrt{2}}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}}.$$

We thus only need to bound the expectation of $\sup_{f \in \mathcal{F}} \big\{ \mathcal{R}(f) - \widehat{\mathcal{R}}(f) \big\}$ and of the similar quantity $\sup_{f \in \mathcal{F}} \big\{ \widehat{\mathcal{R}}(f) - \mathcal{R}(f) \big\}$ (which will typically have the same bound), and add on top of it $\frac{\ell_\infty \sqrt{2}}{\sqrt{n}} \sqrt{\log \frac{2}{\delta}}$, to ensure a high-probability bound.[6]

We now provide a series of bounds to bound these expectations, from simple to more refined, culminating in Rademacher complexities in Section 4.5.

### 4.4.2 Easy case I: quadratic functions

We will show what happens with a quadratic loss function and an $\ell_2$-ball constraint. We remember that in this case $\ell(y, \theta^\top \varphi(x)) = (y - \theta^\top \varphi(x))^2$. From that, we get

$$\begin{aligned}\widehat{\mathcal{R}}(f) - \mathcal{R}(f) \;&=\; \theta^\top \Big( \frac{1}{n} \sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top - \mathbb{E}\big[\varphi(x)\varphi(x)^\top\big] \Big) \theta \\ &\qquad -2\theta^\top \Big( \frac{1}{n} \sum_{i=1}^n y_i \varphi(x_i) - \mathbb{E}\big[y\varphi(x)\big] \Big) + \Big( \frac{1}{n} \sum_{i=1}^n y_i^2 - \mathbb{E}\big[y^2\big] \Big).\end{aligned}$$

Hence, the supremum can be upper-bounded in closed form as

$$\begin{aligned}\sup_{\|\theta\|_2 \leqslant D} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)| \;&\leqslant\; D^2 \Big\| \frac{1}{n} \sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top - \mathbb{E}\big[\varphi(x)\varphi(x)^\top\big] \Big\|_{\rm op} \\ &\qquad +2D \Big\| \frac{1}{n} \sum_{i=1}^n y_i \varphi(x_i) - \mathbb{E}\big[y\varphi(x)\big] \Big\|_2 + \Big| \frac{1}{n} \sum_{i=1}^n y_i^2 - \mathbb{E}\big[y^2\big] \Big|,\end{aligned}$$

where $\|M\|_{\rm op}$ is the operator norm of the matrix $M$ defined as $\|M\|_{\rm op} = \sup_{\|u\|_2=1} \|Mu\|_2$ (for which we have $|u^\top M u| \leqslant \|M\|_{\rm op} \|u\|_2^2$ for any vector $u$).

Thus, to get a uniform bound, we simply need to upper-bound the three *non-uniform* expectations of deviations, and therefore of order $O(1/\sqrt{n})$, and we get an overall uniform deviation bound. This case gives the impression that it should be possible to get such a rate in $O(1/\sqrt{n})$ for other types of losses than the quadratic loss. However, closed-form calculations are impossible, so we must introduce new tools.

[5] For a fixed function $f \in \mathcal{F}$, only one term in the average is changed, with absolute value less than $\ell_\infty$, thus a deviation of at most $\frac{2}{n}\ell_\infty$. This can be extended to the supremum by a simple computation left as an exercise.

[6] When combining two bounds in probability, the union bound leads to the term $2/\delta$ instead of $1/\delta$, see Section 1.2.1.

**Exercise 4.6** *(♦) Provide an explicit bound on* $\sup_{\|\theta\|_2 \leqslant D} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$ *above, and compare it to using Rademacher complexities in Section 4.5. The concentration of averages of matrices from Section 1.2.6 can be used.*

⚠ Note that from now on, in the sections below, unless otherwise stated, we do not require the loss to be convex.

### 4.4.3 Easy case II: Finite number of models

We assume in this section that the loss functions are bounded between 0 and $\ell_\infty$, using the upper-bound $2\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big|$ on the estimation error, and the union bound:

$$\mathbb{P}\Big(\mathcal{R}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) \geqslant t\Big) \;\leqslant\; \mathbb{P}\Big(2\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\Big) \leqslant \sum_{f\in\mathcal{F}} \mathbb{P}\Big(2\big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\Big).$$

We have, for $f \in \mathcal{F}$ fixed, $\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i))$, and we can apply Hoeffding's inequality from Section 1.2.1 to bound each $\mathbb{P}\big(2\big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\big)$, leading to

$$\mathbb{P}\Big(\mathcal{R}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) \geqslant t\Big) \;\leqslant\; \sum_{f\in\mathcal{F}} 2\exp(-2n(t/2)^2/\ell_\infty^2) = 2|\mathcal{F}|\exp(-nt^2/(2\ell_\infty^2)).$$

Thus, by setting $\delta = 2|\mathcal{F}|\exp(-nt^2/2\ell_\infty^2)$, and finding the corresponding $t$, with probability greater than $1-\delta$, we get (using $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$):

$$\begin{aligned}\mathcal{R}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) \;&\leqslant\; t = \frac{\sqrt{2}\ell_\infty}{\sqrt{n}}\sqrt{\log\frac{2|\mathcal{F}|}{\delta}} = \frac{\sqrt{2}\ell_\infty}{\sqrt{n}}\sqrt{\log(2|\mathcal{F}|) + \log\frac{1}{\delta}} \\ &\leqslant\; \sqrt{2}\ell_\infty\sqrt{\frac{\log(2|\mathcal{F}|)}{n}} + \frac{\sqrt{2}\ell_\infty}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}.\end{aligned}$$

**Exercise 4.7** *(♦) In terms of expectation, show that (using the proof of the max of random variables from Section 1.2.4 in Chapter 2, which applies because bounded random variables are sub-Gaussian):*

$$\mathbb{E}\big[\mathcal{R}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{R}(f)\big] \leqslant 2\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big|\Big] \leqslant \ell_\infty\sqrt{\frac{2\log(2|\mathcal{F}|)}{n}}.$$

Thus, according to the bound, learning is possible when the logarithm $\log(|\mathcal{F}|)$ of the number of models is small compared to $n$. This is the first generic control of uniform deviations.

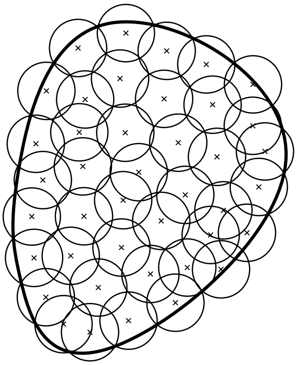
⚠ Note that this is only an upper bound, and learning is possible with infinitely many models (the most classical scenario). See below.

### 4.4.4 Beyond finitely many models through covering numbers (♦)

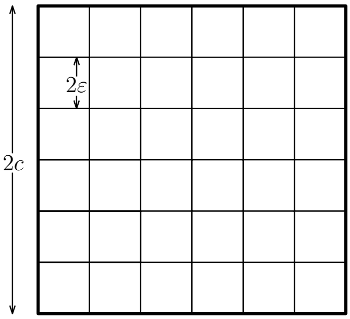
The simple idea behind covering numbers is to deal with function spaces with infinitely many elements by approximating them through a finite number of elements. This is often referred to as an "$\varepsilon$-net argument."

For simplicity, we assume that the loss functions are regular, for example, that they are $G$-Lipschitz-continuous with respect to their second argument.

**Covering numbers.** We assume there exists $m = m(\varepsilon)$ elements $f_1, \dots, f_m$ such that for any $f \in \mathcal{F}$, $\exists i \in \{1, \dots, m\}$ such that $d(f, f_i) \leqslant \varepsilon$. The minimal possible number $m(\varepsilon)$ is the *covering number* of $\mathcal{F}$ at precision $\varepsilon$. See an example below in two dimensions of a covering with Euclidean balls.



The covering number $m(\varepsilon)$ is a non-increasing function of $\varepsilon$. Typically, $m(\varepsilon)$ grows with $\varepsilon$ as a power $\varepsilon^{-d}$ when $\varepsilon \to 0$, where $d$ is the underlying dimension. Indeed, for the $\ell_\infty$-metric, if (in a certain parameterization) $\mathcal{F}$ is included in a ball of radius $c$ in the $\ell_\infty$-ball of dimension $d$, it can be easily covered by $(c/\varepsilon)^d$ cubes of length $2\varepsilon$. See below.



Given that all norms are equivalent in dimension $d$, we get the same dependence in $\varepsilon^{-d}$ of $m(\varepsilon)$ for all bounded subsets of a finite-dimensional vector space, and thus $\log m(\varepsilon)$ grows as $d \log \frac{1}{\varepsilon}$ when $\varepsilon$ tends to zero.

For some sets (e.g., all Lipschitz-continuous functions in $d$ dimensions) $\log m(\varepsilon)$ grows faster, for example as $\varepsilon^{-d}$. See, e.g., Wainwright (2019).

**$\varepsilon$-net argument.** Given a cover of $\mathcal{F}$, for all $f \in \mathcal{F}$, and with $(f_i)_{i\in\{1,\dots,m(\varepsilon)\}}$ the associated cover elements,

$$\begin{aligned} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| &\leqslant \big|\widehat{\mathcal{R}}(f) - \widehat{\mathcal{R}}(f_i)\big| + \big|\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)\big| + \big|\mathcal{R}(f_i) - \mathcal{R}(f)\big| \\ &\leqslant 2G\varepsilon + \sup_{i\in\{1,\dots,m(\varepsilon)\}} \big|\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)\big|. \end{aligned}$$

This implies that using bounds on the expectation of the maximum (Section 1.2.4), which apply because bounded random variables are sub-Gaussian (with the sub-Gaussianity parameter proportional to the almost sure bound):

$$\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big|\Big] \leqslant 2G\varepsilon + \mathbb{E}\Big[\sup_{i\in\{1,\dots,m(\varepsilon)\}} \big|\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)\big|\Big] \leqslant 2G\varepsilon + 2\ell_\infty\sqrt{\frac{2\log(2m(\varepsilon)))}{n}}.$$

Therefore, if $m(\varepsilon) \sim \varepsilon^{-d}$, ignoring constants, we need to balance $\varepsilon + \sqrt{d\log(1/\varepsilon)/n}$, which leads to, with a choice of $\varepsilon$ proportional to $1/\sqrt{n}$, to a rate proportional to $\sqrt{(d/n)\log(n)}$, which shows that the dependence in $n$ is also close to $1/\sqrt{n}$. Unfortunately, unless refined computations of covering numbers or more advanced tools (such as "chaining") are used, this often leads to a non-optimal dependence on dimension and/or number of observations (see, e.g., Wainwright, 2019, for examples of these refinements).

One powerful tool that allows sharp bounds at a reasonable cost is Rademacher complexities (Boucheron et al., 2005) or Gaussian complexities (Bartlett and Mendelson, 2002). In this chapter, we will focus on Rademacher complexity.

## 4.5 Rademacher complexity

We consider $n$ independent and identically distributed random variables $z_1, \dots, z_n \in \mathcal{Z}$, and a class $\mathcal{H}$ of functions from $\mathcal{Z}$ to $\mathbb{R}$. In our context, the space of functions is related to the learning problem as: $z = (x, y)$, and $\mathcal{H} = \{(x, y) \mapsto \ell(y, f(x)),\ f \in \mathcal{F}\}$.

Our goal in this section is to provide an upper-bound on $\sup_{f\in\mathcal{F}} \big\{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big\}$, which happens to be equal to

$$\sup_{h\in\mathcal{H}} \Big\{\mathbb{E}[h(z)] - \frac{1}{n}\sum_{i=1}^n h(z_i)\Big\},$$

where $\mathbb{E}[h(z)]$ denotes the expectation with respect to a variable having the same distribution as all $z_i$'s.

We denote by $\mathcal{D} = \{z_1, \dots, z_n\}$ the data. We define the *Rademacher complexity* of the class of functions $\mathcal{H}$ from $\mathcal{Z}$ to $\mathbb{R}$:

$$\mathrm{R}_n(\mathcal{H}) = \mathbb{E}_{\varepsilon,\mathcal{D}}\Big(\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i h(z_i)\Big), \tag{4.9}$$

where $\varepsilon \in \mathbb{R}^n$ is a vector of independent Rademacher random variable (that is, taking values $-1$ or $1$ with equal probabilities), which is also independent of $\mathcal{D}$. It is a deterministic quantity that only depends on $n$ and $\mathcal{H}$.

In words, the Rademacher complexity is equal to the expectation of the maximal dot-product between values of a function $h$ at the observations $z_i$ and random labels. It measures the "capacity" of the set of functions $\mathcal{H}$. We will see later that it can be computed (or simply upper-bounded) in many interesting cases, leading to powerful bounds. The term "Rademacher average" is also commonly used.

⚠ Be careful with the two notations $\mathrm{R}_n(\mathcal{H})$ (Rademacher complexity) and $\mathcal{R}(f)$ (risk of the prediction function $f$), not to be confused with the feature norm $R$ often used with linear models.

**Exercise 4.8** *Show the following properties of Rademacher complexities (see Bartlett and Mendelson, 2002, for more details):*

*(a)* *If* $\mathcal{H} \subset \mathcal{H}'$, *then* $\mathrm{R}_n(\mathcal{H}) \leqslant \mathrm{R}_n(\mathcal{H}')$.

*(b)* $\mathrm{R}_n(\mathcal{H} + \mathcal{H}') = \mathrm{R}_n(\mathcal{H}) + \mathrm{R}_n(\mathcal{H}')$.

*(c)* *If* $\alpha \in \mathbb{R}$, $\mathrm{R}_n(\alpha\mathcal{H}) = |\alpha| \mathrm{R}_n(\mathcal{H})$.

*(d)* *If* $h_0 : \mathcal{Z} \to \mathbb{R}$, $\mathrm{R}_n(\mathcal{H} + \{h_0\}) = \mathrm{R}_n(\mathcal{H})$.

*(e)* $\mathrm{R}_n(\mathcal{H}) = \mathrm{R}_n(\text{convex hull}(\mathcal{H}))$.

**Exercise 4.9 (Massart's lemma)** *If* $\mathcal{H} = \{h_1, \dots, h_m\}$, *and almost surely we have the bound* $\frac{1}{n}\sum_{i=1}^n h_j(x_i)^2 \leqslant R^2$ *for all* $j \in \{1, \dots, m\}$, *then the Rademacher complexity of the class of functions* $\mathcal{H}$ *satisfies* $\mathrm{R}_n(\mathcal{H}) \leqslant \sqrt{\frac{2\log m}{n}} R$.

### 4.5.1 Symmetrization

First, we relate the Rademacher complexity to the uniform deviation through a general "symmetrization" property, which shows that the Rademacher complexity directly controls the expected uniform deviation.

**Proposition 4.2 (symmetrization)** *Given the Rademacher complexity of* $\mathcal{H}$ *defined in Eq. (4.9), we have:*

$$\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n h(z_i) - \mathbb{E}[h(z)]\bigg)\bigg] \leqslant 2\mathrm{R}_n(\mathcal{H})\ ,\ \mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\mathbb{E}[h(z)] - \frac{1}{n}\sum_{i=1}^n h(z_i)\bigg)\bigg] \leqslant 2\mathrm{R}_n(\mathcal{H}).$$

**Proof** (♦) Let $\mathcal{D}' = \{z'_1, \dots, z'_n\}$ be an independent copy of the data $\mathcal{D} = \{z_1, \dots, z_n\}$. Let $(\varepsilon_i)_{i\in\{1,\dots,n\}}$ be i.i.d. Rademacher random variables, which are also independent of $\mathcal{D}$ and $\mathcal{D}'$. Using that for all $i$ in $\{1, \dots, n\}$, $\mathbb{E}[h(z'_i)|\mathcal{D}] = \mathbb{E}[h(z)]$, we have:

$$\begin{aligned}
\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\mathbb{E}[h(z)] - \frac{1}{n}\sum_{i=1}^n h(z_i)\bigg)\bigg] &= \mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n \mathbb{E}[h(z'_i)|\mathcal{D}] - \frac{1}{n}\sum_{i=1}^n h(z_i)\bigg)\bigg] \\
&= \mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n \mathbb{E}\big[h(z'_i) - h(z_i)\big|\mathcal{D}\big]\bigg)\bigg],
\end{aligned}$$

by definition of the independent copy $\mathcal{D}'$. Then

$$\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\mathbb{E}[h(z)]-\frac{1}{n}\sum_{i=1}^n h(z_i)\bigg)\bigg]\leqslant\mathbb{E}\bigg[\mathbb{E}\bigg(\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n\big[h(z_i')-h(z_i)\big]\bigg)\bigg|\mathcal{D}\bigg)\bigg],$$

using that the supremum of the expectation is less than the expectation of the supremum. Thus, by the towering law of expectation, we get

$$\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\mathbb{E}[h(z)]-\frac{1}{n}\sum_{i=1}^n h(z_i)\bigg)\bigg]\leqslant\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n\big[h(z_i')-h(z_i)\big]\bigg)\bigg].$$

We can now use the symmetry of the laws of $\varepsilon_i$ and $h(z_i')-h(z_i)$, to get:

$$\begin{aligned}
&\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\mathbb{E}[h(z)]-\frac{1}{n}\sum_{i=1}^n h(z_i)\bigg)\bigg]\\
\leqslant\ &\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n\varepsilon_i\big(h(z_i')-h(z_i)\big)\bigg)\bigg]\\
\leqslant\ &\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n\varepsilon_i\big(h(z_i)\big)\bigg)\bigg]+\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n\varepsilon_i\big(-h(z_i)\big)\bigg)\bigg]\\
=\ &2\mathbb{E}\bigg[\sup_{h\in\mathcal{H}}\bigg(\frac{1}{n}\sum_{i=1}^n\varepsilon_i h(z_i)\bigg)\bigg]=2\mathrm{R}_n(\mathcal{H}).
\end{aligned}$$

The reasoning is essentially identical for $\mathbb{E}\left[\sup_{h\in\mathcal{H}}\left(\frac{1}{n}\sum_{i=1}^n h(z_i)-\mathbb{E}[h(z)]\right)\right]\leqslant 2\mathrm{R}_n(\mathcal{H})$. ■

The lemma above only bounds the expectation of the deviation between the empirical average and the expectation by the Rademacher average. Together with concentration inequalities from Section 1.2, we can obtain high-probability bounds, as done in Section 4.4.1 with McDiarmid's inequality.

**Exercise 4.10** *(♦) The Gaussian complexity of a class of functions $\mathcal{H}$ from $\mathcal{Z}$ to $\mathbb{R}$ is defined as $G_n(\mathcal{H})=\mathbb{E}_{\varepsilon,\mathcal{D}}\big(\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i h(z_i)\big)$, where $\varepsilon\in\mathbb{R}^n$ is a vector of independent Gaussian variables with mean zero and variance one. Show that (a) $\mathrm{R}_n(\mathcal{H})\leqslant\sqrt{\frac{\pi}{2}}\cdot G_n(\mathcal{H})$ and (b) $G_n(\mathcal{H})\leqslant\sqrt{2\log(2n)}\cdot\mathrm{R}_n(\mathcal{H})$.*

**Empirical Rademacher complexities (♦).** The Rademacher complexity $\mathrm{R}_n(\mathcal{H})$ defined in Eq. (4.9) is a *deterministic* quantity that depends on the distribution of inputs. When using bound in high-probability through MacDiarmid's inequality in Section 4.4.1, we obtained that if $|h(z)|\leqslant\ell_\infty$ for all $h\in\mathcal{H}$, then with probability greater than $1-\delta$, for all $h\in\mathcal{H}$,

$$\mathbb{E}[h(z)]\leqslant\frac{1}{n}\sum_{i=1}^n h(z_i)+2\mathrm{R}_n(\mathcal{H})+\frac{2\ell_\infty}{\sqrt{n}}\sqrt{\log(1/\delta)}.$$

While we provide below estimates based on simple information on the input distribution, an empirical version can be defined that does not take the expectation with respect to the data, that is,

$$\hat{\mathrm{R}}_n(\mathcal{H}) = \mathbb{E}_\varepsilon\Big(\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i h(z_i)\Big), \tag{4.10}$$

which is now a random quantity that is computable from the training data and the class of functions. We can also use MacDiarmid's inequality to bound the difference between $\mathrm{R}_n(\mathcal{H})$ and $\hat{\mathrm{R}}_n(\mathcal{H})$, obtain a similar high-probability bound as above, that is,

$$\mathbb{E}[h(z)] \leqslant \frac{1}{n}\sum_{i=1}^n h(z_i) + 2\hat{\mathrm{R}}_n(\mathcal{H}) + 4\frac{2\ell_\infty}{\sqrt{n}}\sqrt{\log(2/\delta)},$$

which is now computable.

### 4.5.2 Lipschitz-continuous losses

A particularly appealing property in our context is the following property, sometimes called the "contraction principle," using a simple proof from Meir and Zhang (2003, Lemma 5); see also Ledoux and Talagrand (1991, Section 4.5). See Prop. 4.4 below for a similar result for the Rademacher complexity defined with absolute values (and then with an extra factor of 2).

**Proposition 4.3 (Contraction principle - Lipschitz-continuous functions)** *Given any functions $b$, $a_i : \Theta \to \mathbb{R}$ (no assumption) and $\varphi_i : \mathbb{R} \to \mathbb{R}$ any 1-Lipschitz-functions, for $i = 1, \dots, n$, we have, for $\varepsilon \in \mathbb{R}^n$ a vector of independent Rademacher random variables:*

$$\mathbb{E}_\varepsilon\Big[\sup_{\theta\in\Theta}\Big\{b(\theta) + \sum_{i=1}^n \varepsilon_i\varphi_i(a_i(\theta))\Big\}\Big] \leqslant \mathbb{E}_\varepsilon\Big[\sup_{\theta\in\Theta}\Big\{b(\theta) + \sum_{i=1}^n \varepsilon_i a_i(\theta)\Big\}\Big].$$

**Proof** (♦) We consider a proof by induction on $n$. The case $n = 0$ is trivial, and we show how to go from $n \geqslant 0$ to $n+1$. We thus consider $\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sum_{i=1}^{n+1}\varepsilon_i\varphi_i(a_i(\theta))\Big\}\Big]$ and compute the expectation with respect to $\varepsilon_{n+1}$ explicitly, by considering the two potential values with probability 1/2:

$$\begin{aligned}
&\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta) + \sum_{i=1}^{n+1}\varepsilon_i\varphi_i(a_i(\theta))\Big\}\Big]\\
&= \frac{1}{2}\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sum_{i=1}^n\varepsilon_i\varphi_i(a_i(\theta)) + \varphi_{n+1}(a_{n+1}(\theta))\Big\}\Big]\\
&\qquad +\frac{1}{2}\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sum_{i=1}^n\varepsilon_i\varphi_i(a_i(\theta)) - \varphi_{n+1}(a_{n+1}(\theta))\Big\}\Big],
\end{aligned}$$

which is equal to

$$\begin{aligned}\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\Big[\sup_{\theta,\theta'\in\Theta}\Big\{&\frac{b(\theta)+b(\theta')}{2}\\ &+\sum_{i=1}^n \varepsilon_i \frac{\varphi_i(a_i(\theta))+\varphi_i(a_i(\theta'))}{2} + \frac{\varphi_{n+1}(a_{n+1}(\theta))-\varphi_{n+1}(a_{n+1}(\theta'))}{2}\Big\}\Big],\end{aligned}$$

by assembling the terms. By taking the supremum over $(\theta,\theta')$ and $(\theta',\theta)$, we get

$$\begin{aligned}&\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\Big[\sup_{\theta,\theta'\in\Theta}\Big\{\frac{b(\theta)+b(\theta')}{2}\\ &\qquad+\sum_{i=1}^n \varepsilon_i \frac{\varphi_i(a_i(\theta))+\varphi_i(a_i(\theta'))}{2} + \frac{|\varphi_{n+1}(a_{n+1}(\theta))-\varphi_{n+1}(a_{n+1}(\theta'))|}{2}\Big\}\Big]\\ \leqslant\ &\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\Big[\sup_{\theta,\theta'\in\Theta}\Big\{\frac{b(\theta)+b(\theta')}{2} + \sum_{i=1}^n \varepsilon_i \frac{\varphi_i(a_i(\theta))+\varphi_i(a_i(\theta'))}{2} + \frac{|a_{n+1}(\theta)-a_{n+1}(\theta')|}{2}\Big\}\Big],\end{aligned}$$

using Lipschitz-continuity. We can redo the same sequence of *equalities* with $\varphi_{n+1}$ being the identity to obtain that the last expression above is equal to

$$\begin{aligned}&\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\mathbb{E}_{\varepsilon_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\varepsilon_{n+1}a_{n+1}(\theta)+\sum_{i=1}^n\varepsilon_i\varphi_i(a_i(\theta))\Big\}\Big]\\ \leqslant\ &\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n,\varepsilon_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\varepsilon_{n+1}a_{n+1}(\theta)+\sum_{i=1}^n\varepsilon_i a_i(\theta)\Big\}\Big] \text{ by the induction hypothesis,}\end{aligned}$$

which leads to the desired result. ■

We can apply the contraction principle above to supervised learning situations where $u_i \mapsto \ell(y_i,u_i)$ is $G$-Lipschitz-continuous for all $i$ almost surely (which is possible for regression or when using a convex surrogate for binary classification as presented in Section 4.1), leading to, by the contraction principle (applied conditioned on the data $\mathcal{D}$ to $b=0$, $\Theta=\{(f(x_1),\dots,f(x_n)),\ f\in\mathcal{F}\}\subset\mathbb{R}^n$ and $a_i(\theta)=\theta_i$, $\varphi_i(u_i)=\ell(y_i,u_i)$):

$$\mathbb{E}_\varepsilon\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\ell(y_i,f(x_i))\ \Big|\ \mathcal{D}\Big] \ \leqslant\ G\cdot\mathbb{E}_\varepsilon\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i f(x_i)\ \Big|\ \mathcal{D}\Big],$$

which leads to

$$\mathrm{R}_n(\mathcal{H})\leqslant G\cdot\mathrm{R}_n(\mathcal{F}). \tag{4.11}$$

Thus, the Rademacher complexity of the class of prediction functions controls the uniform deviations of the empirical risk. We consider simple examples below but give before without proof a contraction result that we will need in Section 9.2.3 (See proof by Ledoux and Talagrand, 1991, Theorem 4.12).

**Proposition 4.4 (Contraction principle - absolute values)** *Given any functions $a_i : \Theta \to \mathbb{R}$ (no assumption) and $\varphi_i : \mathbb{R} \to \mathbb{R}$ any 1-Lipschitz-functions such that $\varphi_i(0) = 0$, for $i = 1, \dots, n$, we have, for $\varepsilon \in \mathbb{R}^n$ a vector of independent Rademacher random variables:*

$$\mathbb{E}_\varepsilon\bigg[\sup_{\theta\in\Theta}\Big|\sum_{i=1}^n \varepsilon_i \varphi_i(a_i(\theta))\Big|\bigg] \leqslant 2\,\mathbb{E}_\varepsilon\bigg[\sup_{\theta\in\Theta}\Big|\sum_{i=1}^n \varepsilon_i a_i(\theta)\Big|\bigg].$$

### 4.5.3 Ball-constrained linear predictions

We now assume that $\mathcal{F} = \{f_\theta(x) = \theta^\top \varphi(x),\ \Omega(\theta) \leqslant D\}$ where $\Omega$ is a norm on $\mathbb{R}^d$. We denote the design matrix by $\Phi \in \mathbb{R}^{n\times d}$. We have (with expectations both with respect to $\varepsilon$ and the data):

$$\begin{aligned}
\mathrm{R}_n(\mathcal{F}) &= \mathbb{E}\bigg[\sup_{\Omega(\theta)\leqslant D}\Big\{\frac{1}{n}\sum_{i=1}^n \varepsilon_i \theta^\top \varphi(x_i)\Big\}\bigg] = \mathbb{E}\bigg[\sup_{\Omega(\theta)\leqslant D} \frac{1}{n}\varepsilon^\top \Phi\theta\bigg] \\
&= \frac{D}{n}\mathbb{E}\Big[\Omega^*(\Phi^\top\varepsilon)\Big],
\end{aligned}$$

where $\Omega^*(u) = \sup_{\Omega(\theta)\leqslant 1} u^\top\theta$ is the *dual norm* of $\Omega$. For example, when $\Omega$ is the $\ell_p$-norm, with $p \in [1, \infty]$, then $\Omega^*$ is the $\ell_q$-norm, where $q$ is such that $\frac{1}{p} + \frac{1}{q} = 1$, e.g., $\|\cdot\|_2^* = \|\cdot\|_2$, $\|\cdot\|_1^* = \|\cdot\|_\infty$, and $\|\cdot\|_\infty^* = \|\cdot\|_1$. For more details, see Boyd and Vandenberghe (2004).

Thus, computing Rademacher complexities is equivalent to computing expectations of norms. When $\Omega = \|\cdot\|_2$, we get:

$$\begin{aligned}
\mathrm{R}_n(\mathcal{F}) &= \frac{D}{n}\mathbb{E}\left[\|\Phi^\top\varepsilon\|_2\right] \leqslant \frac{D}{n}\sqrt{\mathbb{E}\left[\|\Phi^\top\varepsilon\|_2^2\right]} \text{ by Jensen's inequality,} \\
&= \frac{D}{n}\sqrt{\mathbb{E}\left[\mathrm{tr}[\Phi^\top\varepsilon\varepsilon^\top\Phi]\right]} = \frac{D}{n}\sqrt{\mathbb{E}\left[\mathrm{tr}[\Phi^\top\Phi]\right]} \text{ using that } \mathbb{E}[\varepsilon\varepsilon^\top] = I, \\
&= \frac{D}{n}\sqrt{\sum_{i=1}^n \mathbb{E}(\Phi^\top\Phi)_i} = \frac{D}{n}\sqrt{\sum_{i=1}^n \mathbb{E}\|\varphi(x_i)\|_2^2} = \frac{D}{\sqrt{n}}\sqrt{\mathbb{E}\|\varphi(x)\|_2^2}. \qquad (4.12)
\end{aligned}$$

We thus obtain a *dimension-independent* Rademacher complexity that we can use in the summary in Section 4.5.4 below.

**Exercise 4.11** *($\ell_1$-norm) Assume that almost surely $\|\varphi(x)\|_\infty \leqslant R$. Show that the Rademacher complexity $\mathrm{R}_n(\mathcal{F})$ for $\mathcal{F} = \{f_\theta(x) = \theta^\top\varphi(x),\ \Omega(\theta) \leqslant D\}$ with $\Omega = \|\cdot\|_1$ is upper-bounded by $RD\sqrt{\frac{2\log(2d)}{n}}$.*

**Exercise 4.12** *(♦) Let $p > 1$, and $q$ such that $1/p + 1/q = 1$. Assume that almost surely $\|\varphi(x)\|_q \leqslant R$. Show that the Rademacher complexity $\mathrm{R}_n(\mathcal{F})$ for $\mathcal{F} = \{f_\theta(x) = \theta^\top\varphi(x),\ \Omega(\theta) \leqslant D\}$, with $\Omega = \|\cdot\|_p$, is upper-bounded by $\frac{RD}{\sqrt{n}}\frac{1}{\sqrt{p-1}}$ (hint: use Exercise 1.19). Recover the result of Exercise 4.11 by taking $p = 1 + \frac{1}{\log(2d)}$.*

### 4.5.4 Putting things together (linear predictions)

We now consider a linear model based on some feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$ and apply the Rademacher results from the previous section to obtain a bound on the estimation error. We then look at the approximation error.

**Estimation error.** With all the elements above, we can now propose the following general result (where no convexity of the loss function is assumed) for the estimation error. Note that there is no explicit dependence on the underlying dimension $d$, which will be important in Chapter 7 where we consider infinite-dimensional feature spaces.

**Proposition 4.5 (Estimation error)** *Assume a $G$-Lipschitz-continuous loss function, linear prediction functions with $\mathcal{F} = \{f_\theta(x) = \theta^\top \varphi(x),\ \|\theta\|_2 \leqslant D\}$, where $\mathbb{E}\|\varphi(x)\|_2^2 \leqslant R^2$. Let $\hat{f} = f_{\hat{\theta}} \in \mathcal{F}$ be the minimizer of the empirical risk, then:*

$$\mathbb{E}\big[\mathcal{R}(f_{\hat{\theta}})\big] \leqslant \inf_{\|\theta\|_2 \leqslant D} \mathcal{R}(f_\theta) + \frac{4GRD}{\sqrt{n}}.$$

**Proof** Using Prop. 4.2 to relate the uniform deviation to the Rademacher average, Eq. (4.11) to take care of the Lipschitz-continuous loss, and Eq. (4.12) to account for the $\ell_2$-norm constraint, we get the desired result. Note that the factor of 4 comes from symmetrization (Prop. 4.2, which leads to a factor of 2), and Eq. (4.8) in Section 4.4 (which leads to another one). ■

**Approximation error.** If we assume that there exists a minimizer $\theta_*$ of $\mathcal{R}(f_\theta)$ over $\mathbb{R}^d$, the approximation error (of using a ball of $\theta$ rather than the whole $\mathbb{R}^d$) is upper-bounded by, following derivations from Section 4.3 (using Cauchy-Schwarz and Jensen's inequalities):

$$\begin{aligned}
\inf_{\|\theta\|_2 \leqslant D} \mathcal{R}(f_\theta) - \mathcal{R}(f_{\theta_*}) &\leqslant G \inf_{\|\theta\|_2 \leqslant D} \mathbb{E}\big[|f_\theta(x) - f_{\theta_*}(x)|\big] \\
&= G \inf_{\|\theta\|_2 \leqslant D} \mathbb{E}\big[|\varphi(x)^\top(\theta - \theta_*)|\big] \\
&\leqslant G \inf_{\|\theta\|_2 \leqslant D} \|\theta - \theta_*\|_2 \mathbb{E}\big[\|\varphi(x)\|_2\big] \leqslant GR \inf_{\|\theta\|_2 \leqslant D} \|\theta - \theta_*\|_2.
\end{aligned}$$

This leads to

$$\mathbb{E}\big[\mathcal{R}(f_{\hat{\theta}})\big] - \mathcal{R}(f_{\theta_*}) \leqslant GR \inf_{\|\theta\|_2 \leqslant D} \|\theta - \theta_*\|_2 + \frac{4GRD}{\sqrt{n}} = GR(\|\theta_*\|_2 - D)_+ + \frac{4GRD}{\sqrt{n}}.$$

We see that for $D = \|\theta_*\|_2$, we obtain the bound $\frac{4GR\|\theta_*\|_2}{\sqrt{n}}$, but this setting requires to know $\|\theta_*\|_2$ which is not possible in practice. If $D$ is too large, the estimation error increases (overfitting). At the same time, if $D$ is too small, the approximation error can quickly kick in (with a value that does not go to zero when $n$ tends to infinity), leading to underfitting. Note that on top of this approximation error, we need to add the incompressible one due to the choice of a linear model.

**Exercise 4.13** *We consider a learning problem with* $1$*-Lipschitz-continuous loss (with respect to the second variable), with a function class* $f_\theta(x) = \theta^\top \varphi(x)$*, with* $\|\theta\|_1 \leqslant D$*, and* $\varphi : \mathcal{X} \to \mathbb{R}^d$ *with* $\|\varphi(x)\|_\infty$ *almost surely less than* $R$*. Given the expected risk* $\mathcal{R}(f_\theta)$ *and the empirical risk* $\widehat{\mathcal{R}}(f_\theta)$*. Show that* $\mathbb{E}\big[\mathcal{R}(f_{\hat{\theta}})\big] \leqslant \inf_{\|\theta\|_1 \leqslant D} \mathcal{R}(f_\theta) + 4RD\sqrt{2\log(2d)/n}$*.*

### 4.5.5 From constrained to regularized estimation (♦)

In practice, it is preferable to penalize by the norm $\Omega(\theta)$ instead of constraining. While the respective sets of solutions when letting the respective constraint and regularization parameters vary are the same, the main reason is that the hyperparameter is easier to find, and the optimization is typically easier. For simplicity, we only consider the $\ell_2$-norm in this section.

We now denote $\hat{\theta}_\lambda$ the minimizer of

$$\widehat{\mathcal{R}}(f_\theta) + \frac{\lambda}{2}\|\theta\|_2^2. \tag{4.13}$$

If the loss is always positive, then $\frac{\lambda}{2}\|\hat{\theta}_\lambda\|_2^2 \leqslant \widehat{\mathcal{R}}(f_{\hat{\theta}_\lambda}) + \frac{\lambda}{2}\|\hat{\theta}_\lambda\|_2^2 \leqslant \widehat{\mathcal{R}}(f_0)$, leading to a bound $\|\hat{\theta}_\lambda\|_2 = O(1/\sqrt{\lambda})$. Thus, with $D = O(1/\sqrt{\lambda})$ in the bound above, this leads to a deviation of $O(1/\sqrt{\lambda n})$, which is not optimal.

We now give an interesting stronger result using the strong convexity of the squared $\ell_2$-norm (with now a convex loss), adapted from Sridharan et al. (2009); Bartlett et al. (2005).

**Proposition 4.6 (Fast rates for regularized objectives)** *Assume the loss function is* $G$*-Lipschitz-continuous and* ***convex****, with linear prediction functions defined by* $f_\theta(x) = \theta^\top \varphi(x)$*, for* $\theta \in \mathbb{R}^d$*, where* $\|\varphi(x)\|_2 \leqslant R$ *almost surely. Let* $\hat{\theta}_\lambda \in \mathbb{R}^d$ *be the minimizer of the regularized empirical risk in Eq. (4.13), then:*

$$\mathbb{E}\big[\mathcal{R}(f_{\hat{\theta}_\lambda})\big] \leqslant \inf_{\theta \in \mathbb{R}^d} \big\{\mathcal{R}(f_\theta) + \frac{\lambda}{2}\|\theta\|_2^2\big\} + \frac{32G^2R^2}{\lambda n}.$$

**Proof** (♦) Let $\mathcal{R}_\lambda(f_\theta) = \mathcal{R}(f_\theta) + \frac{\lambda}{2}\|\theta\|_2^2$, with minimum value $\mathcal{R}_\lambda^*$ attained at $\theta_\lambda^*$. We consider the convex set $\mathcal{C}_\varepsilon = \{\theta \in \mathbb{R}^d,\ \mathcal{R}_\lambda(f_\theta) - \mathcal{R}_\lambda^* \leqslant \varepsilon\}$ for an $\varepsilon > 0$ to be chosen later. If $\hat{\theta}_\lambda \notin \mathcal{C}_\varepsilon$, then, by convexity, there has to be an $\eta$ in the segment $[\theta_\lambda^*, \hat{\theta}_\lambda]$ such that $\mathcal{R}_\lambda(f_\eta) - \mathcal{R}_\lambda^* = \mathcal{R}_\lambda(f_\eta) - \mathcal{R}_\lambda(f_{\theta_\lambda^*}) = \varepsilon$, and $\widehat{\mathcal{R}}_\lambda(f_\eta) \leqslant \widehat{\mathcal{R}}_\lambda(f_{\theta_\lambda^*})$.[7] This implies that

$$\mathcal{R}_\lambda(f_\eta) - \widehat{\mathcal{R}}_\lambda(f_\eta) + \widehat{\mathcal{R}}_\lambda(f_{\theta_\lambda^*}) - \mathcal{R}_\lambda(f_{\theta_\lambda^*}) = \mathcal{R}_\lambda(f_\eta) - \mathcal{R}_\lambda(f_{\theta_\lambda^*}) + \widehat{\mathcal{R}}_\lambda(f_{\theta_\lambda^*}) - \widehat{\mathcal{R}}_\lambda(f_\eta) \geqslant \varepsilon. \tag{4.14}$$

[7] This can be shown by taking $\eta$ at the intersection of the segment $[\theta_\lambda^*, \hat{\theta}_\lambda]$ and the set $\partial\mathcal{C}_\varepsilon$ (the boundary of $\mathcal{C}_\varepsilon$).

By strong convexity, we have, $\mathcal{R}_\lambda(f_\theta) - \mathcal{R}_\lambda^* \geqslant \frac{\lambda}{2}\|\theta - \theta_\lambda^*\|_2^2$ for all $\theta$, and thus $\mathcal{C}_\varepsilon$ is included in the $\ell_2$-ball of center $\theta_\lambda^*$ and radius $\sqrt{2\varepsilon/\lambda}$. Thus, from Eq. (4.14), we get $\sup_{\|\eta-\theta_\lambda^*\|_2 \leqslant \sqrt{2\varepsilon/\lambda}} \Big\{ \mathcal{R}_\lambda(f_\eta) - \mathcal{R}_\lambda(f_{\theta_\lambda^*}) - \big[\widehat{\mathcal{R}}_\lambda(f_\eta) - \widehat{\mathcal{R}}_\lambda(f_{\theta_\lambda^*})\big] \Big\} \geqslant \varepsilon$. Using Section 4.5.3, we have

$$\begin{aligned} &\mathbb{E}\Big[ \sup_{\|\eta-\theta_\lambda^*\|_2 \leqslant \sqrt{2\varepsilon/\lambda}} \Big\{ \mathcal{R}_\lambda(f_\eta) - \mathcal{R}_\lambda(f_{\theta_\lambda^*}) - \big[\widehat{\mathcal{R}}_\lambda(f_\eta) - \widehat{\mathcal{R}}_\lambda(f_{\theta_\lambda^*})\big] \Big\} \Big] \\ \leqslant\ & 2\mathbb{E}\Big[ \sup_{\|\eta-\theta_\lambda^*\|_2 \leqslant \sqrt{2\varepsilon/\lambda}} \Big\{ \frac{1}{n}\sum_{i=1}^n \varepsilon_i [\ell(y_i, \varphi(x_i)^\top \eta) - \ell(y_i, \varphi(x_i)^\top \theta_\lambda^*)] \Big\} \Big] \leqslant 2GR\sqrt{2\varepsilon/\lambda}. \end{aligned}$$

Moreover, by McDiarmid's inequality,

$$\mathbb{P}\Big( \mathcal{R}_\lambda(f_\eta) - \widehat{\mathcal{R}}_\lambda(f_\eta) + \widehat{\mathcal{R}}_\lambda(f_{\theta_\lambda^*}) - \mathcal{R}_\lambda(f_{\theta_\lambda^*}) \geqslant 2GR\sqrt{2\varepsilon/\lambda} + t\frac{2\frac{GR}{\sqrt{n}}\sqrt{2\varepsilon/\lambda}}{\sqrt{2n}} \Big) \leqslant e^{-t^2}.$$

Thus, if $\varepsilon \geqslant 2\frac{GR}{\sqrt{n}}\sqrt{2\varepsilon/\lambda}(1 + \frac{t}{\sqrt{2}})$, that is, if $\varepsilon \geqslant 8\frac{G^2R^2}{\lambda n}(2 + t^2)$, we have the high probability bound $\mathbb{P}\big(\mathcal{R}_\lambda(f_{\hat\theta_\lambda}) - \mathcal{R}_\lambda^* > \varepsilon\big) \leqslant e^{-t^2}$. This leads to, by integration, the final bound $\mathbb{E}\big[\mathcal{R}_\lambda(f_{\hat\theta_\lambda}) - \mathcal{R}_\lambda^*\big] \leqslant \frac{32G^2R^2}{\lambda n}$. ■

Note that we obtain a "fast rate" in $O(R^2/(\lambda n))$, which has a better dependence in $n$ but depends on $\lambda$, which can be very small in practice. One classical choice of $\lambda$ that we have seen in Chapter 3 also applies here, as $\lambda \propto \frac{GR}{\sqrt{n}\|\theta_*\|}$, leading to the slow rate

$$\mathbb{E}\big[\mathcal{R}(f_{\hat\theta_\lambda})\big] \leqslant \mathcal{R}(f_{\theta_*}) + O\Big(\frac{GR}{\sqrt{n}}\|\theta_*\|_2\Big).$$

This result is similar to the one obtained in Chapter 3 for ridge (least-squares) regression, but now for all Lipschitz-continuous losses. Note that the amount of regularization to get the result above still depends on the unknown quantity $\|\theta_*\|_2$. Below, we consider the general case of penalization by a norm, where we will obtain similar results but with a hyperparameter that does not depend on the unknown norm of $\|\theta_*\|_2$.

**Exercise 4.14** *(♦♦) Extend the result in Prop. 4.6 to features that are almost surely bounded in $\ell_p$-norm by $R$, and a regularizer $\psi$ which is strongly-convex with respect to the $\ell_p$-norm, that is, such that for all $\theta, \eta \in \mathbb{R}^d$, $\psi(\theta) \geqslant \psi(\eta) + \psi'(\eta)^\top(\theta - \eta) + \frac{\mu}{2}\|\theta - \eta\|_p^2$, where $\psi'(\eta)$ is a subgradient of $\psi$ at $\eta$.*

**Norm-penalized estimation. (♦♦)** While Proposition 4.6 considered squared $\ell_2$-norm penalization and relied on specific properties of the $\ell_2$-norm, we now consider penalization by *any* (non-squared) norm. That is, we now focus on the following objective function:

$$\widehat{\mathcal{R}}_\lambda(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, \theta^\top \varphi(x_i)) + \lambda\Omega(\theta), \tag{4.15}$$

for any norm $\Omega$ on $\mathbb{R}^d$, with $\Omega^*$ denoting the dual norm. The following proposition provides an estimation rate in $O(1/\sqrt{n})$.

**Proposition 4.7 (Norm-penalized estimation)** *Assume that the unregularized risk* $\mathcal{R}_0(\theta) = \mathbb{E}_{p(x,y)}\big[\ell(y, \theta^\top \varphi(x))\big]$ *is minimized at some* $\theta_* \in \mathbb{R}^d$, *and that the function* $\theta \mapsto \ell(y, \theta^\top\varphi(x))$ *is GR-Lipschitz continuous in* $\theta$ *for* $\Omega(\theta) \leqslant 2\Omega(\theta_*)$, *and that* $\Omega^*(\varphi(x)) \leqslant R$ *almost surely. Denote* $\rho_\Omega = \sup_{\Omega^*(z_1),\dots,\Omega^*(z_n)\leqslant 1} \mathbb{E}_\varepsilon\big[\Omega^*\big(\frac{1}{\sqrt{n}}\sum_{i=1}^n \varepsilon_i z_i\big)\big]$ *where* $\varepsilon \in \{-1,1\}^n$ *is a vector of independent Rademacher random variables. For any* $\delta \in (0,1)$, *and for* $\lambda = \frac{GR}{\sqrt{n}}(\rho_\Omega + 4\sqrt{2\log\frac{1}{\delta}})$, *with probability at least* $1-\delta$, *any minimizer* $\hat\theta_\lambda$ *of Eq. (4.15) satisfies:*

$$\mathcal{R}(\hat\theta_\lambda) \leqslant \mathcal{R}(\theta_*) + \Omega(\theta_*)\frac{3GR}{\sqrt{n}}\Big(\rho_\Omega + 4\sqrt{2\log\frac{1}{\delta}}\Big).$$

**Proof** We consider $\theta_\lambda^*$ a minimizer of the population regularized risk $\mathcal{R}_\lambda(\theta) = \mathcal{R}(\theta) + \lambda\Omega(\theta)$. It satisfies $\Omega(\theta_\lambda^*) \leqslant \Omega(\theta_*)$. Moreover, $\rho_\Omega$ is such that the Rademacher complexity of the set of linear predictors such that $\Omega(\theta) \leqslant D$ for $D \leqslant 2\Omega(\theta_*)$, is less than $\frac{\rho_\Omega GRD}{\sqrt{n}}$ (see Section 4.5.3). For example, for the $\ell_2$-norm, we have $\rho_\Omega = 1$, while for the $\ell_1$-norm, we have $\rho_\Omega = \sqrt{2\log(2d)}$. In terms of losses, for the logistic loss, we have $G = 1$, while for the square loss (with a factor of 1/2) with a model $y = \varphi(x)^\top\theta^* + \varepsilon$ with $|\varepsilon| \leqslant \sigma$ almost surely, we get $G = \sigma + 3R\Omega(\theta^*)$.

Using McDiarmid's inequality like in Section 4.4.1, by fixing any $\theta_0$ such that $\Omega(\theta_0) \leqslant D$, with probability greater than $1-e^{-t^2}$, for all $\theta$ such that $\Omega(\theta) \leqslant 2\Omega(\theta_*)$, $\mathcal{R}(\theta) - \mathcal{R}(\theta_0) \leqslant \widehat{\mathcal{R}}(\theta) - \widehat{\mathcal{R}}(\theta_0) + \frac{\rho_\Omega GRD}{\sqrt{n}} + t\frac{2GRD\sqrt{2}}{\sqrt{n}}$.

We consider the set $\mathcal{C}_{\nu,\varepsilon} = \big\{\theta \in \mathbb{R}^d,\ \Omega(\theta) \leqslant 2\Omega(\theta_\lambda^*),\ \mathcal{R}_\lambda(\theta) - \mathcal{R}_\lambda(\theta_\lambda^*) \leqslant \varepsilon\big\}$. This is a convex set, with boundary $\partial\mathcal{C}_{\nu,\varepsilon} = \big\{\theta \in \mathbb{R}^d,\ \Omega(\theta) \leqslant 2\Omega(\theta_\lambda^*),\ \mathcal{R}_\lambda(\theta) - \mathcal{R}_\lambda(\theta_\lambda^*) = \varepsilon\big\}$ for a well-chosen $\varepsilon$ (that is, the saturated constraint has to be one on the expected risk). Indeed, if $\Omega(\theta) = 2\Omega(\theta_\lambda^*)$, then, using that the optimality conditions for $\theta_\lambda^*$ implies that $\Omega^*(\mathcal{R}'(\theta_\lambda^*)) \leqslant \lambda$:

$$\begin{aligned}
\mathcal{R}_\lambda(\theta) - \mathcal{R}_\lambda(\theta_\lambda^*) &= \mathcal{R}(\theta) - \mathcal{R}(\theta_\lambda^*) + \lambda\Omega(\theta) - \lambda\Omega(\theta_\lambda^*) \text{ by definition,}\\
&\geqslant \mathcal{R}'(\theta_\lambda^*)^\top(\theta - \theta_\lambda^*) + \lambda\Omega(\theta) - \lambda\Omega(\theta_\lambda^*) \text{ by convexity,}\\
&\geqslant -\Omega^*(\mathcal{R}'(\theta_\lambda^*)) \cdot \Omega(\theta - \theta_\lambda^*) + \lambda\Omega(\theta) - \lambda\Omega(\theta_\lambda^*)\\
&\qquad\qquad\qquad\qquad \text{by definition of the dual norm,}\\
&\geqslant -\lambda\Omega(\theta - \theta_\lambda^*) + \lambda\Omega(\theta) - \lambda\Omega(\theta_\lambda^*) \text{ by optimality of } \theta_\lambda^*,\\
&\geqslant 2\lambda\Omega(\theta) - 2\lambda\Omega(\theta_\lambda^*) \text{ by the triangular inequality,}\\
&= 2\lambda\Omega(\theta_\lambda^*) \text{ since we have assumed } \Omega(\theta) = 2\Omega(\theta_\lambda^*).
\end{aligned}$$

Thus, we must impose that $\varepsilon \leqslant 2\Omega(\theta_\lambda^*)$.

We now show that with high probability, we must have $\hat\theta_\lambda \in \mathcal{C}_{\nu,\varepsilon}$. If $\hat\theta_\lambda \notin \mathcal{C}_{\nu,\varepsilon}$, since $\theta_\lambda^* \in \mathcal{C}_{\nu,\varepsilon}$, there has to be an element $\theta$ in the segment $[\theta_\lambda^*, \hat\theta_\lambda]$ which is in $\partial\mathcal{C}_{\nu,\varepsilon}$. Since our risks are convex, we have $\widehat{\mathcal{R}}_\lambda(\theta) \leqslant \max\{\widehat{\mathcal{R}}_\lambda(\theta_\lambda^*), \widehat{\mathcal{R}}_\lambda(\hat{\theta}_\lambda)\} = \widehat{\mathcal{R}}_\lambda(\theta_\lambda^*)$. Thus

$$\widehat{\mathcal{R}}(\theta_\lambda^*) - \widehat{\mathcal{R}}(\theta) - \mathcal{R}(\theta_\lambda^*) + \mathcal{R}(\theta) = \widehat{\mathcal{R}}_\lambda(\theta_\lambda^*) - \widehat{\mathcal{R}}_\lambda(\theta) - \mathcal{R}_\lambda(\theta_\lambda^*) + \mathcal{R}_\lambda(\theta) \geqslant -\mathcal{R}_\lambda(\theta_\lambda^*) + \mathcal{R}_\lambda(\theta) = \varepsilon.$$

Thus if we take, $\varepsilon \geqslant \frac{\rho_\Omega GRD}{\sqrt{n}} + t\frac{2GRD\sqrt{2}}{\sqrt{n}}$, with $D = 2\Omega(\theta_\lambda^*)$, this can only happen with probability less than $\exp(-t^2)$. This leads to the constraint $\varepsilon \geqslant \frac{2GR\Omega(\theta_\lambda^*)}{\sqrt{n}}(\rho_\Omega + 4t\sqrt{2})$. Thus, we can take $\lambda = \frac{GR}{\sqrt{n}}(\rho_\Omega + 4t\sqrt{2})$, and with probability greater than $1 - e^{-t^2}$ we have

$$\mathcal{R}_\lambda(\hat{\theta}_\lambda) - \mathcal{R}_\lambda(\theta_\lambda^*) \leqslant 2\lambda\Omega(\theta_\lambda^*) \leqslant 2\lambda\Omega(\theta^*).$$

Overall, denoting $\delta = e^{-t^2}$, we get that with probability greater than $1 - \delta$

$$\mathcal{R}(\hat{\theta}_\lambda) \leqslant \mathcal{R}(\theta_*) + \Omega(\theta_*)\frac{3GR}{\sqrt{n}}\Big(\rho_\Omega + 4\sqrt{2\log\frac{1}{\delta}}\Big).$$

for $\lambda = \frac{GR}{\sqrt{n}}(\rho_\Omega + 4\sqrt{2\log\frac{1}{\delta}})$. Note that we could get a result in expectation (left as an exercise). The key here is that the value of $\lambda$ does not depend on $\Omega(\theta^*)$. ■

### 4.5.6 Extensions and improvements

In this chapter, we have focused on the simplest situations for empirical risk minimization technique: regression or binary classification with i.i.d. data. Statistical learning theory investigates many more complex cases along several lines:

- **Slower rates than $1/\sqrt{n}$:** In this chapter, we primarily studied the estimation error that decays as $1/\sqrt{n}$. When balancing it with approximation error (by adapting norm constraints or regularization parameters), we will obtain slower rates, but with weaker assumptions, in Chapter 7 (kernel methods) and Chapter 9 (neural networks).

- **Faster rates with discrete outputs:** Further analysis can be carried through when dealing with binary classification, or more generally discrete outputs, with potentially different convergence rates for the convex surrogate and the original loss function (i.e., after thresholding, where sometimes exponential rates can be obtained). This is often done under so-called "low noise" conditions (see, e.g., Koltchinskii and Beznosova, 2005; Audibert and Tsybakov, 2007), as briefly exposed in Section 4.1.4.

- **Other generic learning theory frameworks:** In this chapter, we have focused primarily on the tools of Rademacher averages to obtain generic learning bounds. Other frameworks lead to similar bounds but from different mathematical perspectives. For example, PAC-Bayesian analysis (Catoni, 2007; Zhang, 2006) is described in Section 14.4, while stability-based arguments (Bousquet and Elisseeff, 2002) lead to similar results (see exercise below).

**Exercise 4.15** *(♦) We consider a learning algorithm and a distribution $p$ on $(x, y)$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and two outputs $f, f' : \mathcal{X} \to \mathcal{Y}$ of the learning algorithm on datasets of $n$ observations which differ by a single observation, $|\ell(y, f(x)) - \ell(y, f'(x))| \leqslant \beta_n$, an assumption referred to as "uniform stability". Show that the expected deviation between the expected risk and the empirical risk of the algorithm's output is bounded by $\beta_n$. With the same assumptions as in Prop. 4.6, show that we have $\beta_n = \frac{2G^2R^2}{\lambda n}$ (see Bousquet and Elisseeff, 2002, for more details).*

- **Beyond independent observations:** Much of statistical learning theory deals with the simplifying assumptions that observations are i.i.d. from the same distribution as the one used during the testing phase. This leads to the reasonably simple results presented in this chapter. Several lines of work deal with situations when data are not independent: among them, online learning presented in Chapter 11 shows that many classical algorithms are indeed robust to such dependence. Another avenue coming from statistics is to make some assumptions on the dependence between observations, the most classical one being that the sequence of observations $(x_i, y_i)_{i \geqslant 1}$ form a Markov chain, and thus satisfies "mixing conditions" (see, e.g., Mohri and Rostamizadeh, 2010).
- **Mismatch between training and testing distributions:** In many application scenarios, the testing distribution may deviate from the training distribution: the input distribution of $x$ may be different while the conditional distribution of $y$ given $x$ remains the same, a situation commonly referred to as "covariate shift", or the entire distribution of $(x, y)$ may deviate (often referred to as the need for "domain adaptation"). If no assumption is made on the proximity of these two distributions, no guarantee can be obtained. Several ideas have been explored to derive algorithms and/or guarantees, such as importance reweighting (Sugiyama et al., 2007) or finding projections of the data with similar test and train distributions (Ganin et al., 2016).
- **Semi-supervised learning:** In many applications, many unlabelled observations are available (that is, only with the input $x$ being available). To leverage the abundance of unlabelled data, some assumptions are typically made to show an improvement of learning algorithms, such as the "cluster assumption" (points in the same class tend to cluster together) or "low-density separation" (for classification, decision boundaries tend to be in regions with few input observations). Many algorithms exist, such as Laplacian regularization (see Cabannes et al., 2021, and references therein) or discriminative clustering (Xu et al., 2004; Bach and Harchaoui, 2007).

## 4.6 Model selection (♦)

Throughout this chapter, we have considered a family $\mathcal{F}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ and have obtained generalization bounds for the minimizer $\hat{f} \in \mathcal{F}$ of the empirical risk $\widehat{\mathcal{R}}$. Assuming that the loss function $\ell(y, f(x))$ is almost surely in $[-\ell_\infty, \ell_\infty]$, we have obtained in Section 4.4.1, together with Rademacher complexities in Section 4.5, a bound of the form

$$\sup_{f \in \mathcal{F}} \big| \mathcal{R}(f) - \widehat{\mathcal{R}}(f) \big| \leqslant 2\mathrm{R}_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{n}} \sqrt{2 \log \frac{2}{\delta}}$$

with probability greater than $1 - \delta$, with the Rademacher complexity $\mathrm{R}_n(\mathcal{H})$ of the class of functions $\mathcal{H} = \{(x, y) \mapsto \ell(y, f(x)),\ f \in \mathcal{F}\}$. From Eq. (4.8), this leads to, with probability greater than $1 - \delta$:

$$\mathcal{R}(\hat{f}) \leqslant \widehat{\mathcal{R}}(\hat{f}) + 2\mathrm{R}_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{n}} \sqrt{2 \log \frac{2}{\delta}}, \tag{4.16}$$

which is a data-dependent generalization bound in high probability. Moreover, at the end of Section 4.5.1, we have seen that we could use the empirical Rademacher complexity, which can be more easily computed (with fewer assumptions).

We now consider a finite (but potentially large) number $m$ of models $\mathcal{F}_1, \dots, \mathcal{F}_m$, together with their associated loss function spaces $\mathcal{H}_1, \dots, \mathcal{H}_m$ and their generalization bounds for the empirical risk minimizer based on Rademacher complexities. In this section, we consider how to choose the best corresponding empirical risk minimizer among $\hat{f}_1, \dots, \hat{f}_m$. We consider two approaches, either based on minimizing a penalized data-generalization bound (also referred to as structural risk minimization) or simply using a validation set.

In both cases, we consider a set of positive weights $\pi_1, \dots, \pi_m$ that sum to one. We can typically choose $\pi_i = 1/m$ for all $i \in \{1, \dots, m\}$, we can also consider other choices, in particular when $m$ gets large and we are willing to put more prior weight on certain models.

### 4.6.1 Structural risk minimization

We minimize the data-dependent generalization bounds plus an additional parameter to take into account the prior on models, that is,

$$\hat{\imath} = \arg \min_{i \in \{1, \dots, m\}} \Big\{ \widehat{\mathcal{R}}(\hat{f}_i) + 2\mathrm{R}_n(\mathcal{H}_i) + \frac{2\ell_\infty}{\sqrt{n}} \sqrt{2 \log(1/\pi_i)} \Big\}.$$

We can then use Eq. (4.16) for each of the $m$ models, and with $\pi_i \delta$ instead of $\delta$, use the union bound to get, with probability greater than $1 - \delta$:

$$\mathcal{R}(\hat{f}_{\hat{\imath}}) \quad \leqslant \quad \min_{i \in \{1, \dots, m\}} \Big\{ \inf_{f_i \in \mathcal{F}_i} \mathcal{R}(f_i) + 2\mathrm{R}_n(\mathcal{H}_i) + \frac{2\ell_\infty}{\sqrt{n}} \sqrt{2 \log(1/\pi_i)} \Big\} + \frac{2\ell_\infty}{\sqrt{n}} \sqrt{2 \log(2/\delta)}.$$

For example, when $\pi_1 = \cdots = \pi_m = 1/m$, we thus obtain that the model selection procedure pays an extra price of $\sqrt{\log(m)/n}$ on top of the individual generalization bounds.

### 4.6.2 Selection based on validation set

We assume here that we have kept a proportion $\rho \in (0,1)$ of the training data as a validation set (assuming for simplicity that $\rho n$ is an integer). We then have an empirical risk based on $(1-\rho)n$ observations, we which now denote $\widehat{\mathcal{R}}^{(\text{training})}_{(1-\rho)n}$, and a validation empirical risk denoted $\widehat{\mathcal{R}}^{(\text{validation})}_{\rho n}$. Given the $m$ minimizers $\hat{f}_i$ of the training empirical risks $\widehat{\mathcal{R}}^{(\text{training})}_{(1-\rho)n}(f_i)$ over $f_i \in \mathcal{F}_i$, for $i \in \{1, \dots, m\}$, we choose $\hat{\imath}$ that minimizes the following criterion

$$\widehat{\mathcal{R}}^{(\text{validation})}_{\rho n}(\hat{f}_i) + \frac{2\ell_\infty}{\sqrt{\rho n}} \sqrt{2 \log(1/\pi_i)}$$

(for uniform weights $\pi_1 = \cdots = \pi_m = 1/m$, this is simply the minimizer of the validation risk). We can then simply use Hoeffding's inequality and the union bound to get the generalization bound:

$$\mathcal{R}(\hat{f}_{\hat{\imath}}) \leqslant \min_{i \in \{1,\dots,m\}} \mathcal{R}(\hat{f}_i) + \frac{2\ell_\infty}{\sqrt{\rho n}} \sqrt{2 \log(1/\pi_i)} + \frac{2\ell_\infty}{\sqrt{\rho n}} \sqrt{2 \log(2/\delta)},$$

which shows an extra price proportional to $1/\sqrt{\rho n}$, highlighting the fact that the validation set proportion $\rho \in (0,1)$ should not be too small. We can also obtain a result similar to the one in the section above by using the same generalization bounds based on Rademacher averages, that is,

$$\begin{aligned}\mathcal{R}(\hat{f}_{\hat{\imath}}) \quad &\leqslant \quad \min_{i \in \{1,\dots,m\}} \Big\{ \inf_{f_i \in \mathcal{F}_i} \mathcal{R}(f_i) + 2\mathrm{R}_{(1-\rho)n}(\mathcal{H}_i) + \frac{2\ell_\infty}{\sqrt{\rho n}} \sqrt{2 \log(1/\pi_i)} \Big\} \\ &\qquad + \frac{2\ell_\infty}{\sqrt{\rho n}} \sqrt{2 \log(2/\delta)} + \frac{2\ell_\infty}{\sqrt{(1-\rho)n}} \sqrt{2 \log(2/\delta)}.\end{aligned}$$

Suppose $\rho$ is bounded away from 0 and 1. In that case, we obtain a slightly worse bound than when using data-dependent bounds, with the difference that the performance of validation methods is, in practice, much better than the bound guarantees (while data-dependent bound optimization may not exhibit such adaptivity).

## 4.7 Relationship with asymptotic statistics (♦)

In this last section, we will relate the non-asymptotic analysis presented in this chapter to results from asymptotic statistics (see the comprehensive book by Van der Vaart (2000), which presents this large literature).

To make this concrete, we assume that we have a set of models $\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathbb{R},\ \theta \in \mathbb{R}^d\}$ parameterized by a vector $\theta \in \mathbb{R}^d$. We consider the empirical risk and expected risks (with a slight overloading of notations):

$$\mathcal{R}(\theta) = \mathcal{R}(f_\theta) = \mathbb{E}\big[\ell(y, f_\theta(x))\big] \ \text{ and } \ \widehat{\mathcal{R}}(\theta) = \widehat{\mathcal{R}}(f_\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i)).$$

We assume that we have a loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ (such as for regression or any of the convex surrogates for classification), which is sufficiently differentiable with respect to the second variable, so that results from Van der Vaart (2000) apply (e.g., Theorems 5.21 or 5.41 on "M-estimation", which cover empirical risk minimization). In this section, we will only report their final result and provide an intuitive justification.

We assume that $\theta_* \in \mathbb{R}^d$ is a minimizer of $\mathcal{R}(\theta)$ and that the Hessian $\mathcal{R}''(\theta_*)$ is positive-definite (it has to be positive semi-definite as $\theta_*$ is a minimizer, we assume invertibility on top of it).

We let $\hat{\theta}_n$ denote a minimizer of $\widehat{\mathcal{R}}$. Since $\mathcal{R}'(\theta_*) = 0$, and $\widehat{\mathcal{R}}'(\theta_*) = \frac{1}{n}\sum_{i=1}^n \frac{\partial \ell(y, f_\theta(x))}{\partial \theta}$, by the law of large numbers, $\widehat{\mathcal{R}}'(\theta_*)$ tends to $\mathcal{R}'(\theta_*) = 0$ (e.g., almost surely), and we should thus expect that $\hat{\theta}_n$ (which is defined through $\widehat{\mathcal{R}}'(\hat{\theta}_n) = 0$) tends to $\theta_*$ (all these statements can be made rigorous, see Van der Vaart (2000)).

Then, a Taylor expansion of $\widehat{\mathcal{R}}'$ around $\theta_*$ leads to

$$0 = \widehat{\mathcal{R}}'(\hat{\theta}_n) \approx \widehat{\mathcal{R}}'(\theta_*) + \widehat{\mathcal{R}}''(\theta_*)(\hat{\theta}_n - \theta_*).$$

By the law or large numbers, $\widehat{\mathcal{R}}''(\theta_*)$ tends to $H(\theta_*) = \mathcal{R}''(\theta_*)$ when $n$ tends to infinity, and thus we obtain:

$$\hat{\theta}_n - \theta_* \approx \mathcal{R}''(\theta_*)^{-1}\widehat{\mathcal{R}}'(\theta_*) = H(\theta_*)^{-1}\widehat{\mathcal{R}}'(\theta_*).$$

Moreover, $\widehat{\mathcal{R}}'(\theta_*)$ is the average of $n$ i.i.d. random vectors, and by the central limit theorem, it is asymptotically Gaussian with mean zero and covariance matrix equal to $\frac{1}{n}G(\theta_*) = \frac{1}{n}\mathbb{E}\big[\big(\frac{\partial \ell(y, f_\theta(x))}{\partial \theta}\big)\big(\frac{\partial \ell(y, f_\theta(x))}{\partial \theta}\big)^\top\big|_{\theta=\theta_*}\big]$. Therefore, we (intuitively) obtain that $\hat{\theta}_n$ is asymptotically Gaussian with mean $\theta_*$ and covariance matrix $\frac{1}{n}H(\theta_*)^{-1}G(\theta_*)H(\theta_*)^{-1}$.

This asymptotic result has the nice consequence that:

$$\begin{aligned} \mathbb{E}\big[\|\hat{\theta}_n - \theta_*\|_2^2\big] &\sim \frac{1}{n}\operatorname{tr}\big[H(\theta_*)^{-1}G(\theta_*)H(\theta_*)^{-1}\big] \\ \mathbb{E}\big[\mathcal{R}(\hat{\theta}_n) - \mathcal{R}(\theta_*)\big] &\sim \frac{1}{n}\operatorname{tr}\big[H(\theta_*)^{-1}G(\theta_*)\big]. \end{aligned}$$

For example, for well-specified linear regression (like analyzed in Chapter 3), it turns out that we have $G(\theta_*) = \sigma^2 H(\theta_*)$ (proof left as an exercise), and thus we recover the rate $\sigma^2 d/n$.

**Benefits of the asymptotic analysis.** As shown above, the asymptotic analysis gives a precise picture of the asymptotic behavior of empirical risk minimization. Much more than simply providing an upper-bound on $\mathbb{E}\big[\mathcal{R}(\hat{\theta}_n) - \mathcal{R}(\theta_*)\big]$, it gives also a limit Gaussian distribution for $\hat{\theta}_n$, and a *fast rate* as $O(1/n)$. Moreover, because we have limits, we can compare limits between various learning algorithms and claim (asymptotic) superiority or inferiority of one method over another, which comparing upper bounds cannot achieve.

Thus, an asymptotic analysis does not suffer from the traditional looseness of non-asymptotic bounds that rely on crude approximations, and, while they are valid even for small $n$ and often exhibit the desired behavior of $n$, are overly pessimistic.

**Pitfalls of the asymptotic analysis.** The main drawback of this analysis is that it is... asymptotic. That is, $n$ tends to infinity, and it is impossible to tell without further analysis when the asymptotic behavior will kick in. Sometimes, this is for reasonably small $n$, sometimes for large $n$. Further asymptotic expansions can be carried out, but small sample effects are hard to characterize, particularly when the underlying dimension $d$ gets large.

**Bridging the gap.** Studying the validity of the asymptotic expansion described above can be done in several ways. See, e.g., Ostrovskii and Bach (2021a) (and references therein) for finite-dimensional models, and Chapter 7 for results similar to $\sigma^2 d/n$ when the dimension of the feature space gets infinite.

## 4.8 Summary

In this chapter, we have first introduced convex surrogates for binary classification problems to avoid performing optimization on functions with values in $\{-1, 1\}$. This comes with generalization guarantees that will be extended in Chapter 13 to multiple categories and, more generally, to structured output spaces.

The chapter's core was dedicated to introducing Rademacher complexities, which are flexible tools to study estimation errors in many settings. This led to simple bounds for linear models and ball constraints, which will be extended to infinite-dimensional settings in Chapter 7 and neural networks in Chapter 9. Other frameworks exist to obtain similar bounds, such as the PAC-Bayes framework presented in Section 14.4, often leading to tighter bounds.

While the present chapter was dedicated to the statistical analysis of empirical risk minimizers, the next chapter is dedicated to optimization algorithms traditionally dedicated to finding approximate such minimizers, with notably stochastic gradient descent, which also naturally exhibits good generalization performance.

# Chapter 5

# Optimization for machine learning

**Chapter summary**

- Gradient descent: the workhorse first-order algorithm for optimization, which converges exponentially fast for well-conditioned convex problems.
- Stochastic gradient descent (SGD): the workhorse first-order algorithm for large-scale machine learning, which converges as $1/t$ or $1/\sqrt{t}$, where $t$ is the number of iterations.
- Generalization bounds through stochastic gradient descent: with only a single pass on the data, there is no risk of overfitting, and we obtain generalization bounds for unseen data.
- Variance reduction: when minimizing strongly-convex finite sums, this class of algorithms is exponentially convergent while having a small iteration complexity.

In this chapter, we present optimization algorithms based on gradient descent and analyze their performance, mainly on convex objective functions. We will consider generic algorithms that have applications beyond machine learning and algorithms dedicated to machine learning (such as stochastic gradient methods). See Nesterov (2018); Bubeck (2015) for further details.

## 5.1 Optimization in machine learning

In supervised machine learning, we are given $n$ i.i.d. samples $(x_i, y_i)$, $i = 1, \dots, n$ of a couple of random variables $(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ and the goal is to find a predictor $f : \mathcal{X} \to \mathbb{R}$

with a small risk on unseen data

$$\mathcal{R}(f) := \mathbb{E}[\ell(y, f(x))],$$

where $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a loss function. This loss is typically convex in the second argument (e.g., square loss or logistic loss, see Chapter 4), which is often considered a weak assumption.

In the empirical risk minimization approach described in Chapter 4, we choose the predictor by minimizing the empirical risk over a parameterized set of predictors, potentially with regularization. For a parameterization $\{f_\theta\}_{\theta \in \mathbb{R}^d}$ and a regularizer $\Omega : \mathbb{R}^d \to \mathbb{R}$ (e.g., $\Omega(\theta) = \|\theta\|_2^2$ or $\Omega(\theta) = \|\theta\|_1$), this requires to minimize the function

$$F(\theta) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \Omega(\theta). \tag{5.1}$$

In optimization, the function $F : \mathbb{R}^d \to \mathbb{R}$ is called the *objective function*.

In general, the minimizer has no closed form. Even when it has one (e.g., linear predictor and square loss in Chapter 3), it could be expensive to compute for large problems. We thus resort to iterative algorithms.

**Accuracy of iterative algorithms.** Solving optimization problems with high accuracy is computationally expensive, and the goal is not to minimize the training objective but the error on unseen data.

Then, which accuracy is satisfying in machine learning? If the algorithm returns $\hat{\theta}$, and we define $\theta_* \in \arg\min_\theta \mathcal{R}(f_\theta)$, we have the risk decomposition from Section 2.3.2 (where the approximation error due to the use of a specific set of models $f_\theta$, $\theta \in \Theta$ is ignored):

$$\mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) = \underbrace{\big\{\mathcal{R}(f_{\hat{\theta}}) - \widehat{\mathcal{R}}(f_{\hat{\theta}})\big\}}_{\text{estimation error}} + \underbrace{\big\{\widehat{\mathcal{R}}(f_{\hat{\theta}}) - \widehat{\mathcal{R}}(f_{\theta_*})\big\}}_{\text{optimization error}} + \underbrace{\big\{\widehat{\mathcal{R}}(f_{\theta_*}) - \mathcal{R}(f_{\theta_*})\big\}}_{\text{estimation error}},$$

where we added the second term, the *optimization error*, which will always be negative if $\hat{\theta}$ is the minimizer of $\widehat{\mathcal{R}}$, and is usually upper-bounded by $\widehat{\mathcal{R}}(f_{\hat{\theta}}) - \inf_{\theta \in \Theta} \widehat{\mathcal{R}}(f_\theta)$. It is thus sufficient to reach an optimization accuracy of the order of the estimation error (usually of the order $O(1/\sqrt{n})$ or $O(1/n)$, see Chapter 3 and Chapter 4). Note that for machine learning, the optimization error defined above corresponds to characterizing approximate solutions through function values. While this will be one central focal point in this chapter, we will consider other performance measures.

In this chapter, we will first look at minimization without focusing on machine learning problems (Section 5.2), with both smooth and non-smooth objective functions. We will then look at stochastic gradient descent in Section 5.4, which can be used to obtain bounds on both the training and testing risks. We then briefly present adaptive methods in

Section 5.4.2, bias-variance decompositions for least-squares in Section 5.4.3, and variance reduction in Section 5.4.4.

⚠ The notation $\theta_*$ may mean different things in optimization and machine learning: minimizer of the *regularized empirical risk*, or minimizer of the *expected risk*. For the sake of clarity, we will use the notation $\eta_*$ for the minimizer of empirical (potentially regularized) risk, that is, when we look at optimization problems, and $\theta_*$ for the minimizer of the expected risk, that is, when we look at statistical problems.

⚠ Sometimes, we mention solving a problem with *high* precision. This corresponds to a *low* optimization error.

In this chapter, we primarily focus on gradient descent methods for convex optimization problems, which, in learning terms, correspond to predictors that are linear in their parameter (an assumption that will be relaxed in subsequent chapters) and a convex loss function such as the logistic loss or the square loss. We first consider so-called "batch methods" that do not use the finite sum structure of the objective function in Eq. (5.1) before moving on to the stochastic gradient method that does take into account this structure for enhanced computational efficiency.

## 5.2 Gradient descent

Suppose we want to solve, for a function $F : \mathbb{R}^d \to \mathbb{R}$, the optimization problem

$$\min_{\theta \in \mathbb{R}^d} F(\theta).$$

We assume that we are given access to certain "oracles": the *$k$-th-order oracle* corresponds to the access to: $\theta \mapsto (F(\theta), F'(\theta), \dots, F^{(k)}(\theta))$, that is all partial derivatives up to order $k$. All algorithms will call these oracles; thus, their computational complexity will depend directly on the complexity of this oracle. For example, for least-squares with a design matrix in $\mathbb{R}^{n \times d}$, computing a single gradient of the empirical risk costs $O(nd)$.

In this section, for the algorithms and proofs, we do not assume that the function $F$ is the regularized empirical risk, but this situation will be our motivating example throughout. We will study the following first-order algorithm.

**Algorithm 5.1 (Gradient descent (GD))** *Pick $\theta_0 \in \mathbb{R}^d$ and for $t \geqslant 1$, let*

$$\theta_t = \theta_{t-1} - \gamma_t F'(\theta_{t-1}), \tag{5.2}$$

*for a well (potentially adaptively) chosen step-size sequence $(\gamma_t)_{t \geqslant 1}$.*

For machine learning problems where the empirical risk is minimized, computing the gradient $F'(\theta_{t-1})$ requires computing all gradients of $\theta \mapsto \ell(y_i, f_\theta(x_i))$, and averaging them.

There are many ways to choose the step-size $\gamma_t$, either constant, decaying, or through a line search.[1] In practice, using some form of line search is usually advantageous and is implemented in most applications. See Armijo (1966) and Goldstein (1962) for convergence guarantees with typical procedures. In this chapter, since we want to focus on the simplest algorithms and proofs, we will focus on step-sizes that depend explicitly on problem constants and sometimes on the iteration number. When gradients are not available, gradient estimates may be built from function values (see, e.g., Nesterov and Spokoiny, 2017, and Chapter 11). Note that the differences between convergence rates with and without line searches are generally not significant (see Exercise 5.2 below for quadratic functions). At the same time, practical behavior is significantly improved with line search.

We start with the simplest example, namely convex quadratic functions, where the most important concepts already appear.

### 5.2.1 Simplest analysis: ordinary least-squares

We start with a case where the analysis is explicit: ordinary least squares (see Chapter 3 for the statistical analysis). Let $\Phi \in \mathbb{R}^{n\times d}$ be the design matrix and $y \in \mathbb{R}^n$ the vector of responses. Least-squares estimation amounts to finding a minimizer $\eta_*$ of

$$F(\theta) = \frac{1}{2n}\|\Phi\theta - y\|_2^2. \tag{5.3}$$

⚠ A factor of $\frac{1}{2}$ has been added compared to Chapter 3 to get nicer looking gradients.

The gradient of $F$ is $F'(\theta) = \frac{1}{n}\Phi^\top(\Phi\theta - y) = \frac{1}{n}\Phi^\top\Phi\theta - \frac{1}{n}\Phi^\top y$. Thus, denoting $H = \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d\times d}$ the Hessian matrix (equal for all $\theta$, denoted $\widehat{\Sigma}$ in Chapter 3), minimizers $\eta_*$ are characterized by

$$H\eta_* = \frac{1}{n}\Phi^\top y.$$

Since $\frac{1}{n}\Phi^\top y \in \mathbb{R}^d$ is in the column space of $H$, there is always a minimizer, but unless $H$ is invertible, the minimizer is not unique. But all minimizers $\eta_*$ have the same function value $F(\eta_*)$, and we have, from a simple exact Taylor expansion (and using $F'(\eta_*) = 0$):

$$F(\theta) - F(\eta_*) = F'(\eta_*)^\top(\theta - \eta_*) + \frac{1}{2}(\theta - \eta_*)^\top H(\theta - \eta_*) = \frac{1}{2}(\theta - \eta_*)^\top H(\theta - \eta_*).$$

Two quantities will be important in the following developments: the largest eigenvalue $L$ and the smallest eigenvalue $\mu$ of the Hessian matrix $H$. As a consequence of the convexity of the objective, we have $0 \leqslant \mu \leqslant L$. We denote by $\kappa = \frac{L}{\mu} \geqslant 1$ the *condition number*.

Note that for least-squares, $\mu$ is the lowest eigenvalue of the non-centered empirical covariance matrix and that it is zero as soon as $d > n$, and, in most practical cases, *very* small. When adding a regularizer $\frac{\lambda}{2}\|\theta\|_2^2$ (like in ridge regression), then $\mu \geqslant \lambda$ (but then $\lambda$ typically decreases with $n$, often between $\frac{1}{\sqrt{n}}$ and $\frac{1}{n}$, see Chapter 7 for more details).

[1] See, e.g., https://en.wikipedia.org/wiki/Line_search.

**Closed-form expression.** Gradient descent iterates with fixed step-size $\gamma_t = \gamma$ can be computed in closed form:

$$\theta_t = \theta_{t-1} - \gamma F'(\theta_{t-1}) = \theta_{t-1} - \gamma\big[\frac{1}{n}\Phi^\top(\Phi\theta_{t-1} - y)\big] = \theta_{t-1} - \gamma H(\theta_{t-1} - \eta_*),$$

leading to

$$\theta_t - \eta_* = \theta_{t-1} - \eta_* - \gamma H(\theta_{t-1} - \eta_*) = (I - \gamma H)(\theta_{t-1} - \eta_*),$$

that is, we have a linear recursion, and we can unroll the recursion and now write

$$\theta_t - \eta_* = (I - \gamma H)^t(\theta_0 - \eta_*).$$

We can now look at various measures of performance:

$$\begin{aligned} \|\theta_t - \eta_*\|_2^2 &= (\theta_0 - \eta_*)^\top (I - \gamma H)^{2t}(\theta_0 - \eta_*) \\ F(\theta_t) - F(\eta_*) &= \frac{1}{2}(\theta_0 - \eta_*)^\top (I - \gamma H)^{2t} H (\theta_0 - \eta_*). \end{aligned}$$

The two optimization performance measures differ by the presence of the Hessian matrix $H$ in the measure based on function values.

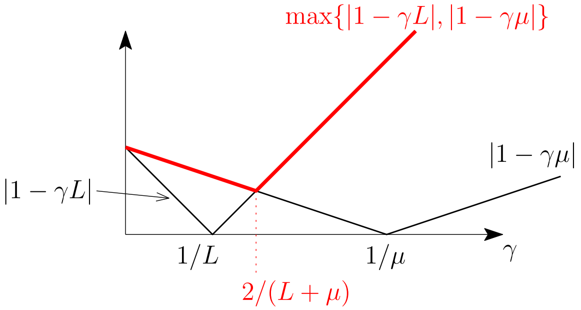
**Convergence in distance to the minimizer.** If we hope to have $\|\theta_t - \eta_*\|_2^2$ going to zero, we need to have a single minimizer $\eta_*$, and thus $H$ has to be invertible, that is $\mu > 0$. Given the form of $\|\theta_t - \eta_*\|_2^2$, we simply need to bound the eigenvalues of $(I - \gamma H)^{2t}$ (since for a positive semi-definite matrix $M$, $u^\top M u \leqslant \lambda_{\max}(M)\|u\|_2^2$ for all vectors $u$).

The eigenvalues of $(I - \gamma H)^{2t}$ are exactly $(1 - \gamma\lambda)^{2t}$ for $\lambda$ an eigenvalue of $H$ (all of them are in the interval $[\mu, L]$). Thus all the eigenvalues of $(I - \gamma H)^{2t}$ have magnitude less than

$$\Big(\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda|\Big)^{2t}.$$

We can then have several strategies for choosing the step-size $\gamma$:

- Optimal choice: one can check that minimizing $\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda|$ is done by setting $\gamma = 2/(\mu + L)$, with an optimal value equal to $\frac{\kappa - 1}{\kappa + 1} = 1 - \frac{2}{\kappa + 1} \in (0, 1)$. See geometric "proof" below.



- Choice independent of $\mu$: with the simpler (slightly smaller) choice $\gamma = 1/L$, we get $\max_{\lambda\in[\mu,L]} |1-\gamma\lambda| = (1-\frac{\mu}{L}) = (1-\frac{1}{\kappa})$, which is only slightly larger than the value for the optimal choice. Note that all step-sizes strictly less than $2/L$ will lead to exponential convergence.

For example, with the weaker choice $\gamma = 1/L$, we get:

$$\|\theta_t - \eta_*\|_2^2 \leqslant \Big(1-\frac{1}{\kappa}\Big)^{2t} \|\theta_0 - \eta_*\|_2^2,$$

which is often referred to as exponential, geometric, or linear convergence.

⚠ The denomination "linear" is sometimes confusing and corresponds to a number of significant digits that grows linearly with the number of iterations.

We can further bound $\big(1-\frac{1}{\kappa}\big)^{2t} \leqslant \exp(-1/\kappa)^{2t} = \exp(-2t/\kappa)$, and thus the characteristic time of convergence is of order $\kappa$. We will often make the calculation $\varepsilon = \exp(-2t/\kappa) \Leftrightarrow t = \frac{\kappa}{2}\log\frac{1}{\varepsilon}$. Thus, for a relative reduction of squared distance to the optimum of $\varepsilon$, we need at most $t = \frac{\kappa}{2}\log\frac{1}{\varepsilon}$ iterations.

For $\kappa = +\infty$, the result remains true but simply says that for all minimizers $\|\theta_t - \eta_*\|_2^2 \leqslant \|\theta_0 - \eta_*\|_2^2$, which is a good sign (the algorithm does not move away from minimizers) but not indicative of any form of convergence. We will need to use a different criterion.

**Convergence in function values.** Using the same step-size $\gamma = 1/L$ as above, and using the upper-bound on eigenvalues of $(I-\gamma H)^{2t}$ (which are all less than $(1-1/\kappa)^{2t}$), we get

$$F(\theta_t) - F(\eta_*) \leqslant \Big(1-\frac{1}{\kappa}\Big)^{2t}[F(\theta_0)-F(\eta_*)] \leqslant \exp(-2t/\kappa)[F(\theta_0)-F(\eta_*)]. \tag{5.4}$$

When $\kappa < \infty$ (that is, $\mu > 0$), we also obtain linear convergence for this criterion, but when $\kappa = \infty$, this is non-informative.

To obtain a convergence rate, we will need to bound the eigenvalues of $(I-\gamma H)^{2t}H$ instead of $(I-\gamma H)^{2t}$. The key difference is that for eigenvalues $\lambda$ of $H$ which are close to zero, $(1-\gamma\lambda)^{2t}$ does not have a strong contracting effect, but they count less as they are multiplied by $\lambda$ in the bound.

We can make this trade-off precise, for $\gamma \leqslant 1/L$, as

$$\begin{aligned} \big|\lambda(1-\gamma\lambda)^{2t}\big| &\leqslant \lambda\exp(-\gamma\lambda)^{2t} = \lambda\exp(-2t\gamma\lambda) \\ &= \frac{1}{2t\gamma}2t\gamma\lambda\exp(-2t\gamma\lambda) \leqslant \frac{1}{2t\gamma}\sup_{\alpha\geqslant 0}\alpha\exp(-\alpha) = \frac{1}{2et\gamma} \leqslant \frac{1}{4t\gamma}, \end{aligned}$$

where we used that $\alpha e^{-\alpha}$ is maximized over $\mathbb{R}_+$ at $\alpha = 1$ (as the derivative is $e^{-\alpha}(1-\alpha)$).

This leads to, with the largest step-size $\gamma = 1/L$:

$$F(\theta_t) - F(\eta_*) \leqslant \frac{1}{8t\gamma}\|\theta_0 - \eta_*\|_2^2 = \frac{L}{8t}\|\theta_0 - \eta_*\|_2^2. \tag{5.5}$$

We can make the following observations:

- ⚠ The convergence results in $\exp(-2t/\kappa)$ in Eq. (5.4) for invertible Hessians or $1/t$ in general in Eq. (5.5) are only upper-bounds! It is good to understand the gap between the bounds and the actual performance, as this is possible for quadratic objective functions.

  For the exponentially convergent case, the lowest eigenvalue $\mu$ dictates the rate for all eigenvalues. So, if the eigenvalues are well-spread (or if only one eigenvalue is very small), there can be quite a strong discrepancy between the bound and the actual behavior.

  For the rate in $1/t$, the bound in eigenvalues is tight when $t\gamma\lambda$ is of order 1, namely when $\lambda$ is of order $1/(t\gamma)$. Thus, to see an $O(1/t)$ convergence rate in practice, we need to have sufficiently many small eigenvalues. As $t$ grows, we often go to a local linear convergence phase where the smallest non-zero eigenvalue of $H$ kicks in. See the simulations and the exercise below.

  **Exercise 5.1** *Let $\mu_+$ be the smallest non-zero eigenvalue of $H$. Show that gradient descent is linearly convergent with the contracting rate $(1 - \mu_+/L)$.*

- From errors to numbers of iterations: as already mentioned, the bound in Eq. (5.4) says that after $t$ steps, the reduction in suboptimality in function values is multiplied by $\varepsilon = \exp(-2t/\kappa)$. This can be reinterpretated as a need of $t = \frac{\kappa}{2} \log \frac{1}{\varepsilon}$ iterations to reach a relative error $\varepsilon$.

- Can an algorithm having the same access to oracles of $F$ do better?

  If we have access to matrix-vector products with the matrix $\Phi$, then the conjugate gradient algorithm can be used with convergence rates in $\exp(-t/\sqrt{\kappa})$ and $1/t^2$ (see Golub and Loan, 1996). With only access to gradients of $F$ (which is a bit weaker), Nesterov acceleration (see Section 5.2.5 below) will also lead to the same convergence rates, which are then optimal (for a sense to be defined later in this chapter and in more details in Chapter 15).

- Can we extend beyond least-squares? The convergence results above will generalize to convex functions (see Section 5.2.2) but with less direct proofs. Non-convex objectives are discussed in Section 5.2.6.

**Experiments.** We consider two quadratic optimization problems in dimension $d = 1000$, with two different decays of eigenvalues $(\lambda_k)_{k\in\{1,\dots,d\}}$ for the Hessian matrix $H$, one as $1/k$ (in blue below) and one in $1/k^2$ (in red below), and for which we plot in Figure 5.1 the performance for function values, both in semi-logarithm plots (left) and full-logarithm plots (right). For slow decays (blue), we see the linear convergence kicking in (line in the left "semi-log" plot), while for fast decays (red), we obtain a polynomial rate that is not exponential (line in the right "log-log" plot). Note that the bound in Eq. (5.5) is very pessimistic and does not lead to the correct power of $t$ (which, as can be checked as an exercise, should be $1/\sqrt{t}$ for $t$ small enough compared to $d$).

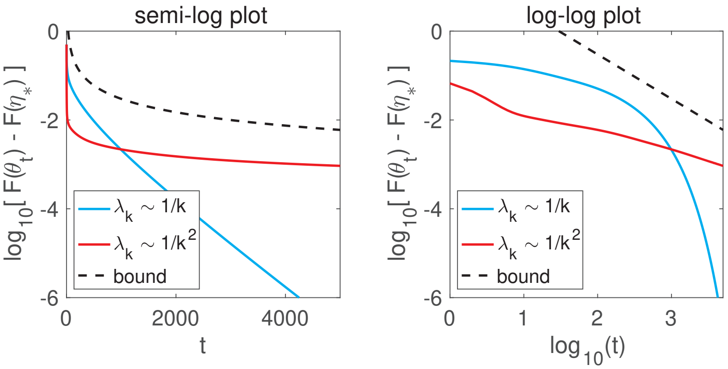


Figure 5.1: Gradient descent on two least-squares problems with step-size $\gamma = 1/L$, and two different sets of eigenvalues $(\lambda_k)_{k\in\{1,\dots,d\}}$ of the Hessian, together with the bound from Eq. (5.5). Left: semi-logarithmic scale. Right: joint logarithmic scale.

**Exercise 5.2** *(exact line search ♦) For the quadratic objective in Eq. (5.3), show that the optimal step-size $\gamma_t$ in Eq. (5.2) is equal to $\gamma_t = \frac{\|F'(\theta_{t-1})\|_2^2}{F'(\theta_{t-1})^\top H F'(\theta_{t-1})}$. Show that when $F$ is strongly-convex, $F(\theta_t) - F(\eta_*) \leqslant \left(\frac{\kappa-1}{\kappa+1}\right)^2 \big[F(\theta_{t-1}) - F(\eta_*)\big]$, and compare the rate with constant step-size gradient descent.*
*Hint: prove and use the Kantorovich inequality* $\sup_{\|z\|_2=1} z^\top H z z^\top H^{-1} z = \frac{(L+\mu)^2}{4\mu L}$.

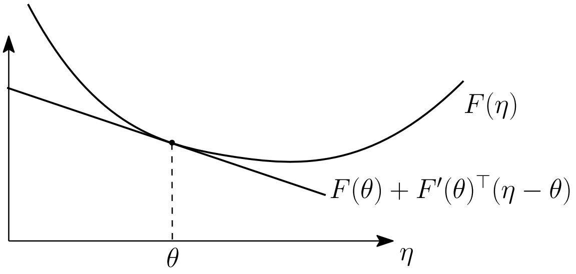
### 5.2.2 Convex functions and their properties

We now wish to analyze GD (and later its stochastic version SGD) in a broader setting. We will always assume convexity, although these algorithms are also used (and can sometimes also be analyzed) when this assumption does not hold (see Section 5.2.6). In other words, convexity is most often used for analysis rather than to define the algorithm.

**Definition 5.1 (Convex function)** *A differentiable function $F : \mathbb{R}^d \to \mathbb{R}$ is said* convex *if and only if*

$$F(\eta) \geqslant F(\theta) + F'(\theta)^\top(\eta - \theta), \qquad \forall \eta, \theta \in \mathbb{R}^d. \tag{5.6}$$

This corresponds to the function $F$ being above its tangent at $\theta$, as illustrated below.



If $f$ is twice-differentiable, this is equivalent to requiring $F''(x) \succcurlyeq 0$, $\forall x \in \mathbb{R}^d$; here $\succcurlyeq$ denotes the semidefinite partial ordering—also called the Löwner order—characterized by $A \succcurlyeq B \Leftrightarrow A - B$ is positive semidefinite, see Boyd and Vandenberghe (2004); Bhatia (2009).

An important consequence that we will use a lot in this chapter is, for all $\theta \in \mathbb{R}^d$ (and using $\eta = \eta_*$)

$$F(\eta_*) \geqslant F(\theta) + F'(\theta)^\top(\eta_* - \theta) \quad \Leftrightarrow \quad F(\theta) - F(\eta_*) \leqslant F'(\theta)^\top(\theta - \eta_*), \tag{5.7}$$

that is, the distance to optimum in function values is upper bounded by a function of the gradient (note that it provides proof that $F'(\theta) = 0$ implies that $\theta$ is a global minimizer of $F$).

A more general definition of convexity (without gradients) is that $\forall \theta, \eta \in \mathbb{R}^d$ and $\alpha \in [0,1]$,

$$F(\alpha\eta + (1-\alpha)\theta) \leqslant \alpha F(\eta) + (1-\alpha)F(\theta),$$

which generalizes to the usual Jensen's inequality below.[2]

**Proposition 5.1 (Jensen's inequality)** *If $F : \mathbb{R}^d \to \mathbb{R}$ is convex and $\mu$ is a probability measure on $\mathbb{R}^d$, then*

$$F\Big(\int_{\mathbb{R}^d} \theta d\mu(\theta)\Big) \leqslant \int_{\mathbb{R}^d} F(\theta) d\mu(\theta). \tag{5.8}$$

*In words: "the image of the average is smaller than the average of the images".*

⚠ When using Jensen's inequality, be extra careful in the direction of the inequality.

**Exercise 5.3** *Assume that the function $F : \mathbb{R}^d \to \mathbb{R}$ is* strictly convex*, that is, $\forall \theta, \eta \in \mathbb{R}^d$ such that $\theta \neq \eta$, and $\alpha \in (0,1)$, $F(\alpha\eta + (1-\alpha)\theta) < \alpha F(\eta) + (1-\alpha)F(\theta)$. Show that there is equality in Jensen's inequality in Eq. (5.8) if and only if the random variable $\theta$ is almost surely constant.*

The class of convex functions satisfies the following stability properties (proofs left as an exercise); for more properties on convex functions, see Boyd and Vandenberghe (2004):

- If $(F_j)_{j \in \{1,\dots,m\}}$ are convex and $(\alpha_j)_{j \in \{1,\dots,m\}}$ are nonnegative, then $\sum_{j=1}^m \alpha_j F_j$ and $\max_{j \in \{1,\dots,m\}} F_j$ are convex.
- If $F : \mathbb{R}^d \to \mathbb{R}$ is convex and $A : \mathbb{R}^{d'} \to \mathbb{R}^d$ is linear then $F \circ A : \mathbb{R}^{d'} \to \mathbb{R}$ is convex.
- If $F : \mathbb{R}^{d_1 + d_2} \to \mathbb{R}$ is convex, so is the function $x_1 \mapsto \inf_{x_2 \in \mathbb{R}^{d_2}} F(x_1, x_2)$ on $\mathbb{R}^{d_1}$.

**Classical machine learning example.** Problems of the form in Eq. (5.1) are convex if the loss $\ell$ is convex in the second variable, $f_\theta(x)$ is linear in $\theta$, and $\Omega$ is convex.

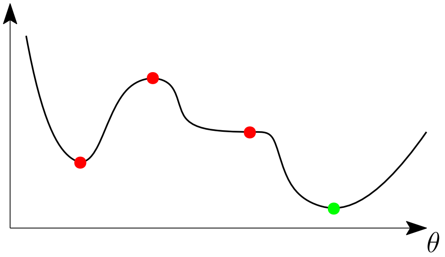
[2] See several applications in `https://francisbach.com/jensen-inequality/`.

**Global optimality from local information.** It is also worth emphasizing the following property (immediate from the definition).

**Proposition 5.2** *Assume that $F : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable. Then $\eta_* \in \mathbb{R}^d$ is a global minimizer of $F$ if and only if*

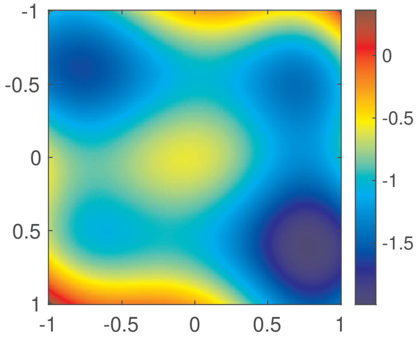
$$F'(\eta_*) = 0.$$

This implies that for convex functions, we only need to look for stationary points. This is *not* the case for potentially non-convex functions. For example, in one dimension below, all red points are stationary points that are not the global minimum (which is in green).



The situation is even more complex in higher dimensions. Note that without convexity assumptions, optimization of Lipschitz-continuous functions will need exponential time in dimension in the worst case (see Section 15.2.2).

**Exercise 5.4** *Identify all stationary points in the function in $\mathbb{R}^2$ depicted below.*



### 5.2.3 Analysis of GD for strongly convex and smooth functions

The analysis of optimization algorithms requires assumptions on the objective functions, like the ones introduced in this section. From these assumptions, additional properties are derived (typically inequalities), and then most convergence proofs look for a "Lyapunov function" (sometimes called a "potential function") that goes down along the iterations. More precisely, if $V : \mathbb{R}^d \to \mathbb{R}_+$ is such that $V(\theta_t) \leqslant (1-\alpha)V(\theta_{t-1})$, then $V(\theta_t) \leqslant (1-\alpha)^t V(\theta_0)$ and we obtain linear convergence. The art is then to find the appropriate

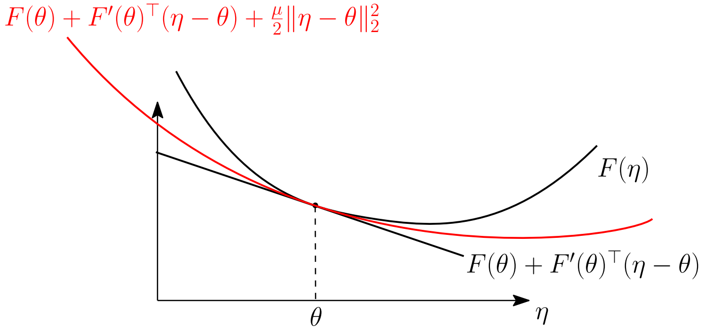
Lyapunov function; for slower convergence rates, weaker forms of decrease for Lyapunov functions will be considered.

We first consider an assumption allowing exponential convergence rates.

**Definition 5.2 (Strong convexity)** *A differentiable function $F$ is said $\mu$-strongly convex, with $\mu > 0$, if and only if*

$$F(\eta) \geqslant F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|_2^2, \quad \forall \eta, \theta \in \mathbb{R}^d. \tag{5.9}$$

The function $F$ is strongly-convex if and only if the function $F$ is strictly above its tangent and the difference is at least quadratic in the distance to the point where the two coincide. This notably allows us to define quadratic lower bounds on $F$. See below.


$F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|_2^2$
$F(\eta)$
$F(\theta) + F'(\theta)^\top(\eta - \theta)$
$\theta$
$\eta$

For twice differentiable functions, this is equivalent to $F''(\theta) \succcurlyeq \mu I$ for all $\theta$, that is, all eigenvalues of $F''(\theta)$ are greater than or equal to $\mu$ (see Nesterov, 2018), but non-smooth functions can be strongly-convex, since, as a consequence of the exercise below, we can add $\frac{\mu}{2}\|\cdot\|_2^2$ to any (potentially non-smooth) convex function to make it $\mu$-strongly-convex.

**Exercise 5.5** *Show that $F : \mathbb{R}^d \to \mathbb{R}$ is $\mu$-strongly-convex if and only if the function $\theta \mapsto F(\theta) - \frac{\mu}{2}\|\theta\|_2^2$ is convex.*

**Exercise 5.6** *Show that if $F : \mathbb{R}^d \to \mathbb{R}$ is $\mu$-strongly-convex, then it has a unique minimizer.*

**Exercise 5.7** *Show that the differentiable function $F : \mathbb{R}^d \to \mathbb{R}$ is $\mu$-strongly convex if and only if for all $\theta, \eta \in \mathbb{R}^d$, $\|F'(\theta) - F'(\eta)\|_2 \geqslant \mu\|\theta - \eta\|_2$.*

**Strong convexity through regularization.** When an objective function $F$ is convex, then $F + \frac{\mu}{2}\|\cdot\|_2^2$ is $\mu$-strongly convex (proof left as an exercise). In practice, in machine learning problems with linear models, so that the empirical risk is convex, strong convexity most often comes from the regularizer (and thus $\mu$ decays with $n$), leading to condition numbers that grow with $n$ (typically in $\sqrt{n}$ or $n$). While the regularizer was added in Section 3.6 to improve generalization, we see in this section that it also leads to faster optimization algorithms, showing that statistical and optimization performances are often aligned.

**Łojasiewicz inequality.** Strong convexity implies that $F$ admits a unique minimizer $\eta_*$, which is characterized by $F'(\eta_*) = 0$. Moreover, this guarantees that the gradient is large when a point is far from optimal (in function values):

**Lemma 5.1 (Łojasiewicz inequality)** *If $F$ is differentiable and $\mu$-strongly convex with unique minimizer $\eta_*$, then we have:*

$$\|F'(\theta)\|_2^2 \geqslant 2\mu(F(\theta) - F(\eta_*)), \quad \forall \theta \in \mathbb{R}^d.$$

**Proof** The right-hand side in Definition 5.2 is strongly convex in $\eta$ and minimized with $\tilde{\eta} = \theta - \frac{1}{\mu}F'(\theta)$. Plugging this value into the bound and taking $\eta = \eta_*$ in the left-hand side, we get $F(\eta_*) \geqslant F(\theta) - \frac{1}{\mu}\|F'(\theta)\|_2^2 + \frac{1}{2\mu}\|F'(\theta)\|_2^2 = F(\theta) - \frac{1}{2\mu}\|F'(\theta)\|_2^2$. The conclusion follows by rearranging. ■

Note that while strong convexity is a sufficient condition for the Łojasiewicz inequality, it is unnecessary (see, e.g., Section 12.1.1).

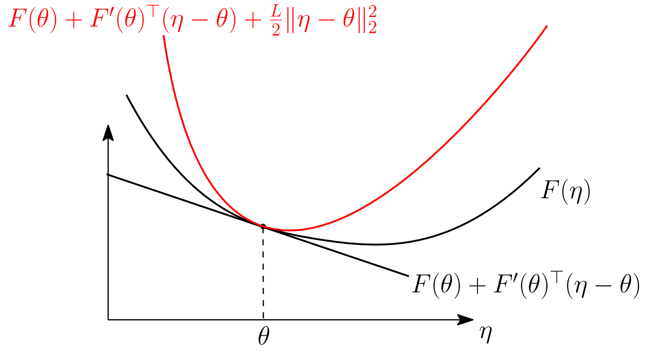
To obtain exponential convergence rates, strong-convexity is typically associated with smoothness, which we now define.

**Definition 5.3 (Smoothness)** *A differentiable function $F$ is said $L$-smooth if and only if*

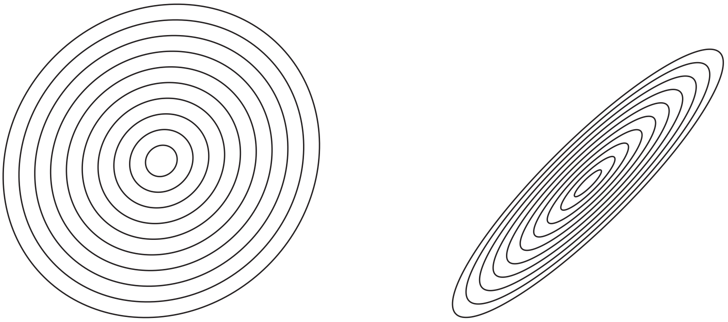
$$|F(\eta) - F(\theta) - F'(\theta)^\top(\eta - \theta)| \leqslant \frac{L}{2}\|\theta - \eta\|^2, \quad \forall \theta, \eta \in \mathbb{R}^d. \tag{5.10}$$

This is equivalent to $F$ having a $L$-Lipschitz-continuous gradient with respect to the $\ell_2$-norm, i.e., $\|F'(\theta) - F'(\eta)\|_2^2 \leqslant L^2\|\theta - \eta\|_2^2$, $\forall \theta, \eta \in \mathbb{R}^d$. For twice differentiable functions, this is equivalent to $-LI \preccurlyeq F''(\theta) \preccurlyeq LI$ (see Nesterov, 2018). If the function is also $\mu$-strongly convex, then all eigenvalues of all Hessians are in the interval $[\mu, L]$.

Note that when $F$ is convex and $L$-smooth, we have a quadratic upper bound that is tight at any given point (strong convexity implies the corresponding lower bound with $L$ replaced by $\mu$). See below.


$F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{L}{2}\|\eta - \theta\|_2^2$
$F(\eta)$
$F(\theta) + F'(\theta)^\top(\eta - \theta)$
$\theta$
$\eta$

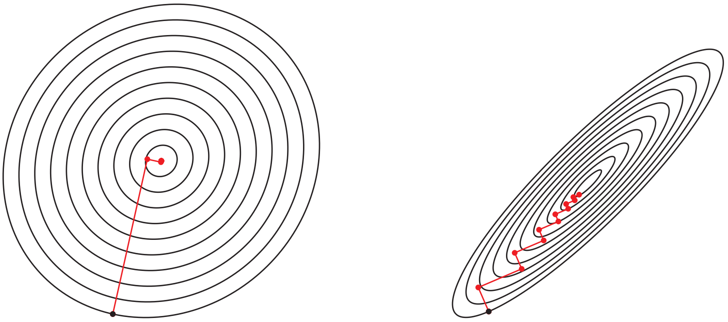
When a function is both smooth and strongly convex, we denote by $\kappa = L/\mu \geqslant 1$ its condition number (we recover the definition of Section 5.2.1 for quadratic functions). See examples below of level sets of functions with varying condition numbers: the condition number impacts the shapes of the level sets.



(small $\kappa = L/\mu$) (large $\kappa = L/\mu$)

The performance of gradient descent will depend on this condition number (see steepest descent below, that is, gradient descent with exact line search): with a small condition number (left), we get fast convergence, while for a large condition number (right), we get oscillations.

**Exercise 5.8** *(♦) We consider the angle $\alpha$ between the descent direction $-F'(\theta)$ and the deviation to optimum $\theta - \eta_*$, defined through $\cos\alpha = \frac{F'(\theta)^\top(\theta-\eta_*)}{\|F'(\theta)\|\cdot\|\theta-\eta_*\|_2}$. Show that for a $\mu$-strongly-convex, $L$-smooth quadratic function, $\cos\alpha \geqslant \frac{2\sqrt{\mu L}}{L+\mu}$ (hint: prove and use the Kantorovich inequality $\sup_{\|z\|_2=1} z^\top H z z^\top H^{-1} z = \frac{(L+\mu)^2}{4\mu L}$). (♦♦) Show that the same result holds without the assumption that $F$ is quadratic (hint: use the co-coercivity of the function $\theta \mapsto F(\theta) - \frac{\mu}{2}\|\theta\|_2^2$, from Prop. 5.4).*



(small $\kappa = L/\mu$) (large $\kappa = L/\mu$)

For machine learning problems, such as linear predictions and smooth losses (square or logistic), we have smooth problems. If we use a squared $\ell_2$-regularizer $\frac{\mu}{2}\|\cdot\|_2^2$ , we get a $\mu$-strongly convex problem. Note that when using regularization, as explained in Chapters 3 and 4, the value of $\mu$ decays with $n$, typically between $1/n$ and $1/\sqrt{n}$, leading to condition numbers between $\sqrt{n}$ and $n$.

In this context, gradient descent on the empirical risk is often called a "batch" technique because all the data points are accessed at every iteration.

In the next proposition, we show that gradient descent converges exponentially for such smooth and strongly-convex problems, thus extending the result for quadratic functions from Section 5.2.1.

**Proposition 5.3 (Convergence of GD for smooth strongly-convex functions)** *Assume that $F$ is $L$-smooth and $\mu$-strongly convex. Choosing $\gamma_t = 1/L$, the iterates $(\theta_t)_{t \geqslant 0}$ of GD on $F$ satisfy*

$$F(\theta_t) - F(\eta_*) \leqslant \Big(1 - \frac{1}{\kappa}\Big)^t (F(\theta_0) - F(\eta_*)) \leqslant \exp(-t/\kappa)(F(\theta_0) - F(\eta_*)).$$

**Proof** By the smoothness inequality in Eq. (5.10) applied to $\theta_{t-1}$ and $\theta_{t-1} - F'(\theta_{t-1})/L$, we have the following descent property, with $\gamma_t = 1/L$,

$$\begin{aligned} F(\theta_t) &= F\big(\theta_{t-1} - F'(\theta_{t-1})/L\big) \leqslant F(\theta_{t-1}) + F'(\theta_{t-1})^\top(-F'(\theta_{t-1})/L) + \frac{L}{2}\|-F'(\theta_{t-1})/L\|_2^2 \\ &= F(\theta_{t-1}) - \frac{1}{L}\|F'(\theta_{t-1})\|_2^2 + \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2. \end{aligned}$$

Rearranging, we get

$$F(\theta_t) - F(\eta_*) \leqslant F(\theta_{t-1}) - F(\eta_*) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2.$$

Using Lemma 5.1, it follows

$$F(\theta_t) - F(\eta_*) \leqslant (1 - \mu/L)(F(\theta_{t-1}) - F(\eta_*)) \leqslant \exp(-\mu/L)(F(\theta_{t-1}) - F(\eta_*)).$$

We conclude by recursion and with the definition $\kappa = L/\mu$. ■

We can make the following observations:

- As mentioned before, we necessarily have $\mu \leqslant L$; the ratio $\kappa := L/\mu$ is called the *condition number*. It is a property of the objective function, which may be hard or easy to minimize. It is not invariant by linear changes of variables $\theta \to A\theta$, where $A$ is an invertible linear map; finding a good $A$ to reduce the condition number is the main principle behind "preconditioning" techniques (see, e.g., Nocedal and Wright, 1999 for more details and the end of Section 5.2.5).
- If we only assume that the function is smooth and convex (not strongly convex), then GD with constant step-size $\gamma = 1/L$ also converges when a minimizer exists, but at a slower rate in $O(1/t)$. See Section 5.2.4 below.
- Choosing the step-size only requires an upper bound $L$ on the smoothness constant (if it is over-estimated, the convergence rate only degrades slightly).

- Writing the update $(\theta_t - \theta_{t-1})/\gamma = -F'(\theta_{t-1})$, this algorithm can be seen, under the smoothness assumption, as the discretization of the gradient flow
$$\frac{d}{dt}\eta(t) = -F'(\eta),$$
where $\eta(t\gamma) \approx \theta_t$. This analogy can lead to several insights and proof ideas (see, e.g., Scieur et al., 2017).
- For this class of functions (convex and smooth), first-order methods exist that achieve a faster rate, showing that gradient descent is not optimal. However, these improved algorithms also have drawbacks (lack of adaptivity, instability to noise, etc.). See below.

**Exercise 5.9** *Compute all constants for $\ell_2$-regularized logistic regression and for ridge regression.*

**Adaptivity.** Note that gradient descent is adaptive to strong convexity: the exact same algorithm applies to both strongly convex and convex cases, and the two bounds apply. This adaptivity is important in practice, as often, locally around the global optimum, the strong convexity constant converges to the minimal eigenvalue of the Hessian at $\eta_*$, which can be significantly larger than $\mu$ (the global constant).

**Fenchel conjugate (♦).** Given some convex function $F : \mathbb{R}^d \to \mathbb{R}$, an important tool is the Fenchel-Legendre conjugate $F^*$ defined as $F^*(\alpha) = \sup_{\theta\in\mathbb{R}^d} \alpha^\top\theta - F(\theta)$. In particular, when we allow extended-value functions (which may take the value $+\infty$), we can represent functions defined on a convex domain, and we have, under simple regularity conditions, that the conjugate of the conjugate of a convex function is the function itself. Thus, any convex function can be seen as a maximum of affine functions. Moreover, suppose the original function is not convex. In that case, the bi-conjugate is often referred to as the convex envelope and is the tightest convex lower-bound (this is often used when designing convex relaxations of non-convex problems). Moreover, using Fenchel conjugation is crucial when dealing with convex duality (which we will not cover in this chapter). See Boyd and Vandenberghe (2004) for details.

**Exercise 5.10** *Let $F$ be an $L$-smooth convex function on $\mathbb{R}^d$. Show that its Fenchel conjugate is $(1/L)$-strongly convex.*

**Exercise 5.11** *Let $F$ be an $L$-smooth convex function on $\mathbb{R}^d$, and $F^*$ its Fenchel conjugate. Show that for any $\theta, z \in \mathbb{R}^d$, we have $F(\theta)+F^*(z)-z^\top\theta \geqslant 0$, if and only if $z = F'(\theta)$ (this is the Fenchel-Young inequality). (♦) Show in addition that $F(\theta) + F^*(z) - z^\top\theta \geqslant \frac{1}{2L}\|z - F'(\theta)\|_2^2$.*

### 5.2.4 Analysis of GD for convex and smooth functions (♦)

To obtain the $1/t$ convergence rate without strong-convexity (like we got in Section 5.2.1 for quadratic functions), we will need an extra property of convex, smooth functions, sometimes called “co-coercivity”. This is an instance of inequalities we need to use to circumvent the lack of closed form for iterations.

**Proposition 5.4 (co-coercivity)** *If $F$ is a convex $L$-smooth function on $\mathbb{R}^d$, then for all $\theta, \eta \in \mathbb{R}^d$, we have:*

$$\frac{1}{L}\|F'(\theta) - F'(\eta)\|_2^2 \leqslant \big[F'(\theta) - F'(\eta)\big]^\top (\theta - \eta).$$

*Moreover, we have: $F(\theta) \geqslant F(\eta) + F'(\eta)^\top(\theta - \eta) + \frac{1}{2L}\|F'(\theta) - F'(\eta)\|^2$.*

**Proof** We will show the second inequality, which implies the first one, by applying it twice with $\eta$ and $\theta$ swapped and summing them.

- Define $H(\theta) = F(\theta) - \theta^\top F'(\eta)$. The function $H : \mathbb{R}^d \to \mathbb{R}$ is convex with global minimum at $\eta$, since $H'(\theta) = F'(\theta) - F'(\eta)$, which is equal to zero for $\theta = \eta$. The function $H$ is also $L$-smooth.
- From the definition of smoothness, we get $H(\theta - \frac{1}{L}H'(\theta)) \leqslant H(\theta) + H'(\theta)^\top(-\frac{1}{L}H'(\theta)) + \frac{L}{2}\|-\frac{1}{L}H'(\theta)\|_2^2$, which is less than $H(\theta) - \frac{1}{2L}\|H'(\theta)\|_2^2$.
- This leads to $F(\eta) - \eta^\top F'(\eta) = H(\eta) \leqslant H(\theta - \frac{1}{L}H'(\theta)) \leqslant H(\theta) - \frac{1}{2L}\|H'(\theta)\|_2^2 = F(\theta) - \theta^\top F'(\eta) - \frac{1}{2L}\|F'(\theta) - F'(\eta)\|_2^2$, which leads to the desired inequality by shuffling terms.

■

We can now state the following convergence result for gradient descent with potentially no strong-convexity. Up to constants, we obtain the same rate for quadratic functions in Eq. (5.5).

**Proposition 5.5 (Convergence of GD for smooth convex functions)** *Assume that $F$ is $L$-smooth and convex, with a global minimizer $\eta_*$. Choosing $\gamma_t = 1/L$, the iterates $(\theta_t)_{t\geqslant 0}$ of GD on $F$ satisfy, for $t > 0$,*

$$F(\theta_t) - F(\eta_*) \leqslant \frac{L}{2t}\|\theta_0 - \eta_*\|_2^2.$$

**Proof** Following Bansal and Gupta (2019), the Lyapunov function that we will choose is

$$V_t(\theta_t) = t[F(\theta_t) - F(\eta_*)] + \frac{L}{2}\|\theta_t - \eta_*\|_2^2,$$

and our goal is to show that it decays along iterations (the requirement is thus weaker than for exponential convergence). We can split the difference in Lyapunov functions into three terms (each with its own color):

$$\begin{aligned} & V_t(\theta_t) - V_{t-1}(\theta_{t-1}) \\ = \ & t[F(\theta_t) - F(\theta_{t-1})] + F(\theta_{t-1}) - F(\eta_*) + \frac{L}{2}\|\theta_t - \eta_*\|_2^2 - \frac{L}{2}\|\theta_{t-1} - \eta_*\|_2^2. \end{aligned}$$

To bound it:

- We use $F(\theta_t)-F(\theta_{t-1}) \leqslant -\frac{1}{2L}\|F'(\theta_{t-1})\|_2^2$ like in the proof of Prop. 5.3.
- We use $F(\theta_{t-1})-F(\eta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1}-\eta_*)$, as a consequence of convexity (function above the tangent at $\theta_{t-1}$), as in Eq. (5.7).
- We use $\frac{L}{2}\|\theta_t-\eta_*\|_2^2 - \frac{L}{2}\|\theta_{t-1}-\eta_*\|_2^2 = -L\gamma(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) + \frac{L\gamma^2}{2}\|F'(\theta_{t-1})\|_2^2$ by expanding the square.

This leads to, with the step-size $\gamma = 1/L$:

$$\begin{aligned} V_t(\theta_t) - V_{t-1}(\theta_{t-1}) &\leqslant t\Big[-\frac{1}{2L}\|F'(\theta_{t-1})\|_2^2\Big] + F'(\theta_{t-1})^\top(\theta_{t-1}-\eta_*) \\ &\qquad\qquad -L\gamma(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) + \frac{L\gamma^2}{2}\|F'(\theta_{t-1})\|_2^2 \\ &= -\frac{t-1}{2L}\|F'(\theta_{t-1})\|_2^2 \ \leqslant 0, \end{aligned}$$

which leads to $t[F(\theta_t)-F(\eta_*)] \leqslant V_t(\theta_t) \leqslant V_0(\theta_0) = \frac{L}{2}\|\theta_0-\eta_*\|_2^2,$, and thus the desired bound $F(\theta_t)-F(\eta_*) \leqslant \frac{L}{2t}\|\theta_0-\eta_*\|_2^2$. ■

The proof above is on purpose mysterious: the choice of Lyapunov function seems arbitrary at first, but all inequalities lead to nice cancellations. These proofs are sometimes hard to design. For an interesting line of work trying to automate these proofs, see https://francisbach.com/computer-aided-analyses/.

**Exercise 5.12** *(alternative convergence proof ♦) We consider an $L$-smooth convex function with a global minimizer $\eta_*$, and gradient descent with step-size $\gamma_t = 1/L$.*

*(a) Show that $\|\theta_t-\eta_*\|_2^2 \leqslant \|\theta_{t-1}-\eta_*\|_2^2$ for all $t \geqslant 1$.*

*(b) Show that $F(\theta_t) \leqslant F(\theta_{t-1}) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2$ for all $t \geqslant 1$.*

*(c) Denoting $\Delta_t = F(\theta_t)-F(\eta_*)$, show that $\Delta_t \leqslant \Delta_{t-1} - \frac{1}{2L\|\theta_0-\eta_*\|^2}\Delta_{t-1}^2$ for all $t \geqslant 1$. Conclude that $\Delta_t \leqslant \frac{2L}{t+4}\|\theta_0-\eta_*\|^2$.*

**Early stopping for machine learning (♦♦).** An inspection of the proof of Prop. 5.5 shows that throughout the proof, a minimizer $\eta_*$ can be replaced by *any* $\eta \in \mathbb{R}^d$, leading to

$$t[F(\theta_t)-F(\eta)] + \frac{L}{2}\|\theta_t-\eta\|_2^2 \leqslant \frac{L}{2}\|\theta_0-\eta\|_2^2. \tag{5.11}$$

When $F(\theta) = \widehat{\mathcal{R}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, \theta^\top\varphi(x_i))$, for a smooth loss function (with constant $G_2$), for linear predictions with feature $\ell_2$-norms smaller than $R$, we have $L \leqslant G_2R^2$. Moreover, if the loss is non-negative, less than $G_0$ at zero predictions, and also Lipschitz-continuous (with constant $G_1$), such as the logistic loss, we showed in Section 4.5.4 that for any $D$,

$$\mathbb{E}\Big[\sup_{\|\theta\|_2 \leqslant D}\big|\mathcal{R}(\theta)-\widehat{\mathcal{R}}(\theta)\big|\Big] \leqslant \frac{4G_1RD}{\sqrt{n}}.$$

Then from Eq. (5.11), assuming we initialize with $\theta_0 = 0$, we get for $\eta = 0$, $\frac{G_2 R^2}{2}\|\theta_t\|_2^2 \leqslant tG_0$, leading to for any $\eta \in \mathbb{R}^d$:

$$\mathbb{E}\big[\mathcal{R}(\theta_t)\big] \leqslant \mathcal{R}(\eta) + \frac{4G_1 RD}{\sqrt{n}}\Big(\|\eta\|_2 + \Big(\frac{2G_0}{G_2 R^2}t\Big)^{1/2}\Big),$$

showing that if $t = o(n)$ (we do not make too many steps), the testing error is controlled. In particular, if $\theta_*$ is a minimizer of the expected risk $\mathcal{R}$, then wit $t = \sqrt{n}$, we obtain $\mathbb{E}\big[\mathcal{R}(\theta_t)\big] - \mathcal{R}(\theta_*) = O(n^{-1/4})$, which will be improved using a different analysis at the end of Section 5.3.

### 5.2.5 Beyond gradient descent (♦)

While gradient descent is the simplest algorithm with a simple analysis, there are multiple extensions that we will only briefly mention (see more details by Nesterov, 2004, 2007):

**Nesterov acceleration.** For strongly-convex functions, a simple modification of gradient descent allows for obtaining better convergence rates. The algorithm is as follows and is based on updating the following iterates:

$$\theta_t = \eta_{t-1} - \frac{1}{L}g'(\eta_{t-1}) \tag{5.12}$$

$$\eta_t = \theta_t + \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}(\theta_t - \theta_{t-1}), \tag{5.13}$$

and the convergence rate is then $F(\theta_t) - F(\eta_*) \leqslant L\|\theta_0 - \eta_*\|^2(1 - \sqrt{\mu/L})^t$, which is equal to $L\|\theta_0 - \eta_*\|^2(1 - 1/\sqrt{\kappa})^t$, that is the characteristic time to convergence goes from $\kappa$ to $\sqrt{\kappa}$. If $\kappa$ is large (typically of order $\sqrt{n}$ or $n$ for machine learning), the gains are substantial. In practice, this leads to significant improvements. See a detailed description and many extensions by d'Aspremont et al. (2021).

For convex functions, we need the extrapolation step to depend on $t$ as follows:

$$\theta_t = \eta_{t-1} - \frac{1}{L}F'(\eta_{t-1}) \tag{5.14}$$

$$\eta_t = \theta_t + \frac{t-1}{t+2}(\theta_t - \theta_{t-1}). \tag{5.15}$$

This simple modification dates back to Nesterov in 1983 and leads to the following convergence rate $F(\theta_t) - F(\eta_*) \leqslant \frac{2L\|\theta_0 - \eta_*\|^2}{(t+1)^2}$. See exercise below, and d'Aspremont et al. (2021) for more details.

Moreover, the last two rates are known to be optimal for the considered problems. For algorithms that access gradients and combine them linearly to select a new query point, it is impossible to have better dimension-independent rates. See Nesterov (2007) and Chapter 15 for more details.

**Exercise 5.13** *(♦♦) For the updates in Eq. (5.12) and Eq. (5.13), show that for* $L(\theta, \eta) = f(\theta) - f(\eta_*) + \frac{\mu}{2}\Big\|\eta - \eta_* + \frac{1+\sqrt{\mu/L}}{\sqrt{\mu/L}}(\theta - \eta)\Big\|_2^2$*, then* $L(\theta_t, \eta_t) \leqslant (1 - \sqrt{\mu/L})L(\theta_{t-1}, \eta_{t-1})$*. Show that this implies a convergence rate proportional to* $(1 - \sqrt{\mu/L})^t$*.*

**Exercise 5.14** *(♦♦) For the updates in Eq. (5.14) and Eq. (5.15), show that for* $L_t(\theta, \eta) = \big(\frac{t+1}{2}\big)^2\big[f(\theta) - f(\eta_*) + \frac{L}{2}\big\|\eta - \eta_* + \frac{t}{2}(\eta - \theta)\big\|_2^2\big]$*, then* $L_t(\theta_t, \eta_t) \leqslant L_{t-1}(\theta_{t-1}, \eta_{t-1})$*. Show that this implies a convergence rate proportional to* $\frac{1}{t^2}$*.*

**Newton method.** Given $\theta_{t-1}$, the Newton method minimizes the second-order Taylor expansion around $\theta_{t-1}$ (or, equivalently, finds a zero of $F'$ by using a first-order Taylor expansion of $F'$ around $\theta_{t-1}$):

$$F(\theta_{t-1}) + F'(\theta_{t-1})^\top(\theta - \theta_{t-1}) + \frac{1}{2}(\theta - \theta_{t-1})^\top F''(\theta_{t-1})^\top(\theta - \theta_{t-1}).$$

The gradient of this quadratic function is $\theta - \theta_{t-1} + F''(\theta_{t-1})^\top(\theta - \theta_{t-1})$, and setting it to zero leads to $\theta_t = \theta_{t-1} - F''(\theta_{t-1})^{-1}F'(\theta_{t-1})$, which is an expensive iteration, as the running-time complexity is $O(d^3)$ in general to solve the linear system. It leads to local quadratic convergence: If $\|\theta_{t-1} - \theta_*\|$ small enough, for some constant $C$, one can show $(C\|\theta_t - \theta_*\|) = (C\|\theta_{t-1} - \theta_*\|)^2$. See Boyd and Vandenberghe (2004) for more details and conditions for global convergence, in particular through the use of "self-concordance", which is a property that relates third and second-order derivatives.

⚠ The denomination "quadratic" is sometimes confusing and corresponds to a number of significant digits that doubles at each iteration.

Note that for machine learning problems, quadratic convergence may be overkill compared to the computational complexity of each iteration since cost functions are averages of $n$ terms and naturally have some uncertainty of order $O(1/\sqrt{n})$.

**Exercise 5.15** *(♦) Assume the function* $F$ *is* $\mu$*-strongly convex, twice differentiable, and such that the Hessian is Lipschitz-continuous, i.e.,* $\|f''(\theta) - f''(\eta)\|_{\rm op} \leqslant M\|\theta - \eta\|_2$*. Using the Taylor formula with integral remainder, show that for the iterates of Newton's method,* $\|\nabla F(\theta_t)\|_2 \leqslant \frac{M}{2\mu^2}\|\nabla F(\theta_{t-1})\|_2^2$*. Show that this implies local quadratic convergence.*

**Proximal gradient descent (♦).** Many optimization problems are said "composite", that is, the objective function $F$ is the sum of a smooth function $G$ and a non-smooth function $H$ (such as a norm). It turns out that a simple modification of gradient descent allows us to benefit from the fast convergence rates of smooth optimization (compared to the slower rates for non-smooth optimization that we would obtain from the subgradient method in the next section).

For this, we need to first see gradient descent as a *proximal method.* Indeed, one may see the iteration $\theta_t = \theta_{t-1} - \frac{1}{L}G'(\theta_{t-1})$, as

$$\theta_t = \arg\min_{\theta \in \mathbb{R}^d} \quad G(\theta_{t-1}) + (\theta - \theta_{t-1})^\top G'(\theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2,$$

where, for a $L$-smooth function $G$, the objective function above is an upper-bound of $G(\theta)$ which is tight at $\theta_{t-1}$ (see Eq. (5.10)).

While this reformulation does not bring much for gradient descent, we can extend this to the composite problem and consider the following iteration, where $H$ is left as is,

$$\theta_t = \arg\min_{\theta \in \mathbb{R}^d} \ G(\theta_{t-1}) + (\theta - \theta_{t-1})^\top G'(\theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2 + H(\theta).$$

It turns out that the convergence rates for $G + H$ are the same as smooth optimization, with potential acceleration (Nesterov, 2007; Beck and Teboulle, 2009).

The crux is to be able to compute the step above, that is, minimize with respect to $\theta$ functions of the form $\frac{L}{2}\|\theta - \eta\|_2^2 + H(\theta)$. When $H$ is the indicator function of a convex set (which is equal to 0 inside the set, and $+\infty$ otherwise), we get projected gradient descent. When $H$ is the $\ell_1$-norm, that is, $H = \lambda\|\cdot\|_1$, this can be shown to be a soft-thresholding step, as for each coordinate $\theta_i = (|\eta_i| - \lambda/L)_+ \frac{\eta_i}{|\eta_i|}$ (proof left as an exercise). See applications to model selection and sparsity-inducing norms in Chapter 8.

**Pre-conditioning (♦).** The convergence rate of gradient descent depends crucially on the condition number $\kappa$, which is not invariant by linear rescaling of the problem. That is, if we (equivalently) aim to minimize $G(\tilde{\theta}) = F(A\tilde{\theta})$ for some invertible matrix $A \in \mathbb{R}^{d\times d}$ and a twice differentiable function $F$, the gradient of $G$ is $G'(\tilde{\theta}) = A^\top F'(A\tilde{\theta})$, and thus gradient descent on $G$ can be written $\tilde{\theta}_t = \tilde{\theta}_{t-1} - \gamma G'(\tilde{\theta}) = \tilde{\theta}_{t-1} - \gamma A^\top F'(A\tilde{\theta}_{t-1})$, which can be rewritten as $\theta_t = \theta_{t-1} - \gamma A A^\top F'(\theta_{t-1})$ with the change of variable $\theta = A\tilde{\theta}$. This is thus equivalent to pre-multiplying the gradient of $F$ by the positive definite matrix $AA^\top$.

This will be advantageous when the condition number of $G$ is smaller than that of $F$. For example, for a quadratic function $F$ with constant Hessian matrix $H \in \mathbb{R}^{d\times d}$, taking $A$ as an inverse square root of $H$ leads to the minimal possible value of the condition number, and thus the pre-conditioned gradient iteration (here equal to the Newton step) converges in one iteration. Such a value of $A$ optimizes the condition number but is not computationally efficient, and various conditioners can be used in practice, based on diagonal approximations of the Hessian, random projections (Martinsson and Tropp, 2020) or incomplete Cholesky factorizations (Golub and Loan, 1996). Such preconditioning is also useful in non-smooth situations (see Section 5.4.2 in the context of SGD).

### 5.2.6 Non-convex objective functions (♦)

For smooth, potentially non-convex objective functions, the best one can hope for is to converge to a stationary point $\theta$ such that $F'(\theta) = 0$. The proof below provides the weaker result that at least one iterate has a small gradient. Indeed, using the same Taylor expansion as the convex case (which is still valid), we get, using the $L$-smoothness of $F$:

$$F(\theta_t) \ \leqslant \ F(\theta_{t-1}) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2,$$

leading to, summing the inequalities above for all iterations between 1 and $t$:

$$\frac{1}{2Lt}\sum_{s=1}^{t}\|F'(\theta_{s-1})\|_2^2 \leqslant \frac{F(\theta_0)-F(\eta_*)}{t}.$$

Thus, there is one $s$ in $\{0,\dots,t-1\}$ for which $\|F'(\theta_s)\|_2^2 \leqslant O(1/t)$. Without further assumptions, this does not imply that any of the iterates is close to a stationary point.

## 5.3 Gradient methods on non-smooth problems

We now relax our assumptions and only require Lipschitz continuity in addition to convexity. The rates will be slower, but extending stochastic gradients will be easier.

**Definition 5.4 (Lipschitz-continuous function)** *A function $F : \mathbb{R}^d \to \mathbb{R}$ is said $B$-Lipschitz-continuous if and only if*

$$|F(\eta)-F(\theta)| \leqslant B\|\eta-\theta\|_2, \qquad \forall \theta,\eta \in \mathbb{R}^d.$$

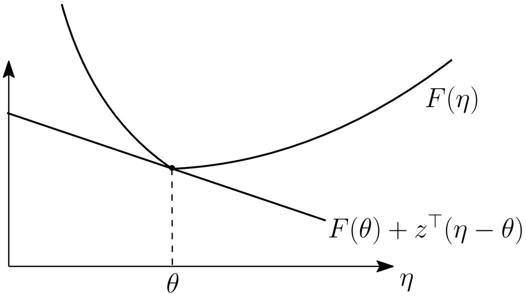
This setting is usually referred to as *non-smooth* optimization.

**Exercise 5.16** *Show that if $F$ is differentiable, $B$-Lipschitz-continuity is equivalent to the assumption $\|F'(\theta)\|_2 \leqslant B$, $\forall \theta \in \mathbb{R}^d$.*

**From gradients to subgradients.** We can apply non-smooth optimization to objective functions that are not differentiable (such as the hinge loss from Section 4.1.2). For convex Lipschitz-continuous objectives, one can show that the function is almost everywhere differentiable (see, e.g., Nekvinda and Zajíček, 1988). In points where it is not, one can define the set of slopes of lower-bounding tangents as the *subdifferential* and any element of it as a *subgradient*. That is, we can define the subdifferential as (see illustration below):

$$\partial F(\theta) = \big\{z \in \mathbb{R}^d,\ \forall \eta \in \mathbb{R}^d,\ F(\eta) \geqslant F(\theta) + z^\top(\eta-\theta)\big\}.$$

For a convex function defined on $\mathbb{R}^d$, the subdifferential happens to be a non-empty convex set at all points $\theta$. Moreover, when $F$ is differentiable with gradient $F'(\theta)$, the subdifferential is reduced to a point, that is, $\partial F(\theta) = \{F'(\theta)\}$. For example, the absolute value $\theta \mapsto |\theta|$ has a subdifferential equal to $[-1,1]$ at zero. See more details by Rockafellar (1997).



The gradient descent iteration is then meant as using any subgradient $z \in \partial F(\theta_{t-1})$ instead of $F'(\theta_{t-1})$, for which we will only need that the function is above the tangent defined by this subgradient. The method is then referred to as the subgradient method (it is not a descent method anymore, that is, the function values may go up occasionally).

**Exercise 5.17** *Compute the subdifferential of $\theta \mapsto |\theta|$ and $\theta \mapsto (1 - y\theta^\top x)_+$, for the label $y \in \{-1, 1\}$ and the input $x \in \mathbb{R}^d$.*

**Convergence rate of the subgradient method.** We can prove convergence of the gradient descent algorithm, now with a decaying step-size and a slower rate than for smooth functions.

⚠ Like for stochastic gradient descent in the next section, and as opposed to gradient descent for smooth functions in the previous section, the objective function for the subgradient method for non-smooth functions may not go down at every iteration.

**Proposition 5.6 (Convergence of the subgradient method)** *Assume that $F$ is convex, $B$-Lipschitz-continuous, and admits a minimizer $\eta_*$ that satisfies $\|\eta_* - \theta_0\|_2 \leqslant D$. By choosing $\gamma_t = \frac{D}{B\sqrt{t}}$ then the iterates $(\theta_t)_{t \geqslant 0}$ of GD on $F$ satisfy*

$$\min_{0 \leqslant s \leqslant t-1} F(\theta_s) - F(\eta_*) \leqslant DB \frac{2 + \log(t)}{2\sqrt{t}}. \tag{5.16}$$

**Proof** We look at how $\theta_t$ approaches $\eta_*$, that is, we try to use $\|\theta_t - \eta_*\|_2^2$ as a Lyapunov function. We have:

$$\begin{aligned} \|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \gamma_t F'(\theta_{t-1}) - \eta_*\|_2^2 \\ &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma_t F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*) + \gamma_t^2 \|F'(\theta_{t-1})\|_2^2. \end{aligned}$$

Combining this with the convexity inequality $F(\theta_{t-1}) - F(\eta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*)$ from Eq. (5.7), using the boundedness of the gradients (that is, $\|F'(\theta_{t-1})\|_2^2 \leqslant B^2$), it follows:

$$\|\theta_t - \eta_*\|_2^2 \leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma_t[F(\theta_{t-1}) - F(\eta_*)] + \gamma_t^2 B^2.$$

We are in a situation where the Lyapunov function $\theta \mapsto \|\theta - \eta_*\|_2^2$ is not decreasing along iterations because of the term $\gamma_t^2 B^2$ above. It is then classical to isolate the negative term $-2\gamma_t[F(\theta_{t-1}) - F(\eta_*)]$ and sum inequalities. Thus, by isolating the distance to optimum in function values, we get:

$$\gamma_t(F(\theta_{t-1}) - F(\eta_*)) \leqslant \frac{1}{2}\Big(\|\theta_{t-1} - \eta_*\|_2^2 - \|\theta_t - \eta_*\|_2^2\Big) + \frac{1}{2}\gamma_t^2 B^2. \tag{5.17}$$

It is sufficient to sum these inequalities to get (in fact, for any $\eta_* \in \mathbb{R}^d$ and not only the minimizer),

$$\frac{1}{\sum_{s=1}^t \gamma_s} \sum_{s=1}^t \gamma_s \left(F(\theta_{s-1}) - F(\eta_*)\right) \leqslant \frac{\|\theta_0 - \eta_*\|_2^2}{2\sum_{s=1}^t \gamma_s} + B^2 \frac{\sum_{s=1}^t \gamma_s^2}{2\sum_{s=1}^t \gamma_s}.$$

As a weighted average, the left-hand side is larger than $\min_{0\leqslant s\leqslant t-1}(F(\theta_s) - F(\eta_*))$, and also larger than $F(\bar{\theta}_t) - F(\eta_*)$ where $\bar{\theta}_t = (\sum_{s=1}^t \gamma_s \theta_{s-1})/(\sum_{s=1}^t \gamma_s)$ by Jensen's inequality.

The upper bound goes to 0 if $\sum_{s=1}^t \gamma_s$ goes to $\infty$ (to forget the initial condition, sometimes called the "bias") and $\gamma_t \to 0$ (to decrease the "variance" term). Let us choose $\gamma_s = \tau/\sqrt{s}$ for some $\tau > 0$. By using the series-integral comparisons below, we get the bound

$$\min_{0\leqslant s\leqslant t-1}(F(\theta_s) - F(\eta_*)) \leqslant \frac{1}{2\sqrt{t}}\Big(\frac{D^2}{\tau} + \tau B^2(1+\log(t))\Big).$$

We choose $\tau = D/B$ (which is suggested by optimizing the previous bound when $\log(t) = 0$), which leads to the result. In the proof, we used the following series-integral comparisons for decreasing functions:

$$\sum_{s=1}^t \frac{1}{\sqrt{s}} \geqslant \sum_{s=1}^t \frac{1}{\sqrt{t}} = \sqrt{t},$$

and $\sum_{s=1}^t \frac{1}{s} \leqslant 1 + \sum_{s=2}^t \frac{1}{s} \leqslant 1 + \int_1^t \frac{ds}{s} = 1 + \log(t)$. ■

The proof scheme above is very flexible. It can be extended in the following directions:

- There is no need to know in advance an upper-bound $D$ on the distance to optimum; we then get, with an arbitrary $D$, with the same step-size $\gamma_t = \frac{D}{B\sqrt{t}}$ a rate of the form $\frac{BD}{2\sqrt{t}}\big(\frac{\|\theta_0-\eta_*\|_2^2}{D^2} + (1+\log(t))\big)$. Moreover, a slightly modified version of the subgradient method removes the need to know the Lipschitz constant. See the exercise below.

  **Exercise 5.18** *We consider the iteration $\theta_t = \theta_{t-1} - \frac{\gamma_t'}{\|F'(\theta_{t-1})\|_2}F'(\theta_{t-1})$. Show that with the step-size $\gamma_t' = D/\sqrt{t}$, we get the guarantee $\min_{0\leqslant s\leqslant t-1} F(\theta_s) - F(\eta_*) \leqslant DB\frac{2+\log(t)}{2\sqrt{t}}$.*

- The algorithm applies to constrained minimization over a convex set by inserting a projection step at each iteration (the proof, which uses the contractivity of orthogonal projections, is essentially the same; see the exercise below).

  **Exercise 5.19** *Let $K \subset \mathbb{R}^d$ be a convex closed set, and denote by $\Pi_K(\theta)$ he orthogonal projection of $\theta$ onto $K$, defined as $\Pi_K(\theta) = \arg\min_{\eta\in K} \|\eta - \theta\|_2^2$.Show that the function $\Pi_K$ is contractive, that is, for all $\theta, \eta \in \mathbb{R}^d$, $\|\Pi_K(\theta) - \Pi_K(\eta)\|_2 \leqslant \|\theta - \eta\|_2$. For the algorithm $\theta_t = \Pi_K(\theta_{t-1} - \gamma_t F'(\theta_{t-1}))$, and $\eta_*$ a minimizer of $F$ on $K$, show that the guarantee of Prop. 5.6 still holds.*

- The algorithm applies to non-differentiable convex and Lipschitz objective functions (using sub-gradients, i.e., any vector satisfying Eq. (5.6) in place of $F'(\theta_t)$).

- The algorithm can be applied to "non-Euclidean geometries", where we consider bounds on the iterates or the gradient with different quantities, such as Bregman divergences. This can be done using the "mirror descent" framework, and for instance, can be applied to obtain multiplicative updates (see, e.g., Juditsky and Nemirovski, 2011a,b; Bubeck, 2015). See more details in the online stochastic case in Section 11.1.3.

**Exercise 5.20** *(♦) Let $F : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function, and $\psi : \mathbb{R}^d \to \mathbb{R}$ a strictly convex function:*

*(a) Show that the minimizer of $F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{1}{2\gamma}\|\eta - \theta\|_2^2$ is equal to $\eta = \theta - \gamma F'(\theta)$.*

*(b) Show that the Bregman divergence $D_\psi(\eta, \theta)$ defined as $D_\psi(\eta, \theta) = \psi(\eta) - \psi(\theta) - \psi'(\theta)^\top(\eta - \theta)$ is non-negative, and equal to zero if and only if $\eta = \theta$.*

*(c) Show that the minimizer of $F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{1}{\gamma} D_\psi(\eta, \theta)$ satisfies $\psi'(\eta) = \psi'(\theta) - \gamma F'(\theta)$. Show that the same conclusion holds if $\psi$ is only defined on an open convex set $K \subset \mathbb{R}^d$, and so that the gradient $\psi'$ is a bijection from $K$ to $\mathbb{R}^d$.*

*(d) Apply to $\psi(\theta) = \sum_{i=1}^d \theta_i \log \theta_i$.*

- Often the uniformly averaged iterate is used, as $\frac{1}{t}\sum_{s=0}^{t-1} \theta_s$. Convergence rates (without the $\log t$ factor) can be obtained with a slightly more involved proof using the Abel summation formula (see also Section 11.1.1).

**Exercise 5.21** *(♦) We consider the same assumptions as Exercise 5.19 and the same algorithm with orthogonal projections. With $D$ the diameter of $K$, show that for the average iterate $\bar{\theta}_t = \frac{1}{t}\sum_{s=0}^{t-1} \theta_s$, we have: $F(\bar{\theta}_t) - F(\theta_*) \leqslant \frac{3BD}{2\sqrt{t}}$.*

- The algorithm with the decaying step-size $\gamma_t$ is an "anytime" algorithm; that is, it can be stopped at any time $t$, and the bound in Eq. (5.16) then applies. Computations are often easier when considering a constant step-size $\gamma$ that depends on the number of iterations $T$ that the user wishes to perform, $T$ being usually referred to as the "horizon". Starting from Eq. (5.17), we get the bound:

$$\frac{1}{T}\sum_{t=1}^{T} F(\theta_{t-1}) - F(\theta_*) \leqslant \frac{D^2}{2\gamma T} + \frac{\gamma B^2}{2}, \tag{5.18}$$

where the optimal $\gamma$ can be obtain as $\gamma = \frac{D}{B\sqrt{T}}$ and an optimized rate of $\frac{DB}{\sqrt{T}}$. We gain on the logarithmic factor, but we no longer have an anytime algorithm (since the bound only applies at time $T$). This applies as well to SGD in Section 5.4. In these situations, a "doubling trick" can be used, leading to an anytime algorithm with the same guarantee but undesired practical behavior as the algorithm makes substantial changes at each iteration, which is a power of two (see the exercise below).

- Stochastic gradients can be used, as presented below (one interpretation is that the subgradient method is so slow that it is robust to noisy gradients).

**Exercise 5.22 (doubling trick for subgradient method)** *We consider an algorithm that successively applies the SGD iteration with step-size $\gamma = \frac{D}{B\sqrt{2^k}}$ during $2^k$ iterations, for $k = 0, 1, \dots$. Show that after $t$ subgradient iterations, the observed best value of $F$ is less than a constant times $DB/\sqrt{t}$.*

**Exercise 5.23** *Compute all constants for $\ell_2$-regularized logistic regression and the support vector machine with linear predictors (Section 4.1).*

**Machine learning with linear predictions and Lipschitz-continuous losses.** For specialized machine learning problems, we can now close the loop on the discussion outlined in Section 5.1 regarding the need to take into account the optimization error on top of the estimation error. For convex Lipschitz-continuous losses (with constant $G$) such as the logistic loss or the hinge loss, for linear predictions with feature $\ell_2$-norms smaller than $R$, a parameter bounded in $\ell_2$-norm by $D$, we showed in Section 4.5.4 that the estimation error was upper-bounded by a constant times $\frac{GRD}{\sqrt{n}}$. The optimization error after $t$ iterations of the subgradient method is upper-bounded by a constant times $\frac{GRD}{\sqrt{t}}$, since the Lipschitz constant of the objective function is $B \leqslant GR$.

Adding these two bounds, there is no need to have the number of iterations $t$ larger than the number of observations $n$. However, since each gradient computation requires $n$ gradient computations for the individual loss functions associated with a single data point, the total number of such gradient computations is $tn \approx n^2$, which is not scalable when $n$ is large. We now show how stochastic gradient descent can turn this number to $n$ with the same upper-bound on the generalization performance.

## 5.4 Convergence rate of stochastic gradient descent (SGD)

For machine learning problems, where $F(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f_\theta(x_i)) + \Omega(\theta)$, at each iteration, the gradient descent algorithm requires computing a "full" gradient $F'(\theta_{t-1})$, which could be costly as it requires accessing the entire data set (all $n$ pairs of observations). An alternative is to instead only compute *unbiased* stochastic estimations of the gradient $g_t(\theta_{t-1})$, i.e., such that

$$\mathbb{E}[g_t(\theta_{t-1})|\theta_{t-1}] = F'(\theta_{t-1}), \tag{5.19}$$

which could be much faster to compute, in particular by accessing fewer observations.

⚠ Note that we need to condition over $\theta_{t-1}$ because $\theta_{t-1}$ encapsulates all the randomness due to past iterations, and we only require "fresh" randomness at time $t$.

⚠ Somewhat surprisingly, this unbiasedness does *not* need to be coupled with a vanishing variance: while there are always errors in the gradient, the use of a decreasing step-size will ensure convergence. If the noise in the gradient is not unbiased, then we only get convergence if the noise magnitudes go to zero (see, e.g., d'Aspremont, 2008; Schmidt et al., 2011 and references therein).

This leads to the following algorithm.

**Algorithm 5.2 (Stochastic gradient descent (SGD))** *Choose a step-size sequence* $(\gamma_t)_{t \geqslant 0}$*, pick* $\theta_0 \in \mathbb{R}^d$ *and for* $t \geqslant 1$*, let*

$$\theta_t = \theta_{t-1} - \gamma_t g_t(\theta_{t-1}),$$

*where* $g_t(\theta_{t-1})$ *satisfies Eq. (5.19).*

**SGD in machine learning.** There are two ways to use SGD for supervised machine learning:

(1) **Empirical risk minimization**: If $F(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f_\theta(x_i))$ then at iteration $t$ we can choose uniformly at random $i(t) \in \{1, \dots, n\}$ and define $g_t$ as the gradient of $\theta \mapsto \ell(y_{i(t)}, f_\theta(x_{i(t)}))$. Here, the randomness comes from the random choice of indices.

There exist "mini-batch" variants where, at each iteration, the gradient is averaged over a random subset of the indices (we then reduce the variance of the gradient estimate, but we use more gradients, and thus the running time increases, see Exercise 5.25). We then converge to a *minimizer* $\eta_*$ *of the empirical risk*.

Note here that since we sample *with replacement*, a given function will be selected several times, even within $n$ iterations. Sampling without replacement can also be studied, but its analysis is more involved (see, e.g., Nagaraj et al., 2019, and references therein).

(2) **Expected risk minimization**: If $F(\theta) = \mathbb{E}[\ell(y, f_\theta(x))]$ is the expected (non-observable) risk, then at iteration $t$ we can take a fresh sample $(x_t, y_t)$ and define $g_t$ as the gradient of $\theta \mapsto \ell(y_t, f_\theta(x_t))$, for which, if we swap the orders of expectation and differentiation, we get the unbiasedness. Note here that to preserve the unbiasedness, only a single pass is allowed (otherwise, this would create dependencies that would break it) and that the randomness comes from the observations $(x_t, y_t)$ themselves.

Here, we *directly minimize the (generalization) risk*. The counterpart is that if we only have $n$ samples, then we can only run $n$ SGD iterations, and when $n$ grows, the iterates will converge to a *minimizer* $\theta_*$ *of the expected risk*.

Note that in practice, multiple passes over the data (that is, using each observation multiple times) lead to better performance. To avoid overfitting, either a regularization term is added to the empirical risk, or the SGD algorithm is stopped before its convergence (and typically when some validation risk stops decreasing), which is referred to as regularization by "early stopping".

We can study the two situations above using the latter one by considering the empirical risk as the expectation with respect to the empirical distribution of the data (and we thus use the notation $\theta_*$ for the global minimizer).

Stochastic gradient descent is not a descent method: the function values often go up, but they go down "on average". See, for example, an illustration in Figure 5.2.

Under the same usual assumptions on the objective functions, we now study SGD with the following extra assumptions:

- (H-1) unbiased gradient: $\mathbb{E}[g_t(\theta_{t-1})|\theta_{t-1}] = F'(\theta_{t-1})$, $\forall t \geqslant 1$,
- (H-2) bounded gradient: $\|g_t(\theta_{t-1})\|_2^2 \leqslant B^2$ almost surely, $\forall t \geqslant 1$.

Assumption (H-2) could be replaced by other regularity conditions (e.g., Lipschitz-continuous gradients). Assumption (H-1) is crucial and is often obtained by considering independent gradient functions $g_t$, for which we have $\mathbb{E}[g_t(\cdot)] = F'(\cdot)$. See Exercise 5.26 for SGD for smooth functions.

**Proposition 5.7 (Convergence of SGD)** *Assume that $F$ is convex, $B$-Lipschitz and admits a minimizer $\theta_*$ that satisfies $\|\theta_* - \theta_0\|_2 \leqslant D$. Assume that the stochastic gradients satisfy (H-1) and (H-2). Then, choosing $\gamma_t = (D/B)/\sqrt{t}$, the iterates $(\theta_t)_{t\geqslant 0}$ of SGD on $F$ satisfy*

$$\mathbb{E}\big[F(\bar{\theta}_t) - F(\theta_*)\big] \leqslant DB\frac{2 + \log(t)}{2\sqrt{t}}.$$

*where $\bar{\theta}_t = (\sum_{s=1}^t \gamma_s \theta_{s-1})/(\sum_{s=1}^t \gamma_s)$.*

We state our bound in terms of the average iterates because the cost of finding the best iterate could be higher than that of evaluating a stochastic gradient (since we cannot compute $F$ in general).

**Proof** We follow essentially the same proof as in the deterministic case, adding some expectations at well-chosen places. We have:

$$\begin{aligned}\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big] &= \mathbb{E}\big[\|\theta_{t-1} - \gamma_t g_t(\theta_{t-1}) - \theta_*\|_2^2\big] \\ &= \mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - 2\gamma_t \mathbb{E}\big[g_t(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big] + \gamma_t^2 \mathbb{E}\big[\|g_t(\theta_{t-1})\|_2^2\big].\end{aligned}$$

We can then compute the expectation of the middle term as:

$$\begin{aligned}\mathbb{E}\big[g_t(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big] &= \mathbb{E}\big[\mathbb{E}\big[g_t(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big|\theta_{t-1}\big]\big] \\ &= \mathbb{E}\big[\mathbb{E}\big[g_t(\theta_{t-1})\big|\theta_{t-1}\big]^\top(\theta_{t-1} - \theta_*)\big] = \mathbb{E}\big[F'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big],\end{aligned}$$

where we have crucially used the unbiasedness assumption (H-1). This leads to

$$\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big] \leqslant \mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - 2\gamma_t \mathbb{E}\big[F'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big] + \gamma_t^2 B^2.$$

Thus, combining with the convexity inequality $F(\theta_{t-1}) - F(\theta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)$ from Eq. (5.7), we get

$$\gamma_t \mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] \leqslant \frac{1}{2}\Big(\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\Big) + \frac{1}{2}\gamma_t^2 B^2. \qquad (5.20)$$

Except for the expectations, this is the same bound as Eq. (5.17), so we can conclude as in the proof of Prop. 5.6. ■

We can make the following observations:

- Averaging of iterates is often performed after a certain number of iterations (e.g., one pass over the data when doing multiple passes): having such a "burn-in" period speeds up the algorithms by forgetting initial conditions faster.
- Many authors consider the projected version of the algorithm where after the gradient step, we orthogonally project onto the ball of radius $D$ and center $\theta_0$. The bound is then exactly the same.
- Like for the subgradient method in Eq. (5.18), we can consider a constant step-size $\gamma$, to obtain

  $$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[F(\theta_{t-1})\big] - F(\theta_*) \leqslant \frac{D^2}{2\gamma T} + \frac{\gamma B^2}{2},$$

  from which we get, $\mathbb{E}\big[F(\bar{\theta}_t)\big] - F(\theta_*) \leqslant \frac{D^2}{2\gamma T} + \frac{\gamma B^2}{2} = \frac{DB}{\sqrt{T}}$, for the specific choice $\gamma = D/(B\sqrt{T})$ (that depends on the horizon $T$, and for the uniformly average iterate.
- The result that we obtain, when applied to single pass SGD, is a generalization bound; that is, after the $n$ iterations, we have an excess risk proportional to $1/\sqrt{n}$, corresponding to the excess risk compared to the best predictor $f_\theta$.

  This is to be compared to using results from Chapter 4 (uniform deviation bounds) and non-stochastic gradient descent. It turns out that the estimation error due to having $n$ observations is exactly the same as the generalization bound obtained by SGD (see Section 4.5.4 in Chapter 4). Still, we need to add on top of the optimization error proportional to $1/\sqrt{t}$ (with the same constants). The bounds match if $t = n$, that is, we run $n$ iterations of gradient descent on the empirical risk. This leads to a running time complexity of $O(tnd) = O(n^2 d)$ instead of $O(nd)$ using SGD, hence the strong gains in using SGD.

  ⚠ We are still comparing upper bounds.
- The bound in $O(BD/\sqrt{t})$ is optimal for this class of problem. That is, as shown for example by Agarwal et al. (2009), among all algorithms that can query stochastic gradients, having a better convergence rate (up to some constants) is impossible. See Section 15.3 for a detailed proof.
- As opposed to the deterministic case, the use of smoothness does not lead to significantly better results (see Exercise 5.26).
- An inspection of the proof shows that we can replace the almost sure bounds $\|g_t(\theta_{t-1})\|_2^2 \leqslant B^2$ by bounds in expectation $\mathbb{E}\big[\|g_t(\theta_{t-1})\|_2^2\big] \leqslant B^2$. For machine learning problems with linear predictions with feature $\ell_2$-norm bounded by $R$ and a $G$-Lipschitz-continuous loss, the gradient $g_t(\theta_{t-1})$ is the gradient of the function $\theta \mapsto \ell(y_t, \varphi(x_t)^\top\theta)$ taken at $\theta_{t-1}$, and thus its squared norm is less than

$G^2 \cdot \|\varphi(x_t)\|_2^2$. An almost sure bound is therefore $G^2R^2$, while a bound in expectation is $G^2 \cdot \mathbb{E}\big[\|\varphi(x_t)\|_2^2\big]$, which is stronger.

- We can obtain a result in high probability, using an extension of Hoeffding's inequality to "differences of martingales", as shown in the exercise below.

**Exercise 5.24 (high-probability bound for SGD (♦))** *Using the same assumptions and notations as in Prop. 5.7, we consider the projected iteration:* $\theta_t = \Pi_D(\theta_{t-1} - \gamma_t g_t)$, *where* $\Pi_D$ *is the orthogonal projection on the* $\ell_2$*-ball of center* 0 *and radius* $D$. *Denoting* $z_t = -\gamma_t(\theta_{t-1} - \theta_*)^\top [g_t - F'(\theta_{t-1})]$, *show that* $\mathbb{E}[z_t|\mathcal{F}_{t-1}] = 0$ *and* $|z_t| \leqslant 4\gamma_t BD$ *almost surely, and that*

$$\gamma_t[F(\theta_{t-1}) - F(\theta_*)] \leqslant \frac{1}{2}\Big(\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\Big) + \frac{1}{2}\gamma_t^2 B^2 + z_t.$$

*Show that with probability at least* $1 - \delta$, *then, for the weighted average* $\bar{\theta}_t$ *defined in Prop. 5.7, for any step-sizes* $\gamma_t$:

$$F(\bar{\theta}_t) \leqslant \frac{2D^2}{\sum_{s=1}^t \gamma_s} + B^2 \frac{\sum_{s=1}^t \gamma_s^2}{2\sum_{s=1}^t \gamma_s} + 4BD \frac{\big(\sum_{s=1}^t \gamma_s^2\big)^{1/2}}{\sum_{s=1}^t \gamma_s} \sqrt{2 \log \frac{1}{\delta}},$$

*and that for a constant-step size,* $\gamma_t = \gamma$, $F(\bar{\theta}_t) \leqslant \frac{2D^2}{\gamma T} + \frac{\gamma B^2}{2} + \frac{4DB}{\sqrt{t}}\sqrt{2 \log \frac{1}{\delta}}$.

**SGD or gradient descent on the empirical risk?** As seen above, the number of iterations to reach a given precision will be larger for stochastic gradient descent than for smooth deterministic gradient descent, but with a complexity that is typically $n$ times faster. Thus, for high precision, that is, low values of $F(\theta) - F(\eta_*)$ (which is not needed for machine learning), the number of iterations of SGD may become prohibitively large, and deterministic full gradient descent could be preferred. However, for low precision and large $n$, SGD is the method of choice (see also recent improvements in Section 5.4.4).

In particular, for the linear prediction case described at the end of Section 5.3, we obtain the exact same rate in Prop. 5.7 as for non-stochastic gradient descent on the empirical risk. If sampling from the $n$ observations with replacement, after $t = n$ steps, the sum of optimization error and optimization error is of the same order $O(\frac{GRD}{\sqrt{n}})$, with now only $n$ accesses to individual loss gradients (instead of $n^2$ with batch methods, thus, with a big improvement). Moreover, with *a single pass over the data*, Prop. 5.7 is *directly* a generalization performance result with the same rate.

**Exercise 5.25** *We consider the mini-batch version of SGD, where at every iteration, we replace* $g_t(\theta_{t-1})$ *by the average of* $m$ *independent samples of stochastic gradients at* $\theta_{t-1}$. *Extend the convergence result of Prop. 5.7.*

**Exercise 5.26** *(♦) We consider independent and identically distributed convex* $L$*-smooth random functions* $f_t : \mathbb{R}^d \to \mathbb{R}$, $t \geqslant 1$, *with expectation* $F : \mathbb{R}^d \to \mathbb{R}$, *that has a minimizer* $\theta_* \in \mathbb{R}^d$. *We consider the SGD recursion* $\theta_t = \theta_{t-1} - \gamma_t f_t'(\theta_{t-1})$, *with* $\gamma_t$ *a deterministic* *step-size sequence. Using co-coercivity (Prop. 5.4), show that*

$$\mathbb{E}\big[\|\theta_t-\theta_*\|_2^2\big] \leqslant \mathbb{E}\big[\|\theta_{t-1}-\theta_*\|_2^2\big]-2\gamma_t(1-\gamma_t L)\mathbb{E}\big[F'(\theta_{t-1})^\top(\theta_{t-1}-\theta_*)\big]+2\gamma_t^2\mathbb{E}\big[\|f_t'(\theta_*)\|_2^2\big].$$

*Extend the proof of Prop. 5.7 to obtain an explicit rate in $O(1/\sqrt{t})$.*

**Exercise 5.27 (non-uniform sampling (♦))** *We consider functions $F:\mathbb{R}^d\times\mathcal{Z}\to\mathbb{R}$, which are convex with respect to the first variable, with a subgradient $F'(\theta,z)$ with respect to the first variable which is bounded in $\ell_2$-norm by a constant $B(z)$ that depends on $z$. We consider a distribution $p$ on $\mathcal{Z}$, and we aim to minimize $\mathbb{E}_{z\sim p}F(\theta,z)$, but we sample from a distribution $q$, with density $dq/dp(z)$ with respect to $p$ to get independent and identically distributed random $z_t$, $t\geqslant 1$. We consider the recursion $\theta_t=\theta_{t-1}-\frac{\gamma}{dq/dp(z_t)}F'(\theta_{t-1},z_t)$. Provide a convergence rate for this algorithm and show how a good choice of $q$ leads to significant improvements over the choice $q=p$ when $B(z)$ is far from uniform in $z$. Apply this result to the support vector machine when applying SGD to the empirical risk.*

### 5.4.1 Strongly convex problems (♦)

We consider the regularized problem $G(\theta)=F(\theta)+\frac{\mu}{2}\|\theta\|_2^2$, with the same assumption as above, and started at $\theta_0=0$. The SGD iteration is then, with $g_t(\theta_{t-1})$ a stochastic (sub)gradient of $F$ at $\theta_{t-1}$:

$$\theta_t=\theta_{t-1}-\gamma_t\big[g_t(\theta_{t-1})+\mu\theta_{t-1}\big]. \tag{5.21}$$

We then have an improved convergence rate in $O(1/t)$ with a different decay for the step-size.

**Proposition 5.8 (Convergence of SGD for strongly-convex problems)** *Assume that $F$ is convex, $B$-Lipschitz and that $F+\frac{\mu}{2}\|\cdot\|_2^2$ admits a (necessarily unique) minimizer $\theta_*$. Assume that the stochastic gradient $g$ satisfies (H-1) and (H-2). Then, choosing $\gamma_t=1/(\mu t)$, the iterates $(\theta_t)_{t\geqslant 0}$ of SGD from Eq. (5.21) satisfy*

$$\mathbb{E}\big[G(\bar\theta_t)-G(\theta_*)\big]\leqslant\frac{2B^2(1+\log t)}{\mu t},$$

*where $\bar\theta_t=\frac{1}{t}\sum_{s=1}^t\theta_{s-1}$.*

**Proof** The beginning of the proof is essentially the same as for convex problems, leading to (with the new terms in blue):

$$\begin{aligned}\mathbb{E}\big[\|\theta_t-\theta_*\|_2^2\big] &= \mathbb{E}\big[\|\theta_{t-1}-\gamma_t(g_t(\theta_{t-1})+\mu\theta_{t-1})-\theta_*\|_2^2\big]\\ &= \mathbb{E}\big[\|\theta_{t-1}-\theta_*\|_2^2\big]-2\gamma_t\mathbb{E}\big[(g_t(\theta_{t-1})+\mu\theta_{t-1})^\top(\theta_{t-1}-\theta_*)\big]\\ &\qquad\qquad+\gamma_t^2\mathbb{E}\big[\|g_t(\theta_{t-1})+\mu\theta_{t-1}\|_2^2\big].\end{aligned}$$

From the iterations in Eq. (5.21), we see that $\theta_t=(1-\gamma_t\mu)\theta_{t-1}+\gamma_t\mu\big[-\frac{1}{\mu}g_t(\theta_{t-1})\big]$ is a convex combination of gradients divided by $-\mu$, and thus $\|g_t(\theta_{t-1})+\mu\theta_{t-1}\|_2^2$ is always less than $4B^2$. Thus

$$\mathbb{E}\big[\|\theta_t-\theta_*\|_2^2\big]\leqslant\mathbb{E}\big[\|\theta_{t-1}-\theta_*\|_2^2\big]-2\gamma_t\mathbb{E}\big[G'(\theta_{t-1})^\top(\theta_{t-1}-\theta_*)\big]+4\gamma_t^2B^2.$$

Therefore, combining with the inequality coming from strong convexity $G(\theta_{t-1}) - G(\theta_*) + \frac{\mu}{2}\|\theta_{t-1} - \theta_*\|_2^2 \leqslant G'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)$ (see Eq. (5.9)), it follows

$$\gamma_t \mathbb{E}[G(\theta_{t-1}) - G(\theta_*)] \leqslant \frac{1}{2}\big((1-\gamma_t\mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \mathbb{E}\|\theta_t - \theta_*\|^2\big) + 2\gamma_t^2 B^2,$$

and thus, now using the specific step-size choice $\gamma_t = 1/(\mu t)$:

$$\begin{aligned}\mathbb{E}[G(\theta_{t-1}) - G(\theta_*)] &\leqslant \frac{1}{2}\big((\gamma_t^{-1} - \mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \gamma_t^{-1}\mathbb{E}\|\theta_t - \theta_*\|^2\big) + 2\gamma_t B^2, \\ &= \frac{1}{2}\big(\mu(t-1)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \mu t\mathbb{E}\|\theta_t - \theta_*\|^2\big) + \frac{2B^2}{\mu t}.\end{aligned}$$

Thus, we get a telescoping sum: summing between all indices between 1 and $t$, and using the bound $\sum_{s=1}^t \frac{1}{s} \leqslant 1 + \log t$, we get the desired result. ■

We can make the following observations:

- For smooth problems, we can show a similar bound of the form $O(\kappa/t)$. For quadratic problems, constant step-sizes can be used with averaging, leading to improved convergence rates (Bach and Moulines, 2013). See the exercise below.

  **Exercise 5.28** (♦) *We consider the minimization of $F(\theta) = \frac{1}{2}\theta^\top H\theta - c^\top\theta$, where $H \in \mathbb{R}^{d\times}$ is positive definite (and thus invertible). We consider the recursion $\theta_t = \theta_{t-1} - \gamma[F'(\theta_{t-1}) + \varepsilon_t]$, where all $\varepsilon_t$'s are independent, with zero mean and covariance matrix equal to $C$. Compute explicity $\mathbb{E}\big[F(\theta_t) - F(\theta_*)\big]$, and provide an upper-bound of $\mathbb{E}\big[F(\bar\theta_t) - F(\theta_*)\big]$, where $\bar\theta_t = \frac{1}{t}\sum_{s=0}^{t-1}\theta_s$.*

- The bound in $O(B^2/\mu t)$ is optimal for this class of problems. That is, as shown for example by Agarwal et al. (2009), among all algorithms that can query stochastic gradients, having a better convergence rate (up to some constants) is impossible (see Section 15.3).

- We note that for the same regularized problem, we could use a step size proportional to $DB/\sqrt{t}$ and obtain a bound proportional to $DB/\sqrt{t}$, which looks worse than $B^2/(\mu t)$ but can, in fact, be better when $\mu$ is very small.

  Note also the loss of adaptivity: the step-size now depends on the problem's difficulty (this was different for deterministic gradient descent). See experiments below for illustrations.

**Exercise 5.29** *With the same assumptions as Prop. 5.8, show that with the step-size $\gamma_t = \frac{2}{\mu(t+1)}$, and with $\bar\theta_t = \frac{2}{t(t+1)}\sum_{s=1}^t s\theta_{s-1}$, we have: $\mathbb{E}\big[G(\bar\theta_t) - G(\theta_*)\big] \leqslant \frac{8B^2}{\mu(t+1)}$.*

**Experiments.** We consider a simple binary classification problem with linear predictors in dimension $d = 40$ (inputs generated from a Gaussian distribution, with binary outputs obtained as the sign of a linear function with additive Gaussian noise), with $n = 400$ observations, and observed features with $\ell_2$-norm bounded by $R$. We consider the hinge

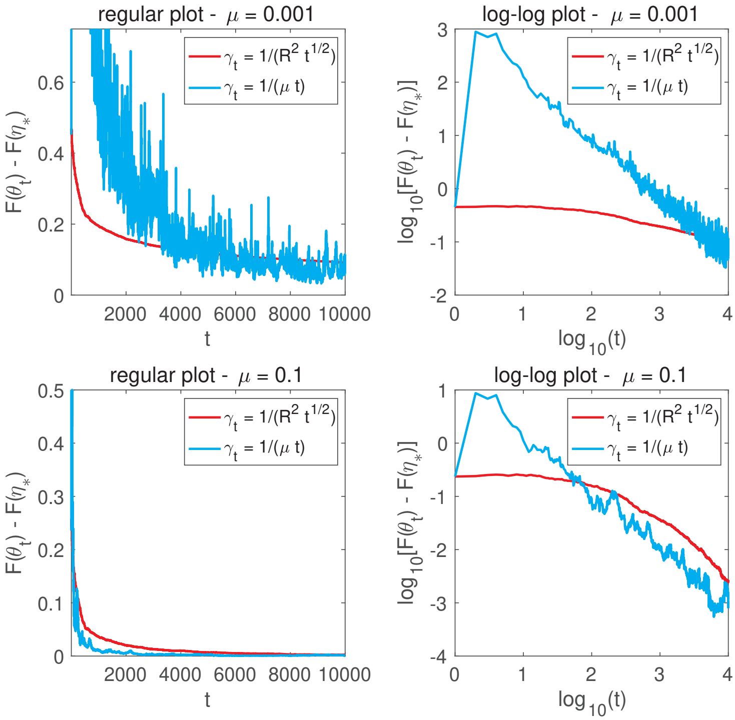


Figure 5.2: Comparison of step-sizes for SGD for the support vector machine, for two values of the regularization parameter $\mu$ (top: large $\mu$, bottom: small $\mu$). The performance is measured with a single run (hence the variability) on the excess training objective (left: regular plot, right: "log-log" plot).

loss with a squared $\ell_2$-regularizer $\frac{\mu}{2}\|\cdot\|_2^2$ (that is, the support vector machine presented in Section 4.1.2). We measure the excess training objective. We consider two values of $\mu$, and compare the two step-sizes $\gamma_t = 1/(R^2\sqrt{t})$ and $\gamma_t = 1/(\mu t)$ in Figure 5.2. We see that for large enough $\mu$, the strongly-convex step-size is better. This is not the case for small $\mu$.

The experiments above highlight the danger of a step-size equal to $1/(\mu t)$. In practice, it is often preferable to use $\gamma_t = \frac{1}{B^2\sqrt{t}+\mu t}$, as shown in the exercise below.

**Exercise 5.30** *(♦♦) With the same assumptions as in Prop. 5.8, with $\gamma_t = \frac{1}{B^2\sqrt{t}+\mu t}$, provide a convergence rate for the averaged iterate.*

### 5.4.2 Adaptive methods (♦)

The discussion on pre-conditioning for gradient descent on smooth functions at the end of Section 5.2.5 can be adapted to stochastic gradient methods for non-smooth problems. In this section, we highlight the potential gains and give references for precise results.

We focus on a linear prediction problem with i.i.d. features bounded in $\ell_2$-norm by $R$, and a convex $G$-Lipschitz-continuous loss function, in the setting of Prop. 5.7. For a constant step-size $\gamma$, in the proof of Prop. 5.7, we obtained an expected excess-risk equal to, starting from $\theta_0 = 0$,

$$\frac{1}{2\gamma t}\|\theta_*\|_2^2 + \frac{\gamma G^2}{2}\operatorname{tr}[\Sigma],$$

where $\Sigma = \mathbb{E}[\varphi(x)\varphi(x)^\top]$ is the covariance matrix of the features. Optimizing over $\gamma$ leads to the overall rate of $\frac{G\|\theta_*\|_2}{\sqrt{t}}\sqrt{\operatorname{tr}[\Sigma]}$.

Like done at the end of Section 5.2.5, pre-multiplying each gradient by the matrix $AA^\top$ is equivalent to minimizing the expectation of $\ell(y, \varphi(x)^\top A\tilde{\theta})$, which itself corresponds to replacing the feature map $\varphi$ by $A^\top\varphi$, and $\theta_*$ by $A^{-1}\theta_*$. The complexity bound above then becomes

$$\frac{1}{2\gamma t}\theta_*^\top(AA^\top)^{-1}\theta_* + \frac{\gamma G^2}{2}\operatorname{tr}[\Sigma AA^\top].$$

The matrix $M = (\gamma AA^\top)^{-1}$, which is the inverse of the matrix multiplying the gradient in the SGD iteration, can be optimized in the specific situation where we restrict the matrix $M$ to be diagonal with diagonal $m \in \mathbb{R}^d$. We then obtain the bound

$$\frac{1}{2t}\|\theta_*\|_\infty^2 \cdot \sum_{j=1}^d m_j + \frac{G^2}{2}\sum_{j=1}^d \frac{\Sigma_{jj}}{m_j},$$

with optimal $m_j$ equal to $\Sigma_{jj}^{1/2}G\sqrt{t}/\|\theta_*\|_\infty$ and an overall rate equal to $\frac{G\|\theta_*\|_\infty}{\sqrt{t}}\sum_{j=1}^d \Sigma_{jj}^{1/2}$, which can be substantially smaller than the corresponding rate with uniform $m$, proportional to $\frac{G\|\theta_*\|_\infty}{\sqrt{t}} d\sqrt{\sum_{j=1}^d \Sigma_{jj}}$; this in particular the case, where the $\Sigma_{jj}$'s have heterogeneous values.

In practice, we can estimate before running the learning algorithm the required elements of $\Sigma$, the (non-centered) covariance matrix of the features, and, more generally, the covariance of the gradients. These quantities can be estimated online, leading to the algorithms Adagrad (Duchi et al., 2011), or Adam (Kingma and Ba, 2014), which come with specific complexity bounds (see, e.g., Défossez et al., 2022).

### 5.4.3 Bias-variance trade-offs for least-squares (♦)

In this section, we consider the least-squares learning problems studied in Chapter 3, that is, we assume that we have i.i.d. observations $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, for $i \geqslant 1$, assuming that there exists a feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$ and $\theta_* \in \mathbb{R}^d$ such that $y_i = \varphi(x_i)^\top\theta_* + \varepsilon_i$, where $\varepsilon_i$ has mean zero and variance $\sigma^2$, and is independent of $x_i$. The goal of this section is to relate the performance of single-pass SGD to (regularized) empirical risk minimization studied in Section 3.3 and Section 3.6, and to study the impact of noise in SGD precisely.

The SGD recursion, often referred to as the least-mean-squares (LMS) recursion, can be written as, with a constant step-size:

$$\theta_t = \theta_{t-1} - \gamma(\theta_{t-1}^\top\varphi(x_t) - y_t)\varphi(x_t) = \theta_{t-1} - \gamma(\theta_{t-1}^\top\varphi(x_t) - \theta_*^\top\varphi(x_t) - \varepsilon_t)\varphi(x_t),$$

leading to

$$\theta_t - \theta_* = (I - \gamma \varphi(x_t)\varphi(x_t)^\top)(\theta_{t-1} - \theta_*) + \gamma \varepsilon_t \varphi(x_t). \tag{5.22}$$

Thus, like in the deterministic case in Section 5.2.1, we obtain a linear dynamical system, this time with random coefficients.

**Classical analysis.** We can first use a similar proof as in previous sections, that is, expanding Eq. (5.22),

$$\begin{aligned}\|\theta_t - \theta_*\|_2^2 &= \|\theta_{t-1} - \theta_*\|_2^2 + \|\gamma \varphi(x_t)\varphi(x_t)^\top(\theta_{t-1} - \theta_*)\|_2^2 \\ &\quad -2\gamma(\theta_{t-1} - \theta_*)^\top \varphi(x_t)\varphi(x_t)^\top(\theta_{t-1} - \theta_*) + \|\gamma \varepsilon_t \varphi(x_t)\|_2^2 \\ &\quad +2\gamma \varepsilon_t \varphi(x_t)^\top (I - \gamma \varphi(x_t)\varphi(x_t)^\top)(\theta_{t-1} - \theta_*),\end{aligned}$$

leading to, with $\mathcal{F}_{t-1}$ the information up to time $t-1$ (generated by $x_1, y_1, \dots, x_{t-1}, y_{t-1}$), and using that $\|\varphi(x_t)\|_2^2 \leqslant R^2$ almost surely, and the inequality $\Sigma = \mathbb{E}\big[\varphi(x_t)\varphi(x_t)^\top\big]$, and $\mathbb{E}\big[\|\varphi(x_t)\|_2^2 \varphi(x_t)\varphi(x_t)^\top\big] \preccurlyeq \Sigma$:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2 \big| \mathcal{F}_{t-1}\big] \leqslant \|\theta_{t-1} - \theta_*\|_2^2 + (\gamma^2 R^2 - 2\gamma)(\theta_{t-1} - \theta_*)^\top \Sigma (\theta_{t-1} - \theta_*) + \gamma^2 \sigma^2 R^2.$$

This leads to, with $F(\theta) - F(\theta_*) = \frac{1}{2}(\theta - \theta_*)^\top \Sigma (\theta - \theta_*)$, for $\gamma \leqslant 1/R^2$,

$$\mathbb{E}\big[F(\theta_{t-1}) - F(\theta_*)\big] \leqslant \frac{1}{2\gamma}\Big(\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\Big) + \frac{\gamma \sigma^2 R^2}{2},$$

and thus, for the average $\bar\theta_t = \frac{1}{t}\sum_{s=1}^t \theta_{s-1}$, using Jensen's inequality,

$$\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta_*)\big] \leqslant \frac{1}{2\gamma t}\|\theta_0 - \theta_*\|_2^2 + \frac{\gamma \sigma^2 R^2}{2},$$

which is a similar result to the non-smooth case but with an explicit bias/variance decomposition where the noise variance $\sigma^2$ explicitly appears, as well as the norm of $\theta_*$. Note that it requires the step-size to depend on the number of total iterations to obtain convergence.

However, for least-squares, a finer analysis can be performed, allowing explicitly for constant step-sizes and a clear relationship with generalization bounds for least-squares outlined in Chapter 3.

**Finer analysis of the LMS recursion (♦♦).** A detailed analysis of the LMS recursion in Eq. (5.22) is out of the scope of this textbook. However, a simplified recursion with essentially the same behavior can be analyzed with simple linear algebra tools. To obtain this simplified recursion, we rewrite Eq. (5.22) as

$$\theta_t - \theta_* = (I - \gamma \Sigma)(\theta_{t-1} - \theta_*) + \gamma \varepsilon_t \varphi(x_t) + \gamma(\Sigma - \varphi(x_t)\varphi(x_t)^\top)(\theta_{t-1} - \theta_*),$$

which is the recursion of the expected risk, corresponding to the term $(I - \gamma\Sigma)(\theta_{t-1} - \theta_*)$, plus additional stochastic terms with zero conditional mean. One of them, $\gamma \varepsilon_t \varphi(x_t)$ is purely "additive" (i.e., it does not depend on $\theta_{t-1}$) and has a constant non-zero variance, while the other one, $\gamma(\Sigma - \varphi(x_t)\varphi(x_t)^\top)(\theta_{t-1} - \theta_*)$ is multiplicative and has a variance that will go to zero as iterates converge to $\theta_*$. The simplified recursion ignores that term, and we now study the recursion (started at $\eta_0 = \theta_0$):

$$\eta_t - \theta_* = (I - \gamma\Sigma)(\eta_{t-1} - \theta_*) + \gamma\varepsilon_t\varphi(x_t), \tag{5.23}$$

which also corresponds to replacing $\varphi(x_t)\varphi(x_t)^\top$ in Eq. (5.22) by its expectation $\Sigma$ .

We can then explicitly unroll the recursion as:

$$\eta_t - \theta_* \quad = \quad (I - \gamma\Sigma)^t(\eta_0 - \theta_*) + \sum_{u=1}^{t} \gamma\varepsilon_u(I - \gamma\Sigma)^{t-u}\varphi(x_u),$$

with two parts, one which only depends on the initialization, that is, $(I - \gamma\Sigma)^t(\eta_0 - \theta_*)$, which is precisely the deterministic recursion from Section 5.2.1, and we refer to it as the "bias" part, and a part that depends on the noise variables $\varepsilon_u$, $u = 1, \dots, t$, which we refer to as the "variance" part. Assuming these noise variables are independent from $x$, the two parts can be considered totally independently when taking expectations.

We then have, for the averaged iterates:

$$\begin{aligned}
\bar\eta_t^{(\mathrm{bias})} - \theta_* &= \frac{1}{t}\sum_{v=0}^{t-1}(I - \gamma\Sigma)^v(\eta_0 - \theta_*) = \frac{1}{t}(\gamma\Sigma)^{-1}\big[I - (I - \gamma\Sigma)^t\big](\eta_0 - \theta_*) \\
\bar\eta_t^{(\mathrm{var})} - \theta_* &= \frac{1}{t}\sum_{v=1}^{t-1}\sum_{u=1}^{v}\gamma\varepsilon_u(I - \gamma\Sigma)^{v-u}\varphi(x_u) = .\frac{\gamma}{t}\sum_{u=1}^{t-1}\sum_{v=u}^{t-1}(I - \gamma\Sigma)^{v-u}\varepsilon_u\varphi(x_u) \\
&= \frac{1}{t}\sum_{u=1}^{t-1}\Sigma^{-1}\big[I - (I - \gamma\Sigma)^{t-u}\big]\varepsilon_u\varphi(x_u),
\end{aligned}$$

leading to

$$\begin{aligned}
\big\|\bar\eta_t^{(\mathrm{bias})} - \theta_*\big\|_\Sigma^2 &= \frac{1}{t^2}(\eta_0 - \theta_*)^\top(\gamma\Sigma)^{-2}\big[I - (I - \gamma\Sigma)^t\big]^2\Sigma(\eta_0 - \theta_*) \\
&\leqslant \frac{1}{\gamma^2 t^2}(\eta_0 - \theta_*)^\top\Sigma^{-1}(\eta_0 - \theta_*), \\
\mathbb{E}\Big[\big\|\bar\eta_t^{(\mathrm{var})} - \theta_*\big\|_\Sigma^2\Big] &= \frac{\sigma^2}{\gamma t^2}\sum_{u=1}^{t-1}\mathrm{tr}\,\Big[\Sigma^2\Sigma^{-2}\big[I - (I - \gamma\Sigma)^{t-u}\big]^2\Big] \leqslant \frac{\sigma^2 d}{t}.
\end{aligned}$$

We thus obtain two terms, the variance term in $\frac{\sigma^2 d}{t}$, which is present because the optimal prediction is not equal to the response, and the bias term in $\frac{1}{\gamma^2 t^2}(\eta_0 - \theta_*)^\top\Sigma^{-1}(\eta_0 - \theta_*)$, which corresponds to the forgetting of initial conditions. It is worth comparing to the same quantities for the non-averaged iterates: the bias term is upper-bounded by (using the same constants) $\frac{1}{\gamma^2 t^2}(\eta_0 - \theta_*)^\top\Sigma^{-1}(\eta_0 - \theta_*)$, but it is typically faster when the lowest eigenvalue of $\Sigma$ is strictly positive. The variance term is only of order $\gamma\sigma^2\,\mathrm{tr}[\Sigma]$ for the variance term (thus. with no convergence). This is illustrated in Figure 5.3.

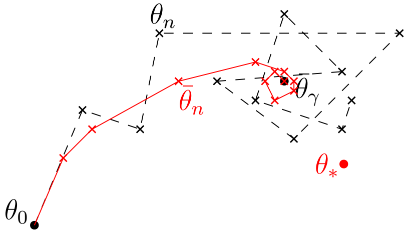


Figure 5.3: The iterates of SGD form a Markov chain, which is homogeneous when the step-size $\gamma$ is constant. It typically converges to a stationary distribution with expectation $\bar{\theta}_\gamma$, which happens to be the global minimum $\theta_*$ for quadratic costs (and with a deviation of $\gamma^2$ in general). The non-average iterates go from the initial condition $\theta_0$ to the vicinity of $\bar{\theta}_\gamma$, while the averaged iterates converge to that expectation $\bar{\theta}_\gamma$.

When $t = n$ iterations are performed, these should be compared to the excess risk for the least-squares estimators defined in Section 3.3 obtained by minimizing the empirical risk (only with the fixed design assumption). The variance term is the same as $\sigma^2 d/n = O(1/n)$, while the bias term is in $O(1/n^2)$ and seems smaller in the dependence in $n$. However, in high-dimensional problems, it can start to be larger for small $n$, highlighting the impact of forgetting initial conditions (see, e.g., Défossez and Bach, 2015).

The analysis provided in this section can be extended in several ways, for the "true" multiplicative noise, with similar results (Bach and Moulines, 2013; Défossez and Bach, 2015), to obtain dimension-free results akin to Section 3.6 (Dieuleveut and Bach, 2016; Dieuleveut et al., 2017), and to go beyond least-squares regression by studying logistic regression (Bach, 2014).

### 5.4.4 Variance reduction (♦)

We consider a finite sum $F(\theta) = \frac{1}{n}\sum_{i=1}^n f_i(\theta)$, where each $f_i$ is $R^2$-smooth (for example, logistic regression with features bounded by $R$ in $\ell_2$-norm), and which is such that $F$ is $\mu$-strongly convex (for example by adding $\frac{\mu}{2}\|\theta\|_2^2$ to each $f_i$, or by benefitting from the strong convexity of the sum). We denote by $\kappa = R^2/\mu$ the condition number of the problem (note that it is more significant than $L/\mu$, where $L$ is the smoothness constant of $F$).

Using SGD, the convergence rate has been shown to be $O(\kappa/t)$ in Section 5.4.1, with iterations of complexity $O(d)$, while for GD, the convergence rate is $O(\exp(-t/\kappa))$ (see Section 5.2.3), but each iteration has complexity $O(nd)$. We now present a result allowing exponential convergence with an iteration cost of $O(d)$.

The idea is to use a form of *variance reduction*, made possible by keeping in memory past gradients. We denote by $z_i^{(t)} \in \mathbb{R}^d$ the version of gradient $i$ stored at time $t$.

The SAGA algorithm (Defazio et al., 2014), which combines the earlier algorithms SAG (Schmidt et al., 2017) and SVRG (Johnson and Zhang, 2013; Zhang et al., 2013), works as follows: at every iteration, an index $i(t)$ is selected uniformly at random in

$\{1,\dots,n\}$, and we perform the iteration

$$\theta_t = \theta_{t-1} - \gamma\Big[f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\Big],$$

with $z_{i(t)}^{(t)} = f'_{i(t)}(\theta_{t-1})$ and all others $z_i^{(t)}$ left unchanged (i.e., the same as $z_i^{(t-1)}$). In words, we add to the update with the stochastic gradient $f'_{i(t)}(\theta_{t-1})$ the corrective term $\frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}$, which has zero expectation with respect to $i(t)$. Thus, since the expectation of $f'_{i(t)}(\theta_{t-1})$ with respect to $i(t)$ is equal to the full gradient $F'(\theta)$, the update is *unbiased* like for regular SGD. The goal is to reduce its variance.

The idea behind variance reduction is that if the random variable $z_{i(t)}^{(t-1)}$ (only considering the source of randomness coming from $i(t)$) is positively correlated with $f'_{i(t)}(\theta_{t-1})$, then the variance is reduced, and larger step-sizes can be used.

As the algorithm converges, then $z_i^{(t)}$ converges to $f'_i(\eta_*)$ (the individual gradient at optimum). We will show that *simultaneously* $\theta_t$ converges to $\eta_*$ *and* $z_i^{(t)}$ converges to $f'_i(\eta_*)$ for all $i$, all at the same speed.

**Proposition 5.9 (Convergence of SAGA)** *If initializing with $z_i^{(0)} = f'_i(\theta_0)$ at the initial point $\theta_0 \in \mathbb{R}^d$, for all $i \in \{1,\dots,n\}$, we have, for the choice of step-size $\gamma = \frac{1}{4R^2}$:*

$$\mathbb{E}\big[\|\theta_t - \eta_*\|_2^2\big] \leqslant \Big(1 - \min\Big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\Big\}\Big)^t \big(1 + \frac{n}{4}\big)\|\theta_0 - \eta_*\|_2^2. \tag{5.24}$$

**Proof** (♦♦) The proof consists in finding a Lyapunov function that decays along iterations.

**Step 1.** We first try our "usual" Lyapunov function, making the differences $\|z_i^{(t)} - f'_i(\eta_*)\|_2^2$ appear, with the update $\theta_t = \theta_{t-1} - \gamma\omega_t$, with $\omega_t = \big[f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\big]$,

$$\begin{aligned}
\|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top \omega_t + \gamma^2\big\|\omega_t\big\|_2^2 \text{ by expanding the square,}\\
\mathbb{E}_{i(t)}\|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1})\\
&\qquad + \gamma^2\mathbb{E}_{i(t)}\Big\|f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\Big\|_2^2,
\end{aligned}$$

using the unbiasedness of the stochastic gradient. We further get

$$\begin{aligned}
\mathbb{E}_{i(t)}\|\theta_t - \eta_*\|_2^2 &\leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1})\\
&\quad + 2\gamma^2\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\big\|_2^2 + 2\gamma^2\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)} + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)}\big\|_2^2,
\end{aligned}$$

using $\|a+b\|_2^2 \leqslant 2\|a\|_2^2 + 2\|b\|_2^2$. To bound $\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\big\|_2^2$, we use co-coercivity of all functions $f_i$ (see Prop. 5.4), to get:

$$\begin{aligned}
\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\big\|_2^2 &= \frac{1}{n}\sum_{i=1}^n \big\|f'_i(\theta_{t-1}) - f'_i(\eta_*)\big\|_2^2 \\
&\leqslant \frac{1}{n}\sum_{i=1}^n R^2[f'_i(\theta_{t-1}) - f'_i(\eta_*)]^\top(\theta_{t-1} - \theta_*) \\
&\leqslant R^2 F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*) \text{ since } \sum_{i=1}^n f'_i(\eta_*) = 0. \quad (5.25)
\end{aligned}$$

To bound $\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)} + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)}\big\|_2^2$, we use $\mathbb{E}_{i(t)}\|Z - \mathbb{E}_{i(t)}Z\|_2^2 \leqslant \mathbb{E}_{i(t)}\|Z\|_2^2$. We thus get

$$\begin{aligned}
\mathbb{E}_{i(t)}\|\theta_t - \eta_*\|_2^2 &\leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + 2\gamma^2 R^2(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \\
&\qquad + 2\gamma^2\frac{1}{n}\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2, \\
&\leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(1 - \gamma R^2)(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \\
&\qquad + 2\frac{\gamma^2}{n}\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2.
\end{aligned}$$

**Step 2.** We see the term $\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2$ appearing, so we try to study how it varies across iterations. We have, by definition of the updates of the vectors $z_i^{(t)}$:

$$\begin{aligned}
\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2 &= \sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2 \\
&\qquad - \big\|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)}\big\|_2^2 + \big\|f'_{i(t)}(\eta_*) - f'_{i(t)}(\theta_{t-1})\big\|_2^2.
\end{aligned}$$

Taking expectations with respect to $i(t)$, we get

$$\begin{aligned}
\mathbb{E}_{i(t)}\Big[\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2\Big] &= \Big(1 - \frac{1}{n}\Big)\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2 + \frac{1}{n}\sum_{i=1}^n \big\|f'_i(\eta_*) - f'_i(\theta_{t-1})\big\|_2^2 \\
&\leqslant \Big(1 - \frac{1}{n}\Big)\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2 + R^2(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}),
\end{aligned}$$

where we use the bound in Eq. (5.25). Thus, for a positive number $\Delta$ to be chosen later,

$$\begin{aligned}
&\mathbb{E}_{i(t)}\Big[\|\theta_t - \eta_*\|_2^2 + \Delta\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2\Big] \\
&\leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma\Big(1 - \gamma R^2 - \frac{R^2\Delta}{2\gamma}\Big)(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \\
&\qquad + \Big[2\frac{\gamma^2}{n\Delta} + (1 - 1/n)\Big]\Delta\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2.
\end{aligned}$$

With $\Delta = 3\gamma^2$ and $\gamma = \frac{1}{4R^2}$, we get $1 - \gamma R^2 - \frac{R^2\Delta}{2\gamma} = \frac{3}{8}$ and $2\frac{\gamma^2}{n\Delta} = \frac{2}{3n}$. Moreover, using the identity $(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \geqslant \mu \|\theta_{t-1} - \eta_*\|_2^2$ as a consequence of strong convexity, we then get:

$$\mathbb{E}_{i(t)}\Big[\|\theta_t - \eta_*\|_2^2 + \Delta \sum_{i=1}^n \big\|f_i'(\eta_*) - z_i^{(t)}\big\|_2^2\Big] \leqslant \Big(1 - \min\Big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\Big\}\Big)\Big[\|\theta_{t-1} - \eta_*\|_2^2 + \Delta \sum_{i=1}^n \big\|f_i'(\eta_*) - z_i^{(t-1)}\big\|_2^2\Big].$$

Thus

$$\mathbb{E}\big[\|\theta_t - \eta_*\|_2^2\big] \leqslant \Big(1 - \min\Big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\Big\}\Big)^t \Big[\|\theta_0 - \eta_*\|_2^2 + \frac{3}{16R^4} \sum_{i=1}^n \big\|f_i'(\eta_*) - z_i^{(0)}\big\|_2^2\Big].$$

If initializing with $z_i^{(0)} = f_i'(\theta_0)$, we get the desired bound by using the Lipschitz-continuity of each $f_i'$, which leads to $(1 + \frac{3n}{16})\|\theta_0 - \eta_*\|_2^2 \leqslant (1 + \frac{n}{4})\|\theta_0 - \eta_*\|_2^2$. This leads to the final bound in Eq. (5.24). ■

We can make the following observations:

- The contraction rate after one iteration is $\big(1 - \min\big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\big\}\big)$, which is less than $\exp\big(-\min\big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\big\}\big)$. Thus, after an "effective pass" over the data, that is, $n$ iterations, the contracting rate is $\exp\big(-\min\big\{\frac{1}{3}, \frac{3\mu n}{16R^2}\big\}\big)$. It is only an effective pass because after we sample $n$ indices with replacement, we will not see all functions (while some will be seen several times).

  In order to have a contracting effect of $\varepsilon$, that is, having $\|\theta_t - \eta_*\|_2^2 \leqslant \varepsilon\|\theta_0 - \eta_*\|_2^2$, we need to have $\exp\big(-t\min\big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\big\}\big)2n \leqslant \varepsilon$, which is equivalent to $t \geqslant \max\big\{3n, \frac{16R^2}{3\mu}\big\}\log\frac{2n}{\varepsilon}$. It just suffices to have $t \geqslant \big(3n + \frac{16R^2}{3\mu}\big)\log\frac{2n}{\varepsilon}$, and thus the running time complexity is equal to $d$ times the minimal number, that is

  $$d\Big(3n + \frac{16R^2}{3\mu}\Big)\log\frac{2n}{\varepsilon}.$$

  This is to be contrasted with batch gradient descent with step-size $\gamma = 1/R^2$ (which is the simplest step-size that can be computed easily), whose complexity is

  $$dn\frac{R^2}{\mu}\log\frac{1}{\varepsilon}.$$

  We replace the product of $n$ and condition number $\kappa = \frac{R^2}{\mu}$ by a sum, which is significant where $\kappa$ is large.

- Multiple extensions of this result are available, such as a rate for non-strongly-convex functions, adaptivity to strong-convexity, proximal extensions, and acceleration. It is also worth mentioning that the need to store past gradients can be alleviated (see Gower et al., 2020, for more details).

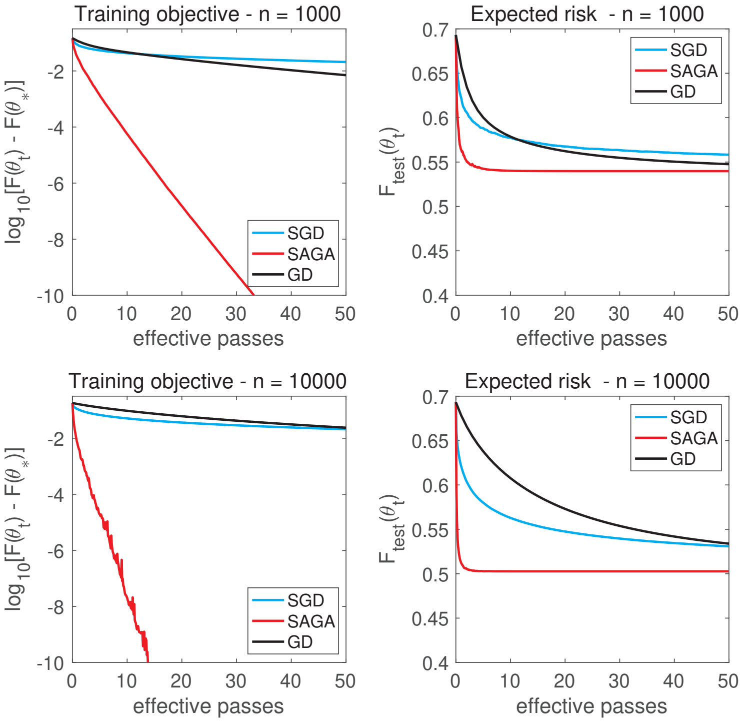


Figure 5.4: Comparison of stochastic gradient algorithms for logistic regression. Top: $n = 1000$, bottom: $n = 10000$. Left: training objective in semi-log plot, right: expected risk estimated with $n$ test points.

- Note that these fast algorithms allow very small optimization errors and that the best testing risks will typically be obtained after a few (10 to 100) passes.

**Experiments.** We consider $\ell_2$-regularized logistic regression, and we compare GD, SGD, and SAGA, all with their corresponding step-sizes coming from the theoretical analysis, with two values of $n$. We use a simple binary classification problem with linear predictors in dimension $d = 40$ (inputs generated from a Gaussian distribution, with binary outputs obtained as the sign of a linear function with additive Gaussian noise), with two different numbers of observations $n$, and regularization parameter $\mu = R^2/n$. See Figure 5.4 (top: small $n$, left: large $n$). We see that for early iterations, SGD dominates GD, while for larger numbers of iterations, GD is faster. This last effect is not seen for large numbers of observations (right), where SGD always dominates GD. SAGA gets to machine precision after 50 effective passes over the data in the two cases. Note also the better performance on the testing data.

## 5.5 Conclusion

**Convex finite-dimensional problems.** We can now provide a summary of convergence rates below, with the main rates we have seen in this chapter (and some that we have not seen) for convex objective functions. We separate between convex and strongly convex, and between smooth and non-smooth, as well as between deterministic and stochastic methods. Below, $L$ is the smoothness constant, $\mu$ the strong convexity constant, $B$ the Lipschitz constant, and $D$ the distance to optimum at initialization.

| | convex | strongly convex |
|---|---|---|
| nonsmooth | deterministic: $BD/\sqrt{t}$<br>stochastic: $BD/\sqrt{t}$ | deterministic: $B^2/(t\mu)$<br>stochastic: $B^2/(t\mu)$ |
| smooth | deterministic: $LD^2/t^2$<br>stochastic: $LD^2/\sqrt{t}$<br>finite sum: $n/t$ | deterministic: $\exp(-t\sqrt{\mu/L})$<br>stochastic: $L/(t\mu)$<br>finite sum: $\exp(-\min\{1/n, \mu/L\}t)$ |

The convergence rates are often written as a number $t$ of accesses to individual gradients to achieve excess function values of $\varepsilon$. This corresponds to inverting each formula for $\varepsilon$ as a function of $t$ to a formula for $t$ as a function of $\varepsilon$. This leads to the following table:

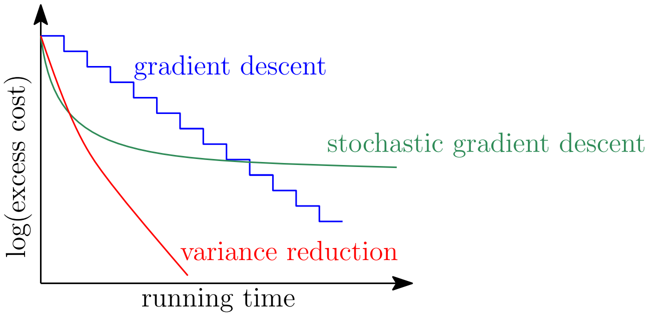
| | convex | strongly convex |
|---|---|---|
| nonsmooth | deterministic: $(BD)^2/\varepsilon^2$<br>stochastic: $(BD)^2/\varepsilon^2$ | deterministic: $B^2/(\varepsilon\mu)$<br>stochastic: $B^2/(\varepsilon\mu)$ |
| smooth | deterministic: $\sqrt{L}D/\sqrt{\varepsilon}$<br>stochastic: $(LD^2)^2/\varepsilon^2$<br>finite sum: $n/\varepsilon$ | deterministic: $\exp(-t\sqrt{\mu/L})$<br>stochastic: $L/(\varepsilon\mu)$<br>finite sum: $\max\{n, L/\mu\}\log(1/\varepsilon)$ |

Like in the rest of the book, where we obtain explicit convergence rates, the homogeneity of all quantities can be checked (see exercise below). In the context optimization, this ensures that algorithms are invariant by change of variable $\theta \to \alpha\theta$ for $\alpha \neq 0$.

**Exercise 5.31** *Check the homogeneity of all quantities of this section (step-size and convergence rates).*

Note that many important themes in optimization have been ignored, such as Frank-Wolfe methods (presented in Chapter 9), coordinate descent, or duality. See Nesterov (2018); Bubeck (2015) for further details. See also Chapter 7 and Chapter 9 for optimization methods for kernel methods and neural networks.

For strongly-convex smooth problems, the following toy figure also provides a good summary, with gradient descent being along a line in a semi-log plot (that is, exponential convergence) but with a staircase effect due to the lack of progress while computing the full gradient, SGD starting fast but having trouble reaching low optimization error, with variance reduction getting the best of both worlds, together with a faster rate of convergence than regular GD.



**Beyond finite-dimensional problems.** Supervised machine learning problems leading to finite-dimensional convex objective functions are essentially problems with prediction functions that are linear in their parameters, with a feature map that can be explicitly computed. In Chapter 7, we extend some of the algorithms seen in this chapter to features that are only available through dot-products $\varphi(x)^\top \varphi(x')$.

**Beyond convex problems.** Complexity bounds can be obtained beyond convex problems, as shown briefly in Section 5.2.6. However, they only certify that the gradient norm will go to zero, not that a global optimum has been approximately reached. Objective functions obtained from neural network training provide an important class of non-convex objective functions that we consider in Chapter 9.

**Generalization bounds: Rademacher or SGD?** In the last chapter, we have shown how to obtain generalization bounds for the constrained or regularized empirical risk minimizer. They relied on Rademacher complexities, which apply to all Lipschitz-continuous loss functions (not necessarily convex). However, they leave open how to obtain algorithmically such minimizers. In this section, we have not only seen algorithms to obtain such minimizers through gradient-based techniques but also single-pass SGD that directly provides the same generalization bound on unseen data for an efficient algorithm. We will see in Section 11.1.3 how this extends to the mirror descent framework to account for non-Euclidean geometries.

These two ways of obtaining generalization bounds will also be compared for multi-category classification in Chapter 13, where SGD-based bounds will lead to better bounds.

# Chapter 6

# Local averaging methods

**Chapter summary**

- First chapter on non-parametric methods that are not based on parametric models and can adapt to complex target functions.
- "Linear" estimators: These are based on assigning weight functions to each observation so that each observation can vote for its label with the corresponding weight (typically non-linear in the input variables).
- Partitioning estimates: the input space is cut into non-overlapping cells, and the predictor is piecewise-constant.
- Nadaraya-Watson estimators (a.k.a. kernel regression): each observation assigns a weight proportional to its distance in input space.
- $k$-nearest-neighbors: each observation assigns an equal weight to its $k$ nearest neighbors.
- Consistency: All of these methods can provably learn complex Lipschitz-continuous non-linear functions with a convergence rate of the form $O(n^{-2/(d+2)})$, where $d$ is the underlying input dimension, leading to the curse of dimensionality.

## 6.1 Introduction

Like in previous chapters, we consider the supervised learning set-up, where we are being given a training set: observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of inputs/outputs, features/variables are assumed independent and identically distributed (i.i.d.) random variables with common distribution $p$. We consider a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, where $\ell(y, z)$ is the loss of predicting $z$ while the true label is $y$.

Our goal is to minimize the expected risk, that is, the generalization performance of a prediction function $f$ from $\mathcal{X}$ to $\mathcal{Y}$:

$$\mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big],$$

where the expectation is taken with respect to the distribution $p$.

⚠ Like in the rest of the book, we assume that the testing distribution is the same as the training distribution.

⚠ Be careful with the randomness or lack thereof of $f$: The estimator $\hat{f}$ we will use depends on the training data and not on the testing data, and thus $\mathcal{R}(\hat{f})$ is random because of the dependence on the training data.

As seen in Chapter 2, the two classical cases are:

- Binary classification: $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$), and $\ell(y, z) = 1_{y \neq z}$ ("0-1" loss). Then $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$.
- Regression: $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). Then $\mathcal{R}(f) = \mathbb{E}(y - f(x))^2$.

As seen in Chapter 2, minimizing the expected risk leads to an optimal "target function," called the Bayes predictor $f^* \in \arg\min \mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big]$. As shown in Section 2.2.3, the optimal predictor can be obtained from the conditional distribution of $y|x$ as

$$f^*(x) \in \arg\min_{z \in \mathcal{Y}} \ \mathbb{E}(\ell(y, z)|x).$$

Note that (a) the Bayes predictor is not unique but that all Bayes predictors lead to the same Bayes risk, and (b) the Bayes risk is usually non-zero (unless the dependence between $x$ and $y$ is deterministic). The goal of supervised machine learning is thus to estimate $f^*$, knowing only the training data $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ *and* the loss $\ell$, with the goal of minimizing the risk or the excess risk $\mathcal{R}(f) - \mathcal{R}^*$. We have the following special cases:

- For binary classification: $\mathcal{Y} = \{0, 1\}$ and $\ell(y, z) = 1_{y \neq z}$, the Bayes predictor is equal to $f^*(x) \in \arg\max_{i \in \{0,1\}} \mathbb{P}(y = i|x)$. This extends naturally to multi-category classification with the Bayes predictor $f^*(x) \in \arg \max\limits_{i \in \{1,\dots,k\}} \mathbb{P}(y = i|x)$.

  If a convex surrogate from Section 4.1.1 is used, such as the logistic loss $\ell(y, z) = \log(1 + \exp(-yz))$ for $z \in \mathbb{R}$, then the target function is $f^*(x) = \log \frac{\mathbb{P}(y=1|x)}{\mathbb{P}(y=-1|x)}$.
- For regression: $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$, the Bayes predictor is $f^*(x) = \mathbb{E}(y|x)$. Moreover, we have $\mathcal{R}(f) - \mathcal{R}^* = \int_{\mathcal{X}} (f(x) - f^*(x))^2 dp(x) = \|f - f^*\|^2_{L_2(dp(x))}$.

In Chapter 3 and Chapter 4, we explored methods based on empirical risk minimization, with explicit finite-dimensional models (often linear in their parameters) that may not be flexible enough to adapt to complex target functions. We now explore methods that can, starting with local averaging methods, which are not based on empirical risk minimization. In subsequent chapters, we will study kernel methods (Chapter 7) and neural networks (Chapter 9).

## 6.2 Local averaging methods

In local averaging methods, we aim at approximating the target function $f^*$ directly *without any form of optimization*. This will be done by approximating the conditional distribution $p(y|x)$ of $y$ given $x$, by some $\widehat{p}(y|x)$. We then replace the target function $f^*(x) \in \arg\min_{z\in\mathcal{Y}} \int_{\mathcal{Y}} \ell(y,z)dp(y|x)$ by $\hat{f}(x) \in \arg\min_{z\in\mathcal{Y}} \int_{\mathcal{Y}} \ell(y,z)d\widehat{p}(y|x)$. These are often called "plug-in" estimators.

In the usual cases, this leads to the following predictions:

- For classification with the 0-1 loss: $\hat{f}(x) \in \arg \max_{j\in\{1,\dots,k\}} \widehat{\mathbb{P}}(y=j|x)$.
- For regression with the square loss: $\hat{f}(x) = \int_{\mathcal{Y}} y \ d\widehat{p}(y|x)$.

### 6.2.1 Linear estimators

In this chapter, we will consider "linear" estimators, where the conditional distribution is of the form

$$\widehat{p}(y|x) = \sum_{i=1}^n \hat{w}_i(x)\delta_{y_i}(y),$$

where $\delta_{y_i}$ is the Dirac probability distribution at $y_i$ (putting a unit mass at $y_i$), and the weight functions $\hat{w}_i : \mathcal{X} \to \mathbb{R}$, $i = 1, \dots, n$, depends on the input data only (for simplicity) and satisfy (almost surely in $x$):

$$\forall x \in \mathcal{X}, \quad \forall i \in \{1, \dots, n\}, \ \hat{w}_i(x) \geqslant 0, \text{ and } \sum_{i=1}^n \hat{w}_i(x) = 1.$$

These conditions ensure that for all $x \in \mathcal{X}$, $\widehat{p}(y|x)$ is a probability distribution.

⚠ Some references allow for the weights not to sum to 1.

For our running examples, this leads to the following predictions:

- For classification: $\hat{f}(x) \in \arg \max_{j\in\{1,\dots,k\}} \sum_{i=1}^n \hat{w}_i(x) 1_{y_i=j}$, that is, each observation $(x_i, y_i)$ votes for its label with weight $\hat{w}_i(x)$, a strategy often called "majority vote".
- For regression: $\mathcal{Y} = \mathbb{R}$: $\hat{f}(x) = \sum_{i=1}^n \hat{w}_i(x) y_i$. This is why the terminology "linear estimators" is sometimes used, since, as a function of the response vector in $\mathbb{R}^n$, the estimator is linear (note that this is the case as well for kernel ridge regression in Chapter 7). If we only consider predictions $\hat{f}(x_i)$ at the observed inputs, the vector $\hat{y} \in \mathbb{R}^n$ of predictions $\hat{y}_i = \hat{f}(x_i)$, for $i \in \{1, \dots, n\}$ is of the form $\hat{y} = Hy$, where the matrix $H \in \mathbb{R}^{n\times n}$, often called the smoothing matrix or the "hat matrix", is such that $H_{ij} = \hat{w}_j(x_i)$.

  Note that on top of being a linear estimator, the estimator satisfies additional properties: if the same constant is added to all outputs, the exact same constant

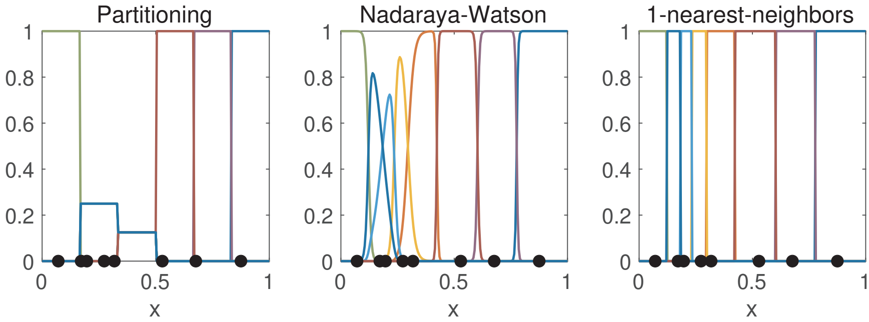


Figure 6.1: Weights of linear estimators in $d = 1$ dimension for the three types of local averaging estimators. The $n = 8$ weight functions $x \mapsto \hat{w}_i(x)$ are plotted with the observations in black.

is added to the prediction function; moreover, given two vectors of outputs $y$ and $y' \in \mathbb{R}^n$, with two prediction functions $\hat{f}$ and $\hat{f}'$, if $y_i \leqslant y'_i$ for all $i \in \{1, \dots, n\}$, then $\hat{f}(x) \leqslant \hat{f}'(x)$ for all $x \in \mathcal{X}$.

**Construction of weight functions.** In most cases, for any $i$, the weight function $\hat{w}_i(x)$ is close to 1 for training points $x_i$ which are close to $x$. We now show three classical ways of building them: (1) partition estimators, (2) Nearest-neighbors, and (3) Nadaraya-Watson estimator (a.k.a. kernel regression). See examples in Figure 6.1.

### 6.2.2 Partition estimators

If $\mathcal{X} = \bigcup_{j \in J} A_j$ is a partition (such that for all distinct $j, j' \in J$, $A_j \cap A_{j'} = \varnothing$) of $\mathcal{X}$ with a countable index set $J$ (which we will assume finite for simplicity, equal to $\{1, \dots, |J|\}$), then we can consider for any $x \in \mathcal{X}$ the corresponding element $A(x)$ of the partition (that is, $A(x)$ is the unique $A_j$, $j \in J$, such that $x \in A_j$), and define

$$\hat{w}_i(x) = \frac{1_{x_i \in A(x)}}{\sum_{i'=1}^n 1_{x_{i'} \in A(x)}}, \tag{6.1}$$

with the convention that if no training data point lies in $A(x)$, then $\hat{w}_i(x)$ is equal to $1/n$ for each $i \in \{1, \dots, n\}$. This implies that each $\hat{w}_i$ is piecewise constant with respect to the partition, that is, for any non-empty cell $A_j$ (that is, for which at least one observation falls in $A_j$), for any $x \in A_j$, the vectors $(\hat{w}_i(x))_{i \in \{1,\dots,n\}}$ has weights equal to $1/n_{A_j}$ for $i \in A_j$, where $n_{A_j}$ is the number of training points in the set $A_j$, and 0 otherwise.

**Equivalence with least-squares regression.** When applied to regression where the estimator is $\hat{f}(x) = \sum_{i=1}^n \hat{w}_i(x) y_i$, then using a partition estimator can be seen as a least-squares estimator with feature vector $\binom{\varphi(x)}{1} = \binom{(1_{x \in A_j})_{j \in J}}{1} \in \mathbb{R}^{|J|+1}$, as we now show.

Indeed, we then aim to estimate $\binom{\theta}{\eta} \in \mathbb{R}^{|J|+1}$ for the prediction function

$$\hat{f}(x) = \sum_{j \in J} \theta_j 1_{x \in A_j} + \eta.$$

From training data $(x_1, y_1), \dots, (x_n, y_n)$, as shown in Chapter 3, we can directly estimate the constant term as $\eta = \bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$, while for the other components, we need to solve the normal equations

$$\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top \theta = \sum_{i=1}^n (y_i - \bar{y})\varphi(x_i).$$

It turns out that the matrix $n\widehat{\Sigma} = \sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top$ is diagonal where for each $j \in J$, $n\widehat{\Sigma}_{jj}$ is equal to $n_{A_j}$ the number of data points lying in cell $A_j$. This implies that for a non-empty cell $A_j$, $\theta_j$ is the average of all $y_i - \bar{y}$, for all $i$ such that $x_i$ lies in $A_j$. Thus, for all $x \in A_j$, the prediction is exactly $\theta_j + \bar{y}$, as obtained from weights in Eq. (6.1). For empty cells, $\theta_j$ is not determined by the normal equations above, and if we set it to zero, we recover our convention of predicting as the mean of all labels.

⚠ Other conventions exist (such as all zero weights when no data point lies in $A(x)$).

This equivalence with least-squares estimation with a diagonal (empirical or not) non-centered covariance matrix makes it attractive for theoretical purposes, as the inversion of the population and expected covariance matrices could be done in closed form.

**Choice of partitions.** There are two standard applications of partition estimators:

- **Fixed partitions**: for example, when $\mathcal{X} = [0,1]^d$, we can consider cubes of length $h$, with $|J| = h^{-d}$ (see example below in $d = 2$ dimension with $|J| = 25$). Note here that the computation time for each $x \in \mathcal{X}$ is not necessarily proportional to $|J|$ but to $n$ (by simply considering the bins where the data lie). This estimator is sometimes called a "regressogram". We need then to choose the bandwidth $h$ (see analysis in Section 6.3.1). See Figure 6.2 for an illustration in one dimension.

| | | | | |
|---|---|---|---|---|
| $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
| $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ |
| $A_{11}$ | $A_{12}$ | $A_{13}$ | $A_{14}$ | $A_{15}$ |
| $A_{16}$ | $A_{17}$ | $A_{18}$ | $A_{19}$ | $A_{20}$ |
| $A_{21}$ | $A_{22}$ | $A_{23}$ | $A_{24}$ | $A_{25}$ |

- **Decision trees**: for data in a hypercube, we can recursively partition it by selecting a variable to split, leading to a maximum reduction in errors when defining the partitioning estimate.[1] A model selection criterion is then needed to control

[1] See more details in https://en.wikipedia.org/wiki/Decision_tree_learning.

the number of cells in the partition (see, e.g., Friedman et al., 2009, Section 9.2). Note here that the partition depends on the labels (so the analysis below does not apply unless the partitioning is learned on data different from the one used for the estimation).

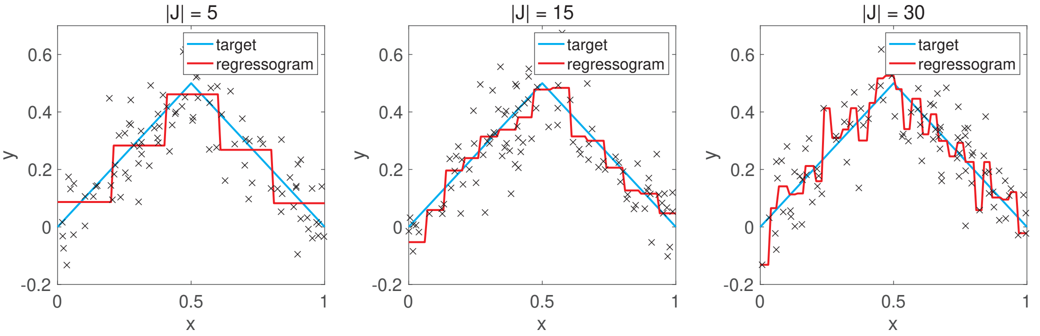


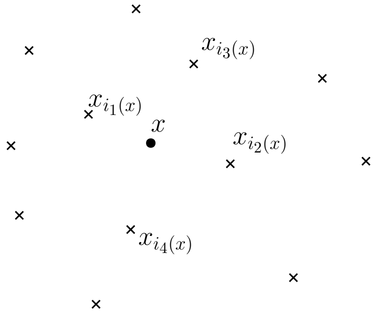
Figure 6.2: Regressograms in $d = 1$ dimension, with three values of $|J|$ (the number of sets in the partition). The $n = 100$ input data points are distributed uniformly on $[0, 1]$, and, for $i \in \{1, \ldots, n\}$, the outputs $y_i$ are equal to $\frac{1}{2} - |x_i - \frac{1}{2}| + \varepsilon_i$, where $\varepsilon_i$ is a Gaussian with mean zero and variance $\sigma^2 = \frac{1}{100}$. We can observe both underfitting ($|J|$ too small) and overfitting ($|J|$ too large). Note that the target function $f^*$ is piecewise affine and that on the affine parts, the estimator is far from linear; that is, the estimator cannot take advantage of extra-regularity (see Section 6.5 for more details).

### 6.2.3 Nearest-neighbors

Given an integer $k \geqslant 1$, and a distance $d$ on $\mathcal{X}$, for any $x \in \mathcal{X}$, we can order the $n$ observations so that

$$d(x_{i_1(x)}, x) \leqslant d(x_{i_2(x)}, x) \leqslant \cdots \leqslant d(x_{i_n(x)}, x),$$

where $\{i_1(x), \ldots, i_n(x)\} = \{1, \ldots, n\}$, and ties are broken randomly[2] (that is, for all $x \in \mathcal{X}$, by sampling randomly once, which indices should come first for each $i$). See the illustration below.

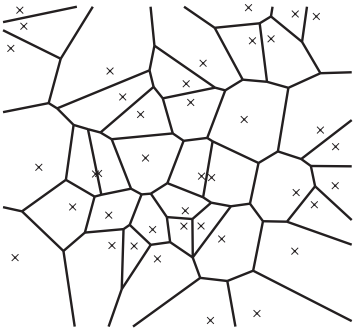


[2] Other conventions share the weights among all ties.

We then define

$$\hat{w}_i(x) = 1/k \text{ if } i \in \{i_1(x), \ldots, i_k(x)\}, \text{ and 0 otherwise.}$$

Given a new input $x \in \mathbb{R}^d$, the nearest neighbor predictor looks at the $k$ nearest points $x_i$ in the data set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and predicts a majority vote among them (for classification) or simply the averaged response (for regression). The number of nearest neighbors is the hyperparameter, which needs to be estimated (typically by cross-validation); see Section 6.3.2 for an analysis. See a one-dimensional example in Figure 6.3. For $k = 1$, the prediction function is piecewise constant, with each constant piece corresponding to a region where a given observation is the nearest neighbor, leading, in two dimensions to the Voronoi diagram below, with all regions displayed.



**Algorithms.** Given a test point $x \in \mathcal{X}$, the naive algorithm looks at all training data points for computing the predicted response. Thus the complexity is $O(nd)$ per test point in $\mathbb{R}^d$. When $n$ is large, this is costly in time and memory. Indexing techniques exist for (potentially approximate) nearest-neighbor search, such as "k-d-trees",[3] with typically a logarithmic complexity in $n$ (but with some additional compiling time), with a memory footprint that can grow exponentially in dimension (see, e.g., Shakhnarovich et al., 2005).

**Exercise 6.1** *What is the pattern of non-zeros of the smoothing matrix $H \in \mathbb{R}^{n \times n}$?*

### 6.2.4 Nadaraya-Watson estimator a.k.a. kernel regression (♦)

Given a "kernel" function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, which is pointwise non-negative, we define

$$\hat{w}_i(x) = \frac{k(x, x_i)}{\sum_{i'=1}^n k(x, x_{i'})},$$

with the convention that if $k(x, x_i) = 0$ for all $i \in \{1, \ldots, n\}$, then $\hat{w}_i(x)$ is equal to $1/n$ for each $i$ (which is the same convention used for estimators based on partitions in Section 6.2.2).

[3] See https://en.wikipedia.org/wiki/K-d_tree.

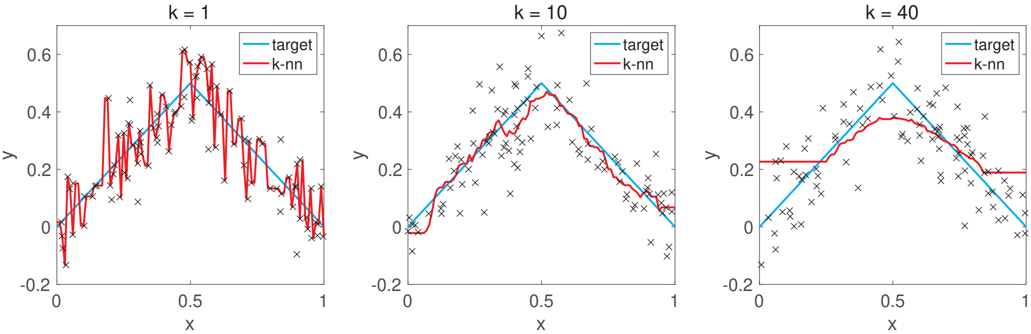
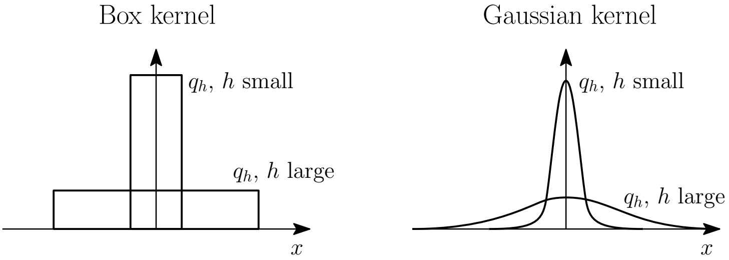


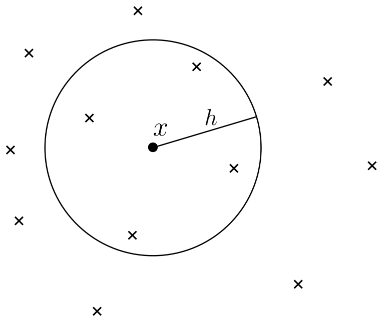
Figure 6.3: $k$-nearest neighbor regression in $d = 1$ dimension, with three values of $k$ (the number of neighbors), with the same data as Figure 6.2. We can observe both underfitting ($k$ too large) and overfitting ($k$ too small).

In most cases where $\mathcal{X} \subset \mathbb{R}^d$, we take $k(x, x') = h^{-d} q\big(\frac{1}{h}(x - x')\big)$ for a certain function $q : \mathbb{R}^d \to \mathbb{R}_+$ that has large values around 0, and $h > 0$ a "bandwidth" parameter to be selected (see analysis in Section 6.3.3). If we assume that $q$ is integrable with an integral equal to one, then $k(\cdot, x')$ is a probability density with mass around $x'$, which gets more concentrated as $h$ goes to zero. See the illustration below for the two typical kernel functions (sometimes called "windows").



Typical examples are:

- Box kernel: $q(x) \propto 1_{\|x\|_2 \leqslant 1}$, which leads to a weight functions $\hat{w}_i$ with many zeros. See below for an illustration in $d = 2$ dimensions.



- Gaussian kernel $q(x) \propto e^{-\|x\|^2/2}$, where we use the fact it is non-negative *pointwise*, as opposed to positive-definiteness in Chapter 7.[4] See a one-dimensional experiment in Figure 6.4.

In terms of algorithms, with a naive algorithm, for every test point, all the input data have to be considered, that is, a complexity proportional to $n$. The same techniques used for efficient $k$-nearest-neighbor search (e.g., k-d-trees) can also be applied here. Algorithms based on the fast Fourier transform can also be used (Silverman, 1982).

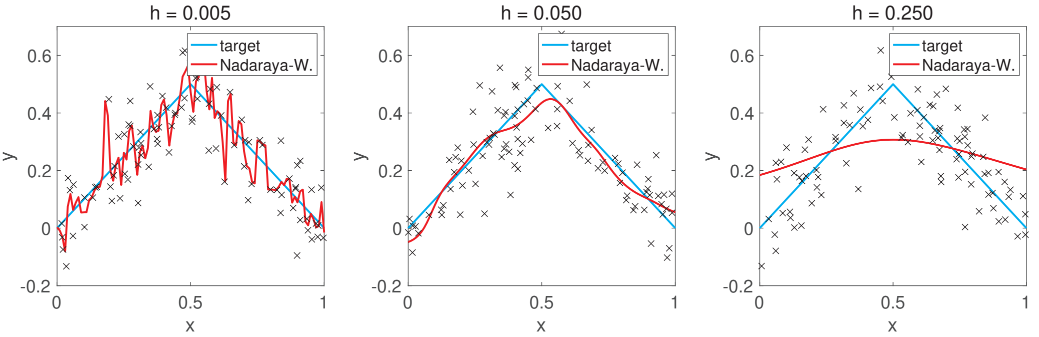


Figure 6.4: Nadaraya-Watson regression in $d = 1$ dimension, with three values of $h$ (the bandwidth), for the Gaussian kernel, with the same data as Figure 6.2. We can observe both underfitting ($h$ too large), or overfitting ($h$ too small).

## 6.3 Generic "simplest" consistency analysis

We consider for simplicity the regression case. For classification, convex surrogate techniques such as those used in Section 4.1 can be used, with, for example, the square loss or the logistic loss (with then a square root calibration function on top of the least-squares excess risk, see Exercise 6.2 below). Still, better rates can be obtained directly (see, e.g., Chen and Shah, 2018; Biau and Devroye, 2015; Audibert and Tsybakov, 2007; Chaudhuri and Dasgupta, 2014).

We make the following generic simplifying assumptions (weaker ones could be considered with more involved proofs):

(H-1) Bounded noise: There exists $\sigma \geqslant 0$ such that $(y - \mathbb{E}(y|x))^2 \leqslant \sigma^2$ almost surely. We could also consider a weaker assumption that the conditional variance $\mathbb{E}\big[(y - \mathbb{E}(y|x))^2|x\big]$ is bounded by $\sigma^2$ almost surely.

(H-2) Regular target function: The target function $f^*(x) = \mathbb{E}(y|x)$ is $B$-Lipschitz-continuous with respect to a distance $d$. For weaker assumptions, see Section 6.4.

We have, with the target function $f^*(x) = \mathbb{E}(y|x)$, at a test point $x \in \mathcal{X}$ (and using that

[4] See also https://francisbach.com/cursed-kernels/

the weights $\hat{w}_i(x)$ sum to one):

$$\begin{aligned}\hat{f}(x) - f^*(x) &= \sum_{i=1}^n y_i \hat{w}_i(x) - \mathbb{E}(y|x)\\ &= \sum_{i=1}^n \hat{w}_i(x)\big[y_i - \mathbb{E}(y_i|x_i)\big] + \sum_{i=1}^n \hat{w}_i(x)\big[\mathbb{E}(y_i|x_i) - \mathbb{E}(y|x)\big]\\ &= \sum_{i=1}^n \hat{w}_i(x)\big[y_i - \mathbb{E}(y_i|x_i)\big] + \sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big].\end{aligned}$$

Given $x_1, \dots, x_n$ (and *because we have assumed the weight functions do not depend on the labels*), the left term has zero expectation, while the right term is deterministic. We thus have, using the independence of all $(x_i, y_i)$, $i = 1, \dots, n$, and for $x$ fixed:

$$\begin{aligned}&\mathbb{E}\big[(\hat{f}(x) - f^*(x))^2 \big| x_1, \dots, x_n\big]\\ = \;& (\mathbb{E}(\hat{f}(x)\big|x_1, \dots, x_n) - f^*(x))^2 + \mathrm{var}\big[\hat{f}(x)\big|x_1, \dots, x_n\big]\\ = \;& \Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2 + \sum_{i=1}^n \hat{w}_i(x)^2 \mathbb{E}\big[\big(y_i - \mathbb{E}(y_i|x_i)\big)^2\big|x_i\big]\\ = \;& \qquad\qquad\text{bias} \qquad\qquad + \qquad \text{variance},\end{aligned}$$

with a "bias" term which is zero if $f^*$ is constant,[5] and a "variance" term which is zero, when $y$ is a deterministic function of $x$ (i.e., $\sigma = 0$). Note that at this point, we only had equalities in the argument; we can now upper-bound as:

$$\begin{aligned}&\mathbb{E}\big[(\hat{f}(x) - f^*(x))^2 \big| x_1, \dots, x_n\big]\\ \leqslant \;& \Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - f^*(x)\big|\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H-1)}, \qquad (6.2)\\ \leqslant \;& \Big[\sum_{i=1}^n \hat{w}_i(x) B d(x_i, x)\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H-2)},\\ \leqslant \;& B^2 \sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using Jensen's inequality.}\end{aligned}$$

Note that in the last inequality, having the weight vector $\hat{w}(x)$ in the simplex is crucial. We then have for the expected excess risk this generic bound we will use for all three cases (partitions, $k$-nn, and Nadaraya-Watson):

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leqslant B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2\Big] dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x). \qquad (6.3)$$

[5] What we call bias in this book is sometimes referred to as the squared bias.

⚠ The expectation is with respect to the training data. The expectation with respect to the testing point $x$ is kept as an integral to avoid confusion.

This upper bound can be divided into two terms:

- A variance term $\sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x)$, that depends on the noise on top of the optimal predictions. Since the weights sum to one, we can write $\sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] = \sum_{i=1}^n \mathbb{E}[(\hat{w}_i(x) - 1/n)^2] + 2/n - 1/n^2$, that is, up to vanishing constant, the variance term measures the deviation to uniform weights.
- A bias term $B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2\Big] dp(x)$, which depends on the regularity of the target function through the constant $B$. It will be small if the weight vectors $\hat{w}(x)$ put most of its mass on observations $x_i$ that are close to $x$.

This leads to two conditions: variance and bias have to go to zero when $n$ grows, corresponding to two simple expressions that depend on the weights. For the variance, the worst case scenario is that $\hat{w}_i(x)^2 \approx \hat{w}_i(x)$, that is, weights are putting all the mass into a single label (usually different for different testing point), thus leading to overfitting. For the bias, the worst-case scenario is that weights are uniform (leading to underfitting).

In the following, we will specialize it for $\mathcal{X}$ a subset of $\mathbb{R}^d$, with a distribution with a density with some minor regularity properties (all will have compact support, that is, $\mathcal{X}$ compact), where we show that a proper setting of the hyperparameters leads to "good" predictions. This will be done for all three cases of local averaging methods.

We look at universal consistency in Section 6.4, where we will list assumption (H-2).

**Exercise 6.2** *For the binary classification problem, with $\mathcal{Y} = \{-1, 1\}$, assume that $f^*(x) = \mathbb{E}(y|x)$ is $B$-Lipschitz-continuous. Using Section 4.1.4, show that the excess risk of the majority vote is upper-bounded by*

$$\bigg(B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2\Big] dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x)\bigg)^{1/2}.$$

### 6.3.1 Fixed partition

For the partitioning estimate defined in Section 6.2.2, we can prove the following convergence rate.

**Proposition 6.1 (Convergence rate for partition estimates)** *Assume bounded noise (H-1) and a Lipschitz-continuous target function (H-2), and a partition of the bounded support $\mathcal{X}$ of $p$, as $\mathcal{X} = \bigcup_{j \in J} A_j$; then for the partitioning estimate $\hat{f}$, we have:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leqslant \Big(8\sigma^2 + \frac{B^2}{2} \operatorname{diam}(\mathcal{X})^2\Big) \frac{|J|}{n} + B^2 \max_{j \in J} \operatorname{diam}(A_j)^2. \quad (6.4)$$

**Optimal trade-off between bias and variance.** Before we look at the proof (which is based on Eq. (6.3)), we can look at the consequence of the bound in Eq. (6.4). We need to balance the terms (up to constants) $\max_{j \in J} \operatorname{diam}(A_j)^2$ and $\frac{|J|}{n}$. In the simplest

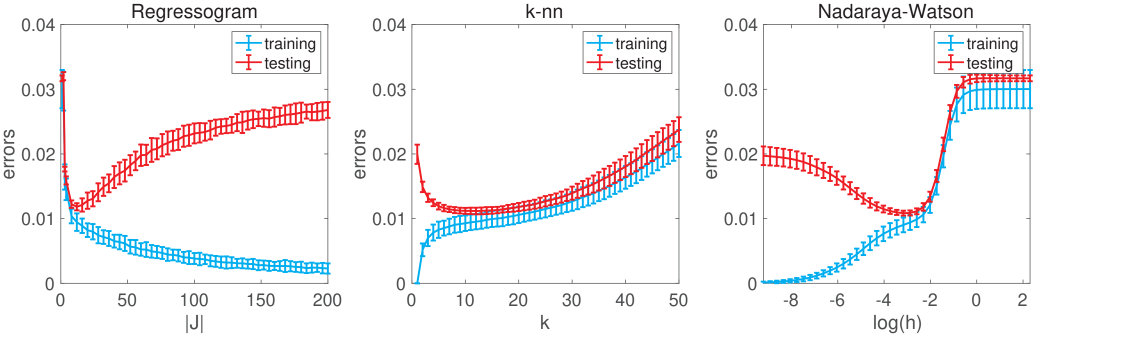


Figure 6.5: Learning curves for all three local averaging methods as a function of the corresponding hyperparameter. Left: regressogram (hyperparameter = number $|J|$ of sets in the partition), middle: $k$-nearest-neighbors (hyperparameter = number of neighbors $k$), right: Nadaraya-Watson (hyperparameter = bandwidth $h$). In all three cases, we see a trade-off between under-fitting and over-fitting.

situation of the unit-cube $[0,1]^d$, with $|J| = h^{-d}$ cubes of length $h$, we get $\frac{|J|}{n} = \frac{1}{nh^d}$ and $\max_{j\in J} \operatorname{diam}(A_j)^2 = h^2$, which, with $h = n^{-1/(2+d)}$ to make them equal, leads to a rate proportional to $n^{-2/(2+d)}$. As shown by Györfi et al. (2006), this rate is optimal for the estimation of Lipschitz-continuous functions (see Chapter 15).

While optimal, this is a very slow rate and a typical example of the curse of dimensionality. For this rate to be small, $n$ has to be exponentially large in dimension. This is unavoidable with so little regularity (only bounded first-order derivatives). In Chapter 7 (and also in Section 6.5), we show how to leverage the smoothness of the target function to get significantly improved bounds (local averaging cannot take strong advantage of such smoothness). In Chapter 8, we will leverage dependence on a small number of variables.

**Experiments.** For the problem shown in Section 6.2, we plot in Figure 6.5 (left plot) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of $|J|$.

**Proof of Proposition 6.1** (♦) We consider an element $A_j$ of the partition with at least one observation in it (a non-empty cell). Then for $x \in A_j$, and $i$ among the indices of the points lying in $A_j$, $\hat{w}_i(x) = 1/n_{A_j}$ where $n_{A_j} \in \{1, \dots, n\}$ is the number of data points lying in $A_j$.

**Variance.** From Eq. (6.3), the variance term is bounded from above by $\sigma^2$ times

$$\sum_{i=1}^n \hat{w}_i(x)^2 = n_{A_j} \frac{1}{n_{A_j}^2} = \frac{1}{n_{A_j}}.$$

If $A_j$ contains no input observations, then all weights are equal to $1/n$, and this sum is equal to $n \times \frac{1}{n^2} = 1/n$ for all $x \in A_j$. Thus, we get

$$\begin{aligned}\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] dp(x) &= \int_{\mathcal{X}} \sum_{j\in J} 1_{x\in A_j} \mathbb{E}\Big[\frac{1}{n_{A_j}} 1_{n_{A_j}>0} + \frac{1}{n} 1_{n_{A_j}=0}\Big] dp(x) \\ &= \sum_{j\in J} \mathbb{P}(A_j) \cdot \mathbb{E}\Big[\frac{1}{n_{A_j}} 1_{n_{A_j}>0} + \frac{1}{n} 1_{n_{A_j}=0}\Big].\end{aligned}$$

Intuitively, by the law of large numbers, $n_{A_j}/n$ tends to $\mathbb{P}(A_j)$, so the variance term is expected to be of the order $\sigma^2 \sum_{j\in J} \mathbb{P}(A_j) \frac{1}{n\mathbb{P}(A_j)} = \sigma^2 \frac{|J|}{n}$, which is to be expected as this is essentially equivalent to least-squares regression with $|J|$ features $(1_{x\in A_j})_{j\in J}$.

More formally, we have $\mathbb{P}(n_{A_j} = 0) = (1 - \mathbb{P}(A_j))^n$, and, using Bernstein's inequality (see Section 1.2.3) for the random variables $1_{x_i\in A_j}$, which have mean and variance upper bounded by $\mathbb{P}(A_j)$, we have: $\mathbb{P}\big(\frac{n_{A_j}}{n} \leqslant \frac{1}{2}\mathbb{P}(A_j)\big) = \mathbb{P}\big(\frac{n_{A_j}}{n} \leqslant \mathbb{P}(A_j) - \frac{1}{2}\mathbb{P}(A_j)\big) \leqslant \exp\big(-\frac{n\mathbb{P}(A_j)^2/4}{2\mathbb{P}(A_j)+2(\mathbb{P}(A_j)/2)/3}\big) \leqslant \exp(-n\mathbb{P}(A_j)/10) \leqslant \frac{5}{n\mathbb{P}(A_j)}$, where we have used $\alpha e^{-\alpha} \leqslant 1/2$ for any $\alpha \geqslant 0$. This leads to the bound

$$\begin{aligned}\sum_{j\in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{1_{n_{A_j}>0}}{n_{A_j}} + \frac{1_{n_{A_j}=0}}{n}\Big] &\leqslant \sum_{j\in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{1_{1\leqslant n_{A_j} \leqslant \frac{n}{2}\mathbb{P}(A_j)}}{n_{A_j}} + \frac{1_{n_{A_j} > \frac{n}{2}\mathbb{P}(A_j)}}{n_{A_j}} + \frac{1_{n_{A_j}=0}}{n}\Big] \\ &\leqslant \sum_{j\in J} \mathbb{P}(A_j) \Big[\mathbb{P}\Big(\frac{n_{A_j}}{n} \leqslant \frac{\mathbb{P}(A_j)}{2}\Big) + \frac{2}{n\mathbb{P}(A_j)} + \frac{1}{n}\mathbb{P}(n_{A_j}=0)\Big] \\ &\leqslant \sum_{j\in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{5}{n\mathbb{P}(A_j)} + \frac{2}{n\mathbb{P}(A_j)} + \frac{1}{n\mathbb{P}(A_j)}\Big] \leqslant \frac{8|J|}{n}.\end{aligned}$$

**Bias.** We have, for $x \in A_j$ and a non-empty cell,

$$\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \leqslant \mathrm{diam}(A_j)^2,$$

with $\sum_{i=1}^n \hat{w}_i(x) d(x,x_i)^2 = \frac{1}{n}\sum_{i=1}^n d(x,x_i)^2 \leqslant \mathrm{diam}(\mathcal{X})^2$ for empty-cells. Thus, separating the cases $n_{A_j} = 0$ and $n_{A_j} > 0$:

$$\begin{aligned}\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x,x_i)^2\Big] dp(x) &\leqslant \sum_{j\in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\mathrm{diam}(A_j)^2 1_{n_{A_j}>0} + 1_{n_{A_j}=0} \mathrm{diam}(\mathcal{X})^2\Big] \\ &\leqslant \sum_{j\in J} \mathbb{P}(A_j) \Big[\mathrm{diam}(A_j)^2 + (1-\mathbb{P}(A_j))^n \mathrm{diam}(\mathcal{X})^2\Big] \\ &= \sum_{j\in J} \mathbb{P}(A_j) \mathrm{diam}(A_j)^2 + \sum_{j\in J} \mathbb{P}(A_j)(1-\mathbb{P}(A_j))^n \cdot \mathrm{diam}(\mathcal{X})^2.\end{aligned}$$

Using that $u(1-u)^n \leqslant 1/(2n)$ for $u \geqslant 0$, we get

$$\int_{\mathcal{X}} \mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x) d(x,x_i)^2\big] dp(x) \leqslant \sum_{j \in J} \mathbb{P}(A_j) \mathrm{diam}(A_j)^2 + \frac{1}{2}\frac{|J|}{n} \times \mathrm{diam}(\mathcal{X})^2,$$

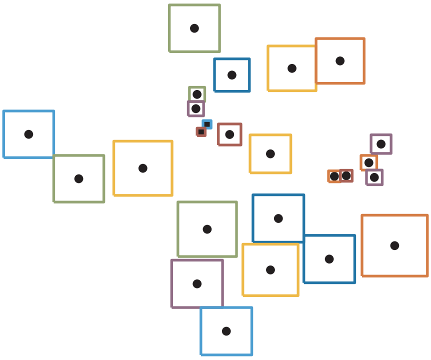
which leads to the desired term. ■

### 6.3.2 $k$-nearest neighbor

Here, since all weights are equal to zero except $k$ of them, which are equal to $\frac{1}{k}$, we have $\sum_{i=1}^n \hat{w}_i(x)^2 = \frac{1}{k}$, so the variance term will go down as soon as $k$ tends to infinity. For the bias term, the needed term $\sum_{i=1}^n \hat{w}_i(x) d(x_i,x)^2$ is equal to the average of the squared distances between $x$ and its $k$-nearest neighbors within $\{x_1,\dots,x_n\}$, and this is less than the expected distance to the $k$-th nearest neighbor $x_{i_k(x)}$, for which the two following lemmas, taken from Biau and Devroye (2015, Theorem 2.4), give an estimate for the $\ell_\infty$-distance, and thus for all distances by equivalence of norms on $\mathbb{R}^d$.

**Lemma 6.1 (distance to nearest neighbor)** *Consider a probability distribution with compact support in $\mathcal{X} \subset \mathbb{R}^d$. Consider $n+1$ points $x_1,\dots,x_n,x_{n+1}$ sampled i.i.d. from $\mathcal{X}$. Then the expected squared $\ell_\infty$-distance between $x_{n+1}$ and its first-nearest-neighbor is less than $4\frac{\mathrm{diam}(\mathcal{X})^2}{n^{2/d}}$ for $d \geqslant 2$, and less than $\frac{2}{n}\mathrm{diam}(\mathcal{X})^2$ for $d=1$.*

**Proof** We denote by $x_{(i)}$ a nearest neighbor of $x_i$ among the other $n$ points. Since all $n+1$ points are i.i.d., we can permute the indices without changing the distributions, and all $\|x_i - x_{(i)}\|_\infty^2$ have the same distribution as $\|x_{n+1} - x_{(n+1)}\|_\infty^2$; thus, we can instead compute $\frac{1}{n+1}\sum_{i=1}^{n+1} \mathbb{E}\big[\|x_i - x_{(i)}\|_\infty^2\big]$. We denote by $R_i = \|x_i - x_{(i)}\|_\infty$ and assume for simplicity $R_i > 0$ for all $i$ (the general case is left as an exercise). Then the sets $B_i = \{x \in \mathbb{R}^d, \ \|x - x_i\|_\infty < \frac{R_i}{2}\}$ are disjoint since for $i \neq j$, $\|x_i - x_j\|_\infty \geqslant \max\{R_i, R_j\}$. See the illustration below in two dimensions, with squares representing the sets $B_i$ centered as $x_i$ (represented as dots).



Moreover, their union has diameter less than $\mathrm{diam}(\mathcal{X}) + \mathrm{diam}(\mathcal{X}) = 2\mathrm{diam}(\mathcal{X})$. Thus, the volume of the union of all sets $B_i$, which is equal to the sum of their volumes, is less than $\big(2\mathrm{diam}(\mathcal{X})\big)^d$. Thus, we have: $\sum_{i=1}^{n+1} R_i^d \leqslant \big(2\mathrm{diam}(\mathcal{X})\big)^d$. Therefore, by Jensen's inequality, for $d \geqslant 2$,

$$\Big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i^2\Big)^{d/2} \leqslant \frac{1}{n+1}\sum_{i=1}^{n+1}(R_i)^d \leqslant \frac{2^d \mathrm{diam}(\mathcal{X})^d}{n+1},$$

leading to the desired result. For $d = 1$, we have $\big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i^2\big) \leqslant \mathrm{diam}(\mathcal{X})\big(\frac{1}{n+1}\sum_{i=1}^{n+1} R_i\big)$, which is less than $\frac{2}{n+1}\mathrm{diam}(\mathcal{X})^2$. ■

**Lemma 6.2 (distance to $k$-nearest-neighbor)** *Let $k \geqslant 1$. Consider a probability distribution with compact support in $\mathcal{X} \subset \mathbb{R}^d$. Consider $n+1$ points $x_1, \dots, x_n, x_{n+1}$ sampled i.i.d. from $\mathcal{X}$. Then the expected squared $\ell_\infty$-distance between $x_{n+1}$ and its $k$-nearest-neighbor is less than $8\mathrm{diam}(\mathcal{X})^2\big(\frac{2k}{n}\big)^{2/d}$ for $d \geqslant 2$, and less than $\frac{8k}{n}\mathrm{diam}(\mathcal{X})^2$ for $d = 1$.*

**Proof** (♦) Without loss of generality, we assume $2k \leqslant n$ (otherwise, the proposed bounds are trivial). We can then divide randomly (and independently) the $n$ first points into $2k$ sets of size approximately $\frac{n}{2k}$. We denote $x_{(k)}^j$ a 1-nearest neighbor of $x_{n+1}$ within the $j$-th set. The squared distance from $x_{n+1}$ to the $k$-nearest neighbor among all first $n$ points is less than the $k$-th smallest of the distances $\|x_{n+1} - x_{(k)}^j\|_\infty^2$, $j \in \{1, \dots, 2k\}$, because we take a $k$-nearest neighbor over a smaller set. This $k$-th smallest distance is less than $\frac{1}{k}\sum_{j=1}^{2k}\|x_{n+1} - x_{(k)}^j\|_\infty^2$ (this is a general fact that the $k$-smallest element among non-negative $p$ elements, is less than their sum divided by $p-k$, applied here for $p = k$).

Thus, using the lemma above on the 1-nearest-neighbor from $\frac{n}{2k}$ points, we get that the desired averaged distance is less than, for $d \geqslant 2$:

$$\frac{1}{k}\sum_{j=1}^{2k} 4\frac{\mathrm{diam}(\mathcal{X})^2}{(\frac{n}{2k})^{2/d}} = 8\frac{\mathrm{diam}(\mathcal{X})^2}{n^{2/d}}(2k)^{2/d}.$$

A similar argument can be extended to $d = 1$ (left as an exercise). ■

Putting things together, we get the following result for the consistency of $k$-nearest-neighbors.

**Proposition 6.2 (Convergence rate for $k$-nearest-neighbors)** *Assume bounded noise (H-1) and a Lipschitz-continuous target function (H-2) with an input distribution with bounded support $\mathcal{X}$. Then for the $k$-nearest-neighbor estimate $\hat{f}$ with the $\ell_\infty$-norm, we have, for $d \geqslant 2$:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2]dp(x) \leqslant \frac{\sigma^2}{k} + 8B^2\mathrm{diam}(\mathcal{X})^2\Big(\frac{2k}{n}\Big)^{2/d}. \tag{6.5}$$

Balancing the two terms above is obtained with $k \propto n^{2/(2+d)}$, and we get the same result as for the other local averaging schemes. See more details by Chen and Shah (2018) and Biau and Devroye (2015).

**Exercise 6.3** *Show that if the Bayes rate is* $0$ *(that is,* $\sigma = 0$*), then* $1$*-nearest-neighbor is consistent.*

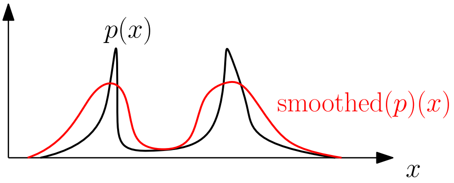
**Experiments.** For the problem shown in Section 6.2, we plot in Figure 6.5 (middle) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of $k$.

### 6.3.3 Kernel regression (Nadaraya-Watson) (♦)

In this section, we assume that $\mathcal{X} = \mathbb{R}^d$, and for simplicity, we assume that the distribution of the inputs has a density (also denoted $p$) with respect to the Lebesgue measure. We also assume that the kernel is of the form $k(x,x') = q_h(x-x') = h^{-d} q(\frac{1}{h}(x-x'))$ for a probability density $q : \mathbb{R}^d \to \mathbb{R}_+$. The function $q_h$ is also a density, which is the density of $hz$ when $z$ has density $q(z)$ (it thus gets more concentrated around 0 as $h$ tends to zero). With these notations, the weights can be written:

$$\hat{w}_i(x) = \frac{q_h(x-x_i)}{\sum_{j=1}^n q_h(x-x_j)}.$$

**Smoothing by convolution.** When performing kernel regression, quantities of the form $\frac{1}{n}\sum_{i=1}^n q_h(x-x_i)g(x_i)$ naturally appear. When the number $n$ of observations goes to infinity, and $x$ is fixed, by the law of large numbers, it tends almost surely to $\int_{\mathbb{R}^d} q_h(x-z)g(z)p(z)dz$, which is exactly the convolution between the function $q_h$ and the function $x \mapsto p(x)g(x)$, which we can denote $[(pg) * q_h](x)$. The function $q_h$ is a probability density that puts most of its weights at a range of values of order $h$, e.g., for kernels like the Gaussian kernel or the box kernel. Thus, convolution will smooth the function $pg$ by averaging values at range $h$. Therefore, when $h$ goes to zero, it converges to the function $pg$ itself. See an example below for $g = 1$.



Note that for this limit to hold, we need to ensure the factors in $n$ and $h^d$ are present.

We can now look at the generalization bound from Eq. (6.3) and see how it applies to kernel regression. We now consider the $\ell_2$-distance for simplicity and consider the variance and bias terms separately, first with an asymptotic informal result where both $h$ tends to zero and $n$ tends to infinity, and then a formal result.

**Variance term.** We have, for a fixed $x \in \mathcal{X}$:

$$n \sum_{i=1}^n \hat{w}_i(x)^2 \quad = \quad \frac{\frac{1}{n}\sum_{i=1}^n q_h(x-x_i)^2}{\big(\frac{1}{n}\sum_{i=1}^n q_h(x-x_i)\big)^2}.$$

Using the law of large numbers and the smoothing reasoning above, this sum $n\sum_{i=1}^n \hat{w}_i(x)^2$ is converging almost surely to

$$\frac{\int_{\mathbb{R}^d} q_h(x-z)^2 p(z)dz}{\big(\int_{\mathbb{R}^d} q_h(x-z)p(z)dz\big)^2} = \frac{[q_h^2 * p](x)}{[q_h * p](x)^2}.$$

When $h$ tends to zero, then the denominator above $[q_h * p](x)^2$ tends to $p(x)^2$ because the bandwidth of the smoothing goes to zero. The numerator above corresponds, up to a multiplicative constant, to the smoothing of $p$ by the density $x \mapsto \frac{q_h(x)^2}{\int_{\mathbb{R}^d} q_h(u)^2 du}$, and is thus asymptotically equivalent to $p(x)\int_{\mathbb{R}^d} q_h(u)^2 du = p(x)h^{-d}\int_{\mathbb{R}^d} q(u)^2 du$.

Overall, when $n$ tends to infinity, and $h$ tends to zero, we get, asymptotically for $x$ fixed:

$$\sum_{i=1}^n \hat{w}_i(x)^2 \sim \frac{1}{nh^d}\frac{1}{p(x)}\int_{\mathbb{R}^d} q(u)^2 du,$$

and thus, still asymptotically,

$$\int_{\mathcal{X}} \Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] p(x)dx \sim \frac{1}{nh^d}\text{vol}(\text{supp}(p))\int_{\mathbb{R}^d} q(u)^2 du,$$

where vol(supp($p$)) is the volume of the support of $p$ in $\mathbb{R}^d$ (the closure of all $x$ for which $p(x) > 0$), which we assume bounded.

**Bias.** With the same intuitive reasoning, we get when $n$ tends to infinity:

$$\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \quad \to \quad \frac{\int_{\mathbb{R}^d} q_h(x-z)\|x-z\|_2^2 p(z)dz}{\int_{\mathbb{R}^d} q_h(x-z)p(z)dz}.$$

The denominator has the same shape as for the variance term and tends to $p(x)$ when $h$ tends to zero. With the change of variable $u = \frac{1}{h}(x-z)$, the numerator is equal to $\int_{\mathbb{R}^d} q_h(x-z)\|x-z\|_2^2 p(z)dz = h^2\int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du$, which is equivalent to $h^2 p(x)\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du$ when $h$ tends to zero. Overall, when $n$ tends to infinity, and $h$ tends to zero, we get:

$$\int_{\mathcal{X}} \Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2\Big] p(x)dx \sim h^2 \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.$$

Therefore, overall we get an *asymptotic* bound proportional to (up to constants depending on $q$):

$$\frac{\sigma^2}{nh^d} + B^2 h^2,$$

leading to the same upper bound as for partitioning estimates by setting $h \propto n^{-1/(d+2)}$.

**Formal reasoning (♦♦).** We can make the informal reasoning above more formal using concentration inequalities, leading to non-asymptotic bounds of the same nature (simply more complicated) that make explicit the joint dependence on $n$ and $h$. We will prove the following result:

**Proposition 6.3 (Convergence rate for Nadaraya-Watson estimation)** *Assume bounded noise (H-1) and a Lipschitz-continuous target function (H-2), and a function $q : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} q(z)dz = 1$, and $\|q\|_\infty = \sup_{z\in\mathbb{R}^d} q(z)$ is finite. Moreover, assume that $p$ has bounded support $\mathcal{X}$ and density upper-bounded by $\|p\|_\infty$. Then for the Nadaraya-Watson estimate $\hat{f}$, we have:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2]dp(x) \leqslant \bigg[\frac{8\|q\|_\infty}{nh^d}\big(1 + \frac{1}{2}\mathrm{diam}(\mathcal{X})^2\big) + 2Bh^2\|p\|_\infty c\bigg] \cdot C_h, \qquad (6.6)$$

*where $c = \int_{\mathbb{R}^d} q(u)\|u\|_2^2du$ and $C_h = \int_{\mathcal{X}} \frac{p(x)}{[q_h * p](x)}dx$.*

With additional assumptions, we can show that the constant $C_h$ remains bounded when $h$ tends to zero (see exercise below). Before giving the proof, we note that the optimal bandwidth parameter is indeed proportional to $h \propto n^{-1/(d+2)}$, with an overall excess risk proportional to $n^{-2/(d+2)}$, like the two other types of estimators.

**Proof of Proposition 6.3** (♦) As for the proof for partitioning estimates, to deal with the denominator in the definition of the weights, we can first use Bernstein's inequality (see Section 1.2.3), applied to the random variables $q_h(x - x_i)$ which is almost surely in $[0, h^{-d}\|q\|_\infty]$, to bound

$$\mathbb{P}\Big(\frac{1}{n}\sum_{i=1}^n q_h(x - x_i) \leqslant \mathbb{E}[q_h(x - z)] - \varepsilon\Big) \leqslant \exp\Big(-\frac{n\varepsilon^2}{2\mathbb{E}[q_h^2(x - z)] + 2\|q\|_\infty h^{-d}\varepsilon/3}\Big).$$

We get with $\varepsilon = \frac{1}{2}\mathbb{E}[q_h(x - z)]$, using $\mathbb{E}[q_h^2(x - z)] \leqslant \|q\|_\infty h^{-d}\mathbb{E}[q_h(x - z)]$, for the event $\mathcal{A}(x) = \{\frac{1}{n}\sum_{i=1}^n q_h(x - x_i) \leqslant \frac{1}{2}\mathbb{E}[q_h(x - z)]\}$:

$$\begin{aligned}\mathbb{P}\big(\mathcal{A}(x)\big) \leqslant\ & \exp\Big(-\frac{\frac{n}{4}(\mathbb{E}[q_h(x - z)])^2}{2\mathbb{E}[q_h^2(x - z)] + \mathbb{E}[q_h(x - z)]h^{-d}\|q\|_\infty/3}\Big)\\ \leqslant\ & \exp\Big(-\frac{\frac{n}{4}\mathbb{E}[q_h(x - z)]}{(7/3)h^{-d}\|q\|_\infty}\Big) \leqslant \frac{\|q\|_\infty}{nh^d\mathbb{E}[q_h(x - z)]}\cdot\frac{1}{e}\frac{28}{3} \leqslant \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x - z)]}, \quad (6.7)\end{aligned}$$

where we have used $\alpha e^{-\alpha} \leqslant 1/e$ for $\alpha \geqslant 0$. We can now bound bias and variance.

**Variance.** For a fixed $x \in \mathcal{X}$, we get

$$\begin{aligned}\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] &= \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)^2\Big]\\ &\leqslant \mathbb{P}(\mathcal{A}(x)) + \frac{4}{\big(n\mathbb{E}[q_h(x - z)]\big)^2}\mathbb{E}\Big[\sum_{i=1}^n q\big(\frac{1}{h}(x - x_i)\big)^2\Big]\\ &\leqslant \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x - z)]} + \frac{4\mathbb{E}\big[q_h(x - z)^2\big]}{n\big[\mathbb{E}q_h(x - z)\big]^2} \leqslant \frac{8\|q\|_\infty}{nh^d\mathbb{E}[q_h(x - z)]}.\end{aligned}$$

Moreover, we have $\mathbb{E}[q_h(x-z)] = \int_{\mathbb{R}^d} dp(x-hu)q(u)du = [p*q_h](x)$. This leads to an overall bound on the variance term as $\displaystyle\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big]p(x)dx \leqslant \frac{8\|q\|_\infty}{nh^d}\int_{\mathcal{X}}\frac{p(x)}{[p*q_h](x)}dx$.

**Bias term.** We have a similar reasoning for the bias term. Indeed, we get for a given $x \in \mathcal{X}$, using the bound in Eq. (6.7):

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] \\
&= \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] \\
&\leqslant \mathbb{P}(\mathcal{A}(x))\cdot \mathrm{diam}(\mathcal{X})^2 + \frac{2}{n\mathbb{E}[q_h(x-z)]}\cdot n\mathbb{E}[q_h(x-z)\|x-z\|_2^2] \\
&\leqslant \frac{4\|q\|_\infty}{nh^d[q_h*p](x)}\cdot \mathrm{diam}(\mathcal{X})^2 + \frac{2h^2}{[q_h*p](x)}\cdot\int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du \\
&\leqslant \frac{4\|q\|_\infty}{nh^d[q_h*p](x)}\cdot \mathrm{diam}(\mathcal{X})^2 + \frac{2h^2\|p\|_\infty}{[q_h*p](x)}\cdot\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.
\end{aligned}$$

This leads to an overall bound on the bias term equal to $\displaystyle\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big]p(x)dx \leqslant \int_{\mathcal{X}}\frac{p(x)}{[q_h*p](x)}dx\cdot\Big[\frac{4\|q\|_\infty}{nh^d}\mathrm{diam}(\mathcal{X})^2 + 2h^2\|p\|_\infty\Big(\int_{\mathbb{R}^d} q(u)\|u\|_2^2du\Big)\Big]$.

Putting things together, we get that the excess risk $\int_{\mathcal{X}}\mathbb{E}[(\hat{f}(x)-f^*(x))^2]dp(x)$ is less than

$$\Big[\frac{8\|q\|_\infty}{nh^d}\big(1+\frac{1}{2}\mathrm{diam}(\mathcal{X})^2\big) + 2Bh^2\|p\|_\infty\Big(\int_{\mathbb{R}^d} q(u)\|u\|_2^2du\Big)\Big]\cdot\int_{\mathcal{X}}\frac{p(x)}{[q_h*p](x)}dx,$$

which is exactly the desired bound. ■

**Exercise 6.4** *Assume that the support $\mathcal{X}$ of the density $p$ of inputs is bounded and that $p$ is strictly positive and continuously differentiable on $\mathcal{X}$. Show that for $h$ small enough (with an explicit upper-bound), then $C_h = \int_{\mathcal{X}}\frac{p(x)}{[q_h*p](x)}dx \leqslant \frac{1}{2}\mathrm{vol}(\mathcal{X})$.*

**Experiments.** For the problem shown in Section 6.2, we plot in Figure 6.5 (right) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of $h$.

## 6.4 Universal consistency (♦)

Above, we have required the following conditions on the weights:

- $\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x) d(x_i,x)^2\Big] dp(x) \to 0$ when $n$ tends to infinity, to ensure that the bias goes to zero.
- $\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] dp(x) \to 0$ when $n$ tends to infinity, to ensure that the variance goes to zero.

This was enough to show consistency when the target function is Lipschitz-continuous in $\mathbb{R}^d$. This also led to a precise rate of convergence, which turns out to be optimal for learning with target functions that are Lipschitz-continuous and for which the curse of dimensionality cannot be avoided (see Chapter 15).

To show universal consistency, that is, consistency for any square-integrable functions, we need an extra (technical) assumption, which was first outlined in Stone's theorem (Stone, 1977), namely that there exists $c > 0$ such that for any non-negative integrable function $h : \mathcal{X} \to \mathbb{R}$, then

$$\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E}\big[\hat{w}_i(x) h(x_i)\big] dp(x) \leqslant c \cdot \int_{\mathcal{X}} h(x) dp(x). \tag{6.8}$$

Below, $h$ will be the squared deviation between two functions.

⚠ Above, we only take the expectation with respect to the training data, while we use the integral notation to take the expectation with respect to the training distribution.

Then for any $\varepsilon > 0$, and for any target function $f^* \in L_2(dp(x))$, we can find a function $g$ which is $B(\varepsilon)$-Lipschitz-continuous and such that $\|f^* - g\|_{L_2(dp(x))} \leqslant \varepsilon$, because the set of Lipschitz-continuous functions is dense in $L_2(dp(x))$ (see, e.g., Ambrosio et al., 2013)).

Then we have, for a given $x \in \mathcal{X}$:

$$\begin{aligned}
& \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2\Big) \\
\leqslant \quad & \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\Big(\big|f^*(x_i) - g(x_i)\big| + \big|g(x_i) - g(x)\big| + \big|g(x) - f^*(x)\big|\Big]^2\Big) \\
\leqslant \quad & 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - g(x_i)\big|\Big]^2\Big) + 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|g(x_i) - g(x)\big|\Big]^2\Big) \\
& +3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|g(x) - f^*(x)\big|\Big]^2\Big) \text{ using the inequality } (a+b+c)^2 \leqslant 3a^2+3b^2+]!3c^2, \\
\leqslant \quad & 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - g(x_i)\big|\Big]^2\Big) + 3\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x) B(\varepsilon) d(x,x_i)\Big]^2\Big) \\
& +3\mathbb{E}\Big(\big|g(x) - f^*(x)\big|^2\Big) \text{ since weights sum to one, and } g \text{ is Lipschitz-continuous.}
\end{aligned}$$

We can further upper-bound $\mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2\Big)$ by

$$\begin{aligned}
& 3\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - g(x_i)\big|^2\Big] + 3B(\varepsilon)^2\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big) \\
& +3\mathbb{E}\Big(\big|g(x) - f^*(x)\big|^2\Big) \text{ using Jensen's inequality on the second term,} \\
\leqslant \;\; & 3c\cdot\mathbb{E}\big[\big|f^*(x) - g(x)\big|^2\big] + 3B(\varepsilon)^2\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big) + 3\mathbb{E}\Big(\big|g(x) - f^*(x)\big|^2\Big),
\end{aligned}$$

using Eq. (6.8). We can now integrate with respect to $x$ to get

$$\begin{aligned}
& \int_{\mathcal{X}} \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]\Big]^2\Big)dp(x) \\
& \qquad \leqslant 3c\cdot\varepsilon^2 + 3B(\varepsilon)^2\int_{\mathcal{X}}\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big)dp(x) + 3\varepsilon^2. \qquad (6.9)
\end{aligned}$$

**Proving universal consistency.** We can then combine the bound above (which gives a bound on the bias) with Eq. (6.2), starting from the excess risk $\int_{\mathcal{X}}\mathbb{E}\big[(\hat{f}(x)-f^*(x))^2\big]dp(x)$ less than

$$\int_{\mathcal{X}} \mathbb{E}\Big(\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f^*(x_i) - f^*(x)\big|\Big]^2\Big)dp(x) + \sigma^2\int_{\mathcal{X}}\mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x)^2\big]dp(x),$$

which is the sum of a bias term and a variance term, and for which, together with Eq. (6.9), we can use the same tools for consistency as for Eq. (6.3).

To prove universal consistency, we fix a certain $\varepsilon > 0$, from which we obtain some Lipschitz constant $B(\varepsilon)$. For such a $B(\varepsilon)$, we know how to make the (squared) bias term $B(\varepsilon)^2\int_{\mathcal{X}}\mathbb{E}\Big(\sum_{i=1}^n \hat{w}_i(x)d(x,x_i)^2\Big)dp(x) + \sigma^2\int_{\mathcal{X}}\mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x)^2\big]dp(x)$ less than $\varepsilon$, by choosing appropriate hyperparameter and number of observations $n$ (see previous sections). Thus, if the extra condition in Eq. (6.8) is satisfied, these three methods are universally consistent. Note that, in general, $n$ has to grow unbounded when $\varepsilon$ tends to zero without any a priori bound.

We can now look at the three cases:

- Partitioning: We have then $c = 2$, and we get universal consistency. Indeed, using the same notations as in Section 6.2.2, we have for any fixed $x \in A_j$, $j \in J$, and $f$ a non-negative function:

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E}\big[\hat{w}_i(x)f(x_i)\big] &= \mathbb{E}\Big[1_{n_{A_j}>0}\frac{1}{n_{A_j}}\sum_{i \text{ s.t. } x_i\in A_j} f(x_i) \; + 1_{n_{A_j}=0}\frac{1}{n}\sum_{i=1}^n f(x_i)\Big] \\
&= \mathbb{E}\Big[1_{n_{A_j}>0}\frac{1}{n_{A_j}}\sum_{i \text{ s.t. } x_i\in A_j} \mathbb{E}[f(x_i)|x_i \in A_j] + 1_{n_{A_j}=0}\frac{1}{n}\sum_{i=1}^n f(x_i)\Big] \\
&\leqslant \mathbb{E}[f(z)|z \in A_j] \; + \mathbb{E}[f(z)],
\end{aligned}$$

where $z$ is distributed as $x$. Thus, integrating with respect to $x$ and summing over $j \in J$, we get:

$$\int_{\mathcal{X}} \sum_{i=1}^{n} \mathbb{E}\big[\hat{w}_i(x) h(x_i)\big]\, dp(x) \leqslant \sum_{j \in J} \Big( \mathbb{P}(A_j)\mathbb{E}[f(z) | z \in A_j] + \mathbb{P}(A_j) \cdot \mathbb{E}[f(z)] \Big) = 2\mathbb{E}[f(z)],$$

which is exactly Eq. (6.8) with $c = 2$.

- Kernel regression: it can be shown using the same type of techniques outlined for consistency for Lipschitz-continuous functions.
- $k$-nearest neighbor: the condition in Eq. (6.8) is not easy to show and is often referred to as Stone's lemma. See Biau and Devroye (2015, Lemma 10.7).

## 6.5 Adaptivity (♦♦)

As shown above, all local averaging techniques achieve the same performance on Lipschitz-continuous functions, which is an unavoidable bad performance when $d$ grows (curse of dimensionality). One extra order of smoothness, that is, on $\mathbb{R}^d$, two bounded derivatives, can be leveraged to lead to a convergence rate proportional to $n^{-4/(4+d)}$ (Wasserman, 2006, Section 5.4). However, the higher smoothness of the target function does not seem to be easy to leverage, that is, even if the target function is very smooth, the local averaging techniques will not be able to attain better convergence rates. The impossibility comes from the bias term which is the square of $\sum_{i=1}^n \hat{w}_i(x)\big[f^*(x_i) - f^*(x)\big]$ in Section 6.3: when $f^*$ is once differentiable, $f^*(x_i) - f^*(x) = O(\|x_i - x\|)$ and this is what we leveraged in the proofs; when $f^*$ is twice differentiable, by a Taylor expansion, $f^*(x_i) - f^*(x) = (x_i - x)^\top (f^*)'(x_i) + O(\|x_i - x\|^2)$, and we can choose weights so that $\sum_{i=1}^n \hat{w}_i(x)(x - x_i) = O(\|x - x_i\|^2)$ (this is possible because the components of $x - x_i$ may take positive and negative values, leading to potential cancellations, see exercise below); but when $f$ is three-times differentiable or more, obtaining a term $O(\|x_i - x\|^3)$ that would come from a Taylor expansion, is only possible if the weights satisfy $\sum_{i=1}^n \hat{w}_i(x)(x - x_i)(x - x_i)^\top = O(\|x_i - x\|^3)$, which is not possible when the weights are non-negative as no cancellations are possible.

Positive-definite kernel methods will provide simple ways in Chapter 7, as well as neural networks in Chapter 9, to leverage smoothness. Among local averaging techniques, there are, however, ways to do it. For example, using locally linear regression, where one solves for any test point $x$,

$$\inf_{\beta_1 \in \mathbb{R}^d,\ \beta_0 \in \mathbb{R}} \sum_{i=1}^{n} \hat{w}_i(x)(y_i - \beta_1^\top x_i - \beta_0)^2.$$

(note that the regular regressogram corresponds to setting $\beta_1 = 0$ above). In other words we solve

$$\inf_{\beta_1 \in \mathbb{R}^d,\ \beta_0 \in \mathbb{R}} \int_{\mathcal{Y}} (y - \beta_1^\top x - \beta_0)^2 d\widehat{p}(y|x).$$

The running time is now $O(nd^2)$ per testing point as we have to solve a linear least-squares (see Chapter 3), but the performance, both empirical and theoretical (Tsybakov, 2008), improves. See an example with the regressogram weights in Figure 6.6.

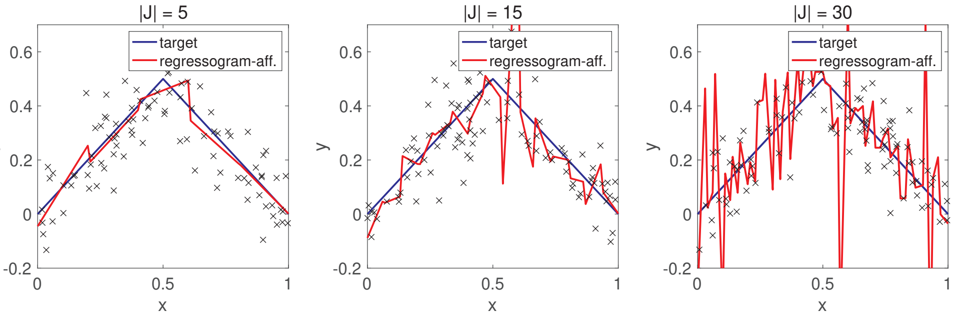


Figure 6.6: Locally linear regression, on the same data as Figure 6.2, for three values of the number $|J|$ of sets within in the partition. Notice the difference with Figure 6.2.

**Exercise 6.5** *(♦) For the Nadaraya Watson estimator, show that when the target function and the kernel are twice continuously differentiable, then the bias term is bounded by a constant times $h^4$. Show that the optimal bandwidth selection leads to a rate proportional to $n^{-4/(4+d)}$.*

## 6.6 Conclusion

In this chapter, we have explored local averaging methods, which leverage the explicit formula for the Bayes predictor and explicitly aim at approximating it, without the need for optimization (as opposed to all other methods presented in this book). While they can potentially adapt to complex prediction functions, they suffer from the curse of dimensionality, that is, the number of observations has to be exponential in dimension for good predictions. Without further assumptions, this is unavoidable, but in the following chapters, we will see that other learning techniques can take advantage of extra assumptions, such as the smoothness of the prediction function (kernels in Chapter 7 and neural networks in Chapter 9), and dependence only a linear projection of the inputs (this will only be possible for neural networks).

Like all techniques presented in this book, local averaging methods can also be used within “ensemble methods” that combine several predictors learned on modifications of the original dataset (see Chapter 10).

# Chapter 7

# Kernel methods

### Chapter summary

– Kernels and representer theorem: learning with infinite-dimensional linear models can be done in a time that depends on the number of observations using a kernel function.
– Kernels on $\mathbb{R}^d$: such models include polynomials and classical Sobolev spaces (functions with square-integrable partial derivatives).
– Algorithms: convex optimization algorithms can be applied with theoretical guarantees and many dedicated developments to avoid the quadratic complexity of computing the kernel matrix.
– Analysis of well-specified models: When the target function is in the associated function space, learning can be done with rates that are independent of dimension.
– Analysis of misspecified models: if the target function is not in the function space, the curse of dimensionality cannot be avoided in the worst-case situations of few existing derivatives of the target function, but the methods are adaptive to any amount of intermediate smoothness.
– Sharp analysis of ridge regression: for the square loss, a more involved analysis leads to optimal rates in various situations in $\mathbb{R}^d$.

In this chapter, we consider positive-definite kernel methods, with only a brief account of the main results. For more details, see Schölkopf and Smola (2001); Shawe-Taylor and Cristianini (2004); Christmann and Steinwart (2008), and teaching slides from Jean-Philippe vert (available from `https://jpvert.github.io/`).

## 7.1 Introduction

In this chapter, we study empirical risk minimization for linear models, that is, prediction functions $f_\theta : \mathcal{X} \to \mathbb{R}$ which are *linear in their parameters* $\theta$, that is, functions of the form $f_\theta(x) = \langle \theta, \varphi(x) \rangle_{\mathcal{H}}$, where $\varphi : \mathcal{X} \to \mathcal{H}$ and $\mathcal{H}$ is a Hilbert space (essentially a Euclidean space with potentially infinite dimension)[1], and $\theta \in \mathcal{H}$. We will often use the notation $\langle \theta, \varphi(x) \rangle$ in this chapter instead of $\langle \theta, \varphi(x) \rangle_{\mathcal{H}}$ when this is not ambiguous.

The key difference with Chapter 3 on least-squares estimation is that (1) we are not restricted to the square loss (although many of the same concepts will play a role, in particular, in the analysis of ridge regression), and (2), we will explicitly allow infinite-dimensional models, thus extending the dimension-free bounds from Chapter 3. The notion of *kernel function* (or simply kernel) $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$ will be particularly fruitful.

**Why is this relevant?** The study of infinite-dimensional linear methods is important for several reasons:

- Understanding linear models in finite but very large input dimensions requires tools from infinite-dimensional analysis.
- Kernel methods lead to simple and stable algorithms, with theoretical guarantees and adaptivity to the smoothness of the target function (as opposed to local averaging techniques). They can be applied in high dimensions, with good practical performance (note that for supervised learning problems with many observations in domains such as computer vision and natural language processing, they do not achieve the state of the art anymore, which is achieved by neural networks presented in Chapter 9).
- They can be easily applied when input observations are not vectors (see Section 7.3.4).
- They are helpful to understand other models such as neural networks (see Chapter 9).

⚠ The type of kernel we consider here is different from the ones in Chapter 6. The ones here are "positive definite," the ones from Chapter 6 are "non-negative". See more details in https://francisbach.com/cursed-kernels/.

## 7.2 Representer theorem

Dealing with infinite-dimensional models initially seems impossible because algorithms cannot be run in infinite dimensions. In this section, we show how the kernel function plays a crucial role in achieving lower-dimensional algorithms.

[1] More precisely, this is a vector space endowed with a inner product and which is complete for the associated normed space topology. See https://en.wikipedia.org/wiki/Hilbert_space for more details.

As a motivation, we consider the optimization problem coming from machine learning with linear models, with data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$:

$$\min_{\theta \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \langle \varphi(x_i), \theta \rangle) + \frac{\lambda}{2} \|\theta\|^2, \tag{7.1}$$

assuming the loss function $\ell$ is already from $\mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ and not from $\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (e.g., hinge loss, logistic loss or least-squares, see Chapter 4).

The key property of the objective function in Eq. (7.1) is that it accesses the input observations $x_1, \dots, x_n \in \mathcal{X}$, only through dot-products $\langle \theta, \varphi(x_i) \rangle$, $i = 1, \dots, n$, and that we penalize using the Hilbertian norm $\|\theta\|$. The following proposition is crucial and has an elementary proof, due to Kimeldorf and Wahba (1971) for Corollary 7.1, and to Schölkopf et al. (2001) for the general form presented below.

**Proposition 7.1 (Representer theorem)** *Consider a feature map $\varphi : \mathcal{X} \to \mathcal{H}$. Let $(x_1, \dots, x_n) \in \mathcal{X}^n$, and assume that the functional $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ is strictly increasing with respect to the last variable, then the infimum of*

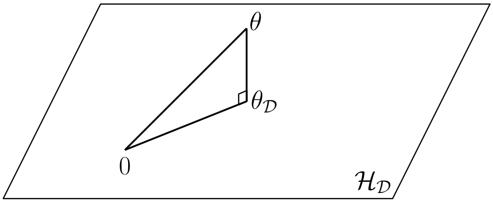
$$\Psi(\langle \theta, \varphi(x_1) \rangle, \dots, \langle \theta, \varphi(x_n) \rangle, \|\theta\|^2)$$

*can be obtained by restricting to a vector $\theta$ of the form*

$$\theta = \sum_{i=1}^{n} \alpha_i \varphi(x_i),$$

*with $\alpha \in \mathbb{R}^n$.*

**Proof** Let $\theta \in \mathcal{H}$, and $\mathcal{H}_{\mathcal{D}} = \big\{ \sum_{i=1}^n \alpha_i \varphi(x_i), \ \alpha \in \mathbb{R}^n \big\} \subset \mathcal{H}$, the linear span of the observed feature vectors. Let $\theta_{\mathcal{D}} \in \mathcal{H}_{\mathcal{D}}$ and $\theta_\perp \in \mathcal{H}_{\mathcal{D}}^\perp$ be such that $\theta = \theta_{\mathcal{D}} + \theta_\perp$, a decomposition which is using the Hilbertian structure of $\mathcal{H}$. Then $\forall i \in \{1, \dots, n\}$, $\langle \theta, \varphi(x_i) \rangle = \langle \theta_{\mathcal{D}}, \varphi(x_i) \rangle + \langle \theta_\perp, \varphi(x_i) \rangle$ with $\langle \theta_\perp, \varphi(x_i) \rangle = 0$, by definition of the orthogonal.



From the Pythagorean theorem, we get: $\|\theta\|^2 = \|\theta_{\mathcal{D}}\|^2 + \|\theta_\perp\|^2$. Therefore, we have:

$$\begin{aligned} \Psi(\langle \theta, \varphi(x_1) \rangle, \dots, \langle \theta, \varphi(x_n) \rangle, \|\theta\|^2) &= \Psi(\langle \theta_{\mathcal{D}}, \varphi(x_1) \rangle, \dots, \langle \theta_{\mathcal{D}}, \varphi(x_n) \rangle, \|\theta_{\mathcal{D}}\|^2 + \|\theta_\perp\|^2) \\ &\geqslant \Psi(\langle \theta_{\mathcal{D}}, \varphi(x_1) \rangle, \dots, \langle \theta_{\mathcal{D}}, \varphi(x_n) \rangle, \|\theta_{\mathcal{D}}\|^2), \end{aligned}$$

with equality if and only if $\theta_\perp = 0$ (since $\Psi$ is strictly increasing with respect to the last variable). Thus

$$\inf_{\theta \in \mathcal{H}} \Psi(\langle \theta, \varphi(x_1) \rangle, \dots, \langle \theta, \varphi(x_n) \rangle, \|\theta\|^2) = \inf_{\theta \in \mathcal{H}_{\mathcal{D}}} \Psi(\langle \theta, \varphi(x_1) \rangle, \dots, \langle \theta, \varphi(x_n) \rangle, \|\theta\|^2),$$

which is exactly the desired result. ■

This implies that the minimizer of Eq. (7.1) can be found among the vectors of the form $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$:

**Corollary 7.1 (Representer theorem for supervised learning)** *For $\lambda > 0$, the infimum of $\frac{1}{n}\sum_{i=1}^n \ell(y_i, \langle \theta, \varphi(x_i)\rangle) + \frac{\lambda}{2}\|\theta\|^2$ can be obtained by restricting to a vector $\theta$ of the form $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$, with $\alpha \in \mathbb{R}^n$.*

It is important to note that there is no assumption on the loss function $\ell$. In particular, no convexity is assumed. This is to be contrasted to the use of duality in Section 7.4.4, where convexity will play a major role and similar $\alpha$'s will be defined (but with some notable differences).

Given Corollary 7.1, we can reformulate the learning problem. We will need the *kernel function* $k$, which is the dot product between feature vectors:

$$k(x, x') = \langle \varphi(x), \varphi(x')\rangle.$$

We then have:

$$\forall j \in \{1, \dots, n\},\ \langle \theta, \varphi(x_j)\rangle = \sum_{i=1}^n \alpha_i k(x_i, x_j) = (K\alpha)_j,$$

where $K \in \mathbb{R}^{n\times n}$ is the *kernel matrix*, such that $K_{ij} = \langle \varphi(x_i), \varphi(x_j)\rangle = k(x_i, x_j)$, and

$$\|\theta\|^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j)\rangle = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K_{ij} = \alpha^\top K \alpha.$$

We can then write:

$$\inf_{\theta \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^n \ell(y_i, \langle \theta, \varphi(x_i)\rangle) + \frac{\lambda}{2}\|\theta\|^2 = \inf_{\alpha \in \mathbb{R}^n} \frac{1}{n}\sum_{i=1}^n \ell(y_i, (K\alpha)_i) + \frac{\lambda}{2}\alpha^\top K \alpha. \tag{7.2}$$

For a test point $x \in \mathcal{X}$, we have $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$.

Thus, the input observations are summarized in the kernel matrix and the kernel function, regardless of the dimension of $\mathcal{H}$. Moreover, explicitly computing the feature vector $\varphi(x)$ is never needed! This is the *kernel trick*, which allows to:

- replace the search space $\mathcal{H}$ by $\mathbb{R}^n$; this is interesting computationally when the dimension of $\mathcal{H}$ is very large (see more details in Section 7.4),
- separate the representation problem (design of kernels on a set $\mathcal{X}$) and the design of algorithms and their analysis (which only use the kernel matrix $K$); this is interesting because a wide range of kernels can be defined for many data types (see more details in Section 7.3).

**Minimum norm interpolation.** The representer theorem can be extended to an interpolating estimator with essentially the same proof (see proposition below).

**Proposition 7.2** *Given $x_1, \dots, x_n \in \mathcal{X}$, and $y \in \mathbb{R}^n$ such that there exists at least one $\theta \in \mathcal{H}$ such that $y_i = \langle \theta, \varphi(x_i) \rangle$ for all $i \in \{1, \dots, n\}$, then among all these $\theta \in \mathcal{H}$ that interpolate the data, the one of minimum norm can be expressed as $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$, with $\alpha \in \mathbb{R}^n$ is such that $y = K\alpha$ (this system must then have a solution).*

## 7.3 Kernels

In the section above, we have introduced the kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as obtained from a dot product $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$. The associated kernel matrix is then a matrix of dot-products (often called a "Gram matrix") and is thus symmetric positive semi-definite (see proof below), that is, all of its eigenvalues are non-negative, or, equivalently, $\forall \alpha \in \mathbb{R}^n$, $\alpha^\top K \alpha \geqslant 0$. It turns out that this simple property is enough to impose the existence of a feature function.

⚠ If $\mathcal{H} = \mathbb{R}^d$, and $\Phi \in \mathbb{R}^{n \times d}$ is the matrix of features (design matrix in the context of regression) with $i$-th row composed of $\varphi(x_i)$, then $K = \Phi\Phi^\top \in \mathbb{R}^{n \times n}$ is the kernel matrix, while $\frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d \times d}$ is the empirical covariance matrix.

**Definition 7.1 (Positive definite kernels)** *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel if and only if all kernel matrices are symmetric positive semi-definite.*

The following important proposition dates back to Aronszajn (1950) and comes with an elegant constructive proof. Note the total absence of assumptions on the set $\mathcal{X}$.

**Proposition 7.3 (Aronszajn, 1950)** *$k$ is a positive definite kernel if and only if there exists a Hilbert space $\mathcal{H}$, and a function $\varphi : \mathcal{X} \to \mathcal{H}$ such that for all $x, x' \in \mathcal{X}$, $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$.*

**Partial proof** We first assume that $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$. Then for any $\alpha \in \mathbb{R}^n$, and points $x_1, \dots, x_n \in \mathcal{X}$, we have:

$$\alpha^\top K \alpha = \sum_{i,j=1}^n \alpha_i \alpha_i \langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}} = \Big\| \sum_{i=1}^n \alpha_i \varphi(x_i) \Big\|_{\mathcal{H}}^2 \geqslant 0.$$

Thus, $k$ is a positive definite kernel.

For the other direction, we consider a positive-definite kernel, and we will construct a space of functions explicitly from $\mathcal{X}$ to $\mathbb{R}$ with a dot-product. We define the set $\mathcal{H}' \subset \mathbb{R}^{\mathcal{X}}$ as the set of linear combinations of kernel functions $\sum_{i=1}^n \alpha_i k(\cdot, x_i)$ for any integer $n$, any set of $n$ points, and any $\alpha \in \mathbb{R}^n$. This is a vector space on which we can define a dot-product through

$$\Big\langle \sum_{i=1}^n \alpha_i k(\cdot, x_i), \sum_{j=1}^m \beta_j k(\cdot, x_j') \Big\rangle = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j k(x_i, x_j'). \tag{7.3}$$

We first check that this is a well-defined function on $\mathcal{H}' \times \mathcal{H}'$, that is, the value does not depend on the chosen representation as a linear combination of kernel functions. Indeed, if we denote $f = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$, then the dot-product is equal to $\sum_{j=1}^m \beta_j f(x_j')$ and depends only on the values of $f$, and not on its representation (and similarly for the function on the right of the dot-product).

This dot-product is bi-linear and always non-negative when applied to the same function (that is, in Eq. (7.3) above, when $\alpha = \beta$ and the points $(x_i)$ and $(x_j')$ are the same, we get a positive number because of the positivity of the kernel $k$). To obtain a dot-product, we only need to show that $\langle f, f \rangle = 0$ implies $f = 0$. This can be shown using Cauchy-Schwarz inequality,[2] leading to for any $x \in \mathcal{X}$, to the sequence of bounds $f(x)^2 = \langle f, k(\cdot, x)\rangle^2 \leqslant \langle f, f\rangle \langle k(\cdot, x), k(\cdot, x)\rangle = \langle f, f\rangle k(x, x)$, leading to $f = 0$ as soon as $\langle f, f \rangle = 0$.

Moreover, this dot product satisfies the two properties for any $f \in \mathcal{H}'$, $x, x' \in \mathcal{X}$:

$$\langle k(\cdot, x), f\rangle = f(x) \text{ and } \langle k(\cdot, x), k(\cdot, x')\rangle = k(x, x').$$

These are called "reproducing properties" and correspond to an explicit construction of the feature map $\varphi(x) = k(\cdot, x)$.

The space $\mathcal{H}'$ is called "pre-Hilbertian" because it is not complete.[3] It can be "completed" into a Hilbert space $\mathcal{H}$ with the same reproducing property. See Aronszajn (1950); Berlinet and Thomas-Agnan (2004) for more details. ■

We can make the following observations:

- $\mathcal{H}$ is called the "feature space," and $\varphi$ the "feature map," that goes from the "input space" $\mathcal{X}$ to the feature space $\mathcal{H}$.
- No assumption is needed about the input space $\mathcal{X}$, and no regularity assumption is needed for $k$. Up to isomorphisms, the feature map and space happen to be unique. The particular space of functions we built is called the *reproducing kernel Hilbert space (RKHS)* associated with $k$, for which $\varphi(x) = k(\cdot, x)$.
- A classical intuitive interpretation of the identity $\langle k(\cdot, x), f\rangle = f(x)$ is that the function evaluation is the dot-product with a function (this is, in fact, another characterization, see the exercise below). This implies that not all Hilbert spaces of real-valued functions on $\mathcal{X}$ are RKHSs. Indeed, for example, if $L_2(\mathbb{R}^d)$ was an RKHS, this would mean that there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $\int_{\mathbb{R}^d} k(x, x') f(x') dx' = f(x)$. In other words, $k(x, x')dx'$ would be a Dirac measure at $x$, which is impossible (as Dirac measures have no density with respect to the Lebesgue measure). Thus $L_2(\mathbb{R}^d)$ is a Hilbert space that is too large to be an RKHS. We will see below that smaller spaces of functions, with square-integrable derivatives of sufficiently high order, will be RKHSs.
- Given a positive-definite kernel $k$, we can thus associate it to some feature map $\varphi$ such that $k(x, y) = \langle \varphi(x), \varphi(y)\rangle_{\mathcal{H}}$, but also to a *space of functions on* $\mathcal{X}$ *with a given*

[2] Cauchy-Schwarz inequality applies to bilinear forms that are symmetric positive semi-definite, but may not be positive defiite, that is, such that $\langle f, f \rangle = 0$ may not imply $f = 0$.

[3] See https://en.wikipedia.org/wiki/Complete_metric_space for definitions.

*norm*, either directly through the RKHS above or by looking at all functions $f_\theta$ of the form $f_\theta(x) = \langle \theta, \varphi(x) \rangle_{\mathcal{H}}$, with a regularization term $\|\theta\|_{\mathcal{H}}^2$. These two views are equivalent.

⚠ From now on, we will denote elements of the Hilbert space $\mathcal{H}$ through the notation $f \in \mathcal{H}$ to highlight the fact that we are considering a space of functions from $\mathcal{X}$ to $\mathbb{R}$, except for optimization algorithms in Section 7.4, where will use the notation $\langle \theta, \varphi(x) \rangle_{\mathcal{H}}$ instead of $f(x)$.

**Exercise 7.1** *(♦) Let $\mathcal{H}$ be a Hilbert space of real-valued functions on $\mathcal{X}$, such that for any $x \in \mathcal{X}$, the linear form $x \mapsto f(x)$ is bounded (that is, $\sup_{f \in \mathcal{H},\ \|f\|_{\mathcal{H}} \leqslant 1} |f(x)|$ is finite). Using the Riesz representation theorem, show that this is a reproducing kernel Hilbert space.*

**Kernel calculus.** The set of positive definite kernels on a set $\mathcal{X}$ is a cone, that is, it is closed under addition and multiplication by a positive constant. In other words, if $k_1$ and $k_2$ are two positive definite kernels and $\lambda_1, \lambda_2 > 0$, then so is $\lambda_1 k_1 + \lambda_2 k_2$. A simple proof follows from considering two feature maps $\varphi_1 : \mathcal{X} \to \mathcal{H}_1$ and $\varphi_2 : \mathcal{X} \to \mathcal{H}_2$, and noticing that $x \mapsto \binom{\lambda_1^{1/2}\varphi_1(x)}{\lambda_2^{1/2}\varphi_2(x)}$ is a feature map for $\lambda_1 k_1 + \lambda_2 k_2$.

Moreover, positive definite kernels are closed under pointwise multiplication, that is, if $k_1$ and $k_2$ are positive definite kernels on the set $\mathcal{X}$, so is, $(x, x') \mapsto k_1(x, x') k_2(x, x')$. For finite-dimensional kernels, where we can consider feature spaces $\mathcal{H}_1 = \mathbb{R}^{d_1}$ and $\mathcal{H}_2 = \mathbb{R}^{d_2}$, the product kernel is associated with a feature space of dimension $d_1 d_2$ and the feature map $x \mapsto \big[\varphi_1(x)_{i_1} \varphi_2(x)_{i_2}\big]_{i_1 \in \{1,\dots,d_1\}, i_2 \in \{1,\dots,d_2\}}$. The general proof is left as an exercise.

**Exercise 7.2** *Show that if $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel, so is the function $(x, x') \mapsto e^{k(x,x')}$.*

**Kernels = features and functions.** A positive-definite kernel thus defines a feature map and a space of functions. Sometimes, the feature map is easy to find, and sometimes it is not. In the next section, we will look at the main examples and describe the associated spaces of functions (and the corresponding norms).

We now look at different ways of building the kernels, by starting first from the feature vector (e.g., linear kernels), from the kernel and explicit feature map (polynomial kernel), from the norm (translation-invariant kernel on $[0, 1]$), or from the kernel without explicit features (translation-invariant kernel on $\mathbb{R}^d$).

### 7.3.1 Linear and polynomial kernels

We start with the most obvious kernels on $\mathcal{X} = \mathbb{R}^d$, for which feature maps are easily found.

**Linear kernel.** We define $k(x,x') = x^\top x'$. It corresponds to a function space composed of linear functions $f_\theta(x) = \theta^\top x$, with an $\ell_2$-penalty $\|\theta\|_2^2$. The kernel trick can be useful when the input data have huge dimension $d$ but are quite sparse (many zeros), such as in text processing, so that the dot-product $x^\top x'$ can be computed in time $o(d)$.

**Polynomial kernel.** For $s$ a postive integer, the kernel $k(x,x') = (x^\top x')^s$ is positive-definite as a integer power of a kernel and can be explicitly expanded as (with the binomial theorem[4]):

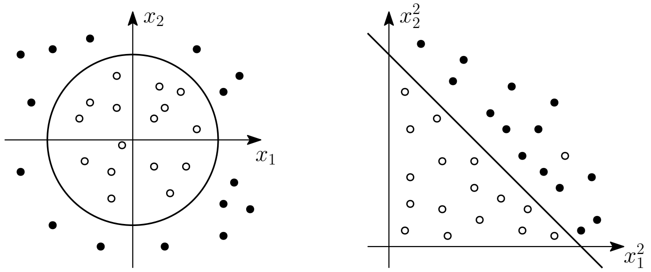
$$k(x,x') = \Big(\sum_{i=1}^d x_i x_i'\Big)^s = \sum_{\alpha_1+\cdots+\alpha_d=s} \binom{s}{\alpha_1,\dots,\alpha_d} \underbrace{(x_1x_1')^{\alpha_1}\cdots(x_dx_d')^{\alpha_d}}_{(x_1^{\alpha_1}\cdots x_d^{\alpha_d})((x_1')^{\alpha_1}\cdots(x_d')^{\alpha_d})},$$

where the sum is over all non-negative integer vectors $(\alpha_1,\dots,\alpha_d)$ that sum to $s$. We have an explicit feature map: $\varphi(x) = \big(\binom{s}{\alpha_1,\dots,\alpha_d}^{\frac{1}{2}} x_1^{\alpha_1}\cdots x_d^{\alpha_d}\big)_{\alpha_1+\cdots+\alpha_d=s}$, and the set of functions is the set of degree-$s$ homogeneous[5] polynomials on $\mathbb{R}^d$, which has dimension $\binom{d+s-1}{s}$.

When $d$ and $s$ grow, the feature space dimension grows as $d^s$, an explicit representation is not desirable, and the kernel trick can be advantageous. Note, however, that the associated norm (which penalizes coefficients of the polynomials) is hard to interpret (as a small change in a single high-order coefficient can lead to significant changes).

**Exercise 7.3** *Show that the kernel $k(x,x') = (1 + x^\top x')^s$ corresponds to the set of all monomials $x_1^{\alpha_1}\cdots x_d^{\alpha_d}$ such that $\alpha_1+\cdots+\alpha_d \leqslant s$. Show that the dimension of the feature space is $\binom{d+s}{s}$.*

As an illustration, when using a polynomial kernel of degree 2, the set of functions that are linear in the feature map is, therefore, the set of quadratic functions. Thus, in a binary classification problem where data can be separated by an ellipsoid, this can be obtained by linear separation in the feature space. See the illustration below.



[4] See https://en.wikipedia.org/wiki/Binomial_theorem .

[5] A function $f : \mathbb{R}^d \to \mathbb{R}$ is said homogeneous if there exists $s \in \mathbb{R}_+$ such that for all $x \in \mathbb{R}^d$, and $\lambda \in \mathbb{R}_+$, $f(\lambda x) = \lambda^s f(x)$.

### 7.3.2 Translation-invariant kernels on $[0, 1]$

We consider $\mathcal{X} = [0, 1]$, and kernels of the form $k(x, x') = q(x - x')$ with a function $q : [0, 1] \to \mathbb{R}$, which is 1-periodic. We will show how they emerge from penalties on the Fourier coefficients of functions, which we now quickly review.[6]

**Fourier series.** We will consider complex-valued functions and use complex exponentials, but all developments could be carried out with cosines and sines. Fourier series corresponds simply to an orthonormal decomposition of square-integrable functions on $[0, 1]$, which are naturally extended to 1-periodic functions on $\mathbb{R}$. More precisely, the set of functions $x \mapsto e^{2im\pi x}$, for $m \in \mathbb{Z}$ is an orthonormal basis of $L_2([0, 1])$. Therefore, any squared integrable functions which are 1-periodic can be expanded in this orthonormal basis, that is, $q(x) = \sum_{m\in\mathbb{Z}} e^{2im\pi x} \hat{q}_m$, with $\hat{q}_m = \int_0^1 q(x) e^{-2im\pi x} dx \in \mathbb{C}$, for $m \in \mathbb{Z}$, obtained by projection $q$ to the element of the basis. The function $q$ is real-valued if and only if for all $m \in \mathbb{Z}$, $\hat{q}_{-m} = \hat{q}_m^*$ (the complex conjugate of $\hat{q}_m$). We will also need Parseval's identity, which is exactly the Pythagorean theorem in the orthonormal basis, that is, $\int_0^1 |q(x)|^2 dx = \sum_{m\in\mathbb{Z}} |\hat{q}_m|^2$.

**Translation-invariant kernels.** When presenting translation-invariant kernels, we can choose to start from the kernel or the associated squared norm. In this section, we start from the squared norm, while in the next one, we start from the kernel.

Given a 1-periodic function $f$ decomposed into its Fourier series as

$$f(x) = \sum_{m\in\mathbb{Z}} e^{2im\pi x} \hat{f}_m,$$

we consider the penalty

$$\sum_{m\in\mathbb{Z}} c_m |\hat{f}_m|^2,$$

with $c \in \mathbb{R}_+^{\mathbb{Z}}$; this penalty can be interpreted through a feature map and the $\ell_2$-norm on $\mathbb{C}^{\mathbb{Z}}$ (the space of functions from $\mathbb{Z}$ to $\mathbb{C}$). Indeed, it corresponds to the feature vector $\varphi(x)_m = e^{2im\pi x}/\sqrt{c_m}$, and $\theta \in \mathbb{C}^{\mathbb{Z}}$, such that $\theta_m = \hat{f}_m \sqrt{c_m}$ (we can easily consider complex-valued features instead of real-valued features if Hermitian dot-products are considered), so that $f(x) = \langle \theta, \varphi(x) \rangle$ and $\sum_{m\in\mathbb{Z}} |\theta_m|^2$ is equal to the norm $\sum_{m\in\mathbb{Z}} c_m |\hat{f}_m|^2$.

Thus the associated kernel is

$$k(x, x') = \sum_{m\in\mathbb{Z}} \varphi(x)_m \varphi(x')_m^* = \sum_{m\in\mathbb{Z}} \frac{e^{2im\pi x}}{\sqrt{c_m}} \frac{e^{-2im\pi x'}}{\sqrt{c_m}} = \sum_{m\in\mathbb{Z}} \frac{1}{c_m} e^{2im\pi(x-x')},$$

which is thus of the form $q(x - x')$ for a 1-periodic function $q$.

What we showed above is that any penalty of the form $\sum_{m\in\mathbb{Z}} c_m |\hat{f}_m|^2$ defines a squared RKHS norm as soon as $c_m$ is strictly positive for all $m \in \mathbb{Z}$, and $\sum_{m\in\mathbb{Z}} \frac{1}{c_m}$ is

[6] See https://en.wikipedia.org/wiki/Fourier_series for more details.

finite. The kernel function is then of the form $k(x,y) = q(x-y)$ with $q$ being 1-periodic, and such that the Fourier series has non-negative real values $\hat{q}_m = c_m^{-1}$. In the other direction, all such kernels are positive-definite (see an extension to $\mathbb{R}^d$ in Section 7.3.3).

**Penalization of derivatives.** For certain penalties based on $(c_m)_m$, there is a natural link with penalties on derivatives, as if $f$ is $s$-times differentiable with squared integrable derivative, we have, by differentiating the Fourier series representation above, $f^{(s)}(x) = \sum_{m\in\mathbb{Z}} (2im\pi)^s e^{2im\pi x} \hat{f}_m$, and thus, from Parseval's theorem (which states that the squared $L_2$-norm of a function is equal to the sum of the square modulus of its Fourier coefficients):

$$\int_0^1 |f^{(s)}(x)|^2 dx = (2\pi)^{2s} \sum_{m\in\mathbb{Z}} m^{2s} |\hat{f}_m|^2.$$

In this chapter, we will consider penalizing such derivatives, leading to Sobolev spaces on $[0,1]$ (see extensions in sections below). The following examples are often considered:

- **Bernoulli polynomials**: we can consider $c_0 = 1$ and $c_m = |m|^{2s}$ for $m \neq 0$, for which the associated norm is $\|f\|_{\mathcal{H}}^2 = \frac{1}{(2\pi)^{2s}} \int_0^1 |f^{(s)}(x)|^2 dx + \big(\int_0^1 f(x)dx\big)^2$. The corresponding kernel $k(x,x')$ can then be written as

  $$k(x,x') = \sum_{m\in\mathbb{Z}} c_m^{-1} e^{2im\pi(x-x')} = 1 + \sum_{m\geqslant 1} \frac{2\cos[2\pi m(x-x')]}{m^{2s}}.$$

  In order to have an expression for $q$ in closed form we notice that if we define $\{x\} = x - \lfloor x \rfloor \in [0,1)$ the fractional part of $x$, the function $x \mapsto \{x\}$ has (by integration by part) an $m$-th Fourier coefficient equal to $\int_0^1 e^{-2im\pi x} x dx = \frac{i}{2m\pi}$. Similarly, the $s$-th power of $\{x\}$ has an $m$-th Fourier coefficient which is an order $s$ polynomial in $m^{-1}$. This implies that $k(x,x')$ has to be an order $s$ polynomial in $\{x-x'\}$, which happens to be related to classical polynomials, as we now show.

  For $s=1$, we have $k(x,x) = 1 + 2\sum_{m\geqslant 1} m^{-2} = 1 + \pi^2/3$; moreover by using the Fourier series expansion $\{t\} = \frac{1}{2} - \frac{1}{2\pi}\sum_{m\geqslant 1} \frac{2\sin[2\pi m t]}{m}$, and integrating, we get

  $$k(x,x') = 2\pi^2\{x-x'\}^2 - 2\pi^2\{x-x'\} + \pi^2/3 + 1 = q(x-x'),$$

  with $q$ plotted in Figure 7.1. For $s \geqslant 1$, we have the closed-form expression $k(x,x') = 1 + (-1)^{s-1}\frac{(2\pi)^{2s}}{(2s)!} B_{2s}(\{x-x'\})$, where $B_{2s}$ the $(2s)$-th Bernoulli polynomial,[7] from which we can "check" the computation above since $B_2(t) = t^2 - t + 1/6$.

  **Exercise 7.4** *Show that for $s=2$, we have for all $x,x' \in [0,1]$, $k(x,x') = q(x-x')$ with $q(t) = 1 - \frac{(2\pi)^4}{24}\big(\{t\}^4 - 2\{t\}^3 + \{t\}^2 - \frac{1}{30}\big)$.*

[7] See https://en.wikipedia.org/wiki/Bernoulli_polynomials.

- **Periodic exponential kernel**: we can consider $c_m = 1 + \alpha^2|m|^2$, for which we also have a closed-form formula, with the penalty $\|f\|_{\mathcal{H}}^2 = \frac{\alpha^2}{(2\pi)^2}\int_0^1 |f'(x)|^2 dx + \int_0^1 |f(x)|^2 dx$.

**Exercise 7.5** *(♦♦♦) Show that we have $k(x,x') = \sum_{m\in\mathbb{Z}} \frac{e^{2im\pi(x-x')}}{1+\alpha^2|m|^2} = q(x-x')$ for $q(t) = \frac{\pi}{\alpha}\frac{\cosh\frac{\pi}{\alpha}(1-2|\{t+1/2\}-1/2|)}{\sinh\frac{\pi}{\alpha}}$. Hint: use the Cauchy residue formula.*[8]

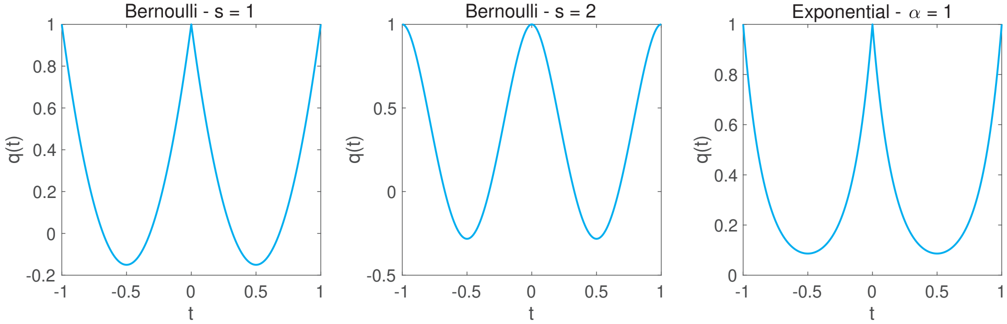


Figure 7.1: Translation-invariant kernels on $[0,1]$, of the form $k(x,x') = q(x-x')$, with $q$ 1-periodic, for the kernels based on Bernoulli polynomials, and the periodic exponential kernel. Kernels are normalized so that $k(x,x) = 1$.

These kernels are mainly used for their simplicity and explicit feature map, which are simpler than the kernels described below which are most used in practice (with similar links with Sobolev spaces). Note also that for the uniform distribution on $[0,1]$, the Fourier basis will be an orthogonal eigenbasis of the covariance operator with eigenvalues $c_m^{-1}$ (see Section 7.6.6).

We saw that for the kernel $q(x-x')$ with Fourier series $\hat{q}_m$ for $q$, the associated norm is $\sum_{m\in\mathbb{Z}} \frac{|\hat{f}_m|^2}{\hat{q}_m}$. We now extend this to Fourier transforms (instead of Fourier series).

### 7.3.3 Translation-invariant kernels on $\mathbb{R}^d$

We consider $\mathcal{X} = \mathbb{R}^d$, and a kernel of the form $k(x,x') = q(x-x')$ with a function $q : \mathbb{R}^d \to \mathbb{R}$, which we refer to as translation-invariant as it is invariant by the addition of the same constant to both arguments. We start with a short review of Fourier transforms.[9]

**Fourier transforms.** The Fourier transform $\hat{f} : \mathbb{R}^d \to \mathbb{C}$ of an integrable function $f : \mathbb{R}^d \to \mathbb{C}$ can be defined through

$$\hat{f}(\omega) = \int_{\mathbb{R}^d} f(x) e^{-i\omega^\top x} dx,$$

[8] See https://francisbach.com/cauchy-residue-formula/.

[9] See https://en.wikipedia.org/wiki/Fourier_transform for more details.

which is then a continuous function of $\omega$. It can naturally be extended to an operator on all square-integrable functions, and under appropriate conditions on $f$, we can recover $f$ from its Fourier transform, that is,

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^\top x} d\omega.$$

Moreover, Parseval's identity leads to $\int_{\mathbb{R}^d} |f(x)|^2 dx = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\hat{f}(\omega)|^2 d\omega$.

**Translation-invariant kernels.** The following proposition gives conditions under which we obtain a positive definite kernel.

**Proposition 7.4 (Bochner's theorem)** *The kernel $k$ is positive definite if and only if $q$ is the Fourier transform of a non-negative Borel measure. Consequently, if $q$ is integrable (for the Lebesgue measure) and its Fourier transform only has non-negative real values, then $k$ is positive definite.*

**Partial proof** We only give the proof of the consequence, which is the only one we need. Since $q$ is integrable, $\hat{q}(\omega) = \int_{\mathbb{R}^d} e^{-i\omega^\top x} q(x) dx$ is defined on $\mathbb{R}^d$ and continuous, and we have through the inverse Fourier transform formula:

$$q(x - x') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{q}(\omega) e^{i(x-x')^\top \omega} d\omega.$$

Let $x_1, \dots, x_n \in \mathbb{R}^d$, let $\alpha_1, \dots, \alpha_n \in \mathbb{R}$. We have:

$$\begin{aligned}
\sum_{s,j=1}^n \alpha_s \alpha_j k(x_s, x_j) = \sum_{s,j=1}^n \alpha_s \alpha_j q(x_s - x_j) &= \frac{1}{(2\pi)^d} \sum_{s,j=1}^n \alpha_s \alpha_j \int_{\mathbb{R}^d} e^{i\omega^\top (x_s - x_j)} \hat{q}(\omega) d\omega \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \Big( \sum_{s,j=1}^n \alpha_s \alpha_j e^{i\omega^\top x_s} (e^{i\omega^\top x_j})^* \Big) \hat{q}(\omega) d\omega \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \Big| \sum_{s=1}^n \alpha_s e^{i\omega^\top x_s} \Big|^2 \hat{q}(\omega) d\omega \geqslant 0,
\end{aligned}$$

which shows the positive-definiteness. See Reed and Simon (1978) and Varadhan (2001, Theorem 2.7) for a proof of the other direction. ■

**Construction of the associated norm.** We give an intuitive (non-rigorous) reasoning: if $q$ is integrable, then $\hat{q}(\omega)$ exists and, we have an explicit representation as

$$k(x, x') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \sqrt{\hat{q}(\omega)} e^{i\omega^\top x} \big( \sqrt{\hat{q}(\omega)} e^{i\omega^\top x'} \big)^* d\omega = \int_{\mathbb{R}^d} \varphi(x)_\omega \varphi(x')^*_\omega d\omega,$$

which is of the form $\langle \varphi(x), \varphi(y) \rangle$, with $\varphi(x)_\omega = \frac{1}{(2\pi)^{d/2}} \sqrt{\hat{q}(\omega)} e^{i\omega^\top x}$ (it is non rigorous because the index $\omega$ belongs to $\mathbb{R}^d$, which is not countable). If we consider a function

$f$ defined as $f(x) = \int_{\mathbb{R}^d} \varphi(x)_\omega \theta_\omega d\omega = \langle \varphi(x), \theta \rangle$, then $\theta_\omega = \frac{1}{(2\pi)^{d/2}} \hat{f}(\omega) / \sqrt{\hat{q}(\omega)}$, and the squared norm of $\theta$ is equal to $\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(w)|^2}{\hat{q}(\omega)} d\omega$, where $\hat{f}$ denotes the Fourier transform of $f$. Therefore, the norm of a function $f \in \mathcal{H}$ should be:

$$\|f\|_{\mathcal{H}}^2 = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(w)|^2}{\hat{q}(\omega)} d\omega.$$

Given the candidate for the norm and the associated dot-product, we can simply check that this is the correct one by showing the reproducing property $\langle f, k(\cdot, x) \rangle$ for this dot-product. Note the similarity with the penalty for the kernel on $[0,1]$ (see more similarity below).

**Link with derivatives.** When $f$ has partial derivatives, then the Fourier transform of $\frac{\partial f}{\partial x_j}$ is equal to $i\omega_j$ times the Fourier transform of $f$. This leads to, using Parseval's theorem, $\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\omega_j|^2 |\hat{f}(w)|^2 d\omega = \int_{\mathbb{R}^d} \left| \frac{\partial f}{\partial x_j}(x) \right|^2 dx$, which extends to higher order derivatives:

$$\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\omega_1^{j_1} \cdots \omega_d^{j_d}|^2 |\hat{f}(w)|^2 d\omega = \int_{\mathbb{R}^d} \left| \frac{\partial^j f}{\partial x_1^{j_1} \cdots \partial x_d^{j_d}}(x) \right|^2 dx, \tag{7.4}$$

for a vector $j \in \mathbb{N}^d$. This will allow us to find corresponding norms by expanding $\hat{q}(\omega)^{-1}$ as sums of monomials. We now consider the main classical examples.

**Exponential kernel.** This is the kernel $q(x - x') = \exp(-\|x - x'\|_2 / r)$, where $r$ is often referred to as the kernel bandwidth (homogeneous to $x$), for which the Fourier transform can be computed as $\hat{q}(\omega) = 2^d \pi^{(d-1)/2} \Gamma((d+1)/2) \frac{r^d}{(1 + r^2 \|\omega\|_2^2)^{(d+1)/2}}$. See Rasmussen and Williams (2006, page 84). Thus, for $d$ odd, $\hat{q}(\omega)^{-1}$ is a sum of monomials, and looking at their orders, we see that the corresponding RKHS norm (that is, the norm on the space of functions on $\mathbb{R}^d$ that our kernel defines) is penalizing all derivatives up to total order $(d+1)/2$, that is, in Eq. (7.4), for all $j \in \mathbb{N}^d$ such that $j_1 + \cdots + j_d \leqslant (d+1)/2$, which is a Sobolev space (fractional for $d$ even).[10]

In particular, for $d = 1$, we have $\hat{q}(\omega) = \frac{2r}{1 + r^2 \omega^2}$, and thus

$$\begin{aligned} \|f\|_{\mathcal{H}}^2 &= \frac{1}{2\pi} \int_{\mathbb{R}} \frac{|\hat{f}(w)|^2}{\hat{q}(\omega)} d\omega = \frac{1}{2r} \frac{1}{2\pi} \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega + \frac{r}{2} \frac{1}{2\pi} \int_{\mathbb{R}} |\omega \hat{f}(\omega)|^2 d\omega \\ &= \frac{1}{2r} \int_{\mathbb{R}} |f(x)|^2 dx + \frac{r}{2} \int_{\mathbb{R}} |f'(x)|^2 dx, \end{aligned}$$

[10] See https://en.wikipedia.org/wiki/Sobolev_space.

and we recover the Sobolev space of functions with squared-integrable derivatives.

⚠ The constant $r$ is homogeneous to the input $x$, while the constant $R$ will be homogeneous to features $\varphi(x)$ (that is, square roots of kernel values). A common rule of thumb is to choose $r$ to be a quantile (such as the median) of all pairwise distances $\|x_i - x_j\|_2$ of the training data.

**Gaussian kernel.** This is the kernel $q(x - x') = \exp(-\|x - x'\|_2^2/r^2)$ (still with a kernel bandwidth $r$), for which the Fourier transform can be explicitly computed as $\hat{q}(\omega) = (\pi r^2)^{d/2} \exp(-r^2\|\omega\|_2^2/4)$. By expanding $\hat{q}(\omega)^{-1}$ through its power series as $\hat{q}(\omega)^{-1} = (\pi r^2)^{d/2} \sum_{s=0}^{\infty} \frac{(r\|\omega\|_2)^{2s}}{4^s s!}$, this corresponds to an RKHS norm which is penalizing all derivatives. Note that all members of this RKHS (the associated function space) are infinitely differentiable and thus much smoother than functions coming from the exponential kernel (the RKHS is smaller).

**Matern kernels and Sobolev spaces.** More generally, one can define a series of kernels so that $\hat{q}(\omega)$ is proportional to $(1 + r^2\|\omega\|_2^2)^{-s}$ for $s > d/2$, to ensure integrability of the Fourier transform. These so-called "Matern kernels" all correspond to Sobolev spaces of order $s$ and can be computed in closed form; see Rasmussen and Williams (2006, page 84). A key fact is that to be an RKHS, a Sobolev space has to have many derivatives when $d$ grows; in particular, having only first-order derivatives ($s = 1$) only leads to an RKHS for $d = 1$, and having $s = 0$ never does.

For $s = \frac{d+3}{2}$, we have $k(x, x') \propto (1 + \sqrt{3}\|x - x'\|_2/r) \exp(-\sqrt{3}\|x - x'\|_2/r)$, and for $s = \frac{d+5}{2}$, we have $k(x, x') \propto (1 + \sqrt{5}\|x - x'\|_2/r + \frac{5}{3}\|x - x'\|_2^2/r^2) \exp(-\sqrt{5}\|x - x'\|_2/r)$. General values $s$ also lead to closed-form formulas (through Bessel functions).

**Density in $L_2(\mathbb{R}^d)$.** For all the kernels below, the set $\mathcal{H}$ is dense in $L_2(\mathbb{R}^d)$ (the set of square-integrable functions with respect to the Lebesgue measure), meaning that all functions in $L_2(\mathbb{R}^d)$ can be approached (with respect to their corresponding norm) by a function in $\mathcal{H}$. This is made quantitative in Section 7.5.2.

⚠ In this chapter, we will consider two spaces of integrable functions, with respect to the Lebesgue measure (which is not a probability measure), which we denote $L_2(\mathbb{R}^d)$, and with respect to the probability measure of the input data, which we denote $L_2(p)$. If $p$ has a density with respect to the Lebesgue measure and this density $\frac{dp}{dx}(x)$ is uniformly bounded, then $L_2(\mathbb{R}^d) \subset L_2(p)$; more precisely, $\|f\|_{L_2(p)} \leqslant \big\|\frac{dp}{dx}\big\|_\infty^{1/2} \|f\|_{L_2(\mathbb{R}^d)}$. However, the converse is not true, simply because being an element of $L_2(\mathbb{R}^d)$ imposes a zero limit at infinity, which being an element of $L_2(p)$ does not impose (moreover, non zero constants are in $L_2(p)$ but not in $L_2(\mathbb{R}^d)$). Note moreover that $\big\|\frac{dp}{dx}\big\|_\infty$ is typically exponential in $d$, and is homogeneous to $r^{-d}$ (in terms of units), where, $r$ is homogeneous to $x$.

**Examples of members of RKHS.** Below, we sampled $n = 10$ random points in $[-1, 1]$ with 10 random responses, and we look for the function $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for

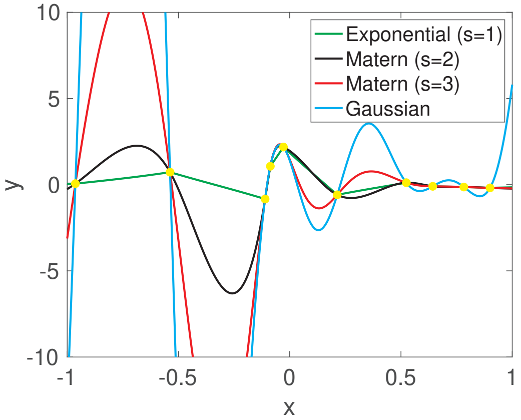


Figure 7.2: Examples of functions in the reproducing kernel Hilbert spaces (RKHS) for several kernels. All functions are the minimum norm interpolators of the yellow points.

all $i \in \{1 \dots, n\}$ and with minimum norm. Given the representer theorem, we can write $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$, and the interpolation condition implies that $K\alpha = y$, and thus $\alpha = K^{-1}y$ (see Prop. 7.2).

We consider several kernels in Figure 7.2, going from close to piecewise affine interpolation to infinitely differentiable functions (for the Gaussian kernel).

### 7.3.4 Beyond vectorial input spaces (♦)

While our theoretical analysis of kernel methods focuses a lot on kernels on $\mathbb{R}^d$ and their link with differentiability properties of the target function, kernels can be applied to a wide variety of problems with various input types. We give below classic examples (see more details by Shawe-Taylor and Cristianini, 2004)

- Set of subsets of a given set $V$: for example, the function $k$ defined as $k(A, B) = \frac{|A \cap B|}{|A \cup B|}$ is a positive definite kernel (classically referred to as the Jaccard index[11]).
- Point clouds: a point cloud in $\mathbb{R}^d$ is a finite subset of $\mathbb{R}^d$, thus with no particular ordering. They occur, for example, in computer vision or graphics. To build a kernel for such objects, a simple first idea is to compute the empirical average of a certain feature vector $\varphi : \mathbb{R}^d \to \mathcal{H}$, and then use a kernel on $\mathcal{H}$. Other kernels may be obtained as functions of the concatenation of the two point clouds (see more details by Cuturi et al., 2005). These constructions extend to probability distributions.
- Text documents/web pages: with the usual "bag of words" assumption, we represent a text document or a web page by considering a vocabulary of "words" (this could be groups of letters, single original words, or groups of words or letters), and counting the number of occurrences of this word in the corresponding document. This gives a typically high-dimensional feature vector $\varphi(x)$ (with dimension the

[11] See https://en.wikipedia.org/wiki/Jaccard_index.

vocabulary size). Using linear functions on this feature provides cheap and stable predictors on such data types (better models that take into account the word order can be obtained, such as neural networks, at the expense of significantly more computational resources). See, e.g., Joulin et al. (2017) for examples.

- Sequences: given some finite alphabet $\mathcal{A}$, we consider the set $\mathcal{X}$ of finite sequences in $\mathcal{A}$ with arbitrary length. A classical infinite-dimensional feature space is indexed by $\mathcal{X}$ itself, and for $y \in \mathcal{X}$, $\varphi(x)_y$ is equal to 1 is $y$ is a subsequence of $x$ (we could also count the number of times the subsequence $y$ appears in $x$, or we could add a weight that depends on $y$, e.g., to penalize longer subsequences). This kernel has an infinite-dimensional feature space, but for two sequences $x$ and $x'$, we can enumerate all subsequences of $x$ and $x'$ and compare them in polynomial time (there exist much faster algorithms, see Gusfield (1997)). These kernels have many applications in bioinformatics.

  The same techniques can be extended to more general combinatorial objects such as trees and graphs (see Shawe-Taylor and Cristianini, 2004).

- Images: before neural networks took over in the years 2010s with the use of large amounts of data, several kernels were designed for images, with often a "bag-of-words" assumption that provides for free invariance by translation. The key is what to consider as "words", i.e., the presence of specific local patterns in the image and the regions under which this assumption is made. See Zhang et al. (2007) for details.

## 7.4 Algorithms

In this section, we briefly mention algorithms aimed at solving

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2, \tag{7.5}$$

for $\ell$ being convex with respect to its second variable. We assume that for all $i \in \{1, \dots, n\}$, $k(x_i, x_i) = \|\varphi(x_i)\|^2 \leqslant R^2$.

### 7.4.1 Representer theorem

We can directly apply the representer theorem, as done in Eq. (7.2) and try to solve

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, (K\alpha)_i) + \frac{\lambda}{2} \alpha^\top K \alpha,$$

which is a convex optimization problem since $\ell$ is assumed convex with respect to the second variable, and $K$ is positive-semidefinite.

In the particular case of the square loss (ridge regression), this leads to

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2n} \|y - K\alpha\|_2^2 + \frac{\lambda}{2} \alpha^\top K \alpha,$$

and setting the gradient to zero, we get $(K^2 + n\lambda K)\alpha = Ky$, with a solution $\alpha = (K + n\lambda I)^{-1}y$, which is not unique as soon as $K$ is not invertible.

However, in general (for the square loss and beyond), it is an ill-conditioned optimization problem because $K$ often has very small eigenvalues (more on this later). When the loss is smooth, the Hessians are equal to $\frac{1}{n}K\,\mathrm{Diag}(h)K + \lambda K$, where $h \in \mathbb{R}^n$ is a vector of second-order derivatives of $\ell$, so that the Hessians are ill-conditioned.

A better alternative is to first compute a square root of $K$ as $K = \Phi\Phi^\top$, where $\Phi \in \mathbb{R}^{n\times m}$, and $m$ the rank of $K$, and solve

$$\min_{\beta\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, (\Phi\beta)_i) + \frac{\lambda}{2}\|\beta\|_2^2,$$

with $\beta = \Phi^\top\alpha$. Note that this corresponds to an explicit feature space representation (that is, the rows of $\Phi$ correspond to features in $\mathbb{R}^m$ for the corresponding data point). For ridge regression, the objective function's Hessian is equal to $\frac{1}{n}\Phi^\top\Phi + \lambda I$, which is well-conditioned because its lowest eigenvalue is greater than $\lambda$ and is thus directly controlled by regularization.

Computing a square root can be done in several ways (through Cholesky decomposition or SVD) (Golub and Loan, 1996), in running time $O(m^2 n)$.

### 7.4.2 Column sampling

To approximate $K$, approximate square roots are a very useful tool, and among various algorithms, approximating $K \in \mathbb{R}^{n\times n}$ from a subset of its columns can be done as $K \approx K(V,I)K(I,I)^{-1}K(I,V)$, where $K(A,B)$ is the sub-matrix of $K$ obtained by taking rows from the set $A \subset \{1,\dots,n\}$ and columns from $B \subset \{1,\dots,n\}$, and $V = \{1,\dots,n\}$. See below for an illustration when $I = \{1,\dots,m\}$ and a partition of the kernel matrix.

| | |
|---|---|
| $K(I,I)$ | $K(I,J)$ |
| $K(J,I)$ | $K(J,J)$ |

This corresponds to an approximate square root $\Phi = K(V,I)K(I,I)^{-1/2} \in \mathbb{R}^{n\times m}$, with $m = |I|$, and it can be computed in time $O(m^2 n)$ (computing the entire kernel matrix is not even needed). Then, the complexity is typically $O(m^2 n)$ instead of $O(n^3)$ (e.g., when using matrix inversion for ridge regression, for faster algorithms, see below), and is thus linear in $n$.

**Exercise 7.6** *(♦) Show that column sampling corresponds to approximating optimally each $\varphi(x_j)$, $j \notin I$, by a linear combination of $\varphi(x_i)$, $i \in I$.*

This approximation technique, often called "Nyström approximation," can be analyzed when the columns are chosen randomly (Rudi et al., 2015).

### 7.4.3 Random features

Some kernels have a special form that leads to specific approximation schemes, that is,

$$k(x,x') = \int_{\mathcal{V}} \varphi(x,v)\varphi(x',v)d\tau(v),$$

where $\tau$ is a probability distribution on some space $\mathcal{V}$ and $\varphi(x,v) \in \mathbb{R}$. We can then approximate the expectation by an empirical average

$$\hat{k}(x,x') = \frac{1}{m}\sum_{j=1}^{m} \varphi(x,v_j)\varphi(x',v_j),$$

where the $v_j$'s are sampled i.i.d. from $\tau$. We can thus use an explicit feature representation $\hat{\varphi}(x) = \big(\frac{1}{\sqrt{m}}\varphi(x,v_j)\big)_{j\in\{1,\dots,m\}}$, and solve

$$\min_{\beta\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \hat{\varphi}(x_i)^\top\beta) + \frac{\lambda}{2}\|\beta\|_2^2.$$

For this scheme to make sense, the number $m$ of random features has to be significantly smaller than $n$, which is often sufficient in practice (see an analysis by Rudi and Rosasco, 2017).

⚠ Note that dimension reduction is here performed independently of the input data (that is, the random feature functions $\varphi(\cdot, v_j)$ are selected before the data are observed, as opposed to column sampling, which is a data-dependent dimension reduction scheme.

The two classic examples are:

- **Translation-invariant kernels**: $k(x,y) = q(x-y) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\hat{q}(\omega)e^{i\omega^\top(x-y)}d\omega$, for which we can take $\varphi(x,\omega) = \sqrt{q(0)}e^{i\omega^\top x} \in \mathbb{C}$, where $\omega$ is sampled from the distribution with density $\frac{1}{(2\pi)^d}\frac{\hat{q}(\omega)}{q(0)}$, which is a Gaussian distribution for the Gaussian kernel. Alternatively, one can use a real-valued feature (instead of a complex-valued one) by using $\sqrt{2}\cos(\omega^\top x + b)$ with $b$ sampled uniformly in $[0, 2\pi]$ (Rahimi and Recht, 2008).
- **Neural networks with random weights**: we can start from an expectation, for which the sampled features are classical, e.g., $\varphi(x,v) = \sigma(v^\top x)$ for some function $\sigma : \mathbb{R} \to \mathbb{R}$. For the "rectified linear unit", that is, $\sigma(\alpha) = \max\{0,\alpha\}$, and for $v$ sampled uniformly on the sphere, we have (proof left as an exercise) $k(x,x') = \frac{\|x\|_2\|x'\|_2}{2(d+1)\pi}\big[(\pi-\eta)\cos\eta + \sin\eta\big]$, where $\cos\eta = \frac{x^\top x'}{\|x\|_2\cdot\|x'\|_2}$ (Le Roux and Bengio, 2007). Therefore, we can view a neural network with a large number of hidden neurons,

with random input weights and not optimized as a kernel method. See a thorough discussion in Chapter 9.

### 7.4.4 Dual algorithms (♦)

For the following two algorithms, we go back to the notation $f(x) = \langle \varphi(x), \theta \rangle$ with $\theta \in \mathcal{H}$ because it is more adapted (and is a direct infinite-dimensional extension of the algorithms from Chapter 5). To solve $\min_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \langle \varphi(x_i), \theta \rangle) + \frac{\lambda}{2}\|\theta\|^2$, for a loss which is convex with respect to the second variable, we can derive a Lagrange dual in the following way (for an introduction to Lagrange duality, see Boyd and Vandenberghe, 2004). We start by reformulating the problem as a constrained problem:

$$\begin{aligned}
& \min_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \langle \varphi(x_i), \theta \rangle) + \frac{\lambda}{2}\|\theta\|^2 \\
= & \min_{\theta \in \mathcal{H},\ u \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, u_i) + \frac{\lambda}{2}\|\theta\|^2 \text{ such that } \forall i \in \{1, \dots, n\},\ \langle \varphi(x_i), \theta \rangle = u_i.
\end{aligned}$$

By Lagrange duality, this is equal to (with $\lambda$ added on top of the regular multiplier $\alpha$ for convenience):

$$\begin{aligned}
& \max_{\alpha \in \mathbb{R}^n} \min_{\theta \in \mathcal{H},\ u \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, u_i) + \frac{\lambda}{2}\|\theta\|^2 + \lambda \sum_{i=1}^n \alpha_i \big(u_i - \langle \varphi(x_i), \theta \rangle\big) \\
= & \max_{\alpha \in \mathbb{R}^n} \Big\{ \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \{\ell(y_i, u_i) + n\lambda\alpha_i u_i]\} + \min_{\theta \in \mathcal{H}} \big\{ \frac{\lambda}{2}\|\theta\|^2 - \lambda \sum_{i=1}^n \alpha_i \langle \varphi(x_i), \theta \rangle \big\} \Big\} \\
& \qquad \text{by reordering,} \\
= & \max_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \{\ell(y_i, u_i) + n\lambda\alpha_i u_i\} - \frac{\lambda}{2} \Big\| \sum_{i=1}^n \alpha_i \varphi(x_i) \Big\|^2 \text{ with } \theta = \sum_{i=1}^n \alpha_i \varphi(x_i), \\
= & \max_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \{\ell(y_i, u_i) + n\lambda\alpha_i u_i\} - \frac{\lambda}{2} \alpha^\top K \alpha,
\end{aligned}$$

with $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$ at optimum. Since the functions $\alpha_i \mapsto \min_{u_i \in \mathbb{R}} \{\ell(y_i, u_i) + n\lambda\alpha_i u_i\}$ are concave (as minima of affine functions), this is a concave maximization problem.

Note the similarity with the representer theorem (existence of $\alpha \in \mathbb{R}^n$ such that $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$) and the dissimilarity (one is a minimization problem, one is maximization problem). Moreover, when the loss is smooth, one can show that the function $\alpha_i \mapsto \min_{u_i \in \mathbb{R}} \{\ell(y_i, u_i) + n\lambda\alpha_i u_i\}$ is a strongly concave function, and thus relatively easy to optimize (in other words, the associated condition numbers of dual problems are smaller than when using the representer theorem).

**Exercise 7.7** *(a) For ridge regression, compute the dual problem and compare the condition number of the primal problem and the condition number of the dual problem; (b) compare the two formulations to the use of normal equations as in Chapter 3, and relate the two using the matrix inversion lemma* $(\Phi\Phi^\top + n\lambda I)^{-1}\Phi = \Phi(\Phi^\top\Phi + n\lambda I)^{-1}$.

### 7.4.5 Stochastic gradient descent (♦)

When minimizing an expectation

$$\min_{\theta \in \mathcal{H}} \mathbb{E}\big[\ell(y, \langle \varphi(x), \theta \rangle)\big] + \frac{\lambda}{2}\|\theta\|^2$$

as in Chapter 5, the stochastic gradient algorithm leads to the recursion

$$\theta_t = \theta_{t-1} - \gamma_t \big[\ell'(y_t, \langle \varphi(x_t), \theta_{t-1}\rangle)\varphi(x_t) + \lambda \theta_{t-1}\big],$$

where $(x_t, y_t)$ is an i.i.d. sample from the distribution defining the expectation, and $\ell'$ is the derivative with respect to the second variable.

When initializing at $\theta_0 = 0$, $\theta_t$ is a linear combination of all $\varphi(x_i)$, $i = 1, \dots, t$, and thus we can write

$$\theta_t = \sum_{i=1}^{t} \alpha_i^{(t)} \varphi(x_i),$$

with $\alpha^{(0)} = 0$, and the recursion in $\alpha$ as

$$\alpha_i^{(t)} = (1 - \gamma_t \lambda)\alpha_i^{(t-1)} \text{ for } i \in \{1, \dots, t-1\}, \text{ and } \alpha_t^{(t)} = -\gamma_t \ell'\Big(y_t, , \sum_{i=1}^{t-1} \alpha_i^{(t-1)} k(x_t, x_i)\Big).$$

The complexity after $t$ iterations is $O(t^2)$ kernel evaluations. The convergence rates from Chapter 5 apply. More precisely, if the loss is $G$-Lipschitz continuous, then, for $F(\theta) = \mathbb{E}\big[\ell(y, \langle \varphi(x), \theta \rangle)\big] + \frac{\lambda}{2}\|\theta\|^2$, we have, for the averaged iterate $\theta_t$ (from Prop. 5.8):

$$\mathbb{E}\big[F(\bar\theta_t)\big] - \inf_{\theta \in \mathcal{H}} F(\theta) \leqslant \frac{2G^2R^2(1 + \log t)}{\lambda t}.$$

When doing a single pass with $t = n$, then $F(\theta)$ is the regularized expected risk, and we obtain a generalization bound, leading to $\mathbb{E}\big[\mathcal{R}(f_{\bar\theta_t})\big] \leqslant \frac{G^2R^2}{\lambda n} + \inf_{f \in \mathcal{H}} \big\{\mathcal{R}(f) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2\big\}$. These bounds are similar to the ones in Section 7.5 below (which assume a regularized empirical risk minimizer is available).

### 7.4.6 "Kernelization" of linear algorithms

Beyond supervised learning, many unsupervised learning algorithms can be "kernelized," such as principal component analysis (as presented in Section 3.9), $K$-means, or canonical correlation analysis.[12] Indeed, these algorithms can be cast only through the matrices of dot-products between observations and can thus be applied after the feature transformation $\varphi : \mathcal{X} \to \mathcal{H}$, and run implicitly only using the kernel function $k(x, x') = \langle \varphi(x), \varphi(x')\rangle$. See Schölkopf and Smola (2001); Shawe-Taylor and Cristianini (2004) for details and exercises below.

[12] See https://en.wikipedia.org/wiki/Canonical_correlation.

**Exercise 7.8 (Kernel principal component analysis)** *We consider $n$ observations $x_1, \dots, x_n$ in a set $\mathcal{X}$ equipped with a positive definite kernel and feature map $\varphi$ from $\mathcal{X}$ to $\mathcal{H}$. Show that the largest eigenvector of the empirical non-centered covariance operator $\frac{1}{n}\sum_{i=1}^n \varphi(x_i) \otimes \varphi(x_i)$ is proportional to $\sum_{i=1}^n \alpha_i \varphi(x_i)$ where $\alpha \in \mathbb{R}^n$ is an eigenvector of the $n \times n$ kernel matrix associated with the largest eigenvalue. Given the RKHS $\mathcal{H}$ associated with the kernel $k$, relate this eigenvalue problem to the maximizer of $\frac{1}{n}\sum_{i=1}^n f(x_i)^2$ subject to $\|f\|_{\mathcal{H}} = 1$.*

**Exercise 7.9 (Kernel $K$-means)** *Show that the $K$-means clustering algorithm[13] can be expressed only using dot-products.*

**Exercise 7.10 (Kernel quadrature)** *We consider a probability distribution $p$ on a set $\mathcal{X}$ equipped with a positive definite kernel $k$ with feature map $\varphi : \mathcal{X} \to \mathcal{H}$. For a function $f$ which is linear in $\varphi$, we want to approximate $\int_{\mathcal{X}} f(x)dp(x)$ from a linear combination $\sum_{i=1}^n \alpha_i f(x_i)$ with $\alpha \in \mathbb{R}^n$.*
*(a) Show that*

$$\Big| \int_{\mathcal{X}} f(x)dp(x) - \sum_{i=1}^n \alpha_i f(x_i) \Big| \leqslant \|f\| \cdot \Big\| \int_{\mathcal{X}} \varphi(x)dp(x) - \sum_{i=1}^n \alpha_i \varphi(x_i) \Big\|.$$

*(b) Express the square of the right-hand side with the kernel function and show how to minimize with respect to $\alpha \in \mathbb{R}^n$.*
*(c) Show that if the points $x_1, \dots, x_n$ are sampled i.i.d. from $p$ and $\alpha_i = 1/n$ for all $i$, then $\mathbb{E}\big\| \int_{\mathcal{X}} \varphi(x)dp(x) - \sum_{i=1}^n \alpha_i \varphi(x_i) \big\|^2 \leqslant \frac{1}{n}\mathbb{E}[k(x,x)]$.*

**Exercise 7.11** *Consider a binary classification problems with data $(x_1, y_1), \dots, (x_n, y_n)$ in $\mathcal{X} \times \{-1, 1\}$, with a positive kernel $k$ defined on $\mathcal{X}$ with feature map $\varphi : \mathcal{X} \to \mathcal{H}$. Let $\mu_+$ (resp. $\mu_-$) be the mean of all feature vectors for positive (resp. negative) labels. We consider the classification rule that predicts $1$ if $\|\varphi(x) - \mu_+\|_{\mathcal{H}}^2 < \|\varphi(x) - \mu_-\|_{\mathcal{H}}^2$ and $-1$ otherwise. Compute the classification rule only using kernel functions and compare it to local averaging methods from Chapter 6.*

## 7.5 Generalization guarantees - Lipschitz-continuous losses

In this section, we consider a $G$-Lipschitz-continuous loss function, and consider a minimizer $\hat{f}_D^{(c)}$ of the constrained problem

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) \text{ such that } \|f\|_{\mathcal{H}} \leqslant D, \tag{7.6}$$

[13] See https://en.wikipedia.org/wiki/K-means_clustering.

and the unique minimizer $\hat{f}_\lambda^{(r)}$ of the regularized problem

$$\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i)) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2. \tag{7.7}$$

We denote by $\mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big]$ the expected risk, and by $f^*$ one of its minimizers (which we assume to be square integrable). We assume $k(x,x) \leqslant R^2$ almost surely.

We can first relate the excess risk to the $L_2$-norm of $f - f^*$, as (using Jensen's inequality)

$$\begin{aligned}\mathcal{R}(f) - \mathcal{R}(f^*) &\leqslant \mathbb{E}[|\ell(y, f(x)) - \ell(y, f^*(x))|] \leqslant G\,\mathbb{E}[|f(x) - f^*(x)|] \\ &\leqslant G\sqrt{\mathbb{E}[|f(x) - f^*(x)|^2]} = G\|f - f^*\|_{L_2(p)},\end{aligned}$$

that is, the excess risk is dominated by the $L_2(p)$-norm of $f - f^*$. For $\mathcal{X} = \mathbb{R}^d$, and probability measures with bounded density with respect to the Lebesgue measure, we have shown that $\|f\|_{L_2(p)} \leqslant \big\|\frac{dp}{dx}\big\|_\infty^{1/2}\|f\|_{L_2(\mathbb{R}^d)}$, so we can replace in upper-bounds the quantity $G\|f - f^*\|_{L_2(p)}$ by $G\big\|\frac{dp}{dx}\big\|_\infty^{1/2}\|f - f^*\|_{L_2(\mathbb{R}^d)}$.

### 7.5.1 Risk decomposition

We now assume that $\sup_{x\in\mathcal{X}} k(x,x) \leqslant R^2$, compatible with the convention in earlier chapters on linear models (e.g., Section 4.5.3) that $\|\varphi(x)\|_{\mathcal{H}}^2 \leqslant R^2$ for all $x \in \mathcal{X}$.

**Constrained problem.** Dimension-free results from Chapter 4 (Prop. 4.5), based on Rademacher complexities, immediately apply, and we obtain that the estimation error is bounded from above by $\frac{4GDR}{\sqrt{n}}$, leading to:

$$\mathbb{E}\big[\mathcal{R}(\hat{f}_D^{(c)})\big] - \mathcal{R}(f^*) \leqslant \frac{4GDR}{\sqrt{n}} + G\inf_{\|f\|_{\mathcal{H}}\leqslant D}\|f - f^*\|_{L_2(p)},$$

(the first term is the estimation error of using the empirical risk minimizer constrained to the ball of RKHS norm less than $D$, the second term is the approximation error).

To find the optimal $D$ (to balance estimation and approximation error), we can minimize the bound with respect to $D$, leading to (using $|a| + |b| \leqslant \sqrt{2(a^2+b^2)}$):

$$\begin{aligned}\inf_{D\geqslant 0}\frac{4GRD}{\sqrt{n}} + G\inf_{\|f\|_{\mathcal{H}}\leqslant D}\|f - f^*\|_{L_2(p)} &= \inf_{f\in\mathcal{H}}\frac{4GR\|f\|_{\mathcal{H}}}{\sqrt{n}} + G\|f - f^*\|_{L_2(p)} \\ &\leqslant G\sqrt{2\inf_{f\in\mathcal{H}}\Big\{\|f - f^*\|_{L_2(p)}^2 + \frac{16R^2}{n}\|f\|_{\mathcal{H}}^2\Big\}}.\end{aligned} \tag{7.8}$$

Note that if we consider $D$ equal to $\frac{\sqrt{n}}{4R}\sqrt{\inf_{f\in\mathcal{H}}\big\{\|f - f^*\|_{L_2(p)}^2 + \frac{16R^2}{n}\|f\|_{\mathcal{H}}^2\big\}}$, we can obtain a bound proportional to what we obtained above.

Overall, we need to understand how the deterministic quantity

$$A(\mu, f^*) = \inf_{f \in \mathcal{H}} \big\{ \|f - f^*\|^2_{L_2(p)} + \mu \|f\|^2_{\mathcal{H}} \big\}$$

goes to zero when $\mu$ goes to zero (note that we define $A(\mu, f^*)$ above through a regularized estimation problem to study trade-offs between estimation and approximation errors, and that this is not a justification to use $16R^2/n$ as a regularization parameter in practice). A few situations are possible:

- If the target function $f^*$ happens to be in $\mathcal{H}$ (well-specified problem), then we have $A(\mu, f^*) \leqslant \mu \|f^*\|^2_{\mathcal{H}}$, and thus it tends to zero as $O(\mu)$. This is the best-case scenario and requires that the target function is sufficiently regular (e.g., with at least $d/2$ derivatives for $\mathcal{X} = \mathbb{R}^d$). Then, using it with $\mu = 16R^2/n$ above, the overall excess risk goes to zero as $O(1/\sqrt{n})$. Moreover, the suggested value of $D$ not surprisingly is exactly $\|f^*\|_{\mathcal{H}}$.
- The target function $f^*$ is not in $\mathcal{H}$ (mis-specified problem), but can be approached arbitrary closely in $L_2(p)$-norm by a function in $\mathcal{H}$; in other words, $f^*$ is in the closure of $\mathcal{H}$ in $L_2(p)$. In this situation, $A(\mu, f^*)$ goes to zero as $\mu$ goes to zero, but without an explicit rate if no further assumptions are made.

  For $\mathcal{X} = \mathbb{R}^d$, and the distribution $p$ of inputs with a bounded density with respect to the Lebesgue measure, and for the translation-invariant kernels from Section 7.3.3, this closure includes all of $L_2(\mathbb{R}^d)$, so this case includes most potential functions. See Section 7.5.2 for explicit rates.
- Otherwise, denoting $\Pi_{\bar{\mathcal{H}}}(f^*)$ the orthogonal projection in $L_2(p)$ of $f^*$ on the closure of $\mathcal{H}$, by the Pythagorean theorem, $A(\mu, f^*) = A(\mu, \Pi_{\bar{\mathcal{H}}}(f^*)) + \|f^* - \Pi_{\bar{\mathcal{H}}}(f^*)\|^2_{L_2(p)}$, that is, there is an incompressible error due to a choice of function space which is not large enough.

Note that we will use the same reasoning for neural networks in Section 9.4.

**Regularized problem (♦).** For the regularized problem, we can use the bound from Chapter 4 (Prop. 4.6):

$$\mathbb{E}\big[\mathcal{R}(\hat{f}^{(r)}_\lambda)\big] - \mathcal{R}(f^*) \quad \leqslant \quad \frac{32G^2R^2}{\lambda n} + \inf_{f \in \mathcal{H}} \Big\{ G\|f - f^*\|_{L_2(p)} + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}} \Big\}.$$

We can now minimize the bound with respect to $\lambda$, with $f$ fixed, as $\lambda = \frac{8RG}{\|f\|_{\mathcal{H}}\sqrt{n}}$, to obtain the bound:

$$G \inf_{f \in \mathcal{H}} \Big\{ \|f - f^*\|_{L_2(p)} + \frac{8R}{\sqrt{n}}\|f\|_{\mathcal{H}} \Big\} \leqslant G\sqrt{2 \inf_{f \in \mathcal{H}} \big\{ \|f - f^*\|^2_{L_2(p)} + \frac{64R^2}{n}\|f\|^2_{\mathcal{H}} \big\}},$$

which is the same bound as for the constrained problem but on a more commonly used optimization problem in practice. Note that for well-specified problems, the suggested regularization parameter is $\lambda = \frac{8RG}{\|f^*\|_{\mathcal{H}}\sqrt{n}}$.

### 7.5.2 Approximation error for translation-invariant kernels on $\mathbb{R}^d$

We first start by analyzing kernel methods' approximation error for translation invariant kernels. Given a distribution $p$ of inputs, the goal is to compute

$$A(\mu, f^*) = \inf_{f \in \mathcal{H}} \|f - f^*\|^2_{L_2(p)} + \mu \|f\|^2_{\mathcal{H}},$$

where $f^*$ is the target function (e.g., the minimizer of the test risk), which we assume is squared-integrable. If $A(\mu, f^*)$ tends to zero when $\mu$ tends to zero for any fixed $f^*$, kernel-based supervised learning leads to universally consistent algorithms.

We assume that $\|f - f^*\|^2_{L_2(p)} \leqslant \frac{C}{r^d}\|f - f^*\|^2_{L_2(\mathbb{R}^d)}$ (e.g., with $C = r^d \|dp/dx\|_\infty$ where $dp/dx$ is the density of $p$), where we have introduced a constant $r$ to preserve homogeneity. Moreover, for simplicity, we assume that $\|f^*\|_{L_2(\mathbb{R}^d)}$ is finite (which implies that $f^*$ has to go to zero at infinity). We now give bounds on

$$\widetilde{A}(\mu, f^*) = \inf_{f \in \mathcal{H}} \frac{1}{r^d}\|f - f^*\|^2_{L_2(\mathbb{R}^d)} + \mu \|f\|^2_{\mathcal{H}},$$

keeping in mind that $A(\mu, f^*) \leqslant C\widetilde{A}(\mu/C, f^*)$. Remember from Section 7.5.1 that if $f^* \in \mathcal{H}$ (best case scenario), then both $A(\mu, f^*)$ and $\widetilde{A}(\mu, f^*)$ are less than $\mu\|f^*\|^2_{\mathcal{H}}$.

**Explicit approximation.** We have, for translation-invariant kernels defined in Section 7.3.3, an explicit formulation of the norm $\|\cdot\|_{\mathcal{H}}$ as $\|f\|^2_{\mathcal{H}} = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} d\omega$, and thus

$$\widetilde{A}(\mu, f^*) = \inf_{\hat{f} \in L_2(\mathbb{R}^d)} \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \Big[\frac{1}{r^d}|\hat{f}(\omega) - \hat{f}^*(\omega)|^2 + \mu \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)}\Big] d\omega.$$

The optimization can be performed independently for each $\omega$, which is a quadratic problem. Setting the derivative with respect to $\hat{f}(\omega)$ to zero leads to $0 = 2\frac{1}{r^d}(\hat{f}(\omega) - \hat{f}^*(\omega)) + 2\mu\frac{\hat{f}(\omega)}{\hat{q}(\omega)}$, and thus $\hat{f}_\mu(\omega) = \frac{\hat{f}^*(\omega)}{1+\mu r^d \hat{q}(\omega)^{-1}}$. In terms of the objective function, we get:

$$\widetilde{A}(\mu, f^*) = \frac{1}{(2\pi r)^d}\int_{\mathbb{R}^d} |\hat{f}^*(\omega)|^2 \Big(1 - \frac{1}{1+\mu r^d \hat{q}(\omega)^{-1}}\Big) d\omega = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} |\hat{f}^*(\omega)|^2 \frac{\mu}{\hat{q}(\omega) + \mu r^d} d\omega.$$

When $\mu$ goes to zero, we see that for each $\omega$, $\hat{f}_\mu(\omega)$ tends to $\hat{f}^*(\omega)$. By the dominated convergence theorem, $\widetilde{A}(\mu, f^*)$ goes to zero when $\mu$ goes to zero.

Without further assumptions, it is impossible to obtain a convergence rate (otherwise, the no-free lunch theorem from Chapter 2 would be invalidated). However, this is possible when assuming regularity properties for $f^*$.

⚠ Note that the universal approximation properties of translation-invariant kernels do not require the kernel bandwidth $r$ to go to zero (as opposed to smoothing kernels from Chapter 6).

**Sobolev spaces (♦).** If we assume that

$$\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}(1+r^2\|\omega\|_2^2)^t|\hat{f}^*(\omega)|^2 d\omega < +\infty \tag{7.9}$$

for some $t > 0$, that is, $f^*$ with squared integrable partial derivatives up to order $t$, then we can further bound:

$$\widetilde{A}(\mu, f^*) \quad \leqslant \quad \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}(1+r^2\|\omega\|_2^2)^t|\hat{f}^*(\omega)|^2 d\omega \times \sup_{\omega\in\mathbb{R}^d}\Big\{\frac{\mu}{\hat{q}(\omega)+\mu r^d}\frac{1}{(1+r^2\|\omega\|_2^2)^t}\Big\}.$$

If we now assume $\hat{q}(\omega) \propto r^d(1+r^2\|\omega\|_2^2)^{-s}$ (Matern kernels), with $s > d/2$ to get an RKHS, then with $t \geqslant s$, $f^* \in \mathcal{H}$, and have $\widetilde{A}(\mu, f^*) = \mu\|f^*\|_{\mathcal{H}}^2$. With $t < s$, that is the function is not inside the RKHS $\mathcal{H}$, then we get a bound proportional to (using $a+b \geqslant \frac{t}{s}a + (1-\frac{t}{s})b \geqslant a^{t/s}b^{1-t/s}$):

$$\sup_{\omega\in\mathbb{R}^d}\Big\{\frac{\mu}{\hat{q}(\omega)+\mu r^d}\frac{1}{(1+r^2\|\omega\|_2^2)^t}\Big\} \leqslant \sup_{\omega\in\mathbb{R}^d}\Big\{\frac{\mu}{\hat{q}(\omega)^{t/s}(\mu r^d)^{1-t/s}}\frac{1}{(1+r^2\|\omega\|_2^2)^t}\Big\} = O(\mu^{t/s}).$$

**Exercise 7.12** *(♦) Find an upper-bound of $\widetilde{A}(\mu, f^*)$ for the same assumption on $f^*$ but with the Gaussian kernel.*

⚠ There are two regularities, with two different constraints: $t \geqslant 0$ for the target function, and $s > d/2$ for the kernel.

**Putting things together.** Thus, for Lipschitz-continuous losses and target functions that satisfy Eq. (7.9), we get an expected excess risk of the order $\big(\widetilde{A}(R^2/n, f^*)\big)^{1/2} = O(n^{-t/(2s)})$, when $t \leqslant s$. For example, when $t = 1$, that is, only first-order derivatives are assumed to be square integrable, then for $s = d/2+1/2$ (exponential kernel), we obtain a rate of $O(n^{-1/(d+1)})$, which is similar to the rate obtained with local averaging techniques in Chapter 6 (note here that we are in Lipschitz-loss set-up, which leads to worse rates, see the square loss in Section 7.6). Thus, kernel methods do not escape the curse of dimensionality (which is unavoidable anyway). *However*, with the proper choice of the regularization parameter, they can benefit from extra smoothness of the target function: in the very favorable case, where $f^* \in \mathcal{H}$, that is $t \geqslant s$, then we obtain a dimension-independent rate of $1/\sqrt{n}$. In intermediate scenarios, the rates are in between. This is why kernel methods are said to be *adaptive to the smoothness* of the target function.

**Approximation bounds (♦).** In some analysis set-ups (such as those explored in Chapter 9), it is required to approximate some $f^*$ up to $\varepsilon$ with the minimum possible RKHS norm. This can be done as follows.

A bound on the quantity $A(\mu, f^*) = \inf_{f\in\mathcal{H}}\big\{\|f-f^*\|_{L_2(p)}^2 + \mu\|f\|_{\mathcal{H}}^2\big\}$ of the form $c\mu^\alpha$

for $\alpha \in (0,1)$ leads to the following bound:

$$\begin{aligned} & \inf_{f\in\mathcal{H}} \|f\|_{\mathcal{H}}^2 \text{ such that } \|f-f^*\|_{L_2(p)} \leqslant \varepsilon \\ = & \inf_{f\in\mathcal{H}} \sup_{\mu\geqslant 0} \|f\|_{\mathcal{H}}^2 + \mu(\|f-f^*\|_{L_2(p)}^2 - \varepsilon^2) \text{ using Lagrangian duality,} \\ = & \sup_{\mu\geqslant 0} \mu A(\mu^{-1}, f^*) - \mu\varepsilon^2 \leqslant \sup_{\mu\geqslant 0} \mu c \mu^{-\alpha} - \mu\varepsilon^2. \end{aligned}$$

The optimal $\mu$ is such that $(1-\alpha)c\mu^{-\alpha} = \varepsilon^2$, leading to an approximation bound proportional to $\varepsilon^{2(1-1/\alpha)} = \varepsilon^{-2(1-\alpha)/\alpha}$.

Applied to $\alpha = t/s$ like before, this leads to an RKHS norm proportional to $\varepsilon^{-(1-\alpha)/\alpha}$ to get an error less than $\|f-f^*\|_{L_2(\mathbb{R}^d)}$ So when $t=1$ (single derivative for the target function), and $s > d/2$ (for the Sobolev kernel), we get a norm of the order $\varepsilon^{-(1/\alpha-1)} = \varepsilon^{-(s-1)} \geqslant \varepsilon^{-d/2+1}$, which explodes exponentially in dimension, which is another way of formulating the curse of dimensionality.

**Relationship between Lipschitz-continuous functions and Sobolev spaces on $\mathbb{R}^d$ (♦♦).** In the previous chapter on local averaging methods, as well as for neural networks (Chapter 9), we will consider Lipschitz-continuous functions on a subset of $\mathbb{R}^d$, which we take here to be the ball of center 0 and radius $r$. To apply results from the current chapter, we need to extend them to a function $g$ on $\mathbb{R}^d$ with controlled squared Sobolev norm with order $t=1$, that is, $\int_{\mathbb{R}^d} \big(|g(x)|^2 + r^2\|g'(x)\|_2^2\big)dx$. Then, the estimation rates for Sobolev space of order $t$, that is, $O(n^{-1/(1+d)})$, applies to Lipshitz-continuous functions on an Euclidean ball.

For this, we also need to impose a bound on the value of $f$ at 0, that is, we assume $|f(0)| \leqslant rD$, and $f$ is $D$-Lipschitz-continuous on the ball of center 0 and radius $r$. We now show that we can extend it to a function $g$ with squared Sobolev norm less than a constant $c_d$ (that depends on $d$) times $R^{d+2}D^2$.

We define the function $g$ which is equal to $f$ on the ball of radius $r$, equal to 0 outside of the ball of radius $2r$, and equal to $g(x) = f(rx/\|x\|_2)(2-\|x\|_2/r)$ for $\|x\|_2 \in [r,2r]$, that is, on each ray $\{ty, t\in[r,2r]\}$, for $y \in \mathbb{R}^d$ of unit norm, the function $g$ goes linearly from $f(y)$ to 0. The function $g$ is continuous and has almost everywhere bounded derivatives. On the ball of radius $2r$, $|g(x)| \leqslant 2rD$, while when $\|x\|_2 \in [r,2r]$, $g'(x) = -\frac{1}{r}f(rx/\|x\|_2)x/\|x\|_2 + \frac{r}{\|x\|_2}(I - xx^\top/\|x\|_2^2)f'(rx/\|x\|_2)(2-\|x\|_2/r)$, leading to, by the Pythagorean theorem, $\|g'(x)\|_2^2 = \frac{1}{r^2}|f(rx/\|x\|_2)|^2 + \frac{r^2}{\|x\|_2^2}(2-\|x\|_2/r)^2\|(I - xx^\top/\|x\|_2^2)f'(rx/\|x\|_2)\|_2^2 \leqslant \frac{1}{r^2}|2rD|^2 + D^2 = 5D^2$. Thus, $\int_{\mathbb{R}^d}\big(|g(x)|^2 + r^2\|g'(x)\|_2^2\big)dx \leqslant 9r^2D^2(2r)^d\frac{\pi^{d/2}}{\Gamma(1+d/2)}$, since the volume of the Euclidean unit ball is equal to $\frac{\pi^{d/2}}{\Gamma(1+d/2)}$. Thus the constant $c_d$ is less than $\frac{9\cdot 2^d\pi^{d/2}}{\Gamma(1+d/2)}$.

## 7.6 Theoretical analysis of ridge regression (♦)

In this section, we provide finer results for ridge regression (that is, square loss and penalization by squared norm) used within kernel methods. Compared to the analysis performed in Section 3.6, there are three difficulties:

(1) we go from fixed design to random design: this will require finer probabilistic arguments to relate population and empirical covariance operators,

(2) we need to go infinite-dimensional: in terms of notations, this will mean not using transposes of matrices, but dot-products, which is a minor modification,

(3) the infimum of the expected risk over linear functions parameterized by $\theta \in \mathcal{H}$ may not be attained by an element of $\mathcal{H}$, but by an element of its closure in $L_2(p)$. This is important, as this allows access to a potentially large set of functions and requires more care.

### 7.6.1 Kernel ridge regression as a "linear" estimator

We consider $n$ i.i.d. observations $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, $i = 1, \dots, n$, and we aim at minimizing, for $\lambda > 0$,

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

Like local averaging methods in Chapter 6, the ridge regression estimator happens to be a "linear" estimator that depends linearly on the response vector (but of course non-linearly in $x$ in general). Indeed, using the representer theorem from Eq. (7.2), the estimator is $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$, with $\alpha \in \mathbb{R}^n$ defined as $\alpha = (K + n\lambda I)^{-1} y$, where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix. We can then write

$$f(x) = \sum_{i=1}^{n} \hat{w}_i(x) y_i,$$

with $\hat{w}(x) = (K + n\lambda I)^{-1} q(x) \in \mathbb{R}^n$, where $q(x) \in \mathbb{R}^n$ is defined as $q_i(x) = k(x, x_i)$. The smoothing matrix $H$ is then equal to $H = K(K + n\lambda I)^{-1}$.

The key differences with local averaging are that (a) the weights do not sum to one, that is, $\sum_{i=1}^n \hat{w}_i(x)$ may be different from one, and (b) the weights are not constrained to be non-negative. While the first difference can be removed using centering (see exercise below), the second one is more fundamental: allowing the weights to be negative will enable the adaptivity to smoothness, which local averaging methods missed (see Section 6.5).

**Exercise 7.13** *We consider the optimization problem $\frac{1}{2n}\|y - \Phi\theta - \eta 1_n\|_2^2 + \frac{\lambda}{2}\|\theta\|_2^2$, where $\Phi \in \mathbb{R}^{n \times d}$ is the design matrix obtained from feature map $\varphi$ and data points $x_1, \dots, x_n$, and $y \in \mathbb{R}^n$, and $1_n \in \mathbb{R}^n$ is the vector of all ones. Show that the optimal value of $\theta$ and $\eta$ are: $\theta = \Phi^\top \alpha$, and $\eta = \frac{1}{n} 1_n^\top (y - \Phi\theta)$, with $\alpha = \Pi_n(\Pi_n K \Pi_n + n\lambda I)^{-1}\Pi_n y$, and $\Pi_n = I - \frac{1}{n} 1_n 1_n^\top$. Show that the prediction function $f(x) = \varphi(x)^\top \theta + \eta$ is of the form $\sum_{i=1}^n \hat{w}_i(x) y_i$ with weights that sum to one.*

**Exercise 7.14** *(♦) For $x_1, \dots, x_n$ equally spaced in $[0,1]$ and for a translation-invariant kernel from Section 7.3.2, compute the eigenvalues of the kernel matrix and the smoothing matrix.*

### 7.6.2 Bias and variance decomposition (♦)

**Beyond fixed-design finite-dimensional analysis.** In Chapter 3, we considered ridge regression in the fixed design setting (where the input data were assumed deterministic) and a finite-dimensional feature space $\mathcal{H}$, and obtained in Prop. 3.7 the following *exact* expression of the excess risk of the ridge regression estimator $\hat\theta_\lambda$, assuming $y_i = \langle \theta_*, \varphi(x_i)\rangle + \varepsilon_i$, with $\varepsilon_i$ independent from $x_i$, and where $\mathbb{E}[\varepsilon_i] = 0$, $\mathbb{E}[\varepsilon_i^2] = \sigma^2$:

$$\mathbb{E}\big[(\hat\theta_\lambda - \theta_*)^\top \widehat\Sigma(\hat\theta_\lambda - \theta_*)\big] = \lambda^2 \theta_*^\top (\widehat\Sigma + \lambda I)^{-2}\widehat\Sigma\theta_* + \frac{\sigma^2}{n}\operatorname{tr}\big[\widehat\Sigma^2(\widehat\Sigma + \lambda I)^{-2}\big]. \tag{7.10}$$

For the random design assumption (the usual machine learning setting), we first need to obtain a value for the expected risk. Moreover, we need to replace the matrix notation to apply to infinite dimensional $\mathcal{H}$ where the minimizer has a potentially infinite norm.

**Modeling assumptions.** We assume that

$$y_i = f^*(x_i) + \varepsilon_i,$$

with for simplicity $\mathbb{E}[\varepsilon_i|x_i] = 0$, and $\mathbb{E}[\varepsilon_i^2|x_i] \leqslant \sigma^2$ almost surely, for some target function $f^* \in L_2(p)$, so that $f^*(x) = \mathbb{E}[y|x]$ is exactly the conditional expectation of $y|x$.

The target function $f^*$ may not be in $\mathcal{H}$. All dot-products will always be in $\mathcal{H}$, while we will specify the corresponding space for norms.

We thus consider the optimization problem:

$$\min_{f \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2, \tag{7.11}$$

with the solution found with algorithms in Section 7.4.

The theoretical analysis of kernel methods typically does not involve the parameters $\alpha \in \mathbb{R}^n$ obtained from the representer theorem.

We have, with $\widehat\Sigma = \frac{1}{n}\sum_{i=1}^n \varphi(x_i) \otimes \varphi(x_i)$ a self-adjoint operator from $\mathcal{H}$ to $\mathcal{H}$ (the empirical covariance operator), a cost function equal to

$$\frac{1}{n}\sum_{i=1}^n y_i^2 + \langle f, \widehat\Sigma f\rangle - 2\Big\langle \frac{1}{n}\sum_{i=1}^n y_i\varphi(x_i), f\Big\rangle + \lambda\langle f, f\rangle,$$

leading to the minimizer $\hat{f}_\lambda$ of Eq. (7.11) equal to:

$$\hat{f}_\lambda = (\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n y_i \varphi(x_i) = (\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n f^*(x_i)\varphi(x_i) + (\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i).$$

We can now compute the (expected) excess risk equal to $\mathbb{E}\big[\|\hat{f}_\lambda - f^*\|^2_{L_2(p)}\big]$ as (using that $\mathbb{E}(\varepsilon_i|x_i) = 0$):

$$\begin{aligned} &\mathbb{E}\big[\|\hat{f}_\lambda - f^*\|^2_{L_2(p)}\big] \\ &= \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i)\Big\|^2_{L_2(p)}\Big] + \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n f^*(x_i)\varphi(x_i) - f^*\Big\|^2_{L_2(p)}\Big]. \end{aligned}$$

The first term is the usual variance term (that depends on the noise on top of the optimal predictions). In contrast, the second is the (squared) bias term (which depends on the regularity of the target function). Before developing the probabilistic argument, we give simplified upper bounds of the two terms.

On top of the non-centered empirical covariance operator $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n \varphi(x_i) \otimes \varphi(x_i)$, we will need its expectation, the covariance operator (from $\mathcal{H}$ to $\mathcal{H}$)

$$\Sigma = \mathbb{E}\big[\varphi(x) \otimes \varphi(x)\big],$$

for the corresponding distribution of the $x_i$'s. A key property relates the $L_2(p)$-norm and the RKHS norm, that is, that for $g \in \mathcal{H}$,

$$\begin{aligned} \|g\|^2_{L_2(p)} &= \int_{\mathcal{X}} g(x)^2 dp(x) = \int_{\mathcal{X}} \langle g, \varphi(x)\rangle^2 dp(x) = \int_{\mathcal{X}} \langle g, \varphi(x) \otimes \varphi(x) g\rangle dp(x) \\ &= \langle g, \Sigma g\rangle = \|\Sigma^{1/2} g\|^2_{\mathcal{H}}. \end{aligned} \tag{7.12}$$

**Variance term.** Starting from variance $= \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1}\frac{1}{n}\sum_{i=1}^n \varepsilon_i\varphi(x_i)\Big\|^2_{L_2(p)}\Big]$, the variance term can be upper-bounded as follows (first using independence and zero means of the variables $\varepsilon_i$). Below, we use the property that for symmetric matrices such that $A \succcurlyeq 0$ and $B \preccurlyeq C$, we have $\mathrm{tr}[AB] \leqslant \mathrm{tr}[AC]$, and Eq. (7.12):

$$\begin{aligned} &\mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i)\Big\|^2_{L_2(p)}\Big] \\ &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}\Big[\mathrm{tr}\,\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma(\widehat{\Sigma} + \lambda I)^{-1}\varepsilon_i^2 \varphi(x_i) \otimes \varphi(x_i)\big)\Big] \\ &\leqslant \frac{\sigma^2}{n}\mathbb{E}\big[\mathrm{tr}\,\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma(\widehat{\Sigma} + \lambda I)^{-1}\widehat{\Sigma}\big)\big] \text{ using } \mathbb{E}[\varepsilon_i^2|x_i] \leqslant \sigma^2, \\ &\leqslant \frac{\sigma^2}{n}\mathbb{E}\Big[\mathrm{tr}\,\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma\big)\Big] \text{ using } (\widehat{\Sigma} + \lambda I)^{-1}\widehat{\Sigma} \preccurlyeq I. \end{aligned} \tag{7.13}$$

This will be the main expression we will bound later.

**Bias term.** We first assume that $f^* \in \mathcal{H}$, that is, the model is well-specified. Then, writing $f^*(x_i) = \langle f^*, \varphi(x_i)\rangle$ (which is possible because $f^* \in \mathcal{H}$), the bias term is equal to

$$\begin{aligned}
\text{bias} &= \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1}\frac{1}{n}\sum_{i=1}^{n} f^*(x_i)\varphi(x_i) - f^*\Big\|_{L_2(p)}^2\Big] && (7.14)\\
&= \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1}\frac{1}{n}\sum_{i=1}^{n} \langle f^*, \varphi(x_i)\rangle\varphi(x_i) - f^*\Big\|_{L_2(p)}^2\Big] && (7.15)\\
&= \mathbb{E}\big[\|(\widehat{\Sigma} + \lambda I)^{-1}\widehat{\Sigma} f^* - f^*\|_{L_2(p)}^2\big]\\
&= \mathbb{E}\Big[\|\lambda \Sigma^{1/2}(\widehat{\Sigma} + \lambda I)^{-1} f^*\|_{\mathcal{H}}^2\Big] = \lambda^2 \mathbb{E}\Big[\langle f^*, (\widehat{\Sigma} + \lambda I)^{-1}\Sigma(\widehat{\Sigma} + \lambda I)^{-1} f^*\rangle\Big], && (7.16)
\end{aligned}$$

where we have used Eq. (7.12) above to re-introduce the operator $\Sigma$. This will be the main expression we will bound later.

**Upper-bound on excess risk.** We have thus shown the following proposition:

**Proposition 7.5** *When $f^* \in \mathcal{H}$, the excess risk of the ridge regression estimator is upper-bounded by:*

$$\mathbb{E}\big[\|\hat{f}_\lambda - f^*\|_{L_2(p)}^2\big] \leqslant \frac{\sigma^2}{n}\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma\big)\Big] + \lambda^2\mathbb{E}\Big[\langle f^*, (\widehat{\Sigma} + \lambda I)^{-1}\Sigma(\widehat{\Sigma} + \lambda I)^{-1} f^*\rangle\Big]. \tag{7.17}$$

Given the expression of the expected variance in Eq. (7.13) and the expected bias in Eq. (7.16), we notice that both the empirical and expected covariance operators appear and that it would be important to replace the empirical one with the expected one. This is possible with extra multiplicative factors, which we now show. Then, we will bound the two terms separately and show how balancing them leads to interesting learning bounds.

### 7.6.3 Relating empirical and population covariance operators

We follow Mourtada and Rosasco (2022) and derive simple relationships between the empirical covariance operator $\widehat{\Sigma}$ and the population operator $\Sigma$, by showing the following lemma dealing with expectations; for high probability bounds, see, e.g., Rudi et al. (2015); Rudi and Rosasco (2017), as well as the end of Section 7.6.4.

**Lemma 7.1 (Mourtada and Rosasco, 2022)** *Assuming i.i.d. data $x_1, \dots, x_n \in \mathcal{X}$, and bounded features $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ for all $x \in \mathcal{X}$; we have, for all $g \in \mathcal{H}$:*

$$\begin{aligned}
\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma\big)\Big] &\leqslant \Big(1 + \frac{R^2}{\lambda n}\Big)\operatorname{tr}\big((\Sigma + \lambda I)^{-1}\Sigma\big) && (7.18)\\
\mathbb{E}\Big[\langle g, (\widehat{\Sigma} + \lambda I)^{-1}\Sigma(\widehat{\Sigma} + \lambda I)^{-1} g\rangle\Big] &\preccurlyeq \lambda^{-1}\Big(1 + \frac{R^2}{\lambda n}\Big)^2\langle g, (\Sigma + \lambda I)^{-1}\Sigma g\rangle. && (7.19)
\end{aligned}$$

**Proof** (♦) The main idea is to introduce a $(n+1)$-th independent observation from the same distribution, write $\Sigma = \mathbb{E}\big[\varphi(x_{n+1}) \otimes \varphi(x_{n+1})\big]$, and use the fact that the observations are "exchangeable", that is, they can be permuted without changing their joint distribution.

We denote $C = \sum_{i=1}^{n+1} \varphi(x_i) \otimes \varphi(x_i)$, and using the matrix inversion lemma (Section 1.1.3), we have

$$\begin{aligned} (C+n\lambda I)^{-1}\varphi(x_{n+1}) &= (n\widehat{\Sigma} + n\lambda I + \varphi(x_{n+1}) \otimes \varphi(x_{n+1}))^{-1}\varphi(x_{n+1}) \\ &= \frac{(n\widehat{\Sigma}+n\lambda I)^{-1}\varphi(x_{n+1})}{1 + \langle \varphi(x_{n+1}), (n\widehat{\Sigma} + n\lambda I)^{-1}\varphi(x_{n+1})\rangle}. \end{aligned} \tag{7.20}$$

Finally, we will use $c = \big\langle \varphi(x_{n+1}), \big(n\widehat{\Sigma} + n\lambda I\big)^{-1}\varphi(x_{n+1})\big\rangle \leqslant \dfrac{R^2}{\lambda n}$. To prove Eq. (7.18), we use Eq. (7.20) above to express $(\widehat{\Sigma}+\lambda I)^{-1}\varphi(x_{n+1})$ in terms of $(C+n\lambda I)^{-1}\varphi(x_{n+1})$, to get:

$$\begin{aligned} \mathbb{E}\Big[ \operatorname{tr}\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma\big)\Big] &= \mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma} + \lambda I)^{-1}\varphi(x_{n+1}) \otimes \varphi(x_{n+1})\big)\Big] \\ &= \mathbb{E}\Big[\big\langle \varphi(x_{n+1}), (\widehat{\Sigma} + \lambda I)^{-1}\varphi(x_{n+1})\big\rangle\Big] \\ &= n\mathbb{E}\Big[(1+c)\big\langle\varphi(x_{n+1}), (C + n\lambda I)^{-1}\varphi(x_{n+1})\big\rangle\Big], \end{aligned}$$

which leads to $\mathbb{E}\big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\big] \leqslant \big(1+\frac{R^2}{\lambda n}\big)\mathbb{E}\big[\big\langle\varphi(x_{n+1}), (C+n\lambda I)^{-1}\varphi(x_{n+1})\big\rangle\big]$. Thus, using that the variables $(x_1, \dots, x_{n+1})$ are exchangeable:

$$\begin{aligned} &\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma} + \lambda I)^{-1}\Sigma\big)\Big] \\ \leqslant\ & \Big(1 + \frac{R^2}{\lambda n}\Big)\frac{1}{n+1}\sum_{i=1}^{n+1}\mathbb{E}\big[\big\langle\varphi(x_i), \big(C + n\lambda I\big)^{-1}\varphi(x_i)\big\rangle\big] \\ =\ & \Big(1 + \frac{R^2}{\lambda n}\Big)\frac{1}{n+1}\mathbb{E}\big[\operatorname{tr}\big(C(C + n\lambda I)^{-1}\big)\big] \text{ since } C = \sum_{i=1}^{n+1}\varphi(x_i) \otimes \varphi(x_i) \\ \leqslant\ & \Big(1 + \frac{R^2}{\lambda n}\Big)\frac{1}{n+1}\big[\operatorname{tr}\big(\mathbb{E}[C](\mathbb{E}[C] + n\lambda I)^{-1}\big)\big] \text{ by Jensen's inequality,}^{14} \\ =\ & \Big(1 + \frac{R^2}{\lambda n}\Big)\frac{1}{n+1}\operatorname{tr}\big((n+1)\Sigma((n+1)\Sigma + n\lambda I)^{-1}\big) \\ \leqslant\ & \Big(1 + \frac{R^2}{\lambda n}\Big)\operatorname{tr}\big(\Sigma(\Sigma + \lambda I)^{-1}\big), \text{ which is exactly Eq. (7.18).} \end{aligned}$$

To prove Eq. (7.19), we use the same technique, that is,

$$\begin{aligned} &\mathbb{E}\big[(\widehat{\Sigma} + \lambda I)^{-1}\Sigma(\widehat{\Sigma} + \lambda I)^{-1}\big] = \mathbb{E}\Big[(\widehat{\Sigma} + \lambda I)^{-1}\varphi(x_n) \otimes \varphi(x_n)(\widehat{\Sigma} + \lambda I)^{-1}\Big] \\ =\ & n^2(1+c)^2\big[(C + n\lambda I)^{-1}\varphi(x_{n+1})\big] \otimes \big[(C + n\lambda I)^{-1}\varphi(x_{n+1})\big]. \end{aligned}$$

This leads to:

$$\begin{aligned}
& \mathbb{E}\big[\langle g, (\widehat{\Sigma}+\lambda I)^{-1}\Sigma(\widehat{\Sigma}+\lambda I)^{-1} g\rangle\big] \\
= \ & n^2 \mathbb{E}\Big[(1+c)^2\langle (C+n\lambda I)^{-1}\varphi(x_{n+1}), g\rangle^2\Big] \\
\leqslant \ & n^2 \big(1+\frac{R^2}{\lambda n}\big)^2 \mathbb{E}\Big[\langle (C+n\lambda I)^{-1}\varphi(x_{n+1}), g\rangle^2\Big] \\
= \ & \frac{n^2}{n+1}\big(1+\frac{R^2}{\lambda n}\big)^2 \mathbb{E}\Big[\langle g, (C+n\lambda I)^{-1} C (C+n\lambda I)^{-1} g\rangle\Big] \text{ by exchangeability,} \\
\leqslant \ & \frac{1}{\lambda}\frac{n}{n+1}\big(1+\frac{R^2}{\lambda n}\big)^2 \mathbb{E}\Big[\langle g, C (C+n\lambda I)^{-1} g\rangle\Big] \\
\leqslant \ & \frac{1}{\lambda}\frac{n}{n+1}\big(1+\frac{R^2}{\lambda n}\big)^2 \langle g, \mathbb{E}[C](\mathbb{E}[C]+n\lambda I)^{-1} g\rangle \text{ by Jensen's inequality,} \\
= \ & \frac{1}{\lambda} n \big(1+\frac{R^2}{\lambda n}\big)^2 \langle g, \Sigma((n+1)\Sigma+n\lambda I)^{-1} g\rangle \leqslant \lambda^{-1}\Big(1+\frac{R^2}{\lambda n}\Big)^2 \langle g, (\Sigma+\lambda I)^{-1}\Sigma g\rangle.
\end{aligned}$$

■

### 7.6.4 Analysis for well-specified problems (♦)

In this section, we assume that $f^* \in \mathcal{H}$. We have the following result for the excess risk, whose proof consists in applying Lemma 7.1 to Eq. (7.17).

**Proposition 7.6 (Well-specified model kernel ridge regression)** *Assume i.i.d. data $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, for $i = 1, \dots, n$, and $y_i = f^*(x_i) + \varepsilon_i$, with $\mathbb{E}[\varepsilon_i | x_i] = 0$ and $\mathbb{E}[\varepsilon_i^2 | x_i] \leqslant \sigma^2$, and $f^* \in \mathcal{H}$. Assume $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$. We have:*

$$\mathbb{E}\big[\|\hat{f}_\lambda - f^*\|^2_{L_2(p)}\big] \leqslant \frac{\sigma^2}{n}\Big(1+\frac{R^2}{\lambda n}\Big) \operatorname{tr}\big((\Sigma+\lambda I)^{-1}\Sigma\big) + \lambda\Big(1+\frac{R^2}{\lambda n}\Big)^2 \langle f^*, \Sigma(\Sigma+\lambda I)^{-1} f^*\rangle. \tag{7.21}$$

This is to be contrasted with Eq. (7.10): we obtain a similar result with $\widehat{\Sigma}$ replaced by $\Sigma$, but with some extra multiplicative constants that are close to one if $R^2/(\lambda n)$ is small. We can further bound $\operatorname{tr}\big((\Sigma+\lambda I)^{-1}\Sigma\big) \leqslant \frac{R^2}{\lambda}$ and $\langle f^*, \Sigma(\Sigma+\lambda I)^{-1} f^*\rangle \leqslant \langle f^*, f^*\rangle$, to get the bound

$$\mathbb{E}\big[\|\hat{f}_\lambda - f^*\|^2_{L_2(p)}\big] \leqslant \sigma^2\frac{R^2}{\lambda n}\Big(1+\frac{R^2}{\lambda n}\Big) + \lambda\Big(1+\frac{R^2}{\lambda n}\Big)^2 \|f^*\|^2_{\mathcal{H}},$$

which is a random design version of the developments in the proof of Prop. 3.8. In such a situation, the choice $\lambda = R^2/\sqrt{n}$ (which does not impose any knowledge of $\|f^*\|_{\mathcal{H}}$) leads to a bound on the excess risk proportional to $(\sigma^2 + R^2\|f^*\|^2_{\mathcal{H}})/\sqrt{n}$.

In finite feature dimensions $d$, we can alternatively bound $\operatorname{tr}\big((\Sigma+\lambda I)^{-1}\Sigma\big) \leqslant \frac{R^2}{\lambda}$ by $d$, then leading to a natural choice $\lambda$ of order $R^2/n$, and an upper-bound of the excess risk proportional to $\sigma^2 d/n + R^2\|f_*\|^2_{\mathcal{H}}/n$.

The last two paragraphs lead to different choices of the regularization parameter, proportional to $R^2/\sqrt{n}$ or $R^2/n$, two classical rules of thumbs within kernel methods.

**Bounds in high-probability (♦♦).** Instead of obtaining bounds in expectation (with respect to the training data), we can obtain high-probability bounds, as briefly shown below for the simplest bound; see, more refined bounds by Rudi et al. (2015); Rudi and Rosasco (2017). Note that they do not rely on Rademacher averages but on direct probabilistic arguments that can only be applied to the square loss.

**Proposition 7.7 (High-probability bound for kernel ridge regression)** *Assume i.i.d. data $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, for $i = 1, \dots, n$, and $y_i = f^*(x_i) + \varepsilon_i$, with $\mathbb{E}[\varepsilon_i | x_i] = 0$ and $\varepsilon_i^2 \leqslant \sigma^2$ almost surely, and $f^* \in \mathcal{H}$. Assume $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ and $n \geqslant \big(\frac{4}{3} + \frac{R^2}{8\lambda}\big) \log \frac{14R^2}{\lambda\delta}$. We have, with a probability greater than $1 - \delta$,*

$$\|\hat{f}_\lambda - f^*\|^2_{L_2(p)} \leqslant \frac{8\sigma^2 R^2}{\lambda n} + 4\lambda \|f^*\|^2_{\mathcal{H}} + \frac{16\sigma^2 R^2}{\lambda n} \log \frac{2}{\delta}. \tag{7.22}$$

**Proof** We first apply Prop. 1.7 with $M_i = \Sigma(\Sigma + \lambda I)^{-1} - (\Sigma + \lambda I)^{-1/2} \varphi(x_i) \otimes \varphi(x_i)(\Sigma + \lambda I)^{-1/2}$, for which we have $V = \frac{R^2}{\lambda} \Sigma(\Sigma + \lambda I)^{-1}$, $\sigma^2 = \frac{R^2}{\lambda}$, $c = 1$, and $t = 1$, leading to

$$\lambda_{\max}\big[(\Sigma + \lambda I)^{-1/2}(\Sigma - \widehat{\Sigma})(\Sigma + \lambda I)^{-1/2}\big] \leqslant \frac{1}{2}$$

with probability greater than $1 - 7\frac{R^2}{\lambda} \exp\big[-\frac{n}{4/3 + R^2/(8\lambda)}\big]$, as soon as $\frac{1}{2} \geqslant \frac{1}{3n} + \frac{R}{\sqrt{\lambda n}}$. This probability is greater than $1 - \delta/2$ as soon as $n \geqslant (4/3 + R^2/(8\lambda)) \log \frac{14R^2}{\lambda\delta}$.

This implies $\Sigma - \widehat{\Sigma} \preccurlyeq \frac{1}{2}(\Sigma + \lambda I)$, $\frac{1}{2}(\Sigma + \lambda I) \preccurlyeq \widehat{\Sigma} + \lambda I$, and thus $(\widehat{\Sigma} + \lambda I)^{-1} \preccurlyeq 2(\Sigma + \lambda I)^{-1}$. Using the Łojasiewicz inequality (Lemma 5.1) on the regularized empirical risk $\widehat{\mathcal{R}}_\lambda(f) = \frac{1}{2n}\langle f - f^*, \widehat{\Sigma}(f - f^*)\rangle - \big\langle \frac{1}{n}\sum_{i=1}^m \varepsilon_i \varphi(x_i), f\big\rangle + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}$, we get:

$$\widehat{\mathcal{R}}_\lambda(f^*) - \widehat{\mathcal{R}}_\lambda(\hat{f}_\lambda) \leqslant \frac{1}{2\lambda} \|\widehat{\mathcal{R}}'_\lambda(f^*)\|^2_{\mathcal{H}}.$$

Using $\widehat{\mathcal{R}}_\lambda(f^*) - \widehat{\mathcal{R}}_\lambda(\hat{f}_\lambda) = \frac{1}{2}\langle f^* - \hat{f}_\lambda, (\widehat{\Sigma} + \lambda I)(f^* - \hat{f}_\lambda)\rangle \geqslant \frac{1}{4}\langle f^* - \hat{f}_\lambda, (\Sigma + \lambda I)(f^* - \hat{f}_\lambda)\rangle = \frac{1}{4}\|\hat{f}_\lambda - f^*\|^2_{L_2(p)}$, we get

$$\|\hat{f}_\lambda - f^*\|^2_{L_2(p)} \leqslant \frac{2}{\lambda}\Big\|\frac{1}{n}\sum_{i=1}^m \varepsilon_i \varphi(x_i) + \lambda f^*\Big\|^2_{\mathcal{H}} \leqslant \frac{4}{\lambda}\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \varphi(x_i)\Big\|^2_{\mathcal{H}} + 4\lambda\|f^*\|^2_{\mathcal{H}}.$$

We thus need a high-probability bound for $\big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \varphi(x_i)\big\|_{\mathcal{H}}$, which we can obtain, with probability greater than $1 - \delta/2$, from McDiarmid's inequality, as

$$\big\|\tfrac{1}{n}\textstyle\sum_{i=1}^n \varepsilon_i \varphi(x_i)\big\|_{\mathcal{H}} \leqslant \tfrac{R\sigma}{\sqrt{n}}\Big(1 + \sqrt{2 \log \tfrac{2}{\delta}}\Big).$$

■

Before analyzing the last proposition and balancing bias and variance, we show how this can be applied beyond well-specified models.

### 7.6.5 Analysis beyond well-specified problems (♦)

In the bound in Eq. (7.21), the only term that requires potentially that $f^* \in \mathcal{H}$ is the bias term $\lambda\langle f^*, (\Sigma+\lambda I)^{-1}\Sigma f^*\rangle$. The following simple lemma is the key to extending to all functions $f^*$ in the closure of $\mathcal{H}$.

**Lemma 7.2** *Given the covariance operator $\Sigma$ and any function $f^* \in \mathcal{H}$, then*

$$\lambda\langle f^*, (\Sigma+\lambda I)^{-1}\Sigma f^*\rangle = \inf_{f\in\mathcal{H}} \Big\{ \|f-f^*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}} \Big\}.$$

**Proof** The optimization problem above can be written as $\inf_{f\in\mathcal{H}} \big\{ \|\Sigma^{1/2}(f-f^*)\|^2_{\mathcal{H}} + \lambda\|f\|^2_{\mathcal{H}} \big\}$, using Eq. (7.12), with solution $f = (\Sigma+\lambda I)^{-1}\Sigma f^*$ and we can simply put back the value in the objective function to get the desired result. ■

**Target function in the closure of $\mathcal{H}$.** By using a limiting argument, we can extend the formula of the bias term in Prop. 7.6 to the general case of $f^* \in L_2(p)$ with Eq. (7.23), in the closure of $\mathcal{H}$ in $L_2(p)$ (because all functions in the closure can be approached by a function in $\mathcal{H}$), leading to

$$\big(1+\frac{R^2}{\lambda n}\big)^2 \inf_{f\in\mathcal{H}} \Big\{ \|f-f^*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}} \Big\}, \tag{7.23}$$

For translation-invariant kernels in $\mathbb{R}^d$ (which are dense in $L_2(\mathbb{R}^d)$), this allows estimating any target function.

**Final result.** Combining the two cases above, we can now show the upper bound for kernel ridge regression in the potentially misspecified case.

**Proposition 7.8 (Mis-specified model kernel ridge regression)** *Assume i.i.d. data $(x_i, y_i) \in \mathcal{X}\times\mathbb{R}$, for $i = 1,\dots,n$, and $y_i = f^*(x_i)+\varepsilon_i$, with $\mathbb{E}[\varepsilon_i|x_i] = 0$ and $\mathbb{E}[\varepsilon_i^2|x_i] \leqslant \sigma^2$. Assume $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ and $f^*$ in the closure of $\mathcal{H}$ in $L_2(p)$. We have:*

$$\mathbb{E}\big[\|\hat{f}_\lambda - f^*\|^2_{L_2(p)}\big] \leqslant \frac{\sigma^2}{n}\Big(1+\frac{R^2}{\lambda n}\Big)\,\mathrm{tr}\,\big((\Sigma+\lambda I)^{-1}\Sigma\big) + \Big(1+\frac{R^2}{\lambda n}\Big)^2 \inf_{f\in\mathcal{H}} \Big\{ \|f-f^*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}} \Big\}. \tag{7.24}$$

Be careful with homogeneity of formulas; e.g., $\frac{R^2}{\lambda n}$ is indeed a constant.

### 7.6.6 Balancing bias and variance (♦)

We can now balance the bias and variance term in the following upper-bound on the expected excess risk,

$$\frac{\sigma^2}{n}\Big(1+\frac{R^2}{\lambda n}\Big)\,\mathrm{tr}\,\big((\Sigma+\lambda I)^{-1}\Sigma\big) + \Big(1+\frac{R^2}{\lambda n}\Big)^2 \inf_{f\in\mathcal{H}} \Big\{ \|f-f^*\|^2_{L_2(p)} + \lambda\|f\|^2 \Big\}.$$

For this section, we will assume that $\mathcal{X} = \mathbb{R}^d$ and that the target function belongs to a Sobolev kernel of order $t > 0$, while the RKHS is a Sobolev space of order $s > d/2$.

We have seen in Section 7.5.2 that the bias term is of order $\big(1+\frac{R^2}{\lambda n}\big)^2 \lambda^{t/s}$ when $s \geqslant t$. For the variance term, we need to study the so-called "degrees of freedom".

**Degrees of freedom.** This is the quantity $\mathrm{tr}\big[\Sigma(\Sigma+\lambda I)^{-1}\big]$, which is decreasing in $\lambda$, from $+\infty$ for $\lambda = 0$ to $0$ for $\lambda = +\infty$. If we know that the eigenvalues $(\lambda_m)_{m\geqslant 0}$ of the covariance operator satisfy

$$\lambda_m \leqslant C(m+1)^{-\alpha},$$

for $\alpha > 1$, then one has, with the change of variable $u = \lambda C^{-1} t^\alpha$ below,

$$\begin{aligned}\mathrm{tr}\big[\Sigma(\Sigma+\lambda I)^{-1}\big] &= \sum_{m\geqslant 0}\frac{\lambda_m}{\lambda_m+\lambda} \leqslant \sum_{m\geqslant 0}\frac{1}{1+\lambda C^{-1}(m+1)^\alpha} \leqslant \int_0^\infty \frac{1}{1+\lambda C^{-1}t^\alpha} \\ &\leqslant \int_0^\infty \lambda^{-1/\alpha} C^{1/\alpha}\frac{1}{\alpha}u^{1/\alpha-1}\frac{du}{1+u} \leqslant O(\lambda^{-1/\alpha}).\end{aligned}$$

It turns out that if the distribution of inputs has a bounded density with respect to the Lebesgue measure, then for our chosen Sobolev space, we have $\alpha = 2s/d$ (see, e.g., Harchaoui et al., 2008, Appendix D).

**Balancing terms (Sobolev spaces).** We thus need to balance $\lambda^{t/s}$ with $\frac{1}{n}\lambda^{-1/\alpha}$, leading to an optimal $\lambda$ proportional to $n^{-(1/\alpha+t/s)^{-1}}$, and a rate proportional to $n^{-\alpha t/(\alpha t+s)}$. This rate is only achievable through our analysis when $\frac{R^2}{n\lambda}$ remains bounded, that is, essentially $\lambda \geqslant R^2/n$, thus, $\frac{1}{\alpha} + \frac{t}{s} \geqslant 1$.

For $\alpha = 2s/d$, we obtain the rate $\frac{1}{n^{2t/(2t+d)}}$, which is valid as long as $\frac{d}{2} + t \geqslant s \geqslant t$. We can make the following observations:

- Except for the constraint $\frac{d}{2}+t \geqslant s \geqslant t$, the upper-bound on the rate obtained after optimizing over $\lambda$ does not depend on the kernel.
- We obtain some form of adaptivity, that is, the rate improves with the regularity of the target function, from the slow rate $\frac{1}{n^{2/(2+d)}}$ when $t = 1$ (recovering the same rate as for local averaging methods[15] in Chapter 6), and that can only be achieved when $s \leqslant d/2+1$, e.g., with the exponential kernel), to the rate $\frac{1}{n^{2s/(2s+d)}}$ when $t = s$, the rate is then always better than $1/\sqrt{n}$ because of the constraint $s > d/2$.
- In order to allow for regularization parameters $\lambda$ which are less than $1/n$, other assumptions are needed. See, e.g., Pillaud-Vivien et al. (2018) and references therein.

[15] In Chapter 6, we assumed the target function to be Lipschitz-continuous, which can be made an element of the Sobolev space of order $t = 1$, with the construction at the end of Section 7.5.2.

## 7.7 Experiments

We consider one-dimensional problems to highlight the adaptivity of kernel methods to the regularity of the target function, with one smooth target and one non-smooth target, and three kernels: exponential kernel corresponding to the Sobolev space of order 1 (top of Figure 7.3), Matern kernel corresponding to the Sobolev space of order 3 (middle), and Gaussian kernel (bottom). In the right plots, dotted lines are affine fits to the log-log learning curves. The regularization parameter for ridge regression is selected to minimize expected risk, and learning curves are obtained by averaging over 20 replications. See results in Figure 7.3. The data

We observe adaptivity for the three kernels: learning is possible even with irregular functions, and the rates are better for smooth target functions. We also note that for kernels with smaller feature spaces (Matern and Gaussian), the performance on the non-smooth target function is worse than for the large feature space (exponential kernel). As highlighted by Bach (2013), this drop in performance is primarily due to a numerical issue (the eigenvalues of the kernel matrix decay exponentially fast, and finite precision arithmetic prevents the use of regularization parameters that are too small).

## 7.8 Conclusion

In this chapter, we have shown how models that are linear in their parameters can be made infinite-dimensional. Algorithmically, this is made possible using the kernel trick that uses only dot-products between the feature maps. Statistically, this leads to models that can adapt to complex prediction functions using the appropriate kernels.

Since algorithms presented in Section 7.4 rely on convex optimization, we obtain precise generalization guarantees that can take into account, estimation, approximation, and optimization errors. A key benefit of positive-definite kernel methods compared to local-averaging techniques is the adaptivity to the smoothness of the prediction function. What is still missing is adaptivity to problems where the optimal prediction function only depends on a subset of the original variables (when applying to inputs in $\mathbb{R}^d$). This will be achieved by neural networks in Chapter 9 at the expense of non-convex optimization problems.

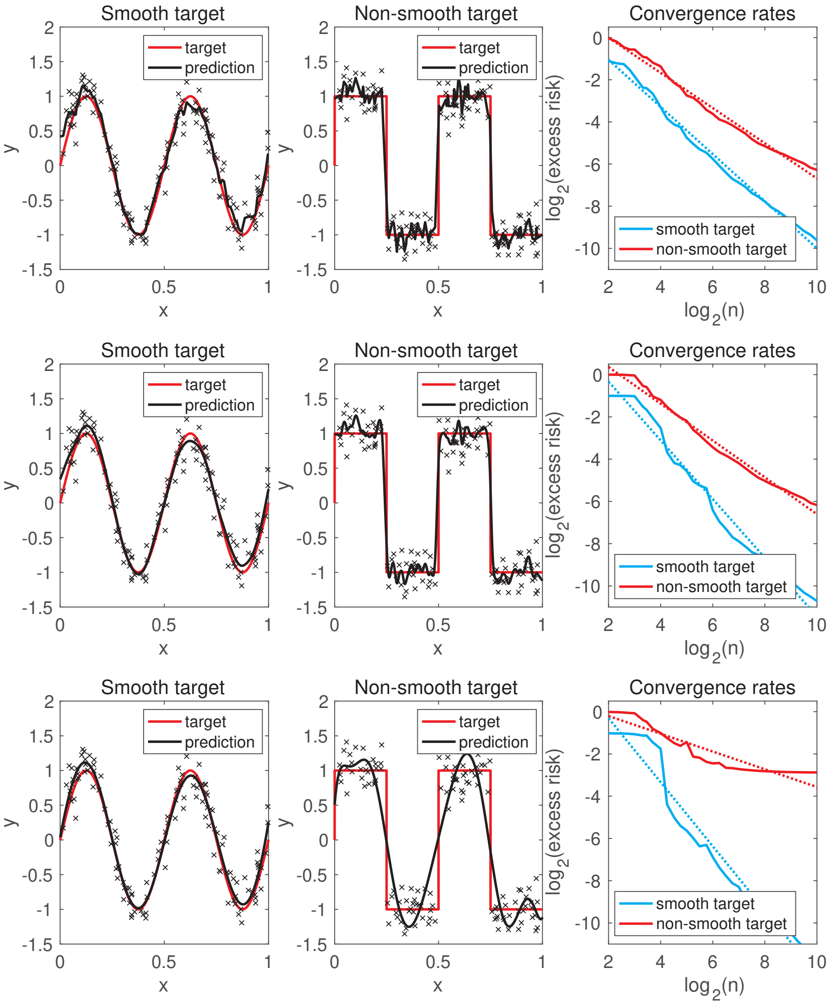


Figure 7.3: Comparison of three kernels, Sobolev space of order 1 (top), Matern kernel corresponding to the Sobolev space of order 3 (middle), and Gaussian kernel (bottom). We consider two different target functions and plot on the right plots the excess risks.

# Chapter 8

# Sparse methods

**Chapter summary**

– Model selection can be performed by adding a specific "sparsity-inducing" penalty on top of the empirical risk.

– $\ell_0$-penalty: For fixed design linear regression, if the optimal predictor has $k$ non-zeros, then we can replace the rate $\frac{\sigma^2 d}{n}$ by $\frac{\sigma^2 k \log d}{n}$ with an $\ell_0$-penalty on the square loss (which is computationally hard).

– $\ell_1$-penalty: With few assumptions, we can get a slow rate proportional to $k\sqrt{\frac{\log d}{n}}$ with an $\ell_1$-penalty and efficient algorithms, while fast rates require strong assumptions on the design matrix in the fixed design setting. In the random design setting, fast rates can be obtained with invertible population covariance matrices.

## 8.1 Introduction

In previous chapters, we have seen the strong effect of the dimensionality of the input space $\mathcal{X}$ on the generalization performance of supervised learning methods in two settings:

- When the target function $f^*$ was only assumed to be Lipschitz-continuous on the set $\mathcal{X} = \mathbb{R}^d$, we saw that the excess risk for $k$-nearest-neigbors, Nadaraya-Watson estimation (Chapter 6), or positive kernel methods (Chapter 7), was scaling as $n^{-2/(d+2)}$.
- When the target function is linear in some features $\varphi(x) \in \mathbb{R}^d$, then the excess risk for unregularized least-squares was scaling as $d/n$.

In these two situations, efficient learning is generally impossible when $d$ is too large (of course, much larger in the linear case).

To improve upon these rates, we study two techniques in this book. The first one is regularization, e.g., by the $\ell_2$-norm, that allows obtaining dimension-independent bounds that cannot improve over the bounds above in the worst-case but are typically adaptive to additional regularity (see Chapter 3 and Chapter 7).

In this chapter, we consider another framework, namely *variable selection*, whose aim is to build predictors that depend only on a small number of variables. The key difficulty is that the identity of the selected variables is not known in advance.

In practice, variable selection is used in mainly two ways:

- The original set of features is already large (for example, in text or web data).
- Given some input $x \in \mathcal{X}$, a large-dimensional feature vector $\varphi(x)$ is built where features are added that could potentially help predict the response, but from which we expect only a small number to be relevant.

⚠ If no good predictor with a small number of active variables exists, these methods are not supposed to work better.

**Linear variable selection.** In this chapter, we focus on *linear* methods, where we assume that we have a feature vector $\varphi(x) \in \mathbb{R}^d$, and we aim to minimize

$$\mathbb{E}[\ell(y, \varphi(x)^\top \theta)]$$

with respect to $\theta \in \mathbb{R}^d$, for some loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$. We will consider two variable selection techniques, namely the penalization by $\|\theta\|_0$ the number of non-zeros in $\theta$ (often called abusively the "$\ell_0$-norm"), or the $\ell_1$-norm. See extensions in Section 8.5.

Non-linear variable selection corresponds to selecting a subset of variables from the $d$ available features $\varphi(x)_1, \dots, \varphi(x)_d$, but with a potentially non-linear model on top of them. This is considered in the context of neural networks in Chapter 9.

**Main focus on least-squares.** These two types of penalties can be applied to all losses, but in this chapter, for simplicity, we will primarily consider the square loss and, in most cases, the fixed design setting (see a thorough description of this setting in Section 3.5), and assume that we have $n$ observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, such that there exists $\theta_* \in \mathbb{R}^d$ for which for $i \in \{1, \dots, n\}$,

$$y_i = \varphi(x_i)^\top \theta_* + \varepsilon_i,$$

where $x_i$ is assumed deterministic, and $\varepsilon_i$ has zero mean and variance $\sigma^2$ (we also assume independence from $x_i$, and sometimes stronger regularity, such as bounded almost surely, or Gaussian). The goal is then to find $\theta \in \mathbb{R}^d$, such that

$$\frac{1}{n}\|\Phi(\theta - \theta_*)\|_2^2 = (\theta - \theta_*)^\top \widehat{\Sigma}(\theta - \theta_*)$$

is as small as possible, where $\Phi \in \mathbb{R}^{n \times d}$ is the design matrix and $\widehat{\Sigma} = \frac{1}{n}\Phi^\top \Phi$ the non-centered empirical covariance matrix. We recall from Chapter 3 that for the ordinary least-squares estimator, the expectation of this excess risk is less than $\sigma^2 d/n$. This is the best possible performance if we make no assumption on $\theta_*$. In this chapter, we assume that $\theta_*$ is sparse, that is, only a few of its components are non-zero, or in other words, $\|\theta_*\|_0 = k$ is small compared to $d$.

The results presented in this section extend beyond the square loss, e.g., to the logistic loss, in a straightforward way for slow rates in $1/\sqrt{n}$ (see the end of Section 8.3.3), with significant additional work for fast rates in $O(1/n)$ (see the end of Section 8.3.4).

### 8.1.1 Dedicated proof technique for constrained least-squares

In this chapter, we consider a more refined proof technique[1] that can extend to constrained versions of least-squares (while our technique in Chapter 3 heavily relies on having a closed form for the estimator, which is not possible in constrained or regularized cases except in few instances, such as ridge regression).

We denote by $\hat\theta$ a minimizer of $\frac{1}{n}\|y - \Phi\theta\|_2^2$ with the constraint that $\theta \in \Theta$, for some subset $\Theta$ of $\mathbb{R}^d$. If $\theta_* \in \Theta$, then we have, by optimality of $\hat\theta$:

$$\|y - \Phi\hat\theta\|_2^2 \leqslant \|y - \Phi\theta_*\|_2^2.$$

By expanding with $y = \Phi\theta_* + \varepsilon$, we get $\|\varepsilon - \Phi(\hat\theta - \theta_*)\|_2^2 \leqslant \|\varepsilon\|_2^2$, leading to, by expanding the norms:

$$\|\varepsilon\|_2^2 - 2\varepsilon^\top\Phi(\hat\theta - \theta_*) + \|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant \|\varepsilon\|_2^2,$$

and thus

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 2\varepsilon^\top\Phi(\hat\theta - \theta_*).$$

We can write it as

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 2\|\Phi(\hat\theta - \theta_*)\|_2 \cdot \varepsilon^\top\Big(\frac{\Phi(\hat\theta - \theta_*)}{\|\Phi(\hat\theta - \theta_*)\|_2}\Big).$$

This reformulation is difficult to deal with because $\hat\theta$ also appears on the right side of the equation. Like done for upper-bounding estimation errors in Chapter 4, we can maximize with respect to $\theta \in \Theta$ to get rid of this randomness, which leads to

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 2\|\Phi(\hat\theta - \theta_*)\|_2 \cdot \sup_{\theta\in\Theta} \varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big), \tag{8.1}$$

where $\hat\theta$ has disappeared from the right-hand side. Finally, isolating $\|\Phi(\hat\theta - \theta_*)\|_2^2$, we get:

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 4 \sup_{\theta\in\Theta} \Big[\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\Big]^2. \tag{8.2}$$

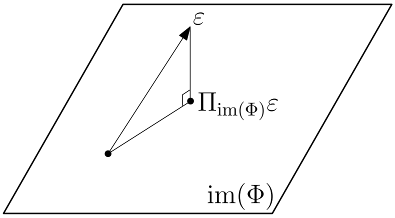
[1] Taken from Philippe Rigollet's lecture notes, see https://math.mit.edu/~rigollet/. See also Rigollet and Tsybakov (2007) for an example of application.

This inequality is true almost surely, and we can take expectation (with respect to $\varepsilon$) to obtain bounds. Therefore, in this chapter, we will compute expectations of maxima of quadratic forms in $\varepsilon$.

For example, when $\Theta = \mathbb{R}^d$ (no constraints), we get, by taking $z = \frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}$, with $\Pi_\Phi = \Pi_{\mathrm{im}(\Phi)}$ the orthogonal projector on the image space $\mathrm{im}(\Phi)$ (which has dimension $\mathrm{rank}(\Phi)$):

$$\mathbb{E}\big[\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \leqslant 4\mathbb{E}\Big[\sup_{z\in\mathrm{im}(\Phi),\|z\|_2=1} \big[\varepsilon^\top z\big]^2\Big].$$

By a simple geometric argument (see below),



we have

$$\sup_{z\in\mathrm{im}(\Phi),\|z\|_2=1} \big[\varepsilon^\top z\big]^2 = \sup_{z\in\mathrm{im}(\Phi),\|z\|_2=1} \big[(\Pi_\Phi\varepsilon)^\top z\big]^2 = \|\Pi_\Phi\varepsilon\|^2,$$

leading to

$$\mathbb{E}\big[\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \leqslant 4\mathbb{E}\Big[\|\Pi_\Phi\varepsilon\|^2\Big] = 4\sigma^2\mathbb{E}\,\mathrm{tr}(\Pi_\Phi^2) = 4\sigma^2\mathrm{rank}(\Phi).$$

We thus get a bound on the excess risk equal to $4\sigma^2 d/n$, which is (because of the constant 4) slightly worse than the direct computation from Chapter 3 (Prop. 3.5) but allows extensions to more complex situations.

This reasoning also allows getting high-probability bounds by adding assumptions on the noise $\varepsilon$. Finally, this also extends to penalized problems (see Section 8.2.2).

### 8.1.2 Probabilistic and combinatorial lemmas

In the proof technique above, we will need to bound expectations of maxima of squared norms of Gaussians, which we now consider. We start with two probabilistic lemmas.

**Lemma 8.1** *If $z \in \mathbb{R}^n$ has a Gaussian distribution with mean 0 and covariance matrix $\sigma^2 I$, then, if $s < \frac{1}{2\sigma^2}$, $\mathbb{E}\big[e^{s\|z\|_2^2}\big] = (1-2\sigma^2 s)^{-n/2}$.*

**Proof** We have, for $\sigma = 1$ (from which we can derive the result for all $\sigma$), and $s < 1/2$ (using independence among the components of $z$):

$$\begin{aligned}\mathbb{E}\big[e^{s\|z\|_2^2}\big] &= \mathbb{E}\big[e^{s\sum_{i=1}^n z_i^2}\big] = \prod_{i=1}^n \mathbb{E}\big[e^{sz_i^2}\big] = \frac{1}{(2\pi)^{n/2}}\prod_{i=1}^n \int_{-\infty}^{\infty} e^{(s-\frac12)z_i^2}dz_i \\ &= \frac{1}{(2\pi)^{n/2}}\prod_{i=1}^n \sqrt{2\pi}\big(1-2s\big)^{-1/2} = (1-2s)^{-n/2}.\end{aligned}$$

■

**Lemma 8.2** *Let $u_1, \dots, u_m$ be $m$ random variables which are potentially dependent, and $s > 0$. Then,* $\mathbb{E}\big[\max\{u_1, \dots, u_m\}\big] \leqslant \frac{1}{s} \log\big(\sum_{i=1}^m \mathbb{E}\big[e^{s u_i}\big]\big)$.

**Proof** Following the reasoning from Section 1.2.4 in Chapter 1, for any $s \in \mathbb{R}$,

$$\mathbb{E}\big[\max\{u_1, \dots, u_m\}\big] \leqslant \frac{1}{s} \log\big(\mathbb{E}\big[e^{s \max\{u_1,\dots,u_m\}}\big]\big) = \frac{1}{s} \log\big(\mathbb{E}\big[\max\{e^{s u_1}, \dots, e^{s u_m}\}\big]\big),$$

which is thus less than $\frac{1}{s} \log\big(\sum_{i=1}^m \mathbb{E}\big[e^{s u_i}\big]\big)$. ■

The previous two lemmas can be combined to upper-bound the expectation of squared norms of Gaussian random variables: if $z_1, \dots, z_m \in \mathbb{R}^n$ are centered (that is, zero-mean) Gaussian random vectors which are potentially dependent, but for which the covariance matrix of $z_i$ has eigenvalues less than $\sigma^2$, we can first use the rotational invariance of Gaussian densities, to assume without loss of generality that the Gaussians have diagonal covariance matrices with components $\sigma_{ij}^2 \leqslant \sigma^2$ (for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$), then, we have from Lemma 8.1,

$$\mathbb{E}[e^{s\|z_i\|_2^2}] = \prod_{j=1}^n \mathbb{E}[e^{s z_{ij}^2}] \leqslant \prod_{j=1}^n (1 - 2\sigma_{ij}^2 s)^{-1/2} \leqslant (1 - 2\sigma^2 s)^{-n/2}.$$

Thus, for $s = \frac{1}{4\sigma^2}$, $\mathbb{E}[e^{s\|z_i\|_2^2}] \leqslant 2^{n/2}$ for all $i \in \{1, \dots, m\}$, and from Lemma 8.2,

$$\mathbb{E}[\max\{\|z_1\|_2^2, \dots, \|z_m\|_2^2\}] \leqslant 4\sigma^2 \log(m 2^{n/2}) = 2n\sigma^2 \log(2) + 4\sigma^2 \log(m),$$

which is to be compared to the expectation of each argument of the max, which is less than $\sigma^2 n$. We pay an additive factor proportional to $\sigma^2 \log(m)$. This will be applied to $m \propto d^k$, leading to the additional term in $\sigma^2 k \log(d)$ for methods based on the $\ell_0$-penalty. The term in $d^k$ comes from the following lemma.

**Lemma 8.3** *Let $d > 0$ and $k \in \{1, \dots, d\}$. Then* $\log \binom{d}{k} \leqslant k(1 + \log \frac{d}{k})$.

**Proof** By recursion on $k$, the inequality is trivial for $k = 1$, and if $\binom{d}{k-1} \leqslant \big(\frac{ed}{k-1}\big)^{k-1}$, then

$$\binom{d}{k} = \binom{d}{k-1} \frac{d-k+1}{k} \leqslant \Big(\frac{ed}{k-1}\Big)^{k-1} \frac{d}{k} \leqslant \Big(\frac{ed}{k}\Big)^{k-1} \Big(1 + \frac{1}{k-1}\Big)^{k-1} \frac{d}{k} \leqslant \Big(\frac{ed}{k}\Big)^{k-1} e \frac{d}{k} = \Big(\frac{ed}{k}\Big)^{k},$$

where we use for $\alpha > 0$, $(1 + \frac{1}{\alpha})^\alpha = \exp(\alpha \log(1 + 1/\alpha)) \leqslant \exp(1) = e$. ■

We now consider two types of variable selection frameworks, one based on $\ell_0$-penalties and one based on $\ell_1$-penalties.

**Exercise 8.1 (Concentration of chi-squared variables)** *We consider $n$ independent standard normal variables $z_1, \dots, z_n$ and the variables $y = z_1^2 + \cdots + z_n^2$. Using Lemma 8.1, show that for any $\varepsilon > 0$, $\mathbb{P}(y \geqslant n(1+\varepsilon)) \leqslant \big(\frac{1+\varepsilon}{\exp(\varepsilon)}\big)^{n/2}$, and for any $\varepsilon \in (0, 1)$, $\mathbb{P}(y \leqslant n(1-\varepsilon)) \leqslant \big(\frac{1-\varepsilon}{\exp(-\varepsilon)}\big)^{n/2}$.*

## 8.2 Variable selection by the $\ell_0$-penalty

In this section, we assume that the target vector $\theta_*$ has $k$ non-zero components, that is, $\|\theta_*\|_0 = k$. We denote by $A = \text{supp}(\theta_*)$ the "support" of $\theta_*$, that is, the subset of $\{1, \dots, d\}$ composed of $j$ such that $(\theta_*)_j \neq 0$. We have $|A| = k$.

**Price of adaptivity.** If we knew the set $A$, then we could simply perform least-squares with the design matrix $\Phi_A \in \mathbb{R}^{n \times |A|}$, where $\Phi_B$ denotes the sub-matrix of $\Phi$ obtained by keeping only the columns from $B$, with an excess risk proportional to $\sigma^2 k/n$ (this is what we call the "oracle" in Section 8.4). Thus, as long as $k$ is small compared to $n$, we can estimate $\theta_*$ correctly, regardless of the potentially large value of $d$.

However, we do not know $A$ in advance, and we have to estimate it. We will see that this will lead to an extra factor of $\log\left(\frac{d}{k}\right) \leqslant \log d$ due to the potentially large number of models with $k$ variables.

### 8.2.1 Assuming $k$ is known

We start by assuming that the cardinality $k$ is known in advance, and we consider Gaussian noise for simplicity (this extends to sub-Gaussian noise as well; see note below).

**Proposition 8.1 (Model selection - known $k$)** *Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance $\sigma^2$, with $\|\theta_*\|_0 \leqslant k$, for $k \leqslant d/2$. Let $\hat{\theta}$ be the minimizer of $\|y - \Phi\theta\|_2^2$ with the constraint that $\|\theta\|_0 \leqslant k$. Then, the (fixed design) excess risk is upper-bounded as:*

$$\mathbb{E}\big[(\hat{\theta} - \theta_*)^\top \widehat{\Sigma}(\hat{\theta} - \theta_*)\big] = \mathbb{E}\Big[\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2\Big] \leqslant 32\sigma^2 \frac{k}{n}\Big(\log\Big(\frac{d}{k}\Big) + 1\Big).$$

**Proof** For any $\theta$ such that $\|\theta\|_0 \leqslant k$, we have $\|\theta - \theta_*\|_0 \leqslant 2k$. Thus, we have, using the bounding technique from Section 8.1.1:

$$\begin{aligned}
\|\Phi(\hat{\theta} - \theta_*)\|_2^2 &\leqslant 4 \sup_{\theta \in \mathbb{R}^d, \|\theta\|_0 \leqslant k} \Big[\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\Big]^2 \text{ from Eq. (8.2)}, \\
&\leqslant 4 \sup_{\theta \in \mathbb{R}^d, \|\theta - \theta_*\|_0 \leqslant 2k} \Big[\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\Big]^2 \text{ from the discussion above}, \\
&= 4 \sup_{B \subset \{1,\dots,d\},\ |B| \leqslant 2k} \ \sup_{\text{supp}(\theta - \theta_*) \subset B} \Big[\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\Big]^2
\end{aligned}$$

by separating by supports. Thus, using the same argument as in Section 8.1.1,

$$\begin{aligned}
\|\Phi(\hat{\theta} - \theta_*)\|_2^2 &\leqslant 4 \sup_{B \subset \{1,\dots,d\},\ |B| \leqslant 2k} \ \sup_{z \in \text{im}(\Phi_B), \|z\|_2 = 1} \big[\varepsilon^\top z\big]^2 \\
&\leqslant 4 \sup_{B \subset \{1,\dots,d\},\ |B| \leqslant 2k} \|\Pi_{\Phi_B}\varepsilon\|^2 \leqslant 4 \sup_{B \subset \{1,\dots,d\},\ |B| = 2k} \|\Pi_{\Phi_B}\varepsilon\|^2,
\end{aligned}$$

because $\|\Pi_{\Phi_B}\varepsilon\|^2$ is non-decreasing in $B$.

The random variable $\|\Pi_{\Phi_B}\varepsilon\|^2$ has an expectation which is less than $2k$. Given that there are $\binom{d}{2k} \leqslant \left(\frac{ed}{2k}\right)^{2k}$ sets $B$ of cardinality $2k$ (bound from Lemma 8.3), we should expect, with concentration inequalities from Section 8.1.2, that we pay a price of $\log\left[\left(\frac{ed}{2k}\right)^{2k}\right] \approx k \log\frac{d}{k}$. We will make this reasoning formal.

Indeed, $\Pi_{\Phi_B}\varepsilon$ is normally distributed with isotropic covariance matrix of dimension $|B| \leqslant 2k$, and thus we have for $s\sigma^2 < 1/2$, from Lemma 8.1:

$$\mathbb{E}\left[e^{s\|\Pi_{\Phi_B}\varepsilon\|^2}\right] \leqslant (1-2\sigma^2 s)^{-k}.$$

Therefore, with $s = 1/(4\sigma^2)$, for which $(1-2\sigma^2 s)^{-k} = 2^k$, we get, from Lemma 8.2:

$$\begin{aligned}\mathbb{E}\left[\|\Phi(\hat\theta-\theta_*)\|_2^2\right] &\leqslant 16\sigma^2 \log\left(\binom{d}{2k}2^k\right)\\ &\leqslant 16\sigma^2 \log\left(\left(\frac{ed}{2k}\right)^{2k}2^k\right) = 16\sigma^2\Big(2k\log\left(\frac{d}{k}\right) + (2-\log 2)k\Big).\end{aligned}$$

This leads to the desired result. ■

We can make the following observations:

- The term $k\log(d/k)$ comes from the logarithm of the number $m$ of subsets of $\{1,\dots,d\}$ of size $2k$, which is a result of the expectation of the maximum of $m$ squared norms of Gaussians.
- The assumption that $k < d/2$ is not a real issue, as when $k \geqslant d/2$, then the classical bound $\sigma^2 d/n$ is of the same order as $\sigma^2 k\log(d/k)/n$.
- The result extends beyond Gaussian noise, that is, for all sub-Gaussian $\varepsilon_i$, for which $\mathbb{E}[e^{s\varepsilon_i}] \leqslant e^{s^2\tau^2}$ for all $s > 0$ (for some $\tau > 0$), or, equivalently $\mathbb{P}(|\varepsilon_i| > t) = O(e^{-ct^2})$ for some $c > 0$.
- The result extends if the minimization of the empirical risk is only done approximately.
- This result is not improvable by any algorithm (polynomial time or not), see, e.g., Giraud (2014, Theorem 2.3) and Chapter 15.

**Algorithms.** In terms of algorithms, essentially all subsets of size $k$ have to be looked at for exact minimization, with a cost proportional to $O(d^k)$, which is a problem when $k$ gets large. There are, however, two simple algorithms that only come with guarantees when such fast rates are available for $\ell_1$-regularization (see Section 8.3.4, and Zhang, 2009).

- **Greedy algorithm:** Starting from the empty set, variables are added one by one, maximizing the resulting cost reduction. This is often referred to as orthogonal matching pursuit (Pati et al., 1993).
- **Iterative sorting:** Starting from $\theta_0 = 0$, the iterative algorithm goes as follows at iteration $t$; the upper bound (based on the $L$-smoothness of the quadratic loss,

with $L = \lambda_{\max}(\frac{1}{n}\Phi^\top \Phi)$, see Chapter 5):

$$\frac{1}{n}\|y - \Phi\theta_{t-1}\|_2^2 - \frac{2}{n}(y - \Phi\theta_{t-1})^\top \Phi(\theta - \theta_{t-1}) + L\|\theta - \theta_{t-1}\|_2^2$$

on the cost function $\frac{1}{n}\|y - \Phi\theta\|_2^2$ is built and minimized with respect to $\|\theta\|_0 \leqslant k$ to obtain $\theta_t$. This is done (proof left as an exercise) by computing the unconstrained minimizer $\theta_{t-1} + \frac{1}{L}\frac{1}{n}\Phi^\top(y - \Phi\theta_{t-1})$, and selecting the $k$ largest components.

### 8.2.2 Estimating $k$ (♦)

In practice, regardless of the computational cost, one also needs to estimate $k$. A classical idea to consider penalized least-squares and minimize

$$\frac{1}{n}\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_0. \tag{8.3}$$

This is a hard problem to solve, which essentially requires looking at all $2^d$ subsets. For a well-chosen $\lambda$, this (almost) leads to the same performance as if $k$ were known.

**Proposition 8.2 (Model selection - $\ell_0$-penalty)** *Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance $\sigma^2$, with $\|\theta_*\|_0 \leqslant k$. Let $\hat\theta$ be a minimizer of Eq. (8.3). Then, for $\lambda = \frac{8\sigma^2}{n}\log(2d)$, we have:*

$$\mathbb{E}\Big[\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2\Big] \leqslant \frac{16k\sigma^2}{n}\big[2 + \log(d)\big] + \frac{16\sigma^2}{n}.$$

**Proof** We follow the same proof technique than in Section 8.1.1, but now for regularized problems. We have by optimality of $\hat\theta$:

$$\|y - \Phi\hat\theta\|_2^2 + n\lambda\|\hat\theta\|_0 \leqslant \|y - \Phi\theta_*\|_2^2 + n\lambda\|\theta_*\|_0,$$

which leads to, using the inequality $2ab \leqslant 2a^2 + \frac{1}{2}b^2$, and the same arguments that led to Eq. (8.1):

$$\begin{aligned}
\|\Phi(\hat\theta - \theta_*)\|_2^2 &\leqslant 2\|\Phi(\hat\theta - \theta_*)\|_2 \cdot \varepsilon^\top\Big(\frac{\Phi(\hat\theta - \theta_*)}{\|\Phi(\hat\theta - \theta_*)\|_2}\Big) + n\lambda\|\theta_*\|_0 - n\lambda\|\hat\theta\|_0 \\
&\leqslant 2\Big(\varepsilon^\top\Big(\frac{\Phi(\hat\theta - \theta_*)}{\|\Phi(\hat\theta - \theta_*)\|_2}\Big)\Big)^2 + \frac{1}{2}\|\Phi(\hat\theta - \theta_*)\|_2^2 + n\lambda\|\theta_*\|_0 - n\lambda\|\hat\theta\|_0,
\end{aligned}$$

leading to, by taking the supremum over $\theta \in \mathbb{R}^d$:

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant \sup_{\theta\in\mathbb{R}^d}\Big\{4\Big(\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\Big)^2 + 2n\lambda\|\theta_*\|_0 - 2n\lambda\|\theta\|_0\Big\}.$$

We then take the supremum by layers, as $\sup_{\theta\in\mathbb{R}^d} = \sup_{k'\in\{1,\dots,d\}} \sup_{|B|=k'} \sup_{\mathrm{supp}(\theta)\subset B}$, that is, using the same derivations as for Prop. 8.1 ($A$ is the support of $\theta_*$):

$$\begin{aligned}
&\mathbb{E}\big[\|\Phi(\hat\theta-\theta_*)\|_2^2\big]\\
\leqslant\ & \mathbb{E}\bigg[\sup_{k'\in\{1,\dots,d\}} \sup_{|B|=k'} \sup_{\mathrm{supp}(\theta)\subset B} \bigg\{4\Big(\varepsilon^\top\Big(\frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}\Big)\Big)^2 + 2n\lambda\|\theta_*\|_0 - 2n\lambda k'\bigg\}\bigg]\\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + 4\mathbb{E}\bigg[\sup_{k'\in\{1,\dots,d\}} \sup_{|B|=k'} \Big\{\|\Pi_{\Phi_{A\cup B}}\varepsilon\|^2 - \frac{n\lambda}{2}k'\Big\}\bigg].
\end{aligned}$$

We thus get with the same reasoning as in Section 8.2.1 (based on the probabilistic lemmas from Section 8.1.2), using $s = \frac{1}{4\sigma^2}$ within Lemma 8.2:

$$\begin{aligned}
&\mathbb{E}\big[\|\Phi(\hat\theta-\theta_*)\|_2^2\big]\\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + 16\sigma^2\log\Big(\sum_{k'=1}^d \binom{d}{k'} 2^{k'+\|\theta^*\|_0}\exp\big(-\frac{n\lambda k'}{8\sigma^2}\big)\Big)\\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + 16\sigma^2\|\theta^*\|_0\log(2) + 16\sigma^2\log\Big(\sum_{k'=1}^d \binom{d}{k'}\exp\big(k'\big(\log(2)-\frac{n\lambda}{8\sigma^2}\big)\big)\Big)\\
\leqslant\ & (2n\lambda + 16\log(2)\sigma^2)\|\theta_*\|_0 + 16\sigma^2 d\log\Big(1+\exp\big(\log(2)-\frac{n\lambda}{8\sigma^2}\big)\Big).
\end{aligned}$$

To find a good regularization parameter, we can then approximately minimize the bound above with respect to $\lambda$. We obtain a good balance of the two terms by having $-\log d = \log(2) - \frac{n\lambda}{8\sigma^2}$, that is, $\lambda = \frac{8\sigma^2}{n}\log(2d)$, for which we get:

$$\mathbb{E}\big[\|\Phi(\hat\theta-\theta_*)\|_2^2\big] \leqslant (2n\lambda + 16\log(2)\sigma^2)\|\theta_*\|_0 + 16\sigma^2 \leqslant 16\sigma^2\big((\log(d)+2)\|\theta_*\|_0 + 1\big),$$

and get the desired result. ■

We can make the following observations:

- Penalties on the number of parameters on top of the empirical risk can be obtained from various perspectives, for square loss depending on whether the noise variance is known or more generally for other losses. For example, the Bayesian information criterion (BIC) penalizes by penalty proportional to $\|\theta\|_0\log n$ (often a smaller penalty than proposed here).
- Note that we need to know $\sigma^2$ in advance to compute $\lambda$, which can be a problem in practice. See Giraud et al. (2012) for more details and alternative formulations.
- The three most important aspects are that: (1) the bound does not require any assumption on the design matrix $\Phi$, (2) we observe a positive high-dimensional phenomenon, where $d$ only appears as $\frac{\log d}{n}$, but (3) only exponential-time algorithms are possible for solving the problem with guarantees (see algorithms below).

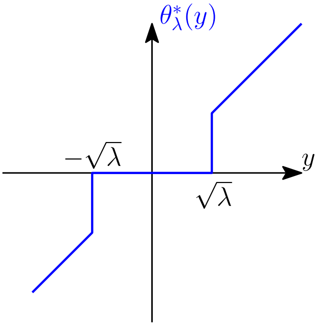
**Exercise 8.2** (♦) *With a penalty proportional to $\|\theta\|_0\log\frac{d}{\|\theta\|_0}$, show the same bound than for $k$ known.*

**Algorithms.** We can extend the two algorithms from Section 8.2.1 for the penalized case:

- **Forward-backward algorithm to minimize a function of a set $B$:** Starting from the empty set $B = \varnothing$, at every step of the algorithm, one tries both a forward algorithm (adding a node to $B$) and a backward algorithm (removing a node from $B$), and only perform a step if it decreases the overall cost function. See an analysis by Zhang (2011).
- **Iterative hard-thresholding:** compared to the constrained case, we minimize

$$\frac{1}{n}\|y-\Phi\theta_{t-1}\|_2^2 - \frac{2}{n}(y-\Phi\theta_{t-1})^\top\Phi(\theta-\theta_{t-1}) + L\|\theta-\theta_{t-1}\|_2^2 + \lambda\|\theta\|_0,$$

with $L = \lambda_{\max}(\frac{1}{n}\Phi^\top\Phi)$, which can also be computed in closed form (by iterative hard thresholding). That is, with $\theta_t = \theta_{t-1} + \frac{1}{nL}\Phi^\top(y - \Phi\theta_{t-1})$, all components $(\theta_t)_j$ such that $|(\theta_t)_j|^2 \geqslant \frac{\lambda}{L}$, are left unchanged and all others are set to zero. Indeed, for one-dimensional problems, the minimizer of $|\theta - y|^2 + \lambda 1_{\theta\neq 0}$ is $\theta^*_\lambda(y) = 0$ if $|y|^2 \leqslant \lambda$ and $\theta^*_\lambda(y) = y$ otherwise (see below).



This is referred to as "iterative hard thresholding" (while for the $\ell_1$-norm, this will be iterative *soft* thresholding) because a component is either kept intact or set exactly to zero, leading to a discontinuous behavior. See an analysis by Blumensath and Davies (2009).

## 8.3 Variable selection by $\ell_1$-regularization

We now consider a computationally efficient alternative to $\ell_0$-penalties, namely using $\ell_1$-penalties, by minimizing, for the square loss:

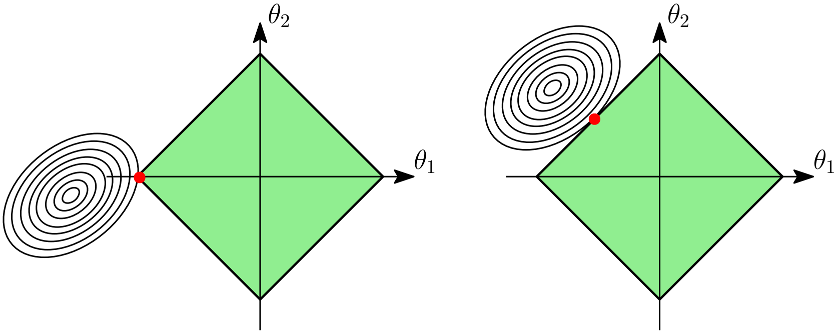
$$\frac{1}{2n}\|y-\Phi\theta\|_2^2 + \lambda\|\theta\|_1. \tag{8.4}$$

This is a convex optimization problem on which algorithms from Chapter 5 can be applied (see instances below). It is often called the "Lasso" problem, for "least absolute shrinkage and selection operator" (Tibshirani, 1996).

We first present algorithms dedicated to solving the optimization problem, then present "slow rate" analyses leading to excess risks in $O(1/\sqrt{n})$, first in the random design case with Lipschitz-continuous losses, and then in the fixed design case with the square loss. We then present "fast rate" analyses leading to excess risks in $O(1/n)$.

### 8.3.1 Intuition and algorithms

**Sparsity-inducing effect.** Unlike the squared $\ell_2$-norm used in ridge regression, the $\ell_1$-norm is non-differentiable, and its non-differentiability is not limited to $\theta = 0$ but occurs in many other points. To see this, we can look at the $\ell_1$-ball and its different geometry compared to the $\ell_2$-ball. This is directly relevant to situations where we constrain the value of the norm instead of penalizing it.



As shown above, where we represent the level set of a potential loss function, the solution of minimizing the loss subject to the $\ell_1$-constraint (in green) is obtained when level sets are "tangent" to the constraint set. In the right part, this is obtained at a point away from the axes, but on the left part, this is achieved at one of the corners of the $\ell_1$-ball, which are points where one of the components of $\theta$ is equal to zero. Such corners are "attractive", that is, minimizers tend to be precisely at these corners, and this exactly leads to sparse solutions.

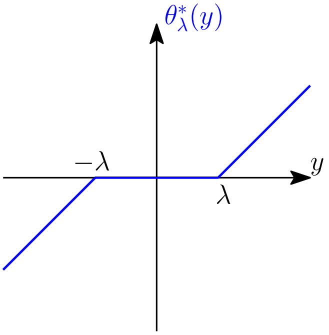
The $\ell_1$-norm is also often introduced as the "convex relaxation" of the $\ell_0$-penalty. Indeed, the $\ell_1$-norm is the convex envelope (the largest convex function which is a lower-bound) of the $\ell_0$-penalty on the set $[-1, 1]^d$ (proof left as an exercise). While this provides some intuition about the $\ell_1$-norm and its potential generalization to other sparse situations, this does not directly justify its good behavior on sparse problems.

**One-dimensional problem.** Another classical way to understand the sparsity-inducing effect is to consider the one-dimensional problem:

$$\min_{\theta \in \mathbb{R}} \ F(\theta) = \frac{1}{2}(y - \theta)^2 + \lambda|\theta|.$$

Since $F$ is strongly-convex, it has a unique minimizer $\theta^*_\lambda(y)$. For $\lambda = 0$ (no regularization), we have $\theta^*_0(y) = y$, while for $\lambda > 0$, by computing left and right derivatives at zero (to be done as an exercise), one can check that $\theta^*_\lambda(y) = 0$ if $|y| \leqslant \lambda$, and $\theta^*_\lambda(y) = y - \lambda$

for $y > \lambda$, and $\theta^*_\lambda(y) = y + \lambda$ for $y < -\lambda$, which can be put all together as $\theta^*_\lambda(y) = \max\{|y| - \lambda, 0\} \operatorname{sign}(y)$, which is depicted below. This is referred to as iterative soft thresholding (this will be useful for proximal methods below).



Note that the minimizer is either sent to zero or shrunk toward zero.

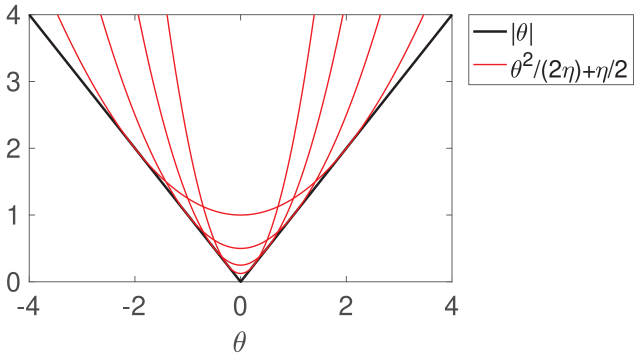
**Optimization algorithms.** We can adapt algorithms from Chapter 5 to the problem in Eq. (8.4).

- **Iterative soft-thresholding:** We can apply proximal methods to the objective function of the form $F(\theta) + \lambda\|\theta\|_1$ for $F(\theta) = \frac{1}{2n}\|y - \Phi\theta\|_2^2$, for which the gradient is $F'(\theta) = -\frac{1}{n}\Phi^\top(y - \Phi\theta)$. The plain (non-accelerated) proximal method recursion is
$$\theta_t = \arg\min_{\theta \in \mathbb{R}^d} F(\theta_{t-1}) + F'(\theta_{t-1})^\top(\theta - \theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2 + \lambda\|\theta\|_1,$$
with $L = \lambda_{\max}(\frac{1}{n}\Phi^\top\Phi)$. This leads to $(\theta_t)_j = \max\{|(\eta_t)_j| - \lambda, 0\} \operatorname{sign}((\eta_t)_j)$, for $\eta_t = \theta_{t-1} - \frac{1}{L}F'(\theta_{t-1})$. This simple algorithm can also be accelerated. The convergence rate then depends on the invertibility of $\frac{1}{n}\Phi^\top\Phi$ (if invertible, we get an exponential convergence rate in $t$, with only $O(1/t)$ otherwise).
- **Coordinate descent:** Although the $\ell_1$-norm is a non-differentiable function, coordinate descent can be applied (because the $\ell_1$-norm is "separable"). At each iteration, we select a coordinate to update (at random or by cycling) and optimize with respect to this coordinate, which is a one-dimensional problem that can be solved in closed form. The convergence properties are similar to proximal methods (Fercoq and Richtárik, 2015).

  **Exercise 8.3** *Provide a closed-form expression for the iteration of the coordinate descent algorithm described above.*

**$\eta$-trick.** The non-differentiability of the $\ell_1$-norm may also be treated through the simple identity:
$$|\theta_j| = \inf_{\eta_j > 0} \frac{\theta_j^2}{2\eta_j} + \frac{\eta_j}{2},$$

where the minimizer is attained at $\eta_j = |\theta_j|$. See below an example in one dimension, with $|\theta|$ and several quadratic upper bounds.



This leads to the reformulation of Eq. (8.4) as

$$\inf_{\theta\in\mathbb{R}^d} \frac{1}{2n}\|y-\Phi\theta\|_2^2 + \lambda\|\theta\|_1 = \inf_{\eta\in\mathbb{R}_+^d}\inf_{\theta\in\mathbb{R}^d} \frac{1}{2n}\|y-\Phi\theta\|_2^2 + \frac{\lambda}{2}\sum_{j=1}^d \frac{\theta_j^2}{\eta_j} + \frac{\lambda}{2}\sum_{j=1}^d \eta_j,$$

and alternating optimization algorithms can be used: (a) minimizing with respect to $\eta$ when $\theta$ is fixed can be done in closed form as $\eta_j = |\theta_j|$, while minimizing with respect to $\theta$ when $\eta$ is fixed is a quadratic optimization problem which can be solved by a linear system.[2]

**Optimality conditions (♦).** To study the estimator defined by Eq. (8.4), it is often necessary to characterize when a certain $\theta$ is optimal or not, that is, to derive optimality conditions.

Since the objective function $H(\theta) = F(\theta) + \lambda\|\theta\|_1$ is not differentiable, we need other tools than having the gradient equal to zero. The gradient looks only at $d$ directions (along the coordinate axis), while, in the non-smooth context, we need to look at all directions, that is, for all $\Delta \in \mathbb{R}^d$, we require that the directional derivative,

$$\partial H(\theta,\Delta) = \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\big[H(\theta+\varepsilon\Delta) - H(\theta)\big],$$

is non-negative. That is, we need to go up in all directions. When $H$ is differentiable at $\theta$, then $\partial H(\theta,\Delta) = H'(\theta)^\top\Delta$, and the positivity for all $\Delta$ is equivalent to $H'(\theta) = 0$.

For $H(\theta) = F(\theta) + \lambda\|\theta\|_1$, we have:

$$\partial H(\theta,\Delta) \quad = \quad F'(\theta)^\top\Delta + \lambda\sum_{j,\ \theta_j\neq 0}\operatorname{sign}(\theta_j)\Delta_j + \lambda\sum_{j,\ \theta_j=0}|\Delta_j|.$$

[2] See more details in `https://www.di.ens.fr/~fbach/ltfp/etatrick.html` and by Bach et al. (2012a, Section 5).

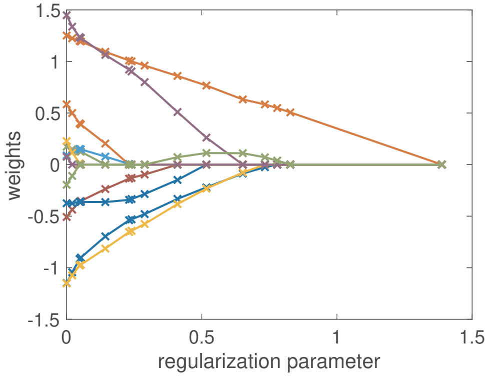


Figure 8.1: Regularization path for a Lasso problem in dimension $d = 32$ and $n = 32$ input observations sampled from a standard Gaussian distribution with 4 non-zero weights equal to $-1$ or $+1$, and outputs generated with additive Gaussian noise with unit variance. The random seed was chosen so that at least one weight comes in and out in the regularization path.

It is separable in $\Delta_j$, $j = 1, \dots, d$, and it is non-negative for all $j$, if and only if all components that depend on $\Delta_j$ are non-negative.

When $\theta_j \neq 0$, then this requires $F'(\theta)_j + \lambda \operatorname{sign}(\theta_j) = 0$, while when $\theta_j = 0$, then we need $F'(\theta)_j \Delta_j + \lambda |\Delta_j| \geqslant 0$ for all $\Delta_j$, which is equivalent to $|F'(\theta)_j| \leqslant \lambda$. This leads to the set of conditions:

$$\begin{cases} F'(\theta)_j + \lambda \operatorname{sign}(\theta_j) = 0, & \forall j \in \{1, \dots, d\} \text{ such that } \theta_j \neq 0, \\ |F'(\theta)_j| \leqslant \lambda, & \forall j \in \{1, \dots, d\} \text{ such that } \theta_j = 0. \end{cases}$$

See Giraud (2014) for more details. Note that we could have also used subgradients to derive these optimality conditions (derivations left as an exercise).

**Homotopy method (♦♦).** We assume for simplicity that $\Phi^\top \Phi$ is invertible so that the minimizer $\theta(\lambda)$ is unique. Given a certain sign pattern for $\theta$, optimality conditions are all convex in $\lambda$ and thus define an interval in $\lambda$ where the sign is constant. Given the sign, then the solution $\theta(\lambda)$ is affine in $\lambda$, leading to a piecewise affine function in $\lambda$ (see an example of a regularization path in Figure 8.1).

If we know the breakpoints in $\lambda$ and the associated signs, we can compute all solutions for all $\lambda$. This is the source of the homotopy algorithm for Eq. (8.4), which starts with large $\lambda$ and builds the path of solutions by computing break points one by one. See more details by Osborne et al. (2000); Mairal and Yu (2012).

### 8.3.2 Slow rates - random design

In this section, we consider Lipschitz-continuous loss functions and, thus, an empirical risk of the form

$$\widehat{\mathcal{R}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \varphi(x_i)^\top \theta),$$

with $\ell$ having Lipschitz constant $G$ with respect to the second variable. We assume that the expected risk $\mathcal{R}(\theta) = \mathbb{E}[\ell(y, \varphi(x)^\top \theta)]$ is minimized at a certain $\theta_* \in \mathbb{R}^d$ and, for simplicity, we consider the estimator $\hat{\theta}_D$ obtained by minimizing $\widehat{\mathcal{R}}(\theta)$ with the constraint that $\|\theta\|_1 \leqslant D$, where we will use tools from Section 4.5.4 (we could also consider penalized formulation using Section 4.5.5). We assume that $\|\varphi(x)\|_\infty \leqslant R$ almost surely.

From Section 4.5.4, we get that,

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_D)\big] \leqslant \inf_{\|\theta\|_1 \leqslant D} \mathcal{R}(\theta) + 4G\mathrm{R}_n(\mathcal{F}_D),$$

where $\mathrm{R}_n(\mathcal{F}_D)$ is the Rademacher complexity of the set of linear predictors with weight vectors bounded by $D$ in $\ell_1$-norm, which we can compute as:

$$\mathrm{R}_n(\mathcal{F}_D) = \mathbb{E}\Big[\sup_{\|\theta\|_1 \leqslant D} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^{\mathrm{R}} \varphi(x_i)^\top \theta\Big] = D\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^{\mathrm{R}} \varphi(x_i)\Big\|_\infty\Big],$$

where $\varepsilon_i^{\mathrm{R}} \in \{-1, 1\}$ are Rademacher random variables. We can now compute a bound on the expectation, first conditioned on the data. Indeed, $\varepsilon_i^{\mathrm{R}} \varphi(x_i)$ has conditional zero mean and is bounded in absolute value by $R$. It is thus sub-Gaussian with constant $R$ (see Section 1.2.1, which implies that $\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^{\mathrm{R}} \varphi(x_i)$ is sub-Gaussian with constant $R^2/n^2$). We can then use Prop. 1.4, to get that the maximum of the $2d$ sub-Gaussian variables is less than $\big(2R^2 \log(2d)/n^2\big)^{1/2}$. This leads to:

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_D)\big] \leqslant \inf_{\|\theta\|_1 \leqslant D} \mathcal{R}(\theta) + \frac{GRD\sqrt{\log(2d)}}{\sqrt{n}}.$$

When $D$ is large enough, e.g., $D = \|\theta_*\|_1$, then we get an excess risk bounded by $GRD\sqrt{\log(2d)}/\sqrt{n}$. If $\theta_*$ has only $k$ non-zero, its $\ell_1$-norm will typically grow as $O(k)$, and we see a high-dimensional phenomenon with a bound proportional to $k\sqrt{\log d}/\sqrt{n}$, where $d$ can be much larger than $n$, as long as $k^2 \log(d)/n$ is small. This is a "slow" rate because of the dependence in $n$, which is $O(1/\sqrt{n})$ rather than in $O(1/n)$.

### 8.3.3 Slow rates - fixed design (square loss)

We now look at the fixed design setting with the square loss. We first consider an analysis based on simple tools with no assumptions on the design matrix $\Phi$. We will see that we can deal with high-dimensional inference problems where $d$ can be large, but it will be with rates in $1/\sqrt{n}$ and not $1/n$, hence the denomination "slow".

We study the penalization by a general norm $\Omega : \mathbb{R}^d \to \mathbb{R}$ with dual norm $\Omega^*$ defined as $\Omega^*(z) = \sup_{\Omega(\theta) \leqslant 1} z^\top \theta$ (see Exercise 8.4 below for classical examples). We thus denote by $\hat{\theta}$ a minimizer of

$$\frac{1}{2n} \|y - \Phi\theta\|_2^2 + \lambda \Omega(\theta). \tag{8.5}$$

We start with a lemma characterizing the excess risk in two situations: (a) where $\lambda$ is large enough and (b) in the general case.

**Lemma 8.4** *Let $\hat{\theta}$ be a minimizer of Eq. (8.5).*

*(a) If $\Omega^*(\Phi^\top \varepsilon) \leqslant \frac{n\lambda}{2}$, then we have $\Omega(\hat{\theta}) \leqslant 3\Omega(\theta_*)$ and $\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leqslant 3\lambda\Omega(\theta_*)$.*

*(b) In all cases, $\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leqslant \frac{4}{n}\|\varepsilon\|_2^2 + 4\lambda\Omega(\theta_*)$.*

**Proof** We have, like in Section 8.1.1, by optimality of $\hat{\theta}$ for Eq. (8.5):

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leqslant 2\varepsilon^\top \Phi(\hat{\theta} - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}).$$

Then, with the dual norm $\Omega^*(z) = \sup_{\Omega(\theta) \leqslant 1} z^\top \theta$, assuming that $\Omega^*(\Phi^\top \varepsilon) \leqslant \frac{n\lambda}{2}$, and using the triangle inequality:

$$\begin{aligned} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 &\leqslant 2\Omega^*(\Phi^\top \varepsilon)\Omega(\hat{\theta} - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}) \\ &\leqslant n\lambda\Omega(\hat{\theta} - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}) \\ &\leqslant n\lambda\Omega(\hat{\theta}) + n\lambda\Omega(\theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}) \leqslant 3n\lambda\Omega(\theta_*) - n\lambda\Omega(\hat{\theta}). \end{aligned}$$

This implies that $\Omega(\hat{\theta}) \leqslant 3\Omega(\theta_*)$ and $\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leqslant 3\lambda\Omega(\theta_*)$.

We also have a general bound through:

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leqslant 2\|\varepsilon\|_2 \|\Phi(\hat{\theta} - \theta_*)\|_2 + 2n\lambda\Omega(\theta_*),$$

which leads to, using the identity $2ab \leqslant \frac{1}{2}a^2 + 2b^2$,

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leqslant \frac{1}{2}\|\Phi(\hat{\theta} - \theta_*)\|_2^2 + 2\|\varepsilon\|_2^2 + 2n\lambda\Omega(\theta_*),$$

which leads to the desired bound. ■

**Exercise 8.4** *For $p \in [1, \infty]$, show that the dual of the $\ell_p$-norm is the $\ell_q$-norm, for $\frac{1}{p} + \frac{1}{q} = 1$.*

We can now use the lemma above to compute the excess risk of the Lasso, for which $\Omega = \|\cdot\|_1$ and $\Omega^*(\Phi^\top \varepsilon) = \|\Phi^\top \varepsilon\|_\infty$.[3] The key is to note that since $\|\Phi^\top \varepsilon\|_\infty$ is a maximum of $2d$ zero-mean terms that scale as $\sqrt{n}$, according to Section 1.2.4, its maximum scales as $\sqrt{n \log(d)}$, and we will apply the lemma above when $\lambda$ is larger than $\sqrt{\log(d)/n}$. We denote by $\|\widehat{\Sigma}\|_\infty$ the largest element of the matrix $\widehat{\Sigma}$ in absolute value.

[3] Developments similar to Prop. 4.7 in Section 4.5.5 for general norms could also be carried out.

**Proposition 8.3 (Lasso - slow rate)** *Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance $\sigma^2$. Let $\hat\theta$ be the minimizer of Eq. (8.4). Then, for $\lambda = \frac{2\sigma}{\sqrt{n}}\sqrt{2\|\widehat\Sigma\|_\infty}\sqrt{\log(d) + \log\frac{1}{\delta}}$, we have, with probability greater than $1-\delta$:*

$$\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 3\|\theta_*\|_1 \cdot \frac{2\sigma}{\sqrt{n}}\sqrt{2\|\widehat\Sigma\|_\infty}\sqrt{\log(d) + \log\frac{1}{\delta}}.$$

**Proof** For each $j$, the random variable $(\Phi^\top\varepsilon)_j$ is Gaussian with mean zero and variance $n\sigma^2\widehat\Sigma_{jj}$. Thus, we get from the union bound and from the fact that for a standard Gaussian variable $z$, $\mathbb{P}(|z| \geqslant t) \leqslant \exp(-t^2/2)$:[4]

$$\mathbb{P}\Big(\|\Phi^\top\varepsilon\|_\infty > \frac{n\lambda}{2}\Big) \leqslant \sum_{j=1}^d \mathbb{P}\Big(|\Phi^\top\varepsilon|_j > \frac{n\lambda}{2}\Big) \leqslant \sum_{j=1}^d \exp\Big(\frac{-n\lambda^2}{8\sigma^2\widehat\Sigma_{jj}}\Big) \leqslant d\exp\Big(\frac{-n\lambda^2}{8\sigma^2\|\widehat\Sigma\|_\infty}\Big) = \delta.$$

Thus, with probability greater than $1-\delta$, we can apply the first part of Lemma 8.4, and therefore the error is less than $3\lambda\|\theta^*\|_1$. For a result in expectation, see the exercise below. ■

⚠ Check homogeneity!

We already observe some high-dimensional phenomenon with the term $\sqrt{\frac{\log d}{n}}$, where $n$ can be much larger than $d$ (if, of course, we assume that the optimal predictor $\theta_*$ is sparse, so that $\|\theta_*\|_1$ does not grow with $d$). Note that the proposed regularization parameter depends on the unknown noise variance. A simple trick known as the "square root Lasso" allows to avoid that dependence on $\sigma$ (see Giraud, 2014, Section 5.4), by minimizing $\frac{1}{\sqrt{n}}\|y - \Phi\theta\|_2 + \lambda\|\theta\|_1$.

The proposition above suggests a regularization parameter $\lambda$ proportional to $1/\sqrt{n}$, which does enable estimation in high-dimensional situations but can also add a significant bias because all non-zero components of $\hat\theta$ are shrunk towards zero. See Section 8.5 for methods to alleviate this effect.

**Exercise 8.5** (♦) *With the same assumptions as Prop. 8.3, and with the choice of regularization parameter $\lambda = 4\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat\Sigma\|_\infty}$, show the following bound in expectation: $\mathbb{E}\big[\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \leqslant 32\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat\Sigma\|_\infty}\|\theta_*\|_1 + \frac{32}{n}\sigma^2$.*

**Beyond square loss.** The slow rates proportional to $\|\theta_*\|_1\sqrt{(\log d)/n}$ for regularization by the $\ell_1$-norm can also be achieved for Lipschitz-continuous losses (such as the logistic loss and the hinge loss), as shown in Prop. 4.7 in Section 4.5.5.

[4] We have for $t \geqslant 0$: $e^{t^2/2}\mathbb{P}(|z| \geqslant t) = \frac{2}{\sqrt{2\pi}}\int_t^{+\infty} e^{t^2/2 - s^2/2}dt \leqslant \frac{2}{\sqrt{2\pi}}\int_t^{+\infty} e^{-(s-t)^2/2}dt = 1$.

### 8.3.4 Fast rates (♦)

We now consider conditions to obtain a fast rate with a leading term proportional to $\sigma^2 \frac{k \log d}{n}$, which is the same as for $\ell_0$-penalty, but with tractable algorithms. This will come with extra (very) strong conditions on the design matrix $\Phi$.

We start with a simple (but crucial) lemma, characterizing the solution of Eq. (8.4) in terms of the support $A$ of $\theta_*$.

**Lemma 8.5** *Let $\hat\theta$ be a minimizer of Eq. (8.5). Assume $\|\Phi^\top \varepsilon\|_\infty \leqslant \frac{n\lambda}{2}$. If $\Delta = \hat\theta - \theta_*$, then $\|\Delta_{A^c}\|_1 \leqslant 3\|\Delta_A\|_1$ and $\|\Phi\Delta\|_2^2 \leqslant 3n\lambda\|\Delta_A\|_1$.*

**Proof** We have, like in previous proofs (e.g., Lemma 8.4), with $\Delta = \hat\theta - \theta_*$, and $A$ the support of $\theta_*$:

$$\|\Phi\Delta\|_2^2 \leqslant 2\varepsilon^\top \Phi\Delta + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat\theta\|_1.$$

Then, assuming that $\|\Phi^\top\varepsilon\|_\infty \leqslant \frac{n\lambda}{2}$,

$$\begin{aligned}
\|\Phi\Delta\|_2^2 &\leqslant 2\|\Phi^\top\varepsilon\|_\infty \|\Delta\|_1 + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat\theta\|_1 \\
\|\Phi\Delta\|_2^2 &\leqslant n\lambda\|\Delta\|_1 + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat\theta\|_1.
\end{aligned}$$

We then use, by using the decomposability of the $\ell_1$-norm and the triangle inequality:

$$\|\theta_*\|_1 - \|\hat\theta\|_1 = \|(\theta_*)_A\|_1 - \|\theta_* + \Delta\|_1 = \|(\theta_*)_A\|_1 - \|(\theta_* + \Delta)_A\|_1 - \|\Delta_{A^c}\|_1 \leqslant \|\Delta_A\|_1 - \|\Delta_{A^c}\|_1,$$

to get

$$\begin{aligned}
\|\Phi\Delta\|_2^2 &\leqslant n\lambda\|\Delta\|_1 + 2n\lambda(\|\theta_*\|_1 - \|\hat\theta\|_1) \leqslant n\lambda\|\Delta\|_1 + 2n\lambda(\|\Delta_A\|_1 - \|\Delta_{A^c}\|_1) \\
&\leqslant n\lambda(\|\Delta_A\|_1 + \|\Delta_{A^c}\|_1) + 2n\lambda(\|\Delta_A\|_1 - \|\Delta_{A^c}\|_1) = 3n\lambda\|\Delta_A\|_1 - n\lambda\|\Delta_{A^c}\|_1.
\end{aligned}$$

This leads to $\|\Delta_{A^c}\|_1 \leqslant 3\|\Delta_A\|_1$ and the other desired inequality. ■

We can now add an extra assumption that will make the proof go through, namely that there exists a constant $\kappa > 0$ such that

$$\frac{1}{n}\|\Phi\Delta\|_2^2 \geqslant \kappa\|\Delta_A\|_2^2 \tag{8.6}$$

for all $\Delta$ that satisfies the condition $\|\Delta_{A^c}\|_1 \leqslant 3\|\Delta_A\|_1$. This is called the "restrictive eigenvalue property" because if the smallest eigenvalue of $\frac{1}{n}\Phi^\top\Phi$ is greater than $\kappa$, the condition is satisfied (but this is only possible if $n \geqslant d$). The relevance of this assumption is discussed in Section 8.3.5.

This leads to the following proposition.

**Proposition 8.4 (Lasso - fast rate)** *Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance $\sigma^2$. Let $\hat\theta$ be the minimizer of Eq. (8.4). Then, for $\lambda = \frac{2\sigma}{\sqrt{n}}\sqrt{2\|\widehat\Sigma\|_\infty}\sqrt{\log(2d) + \log\frac{1}{\delta}}$, we have, if Eq. (8.6) is satisfied, and with probability greater than $1-\delta$:*

$$\mathbb{E}\Big[\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2\Big] \leqslant \frac{72|A|\sigma^2}{n}\frac{\|\widehat\Sigma\|_\infty}{\kappa}\Big(\log(2d) + \log\frac{1}{\delta}\Big).$$

**Proof** (♦) We have, when $\lambda$ is large enough, and by application of Lemma 8.5, and using Eq. (8.6):

$$\|\Delta_A\|_1 \leqslant |A|^{1/2}\|\Delta_A\|_2 \leqslant \frac{|A|^{1/2}}{\sqrt{n\kappa}}\|\Phi\Delta\|_2 \leqslant \frac{|A|^{1/2}}{\sqrt{n\kappa}}\sqrt{3n\lambda\|\Delta_A\|_1},$$

which leads to $\|\Delta_A\|_1 \leqslant \dfrac{3|A|\lambda}{\kappa}$. We then get $\frac{1}{n}\|\Phi\Delta\|_2^2 \leqslant \frac{9|A|\lambda^2}{\kappa}$, which leads to the desired result.

■

The dominant part of the rate is proportional to $\sigma^2 k \frac{\log d}{n}$, which is a fast rate but depends crucially on a very strong assumption. Such results can be extended beyond the square loss using the notion of self-concordance (see, e.g., Ostrovskii and Bach, 2021b, and references therein).

**Exercise 8.6** *(♦♦) With the same assumptions as Prop. 8.4, with the choice of regularization parameter* $\lambda = 4\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat{\Sigma}\|_\infty}$*, show that we have the bound in expectation* $\mathbb{E}\big[\frac{1}{n}\|\Phi(\hat{\theta}-\theta_*)\|_2^2\big] \leqslant \frac{144|A|\sigma^2\|\widehat{\Sigma}\|_\infty}{\kappa}\frac{\log(dn)}{n} + \frac{24}{n}\sigma^2 + \frac{32}{dn^2}\|\theta_*\|_1\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat{\Sigma}\|_\infty}$.

### 8.3.5 Zoo of conditions (♦♦)

Conditions to obtain fast rates are plentiful: they all assume low correlation among predictors, which is rarely the case in practice (in particular, if there are two equal features, they are never satisfied).

**Restricted eigenvalue property (REP).** The most direct condition is the so-called restricted eigenvalue property (REP), which is exactly Eq. (8.6), with the supremum taken over the unknown set $A$ of cardinality less than $k$:

$$\inf_{|A|\leqslant k}\ \inf_{\|\Delta_{A^c}\|_1\leqslant 3\|\Delta_A\|_1} \frac{\|\Phi\Delta\|_2^2}{n\|\Delta_A\|_2^2} \geqslant \kappa > 0. \tag{8.7}$$

**Mutual incoherence condition.** A simpler one to check, but stronger, is the mutual incoherence condition:

$$\sup_{i\neq j}|\widehat{\Sigma}_{ij}| \leqslant \frac{\min_{j\in\{1,\dots,d\}}\widehat{\Sigma}_{jj}}{14k}, \tag{8.8}$$

which states that all cross-correlation coefficients are small (pure decorrelation would set them to zero).

This is weaker than the REP condition above. Indeed, by expanding, we have:

$$\begin{aligned}\|\Phi\Delta\|_2^2 &= \|\Phi_A\Delta_A + \Phi_{A^c}\Delta_{A^c}\|_2^2 = \|\Phi_A\Delta_A\|_2^2 + 2\Delta_A^\top\Phi_A^\top\Phi_{A^c}\Delta_{A^c} + \|\Phi_{A^c}\Delta_{A^c}\|_2^2\\ &\geqslant \|\Phi_A\Delta_A\|_2^2 + 2\Delta_A^\top\Phi_A^\top\Phi_{A^c}\Delta_{A^c}.\end{aligned}$$

Moreover, we have:

$$\begin{aligned}\Delta_A^\top \widehat{\Sigma}_{AA}\Delta_A &= \Delta_A^\top \operatorname{Diag}(\operatorname{diag}(\widehat{\Sigma}_{AA}))\Delta_A + \Delta_A^\top(\widehat{\Sigma}_{AA} - \operatorname{Diag}(\operatorname{diag}(\widehat{\Sigma}_{AA}))\Delta_A \\ &\geqslant \min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}\big(\|\Delta_A\|_2^2 - \frac{1}{14k}\|\Delta_A\|_1^2\big),\end{aligned}$$

and

$$|\Delta_A^\top \Phi_A^\top \Phi_{A^c}\Delta_{A^c}| \leqslant \frac{\min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}}{14k}\|\Delta_{A^c}\|_1\|\Delta_A\|_1 \leqslant \frac{3\min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}}{14k}\|\Delta_A\|_1^2.$$

This leads to $\frac{1}{n}\|\Phi\Delta\|_2^2 \geqslant \min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}\big(\|\Delta_A\|_2^2 - \frac{7}{14k}\|\Delta_A\|_1^2\big)$, which is greater than $\min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}\big(\|\Delta_A\|_2^2 - \frac{7k}{14k}\|\Delta_A\|_2^2\big) = \kappa\|\Delta_A\|_2^2$, with $\kappa = \min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}/2$, thus leading to the REP condition in Eq. (8.7).

**Restricted isometry property.** One of the earlier conditions was the restricted isometry property: all eigenvalues of submatrices of $\widehat{\Sigma}$ of size less than $2k$, are between $1-\delta$ and $1+\delta$ for $\delta$ small enough. See Giraud (2014); Wainwright (2019) for details.

**Gaussian designs (♦).** It is not obvious that the conditions above are non-trivial (that is, there may exist no matrix with good sizes $d$ and $n$ for $k$ large enough). For our results to be non-trivial, we need that $k\frac{\log d}{n}$ to be small but not too small. In this paragraph, we show, without proof, that when sampling from Gaussian distributions, the assumptions above are satisfied. This is a first step towards a random design assumption.

**Proposition 8.5 (Wainwright, 2019, Theorem 7.16)** *If sampling $\varphi(x)$ from a Gaussian with mean zero and covariance matrix $\Sigma$, then with probability greater than $1 - \frac{e^{-n/32}}{1-e^{-n/32}}$, the REP property is satisfied with $\kappa = \frac{1}{16}\lambda_{\min}(\Sigma)$ as soon as $k\frac{\log d}{n} \leqslant \frac{1}{3200}\frac{\lambda_{\min}(\Sigma)}{\|\Sigma\|_\infty}$.*

The proposition above is hard to prove; the following exercise proposes to establish a weaker result, showing that the guarantees for the maximal cardinality $k$ of the support have to be smaller.

**Exercise 8.7** *(♦♦♦) If sampling $\varphi(x)$ from a Gaussian with mean zero and covariance matrix identity, then with large probability, for $n$ greater than a constant times $k^2\frac{\log d}{n}$, the mutual incoherence property in Eq. (8.8) is satisfied.*

**Model selection and irrepresentable condition (♦).** Given that the Lasso aims at performing variable selection, it is natural to study its capacity to find the support of $\theta_*$, that is, the set of non-zero variables. It turns out that it also depends on some conditions on the design matrix, which are stronger than the REP conditions, and called the "irrepresentable condition", and also valid for Gaussian random matrices with similar scalings between $n$, $d$, and $k$. See Giraud (2014); Wainwright (2019) for details.

Algorithmic and theoretical tools are similar to "compressed sensing", where the design matrix represents a set of measurements, which the user/theoretician can choose. In this context, sampling from i.i.d. Gaussians makes sense. For machine learning and statistics, the design matrix is the data and comes as it is, often with strong correlations.

### 8.3.6 Random design (♦)

In this section, we study the Lasso in the random design setting instead of the fixed design setting. For slow rates in $1/\sqrt{n}$, we can directly use Section 4.5.5 to get the exact same slow rate as for fixed design. In this section, we will only consider fast rates.

We now consider the well-specified Lasso case, where the expected risk is equal to $\mathcal{R}(\theta) = \frac{\sigma^2}{2} + \frac{1}{2}(\theta-\theta^*)^\top\Sigma(\theta-\theta^*)$. We assume that $\lambda_{\min}(\Sigma) \geqslant \mu \geqslant 0$, that is, the *expected* risk is $\mu$-strongly convex (and not the empirical risk).

We assume that $y_i = \varphi(x_i)^\top\theta^* + \varepsilon_i$, and denote $\Phi \in \mathbb{R}^{n\times d}$ the design matrix, as well as $\varepsilon \in \mathbb{R}^n$ the vector of noises, which we assume independent and sub-Gaussian. Therefore we have

$$\widehat{\mathcal{R}}(\theta) = \frac{1}{2n}\|\Phi(\theta-\theta^*)-\varepsilon\|_2^2 = \frac{1}{2}(\theta-\theta^*)^\top\widehat{\Sigma}(\theta-\theta^*) - (\theta-\theta^*)^\top\Big(\frac{1}{n}\Phi^\top\varepsilon\Big) + \frac{1}{2n}\|\varepsilon\|_2^2, \tag{8.9}$$

where $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top = \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d\times d}$ is the empirical non-centered covariance matrix.

We will need that $\big\|\frac{1}{n}\Phi^\top\varepsilon\big\|_\infty = \big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i\varphi(x_i)\big\|_\infty$ is small enough, as well as the error in the covariance matrix $\big\|\widehat{\Sigma}-\Sigma\big\|_\infty$. Assuming that $\varepsilon$ is sub-Gaussian with constant $\sigma^2$, and that $\|\varphi(x)\|_\infty \leqslant R$ almost surely, we get that, using results from Section 1.2.1,

$$\mathbb{P}\Big(\big\|\frac{1}{n}\Phi^\top\varepsilon\big\|_\infty \geqslant \frac{\sigma R t}{\sqrt{n}}\Big) \leqslant 2d\exp(-t^2/2) \text{ and } \mathbb{P}\Big(\big\|\widehat{\Sigma}-\Sigma\big\|_\infty \geqslant \frac{R^2 t}{\sqrt{n}}\Big) \leqslant 2d(d+1)/2\exp(-t^2/2).$$

Thus, the probability that at least one is satisfied is less than $d(d+3)\exp(-t^2/2) \leqslant 4d^2\exp(-t^2/2)$.

We now assume that $\big\|\frac{1}{n}\Phi^\top\varepsilon\big\|_\infty \leqslant \frac{\sigma R t}{\sqrt{n}}$ and $\big\|\widehat{\Sigma}-\Sigma\big\|_\infty \leqslant \frac{R^2 t}{\sqrt{n}}$, which happens with probability at least $1-4d^2\exp(-t^2/2)$. From Lemma 8.5, we know that if $\lambda \geqslant 2\big\|\frac{1}{n}\Phi^\top\varepsilon\big\|_\infty$, then we have, with $\hat{\Delta} = \hat{\theta}_\lambda - \theta^*$, and $A$ the support of $\theta^*$:

$$\|\hat{\Delta}_{A^c}\|_1 \leqslant 3\|\hat{\Delta}_A\|_1 \text{ and } \|\hat{\theta}_\lambda\|_1 \leqslant 3\|\theta^*\|_1.$$

Let $v = \mathcal{R}(\hat\theta_\lambda) - \mathcal{R}(\theta^*)$. We have:

$$\begin{array}{rcl} v & \leqslant & \mathcal{R}(\hat\theta_\lambda) - \mathcal{R}(\theta^*) - \widehat{\mathcal{R}}_\lambda(\hat\theta_\lambda) + \widehat{\mathcal{R}}_\lambda(\theta^*) \text{ since } \hat\theta_\lambda \text{ minimizes } \widehat{\mathcal{R}}_\lambda, \\ & = & \mathcal{R}(\hat\theta_\lambda) - \mathcal{R}(\theta^*) - \widehat{\mathcal{R}}(\hat\theta_\lambda) + \widehat{\mathcal{R}}(\theta^*) + \lambda\|\theta^*\|_1 - \lambda\|\hat\theta_\lambda\|_1 \text{ by definition of } \widehat{\mathcal{R}}_\lambda, \\ & = & \frac{1}{2}\hat\Delta^\top(H - \hat H)\hat\Delta + \hat\Delta^\top\Big(\frac{1}{n}\Phi^\top\varepsilon\Big) + \lambda\|\theta^*\|_1 - \lambda\|\hat\theta_\lambda\|_1 \text{ using Eq. (8.9) }, \\ & \leqslant & \frac{1}{2}\big\|\widehat\Sigma - \Sigma\big\|_\infty \cdot \|\hat\Delta\|_1^2 + \big\|\frac{1}{n}\Phi^\top\varepsilon\big\|_\infty \cdot \|\hat\Delta\|_1 + \lambda\|\hat\Delta\|_1 \text{ using norm inequalities}, \\ & \leqslant & \frac{\sigma R t}{\sqrt{n}} \cdot \|\hat\Delta\|_1 + \frac{R^2 t}{2\sqrt{n}} \cdot \|\hat\Delta\|_1^2 + \lambda\|\hat\Delta\|_1 \text{ using our assumptions.} \end{array}$$

Moreover, we have, since $\lambda_{\min}(\Sigma) \geqslant \mu$, $v = \mathcal{R}(\hat\theta_\lambda) - \mathcal{R}(\theta^*) \geqslant \frac{\mu}{2}\|\hat\Delta\|_2^2 \geqslant \frac{\mu}{2|A|}\|\hat\Delta_A\|_1^2$, leading to $\|\hat\Delta\|_1 \leqslant 4\|\hat\Delta_A\|_1 \leqslant 4\sqrt{\frac{2|A|v}{\mu}}$. We also have $\|\hat\Delta\|_1 \leqslant \|\theta^*\|_1 + \|\hat\theta_\lambda\|_1 \leqslant \|\theta^*\|_1 + 3\|\theta^*\|_1 \leqslant 4\|\theta^*\|_1$. We thus get, with $\lambda = \frac{2\sigma R t}{\sqrt{n}}$, two inequalities:

$$v \leqslant \frac{3\sigma R t}{\sqrt{n}} \cdot \|\hat\Delta\|_1 + \frac{R^2 t}{2\sqrt{n}} \cdot \|\hat\Delta\|_1^2 \text{ and } \|\hat\Delta\|_1 \leqslant 4\sqrt{\frac{2|A|v}{\mu}}. \tag{8.10}$$

If $1 \geqslant \frac{32R^2 t}{\sqrt{n}}\frac{|A|}{\mu}$, then the last term in the first inequality in Eq. (8.10) is less than $\frac{v}{2}$, and we get $\frac{v}{2} \leqslant \frac{3\sigma R t}{\sqrt{n}} 4\sqrt{\frac{2|A|v}{\mu}}$, that is, $\sqrt{v} \leqslant \frac{24\sigma R t}{\sqrt{n}}\sqrt{\frac{2|A|}{\mu}}$. This leads to, with $\lambda = \frac{2\sigma R}{\sqrt{n}}t = \frac{2\sigma R}{\sqrt{n}}\sqrt{2\log\frac{4d^2}{\delta}}$, with probability greater than $1-\delta$,

$$\mathcal{R}(\hat\theta_\lambda) - \mathcal{R}(\theta^*) \leqslant 2304 \cdot \frac{R^2}{\mu}\frac{\sigma^2|A|}{n}\log\frac{4d^2}{\delta}.$$

**Exercise 8.8** *With the notations above, show that if $\mu = 0$, from Eq. (8.10) we can recover the slow rate* $\mathcal{R}(\hat\theta_\lambda) - \mathcal{R}(\theta^*) \leqslant \frac{4R\|\theta^*\|_1}{\sqrt{n}}(3\sigma + 2R\|\theta^*\|_1)\sqrt{2\log\frac{4d^2}{\delta}}$.

## 8.4 Experiments

In this section, we perform a simple experiment on Gaussian design matrices, where all entries in $\Phi \in \mathbb{R}^{n\times d}$ are sampled independently from a standard Gaussian distribution, with $n = 64$, and varying $d$. Then $\theta_*$ is taken to be zero except on $k = 4$ components where it is randomly equal to $-1$ or 1. We consider $\sigma = \sqrt{k}$ (to have a constant signal-to-noise ratio when $k$ varies). We perform 128 replications. For each method and each value of its hyperparameter, we averaged the test risk over the 128 replications and reported the minimum value (with respect to the hyperparameter). We compare the following three methods in Figure 8.2:

- Ridge regression: penalty by $\lambda\|\theta\|_2^2$.

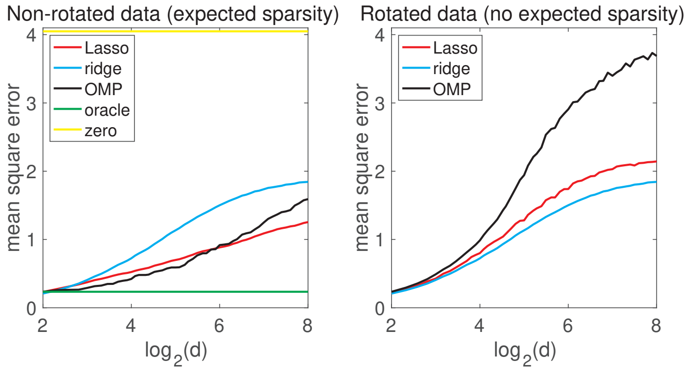


Figure 8.2: Comparison of estimators on least-squares regression: problem with sparse optimal predictor (left), and non-sparse optimal predictor (right).

- Lasso regression: penalty by $\lambda\|\theta\|_1$.
- Orthogonal matching pursuit (greedy forward method), with hyperparameter $k$ (the number of included variables).

We compare two situations: (1) non-rotated data (exactly the model above), and (2) rotated data, where we replace $\Phi$ by $\Phi R$ and $\theta_*$ by $R^\top \theta_*$, where $R$ is a random rotation matrix. For the rotated data, we do not expect sparse solutions. Hence, sparse methods are not expected to work better than ridge regression (and OMP performs significantly worse because once the support is chosen, there is no regularization). Note that the two curves for ridge regression are exactly the same (as expected from rotation invariance of the $\ell_2$-norm). The oracle performance corresponds to the estimator where the true support is given.

Sparse methods make assumptions regarding the best predictor. Like all assumptions, when this assumed prior knowledge is not correct, the method does not perform better.

## 8.5 Extensions

Sparse methods are more general than the $\ell_1$-norm and can be extended in several ways:

- **Group penalties:** in many cases, $\{1, \dots, d\}$ is partitioned into $m$ subsets $A_1, \dots,$ $A_m$, and the goal is to consider “group sparsity,” that is, if we select one variable within a group $A_j$, the entire group should be selected. Such behavior can be obtained using the penalty $\sum_{i=1}^m \|\theta_{A_i}\|_2$ or $\sum_{i=1}^m \|\theta_{A_i}\|_\infty$. This is particularly used when the output $y$ is multi-dimensional (such as in multivariate regression or multi-category classification) to select variables relevant to all outputs. See, e.g., Giraud (2014) for details.

  **Exercise 8.9** *Assuming that the design matrix $\Phi$ is orthogonal, compute the min-*

*imizer of* $\frac{1}{2n}\|y - \Phi\theta\|_2^2 + \lambda \sum_{i=1}^m \|\theta_{A_i}\|_2$.

- **Structured sparsity:** It is also possible to favor other specific patterns for the selected variables, such as blocks, trees, etc., when such prior knowledge is needed. See Bach et al. (2012b) for details.

  **Exercise 8.10** *We consider the* $d$ *(overlapping) sets* $A_i = \{1, \dots, i\}$, *and the norm* $\sum_{i=1}^d \|\theta_{A_i}\|_2$. *Show that penalization with this norm will tend to select patterns of non-zeros of the form* $\{i+1, \dots, d\}$.

- **Nuclear norm:** When learning on matrices, a natural form of sparsity is for a matrix to have a low rank. This can be achieved by penalizing the sum of singular values of a matrix, a norm called the nuclear norm or the trace norm. See Bach (2008) and references therein.

  **Exercise 8.11** *Compute the minimizer of* $\frac{1}{2n}\|Y - \Theta\|_F^2 + \lambda\|\Theta\|_*$, *where* $\|M\|_F$ *is the Frobenius norm and* $\|M\|_*$ *the nuclear norm.*

- **Multiple kernel learning:** the group penalty can be extended when the groups have an infinite dimension and $\ell_2$-norms are replaced by RKHS norms defined in Chapter 7. This becomes a tool for learning the kernel matrix from data. See Bach et al. (2012a) for details.

- **Elastic net:** often, when both effects of the $\ell_1$-norm (sparsity) and the squared $\ell_2$-norm (strong-convexity) are desired, we can sum the two, which is referred to as the "elastic net" penalty. This leads to a strongly-convex optimization problem, which is numerically better behaved.

- **Concave penalization and debiasing:** to obtain a sparsity-inducing effect, the penalty in the $\ell_1$-norm has to be quite large, such as in $1/\sqrt{n}$, which often creates a strong bias in the estimation once the support is selected. There are several ways of debiasing the Lasso, an elegant one being to use a "concave" penalty. That is, we use $\sum_{i=1}^d a(|\theta_i|)$ where $a$ is a concave increasing function on $\mathbb{R}^+$, such as $a(u) = u^\alpha$ for $\alpha \in (0,1)$. This leads to a non-convex optimization problem, where iterative weighted $\ell_1$-minimization provides natural algorithms (see Mairal et al., 2014, and references therein).

## 8.6 Conclusion

In this chapter, we have considered sparse methods based on the penalization by the $\ell_0$ or $\ell_1$ penalties of the weight vector of a linear model. For the square loss, $\ell_0$-penalties led to an excess risk proportional to $\sigma^2 k \log(d)/n$, with a price of adaptivity of $\log(d)$, with few conditions on the problem, but no provably computationally efficient procedures. On the contrary, $\ell_1$-norm penalization can be solved efficiently with appropriate convex optimization algorithms (such as proximal methods) but only obtained a slow rate proportional to $\sqrt{\log(d)/n}$, exhibiting a high-dimensional phenomenon, but a worse dependence in $n$.

Fast rates can only be obtained with stronger assumptions on the covariance matrix of the features.

This chapter was limited to linear models, and in the next chapter on neural networks, we will see how models that are non-linear in their parameters can lead to non-linear variable selection, still exhibiting a high-dimensional phenomenon but at the expense of a harder optimization.

# Chapter 9

# Neural networks

**Chapter summary**

– Neural networks are flexible models for non-linear predictions. They can be studied in terms of the three errors usually related to empirical risk minimization: optimization, estimation, and approximation errors. In this chapter, we focus primarily on single hidden layer neural networks, which are linear combinations of simple affine functions with additional non-linearities.

– Optimization error: as the prediction functions are non-linearly dependent on their parameters, we obtain non-convex optimization problems with only guaranteed convergence to stationary points.

– Estimation error: the number of parameters is not the driver of the estimation error, as the norms of the various weights play an important role, with explicit rates in $O(1/\sqrt{n})$ obtained from Rademacher complexity tools.

– Approximation error: for the "ReLU" activation function, the universal approximation properties can be characterized and are superior to kernel methods because they are adaptive to linear latent variables.

## 9.1 Introduction

In supervised learning, the main focus has been put on methods to learn from $n$ observations $(x_i, y_i), i = 1, \dots, n$, with $x_i \in \mathcal{X}$ (input space) and $y_i \in \mathcal{Y}$ (output/label space). As presented in Chapter 4, a large class of methods relies on minimizing a regularized empirical risk with respect to a function $f : \mathcal{X} \to \mathbb{R}$ where the following cost function is minimized:

$$\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \Omega(f),$$

where $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a loss function, and $\Omega(f)$ is a regularization term. Typical examples were:

- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y_i, f(x_i)) = \frac{1}{2}(y_i - f(x_i))^2$.
- **Classification:** $\mathcal{Y} = \{-1, 1\}$ and $\ell(y_i, f(x_i)) = \Phi(y_i f(x_i)))$ where $\Phi$ is convex, e.g., $\Phi(u) = \max\{1 - u, 0\}$ (hinge loss leading to the support vector machine) or $\Phi(u) = \log(1 + \exp(-u))$ (leading to logistic regression). See more examples in Section 4.1.1.

The class of prediction functions we have considered so far were (with their "pros" and "cons"):

- **Linear functions in some explicit features:** given a feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$, we consider $f(x) = \theta^\top \varphi(x)$, with parameters $\theta \in \mathbb{R}^d$, as analyzed in Chapter 3 (for least-squares) and Chapter 4 (for Lipschitz-continuous losses).

  – *Pros:* Simple to implement, as this leads to convex optimization with gradient descent algorithms, with running time complexity in $O(nd)$, as shown in Chapter 5, and theoretical guarantees which are not necessary scaling badly with dimension $d$ if regularizers are used ($\ell_2$ or $\ell_1$).
  – *Cons:* Only applies to linear functions on explicit (and fixed feature spaces), so they can underfit the data.

- **Linear functions in some implicit features through kernel methods:** the feature map can have arbitrarily large dimension, that is, $\varphi(x) \in \mathcal{H}$ where $\mathcal{H}$ is a Hilbert space, accessed through the kernel function $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$, as presented in Chapter 7.

  – *Pros:* Non-linear flexible predictions, simple to implement, can be used as convex optimization algorithms with strong guarantees. Provides adaptivity to the regularity of the target function, allowing higher-dimensional applications than local averaging methods from Chapter 6.
  – *Cons:* Running-time complexity up to $O(n^2)$ with algorithms from Section 7.4 (but this scaling can be improved with appropriate techniques also discussed in the same section, such as column sampling or random features). The method may still suffer from the curse of dimensionality for target functions that are not smooth enough.

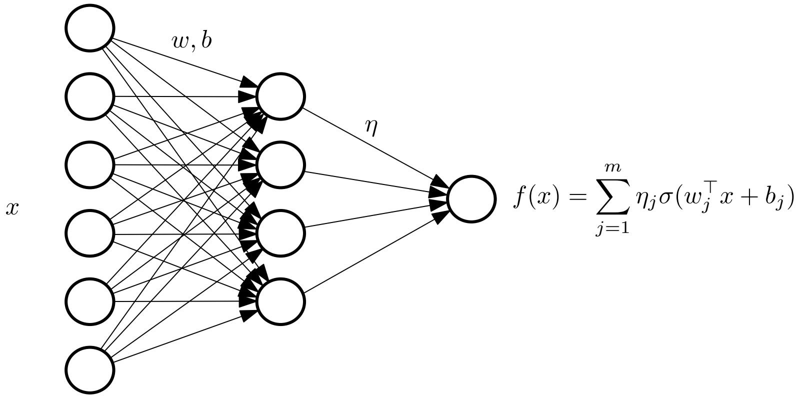
This chapter aims to explore another class of functions for non-linear predictions, namely neural networks, that come with additional benefits, such as more "adaptivity to linear latent variables", but comes with some potential drawbacks, such as a harder optimization problem.

## 9.2 Single hidden layer neural network

We consider $\mathcal{X} = \mathbb{R}^d$ and the set of prediction functions that can be written as

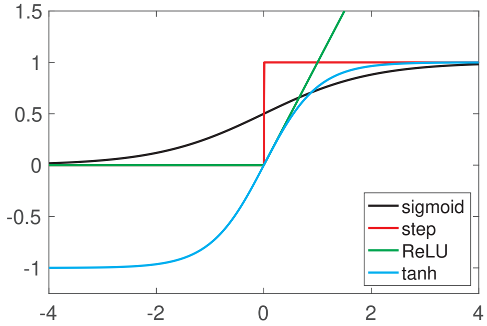
$$f(x) = \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j), \tag{9.1}$$

where $w_j \in \mathbb{R}^d$, $b_j \in \mathbb{R}$, $j = 1, \dots, m$, are the "input weights", $\eta_j \in \mathbb{R}$, , $j = 1, \dots, m$, are the "output weights", and $\sigma$ is an "activation function". This is often represented as a graph (see below). The same architecture can also be considered with $\eta_j \in \mathbb{R}^k$, for $k > 1$ to deal with multi-category classification (see Section 13.1).



The activation function is typically chosen from one of the following examples (see plot below):

- sigmoid $\sigma(u) = \frac{1}{1+e^{-u}}$,
- step function $\sigma(u) = 1_{u>0}$,
- "rectified linear unit" (ReLU) $\sigma(u) = (u)_+ = \max\{u, 0\}$, which will be the main focus of this chapter.
- hyperbolic tangent $\sigma(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$.



The function $f$ is defined as the linear combination of $m$ functions $x \mapsto \sigma(w_j^\top x + b_j)$, which are the "hidden neurons".[1]

⚠ The constant terms $b_j$ are sometimes referred to as "biases", which is unfortunate in a statistical context, as it already has a precise meaning within the bias/variance trade-off (see Chapter 3).

⚠ Do not get confused by the name "neural network" and its biological inspiration. This inspiration is not a proper justification for its behavior on machine learning problems.

**Cross-entropy loss and sigmoid activation function for the last layer.** Following standard practice, we are not adding a non-linearity to the last layer; note that if we were to use an additional sigmoid activation and consider the cross-entropy loss for binary classification, we would exactly be using the logistic loss on the output without an extra activation function.

Indeed, if we consider $g(x) = \frac{1}{1+\exp(-f(x))} \in [0,1]$, and given an output variable $y \in \{-1,1\}$, the so-called "cross-entropy loss", an instance of maximum likelihood (see more details in Chapter 14), is equal to $-\frac{1+y}{2}\log g(x) - \frac{1-y}{2}\log(1-g(x))$. It can be rewritten as $\log\big(1+\exp(-yf(x))\big)$, which is exactly the logistic loss defined in Section 4.1.1, applied to $f(x)$.

**Theoretical analysis of neural networks.** As with any method based on empirical risk minimization, we have to study the three classical aspects: (1) optimization (convergence properties of algorithms for minimizing the risk), (2) estimation error (effect of having a finite amount of data on the prediction performance), and (3) approximation error (effect of having a finite number of parameters or a constraint on the norm of these parameters).

### 9.2.1 Optimization

To find parameters $\theta = \{(\eta_j),(w_j),(b_j)\} \in \mathbb{R}^{m(d+2)}$, empirical risk minimization can be applied, and the following optimization problem has to be solved:

$$\min_{\theta \in \mathbb{R}^{m(d+2)}} \frac{1}{n}\sum_{i=1}^{n} \ell\Big(y_i, \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x_i + b_j)\Big),$$

with potentially additional regularization (often squared $\ell_2$-norm of all weights).

⚠ Note that (as discussed in Chapter 5) the true objective is to perform well on unseen data, and the optimization problem above is just a means to an end.

This is a non-convex optimization problem where the gradient descent algorithms from Chapter 5 can be applied without a strong guarantee beyond obtaining a vector with a

[1] See https://playground.tensorflow.org/ for a nice interactive illustration.

small gradient norm (Section 5.2.6). See below for recent results on providing qualitative global convergence guarantees when $m$ is large.

While stochastic gradient descent remains an algorithm of choice (with also a good generalization behavior as discussed in Section 5.4), several algorithmic improvements have been observed to lead to better stability and performance: specific step-size decay schedules, preconditioning like presented in Section 5.4.2 (Duchi et al., 2011), momentum (Kingma and Ba, 2014), batch-normalization (Ioffe and Szegedy, 2015) or layer-normalization (Ba et al., 2016) to make the optimization better behaved, but overall, the objective function is non-convex, and it remains challenging to understand precisely why gradient-based methods perform well in practice, particularly for deeper networks (some elements are presented below and in Chapter 12). See also boosting procedures in Section 10.3 and Chapter 12, which learn neuron weights incrementally.

**Global convergence of gradient descent for infinite widths (♦).** It turns out that global convergence can be shown for this non-convex optimization problem (Chizat and Bach, 2018; Bach and Chizat, 2022), with tools that go beyond the scope of this book and which are partially described in Chapter 12.[2]

We simply show some experimental evidence below for a simple one-dimensional set-up, where we compare several runs of stochastic gradient descent (SGD) where observations are only seen once (so no overfitting is possible) and with random initializations, on a regression problem with deterministic outputs, thus with the optimal testing error (the Bayes rate) being equal to zero. We show in Figure 9.1 the estimated predictors and the corresponding testing errors with 20 different initializations. We see that small errors are never achieved when $m = 5$ (which is sufficient to attain zero testing errors). With $m = 20$ neurons, SGD finds the optimal predictor for most restarts. When $m = 100$, all restarts have the desired behaviors, highlighting the benefits of over-parameterization.

### 9.2.2 Rectified linear units and homogeneity

From now on, we will mostly focus on the rectified linear unit $\sigma(u) = u_+$. The main property we will leverage is its "positive homogeneity", that is, for $\alpha > 0$, $(\alpha u)_+ = \alpha u_+$. This implies that in the definition of the prediction function as the sum of terms $\eta_j(w_j^\top x + b_j)_+$, we can freely multiply $\eta_j \in \mathbb{R}$ by a positive scalar $\alpha_j$ and divide $(w_j, b_j) \in \mathbb{R}^{d+1}$ by the same $\alpha_j$, without changing the prediction function.

This has a particular effect when using a squared $\ell_2$-regularizer on all weights, which is standard, either explicitly (by adding a penalty to the cost function) or implicitly (see Section 12.1). Indeed, we consider penalizing $\eta_j^2 + \|w_j\|_2^2 + b_j^2/R^2$ for each $j \in \{1, \dots, m\}$, where we have added the factor $R^2$ on the constant term for homogeneity reasons between the slope $w_j$ and the constant term $b_j$ ($R$ will be a bound on the $\ell_2$-norm of input data). Dealing with unit homogeneity between $\eta_j$ and $(w_j, b_j/R)$ does not matter, because of the invariance by rescaling described below.

[2] See also https://francisbach.com/gradient-descent-neural-networks-global-convergence/ for more details.

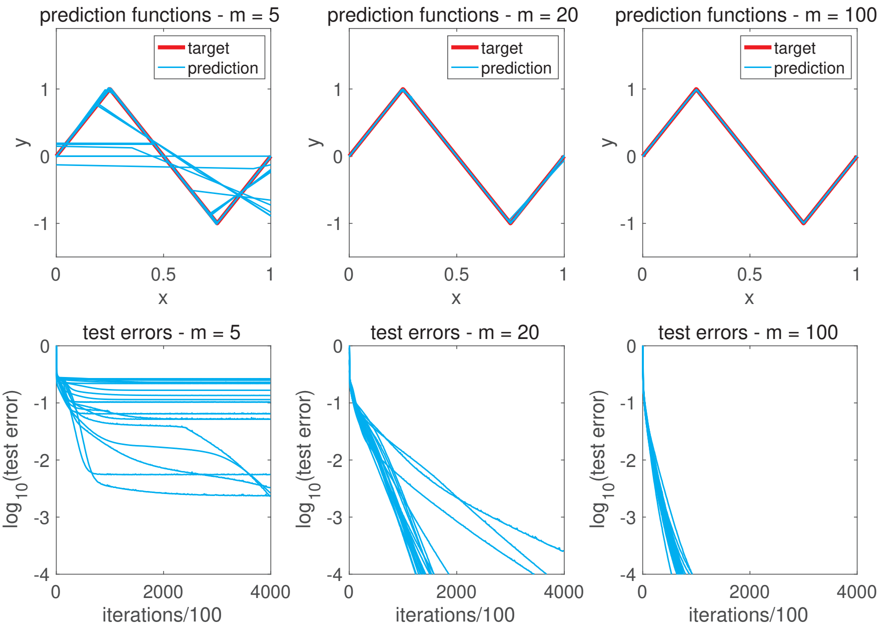


Figure 9.1: Comparison of optimization behavior for different numbers $m$ of neurons, for ReLU activations (left: $m = 5$, middle: $m = 20$, and right: $m = 100$). The neural network used to generate the data (without noise) has 4 hidden neurons. Top: examples of final prediction functions at convergence, bottom: plot of test errors vs. number of iterations.

Optimizing with respect to a scaling factor $\alpha_j$ above (which impacts only the regularizer), we have to minimize $\alpha_j^2\eta_j^2+\big(\|w_j\|_2^2+b_j^2/R^2\big)/\alpha_j^2$, with $\alpha_j^2 = \big[(\|w_j\|_2^2+b_j^2/R^2)^{1/2}\big]/|\eta_j|$ as a minimizer, and with the optimal value of the penalty equal to $2|\eta_j|(\|w_j\|_2^2+b_j^2/R^2)^{1/2}$.

For the theoretical analysis, we can thus choose to normalize each $(w_j, b_j)$ to have unit norm $\|w_j\|_2^2 + b_j^2/R^2 = 1$, and use the penalty $|\eta_j|$ for each $j \in \{1, \dots, m\}$, and thus use an overall $\ell_1$-norm penalty on $\eta$, that is, $\|\eta\|_1$ (we will consider other normalizations for the input weights below, either to ease the exposition, or to induce another behavior, e.g., by using $\ell_1$-norms on the $w_j$'s). We now focus on this choice of regularization in the following sections.

In this chapter, $R$ denotes an almost upper-bound on $x$ directly, and not on a feature map $\varphi(x)$ (as done in earlier chapters).

### 9.2.3 Estimation error

To study the estimation error, we will consider that the parameters of the network are constrained, that is, $\|w_j\|_2^2 + b_j^2/R^2 \leqslant 1$ for each $j \in \{1, \dots, m\}$, and $\|\eta\|_1 \leqslant D$. This defines a set $\Theta$ of allowed parameters. Note that we use $\|w_j\|_2^2 + b_j^2/R^2 \leqslant 1$ instead of $\|w_j\|_2^2 + b_j^2/R^2 = 1$ (as suggested above) as it does not impact the bound on estimation error (and uniform convergence results on a bigger set of functions apply to a smaller set).

We can then compute the Rademacher complexity of the associated class $\mathcal{F}$ of functions we just defined, using tools from Chapter 4 (Section 4.5). We assume that almost surely, $\|x\|_2 \leqslant R$, that is, the input data are bounded in $\ell_2$-norm by $R$.

Following the developments of Section 4.5 on Rademacher averages, we denote by $\mathcal{G} = \{(x, y) \mapsto \ell(y, f(x)),\ f \in \mathcal{F}\}$, the set of loss functions for a prediction function $f \in \mathcal{F}$ (which is here the set of neural network models $f_\theta$ with parameters $\theta$ such that $\theta \in \Theta$). Note that following Section 4.5.3, we consider a constraint on $\|\eta\|_1$, but we could also penalize, which is closer to practice and can be tackled with tools from Section 4.5.5.

We have, by definition of the Rademacher complexity $\mathrm{R}_n(\mathcal{G})$ of $\mathcal{G}$, and taking expectations with respect to the data $(x_i, y_i)$, $i = 1, \dots, n$ (which are assumed i.i.d.) and the independent Rademacher random variables $\varepsilon_i \in \{-1, 1\}$, $i = 1, \dots, n$:

$$\mathrm{R}_n(\mathcal{G}) = \mathbb{E}\Big[\sup_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \ell(y_i, f_\theta(x_i))\Big].$$

This quantity is known to provide an upper-bound on the expected risk (e.g., testing error) $\mathcal{R}(\hat{f})$ of the minimizer $\hat{f} \in \mathcal{F}$ of the empirical risk, through the estimation error, as (using symmetrization from Prop. 4.2 and Eq. (4.8) from Section 4.4):

$$\mathbb{E}\Big[\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f)\Big] \leqslant 4\mathrm{R}_n(\mathcal{G}).$$

We can now use properties of Rademacher complexities presented in Section 4.5, particularly their nice handling of non-linearities. Assuming the loss is almost surely $G$-Lipschitz-continuous with respect to the second variable, using Proposition 4.3 from Chapter 4 that allows getting rid of the loss, we get the bound:

$$\mathrm{R}_n(\mathcal{G}) \leqslant G \cdot \mathbb{E}\Big[\sup_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f_\theta(x_i)\Big] = G \cdot \mathbb{E}\Big[\sup_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^n \sum_{j=1}^m \eta_j \varepsilon_i \sigma(w_j^\top x_i + b_j)\Big].$$

Using the $\ell_1$-constraint on $\eta$ and using $\sup_{\|\eta\|_1 \leqslant D} z^\top \eta = D\|z\|_\infty$, we can directly maximize with respect to $\eta \in \mathbb{R}^m$, leading to (note that another $\ell_p$-constraint on $\eta$, with $p \neq 1$, would be harder to deal with):

$$\mathrm{R}_n(\mathcal{G}) \leqslant G \cdot \mathbb{E}\Big[\sup_{j\in\{1,\dots,m\}} \sup_{\|w_j\|_2^2 + b_j^2/R^2 \leqslant 1} D\Big|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \sigma(w_j^\top x_i + b_j)\Big|\Big].$$

Since the ReLU activation function $\sigma$ is 1-Lipschitz continuous and satisfies $\sigma(0) = 0$, we get, this time using the extension of Proposition 4.3 from Chapter 4 to Rademacher complexities defined with an absolute value (that is, Prop. 4.4), which adds an extra factor of 2:

$$\mathrm{R}_n(\mathcal{G}) \leqslant 2GD \cdot \mathbb{E}\Big[\sup_{j\in\{1,\dots,m\}} \sup_{\|w_j\|_2^2 + b_j^2/R^2 \leqslant 1} \Big|w_j^\top\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big) + b_j\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)\Big|\Big].$$

We can now perform the optimization with respect to $(w_j, b_j)$ in closed form (which can be done using Cauchy-Schwarz inequality), with the same value for all $j \in \{1, \dots, m\}$, leading to:

$$\mathrm{R}_n(\mathcal{G}) \leqslant 2GD \cdot \mathbb{E}\Big[\Big(\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_2^2 + R^2\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big)^{1/2}\Big].$$

We thus get, using Jensen's inequality (here of the form $\mathbb{E}[Z] \leqslant \sqrt{\mathbb{E}[Z^2]}$), as well as the independence, zero mean, and unit variance of $\varepsilon_1, \dots, \varepsilon_n$:

$$\begin{aligned} \mathrm{R}_n(\mathcal{G}) &\leqslant 2GD\Big(\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_2^2 + R^2\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big]\Big)^{1/2} \\ &= 2GD\Big(\frac{1}{n}\mathbb{E}[\|x\|_2^2] + \frac{R^2}{n}\Big)^{1/2} \leqslant \frac{2GDR\sqrt{2}}{\sqrt{n}}. \end{aligned} \tag{9.2}$$

Thus, we get the following proposition, with a bound proportional to $1/\sqrt{n}$ *with no explicit dependence in the number of parameters.*

**Proposition 9.1** *Let $\mathcal{G}$ be the class of functions $(y, x) \mapsto \ell(y, f(x))$ where $f$ is a neural network defined in Eq. (9.1), with the constraint that $\|\eta\|_1 \leqslant D$, $\|w_j\|_2^2 + b_j^2/R^2 \leqslant 1$ for all $j \in \{1, \dots, m\}$. If the loss function is G-Lipschitz-continuous and the activation function $\sigma$ is the ReLU, the Rademacher complexity is upper bounded as*

$$\mathrm{R}_n(\mathcal{G}) \leqslant \frac{4GDR}{\sqrt{n}}.$$

The proposition above allows for the estimation error to be bounded for neural networks, as the maximal deviation between expected risk and empirical risk over all potential networks with bounded parameters is bounded in expectation by four times the Rademacher complexity above.

This will be combined with a study of the approximation properties in Section 9.3, with a summary in Section 9.4.

⚠ For the estimation error, the number of parameters is irrelevant!
What counts is the overall norm of the weights.

We will see in Chapter 12 some recent results showing how optimization algorithms add an implicit regularization that leads to provable generalization in over-parameterized neural networks (that is, networks with many hidden units).

**Exercise 9.1** *(♦) Provide a bound similar to Prop. 9.1 for the constraint* $\|w_j\|_1 + |b_j|/R \leqslant 1$*, where* $R$ *denotes the supremum of* $\|x\|_\infty$ *over all* $x$ *in the support of its distribution.*

Before moving on to approximation properties of neural networks, we note that the reasoning above to compute the Rademacher complexity can be extended by recursion to deeper networks, as the following exercise shows (see, e.g., Neyshabur et al., 2015, for further results).

**Exercise 9.2** *(♦) We consider a* $1$*-Lipschitz-continuous activation function* $\sigma$ *such that* $\sigma(0) = 0$*, and the classes of functions defined recursively as* $\mathcal{F}_0 = \{x \mapsto \theta^\top x, \|\theta\|_2 \leqslant D_0\}$*, and, for* $i = 1, \dots, M$*,* $\mathcal{F}_i = \{x \mapsto \sum_{j=1}^{m_i} \theta_j \sigma(f_j(x)), f_j \in \mathcal{F}_{i-1}, \|\theta\|_1 \leqslant D_i\}$*, corresponding to a neural network with* $M$ *layers. Assuming that* $\|x\|_2 \leqslant R$ *almost surely, show by recursion that the Rademacher complexity satisfies* $\mathrm{R}_n(\mathcal{F}_M) \leqslant 2^M \frac{R}{\sqrt{n}} \prod_{i=0}^M D_i$*.*

## 9.3 Approximation properties

As seen above, the estimation error for constrained output weights grows as $\|\eta\|_1/\sqrt{n}$, where $\eta$ is the vector of output weights and is independent of the number $m$ of neurons. Three important questions will be tackled in the following sections:

- Universality: Can we approximate any prediction function with a sufficiently large number of neurons?
- Bound on approximation error: What is the associated approximation error so that we can derive generalization bounds? How can we use the control of the $\ell_1$-norm $\|\eta\|_1$, particularly when the number of neurons $m$ is allowed to tend to infinity?
- Finite number of neurons: What is the number of neurons required to reach such a behavior?

For this, we need to understand the space of functions that neural networks span and how they relate to the smoothness properties of the function (like we did for kernel methods in Chapter 7).

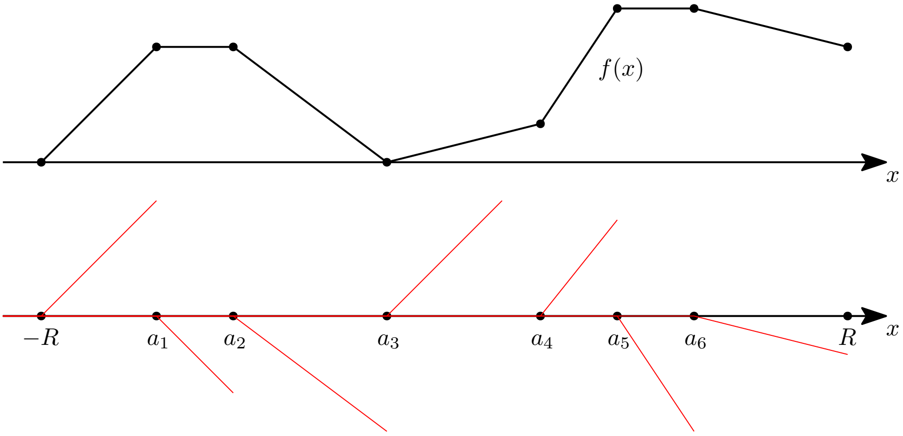
In this section, like in the previous section, we focus primarily on the ReLU activation function, noting that universal approximation results exist as soon as $\sigma$ is not a polynomial (Leshno et al., 1993). We start with a simple non-quantitative argument to show universality in one dimension (and then in all dimensions) before formalizing the function space obtained by letting the number of neurons go to infinity.

### 9.3.1 Universal approximation property in one dimension

We start with simple non-quantitative arguments.

**Approximation of piecewise affine functions.** Since each individual function $x \mapsto \eta_j(w_j x + b_j)_+$ is piecewise affine, the output of a neural network has to be piecewise affine. It turns out that all piecewise affine functions with $m-2$ kinks in the open interval $(-R, R)$ can be represented by $m$ neurons on $[-R, R]$.

Indeed, as illustrated below with $m = 8$, if we assume that the function $f$ is such that $f(-R) = 0$, with kinks $a_1 < \cdots < a_{m-2}$ on $(-R, R)$, we can approximate it on $[-R, a_1]$ by the function $v_1(x+R)_+$ where $v_1$ is the slope of $f$ on $[-R, a_1]$. The approximation is tight on $[-R, a_1]$. To have a tight approximation on $[a_1, a_2]$ without perturbing the approximation on $[-R, a_1]$, we can add to the approximation $v_2(x - a_1)_+$ where $v_2$ is exactly what is needed to compensate the change in slope of $f$. By pursuing the reasoning, we can represent the function on $[-R, R]$ exactly with $m-1$ neurons.



To remove the constraint that $f(-R) = 0$, we can simply notice that $\frac{1}{2R}(x+R)_+ + \frac{1}{2R}(-x+R)_+$ is equal to 1 on $[-R, R]$. Thus, with one additional neuron (only one since $(x+R)_+$ has already been used), we can represent any piecewise-affine function with $m-2$ kinks with $m$ neurons.

**Universal approximation properties.** Now that we can represent precisely all piecewise affine functions on $[-R, R]$, we can use classical approximation theorems for functions on $[-R, R]$. They come in different flavors depending on the norm we use to characterize the approximation. For example, continuous functions can be approximated by piecewise affine functions with arbitrary precision in $L_\infty$-norm (defined as the maximal value of $|f(x)|$ for $x \in [-R, R]$) by simply taking the piecewise interpolant from a grid (see quantitative arguments in Section 9.3.3). With a weaker criterion such as the $L_2$-norm (with respect to the Lebesgue measure), we can approximate any function in $L_2$ (see, e.g., Rudin, 1987). This can be extended to any dimension $d$ by using the Fourier transform representation as $f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^\top x} d\omega$ and approximating the one-dimensional functions sine and cosine as linear superpositions of ReLU's. See a more formal quantitative argument in Section 9.3.4.

To obtain precise bounds in all dimensions, in terms of the number of kinks or the $\ell_1$-norm of output weights, we first need to define the limit when the number of neurons can be unbounded.

### 9.3.2 Infinitely many neurons and variation norm

In this section, we consider neural networks of the form $f(x) = \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j)$, where the input weights are constrained, that is, $(w_j, b_j/R) \in K$, for $K$ a compact subset of $\mathbb{R}^{d+1}$, such as the unit $\ell_2$-sphere (but we will consider a slightly different set at the end of this section). In subsequent sections, we will primarily consider the ReLU activation $\sigma$, but this is not needed in this section (where boundedness or Lipschitz-continuity are sufficient).

In this section, for a function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the $\ell_2$-ball of radius $R$ and center 0 in $\mathbb{R}^d$, we want to study the limit when $m \to \infty$, of the smallest $\ell_1$-norm of $\eta$ for a function $f$ representable with $m$ neurons and output weights $\eta$, that is,

$$\gamma_1^{(m)}(f) = \inf_{\eta_j \in \mathbb{R}, (w_j, b_j) \in K, \ \forall j \in \{1,\dots,m\}} \|\eta\|_1 \ \text{ such that } \ \forall x \in \mathcal{X}, \ f(x) = \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j).$$

The index 1 in $\gamma_1^{(m)}$ will become natural when we compare with kernels in Section 9.5. When $f$ is not representable by $m$ neurons, we let $\gamma_1^{(m)}(f) = +\infty$. By construction the sequence $\gamma_1^{(m)}(f)$ is non-increasing in $m$, and non-negative. Thus, it has a limit when $m$ tends to $\infty$, which we denote $\gamma_1(f)$, which is infinite when $f$ cannot be approximated by a neural network with finitely many neurons and output weights bounded in $\ell_1$-norm. The function $\gamma_1$ is positively homogeneous, that is $\gamma_1(\lambda f) = \lambda \gamma_1(f)$ when $\lambda > 0$ and sub-additive, that is, $\gamma_1(f + g) \leqslant \gamma_1(f) + \gamma_1(g)$. Moreover, one can show that $\gamma_1(f) = 0$ implies that $f = 0$ (proofs left as an exercise). Thus, $\gamma_1$ is a norm on the set of functions $f : \mathcal{X} \to \mathbb{R}$ such that $\gamma_1(f) < +\infty$. It is possible to "complete" this space by adding limits of functions such that their $\gamma_1$-norm remains bounded. We then obtain a Banach space $\mathcal{F}_1$ of functions, with a norm $\gamma_1$, often referred to as the "variation norm" (Kurková and Sanguineti, 2001). This characterizes the set of functions that can be represented by neural networks with bounded $\ell_1$-norm of output weights, regardless of the number of neurons.

**Formulation through measures.** We can write $f(x) = \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j)$, as $f(x) = \int_K (w^\top x + b)_+ d\nu(w,b)$, for $\nu$ the measure $\nu = \sum_{j=1}^m \eta_j \delta_{(w_j,b_j)}$ where $\delta_{(w_j,b_j)}$ is the Dirac measure at $(w_j,b_j)$. Then, the penalty can be written as $\|\eta\|_1 = \int_K |d\nu(w,b)|$, which is the "total variation" of $\nu$.[3] We can therefore see $\gamma_1(f)$ as the infimum of all $\int_K |d\nu(w,b)|$ such that $\forall x \in \mathcal{X}$, $f(x) = \int_K (w^\top x + b)_+ d\nu(w,b)$, and $\nu$ is supported on a countable set. By a density argument (every measure is the "weak" limit of empirical measures), we can remove the constraint on the support and get that

$$\gamma_1(f) = \inf_{\nu \in \mathcal{M}(K)} \int_K |d\nu(w,b)| \text{ such that } \forall x \in \mathcal{X},\ f(x) = \int_K \sigma(w^\top x + b) d\nu(w,b), \quad (9.3)$$

where $\mathcal{M}(K)$ is the set of measures on $K$ with finite total variation (see Bach, 2017, and references therein for more details and formal definitions). This formulation is typically easier to deal with for sufficiently smooth functions, as the optimal measure $\nu$ is typically not a finite sum of Diracs (see examples below, in particular in one dimension). Note that when $K$ is a compact convex set, then the norm $\gamma_1$ is the same when using $K$ in its definition or replacing it by its boundary (that is, we can choose the unit $\ell_2$-sphere or the unit $\ell_2$-ball).

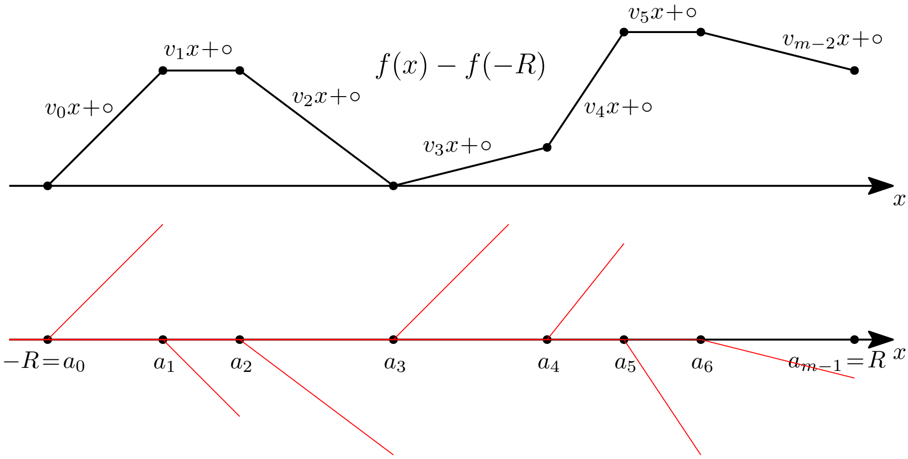
**Studying the approximation properties of $\mathcal{F}_1$.** Now that we have defined the function space through Eq. (9.3), we need to describe the set of functions with finite norm and relate this norm to classical smoothness properties (like done for kernel methods in Chapter 7). To do so, we consider a smaller set than the unit $\ell_2$-ball, that is, the set of $(w, b/R)$ such that $\|w\|_2 = 1/\sqrt{2}$ and $|b| \leqslant R/\sqrt{2}$, which is enough to obtain upper-bounds on the approximation errors. For simplicity, and losing a factor $\sqrt{2}$, we consider the normalization $K = \{(w,b) \in \mathbb{R}^{d+1},\ \|w\|_2 = 1,\ |b| \leqslant R\}$, and consider the norm $\gamma_1$ defined in Eq. (9.3) with this set $K$. Note that for $d = 1$, we have $K = \{(w,b) \in \mathbb{R}^2,\ w \in \{-1,1\},\ |b| \leqslant R\}$. We could stick to the $\ell_2$-sphere, but our particular choice of $K$ leads to simpler formulas.

### 9.3.3 Variation norm in one dimension

The ReLU activation function is specific and leads to simple approximation properties in the interval $[-R,R]$. As already qualitatively described in Section 9.3.1, we start with piecewise affine functions, which, given the shape of the ReLU activation, should be easy to approximate (and immediately lead to universal approximation results as all "reasonable" functions can be approximated by piecewise affine functions). See more details by Breiman (1993); Barron and Klusowski (2018).

[3] When $\nu$ has a density $d\nu/d\tau$ with respect to a base measure $\tau$, then the total variation is defined as the integral $\int_K |d\nu/d\tau(w,b)| d\tau(w,b)$ and is independent of the choice of $\tau$. See https://en.wikipedia.org/wiki/Total_variation for more details.

**Piecewise affine functions.** We can make the reasoning in Section 9.3.1 quantitative. We consider a continuous piecewise affine function on $[-R, R]$ with specific knots at each $R = a_0 < a_1 < \cdots < a_{m-2} < a_{m-1} = R$, so that on $[a_j, a_{j+1}]$, $f$ is affine with slope $v_j$, for $j \in \{0, \dots, m-2\}$.



We can first start to fit the function $x \mapsto f(x) - f(-R)$ (which is equal to 0 at $x = R$) on $[a_0, a_1] = [-R, a_1]$, as $g_0(x) = v_0(x - a_0)_+$. For $x > a_0$, this approximation has slope $v_0$. In order for the approximation to be exact on $[a_1, a_2]$ (while not modifying the function on $[a_0, a_1]$), we consider $g_1(x) = g_0(x) + (v_1 - v_0)(x - a_1)_+$, which is now exact on $[a_0, a_2]$; we can pursue recursively by considering, for $j \in \{1, \dots, m-2\}$

$$g_j(x) = g_{j-1}(x) + (v_j - v_{j-1})(x - a_j)_+,$$

which is equal to $f(x) - f(-R)$ for $x \in [a_0, a_{j+1}]$. We can thus represent $f(x) - f(-R)$ on $[a_0, a_{m-1}] = [0, R]$ exactly with $g_{m-2}(x)$. We have:

$$g_{m-2}(x) = v_0(x - a_0)_+ + \sum_{j=1}^{m-2} (v_j - v_{j-1})(x - a_j)_+.$$

In other words, we can represent any piecewise affine function as (using that on $[-R, R]$, $(x - a_0)_+ = (x + R)_+ = x + R$):

$$f(x) = f(-R) + v_0(x + R) + \sum_{j=1}^{m-2} (v_j - v_{j-1})(x - a_j)_+. \tag{9.4}$$

To obtain a representation that is invariant by a sign change, we also consider the same representation starting from the right (which can, for example, be obtained by applying the one above to $x \mapsto f(-x)$):

$$f(x) = f(R) + v_{m-2}(R - x) + \sum_{j=1}^{m-2} (v_{j-1} - v_j)(a_j - x)_+. \tag{9.5}$$

Note that this also shows that such representations are not unique. By averaging Eq. (9.4) and Eq. (9.5), and using that $\frac{1}{2R}(x+R)_+ + \frac{1}{2R}(-x+R)_+$ is equal to 1 on $[-R,R]$, we get:

$$\begin{aligned} f(x) &= \frac{1}{2}\big[f(R)+f(-R)\big]\Big[\frac{1}{2R}(x+R)_+ + \frac{1}{2R}(-x+R)_+\Big] \\ &\quad +\frac{1}{2}v_0(x+R)_+ - \frac{1}{2}v_{m-2}(-x+R)_+ + \frac{1}{2}\sum_{j=1}^{m-2}(v_j-v_{j-1})\big[(x-a_j)_+ + (a_j-x)_+\big], \end{aligned}$$

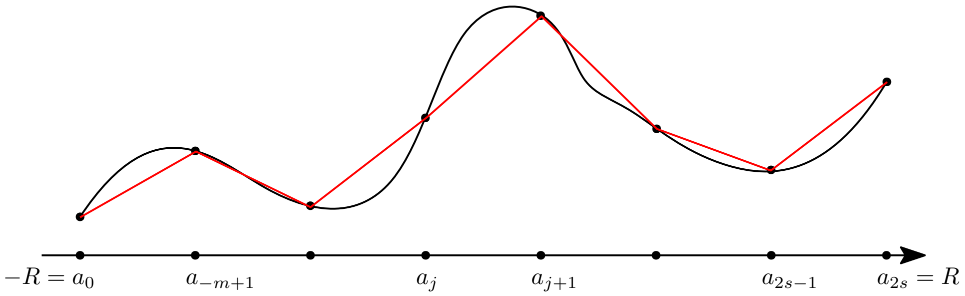
and thus, by construction of the norm $\gamma_1$, we have

$$\gamma_1(f) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+v_0\Big| + \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-v_{m-2}\Big| + \sum_{j=1}^{m-2}|v_j-v_{j-1}|.$$

The norm is thus upper-bounded by the values of $f$ and its derivatives at the boundaries of the interval and the sums of the changes in slope.

**Twice continuously differentiable functions.** We consider a twice continuously differentiable function $f$ on $[-R,R]$, and we would like to express it as a continuous linear combination of functions $x \mapsto (\pm x+b)_+$. We will consider two arguments: one through approximation by piecewise affine functions and one through the Taylor formula with integral remainder.

**Piecewise-affine approximation.** We consider equally-spaced knots $a_j = -R + \frac{j}{s}R$, for $j \in \{0,\dots,2s\}$, and the piecewise affine interpolation from values $a_j, f(a_j)$, with $j \in \{0,\dots,2s\}$, for $s$ that will tend to infinity (see illustration below, where we have $m-1=2s$).



For the approximant $\hat{f}$, the value $v_0$ is equal to $s[f(-R+1/s)-f(-R)] \sim f'(-R)$, and $v_{2s-1} = s[f(R)-f(R-1/s)] \sim f'(R)$ when $s$ tends to infinity, while the differences in slopes $|v_j - v_{j-1}|$ are equal to

$$\Big|\tfrac{s}{R}\big(f(\tfrac{j+1}{s}R)-f(\tfrac{j}{s}R)\big) - \tfrac{s}{R}\big(f(\tfrac{j}{s}R)-f(\tfrac{j-1}{s}R)\big)\Big| = \tfrac{s}{R}\Big|f(\tfrac{j+1}{s}R)-2f(\tfrac{j}{s}R)+f(\tfrac{j-1}{s}R)\Big|,$$

which is equivalent to $\frac{R}{s}\big|f''(\frac{j}{s}R)\big|$ when $s \to +\infty$ (using a second-order Taylor expansion).

Thus, the approximant $\hat{f}$ has $\gamma_1$-norm bounded asymptotically as

$$\gamma_1(\hat{f}) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+f'(-R)\Big|+\frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-f'(R)\Big|+\frac{R}{s}\sum_{j=1}^{2s-1}\big|f''(\tfrac{j}{s}R)\big|.$$

The last term $\frac{R}{s}\sum_{j=1}^{2s-1}\big|f''(\frac{j}{s}R)\big|$ tends to $\int_{-R}^{R}|f''(x)|dx$. Thus, letting $s$ tend to infinity, we get (informally here, as the reasoning below will make it more formal):

$$\gamma_1(f) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+f'(-R)\Big|+\frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-g'(R)\Big|+\int_{-R}^{R}|f''(x)|dx. \tag{9.6}$$

This notably shows that if the number of neurons is allowed to grow, then the $\ell_1$-norm of the weights remain bounded by the quantity above to represent the function $f$ exactly.

**Direct proof through Taylor formula.** Eq. (9.6) above can be extended to continuous functions, which are only twice differentiable almost everywhere with integrable second-order derivatives. In this section, we assume that the function $f$ is twice continuously differentiable and can extend to only integrable second-derivatives by a density argument (see, e.g., Rudin, 1987). For such a function, using the Taylor formula with integral remainder, we have, for $x \in [-R, R]$, using the fact that $(x-b)_+ = 0$ as soon as $b \geqslant x$:

$$\begin{aligned} f(x) &= f(-R) + f'(-R)(x+R) + \int_{-R}^{x} f''(b)(x-b)db \\ &= f(-R) + f'(-R)(x+R) + \int_{-R}^{R} f''(b)(x-b)_+ db. \end{aligned}$$

We also have the symmetric version (obtained by applying the one above to $x \mapsto f(-x)$, replacing $x$ by $-x$, and by a change of variable $b \to -b$ in the integral):

$$f(x) = f(R) - f'(R)(R-x) + \int_{-R}^{R} f''(b)(-x-b)_+ db.$$

By averaging the two equalities, we get:

$$\begin{aligned} f(x) &= \frac{1}{2}\Big[\frac{f(-R)+f(R)}{2R} + f'(-R)\Big](x+R) + \frac{1}{2}\Big[\frac{f(-R)+f(R)}{2R} - f'(-R)\Big](R-x) \\ &\qquad + \frac{1}{2}\int_{-R}^{R} f''(b)(x-b)_+ db - \frac{1}{2}\int_{-R}^{R} f''(b)(-x-b)_+ db. \end{aligned}$$

This leads to the exact same upper-bound on $\gamma_1(f)$ as obtained from piecewise affine interpolation:

$$\gamma_1(f) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+f'(-R)\Big|+\frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-f'(R)\Big|+\int_{-R}^{R}|f''(x)|dx. \tag{9.7}$$

One can check that the upper bound is indeed a norm (left as an exercise). Moreover, the upper bound happens to be tight (see the exercise below).

We will also use a simpler upper-bound, obtained from the triangle inequality:

$$\gamma_1(f) \leqslant \frac{1}{2R}\big|f(-R)+f(R)\big| + \frac{1}{2}\big|f'(R)+f'(-R)\big| + \int_{-R}^{R} |f''(x)|dx. \tag{9.8}$$

**Exercise 9.3** (♦) *Show that the upper-bound in Eq. (9.7) is in fact an equality.*

**Exercise 9.4** (♦) *Show that the minimum norm interpolant from ordered observations $-R < x_1 < \cdots < x_n < R$, $y_1, \dots, y_n \in \mathbb{R}$, is equal to the piecewise-affine interpolant on $[x_1, x_n]$.*

### 9.3.4 Variation norm in arbitrary dimension

If we assume that $f$ is continuous on the ball of center zero and radius $R$, then the Fourier transform $\hat{f}(\omega) = \int_{\mathbb{R}^d} f(x)e^{-i\omega^\top x}dx$ is defined everywhere, and we can write $f$ as the inverse Fourier transform of $\hat{f}$, that is,

$$f(x) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} \hat{f}(\omega)e^{i\omega^\top x}d\omega. \tag{9.9}$$

To compute an upper-bound on $\gamma_1(f)$, it suffices to upper-bound for each $\omega \in \mathbb{R}^d$, $\gamma_1(x \mapsto e^{i\omega^\top x})$ (using complex-valued functions, for which the developments of the previous section still apply, or using sines and cosines), which is possible because we have the representation from Section 9.3.3 and Eq. (9.8) applied to $g : u \mapsto e^{iu\|\omega\|_2}$, for $u \in [-R, R]$,

$$e^{iu\|\omega\|_2} = \int_{-R}^{R} \eta_+(b, \|\omega\|_2)(u-b)_+db + \int_{-R}^{R} \eta_-(b, \|\omega\|_2)(-u-b)_+db,$$

with $\int_{-R}^{R} |\eta_+(b, \|\omega\|_2)|db + \int_{-R}^{R} |\eta_-(b, \|\omega\|_2)|db \leqslant \frac{1}{R} + \|\omega\|_2 + 2R\|\omega\|_2^2 \leqslant \frac{2}{R}(1+2R^2\|\omega\|_2^2)$. We can, therefore, decompose the function defined on the ball of center 0 and radius $R$:

$$\begin{aligned} e^{i\omega^\top x} &= e^{i(x^\top\omega/\|\omega\|_2)\|\omega\|_2} \\ &= \int_{-R}^{R} \eta_+(b, \|\omega\|_2)(x^\top(\omega/\|\omega\|_2)-b)_+db + \int_{-R}^{R} \eta_-(b, \|\omega\|_2)(x^\top(-\omega/\|\omega\|_2)-b)_+db, \end{aligned}$$

with weights being in the correct constraint set (unit norm for the slopes $\omega/\|\omega\|_2$ and constant terms $|b| \leqslant R$), leading to

$$\gamma_1(x \mapsto e^{i\omega^\top x}) \leqslant \frac{2}{R}(1+2R^2\|\omega\|_2^2).$$

Thus, we obtain, from Eq. (9.9) and the triangular inequality for the norm $\gamma_1$:

$$\gamma_1(f) \leqslant \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} |\hat{f}(\omega)|\gamma_1(x \mapsto e^{i\omega^\top x})d\omega \leqslant \frac{1}{(2\pi)^d}\frac{2}{R}\int_{\mathbb{R}^d} |\hat{f}(\omega)|(1+2R^2\|\omega\|_2^2)d\omega. \tag{9.10}$$

Given a function $g : \mathbb{R}^d \to \mathbb{R}$, $\int_{\mathbb{R}^d} |\hat{g}(\omega)| d\omega$ is a measure of smoothness of $g$, and so $\gamma_1(f)$ being finite imposes that $f$ and all second-order derivatives of $f$ have this form of smoothness. The right-hand side of Eq. (9.10) is often referred to as the "Barron norm", named after Barron (1993, 1994). See Klusowski and Barron (2018) for more details.

To relate the norm $\gamma_1$ to other function spaces such as Sobolev spaces, we will consider further upper bounds (and relate them to another norm $\gamma_2$ in Section 9.5).

### 9.3.5 Precise approximation properties

**Precise rates of approximation.** In this section, we will relate the space $\mathcal{F}_1$ to Sobolev spaces, bounding, using Cauchy-Schwarz inequality, the norm $\gamma_1$ as:

$$\begin{aligned}\gamma_1(f) &\leqslant \frac{1}{(2\pi)^d}\frac{2}{R}\int_{\mathbb{R}^d} |\hat{f}(\omega)|(1+2R^2\|\omega\|_2^2)d\omega \quad \text{from Eq. (9.10)},\\ &= \frac{1}{(2\pi)^d}\frac{2}{R}\int_{\mathbb{R}^d} |\hat{f}(\omega)|(1+2R^2\|\omega\|_2^2)^{d/4+5/4}\frac{d\omega}{(1+2R^2\|\omega\|_2^2)^{d/4+1/4}}\\ &\leqslant \frac{1}{(2\pi)^d}\frac{2}{R}\sqrt{\int_{\mathbb{R}^d} |\hat{f}(\omega)|^2(1+2R^2\|\omega\|_2^2)^{d/2+5/2}d\omega}\sqrt{\int_{\mathbb{R}^d}\frac{d\omega}{(1+2R^2\|\omega\|_2^2)^{d/2+1/2}}}, \quad (9.11)\end{aligned}$$

which is a constant times $\sqrt{\int_{\mathbb{R}^d} |\hat{f}(\omega)|^2(1+2R^2\|\omega\|_2^2)^s d\omega}$, which is exactly the Sobolev norm from Chapter 7, with $s = \frac{d}{2} + \frac{5}{2}$ derivatives, which is a reproducing kernel Hilbert space (RKHS) since $s > d/2$.

Thus, all approximation properties from Chapter 7 apply (see there for precise rates, and their application to generalization bounds in Section 9.4). Note, however, that, *using this reasoning*, if we start from a Lipschitz-continuous function, then to approximate it up to $L_2(\mathbb{R}^d)$-norm $\varepsilon$ requires a $\gamma_1$-norm growing as $\varepsilon^{-(s-1)} \geqslant \varepsilon^{-(d/2+3/2)}$ (as obtained at the end of Section 7.5.2 of Chapter 7). Thus, in the generic situation where no particular directions are preferred, using $\mathcal{F}_1$ (neural networks) is not really more advantageous than using kernel methods (see also more details in Section 9.4 and Section 9.5). This changes drastically when such linear structures are present, as shown below.

**Adaptivity to linear latent variables.** We consider a target function $f^*$ that depends only on a $r$-dimensional projection of the data, that is, $f^*$ is of the form $f^*(x) = g(V^\top x)$, where $V \in \mathbb{R}^{d\times r}$ is full rank and has all singular values less than 1, and $g : \mathbb{R}^r \to \mathbb{R}$. Without loss of generality, we can assume that $V$ has orthonormal columns. Then if $\gamma_1(g)$ is finite (for the function $g$ defined on $\mathbb{R}^r$), it can be written as

$$g(z) = \int_{\mathbb{R}^{r+1}} (w^\top z + b)_+ d\mu(w,b),$$

with $\mu$ supported on $\{(w,b) \in \mathbb{R}^{r+1},\ \|w\|_2 \leqslant 1, |b| \leqslant R\}$, and $\gamma_1(g) = \int_{\mathbb{R}^{r+1}} |d\mu(w,b)|$. We can then use this representation of $g$ to obtain a representation of $f^*$ as:

$$f^*(x) = g(V^\top x) = \int_{\mathbb{R}^{r+1}} ((Vw)^\top x + b)_+ d\mu(w,b).$$

Since $\|Vw\|_2 \leqslant 1$ as soon as $\|w\|_2 \leqslant 1$, and $V$ have orthonormal columns, the measure $\mu$ on $(w, b)$ defines a measure for $(Vw, b)$ on $\{(w', b) \in \mathbb{R}^{d+1},\ \|w'\|_2 \leqslant 1, |b| \leqslant R\}$ with a total variation which is less than the one of $\mu$. Thus $\gamma_1(f^*) \leqslant \int_{\mathbb{R}^{r+1}} |d\mu(w,b)| = \gamma_1(g)$. In other words, the approximation properties of $g$ translate to $f^*$, and thus, we pay only the price of these $r$ dimensions and not all $d$ variables, *without* the need to know $V$ in advance. For example, (a) if $g$ has more than $r/2 + 5/2$ squared integrable derivatives, then $\gamma_1(g)$ and thus $\gamma_1(f^*)$ is finite, or (b) if $g$ is Lipschitz-continuous, then both $g$ and $f$ can be approached in $L_2(\mathbb{R}^d)$ with error $\varepsilon$ with a function with $\gamma_1$-norm of order $\varepsilon^{-(r/2+5/2)}$, thus escaping the curse of dimensionality. See Bach (2017) for more details and precise learning rates in Section 9.4.

⚠ Kernel methods do not have such adaptivity. In other words, as shown in Section 9.5, using the $\ell_2$-norm instead of the $\ell_1$-norm on the output weights leads to worse performance.

We will combine these approximation results with the estimation error results in Section 9.4.

### 9.3.6 From the variation norm to a finite number of neurons (♦)

Given a measure $\mu$ on $\mathbb{R}^d$, and a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $\gamma_1(g)$ is finite, we would like to find a set of $m$ neurons $(w_j, b_j) \in K \subset \mathbb{R}^{d+1}$ (which is the compact support of all measures that we consider), such that the associated function defined through

$$f(x) = \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j)$$

is close to $g$ for the norm $L_2(\mu)$.

Since input weights are fixed in $K$, the bound on $\gamma_1(g)$ should translate to a bound on the $\ell_1$-norm of $\eta$: $\|\eta\|_1 \leqslant \gamma_1(g)$. The set of functions $f$ such that $\gamma_1(f) \leqslant \gamma_1(g)$ is the convex hull of functions $s\gamma_1(g)\sigma(w^\top x + b)$, for $s \in \{-1, 1\}$, and $\|w\|_2 = 1$, $|b| \leqslant R$. Thus, we are faced with the problem of approximating elements of a convex hull as an explicit linear combination of extreme points, if possible, with as few extreme points as possible.

In finite dimension, Carathéodory's theorem says that the number of such extreme points can be taken equal to the dimension to get an exact representation. In our case of infinite dimensions, we need an approximate version of Carathéodory's theorem. It turns out that we can create a "fake" optimization problem of minimizing the squared $L_2$-norm (for the input data distribution $p$) $\|f - g\|^2_{L_2(p)}$ such that $\gamma_1(f) \leqslant \gamma_1(g)$, whose solution is $f = g$, with an algorithm that constructs an approximate solution from extreme points. This will be achieved by the Frank-Wolfe algorithm (a.k.a. conditional gradient algorithm). This algorithm is applicable more generally; for more details, see Jaggi (2013); Bach (2015).

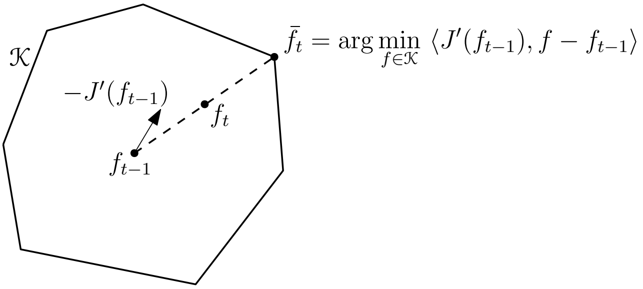
**Frank-Wolfe algorithm.** We thus make a detour by considering an algorithm defined in a Hilbert space $\mathcal{H}$, such that $\mathcal{K}$ is a bounded convex set and $J$ a convex, smooth function from $\mathcal{H}$ to $\mathbb{R}$, that is such that there exists a gradient function $J' : \mathcal{H} \to \mathcal{H}$ such that for all elements $f, g$ of $\mathcal{H}$ (which is the traditional smoothness condition from Section 5.2.3):

$$J(g) + \langle J'(g), f - g\rangle_{\mathcal{H}} \leqslant J(f) \leqslant J(g) + \langle J'(g), f - g\rangle_{\mathcal{H}} + \frac{L}{2}\|f - g\|^2_{\mathcal{H}}.$$

The goal is to minimize $J$ on the bounded convex set $\mathcal{K}$, with an algorithm that only requires access to the set $\mathcal{K}$ through a "linear minimization" oracle (i.e., through minimizing linear functions), as opposed to the projection oracle that we required in Section 5.2.5.

We consider the following recursive algorithm, started from a vector $f_0 \in \mathcal{K}$:

$$\begin{aligned} \bar{f}_t &\in \arg\min_{f\in\mathcal{K}} \ \langle J'(f_{t-1}), f - f_{t-1}\rangle_{\mathcal{H}}, \\ f_t &= \frac{t-1}{t+1} f_{t-1} + \frac{2}{t+1}\bar{f}_t = f_{t-1} + \frac{2}{t+1}(\bar{f}_t - f_{t-1}). \end{aligned} \tag{9.12}$$


$\mathcal{K}$
$\bar{f}_t = \arg\min_{f\in\mathcal{K}} \ \langle J'(f_{t-1}), f - f_{t-1}\rangle$
$-J'(f_{t-1})$
$f_t$
$f_{t-1}$

Because $\bar{f}_t$ is obtained by minimizing a linear function on a bounded convex set, we can restrict the minimizer $\bar{f}_t$ to be extreme points of $\mathcal{K}$, so that, $f_t$ is the convex combination of $t$ such extreme points $\bar{f}_1, \dots, \bar{f}_t$ (note that the first point $f_0$ disappears from the convex combination). We now show that

$$J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \frac{2L}{t+1}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2.$$

**Proof of convergence rate (♦).** This is obtained by using smoothness:

$$\begin{aligned} J(f_t) &\leqslant J(f_{t-1}) + \langle J'(f_{t-1}), f_t - f_{t-1}\rangle_{\mathcal{H}} + \frac{L}{2}\|f_t - f_{t-1}\|^2_{\mathcal{H}} \\ &= J(f_{t-1}) + \frac{2}{t+1}\langle J'(f_{t-1}), \bar{f}_t - f_{t-1}\rangle_{\mathcal{H}} + \frac{4}{(t+1)^2}\frac{L}{2}\|\bar{f}_t - f_{t-1}\|^2_{\mathcal{H}} \\ &\leqslant J(f_{t-1}) + \frac{2}{t+1}\min_{f\in\mathcal{K}} \ \langle J'(f_{t-1}), f - f_{t-1}\rangle_{\mathcal{H}} + \frac{4}{(t+1)^2}\frac{L}{2}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2. \end{aligned}$$

By convexity of $J$, we have for all $f \in \mathcal{K}$, $J(f) \geqslant J(f_{t-1}) + \langle J'(f_{t-1}), f - f_{t-1}\rangle_{\mathcal{H}}$, leading to $\inf_{f\in\mathcal{K}} J(f) \geqslant J(f_{t-1}) + \inf_{f\in\mathcal{K}}\langle J'(f_{t-1}), f - f_{t-1}\rangle_{\mathcal{H}}$. Thus, we get

$$J(f_t) - \inf_{f\in\mathcal{K}} J(f) \quad\leqslant\quad \big[J(f_{t-1}) - \inf_{f\in\mathcal{K}} J(f)\big]\frac{t-1}{t+1} + \frac{4}{(t+1)^2}\frac{L}{2}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2$$

leading to

$$\begin{aligned} t(t+1)\big[J(f_t) - \inf_{f\in\mathcal{K}} J(f)\big] \quad&\leqslant\quad (t-1)t\big[J(f_{t-1}) - \inf_{f\in\mathcal{K}} J(f)\big] + 2L\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2 \\ &\leqslant\quad 2Lt\,\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2 \text{ by using a telescoping sum,}\end{aligned}$$

and thus $J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \frac{2L}{t+1}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2$, as claimed earlier.

**Exercise 9.5** *Show that if we replace Eq. (9.12) by* $f_t = \frac{t-1}{t}f_{t-1} + \frac{1}{t}\bar{f}_t$, $f_t$ *is the uniform convex combination of* $\bar{f}_1, \dots, \bar{f}_t$, *and that,* $J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \frac{2L\log(t+1)}{t+1}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2$.

**Application to approximate representations with a finite number of neurons.** We can apply this to $\mathcal{H} = L_2(\mathbb{R}^d)$ and $J(f) = \|f - g\|^2_{L_2(p)}$, leading to $L = 2$, with $\mathcal{K} = \{f \in L_2(\mathbb{R}^d),\ \gamma_1(f) \leqslant \gamma_1(g)\}$ for which the set of extreme points are exactly single neurons $s\sigma(w^\top \cdot + b)$ scaled by $\gamma_1(g)$ and with an extra sign $s \in \{-1, 1\}$.

We thus obtain after $t$ steps a representation of $f$ with $t$ neurons for which

$$\|f - g\|^2_{L_2(p)} \leqslant \frac{4\gamma_1(g)^2}{t+1}\sup_{(w,b)\in\mathcal{K}}\|\sigma(w^\top\cdot + b)\|^2_{L_2(p)}.$$

Thus, it is sufficient to have $t$ of order $O(\gamma_1(g)^2/\varepsilon^2)$ to achieve $\|f-g\|_{L_2(\mu)} \leqslant \varepsilon$. Therefore, the norm $\gamma_1(g)$ directly controls the approximability of the function $g$ by a finite number of neurons and tells us how many neurons should be used for a given target function. For the ReLU activation, the bound above becomes: $\|f - g\|^2_{L_2(p)} \leqslant \frac{16R^2\gamma_1(g)^2}{t+1}$; note that the dependence of the number of neurons in $\varepsilon$ as $\varepsilon^{-2}$ is not optimal, as it can be improved to $\varepsilon^{-2d/(d+3)}$ (see Bach, 2017, and references therein).

**Application to neural network fitting.** The Frank-Wolfe algorithm can be used to fit a neural network from data by minimizing the empirical risk of a function $f$, which is constrained to have a norm $\gamma_1$ bounded by a fixed constant $D$. After $t$ iterations, the general convergence result above leads to an approximate minimizer with an explicit provable convergence guarantee in $O(1/t)$.

However, as above, the corresponding set of extreme points are single neurons of the form $s\sigma(w^\top \cdot + b)$ scaled by $D$ and with an extra sign $s \in \{-1, 1\}$. Therefore, to implement the linear minimization oracle, given the derivative $\alpha_i$ of the loss function associated with the $i$-th observation, for $i = 1, \dots, n$, we need to minimize with respect to $s, w, b$ the quantity $\sum_{i=1}^n s\alpha_i\sigma(w^\top x_i + b)$, for input observations $x_i \in \mathbb{R}^d$, $i = 1, \dots, n$, for which there is no known polynomial-time algorithms. Thus, we do not obtain through the Frank-Wolfe algorithm a polynomial-time algorithm (see more details by Bach, 2017).

This incremental approach to estimating a neural network is related to boosting procedures that we present in Section 10.3.

## 9.4 Generalization performance for neural networks

We can now consider putting both estimation and approximation errors together, using tools from Section 7.5.1, that give a rate for constrained optimization (this is done for simplicity, as using tools from Section 4.5.5, we could get similar results for penalized problems).

We thus minimized the empirical risk for a $G$-Lipschitz-continuous loss subject to $\gamma_1(f) \leqslant D$. From Section 9.2.3, we get an estimation error less than $\frac{16GDR}{\sqrt{n}}$, on which we need to add $G \inf_{\gamma_1(f) \leqslant D} \|f - f^*\|_{L_2(p)}$, where $f^*$ is the target function, minimizer of the expected risk. Following the same reasoning as in Section 7.5.1, optimizing over $D$ leads to an upper-bound of the form (where the constant is 256 rather than 16 in Eq. (7.8) because the extra factor of 4 in the estimation error):

$$\varepsilon_n = 2G\sqrt{\inf_{f \in \mathcal{F}_1} \Big\{ \|f - f^*\|^2_{L_2(p)} + \frac{256R^2}{n}\gamma_1(f)^2 \Big\}}. \tag{9.13}$$

As shown in Section 7.5, given this bound, we can recover the bound $D$ as $\frac{\sqrt{n}}{16RG}\varepsilon_n$, and thus, using Section 9.3.6 which shows how to approximate a function in $\mathcal{F}_1$ by finitely many neurons, we will lose an additional factor $\varepsilon_n$ with a number of neurons greater than $m \geqslant \frac{16D^2R^2G^2}{\varepsilon_n^2}$ which is exactly equal to a constant times $n$, that is, with this analysis, there is no need to have a number of neurons that greatly exceeds the number of observations.

We can now look at a series of structural assumptions on the target function $f^*$, for which we will see that neural networks provide adaptivity if the regularization parameter is well-chosen:

- No assumption: If we assume that $f_*$ is Lipschitz-continuous on the ball of center 0 and radius $R$, then, as shown at the end of Section 7.5.2, $f_*$ can be extended to a function in the Sobolev space of order 1. Using the comparison of $\gamma_1$ with the Sobolev norm of order $s = \frac{d}{2} + \frac{5}{2}$ in Eq. (9.11), we can reuse the results from kernel methods in Section 7.5.2, and obtain a rate of $O(1/n^{1/(2s)}) = 1/n^{1/(d+5)}$, which exhibits the curse of dimensionality, which cannot anyway be much improved, as the optimal performance has to be larger than $1/n^{1/(d+2)}$ (see Chapter 15).
- Linear latent-variable: If we now assume that $f_*$ depends on an $r$-dimensional *unknown* subspace, then we can reuse the same reasoning on the projected subspace, compare the norm $\gamma_1$ projected to the subspace (like done in Section 9.3.5) to the Sobolev norm on the same projected subspace, thus of order $s = r/2 + 5/2$ (instead of $d/2 + 5/2$). This leads to an estimation rate for the excess risk proportional

$1/n^{1/(r+5)}$ (with constants independent from $d$). This is where neural networks have a strong advantage over kernel methods and sparse methods: they can perform variable selection with non-linear predictions.

- "Teacher network": if we assume that $f^*$ is the linear combination of $k$ hidden neurons, then we obtain a convergence rate proportional to $k/\sqrt{n}$, as the norm $\gamma_1(f^*)$ is proportional to $k$.

**Exercise 9.6** *Consider target functions of the form $f^*(x) = \sum_{j=1}^k f_j(w_j^\top x)$ for one-dimensional Lipschitz-continuous functions. Provide an upper bound on excess risk proportional to $k/n^{1/6}$.*

Note that these rates are not as good as Bach (2017), since the exponent $s = \frac{d}{2} + \frac{5}{2}$ is not optimal, and in fact, a more careful analysis, as outlined in Section 9.5 would lead $s = \frac{d}{2} + \frac{3}{2}$, with a similar dependence on dimension.

**Non-linear variable selection (♦).** In this chapter, we focus primarily on $\ell_2$-norm constraints or penalty on the weight vectors $w_1, \dots, w_m \in \mathbb{R}^d$ of a neural network, but all developments can be carried out with the $\ell_1$-norm, leading to the high-dimensional behavior detailed in Section 8.3.3, but this time selecting variable with a *non-linear* prediction on top of them. For the rest of this section, we assume that $\|x\|_\infty \leqslant R$ almost surely.

The analysis has to be adapted for both the estimation error and the approximation error. For the estimation error, in the derivations of Section 9.2.3, we simply need to replace Eq. (9.2) by

$$\begin{aligned} \mathrm{R}_n(\mathcal{G}) &\leqslant 2GD\bigg(\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_\infty^2 + R^2\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big]\bigg)^{1/2} \qquad (9.14)\\ &\leqslant 2GD\bigg(\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_\infty^2\Big]\bigg)^{1/2} + 2GDR\bigg(\mathbb{E}\Big[\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big]\bigg)^{1/2}\\ &\leqslant 2GDR\frac{\sqrt{2\log(2d)}}{\sqrt{n}} + \frac{2GDR}{\sqrt{n}} \leqslant 4GRD\sqrt{\frac{\log(4d)}{n}}, \end{aligned}$$

using expectations of maxima from Section 1.2.4.

Thus, in estimation rates, we need to consider instead of Eq. (9.13)

$$\varepsilon_n = 2G\sqrt{\inf_{f\in\mathcal{F}_1}\Big\{\|f-f^*\|_{L_2(p)}^2 + \frac{256R^2\log(4d)}{n}\gamma_1(f)^2\Big\}}$$

(note the extra factor $\log(4d)$ and the definition of $R$ as an $\ell_\infty$-bound). Regarding approximation error, we simply use the bound $\|w\|_1 \leqslant \sqrt{k}\|w\|_2$ if $w$ has only $k$ non-zero elements. Thus, if the target function $f^*$ is a Lipschitz-continuous of only $k$ (unknown) variables, we can use the approximation result for $\ell_2$-norm constraints, with an extra dependence on $k$ (which we already had). Thus, overall, the estimation rate of the excess risk is proportional to a constant depending on $k$, times $\big(\frac{\log(4d)}{n}\big)^{1/(k+3)}$, thus with a high-dimensional estimation rate, where $d$ only appears logarithmically.

## 9.5 Relationship with kernel methods (♦)

In this section, we relate our function space $\mathcal{F}_1$ to a simpler function space $\mathcal{F}_2$ that will, in the overparameterized regime when $m$ tends to $+\infty$, correspond to only optimizing the output layer.

### 9.5.1 From a Banach space $\mathcal{F}_1$ to a Hilbert space $\mathcal{F}_2$ (♦)

Following the notations of Section 9.3.2, given a fixed probability measure $\tau$ on $K \subset \mathbb{R}^{d+1}$, we can define another norm as

$$\gamma_2^2(f) = \inf_{\nu \in \mathcal{M}(K)} \int_K \Big| \frac{d\nu(w,b)}{d\tau(w,b)} \Big|^2 d\tau(w,b) \text{ such that } \forall x \in \mathcal{X},\ f(x) = \int_K \sigma(w^\top x + b) d\nu(w,b). \tag{9.15}$$

By construction (and by Jensen's inequality), $\gamma_1(f) \leqslant \gamma_2(f)$, so the space $\mathcal{F}_2$ of functions $f$ such that $\gamma_2(f) < +\infty$ is included in $\mathcal{F}_1$ (moreover, $\gamma_2$ depends on the choice of the base measure $\tau$, while $\gamma_1$ does not).

Moreover, as shown in the proposition below, the space $\mathcal{F}_2$ is a reproducing kernel Hilbert space on $\mathcal{X} = \{x \in \mathbb{R}^d, \|x\|_2 \leqslant R\}$, as defined in Chapter 7.

**Proposition 9.2** *The space $\mathcal{F}_2$ is the reproducing kernel Hilbert space associated with the positive definite kernel function*

$$k(x,x') = \int_K \sigma(w^\top x + b)\sigma(w^\top x' + b) d\tau(w,b). \tag{9.16}$$

**Proof** For a formal proof for all compact sets $K$, see Bach (2017, Appendix A). We only provide a proof for finite $K$ and $\tau$ the uniform probability measure on $K$, we then have, $\gamma_2^2(f) = \inf_{\nu \in \mathbb{R}^K} \frac{1}{|K|} \sum_{(w,b)\in K} \nu_k^2$ such that $f(x) = \frac{1}{|K|}\sum_{(w,b)\in K} \nu_k \sigma(w^\top x + b)$, which corresponds to penalizing the $\ell_2$-norm of $\theta = \frac{1}{\sqrt{|K|}}\nu \in \mathbb{R}^K$ for $f(x) = \theta^\top \varphi(x)$, and $\varphi(x)_{(w,b)} = \frac{1}{|K|^{1/2}}\sigma(w^\top x + b)$. We thus exactly get the desired kernel $k(x,x') = \frac{1}{|K|}\sum_{(w,b)\in K} \sigma(w^\top x + b)\sigma(w^\top x' + b)$. ■

**Interpretation in terms of random features.** As already mentioned in Section 7.4, the kernel defined in Eq. (9.16) can be approximated by sampling uniformly at random from $\tau$, $m$ points $(w_j, b_j)$, $j = 1, \dots, m$, and approximating $k(x,x')$ by

$$\hat{k}(x,x') = \frac{1}{m}\sum_{j=1}^m \sigma(w_j^\top x + b_j)\sigma(w_j^\top x' + b_j).$$

This corresponds to using $f(x) = \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j)$, with a penalty proportional to $m\|\eta\|_2^2$. Thus random features correspond to only optimizing with respect to the output weights while keeping the input weights fixed (while for $\gamma_1$, we optimize over all weights).

Therefore, infinite width networks where input weights are random and only output weights are learned are, in fact, kernel methods in disguise (Neal, 1995; Rahimi and Recht, 2008).

This kernel can be computed in closed form for simple activations and distributions of weights; see Section 9.5.2 and Cho and Saul (2009); Bach (2017). Thus, the same regularization properties may be achieved with algorithms from Chapter 7 (which are based on convex optimization and therefore come with guarantees). Note that, as shown in Section 7.4, a common strategy for kernels defined as expectations is to use the *random feature* approximation $\hat{k}(x, x')$, that is, here, use the neural network representation explicitly.

⚠ The kernel approximation corresponds to input weights $w_j, b_j$ sampled randomly and *held fixed*. Only the output weights $\eta_j$ are optimized.

⚠ Because Dirac measures are not square integrable, the prediction function $x \mapsto \sigma(w^\top x + b)$, that is, a single neuron, is typically not in the RKHS, which is typically composed of smooth functions. See the examples below.

**Link between the two norms.** To relate the two norms more precisely, we rewrite $\gamma_1$ using the fixed probability measure $\tau$, as

$$\gamma_1(f) = \inf_{\eta: K \to \mathbb{R}} \int_K |\eta(w,b)| d\tau(w,b) \text{ such that } \forall x \in \mathcal{X}, f(x) = \int_K \sigma(w^\top x + b)\eta(w,b) d\tau(w,b).$$

The only difference with the squared RKHS norm above is that we consider the $L_1$-norm instead of the squared $L_2$-norm of $\eta$ (with respect to the probability measure $\tau$). The minimum achievable norm is exactly $\gamma_1(f)$.

Note that typically, the infimum over all $\eta$ is not achieved as the optimal measure in Eq. (9.3) may not have a density with respect to $\tau$. Because we use an $L_1$-norm penalty, the measures $\mu(w,b) = \eta(w,b)\tau(w,b)$ can span in the limit all measures $\mu(w,b)$ with finite total variation $\displaystyle\int_{\mathbb{R}^{d+1}} |d\mu(\eta, b)| = \int_{\mathbb{R}^{d+1}} |\eta(w,b)| d\tau(w,b)$.

Overall, we have the following properties (see Table 9.1 for a summary):

- Because of Jensen's inequality, we have $\gamma_1(f) \leqslant \gamma_2(f)$, and thus $\mathcal{F}_2 \subset \mathcal{F}_1$, that is the space $\mathcal{F}_1$ contains many more functions.
- ⚠ A single neuron is in $\mathcal{F}_1$ with $\gamma_1$-norm less than one, as the mass of a Dirac is equal to one.

| $\mathcal{F}_2$ | $\mathcal{F}_1$ |
|---|---|
| Hilbert space | Banach space |
| $\gamma_2(f)^2 = \inf \int_{\mathbb{R}^{d+1}} \lvert\eta(w,b)\rvert^2 d\tau(w,b)$<br>s. t. $f(x)=\int_{\mathbb{R}^{d+1}} \eta(w,b)\sigma(w^\top x+b)d\tau(w,b)$ | $\gamma_1(f) = \inf \int_{\mathbb{R}^{d+1}} \lvert\eta(w,b)\rvert d\tau(w,b)$<br>s. t. $f(x)=\int_{\mathbb{R}^{d+1}} \eta(w,b)\sigma(w^\top x+b)d\tau(w,b)$ |
| Smooth functions<br>Single neurons $\notin \mathcal{F}_2$ | Potentially non-smooth functions<br>Single neurons $\in \mathcal{F}_1$ |

Table 9.1: Summary of properties of the norms $\gamma_1$ and $\gamma_2$.

### 9.5.2 Kernel function (♦♦)

We can compute in closed form the kernel function, which is only useful computationally if the number of random features $m$ is larger than the number of observations (when using the kernel trick is advantageous, as outlined in Section 7.4).

In one dimension, with $w$ uniform on the unit sphere, that is, $w \in \{-1, 1\}$, and with $b$ uniform on $[-R, R]$, we have the following kernel

$$k(x,x') = \frac{1}{4R}\int_{-R}^{R}\Big((x-b)_+(x'-b)_+ + (-x-b)_+(-x'-b)_+\Big)db.$$

After a short calculation left as an exercise (see also Bach, 2023b), we can compute it in closed form as:

$$k(x,x') = \frac{R^2}{6} + \frac{xx'}{2} + \frac{1}{24R}|x-x'|^3.$$

In higher dimension, we have:

$$k(x,x') = \int_{\|w\|_2=1}\frac{1}{2R}\int_{-R}^{R}(w^\top x+b)_+(w^\top x'+b)_+ db d\tau(w),$$

where $\tau$ is the uniform distribution on the sphere. After a longer calculation, also left as an exercise (see Bach, 2023b), we get:

$$k(x,x') = \frac{R^2}{6} + \frac{1}{2d}x^\top x' + \frac{1}{24R}\frac{\Gamma(2)\Gamma(\frac{d}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{d}{2}+\frac{3}{2})}\|x-x'\|_2^3. \tag{9.17}$$

See Figure 9.2 for comparing the RKHS (corresponding to $m = +\infty$ neurons) and the approximation with finite $m$.

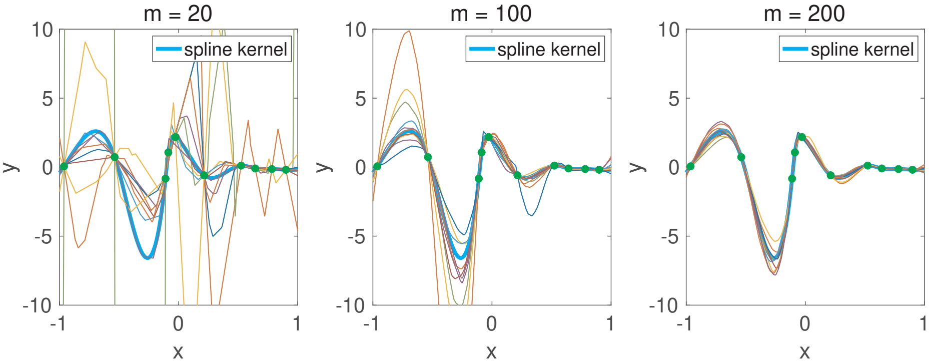


Figure 9.2: Examples of functions in the reproducing kernel Hilbert space $\mathcal{F}_2$ and its approximation based on random features, with $m = 20, 100$, and 200. All functions are the minimum norm interpolators of the green points. This is to be contrasted with the Banach space $\mathcal{F}_1$, where the minimum norm interpolator is the piecewise affine interpolator (see Exercise 9.4), and can be achieved with $m = n$ neurons, where $n$ is the number of observed points.

### 9.5.3 Upper-bound on RKHS norm (♦♦)

We can now find upper bounds on the norm $\gamma_2$. We can either use the kernel function from Eq. (9.17) or the random feature interpretation from Eq. (9.15). We first use the random feature interpretation in one dimension.

**Upper-bound on RKHS norm $\gamma_2$ in one dimension.** Using the same reasoning as the end of Section 9.5, we can get an upper-bound on $\gamma_2(f)$ by decomposing $f$ as

$$f(x) = \int_{-R}^{R} \eta_+(b)(x-b)_+ \frac{db}{4R} + \int_{-R}^{R} \eta_-(b)(-x-b)_+ \frac{db}{4R},$$

with now $\gamma_2(f)^2 \leqslant \displaystyle\int_{-R}^{R} \eta_+(b)^2 \frac{db}{4R} + \int_{-R}^{R} \eta_-(b)^2 \frac{db}{4R}$.

By using as in Section 9.3.3 the Taylor expansion with integral remainder, we get, for any twice differentiable function $f$ on $[-R, R]$:

$$\begin{aligned} f(x) &= \frac{1}{2}f(-R) + \frac{1}{2}f(R) + \frac{1}{2}f'(-R)(x+R) - \frac{1}{2}f'(R)(-x+R) \\ &\qquad + \frac{1}{2}\int_{-R}^{R} f''(b)(x-b)_+ db - \frac{1}{2}\int_{-R}^{R} f''(b)(-x-b)_+ db \\ &= \frac{1}{2}\big[f'(R) + f'(-R)\big] + \frac{1}{2}\big[R(f'(-R) - f'(R)) + f(-R) + f(R)\big] \\ &\qquad + \frac{1}{2}\int_{-R}^{R} f''(b)(x-b)_+ db - \frac{1}{2}\int_{-R}^{R} f''(b)(-x-b)_+ db. \end{aligned}$$

We can now use explicit representations of constants and linear functions, *without* Diracs, as we need finite $L_2$-norms, as:

$$\begin{aligned} x &= \int_{-R}^{R} \frac{(x-b)_+ - (-x-b)_+}{2R} db = \int_{-R}^{R} \frac{x}{2R} db \\ -\frac{R^2}{6} &= \int_{-R}^{R} b(x-b)_+ \frac{db}{4R} + \int_{-R}^{R} b(-x-b)_+ \frac{db}{4R}. \end{aligned}$$

After a short calculation left as an exercise, this leads to

$$\gamma_2(f)^2 \leqslant 2R \int_{-R}^{R} f''(x)^2 dx + \big[f'(R) + f'(-R)\big]^2 + 3\big[R(f'(R) - f'(-R)) - f(-R) - f(R)\big]^2, \tag{9.18}$$

which happens to be an equality (which can be shown by showing that this defines a dot-product, for which $\langle f, k(\cdot, x)\rangle = f(x)$, see Bach, 2023b).

**Exercise 9.7** *Show that the upper bound on $\gamma_2$ from Eq. (9.18) is larger than the bound on $\gamma_1$ from Eq. (9.7).*

The main difference with $\gamma_1$ is that the second-derivative is penalized by an $L_2$-norm and not by an $L_1$-norm, and that this $L_2$-norm can be infinite when the $L_1$-norm is finite, the classic example being for the hidden neuron functions $(x-b)_+$.

⚠ The RKHS is combining infinitely many hidden neuron functions $(x-b)_+$, none of them are inside the RKHS,

⚠ This smoothness penalty does not allow the ReLU to be part of the RKHS. However, this is still a universal penalty (as the set of functions with squared integrable second derivative is dense in $L_2$).

**Upper-bound on RKHS norm $\gamma_2$ in all dimensions.** We can first find a bound directly from the one on $\gamma_1$ in Eq. (9.10), which is exactly Eq. (9.11), ending up with the restriction on the ball of center 0 and radius $R$ of the Sobolev space corresponding to square integrable $s = \frac{d}{2} + \frac{5}{2}$ derivatives on $\mathbb{R}^d$. It turns out this provides a bound on $\gamma_2$ (as can be shown by reproducing the reasoning from Section 9.3.4).

However, this bound is not optimal, which can already be seen in dimension $d = 1$, where we obtain $s = 3$ instead of $s = 2$. It turns out that, in general, it is possible to show $\gamma_2$ is less than a Sobolev norm with index $s = \frac{d}{2} + \frac{3}{2}$. This can be done by drawing links with multivariate splines (Wahba, 1990; Bach, 2023b).

## 9.6 Experiments

We consider the same experimental set-up as Section 7.7, that is, one-dimensional problems to highlight the adaptivity of neural network methods to the regularity of the target function, with smooth targets and non-smooth targets. We consider several values for the

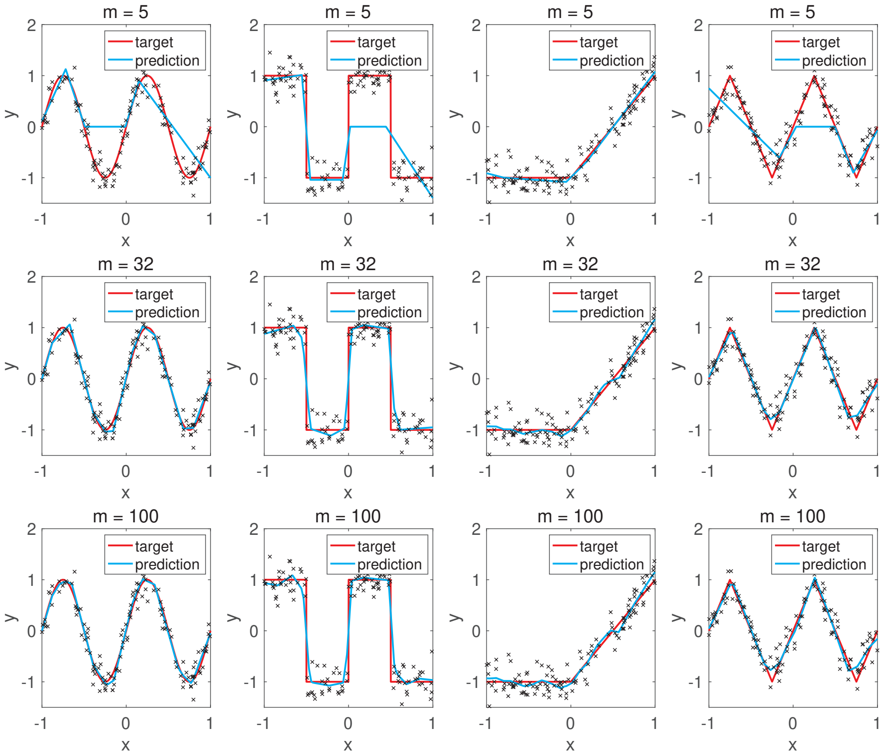


Figure 9.3: Fitting one-dimensional functions with various numbers of neurons $m$ and no additional regularization (top: $m = 5$, middle: $m = 32$, bottom: $m = 100$), with four different prediction problems (one per column).

number $m$ of hidden neurons, and we consider a neural network with ReLU activation functions and an additional global constant term. Training is done by stochastic gradient descent with a small constant step-size and random initialization.

Note that for small $m$, while a neural network with the same number of hidden neurons could fit the data better, optimization is unsuccessful (SGD gets trapped in a bad local minimum). Moreover, between $m = 32$ and $m = 100$, we do not see any overfitting, highlighting the potential under-fitting behavior of neural networks. See also https://francisbach.com/quest-for-adaptivity/.

## 9.7 Extensions

Fully-connected single-hidden layer neural networks are far from what is used in practice, particularly in computer vision and natural language processing. Indeed, state-of-the-art performance is typically achieved with the following extensions:

- **Going deep with multiple layers:** The most simple form of deep neural network is a multilayer fully connected neural network. Ignoring the constant terms for simplicity, it is of the form $f(x^{(0)}) = y^{(L)}$ with input $x^{(0)}$ and output $y^{(L)}$ given by:

$$\begin{aligned} y^{(k)} &= (W^{(k)})^\top x^{(k-1)} \\ x^{(k)} &= \sigma(y^{(k)}), \end{aligned}$$

where $W^{(\ell)}$ is the matrix of weights for layer $\ell$. For these models, obtaining simple and powerful theoretical results is still an active area of research, in terms of approximation, estimation, or optimization errors. See, e.g., Lu et al. (2020); Ma et al. (2020); Yang and Hu (2021). Among these results, the "neural tangent kernel" provides another link between neural networks and kernel methods beyond the one described in Section 9.5 and that applies more generally (see, e.g., Jacot et al., 2018; Chizat et al., 2019).

- **Residual networks:** An alternative to stacking layers one after the other like above is to introduce a different architecture of the form:

  $$\begin{aligned} y^{(k)} &= (W^{(k)})^\top x^{(k-1)} \\ x^{(k)} &= x^{(k-1)} + \sigma(y^{(k)}). \end{aligned}$$

  The direct modeling of $x^{(k)} - x^{(k-1)}$ instead of $x^{(k)}$ through an extra non-linearity, originating from He et al. (2016), can be seen as a discretization of an ordinary differential equation (Chen et al., 2018).

- **Convolutional neural networks:** To tackle large data and improve performances, it is important to leverage prior knowledge about the typical data structure to process. For instance, for signals, images, or videos, it is important to take into account the translation invariance (up to boundary issues) of the domain. This is done by constraining the linear operators involved in the linear part of the neural networks to respect some form of translation invariance and, thus, to use convolutions. See Goodfellow et al. (2016) for details. This can be extended beyond grids to topologies expressed in terms of graphs, leading to graph neural networks (see, e.g., Bronstein et al., 2021).

## 9.8 Conclusion

In this chapter, we have focused primarily on neural networks with one-hidden layers and provided guarantees on the approximation and estimation errors, which show that this class of models, if empirical risk minimization can be performed, leads to a predictive performance that improves on kernel methods from Chapter 7, by being adaptive to linear latent variables (e.g., dependence on an unknown linear projection of the data). In particular, we highlight that having a number of neurons in the order of the number of observations is not detrimental to good generalization performance, as long as the norm of the weights is controlled.

We pursue the study of over-paramaterized models in Chapter 12, where we show how optimization algorithms can both globally converge and leads to implicit biases.

# Part III

# Special topics

# Chapter 10

# Ensemble learning

**Chapter summary**

- Combining several predictors learned on modified versions of the original dataset can have computational and/or statistical benefits.
- Averaging predictors on several reshuffled/resampled/uniformly projected data sets will typically lower the estimator's variance with a potentially limited increase in bias.
- Boosting: iteratively refining the prediction function by re-training on a reweighted dataset in a greedy fashion is an efficient way of building task-dependent features.

Given a supervised learning algorithm $\mathcal{A}$ that goes from datasets $\mathcal{D}$ to prediction rules $\mathcal{A}(\mathcal{D}) : \mathcal{X} \to \mathcal{Y}$, can we run it several times on different datasets constructed from the same original one, and combine the results to get a better overall predictor? The combination is typically a "linear" combination: like for local averaging methods, which combine labels from close-by inputs, we combine the predicted labels from the estimators learned on different datasets. For regression ($\mathcal{Y} = \mathbb{R}$), this is done by simply linearly combining predictions; for classification, this is done by a weighted majority vote or by linearly combining real-valued predictions when convex surrogates are used (such as the logistic loss). For linear models (in their parameters), such linear combinations do not lead to new functions that could not be accessed initially. Still, for non-linear models, this leads to new functions with typically better approximation properties.

The construction of a new dataset given an old one $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, is typically done by giving a different weight $v_i \in \mathbb{R}_+$ to each $(x_i, y_i)$. When the weights are integer-valued, this can be implemented by duplicating the corresponding observations several times (as many times as the integer weight) and then using an existing algorithm for (regularized) empirical risk minimization on the enlarged dataset. In particular, for stochastic gradient descent on the empirical risk, this can be implemented by sampling each observation $(x_i, y_i)$ according to its weight $v_i$ (which then does not need to be an integer). Note, however, that most learning techniques, particularly those based on empirical risk minimization, can directly accommodate arbitrary weights.

In this chapter, we consider two classes of techniques:

- *Bagging / averaging techniques*: datasets are constructed *in parallel*, and the weights are typically random and "uniform" (for example, distributed uniformly or constant). A similar effect can be obtained by modifying the original dataset using random projections. This is studied in Section 10.1 and in Section 10.2.
- *Boosting techniques*: datasets are constructed *sequentially*, and these weights are adapted from previous datasets and thus not uniformly distributed. This is studied in Section 10.3.

The benefits of each combination technique will depend strongly on the original predictor, with three classes that we have considered in earlier chapters:

- Local averaging methods: they will be well-adapted to all ensemble learning techniques, in particular for predictors with high-variance, such as 1-nearest-neighbor estimation.
- Empirical risk minimization with non-linear models: from a set of functions $\varphi(w, \cdot)$, with $w \in \mathcal{W}$, then linear combinations increase the set of models to $\int_{\mathcal{W}} \varphi(w,x)d\nu(w)$, for $\nu$ a signed measure on $\mathcal{W}$. These will be adapted to boosting techniques (we already saw some of them in Chapter 9 in the context of neural networks).
- Empirical risk minimization with linear models (linearity in the model's parameters): the overall model class remains the same by taking linear combinations. Thus, these are typically not adapted to ensemble learning techniques unless some variable/feature selection is added (as we do in Section 10.2).

## 10.1 Averaging / bagging

In this section, for simplicity, we consider the regression case with the square loss where we have an explicit bias/variance decomposition, noting that most results extend beyond that situation. See Exercise 10.1 below.

### 10.1.1 Independent datasets

The idea of bagging, and more generally of averaging methods, is to average predictions from estimators learned from datasets that are as independent as possible. In an idealized situation, we have $m$ independent datasets of size $n$, composed of i.i.d. observations from the same distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$. We obtain for each of them an estimator $\hat{f}_\lambda^{(j)}$, where $j \in \{1, \dots, m\}$, and $\lambda$ is an associated hyperparameter specific to the learning procedure. The new predictor is $\hat{f}_\lambda^{\text{bag}}$ is simply the average of all $\hat{f}_\lambda^{(j)}$, $j = 1, \dots, m$.

If we denote $\text{bias}^{(j)}(x) = \mathbb{E}[\hat{f}_\lambda^{(j)}(x)] - f^*(x)$, and $\text{var}^{(j)}(x) = \text{var}\big[\hat{f}_\lambda^{(j)}(x)\big]$ (assuming $x$

is fixed and only taking expectations with respect to the data), then they are the same for all $j \in \{1, \dots, m\}$, and the bias of $\hat{f}_\lambda^{\rm bag}$ is the same as the base bias for a single dataset (and thus so is the squared bias). At the same time, the variance is divided by $m$ because the datasets are assumed to be independent.

Thus, in the bias/variance trade-off, the selected hyperparameter will typically select a higher variance (or equivalently lower bias) estimator than for $m = 1$. We now give a few examples.

**$k$-nearest neighbor regression.** We consider the analysis from Section 6.3.2 on prediction problems over $\mathcal{X} \subset \mathbb{R}^d$, where we showed in Prop. 6.2 that the (squared) bias was upper-bounded by $8B^2 \text{diam}(\mathcal{X})^2 \big(\frac{2k}{n}\big)^{2/d}$ (for $d \geqslant 2$). At the same time, the variance was bounded by $\frac{\sigma^2}{k}$, where $\sigma^2$ is a bound on the noise variance on top of the target function $f^*$, while $B$ is the Lipschitz-constant of the target function. Thus, with $m$ replications, we get an excess risk upper-bounded by

$$\frac{\sigma^2}{k{\color{red}m}} + 8B^2 \text{diam}(\mathcal{X})^2 \Big(\frac{2k}{n}\Big)^{2/d}.$$

When optimizing the bound above with respect to $k$, we get that $k^{1+2/d} \propto \frac{n^{2/d}}{m}$, leading to $k \propto \frac{1}{m^{d/(2+d)}} n^{2/(2+d)}$. Compared to Section 6.3.2, we obtain a smaller number of neighbors (which is consistent with favoring higher variance estimators). The overall excess risk ends up being proportional to $1/(mn)^{2/(d+2)}$, which is exactly the rate for a dataset of $N = mn$ observations.

Thus, dividing a dataset of $N$ observations in $m$ chunks of $n = N/m$ observations, estimating independently, and combining linearly does not lead to an overall improved statistical behavior compared to learning all at once. Still, it can have significant computational advantages when the $m$ estimators can be computed in parallel (and totally independently). We thus obtain a distributed algorithm with the same worst-case predictive performance as for a single machine.

Note here that there is an upper bound on the number of replications (and thus the ability for parallelization) to get the same (optimal rate), as we need $k$ to be larger than one, and thus, $m$ cannot grow larger than $n^{2/d}$.

**Exercise 10.1** *We consider $k$-nearest neighbor multi-category classification with a majority vote rule. What is the optimal choice of $m$ when using independent datasets?*

**Ridge regression.** Following the analysis from Section 7.6.6, the variance of the ridge regression estimator was proportional to $\frac{\sigma^2}{n} \lambda^{-1/\alpha}$ and the bias proportional to $\lambda^{t/s}$ (see precise definitions in Section 7.6.6). With $m$ replications, we thus get an excess risk proportional to $\frac{\sigma^2}{nm} \lambda^{-1/\alpha} + \lambda^{t/s}$, and the averaged estimator behaves like having $N = nm$ observations. Again, with the proper choice of regularization parameter (lower $\lambda$ than for the full dataset), there is no statistical advantage. Still, there may be a computational one, not only for parallel processing but also with a single machine, as the training time for ridge regression is super-linear in the number of observations (see the exercise below).

**Exercise 10.2** *Assuming that obtaining an estimator for ridge regression has running-time complexity $O(n^\beta)$ for $\beta \geqslant 1$ for $n$ observations, what is the complexity of using a split of the data into $m$ chunks? What is the optimal value of $m$?*

**Beyond independent datasets.** Having independent datasets may not be possible, and one typically needs to artificially "create" such replicated datasets from a single one, which is precisely what bagging methods will do in the next section, with still a reduced variance, but this time a potentially higher bias.

### 10.1.2 Bagging

We consider data sets $\mathcal{D}^{(b)}$, obtained with random weights $v_i^{(b)} \in \mathbb{R}_+$, $i = 1, \dots, n$. For the bootstrap, we consider $n$ samples from the original $n$ data points with replacement, which corresponds to $v_i^{(b)} \in \mathbb{N}$, $i = 1, \dots, n$, that sum to $n$. Such sets of weights are sampled independently $m$ times. We study $m = \infty$ for simplicity, that is, infinitely many replications (in practice, the infinite $m$ behavior can be achieved with moderate $m$'s). Infinitely many bootstrap replications lead to a form of stabilization, which is important for highly variable predictors (which usually imply a large estimation variance).

For linear estimators (in the definition of Section 6.2.1) with the square loss, such as kernel ridge regression or local averaging, this leads to another linear estimator. Therefore, this provides alternative ways of regularizing, which typically may not provide a strong statistical gain over existing methods but provide a computational gain, in particular when each estimator is very efficient to compute. Overall, as shown below for 1-nearest-neighbor, bagging will lower variance while increasing the bias, thus leading to trade-offs that are common in regularizing methods. See also the end of Section 10.2 for a short description of "random forests", which is also (partially) based on bagging.

For simplicity, we will consider averaging estimators obtained by randomly selecting $s$ observations from the $n$ available ones, doing this many times (infinitely many for the analysis), and averaging the predictions.

**Exercise 10.3** *Show that when sampling $n$ elements* with *replacement from $n$ items, the expected fraction of distinct items is $1 - (1 - 1/n)^n$, and that it tends to $1 - 1/e$ when $n$ tends to infinity.*

**One-nearest neighbor regression.** We focus on the 1-nearest neighbor estimator where the strong effect of bagging is striking. The analysis below follows from Biau et al. (2010). The key observation is that if we denote $(x_{(i)}(x), y_{(i)}(x))$ the pair of observations which is the $i$-th nearest neighbor of $x$ from the dataset $x_1, \dots, x_n$ (ignoring ties), then we can write the bagged estimate as

$$\hat{f}(x) = \sum_{i=1}^{n} V_i y_{(i)}(x),$$

where the non-negative weights $V_i$ sum to one, and *do not depend on* $x$. The weight $V_i$ is the probability that the $i$-th nearest neighbor of $x$ is the 1-nearest-neighbor of $x$ in a uniform subsample of size $s$. We consider sampling without replacement and leave sampling with replacement as an exercise (see Biau et al., 2010, for more details). We assume $s \geqslant 2$.

To select the $i$-th nearest neighbor as the 1-nearest-neighbor in a subsample, we need that the $i$-th nearest neighbor is selected but none of the closer neighbors, which leaves $s-1$ elements to choose among $n-i$ possibilities. This shows, that if $i > n-s+1$, then $V_i = 0$, while otherwise $V_i = \binom{n}{s}^{-1}\binom{n-i}{s-1}$, as the total number of subsets of size $s$ is $\binom{n}{s}$ and there are $\binom{n-i}{s-1}$ relevant ones.

We can now use the reasoning from Section 6.3.2. Since for any $x$, the weights given to each observation (once they are ordered in terms of distance to $x$) are $V_1, \dots, V_n$, the variance term is equal to $\sum_{i=1}^{n} V_i^2$. To obtain a bound, we note that for $i \leqslant n-s+1$,

$$V_i = \frac{s}{n-s+1} \frac{\prod_{j=0}^{s-2}(n-i-j)}{\prod_{j=0}^{s-2}(n-j)} = \frac{s}{n-s+1} \prod_{j=0}^{s-2} (1 - \frac{i}{n-j}) \leqslant \frac{s}{n-s+1} \prod_{j=0}^{s-2} (1 - \frac{i}{n}),$$

leading to, upper-bounding the sum by an integral:

$$\begin{aligned} \sum_{i=1}^{n} V_i^2 &\leqslant \frac{s^2}{(n-s+1)^2} \sum_{i=1}^{n} \big(1 - \frac{i}{n}\big)^{2(s-1)} \leqslant \frac{ns^2}{(n-s+1)^2} \int_0^1 (1-t)^{2(s-1)} dt \\ &\leqslant \frac{ns^2}{(n-s+1)^2} \frac{1}{2s-1} \leqslant \frac{ns}{(n-s+1)^2} = \frac{s}{n} \frac{1}{(1+1/n-s/n)^2}. \end{aligned}$$

For the bias term, we need to bound $\sum_{i=1}^{n} V_i \cdot \mathbb{E}\big[\|x - x_{(i)}(x)\|^2\big]$, where the expectation is with respect to the data and the test point $x$. We note here that by definition of $V_i$, and conditioning on the data and $x$, this is the expectation of the distance to the first nearest neighbor from a random sample of size $s$, and thus, by Lemma 6.1, less than $4\mathrm{diam}(\mathcal{X})^2 \frac{1}{s^{2/d}}$ if $d \geqslant 2$ (which we now assume).

Thus, the overall excess risk is less than

$$4B^2 \mathrm{diam}(\mathcal{X})^2 \frac{1}{s^{2/d}} + \frac{s}{n} \frac{1}{(1+1/n-s/n)^2},$$

which we can balance by choosing $s^{1+2/d} \propto n$, leading to the same performance as $k$-nearest neighbor for a well chosen $k$, but now with a bagged esimate.

In Figure 10.1, simulations in one dimension are plotted, showing the regularizing effects of bagging; we see that when $s = n$ (no subsampling), we recover the 1-nearest neighbor estimate, and when $s$ decreases, the variance indeed decreases, while the bias increases.

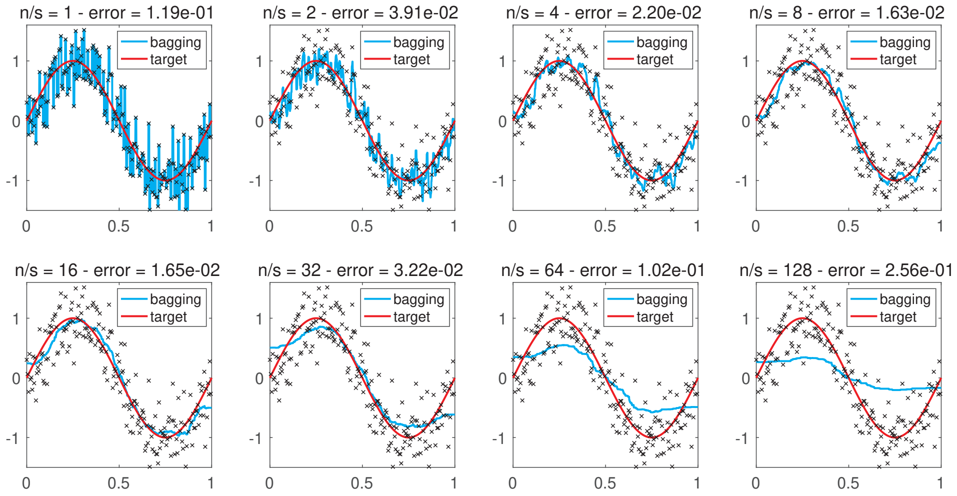


Figure 10.1: Subsampling estimates with $m = 20$ subsampled datasets, for varying subsampling ratios $n/s$, with an estimation of the testing error. When $n/s$ is equal to one, we recover the 1-nearest neighbor classifier (which overfits), and when $n/s$ grows, we get better fits until underfitting kicks in.

## 10.2 Random projections and averaging

In the previous section, we reweighted observations to be able to re-run the original algorithm. This can also be done through random projections of all observations. Such random projections can be performed in several ways: (a) for data in $\mathbb{R}^d$ by selecting $s$ of the $d$ variables, (b) still for data in $\mathbb{R}^d$, by projecting the data in a more general $s$-dimensional subspace, (c) for kernel methods, using random features such as presented in Section 7.4. Such random projections can also reduce the number of samples while keeping the dimension fixed.

In this section, we consider random projections for ordinary least-squares (with the same notation as in Chapter 3, with $y \in \mathbb{R}^n$ the response vector and $\Phi \in \mathbb{R}^{n\times d}$ the design matrix), in two settings:

(a) *Sketching*: replacing $\min_{\theta\in\mathbb{R}^d} \|y-\Phi\theta\|_2^2$ by $\min_{\theta\in\mathbb{R}^d} \|Sy-S\Phi\theta\|_2^2$, where $S \in \mathbb{R}^{s\times n}$ is an i.i.d. Gaussian matrix (with independent zero mean and unit variance elements). This is an idealization of subsampling done in the previous section. Here we typically have $n > s > d$ (more observations than the feature dimension), and one of the benefits of sketching is to be able to store a reduced representation of the data ($\mathbb{R}^{s\times d}$ instead of $\mathbb{R}^{n\times d}$).

(b) *Random projection*: replacing $\min_{\theta\in\mathbb{R}^d} \|y - \Phi\theta\|_2^2$ by $\min_{\eta\in\mathbb{R}^s} \|y - \Phi S\eta\|_2^2$, where $S \in \mathbb{R}^{d\times s}$ is a more general sketching matrix. Here, we typically have $d > n > s$ (high-dimensional situation). The benefits of random projection are two-fold: reduction in computation time and regularization. This corresponds to replacing the corresponding feature vectors $\varphi(x) \in \mathbb{R}^d$ by $S^\top \varphi(x) \in \mathbb{R}^s$. We will consider Gaussian matrices, but also subsampling, and draw connections with kernel methods.

In the following sections, we study these precisely for the ordinary least-squares framework (it could also be done for ridge regression). We first briefly mention a commonly used related approach.

**Random forests.** A popular algorithm called random forests (Breiman, 2001) mixes both dimension reduction by projection and bagging: decision trees are learned on a bootstrapped sample of the data, with selecting a random subset of features at every splitting decision. This algorithm has nice properties (invariance to rescaling of the variables, robustness in high dimension due to the random feature selection) and can be extended in many ways. See Biau and Scornet (2016) for details.

### 10.2.1 Gaussian sketching

Following Section 3.3 on ordinary least-squares, we consider a design matrix $\Phi \in \mathbb{R}^{n\times d}$ with rank $d$ (that is, $\Phi^\top\Phi \in \mathbb{R}^{d\times d}$ invertible), which implies $n \geqslant d$. We consider $s > d$ Gaussian random projections, with typically $s \leqslant n$, but this is not necessary in the analysis below.

The estimator $\hat{\theta}^{(j)}$ is obtained by using $S^{(j)} \in \mathbb{R}^{s\times n}$, with $j = 1, \dots, m$, where $m$ denotes the number of replications. We then consider $\hat{\theta} = \frac{1}{m}\sum_{j=1}^m \hat{\theta}^{(j)}$. When $m = 1$, this is a single sketch.

We will consider the same assumptions as in Section 3.5, that is, $y = \Phi\theta_* + \varepsilon$, where $\varepsilon \in \mathbb{R}^n$ has independent zero-mean components with variance $\sigma^2$, and $\theta_* \in \mathbb{R}^d$. Our goal is to compute the fixed design error $\frac{1}{n}\mathbb{E}_{\varepsilon,S}\|\Phi\hat{\theta} - \Phi\theta_*\|_2^2$, where we take both expectations, with respect to the learning problem (in the fixed design setting, the noise vector $\varepsilon$) and the added randomization (the sketching matrices $S^{(j)}$, $j = 1, \dots, m$).

To compute this error, we first need to compute expectations and variances with respect to the random projections, assuming that $\varepsilon$ is fixed.

Since the Gaussian matrices $S^{(j)}$ are invariant by left and right multiplication by an orthogonal matrix, we can assume that the singular value decomposition of $\Phi = UDV^\top$, where $V \in \mathbb{R}^{d\times d}$ is orthogonal (i.e., $V^\top V = VV^\top = I$), $D \in \mathbb{R}^{d\times d}$ is an invertible diagonal matrix, and $U \in \mathbb{R}^{n\times d}$ has orthonormal columns (i.e., $U^\top U = I$), is such that $U = \binom{I}{0}$, and that we can write $S^{(j)} = \big(S_1^{(j)}\ S_2^{(j)}\big)$ with $S_1^{(j)} \in \mathbb{R}^{s\times d}$ and $S_2^{(j)} \in \mathbb{R}^{s\times(n-d)}$. We can also split $y$ as $y = \binom{y_1}{y_2}$ for $y_1 \in \mathbb{R}^d$ and $y_2 \in \mathbb{R}^{n-d}$.

We can write down the normal equations that define $\hat{\theta}^{(j)} \in \mathbb{R}^d$, for each $j \in \{1, \dots, m\}$, that is, $(\Phi^\top (S^{(j)})^\top S^{(j)}\Phi)\hat{\theta}^{(j)} = \Phi^\top (S^{(j)})^\top S^{(j)} y$, leading to the following closed-form estimators $\hat{\theta}^{(j)} = (\Phi^\top (S^{(j)})^\top S^{(j)}\Phi)^{-1}\Phi^\top (S^{(j)})^\top S^{(j)} y$.[1] Using the assumptions above regarding the SVD of $\Phi$, we have: $S^{(j)}\Phi = S_1^{(j)} DV^\top$. We can then expand the prediction

[1] If $s \geqslant d$, then $S^{(j)}\Phi$ has almost surely rank $d$, and thus $\hat{\theta}^{(j)}$ is uniquely defined.

vector in $\mathbb{R}^n$ as:

$$\begin{aligned}
\Phi\hat{\theta}^{(j)} &= \Phi(\Phi^\top (S^{(j)})^\top S^{(j)}\Phi)^{-1}\Phi^\top (S^{(j)})^\top S^{(j)} y \\
&= \begin{pmatrix} I \\ 0 \end{pmatrix} DV^\top (VD(S_1^{(j)})^\top S_1^{(j)} DV^\top)^{-1} VD(S_1^{(j)})^\top S^{(j)} y \\
&= \begin{pmatrix} I \\ 0 \end{pmatrix} ((S_1^{(j)})^\top S_1^{(j)})^{-1}(S_1^{(j)})^\top S^{(j)} y = \begin{pmatrix} I \\ 0 \end{pmatrix} ((S_1^{(j)})^\top S_1^{(j)})^{-1}(S_1^{(j)})^\top (S_1^{(j)} y_1 + S_2^{(j)} y_2) \\
&= \begin{pmatrix} y_1 + ((S_1^{(j)})^\top S_1^{(j)})^{-1}(S_1^{(j)})^\top S_2^{(j)} y_2 \\ 0 \end{pmatrix}.
\end{aligned}$$

Thus, since $\mathbb{E}[S_2^{(j)}] = 0$ and $S_2^{(j)}$ is independent of $S_1^{(j)}$, we get $\mathbb{E}_{S^{(j)}}\big[\Phi\hat{\theta}^{(j)}\big] = \binom{y_1}{0}$, which happens to be exactly the OLS estimator $\Phi\hat{\theta}_{\text{OLS}} = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top y = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} y$. Moreover, we have the model $y = \Phi\theta_* + \varepsilon$ and, if we split $\varepsilon$ as $\varepsilon = \binom{\varepsilon_1}{\varepsilon_2}$, we have $y = \binom{I}{0} DV^\top \theta_* + \varepsilon$, and thus $y_2 = \varepsilon_2$. We thus get:

$$\mathbb{E}_{S^{(j)}}\Big[\big\|\Phi\hat{\theta}^{(j)} - \mathbb{E}_{S^{(j)}}\Phi\hat{\theta}^{(j)}\big\|^2\Big] = \mathbb{E}_{S^{(j)}}\big[\|((S_1^{(j)})^\top S_1^{(j)})^{-1}(S_1^{(j)})^\top S_2^{(j)}\varepsilon_2\|_2^2\big].$$

Taking the expectation with respect to $\varepsilon$, and using expectations for the (inverse) Wishart distribution,[2] this leads to

$$\begin{aligned}
\mathbb{E}_{\varepsilon, S^{(j)}}\Big[\big\|\Phi\hat{\theta}^{(j)} - \mathbb{E}_{S^{(j)}}\Phi\hat{\theta}^{(j)}\big\|^2\Big] &= \sigma^2 \mathbb{E}_{S^{(j)}}\big[\operatorname{tr}\big((S_2^{(j)})^\top (S_1^{(j)}((S_1^{(j)})^\top S_1^{(j)})^{-2}(S_1^{(j)})^\top S_2^{(j)}\big)\big] \\
&= (n-d)\sigma^2 \mathbb{E}_{S_1^{(j)}}\big[\operatorname{tr}\big(((S_1^{(j)})^\top S_1^{(j)})^{-1}\big)\big] = \frac{d}{s-d-1}(n-d)\sigma^2.
\end{aligned}$$

We can now compute the overall expected generalization error:

$$\begin{aligned}
\frac{1}{n}\mathbb{E}_{\varepsilon, S^{(j)}}\Bigg[\bigg\|\frac{1}{m}\sum_{j=1}^m \Phi\hat{\theta}^{(j)} - \Phi\theta_*\bigg\|_2^2\Bigg] &= \frac{1}{n}\mathbb{E}_\varepsilon\Bigg[\bigg\|\frac{1}{m}\sum_{j=1}^m \mathbb{E}_{S^{(j)}}\big[\Phi\hat{\theta}^{(j)}\big] - \Phi\theta_*\bigg\|_2^2\Bigg] \\
&\qquad + \frac{1}{nm}\mathbb{E}_{\varepsilon, S^{(1)}}\Big[\big\|\Phi\hat{\theta}^{(1)} - \mathbb{E}_{S^{(1)}}\Phi\hat{\theta}^{(1)}\big\|^2\Big] \\
&= \frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\Phi\hat{\theta}_{\text{OLS}} - \Phi\theta_*\big\|_2^2\Big] + \sigma^2\frac{d}{nm}\frac{n-d}{s-d-1} \\
&= \sigma^2\frac{d}{n} + \sigma^2\frac{d}{nm}\frac{n-d}{s-d-1}.
\end{aligned}$$

Thus, when $m$ or $s$ tends to infinity, we recover the traditional OLS behavior, while for $m$ and $s$ finite, the performance degrades gracefully. Moreover, when $s = n$, even for $m = 1$, we get essentially twice the performance of the OLS estimator. We note that to get the same performance as OLS (up to a factor of 2), we need $m = \frac{n-d}{s-d-1} \sim \frac{n}{s}$ replications.

[2] If $S \in \mathbb{R}^{a\times b}$ has independent standard Gaussian components, then $\mathbb{E}[(S^\top S)^{-1}] = \frac{1}{a-b-1}I$ if $a > b+1$, and $\mathbb{E}[SS^\top] = bI$; see https://en.wikipedia.org/wiki/Inverse-Wishart_distribution.

As in the previous section, there is no statistical gain (here, compared to OLS), but only potentially a computational one (because some computations may be done in parallel and of reduced storage). See, e.g., Dobriban and Liu (2019) for other criteria and sketching matrices.

**Beyond Gaussian sketching.** In this section, we have chosen a Gaussian sketching matrix $S$. This made the analysis simple because of the properties of the Gaussian distribution (invariance by rotation and availability of exact expectations for inverse Wishart distributions). The analysis can be extended with more complex tools to other random sketching matrices with more attractive computational properties, such as with many zeros, leading to subsampling observations or dimensions. See Wang et al. (2018); Dobriban and Liu (2019) and references therein. For random projections below, our analysis will apply to more general sketches.

### 10.2.2 Random projections

We also consider the fixed design set-up, with a design matrix $\Phi \in \mathbb{R}^{n\times d}$ and a response vector of the form $y = \Phi\theta_* + \varepsilon$. We now assume that $d > n$ (high-dimensional set-up) and that the rank of $\Phi$ is $n$. For each $j \in \{1,\dots,n\}$, we consider a sketching matrix $S^{(j)} \in \mathbb{R}^{d\times s}$, for $s \leqslant n$ sampled independently from a distribution to be determined (we only assume that almost surely, its rank is equal to $s$). We then consider $\hat\eta^{(j)}$ as a minimizer of $\min_{\eta\in\mathbb{R}^s} \|y - \Phi S^{(j)}\eta\|_2^2$. For simplicity, we assume that the matrix $\Phi S^{(j)}$ has rank $s$, which is the case almost surely for Gaussian projections; this implies that $\hat\eta^{(j)}$ is unique, but our result applies in all situations as we are only interested in the denoised response vector. We now consider the average $\hat\theta = \frac{1}{m}\sum_{j=1}^m S^{(j)}\hat\eta^{(j)}$.

We thus consider the estimator $\hat\eta^{(j)} = \big((S^{(j)})^\top\Phi^\top\Phi S^{(j)}\big)^{-1}(S^{(j)})^\top\Phi^\top y \in \mathbb{R}^s$, obtained from the normal equation $(S^{(j)})^\top\Phi^\top\Phi S^{(j)}\hat\eta^{(j)} = (S^{(j)})^\top\Phi^\top y$, with denoised response vector

$$\hat y^{(j)} = \Phi S^{(j)}\hat\eta^{(j)} = \Phi S^{(j)}\big((S^{(j)})^\top\Phi^\top\Phi S^{(j)}\big)^{-1}(S^{(j)})^\top\Phi^\top y \in \mathbb{R}^n.$$

Denoting $\Pi^{(j)} = \Phi S^{(j)}\big((S^{(j)})^\top\Phi^\top\Phi S^{(j)}\big)^{-1}(S^{(j)})^\top\Phi^\top$, it is of the form $\hat y^{(j)} = \Pi^{(j)}y$. The matrix $\Pi^{(j)}$ is almost surely an orthogonal projection matrix into an $s$-dimensional vector space, and its expectation is denoted $\Delta \in \mathbb{R}^{n\times n}$, and is such that $\operatorname{tr}(\Delta) = s$. We have moreover $0 \preccurlyeq \Delta \preccurlyeq I$, that is, all eigenvalues of $\Delta$ are between 0 and 1.

We can then compute expectations and variances:

$$\begin{aligned}
\mathbb{E}_{S^{(j)}}\big[\hat y^{(j)}\big] &= \mathbb{E}_{S^{(j)}}\big[\Pi^{(j)}y\big] = \Delta y = \Delta\big[\Phi\theta_* + \varepsilon\big] = \Delta\varepsilon + \Delta\Phi\theta_* \\
\mathbb{E}_{S^{(j)}}\big[\hat y^{(j)}\big] - \Phi\theta_* &= \Delta\varepsilon + [\Delta - I]\Phi\theta_* \\
\mathbb{E}_{S^{(j)}}\big\|\hat y^{(j)} - \mathbb{E}_{S^{(j)}}\big[\hat y^{(j)}\big]\big\|_2^2 &= \mathbb{E}_{S^{(j)}}\big\|(\Pi^{(j)} - \Delta)y\big\|_2^2 = y^\top\mathbb{E}_{S^{(j)}}\Big[\big(\Pi^{(j)} - \Delta\big)^2\Big]y \\
&= y^\top\mathbb{E}_{S^{(j)}}\Big[\Pi^{(j)} - \Delta\Pi^{(j)} - \Pi^{(j)}\Delta + \Delta^2\Big]y \text{ since } \Pi^{(j)}\Pi^{(j)} = \Pi^{(j)}, \\
&= y^\top\big(\Delta - \Delta^2\big)y.
\end{aligned}$$

Thus, the overall (fixed design) expected generalization error is equal to:

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}_{\varepsilon,S}\bigg\|\frac{1}{m}\sum_{j=1}^{m}\hat{y}^{(j)} - \Phi\theta_*\bigg\|_2^2 \\
= \ & \frac{1}{n}\mathbb{E}_{\varepsilon}\Big[\big\|\mathbb{E}_{S^{(1)}}\big[\hat{y}^{(1)}\big] - \Phi\theta_*\big\|_2^2 + \frac{1}{m}\mathbb{E}_{S^{(1)}}\big\|\hat{y}^{(1)} - \mathbb{E}_{S^{(1)}}\big[\hat{y}^{(1)}\big]\big\|_2^2\Big] \\
& \hspace{8cm} \text{by taking expectations with respect to all } S^{(j)}, \\
= \ & \frac{1}{n}\mathbb{E}_{\varepsilon}\Big[\big\|\Delta\varepsilon + [\Delta - I]\Phi\theta_*\big\|_2^2 + \frac{1}{m}y^\top\big(\Delta - \Delta^2\big)y\Big] \text{ using the expressions above,} \\
= \ & \frac{\sigma^2}{n}\operatorname{tr}(\Delta^2) + \frac{1}{n}\theta_*^\top\Phi^\top[I-\Delta]^2\Phi\theta_* + \frac{1}{nm}\big[\sigma^2(\operatorname{tr}(\Delta) - \operatorname{tr}(\Delta^2)) + \theta_*^\top\Phi^\top\big(\Delta - \Delta^2\big)\Phi\theta_*\big] \\
& \hspace{3cm} \text{using the model } y = \Phi\theta_* + \varepsilon \text{ and the fact that } \mathbb{E}[\varepsilon] = 0 \text{ and } \mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I, \\
= \ & \frac{\sigma^2}{n}\big(1 - \frac{1}{m}\big)\operatorname{tr}(\Delta^2) + \frac{\sigma^2 s}{nm} + \frac{1}{n}\theta_*^\top\Phi^\top[\Delta - I]^2\Phi\theta_* + \frac{1}{nm}\theta_*^\top\Phi^\top\big(\Delta - \Delta^2\big)\Phi\theta_* \\
= \ & \frac{\sigma^2}{n}\big(1 - \frac{1}{m}\big)\operatorname{tr}(\Delta^2) + \frac{\sigma^2 s}{nm} + \frac{1}{n}\theta_*^\top\Phi^\top\big[I - \Delta + (\frac{1}{m} - 1)(\Delta - \Delta^2)\big]\Phi\theta_* \\
\leqslant \ & \frac{\sigma^2 s}{n} + \frac{1}{n}\theta_*^\top\Phi^\top\big[I - \Delta\big]\Phi\theta_*, \text{ since } \Delta^2 \preccurlyeq \Delta,
\end{aligned}$$

which is the value for $m = 1$ (single replication). Note that the expectation (before taking the bound) decreases in $m$. We now follow Kabán (2014); Thanei et al. (2017) to bound the matrix $I - \Delta$.

Since $\Delta$ is the expectation of a projection matrix, we already know that $0 \preccurlyeq \Delta \preccurlyeq I$. We omit the superscript ${}^{(j)}$ for clarity, and consider $\Pi = \Phi S\big(S^\top\Phi^\top\Phi S\big)^{-1}S^\top\Phi$. For any vector $z \in \mathbb{R}^n$, we consider:

$$\begin{aligned}
z^\top(I - \Delta)z \ &= \ \mathbb{E}_S\big[z^\top(I - \Pi)z\big] = \mathbb{E}_S\big[z^\top z - z^\top\Phi S\big(S^\top\Phi^\top\Phi S\big)^{-1}S^\top\Phi^\top z\big] \\
&= \ \mathbb{E}_S\Big[\min_{u\in\mathbb{R}^s}\|z - \Phi S u\|_2^2\Big] \text{ by definition of projections,} \\
&\leqslant \ \mathbb{E}_S\Big[\min_{v\in\mathbb{R}^d}\|z - \Phi S S^\top v\|_2^2\Big] \text{ by minimizing over a smaller subspace,} \\
&\leqslant \ \min_{v\in\mathbb{R}^d}\mathbb{E}_S\Big[\|z - \Phi S S^\top v\|_2^2\Big] \text{ by properties of the expectation.}
\end{aligned}$$

We can expand to get:

$$\mathbb{E}_S\Big[\|z - \Phi S S^\top v\|_2^2\Big] \ = \ \|z\|_2^2 - 2z^\top\Phi\mathbb{E}_S\big[SS^\top\big]v + v^\top\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]v,$$

leading to, after selecting the optimal $v$ as $v = \big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top z$,

$$z^\top(I - \Delta)z \ \leqslant \ z^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)z.$$

We then need to apply to $z = \Phi\theta_*$, and get:

$$\theta_*^\top\Phi^\top\big[I - \Delta\big]\Phi\theta_* \leqslant \theta_*^\top\Phi^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_*.$$

Thus, we get an overall upper bound of

$$\frac{\sigma^2 s}{n} + \frac{1}{n}\theta_*^\top \Phi^\top \Big( I - \Phi \mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_*.$$

As shown below for special cases, we obtain a bias-variance trade-off similar to Eq. (3.6) for ridge regression in Section 3.6, but now with random projections. Note that in the fixed design setting, there is no explosion of the testing performance when $s = n$ (as opposed to the random design setting studied in Section 12.2 in the context of "double descent").

**Gaussian projections.** If we assume Gaussian random projections, with $S \in \mathbb{R}^{d\times s}$ with independent standard Gaussian components, we get, from properties of the Wishart distribution:[3]

$$\mathbb{E}_S[SS^\top] = sI \text{ and } \mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big] = s(s+1)\Phi^\top\Phi + s\,\mathrm{tr}(\Phi^\top\Phi)I.$$

We then get:

$$\begin{aligned}
\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_* &\leqslant \theta_*^\top \Phi^\top \Big( I - \Phi \mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_* \\
&= \theta_*^\top \Phi^\top \Big( I - s^2\Phi\big(s(s+1)\Phi^\top\Phi + s\,\mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\Phi^\top\Big)\Phi\theta_* \\
&= \theta_*^\top\Phi^\top\Phi\big(\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)\big((s+1)\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&\leqslant 2\,\mathrm{tr}(\Phi^\top\Phi)\cdot\theta_*^\top\Phi^\top\Phi\big((s+1)\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&\leqslant 2\,\mathrm{tr}(\Phi^\top\Phi)\frac{\|\theta_*\|_2^2}{s+1}.
\end{aligned}$$

The overall excess risk is then less than

$$\frac{\sigma^2 s}{n} + \frac{2}{n}\,\mathrm{tr}(\Phi^\top\Phi)\frac{\|\theta_*\|_2^2}{s+1}, \tag{10.1}$$

which is exactly of the form obtained for ridge regression in Eq. (3.6) with $s \sim \frac{\mathrm{tr}(\Phi^\top\Phi)}{\lambda}$. We can consider other sketching matrices with additional properties, such as sparsity (see the exercise below).

**Exercise 10.4** *We consider a sketching matrix $S \in \mathbb{R}^{d\times s}$, where each column is equal to one of the $d$ canonical basis vectors of $\mathbb{R}^d$, selected uniformly at random and independently. Compute $\mathbb{E}[SS^\top]$, as well as, $\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]$, as well as a bound similar to Eq. (10.1).*

**Kernel methods. (♦)** The random projection idea can be extended to kernel methods from Chapter 7. We consider the kernel matrix $K = \Phi\Phi^\top \in \mathbb{R}^{n\times n}$, and the assumption

[3] If $W = S_1 S_1^\top$ for $S_1 \in \mathbb{R}^{n\times s}$ with independent standard Gaussian components, then $\mathbb{E}[W] = sI$ and for an $n\times n$ diagonal matrix $D$ we have $\mathbb{E}[WD^2W] = s(s+1)D^2 + s\,\mathrm{tr}(D^2)I$.

$y = \Phi\theta_* + \varepsilon$ with $\|\theta_*\|_2$ bounded, is turned into $y = y_* + \varepsilon$ with $y_*^\top K^{-1} y_*$ bounded. This corresponds to $y_* = K\alpha$, with an RKHS norm $\alpha^\top K \alpha$. We then consider a random "sketch" $\hat{\Phi} \in \mathbb{R}^{n\times s}$ and an approximate kernel matrix $\hat{K}$. We then obtain an estimate $\hat{y} = \hat{\Phi}(\hat{\Phi}^\top\hat{\Phi})^{-1}\hat{\Phi}^\top y$. The matrix $\Pi$ above is then $\Pi = \hat{\Phi}(\hat{\Phi}^\top\hat{\Phi})^{-1}\hat{\Phi}^\top$, and for the analysis, we need to compute its expectation $\Delta$. We have, following the same reasoning as above, for an arbitrary deterministic $z \in \mathbb{R}^n$:

$$\begin{aligned} z^\top(I-\Delta)z &= \mathbb{E}_{\hat{\Phi}}\big[z^\top(I-\Pi)z\big] = \mathbb{E}_{\hat{\Phi}}\big[z^\top z - z^\top\hat{\Phi}\big(\hat{\Phi}^\top\hat{\Phi}\big)^{-1}\hat{\Phi}^\top z\big] \\ &= \mathbb{E}_{\hat{\Phi}}\Big[\min_{u\in\mathbb{R}^s}\|z-\hat{\Phi}u\|_2^2\Big] \text{ by definition of projections,} \\ &\leqslant \mathbb{E}_{\hat{\Phi}}\Big[\min_{v\in\mathbb{R}^n}\|z-\hat{\Phi}\hat{\Phi}^\top v\|_2^2\Big] \text{ by minimizing over a smaller subspace,} \\ &\leqslant \min_{v\in\mathbb{R}^n}\mathbb{E}_{\hat{\Phi}}\Big[\|z-\hat{\Phi}\hat{\Phi}^\top v\|_2^2\Big] \text{ by properties of the expectation.} \end{aligned}$$

We can expand to get:

$$\mathbb{E}_{\hat{\Phi}}\Big[\|z-\hat{\Phi}\hat{\Phi}^\top v\|_2^2\Big] = \|z\|_2^2 - 2z^\top\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]v + v^\top\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]v,$$

leading to, after selecting the optimal $v$ as $v = \big(\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]\big)^{-1}\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]z$,

$$z^\top(I-\Delta)z \leqslant z^\top\Big(I - \mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\big(\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]\big)^{-1}\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\Big)z.$$

We then need to apply to $z = y_*$, we get that

$$\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_* \leqslant y_*^\top\Big(I - \mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\big(\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]\big)^{-1}\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\Big)y_*.$$

We can, for example, consider each column of $\hat{\Phi}$ to be sampled from a normal distribution with mean zero and covariance matrix $K$, for which we have:

$$\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big] = sK \quad\text{and}\quad \mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big] = s(s+1)K^2 + s\,\mathrm{tr}(K)\cdot K.$$

This leads to the bound $\dfrac{\sigma^2 s}{n} + \dfrac{2}{n}\,\mathrm{tr}(K)\dfrac{y_*^\top K^{-1}y_*}{s+1}$, which is exactly the bound in Eq. (10.1) in the kernel context. However, it is not interesting in practice as it requires the computation of the kernel matrix $K$ and typically a square root to sample from the multivariate Gaussian distribution, which has running-time complexity $O(n^3)$.

In practice, many kernels come with a random feature expansion of the form $k(x,x') = \mathbb{E}_v\big[\varphi(x,v)\varphi(x',v)\big]$, such that $|\varphi(x,v)| \leqslant R$ almost surely (as presented in Section 7.4). We can then take for each column of $\hat{\Phi}$ the vector $(\varphi(x_1,v),\dots,\varphi(x_n,v))^\top \in \mathbb{R}^n$, for a random independent $v$. We have then $\mathbb{E}\big[\hat{\Phi}\hat{\Phi}^\top\big] = sK$ by construction, while a short calculation left as an exercise shows that the second-order moment can be bounded as

$$\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big] \preccurlyeq s(s-1)K + nsR^2K.$$

This leads to the bound $\dfrac{\sigma^2 s}{n} + \dfrac{2}{n}R^2\dfrac{y_*^\top K^{-1}y_*}{s+1}$, which is almost the same as above, but with now an efficient practical algorithm.

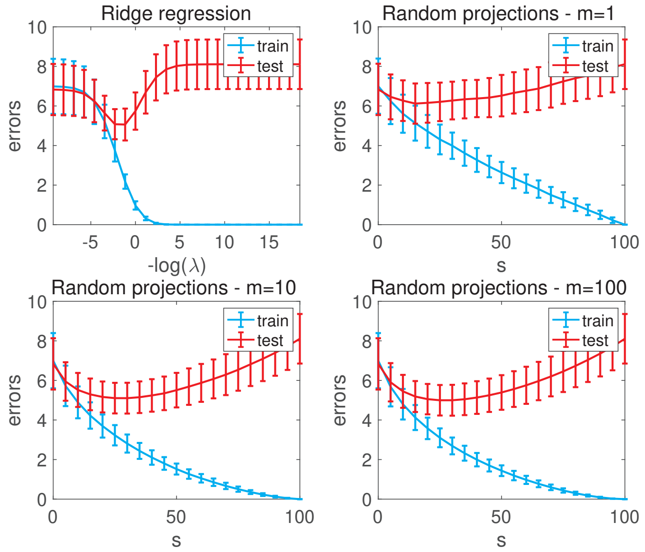


Figure 10.2: Polynomial regression in dimension 20, with polynomials of degree at most 2, with $n = 100$. Top left: training and testing errors for ridge regression in the fixed design setting (the input data are fixed, and only the noise variables are resampled for computing the test error). All other plots: training and testing errors for Gaussian random projections, with different numbers of random projections, $m = 1$ (top right), $m = 10$ (bottom left), and $m = 100$ (bottom right). All curves are averaged over 100 replications (of the noise variables and the random projections).

**Experiments.** In Figure 10.2, we consider a polynomial regression problem in dimension $d_{\mathcal{X}} = 20$, with polynomials of degree at most 2, and thus a feature space of dimension $d = 1 + d_{\mathcal{X}} + d_{\mathcal{X}}(d_{\mathcal{X}} + 1)/2 = 231$, and compare ridge regression with Gaussian random projections. We see a better performance as $m$ grows, consistent with our bounds.

**Johnson-Lindenstrauss lemma (♦).** A related classical result in Gaussian random projections shows that $n$ feature vectors $\varphi_1, \dots, \varphi_n \in \mathbb{R}^d$ can be "well-represented" in dimension $s$ by Gaussian random projections, with $s$ growing only logarithmically in $n$, and independently of the underlying dimension. The following lemma shows that all pairwise distances are preserved (a small modification would lead to all dot-products).

**Lemma 10.1 (Johnson and Lindenstrauss, 1984)** *Given $\varphi_1, \dots, \varphi_n \in \mathbb{R}^d$, let $S \in \mathbb{R}^{d\times s}$ be random matrix with independent standard Gaussian random variables. Then for any $\varepsilon \in (0, 1/2)$ and $\delta \in (0, 1)$, if $s \geqslant \frac{6}{\varepsilon^2} \log \frac{n^2}{\delta}$, with probability greater than $1 - \delta$, we have:*

$$\forall i, j \in \{1, \dots, n\},\ (1-\varepsilon)\|\varphi_i - \varphi_j\|_2^2 \leqslant \big\|s^{-1/2}S^\top\varphi_i - s^{-1/2}S^\top\varphi_j\|_2^2 \leqslant (1+\varepsilon)\|\varphi_i - \varphi_j\|_2^2. \tag{10.2}$$

**Proof (♦)** Let $\psi \in \mathbb{R}^d$ with $\ell_2$-norm equal to one. The random variable $y = \psi^\top SS^\top\psi$ is the sum of $s$ random variables $\psi^\top S_{\cdot j} S_{\cdot j}^\top \psi$, for $S_{\cdot j}$ the $j$-th column of $S$, $j \in \{1, \dots, s\}$. Each of these is the square of $S_{\cdot j}^\top \psi$ which is Gaussian with mean zero and variance equal to $\|\psi\|_2^2 = 1$. Thus, $y$ is a chi-squared random variable. We can thus apply concentration results from Exercise 8.1, leading to

$$\mathbb{P}\big(|y - s| \geqslant s\varepsilon\big) \leqslant \Big(\frac{1-\varepsilon}{\exp(-\varepsilon)}\Big)^{s/2} + \Big(\frac{1+\varepsilon}{\exp(\varepsilon)}\Big)^{s/2}.$$

We can then use the inequality $\log(1+u) \leqslant u - \frac{u^2}{3}$ for any $|u| \leqslant 1/2$, leading to the probability bound $\mathbb{P}\big(|y - s| \geqslant s\varepsilon\big) \leqslant 2\exp\big(-\frac{s}{2}\frac{\varepsilon^2}{3}\big)$. We then apply the reasoning above to the $n(n-1)/2$ vectors $\varphi_i - \varphi_j$, for $i \neq j$, leading to, using a union bound, a probability that Eq. (10.2) is not satisfied with probability less than $n^2 \exp(-s\varepsilon^2/6)$, leading to the desired result. ■

In our context of least-squares regression, the Johnson-Lindenstrauss lemma shows that the kernel matrix is preserved by random projections so that predictions with the projected data should be close to predictions with the original data. The results in this section provide a direct proof aiming at characterizing directly the predictive performance of such random projections.

## 10.3 Boosting

In the previous section, we focused on uniformly combining the outputs (e.g., plain averaging) of estimators obtained by randomly reweighted versions of the original datasets. Reweighting was performed independently of the performance of the resulting prediction functions, and the training procedures for all predictors could be done in parallel. In this section, we explore *sequential* reweightings of the training datasets that depend on the mistakes made by the current prediction functions.

In the early boosting procedures adapted to binary classification, the original learning procedure was used directly on a reweighted version, e.g., Adaboost (see, e.g., Freund et al., 1999). Our analysis will be carried out for boosting procedures, often referred to as "gradient boosting", which are adapted to real-valued outputs, as done in the rest of the book (noting that for classification, we can use convex surrogates).

The theory of boosting is rich, with many connections, and in this section, we only provide a consistency proof in the simplest setting. See Schapire and Freund (2012) for more details.

### 10.3.1 Problem set-up

Given an input space $\mathcal{X}$, and $n$ observations $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, $i = 1, \dots, n$, we are given a set of predictors $\varphi(w, \cdot) : \mathcal{X} \to \mathbb{R}$, for $w \in \mathcal{W}$, with $\mathcal{W}$ typically a compact subset of a finite-dimensional vector space.

The main assumption is that given weights $\alpha \in \mathbb{R}^n$, one can "easily" find the function $\varphi(w,\cdot)$ that minimizes with respect to $w \in \mathcal{W}$

$$\sum_{i=1}^n \alpha_i \varphi(w, x_i),$$

that is, the dot-product between $\alpha$ and the $n$ outputs of $\varphi(w,\cdot)$ on the $n$ observations. In this section, for simplicity, we assume that this minimization can be done exactly. This is often referred to as the "weak learner" assumption. Many examples are available, such as:

- Linear stumps for $\mathcal{X} = \mathbb{R}^d$: $\varphi(w,x) = \pm(w_0^\top x + w_1)_+$, with sometimes the restriction that $w$ has only non-zero components along a single coordinate (where the weak learning tractability assumption is indeed verified). This will lead to a predictor, which is a one-hidden layer neural network, but learned sequentially (rather than by gradient descent on the empirical risk). In the context of binary classification, the weak learners are sometimes thresholded to values in $\{-1, 1\}$ by taking their signs.
- Decision trees for $\mathcal{X} = \mathbb{R}^d$: we consider here the space of piecewise constant functions of $x$, where the pieces with constant values are obtained by recursively partitioning the input space into half-spaces with normals along one of the coordinate axes. In this situation, the set of functions is more easily characterized through the estimation algorithm. See Chen and Guestrin (2016) for an efficient implementation of a boosting algorithm based on decision trees (referred to as "XGBoost").

In this section, we assume bounded features, that is, for all $w \in \mathcal{W}$, and inputs $x \in \mathcal{X}$, $|\varphi(x,w)| \leqslant R$. Moreover, for simplicity, we assume that the set of feature functions $\{\varphi(\cdot,w),\ w \in \mathcal{W}\}$ is centrally symmetric, which is the case for the examples above.

Boosting procedures will make sequential calls to the weak learner oracle that outputs $w_1, \dots, w_t \in \mathcal{W}$ with $t$ the number of iterations, and *linearly combine* the function $\varphi(w_1,\cdot), \dots, \varphi(w_t,\cdot)$. Therefore, the set of predictors that are explored are not only the functions $\varphi(w,\cdot)$, but all linear combinations, that is, functions of the form

$$f(x) = \int_{\mathcal{W}} \varphi(w,x) d\nu(w), \tag{10.3}$$

for $\nu$ a (signed) measure on $\mathcal{W}$, which we assume to have finite mass.

To avoid overfitting, some norm that will be explicitly or implicitly controlled needs to be defined. As done in Section 9.3.2 with neural networks, we will consider an $L_1$-norm, namely the total variation of $\nu$, that is:

$$\int_{\mathcal{W}} |d\nu(w)|.$$

Note that since we have assumed that the features are centrally symmetric, assuming that $\nu$ is a positive measure does not change anything.

For functions $f : \mathcal{X} \to \mathbb{R}$ that can be represented as integrals in Eq. (10.3), the minimal value of $\int_{\mathcal{W}} |d\nu(w)|$ is referred to as the "variation norm" (Kurková and Sanguineti, 2001), or the "atomic norm" (Chandrasekaran et al., 2012), of $f$, and the set of functions with finite norm will be denoted $\mathcal{F}_1$, with a norm $\gamma_1$. Like in Section 9.3.2, this is to distinguish it from the squared norm $\int_{\mathcal{W}} \big| \frac{d\nu(w)}{d\tau(w)} \big|^2 d\tau(w)$ for a fixed positive measure $\tau$, which corresponds to a reproducing kernel Hilbert space (see Chapter 7).

Note that by definition, for any $w \in \mathcal{W}$, $\gamma_1(\varphi(w,\cdot)) \leqslant 1$, since we can represent this function by the measure $\nu = \delta_w$. Since we will optimize over the realizations of the features on the data, we denote by $\psi(w) \in \mathbb{R}^n$ the vector so that $\psi(w)_i = \varphi(x_i, w)$. Since $|\varphi(x,w)| \leqslant R$ for all $w$ and $x$, $\|\psi(w)\|_2 \leqslant R\sqrt{n}$ for all $w$. By restricting to values on $x_1, \dots, x_n$, we obtain a penalty defined on $\mathbb{R}^n$ with a definition similar to $\gamma_1$ defined on functions from $\mathcal{X}$ to $\mathbb{R}$, with more properties we will need for our proofs.

**Gauge function.** We define the function $\gamma : \mathbb{R}^n \to \mathbb{R}$ as the infimum of $\int_{\mathcal{W}} |d\nu(w)|$ over all positive measures such that $u = \int_{\mathcal{W}} \psi(w) d\nu(w)$. This function is usually referred to as the "gauge" function associated with the convex hull of all $\psi(w)$, $w \in \mathcal{W}$ (Rockafellar, 1997). The gauge function $\gamma$ is always convex and positively homogeneous. Since we further assumed central symmetry of the features, that is, the set $\{\psi(w), w \in \mathcal{W}\} \subset \mathbb{R}^n$ is centrally symmetric, $\gamma(-u) = \gamma(u)$ for all $u \in \mathbb{R}^n$. Given our bounded norm assumption $\|\psi(w)\|_2 \leqslant R\sqrt{n}$, we have, for any $u$ such that $\gamma(u)$ is finite (and with associated measure $\nu$), $\|u\|_2 = \big\| \int_{\mathcal{W}} \psi(w) d\nu(w) \big\|_2 \leqslant \int_{\mathcal{W}} \|\psi(w)\|_2 |d\nu(w)| \leqslant R\sqrt{n}\gamma(u)$.

The gauge function may not be a norm since it may not be finite everywhere, that is, there may exist $u \in \mathbb{R}^n$ which cannot be expressed as a linear combination of feature vectors $\psi(w)$, $w \in \mathcal{W}$. We may, however, define a notion of dual gauge function, called a "polar" gauge $\gamma^* : \mathbb{R}^n \to \mathbb{R}$, as $\gamma^*(v) = \sup_{w \in \mathcal{W}} \psi(w)^\top v$, which leads to a form a Cauchy-Schwarz inequality, as $u^\top v \leqslant \gamma(u)\gamma^*(v)$ (see Rockafellar, 1997, Chapter 15, for more details).

**Assumptions.** Following our traditional empirical risk minimization framework presented in Chapter 4, we consider a loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, both for regression and classification. Since we will need differentiable loss functions, our developments are restricted to the logistic loss, the exponential loss, and the square loss. We denote by $\ell_i : \mathbb{R} \to \mathbb{R}$ the loss the observation $(x_i, y_i)$, that is, $\ell_i(u_i) = \ell(y_i, u_i)$. We thus consider the logistic loss $\ell_i(u_i) = \log(1 + \exp(-y_i u_i))$ and the exponential loss $\ell_i(u_i) = \exp(-y_i u_i)$ when $y_i \in \{-1, 1\}$, or the square loss $\ell_i(u_i) = \frac{1}{2}(y_i - u_i)^2$ when $y_i \in \mathbb{R}$.

In our optimization convergence proofs in Section 10.3.5, we will need that each loss $\ell_i$ is smooth, with smoothness constant $G_2$ (e.g., $1/4$ for the logistic loss and 1 for the square loss, and $+\infty$ for the exponential loss). This leads to a loss function $F : \mathbb{R}^n \to \mathbb{R}$, defined as $F(u) = \frac{1}{n} \sum_{i=1}^n \ell_i(u_i)$, which is $(G_2/n)$-smooth. For the statistical consistency proof, we will also need that the loss functions are $G_1$-Lipschitz continuous, which only applies to logistic regression, and that $\ell_i(0)$ has a uniform bound $G_0$ (for logistic regression, $G_0 = \log 2$).

**Finite $\mathcal{W}$.** While boosting methods can be applied for any compact set $\mathcal{W}$ (as long as the minimization oracle is available), an interesting special case corresponds to finite sets

$\mathcal{W} = \{w_1, \dots, w_d\}$. The optimization problem that we aim to solve is the minimization of $F(u)$, for $u$ in the span of all $\psi(w_1), \dots, \psi(w_d)$, which we can rewrite as:

$$\min_{u \in \mathbb{R}^n} F(u) \text{ such that } \exists \alpha \in \mathbb{R}^d, \ u = \sum_{j=1}^d \alpha_j \psi(w_j) = \min_{\alpha \in \mathbb{R}^d} F\Big( \sum_{j=1}^d \alpha_j \psi(w_j) \Big). \tag{10.4}$$

We can thus either see this as an optimization problem in $u$ *or* in $\alpha$, and, given our assumptions regarding central symmetry, the norm $\gamma$ on $u$ is upper-bounded by the $\ell_1$-norm of $\alpha$. Seeing Eq. (10.4) as a problem in $u$ may be advantageous because of strong-convexity properties that could be lost for the problem in $\alpha$ (in particular when $n \leqslant d$): for example, for the square loss, where $F$ is strongly-convex, the optimization problem in $u$ is strongly-convex, and thus exhibits linear convergence, while the problem in $\alpha$ is not strongly-convex (but still exhibits linear convergence for other reasons, see Section 12.1.1).

### 10.3.2 Incremental learning

The simplest version of boosting-like algorithms aims to construct linear combinations of functions of the form $x \mapsto \varphi(w_t, x)$ by selecting incrementally $w_t \in \mathcal{W}$. Starting from the function $g_0 = 0$, we thus consider the simplest update

$$g_t = g_{t-1} + b_t \varphi(w_t, \cdot), \tag{10.5}$$

where the linear combination coefficients for $\varphi(w_1, \cdot), \dots, \varphi(w_{t-1}, \cdot)$ are not changed once they are computed. Given the empirical risk $\widehat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i))$, a natural criterion for the choice of $b_t \in \mathbb{R}$ and $w_t \in \mathcal{W}$ is to solve the optimization problem

$$\min_{b_t \in \mathbb{R}_+, \ w_t \in \mathcal{W}} \widehat{\mathcal{R}}\big(g_{t-1} + b_t \varphi(w_t, \cdot)\big). \tag{10.6}$$

With our notations, and since only values at $x_1, \dots, x_n$ are used for the functions $g_t$, we can represent them with their values on these points, that is, by a vector $u_t \in \mathbb{R}^n$ such that $(u_t)_i = g_t(x_i)$ for all $i \in \{1, \dots, n\}$. The update in Eq. (10.5) then becomes

$$u_t = u_{t-1} + b_t \psi(w_t),$$

and the update in Eq. (10.6) becomes

$$\min_{b_t \in \mathbb{R}_+, \ w_t \in \mathcal{W}} F\big(u_{t-1} + b_t \psi(w_t)\big). \tag{10.7}$$

This minimization is easily done in two situations: for the square loss, leading to matching pursuit (Mallat and Zhang, 1993) and for the exponential loss, leading to Adaboost (Freund and Schapire, 1996). We now present these two classical algorithms and some elements of analysis of their convergence rates for optimizing the empirical risk. We then analyze the expected risk, which is more involved when the goal is to obtain a convergence rate with early stopping.

### 10.3.3 Matching pursuit

Matching pursuit corresponds to the iteration in Eq. (10.7) for the square loss, with applications beyond machine learning, in particular in signal processing (Mallat and Zhang, 1993). For simplicity, only in this section, we assumed that each $x \mapsto \varphi(x, w)$, for $w \in \mathcal{W}$, is normalized on the data, that is $\sum_{i=1}^n \varphi(x_i, w)^2 = \|\psi(w)\|_2^2 = n$. This implies that for all $u \in \mathbb{R}^n$, $\|u\|_2 \leqslant \sqrt{n}\gamma(u)$.

In our context of empirical risk minimization, the square loss corresponds to $F(u) = \frac{1}{2n}\|y-u\|_2^2$, and, because of the normalization, we have:

$$\begin{array}{rcl} F(u_t) & = & F(u_{t-1}) + F'(u_{t-1})^\top (u_t - u_{t-1}) + \dfrac{1}{2n}\|u_t - u_{t-1}\|_2^2 \\ & = & F(u_{t-1}) + F'(u_{t-1})^\top b_t \psi(w_t) + \dfrac{b_t^2}{2}. \end{array}$$

Optimizing with respect to $b_t \in \mathbb{R}$ leads to $b_t = -F'(u_{t-1})^\top \psi(w_t)$, leading to the optimal value

$$F(u_{t-1}) - \frac{1}{2}|F'(u_{t-1})^\top \psi(w_t)|^2.$$

Since $F'(u_{t-1}) = \frac{1}{n}(u_{t-1} - y)$, the iteration can then be written as, initialized with $u_0 = 0$, for $t \geqslant 1$:

$$\left\{ \begin{array}{rcl} w_t & = & \displaystyle\arg\max_{w \in \mathcal{W}} \big|(u_{t-1} - y)^\top \psi(w)\big| \\ u_t & = & u_{t-1} - \frac{1}{n}\big|(u_{t-1} - y)^\top \psi(w_t)\big|\psi(w_t) = u_{t-1} - \frac{1}{n}\gamma^*(u_{t-1} - y)\psi(w_t), \end{array} \right.$$

by definition of the polar gauge function $\gamma^*$.

**Slow convergence.** The minimizer of $F(u) = \frac{1}{2n}\|u - y\|_2^2$ is $u_* = y$. It may or may not be such that $\gamma(y)$ is finite. In this section on matching pursuit, we assume it is, but we consider the general case in Section 10.3.5. It turns out that the penalty $\gamma(y)$ provides an explicit control of the convergence rate of $u_t$ towards $y$. Indeed, it can be shown that the matching pursuit algorithm converges with a rate proportional to $\gamma(y)$, that is,

$$\frac{1}{n}\|y - u_t\|_2^2 \leqslant \gamma(y)^2 t^{-1/3}.$$

See DeVore and Temlyakov (1996) for a detailed result (proved in Exercise 10.6), Section 10.3.5 for a related result for all smooth loss functions (and with a detailed proof), and Sil'nichenko (2004) for an improved dependence on $t$, and Klusowski and Siegel (2023) for lower bounds.

**Fast convergence of the empirical risk.** As already obtained by Mallat and Zhang (1993), exponential rates can be obtained with the stronger assumption that $\gamma$ is a norm on $\mathbb{R}^n$, and then we have by equivalence of norms: $\sqrt{n}\kappa\gamma(u) \leqslant \|u\|_2$, and $\gamma^*(v) \geqslant \kappa\sqrt{n}\|v\|_2$, for a constant $\kappa > 0$ which has to be less than 1, since $\|u\|_2 \leqslant \sqrt{n}\gamma(u)$. For finite sets $\mathcal{W} = \{w_1, \dots, w_d\}$, this corresponds to the kernel matrix $\sum_{i=1}^d \psi(w_i)\psi(w_i)^\top \in$

$\mathbb{R}^{n\times n}$ being invertible. As shown below, this ensures constant multiplicative progress across matching pursuit iterations. Indeed, we then have:

$$\frac{1}{2n}\|y-u_t\|_2^2 = \frac{1}{2n}\|y-u_{t-1}\|_2^2 - \frac{1}{2n^2}\gamma^*(u_{t-1}-y)^2 \leqslant (1-\kappa^2)\cdot\frac{1}{2n}\|y-u_{t-1}\|_2^2,$$

leading to exponential convergence.

**Exercise 10.5** *(♦) Orthogonal matching pursuit is a modification of matching pursuit which, once $w_t \in \mathcal{W}$ has been selected, defines $u_t$ as the minimizer of $F$ over the span of all previously selected feature vectors $\psi(w_1),\dots,\psi(w_t)$. Show that $\frac{1}{n}\|y-u_t\|_2^2 \leqslant \gamma(y)^2 t^{-1}$.*

### 10.3.4 Adaboost

Adaboost (Freund and Schapire, 1996) corresponds to the binary classification case, where we assume that $\varphi(x,w)\in\{-1,1\}$, that is, all weak learners are already classification functions, or, equivalently, $\psi(w)\in\{-1,1\}^n$, and we use the exponential loss, that is,

$$F(u) = \frac{1}{n}\sum_{i=1}^n \exp(-y_i u_i).$$

We can then implement Eq. (10.7) by solving

$$\min_{b_t\in\mathbb{R}, w_t\in\mathcal{W}} F(u_{t-1}+b_t\psi(w_t)) = \min_{b_t\in\mathbb{R}, w_t\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \exp(-y_i(u_{t-1})_i)\exp(-b_t y_i\psi(w_t)_i).$$

Using the fact that $y_i\psi(w_t)_i\in\{-1,1\}$ for all $i\in\{1,\dots,n\}$, this is equivalent to

$$\min_{b_t\in\mathbb{R}, w_t\in\mathcal{W}} \sum_{i=1}^n \Big\{\frac{e^{-b_t}}{n}\frac{1+y_i\psi(w_t)_i}{2} + \frac{e^{b_t}}{n}\frac{1-y_i\psi(w_t)_i}{2}\Big\}e^{-y_i(u_{t-1})_i},$$

with an objective function proportional to $\sum_{i=1}^n\big\{\frac{e^{-b_t}}{n}\frac{1+y_i\psi(w_t)_i}{2}+\frac{e^{b_t}}{n}\frac{1-y_i\psi(w_t)_i}{2}\big\}\pi_i$, where $\pi$ is a vector in the simplex defined as $\pi_i = \frac{e^{-y_i(u_{t-1})_i}}{\sum_{j=1}^n e^{-y_j(u_{t-1})_j}}$.

Given $w_t\in\mathcal{W}$, the optimal $b_t$ is obtained by minimizing a function of the form $e^{-b_t}a_- + e^{b_t}a_+$, for some constants $a_+$ and $a_-$, which is attained as $b_t = \frac{1}{2}\log\frac{a_-}{a_+}$, with optimal value equal to $2\sqrt{a_-a_+}$. Thus, the optimal $b_t$ is equal to

$$b_t = \frac{1}{2}\log\frac{1+\sum_{i=1}^n y_i\psi(w_t)_i\pi_i}{1-\sum_{i=1}^n y_i\psi(w_t)_i\pi_i},$$

and the resulting objective function (that depends on $w_t$) is equal to

$$F(u_{t-1})\Big[1-\Big(\sum_{i=1}^n y_i\psi(w_t)_i\pi_i\Big)^2\Big].$$

We can thus obtain $w_t$ by maximizing $\big|\sum_{i=1}^n y_i\psi(w_t)_i\pi_i\big|$. Since we have assumed central symmetry of the weights, we can equivalently maximize $\sum_{i=1}^n y_i\psi(w_t)_i\pi_i$, which corresponds to finding the weak learner with minimal 0-1 classification error weighted by $\pi$. We thus get the following iteration:

$$\begin{cases} \pi_i = \dfrac{e^{-y_i(u_{t-1})_i}}{\sum_{j=1}^n e^{-y_j(u_{t-1})_j}} \text{ for } i \in \{1,\dots,n\} \\ w_t \in \arg\max\limits_{w\in\mathcal{W}} \displaystyle\sum_{i=1}^n y_i\psi(w_t)_i\pi_i \\ u_t = u_{t-1} + \dfrac{1}{2}\log\dfrac{1+\sum_{i=1}^n y_i\psi(w_t)_i\pi_i}{1-\sum_{i=1}^n y_i\psi(w_t)_i\pi_i}\psi(w_t). \end{cases}$$

After this iteration, we have

$$F(u_t) = F(u_{t-1})\Big[1 - \Big(\sum_{i=1}^n y_i\psi(w_t)_i\pi_i\Big)^2\Big].$$

Therefore, the empirical risk (with the exponential loss) strictly decreases if the weak learner gets an empirical weighted 0-1 loss strictly less than 1/2 (corresponding to the dot product with $y$ being strictly positive). If the error rate is always less than a constant, an assumption referred to as weak learnability, we obtain a linear convergence. Note that if we make the same assumption as in matching pursuit in Section 10.3.3, then $\sum_{i=1}^n y_i\psi(w_t)_i\pi_i = \gamma^*(\pi\circ y) \geqslant \rho\sqrt{n}\|\pi\|_2 \geqslant \rho\|\pi\|_1 \geqslant \rho$, and we have a similar exponential convergence rate.

### 10.3.5 Greedy algorithm based on gradient boosting

In this section, we describe a boosting algorithm which, at each iteration, performs a first-order Taylor expansion at the current point (which requires to compute derivatives of the loss functions) and find the weak learner $x \mapsto \varphi(w,x)$ that makes most progress for this approximation of the risk. We thus consider the following "greedy" algorithm, starting from the zero function $g_0 = 0$, and iterating over $t \geqslant 1$:

- Loss gradient computations: compute $\alpha_i = \ell_i'(g_{t-1}(x_i))$ for $i \in \{1,\dots,n\}$.
- Weak learner: compute $w_t \in \mathcal{W}$ that minimizes $\sum_{i=1}^n \alpha_i\varphi(w,x_i)$ with respect to $w \in \mathcal{W}$. Equivalently, using our notations in $\mathbb{R}^n$, we minimize $F'(u_{t-1})^\top\psi(w)$ with respect to $w \in \mathcal{W}$.
- Function update: take $g_t = g_{t-1} + b_t\varphi(w_t,\cdot)$ for a coefficient $b_t \in \mathbb{R}_+$ that optimizes an upper-bound on the empirical risk. This corresponds to $u_t = u_{t-1} + b_t\psi(w_t)$.

After time $t$, the prediction function $g_t$ will be a linear combination of the functions $\varphi(w_u,\cdot)$, for $u \in \{1,\dots,t\}$, with only $t$ atoms, thus leading to sparse combinations (in other words, the estimated measure $\nu$ is a sum of Diracs). For the square loss, this will exactly be the matching pursuit algorithm presented in Section 10.3.3. In general, these algorithms are often referred to as "gradient boosting" procedures (Friedman, 2001).

We first provide a generic convergence result for the empirical risk (which goes beyond machine learning problems), before proving a convergence rate for the expected risk in Section 10.3.6. We focus on smooth loss functions for the optimization result, while we require a smooth and Lipschitz-continuous loss function for the statistical analysis (such as the logistic loss). For consistency results for the exponential loss, see Bartlett and Traskin (2007).

With our smoothness assumption, we can define the upper-bound on $F(u_t)$ as (using the definition of smoothness in Eq. (5.10)):

$$\begin{aligned} F(u_t) & \leqslant & F(u_{t-1}) + F'(u_{t-1})^\top (u_t - u_{t-1}) + \frac{L}{2}\|u_t - u_{t-1}\|_2^2 \\ & \leqslant & F(u_{t-1}) + b_t F'(u_{t-1})^\top \psi(w_t) + \frac{L}{2} b_t^2 \|\psi(w_t)\|_2^2 \\ & & \text{using the expression } u_t = u_{t-1} + b_t\psi(w_t), \\ & \leqslant & F(u_{t-1}) + b_t F'(u_{t-1})^\top \psi(w_t) + \frac{L}{2} b_t^2 C^2, \end{aligned} \tag{10.8}$$

if $L$ is the smoothness constant of $F$ and $C$ an uniform upper-bound on all $\|\psi(w)\|_2$, $w \in \mathcal{W}$. This naturally leads to the iteration

$$\left\{ \begin{array}{l} w_t \in \arg\max_{w \in \mathcal{W}} \ F'(u_{t-1})^\top \psi(w) \\ u_t = u_{t-1} - \frac{1}{LC^2} F'(u_{t-1})^\top \psi(w_t), \end{array} \right. \tag{10.9}$$

which we can now analyze, to obtain upper-bounds on both function values and the gauge functions of the iterates.

**Proposition 10.1 (convergence of gradient boosting algorithm)** *We consider an $L$-smooth function $F : \mathbb{R}^n \to \mathbb{R}$; we assume that $\psi : \mathcal{W} \to \mathbb{R}^n$ is such that $\|\psi(w)\|_2 \leqslant C$ for all $w \in \mathcal{W}$, and the associated gauge function $\gamma$ is centrally symmetric. Consider the iteration in Eq. (10.9). Then for any $v \in \mathbb{R}^n$, and $t > 0$, we have:*

$$\big(F(u_t) - F(v)\big)_+ \leqslant \Big(\frac{2LC^2 \gamma(u_0 - v)^2 (F(u_0) - F(v))_+^4}{t}\Big)^{1/5},$$

*and*

$$\gamma(u_t) \leqslant \gamma(u_0) + \frac{\sqrt{t}}{LC^2}\Big(2LC^2[F(u_0) - F(u_t)]\Big)^{1/2}.$$

**Proof** We have by construction of the iteration and from Eq. (10.8):

$$F(u_t) - F(v) \leqslant F(u_{t-1}) - F(v) - \frac{1}{2LC^2}\gamma^*(F'(u_{t-1}))^2. \tag{10.10}$$

Moreover, using the convexity of $F$ and properties of gauge functions, we have:

$$F(u_t) - F(v) \leqslant F'(u_t)^\top (u_t - v) \leqslant \gamma^*(F'(u_t))\gamma(u_t - v). \tag{10.11}$$

Finally, using the triangular inequality for $\gamma$, we obtain, from Eq. (10.9), $\gamma(u_t - v) \leqslant \gamma(u_{t-1} - v) + \frac{1}{LC^2}\gamma^*(F'(u_{t-1}))$, leading to, by recursion, $\gamma(u_t - v) \leqslant \Gamma_t$, where $\Gamma_t =$

$\gamma(u_0 - v) + \frac{1}{LC^2}\gamma^*(F'(u_{t-1})) + \cdots + \frac{1}{LC^2}\gamma^*(F'(u_0))$. We define $\Delta_t = (F(u_t) - F(v))_+$. From Eq. (10.10), we get:

$$\Delta_t \leq \left(\Delta_{t-1} - \tfrac{1}{2LC^2}\gamma^*(F'(u_{t-1}))^2\right)_+, \tag{10.12}$$

and from Eq. (10.11), we get $\Delta_t \leqslant \Gamma_t \gamma^*(F'(u_t))$. Thus,

$$\begin{aligned}\Delta_t\Gamma_t^{-2} &\leqslant \Delta_t\Gamma_{t-1}^{-2} \leqslant \left(\Delta_{t-1}\Gamma_{t-1}^{-2} - \tfrac{1}{2LC^2}\Gamma_{t-1}^{-2}\gamma^*(F'(u_{t-1}))^2\right)_+ \\ &\leqslant \left(\Delta_{t-1}\Gamma_{t-1}^{-2} - \tfrac{1}{2LC^2}(\Delta_{t-1}\Gamma_{t-1}^{-2})^2\right)_+.\end{aligned}$$

This leads to[4]

$$\Delta_t\Gamma_t^{-2} \leqslant \frac{1}{\frac{t}{2LC^2} + \Gamma_0^2\Delta_0^{-1}} \leqslant \frac{2LC^2}{t}. \tag{10.13}$$

Moreover, by definition of $\Gamma_t$ and using Eq. (10.11), we have:

$$\Gamma_t = \Gamma_{t-1} + \frac{1}{LC^2}\gamma^*(F'(u_{t-1})) \leqslant \Gamma_{t-1}\left(1 + \frac{1}{LC^2}\frac{\gamma^*(F'(u_{t-1})^2}{\Delta_{t-1}}\right).$$

Thus, by taking the product of the square of Eq. (10.12) and the previous inequality, we get:

$$\Gamma_t\Delta_t^2 \leqslant \Gamma_{t-1}\Delta_{t-1}^2\left(1 + \frac{1}{LC^2}\gamma^*(F'(u_{t-1}))^2\right)\left(1 - \frac{1}{2}\frac{1}{LC^2}\gamma^*(F'(u_{t-1}))^2\right)_+^2.$$

Since $(1 - \varepsilon/2)_+^2(1 + \varepsilon) \leqslant 1$ for all $\varepsilon \geqslant 0$, this leads to $\Gamma_t\Delta_t^2 \leqslant \Gamma_{t-1}\Delta_{t-1}^2$, and thus $\Gamma_t\Delta_t^2 \leqslant \Gamma_0\Delta_0^2$. Taking the product of the square of the inequality above with Eq. (10.13), this leads to

$$\Delta_t^5 = (\Gamma_t\Delta_t^2)^2 \cdot \Delta_t\Gamma_t^{-2} \leqslant (\Gamma_0\Delta_0^2)^2\frac{2LC^2}{t},$$

which leads to the first result.

We can also bound the norm $\gamma(u_t)$ as follows, using Eq. (10.10):

$$\begin{aligned}\gamma(u_t) &\leqslant \gamma(u_0) + \frac{1}{LC^2}\sum_{i=1}^t \gamma^*(F'(u_{i-1})) \leqslant \gamma(u_0) + \frac{\sqrt{t}}{LC^2}\left(\sum_{i=1}^t \gamma^*(F'(u_{i-1}))^2\right)^{1/2} \\ &\leqslant \gamma(u_0) + \frac{\sqrt{t}}{LC^2}\left(2LC^2[F(u_0) - F(u_t)]\right)^{1/2}.\end{aligned}$$

■

We will need the flexibility of having an arbitrary $v \in \mathbb{R}^n$ in the statistical consistency proof, but when $v$ is chosen as the minimizer $u_*$ of $F$ (then assumed to exist), we get a more traditional optimization bound:

$$F(u_t) - F(u_*) \leqslant \left(\frac{2LC^2\gamma(u_0 - u_*)^2(F(u_0) - F(u_*))^4}{t}\right)^{1/5} \leqslant \frac{LC^2\gamma(u_0 - u_*)^2}{t^{1/5}},$$

[4] We can use the following lemma, whose proof is left as an exercise: if $(a_t)$ is a non-increasing non-negative sequence such that $a_t \leqslant (a_{t-1} - ca_{t-1}^2)_+$ for all $t \geqslant 1$, then $a_t \leqslant \frac{1}{t/c + 1/a_0}$ for all $t \geqslant 0$.

which can be compared to the bound for regular gradient descent (Prop. 5.5), which is $\frac{L}{2t}\|u_0 - u_*\|_2^2 \leqslant \frac{LC^2}{2t}\gamma(u_0 - u_*)$, with a better dependence on $t$, but with iterates that cannot be expressed as linear combinations of at most $t$ iterates.

As done in the next section, assuming that $u_0 = 0$ and $F$ is non-negative everywhere, this leads to:

$$\big(F(u_t) - F(v)\big)_+ \leqslant \Big(\frac{2LC^2\gamma(v)^2 F(0)^4}{t}\Big)^{1/5} \quad \text{and} \quad \gamma(u_t) \leqslant \frac{\sqrt{2t}}{\sqrt{LC^2}} F(0)^{1/2}.$$

The proposition above shows that the gauge function $\gamma$ controls the convergence of the gradient-boosting algorithm, in the same way that the Euclidean norm controls the convergence of gradient descent. For finite sets $\mathcal{W}$, where the gauge function is essentially an $\ell_1$-norm in a reparameterization, the link with $\ell_1$-norm penalization can be made explicit (see, e.g., Rosset et al., 2004, for details).

**Exercise 10.6** *(♦) Show that when the function $F$ is quadratic, then have the guarantee: $F(u_t) - F(u_*) \leqslant \frac{LC^2}{2t^{1/3}}\gamma(u_0 - u_*)^2$. Hint: replace Eq. (10.11) by $F(u_t) - F(u_*) = \frac{1}{2}F'(u_t)^\top(u_t - u_*)$.*

### 10.3.6 Convergence of expected risk

In order bound the expected risk, we need to relate empirical risk $\widehat{\mathcal{R}}$ and expected risk $\mathcal{R}$, for functions $f$ with bounded penalty $\gamma_1(f)$. To study the generalization performance of constraining or penalizing by the variation norm defined above, we can naturally use the general framework of Rademacher complexities presented in Section 4.5.

**Statistical performance through Rademacher complexities.** The uniform deviations for the set of predictors $g : \mathcal{X} \to \mathbb{R}$ such that $\gamma_1(g) \leqslant D$ on i.i.d. data $x_1, \dots, x_n$ are controlled by the quantity

$$\mathbb{E}\Big[\sup_{\gamma_1(g)\leqslant D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)\Big] = D \cdot \mathbb{E}\Big[\sup_{w\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i \varphi(x_i, w)\Big], \tag{10.14}$$

where the expectation is taken both with respect to the data $x_1, \dots, x_n$ and the independent Rademacher random variables $\varepsilon_1, \dots, \varepsilon_n \in \{-1, 1\}$.

In Section 9.2.3, we computed an upperbound proportional to $DR/\sqrt{n}$ for $\varphi(x, w)$ of the form $\sigma(x^\top w)$ (which corresponds to learning a one-hidden layer neural network), with an extra factor of $\sqrt{\log d}$ for $\ell_1$-norm constraint on neural network weights, showing that although the set $\mathcal{W}$ is infinite, we can bound the uniform deviations. See another example in the exercise below. In the following, we will assume that

$$\mathbb{E}\Big[\sup_{\gamma_1(g)\leqslant D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)\Big] \leqslant \frac{DR}{\sqrt{n}}\rho_\varphi, \tag{10.15}$$

for a universal constant $\rho_\varphi > 0$.

**Exercise 10.7** *Given a metric space $\mathcal{X}$ with distance $d$ and finite diameter, we consider $\varphi(w) = \sigma(d(x,w))$, for $w \in \mathcal{W} = \mathcal{X}$. Compute an upper-bound on the Rademacher complexity in Eq. (10.14).*

**Generalization bound.** We can now state our main statistical result about gradient boosting. To obtain such bounds, an additional norm (see, e.g., Lugosi and Vayatis, 2004) or cardinality (Barron et al., 2008) constraint is often added. In this section, we show how early stopping is enough to obtain rates of convergences for the gradient-boosting procedures defined in Section 10.3.5.

**Proposition 10.2** *Assume the feature maps $\varphi$ form a centrally symmetric set and that they are uniformly bounded by $R$, and satisfy Eq. (10.15). Assume the loss function $\ell$ is non-negative, $G_2$-smooth and $G_1$-Lipschitz-continuous with respect to the second variable, and that $\ell(y,0) \leqslant G_0$ almost surely. If $g_t$ denotes the $t$-th iterate of the gradient boosting procedure, then, for any function $f : \mathcal{X} \to \mathbb{R}$,*

$$\mathbb{E}\big[\mathcal{R}(g_t)\big] \leqslant \mathcal{R}(f) + \Big[\sqrt{2t}\frac{G_0^{1/2}}{G_2^{1/2}} + R\gamma_1(f)\Big] \cdot 2G_1 \cdot \frac{\rho_\varphi}{\sqrt{n}} + \frac{(R\gamma_1(f))^{2/5}}{t^{1/5}}(2G_2G_0^4)^{1/5}. \quad (10.16)$$

**Proof** For any function $f$ such that $\gamma_1(f)$ is finite, we have:

$$\begin{aligned}\mathcal{R}(g_t) - \mathcal{R}(f) &= \mathcal{R}(g_t) - \widehat{\mathcal{R}}(g_t) + \widehat{\mathcal{R}}(g_t) - \widehat{\mathcal{R}}(f) + \widehat{\mathcal{R}}(f) - \mathcal{R}(f) \\ &\leqslant \sup_{\gamma_1(g) \leqslant \gamma_1(g_t)} \big\{\mathcal{R}(g) - \widehat{\mathcal{R}}(g)\big\} + \sup_{\gamma_1(g) \leqslant \gamma_1(f)} \big\{\widehat{\mathcal{R}}(g) - \mathcal{R}(g)\big\} + \widehat{\mathcal{R}}(g_t) - \widehat{\mathcal{R}}(f).\end{aligned}$$

We then apply Proposition 10.1 with $C = R\sqrt{n}$ and $L = G_2/n$, with $F(0) \leqslant G_0$. This leads to $\gamma_1(g_t) \leqslant \frac{\sqrt{2t}}{\sqrt{G_2R^2}}G_0^{1/2}$, and $\widehat{\mathcal{R}}(g_t) - \widehat{\mathcal{R}}(f) \leqslant \big(\frac{2G_2R^2\gamma_1(f)^2G_0^4}{t}\big)^{1/5}$. Thus, using properties of Rademacher averages from Section 4.5,

$$\mathbb{E}\big[\mathcal{R}(g_t) - \mathcal{R}(f)\big] \leqslant \Big[\frac{\sqrt{2t}}{\sqrt{G_2R^2}}G_0^{1/2} + \gamma_1(f)\Big] \cdot 2G_1 \cdot \frac{\rho_\varphi R}{\sqrt{n}} + \Big(\frac{2G_2R^2\gamma_1(f)^2G_0^4}{t}\Big)^{1/5},$$

which leads to the desired result. ■

Up to constants, the bound in Eq. (10.16) is of the form $\mathcal{R}(f) + \frac{\sqrt{t}}{\sqrt{n}} + \frac{R\gamma_1(f)}{\sqrt{n}} \cdot \rho_\varphi + \frac{(R\gamma_1(f))^{2/5}}{t^{1/5}}$. We can optimize with respect to the number $t$ of iterations, and if we take it of order $t \sim n^{5/7}(R\gamma_1(f))^{4/7}$, then this leads to $\mathcal{R}(f) + \frac{R\gamma_1(f)}{\sqrt{n}} \cdot \rho_\varphi + \big(\frac{R\gamma_1(f)}{\sqrt{n}}\big)^{2/7}$.

Assuming for simplicity that $\rho_\varphi$ is a constant (like for neural networks), the dominant term is $\mathcal{R}(f) + \big(\frac{R\gamma_1(f)}{\sqrt{n}}\big)^{2/7}$. If the Bayes predictor $f_*$ is such that $\gamma_1(f^*)$ is finite, we immediately get an excess risk that goes to zero as $\big(\frac{R\gamma_1(f_*)}{\sqrt{n}}\big)^{2/7}$. If the model we consider is misspecified, then, like in Section 7.5.1 for kernel methods, and Section 9.4 for neural networks, we could compute the resulting approximation error to obtain precise rates depending on properties of the Bayes predictor $f^*$.

**Comparison with explicit constraint on $\gamma_1$.** The bound above is obtained by early-stopping the boosting algorithms before they overfit. An alternative method is to minimize the empirical risk subject to the constraint $\gamma_1(f) \leqslant D$, which can be done with the Frank-Wolfe algorithm described in Section 9.3.6, with the same access to the weak-learner oracle and an optimization error proportional to $R^2D^2/t$ after $t$ iterations. Together with the estimation error in $\rho_\varphi \frac{RD}{\sqrt{n}}$, we can take $t = RDn^{1/2}$ steps of the Frank-Wolfe algorithms to get an excess risk less than $\mathcal{R}(f)$ plus a constant times $\frac{\rho_\varphi RD}{\sqrt{n}}$ for any $f$ such that $\gamma_1(f) \leqslant D$. With the optimal choice of $D$, this leads to $\mathcal{R}(f)$ plus a constant times $\frac{\rho_\varphi R\gamma_1(f)}{\sqrt{n}}$, which is significantly better than for boosting. This, however, requires to set the constant $D$.

**Comparison with early stopping for gradient descent.** Compared to Section 5.3, where we used gradient descent, and the rates obtained for early stopping were the same as for constrained optimization, our analysis of boosting leads to consistent estimation, but with slightly worse rates.

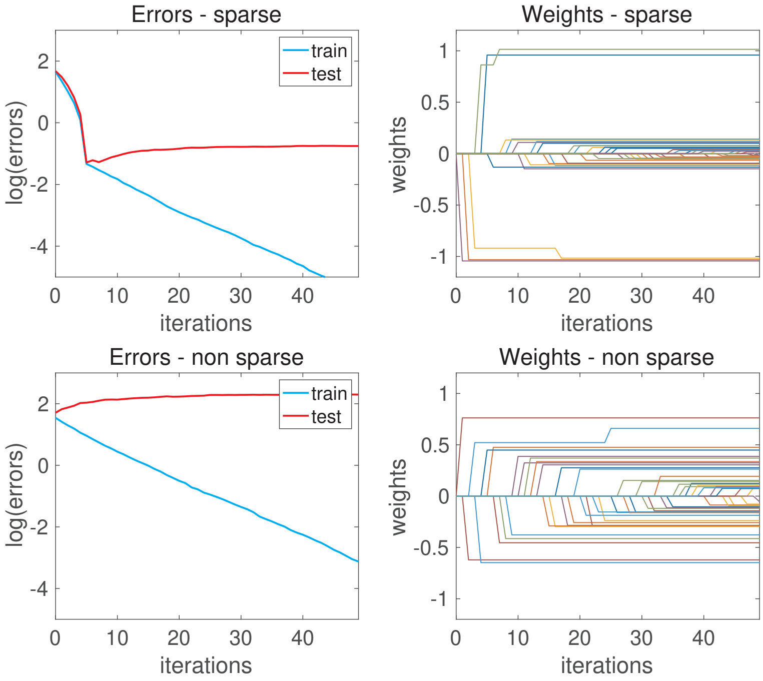


Figure 10.3: Matching pursuit algorithm on a problem with a sparse solution (top) and a non-sparse solution (bottom). Left: plots of training and testing errors. Right: plots of weights.

### 10.3.7 Experiments

In this section, we compare the boosting / conditional gradient algorithm described earlier on a simple linear regression task with feature selection. This corresponds to using $F(u) = \frac{1}{2n}\|y-u\|_2^2$, which is $(1/n)$-smooth and $(1/n)$-strongly convex, and $\gamma(u) = \inf_{w\in\mathbb{R}^d} \|w\|_1$ such that $u = \Phi w$.

We consider $n = 100$ observations in dimension $d = 1000$, sampled from a standard Gaussian random vector. A predictor $\beta_*$ with $k = 5$ non-zero values in $\{-1, 1\}$ and data are generated from a linear model with Gaussian noise. We then compare the iterates of the boosting algorithm in terms of prediction errors (left plots) and variations of weights across iterations (right plots).

We also consider a rotation of the data so that this is no longer a sparse problem (bottom plot). We observe linear convergence of the training errors, as proved above, but with overfitting at convergence, strong for the non-sparse case and weak for the sparse case.

## 10.4 Conclusion

In this chapter, we have presented a brief overview of ensemble learning procedures, which rely on using the same "base" learning procedures on several datasets. Bagging procedures consider several often parallel and independent runs on randomly modified datasets, while boosting changes the weight on each observation sequentially. Moreover, boosting is an instance of computational regularization, where overfitting is avoided by early stopping an optimization algorithm that would converge to a minimizer of the empirical risks if not stopped. The implicit bias in boosting is that of an $\ell_1$-norm; in Section 12.1, we analyze the implicit bias of gradient decent, when run to convergence, with a link to $\ell_2$-penalties.

# Chapter 11

# From online learning to bandits

**Chapter summary**

- Beyond empirical and expected risk minimization, more complex settings can be considered.
- Online convex optimization with gradients: stochastic gradient descent still works with the regret criterion and potentially adversarial functions, with essentially the same rates. The mirror descent framework is adapted to non-Euclidean geometries.
- Zero-th order optimization: Randomization can be used to obtain a stochastic gradient from function values with an additional dimension dependence.
- Multi-armed bandits: in the regret minimization framework, to tackle exploration/exploitation trade-offs, several algorithms can be used, from simple algorithms based on alternating exploration and exploitation to more refined ones utilizing the principle of "optimism in the face of uncertainty."

In traditional stochastic optimization such as presented in Chapter 5 (e.g., Section 5.4), we observe a sequence of gradients of loss functions obtained from a pair of observations $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$:

$$F_t'(\theta_{t-1}) = \frac{\partial \ell(y_t, f_\theta(x_t))}{\partial \theta}\Big|_{\theta=\theta_{t-1}},$$

and our performance measure was

$$\mathbb{E}\big[F(\theta_t)\big] - F_*,$$

where the expectation is taken with respect to the training data, and $F(\theta) = \mathbb{E}[\ell(y_s, f_\theta(x_s))]$ is the expected test error, assuming that all $(x_s, y_s)$ (and thus the individual loss functions $F_s(\theta) = \ell(y_s, f_\theta(x_s)))$, $s = 1, \dots, t$, are independent and identically distributed, and $F_*$ is the minimal value of $F$, that is, $F_* = \inf_{\theta \in \mathcal{C}} F(\theta)$, where $\mathcal{C}$ is the optimization domain.

There are several important extensions corresponding to specific applications:

- **Regret instead of final performance:** The performance criterion can take into account performance along iterations such as $\frac{1}{t}\sum_{s=1}^{t} F(\theta_{s-1})$, and not only at the last iteration, that is $F(\theta_t)$. This is important when the loss functions can be interpreted as actual financial losses incurred while learning the parameter $\theta$ (such as in applications in advertising or finance).

  Performance measures such as the *regret* can then be considered, here equal to

  $$\frac{1}{t}\sum_{s=1}^{t} F(\theta_{s-1}) - \inf_{\theta \in \mathcal{C}} F(\theta),$$

  often after taking the expectation (since $\theta_s$ is random because it depends on the past data).

  ⚠ In this book, we choose to study what is often called the *normalized* regret since we divide $\sum_{s=1}^{t} \big[F(\theta_s) - \inf_{\theta \in \mathcal{C}} F(\theta)\big]$ by $t$. This is done to make comparisons with the usual stochastic framework easier.

- **Adversarial instead of stochastic:** The consideration of the regret criterion opens up the possibility for functions $F_s$ to be different or sampled from different distributions, with a potentially adversarial choice that depends on the past. The regret is then $\frac{1}{t}\sum_{s=1}^{t} F_s(\theta_{s-1}) - \inf_{\theta \in \mathcal{C}} \frac{1}{t}\sum_{s=1}^{k} F_s(\theta)$, which is the comparison to the optimal constant prediction. This allows it to be robust to adversarial functions and adapted to non-stationary environments where very few assumptions can be made. Note here that the regret can be negative. This is presented in Section 11.1.

- **Partial feedback (zero-th order):** Independently of the regret framework, the feedback given to the algorithm may be less precise than the full gradient (e.g., only the function value). This is crucial in applications where function values are expensive to obtain without access to gradients.

  This is the domain of zero-th order optimization, which can be treated through gradient-based algorithms (Section 11.2) or the framework of multi-armed bandits (Section 11.3).

In this chapter, we briefly cover three topics from this large body of literature. For more details, see Shalev-Shwartz (2011); Bubeck and Cesa-Bianchi (2012); Hazan (2022); Slivkins (2019); Lattimore and Szepesvári (2020). This chapter aims to give the main ideas, how they differ from classical learning theory (using the unified notations we provide in this book), and to encourage readers to study these references.

## 11.1 First-order online convex optimization

In this section, we consider a sequence of arbitrary deterministic real-valued convex functions $F_s : \mathbb{R}^d \to \mathbb{R}$, $s \geqslant 1$, and a compact convex set $\mathcal{C}$. The goal of online convex optimization is, starting from a certain $\theta_0 \in \mathcal{C}$, to obtain a sequence $(\theta_s)_{s \geqslant 1}$ so that the regret at time $t$, defined as

$$\frac{1}{t}\sum_{s=1}^{t} F_s(\theta_{s-1}) - \inf_{\theta\in\mathcal{C}} \frac{1}{t}\sum_{s=1}^{t} F_s(\theta), \tag{11.1}$$

is as small as possible.

We assume that at time $s$, we can access a gradient of $F_s$ at any point $\theta_{s-1} \in \mathcal{C}$ that depends on past information. We also consider the possibility that we only observe a random, unbiased version $g_s$, that is, if $\mathcal{F}_s$ denotes the information up to (and including) time $s$,

$$\mathbb{E}\big[g_s|\mathcal{F}_{s-1}\big] = F_s'(\theta_{s-1}).$$

Given the added randomness, we consider the expected regret as a criterion.

For simplicity, we assume that almost surely, $\|g_s\|_2^2 \leqslant B^2$ (which in the context of machine learning corresponds to Lipschitz-continuous loss functions, which include the logistic loss, the hinge loss, and the square loss since we have assumed that we optimize on a bounded set[1]).

**Applications.** This is adapted to a non-stationary environment, where the data distribution varies over time, either stochastically or even adversarially (based on earlier predictions).

In this section, we only present the non-smooth case. The smooth case will be proposed as exercises but leads to similar results compared to the regular stochastic case.

### 11.1.1 Convex case

We consider the projected stochastic gradient descent recursion:

$$\theta_s = \Pi_{\mathcal{C}}(\theta_{s-1} - \gamma_s g_s),$$

for a certain positive step-size $\gamma_s$ (which we assume deterministic for simplicity), where $\Pi_{\mathcal{C}}$ is the orthogonal projection onto the set $\mathcal{C}$. We then have, for any $\theta \in \mathcal{C}$ (as opposed to a fixed $\theta = \eta_*$ the global optimum, like in regular optimization in Chapter 5),

$$\begin{aligned}
\|\theta_s - \theta\|_2^2 &\leqslant \|\theta_{s-1}-\theta\|_2^2 - 2\gamma_s g_s^\top(\theta_{s-1}-\theta) + \gamma_s^2 B^2 \text{ by contractivity of projections,}\\
\mathbb{E}\big[\|\theta_s-\theta\|_2^2|\mathcal{F}_{s-1}\big) &\leqslant \|\theta_{s-1}-\theta\|_2^2 - 2\gamma_s F_s'(\theta_{s-1})^\top(\theta_{s-1}-\theta) + \gamma_s^2 B^2,\\
&\qquad\qquad \text{using the unbiasedness of the gradient,}\\
&\leqslant \|\theta_{s-1}-\theta\|_2^2 - 2\gamma_s\big[F_s(\theta_{s-1}) - F_s(\theta)\big] + \gamma_s^2 B^2, \text{ using convexity.}
\end{aligned}$$

Taking full expectations and isolating $F_s(\theta_{s-1}) - F_s(\theta)$, we get:

$$\mathbb{E}\big[F_s(\theta_{s-1}) - F_s(\theta)\big] \leqslant \frac{1}{2\gamma_s}\Big(\mathbb{E}\big[\|\theta_{s-1}-\theta\|_2^2\big] - \mathbb{E}\big[\|\theta_s-\theta\|_2^2\big]\Big) + \frac{\gamma_s}{2}B^2.$$

[1] The square loss is not Lipschitz-continuous on an unbounded domain, but is, once constrained to a bounded domain.

We can then sum between $s = 1$ to $s = t$, to obtain

$$\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big]-\frac{1}{t}\sum_{s=1}^{t}F_s(\theta) \leqslant \frac{1}{t}\sum_{s=1}^{t}\frac{1}{2\gamma_s}\Big(\mathbb{E}\big[\|\theta_{s-1}-\theta\|_2^2\big]-\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big]\Big)+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2.$$

At this point, the proof is exactly the same as the one of Prop. 5.7, with only the appearances of functions $F_s$ that depend on $s$.

In Chapter 5 (that is, the proof of Prop. 5.7), we considered non-uniform averaging, which is not adapted to the online setting (because the regret is based on a uniform average). We could also use a constant step-size that depends on the horizon $t$ (which then needs to be known in advance). By using Abel's summation formula (discrete integration by part), we can use a time-dependent step-size sequence $(\gamma_s)$, as, using the notation $\delta_s = \mathbb{E}\big[\|\theta_s - \theta\|_2^2\big]$, and for decreasing step-sizes:

$$\begin{aligned}
\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big]-\frac{1}{t}\sum_{s=1}^{t}F_s(\theta) &\leqslant \frac{1}{t}\sum_{s=1}^{t}\frac{1}{2\gamma_s}\big(\delta_{s-1}-\delta_s\big) + \frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2 \\
&\qquad\qquad \text{from the last equation,} \\
&= \frac{1}{t}\sum_{s=1}^{t-1}\delta_s\big(\frac{1}{2\gamma_{s+1}}-\frac{1}{2\gamma_s}\big)+\frac{\delta_0}{2t\gamma_1}-\frac{\delta_t}{2t\gamma_t}+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2 \\
&\qquad\qquad \text{using Abel's summation formula,} \\
&\leqslant \frac{1}{t}\sum_{s=1}^{t-1}\mathrm{diam}(\mathcal{C})^2\big(\frac{1}{2\gamma_{s+1}}-\frac{1}{2\gamma_s}\big)+\frac{\mathrm{diam}(\mathcal{C})^2}{2t\gamma_1}+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2 \\
&\qquad\qquad \text{using that } \delta_s \leqslant \mathrm{diam}(\mathcal{C})^2 \text{ for all } s, \\
&= \frac{\mathrm{diam}(\mathcal{C})^2}{2t\gamma_t}+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2.
\end{aligned}$$

By choosing $\gamma_s = \frac{\mathrm{diam}(\mathcal{C})}{B\sqrt{s}}$, we get, using the same inequalities as for the proof of Prop. 5.7:

$$\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big]-\frac{1}{t}\sum_{s=1}^{t}F_s(\theta) \leqslant \frac{3B\mathrm{diam}(\mathcal{C})}{2\sqrt{t}}. \tag{11.2}$$

This is exactly the expected regret and essentially the same bound as stochastic optimization in Section 5.4. Note that from such a bound, if all $F_s$'s are equal, we can do an "online-to-batch" conversion using Jensen's inequality and exactly get the bound for regular projected stochastic gradient descent (which is no surprise, as the proof is essentially the same).

We show in Section 11.1.4 that the rate in Eq. (11.2) is, up to constants, the best possible over all Lipschitz-continuous functions over a compact set.

**Exercise 11.1** *(♦) In the unconstrained online optimization with smooth functions, that is, assuming that each $F_t$ is $L$-smooth, and $\mathcal{C} = \mathbb{R}^d$, provide a regret bound for online gradient descent.*

### 11.1.2 Strongly-convex case (♦)

Assuming strong-convexity (e.g., by adding $\frac{\mu}{2}\|\theta\|_2^2$ to the objective function), we will get a rate proportional to $B^2\log(k)/(\mu k)$. Indeed, assuming that the functions $F_s$ are all $\mu$-strongly-convex on $\mathcal{C}$. We can indeed modify the proof above with the step-size $\gamma_s = 1/(\mu s)$, to get (with modifications in red):

$$\begin{aligned}
\|\theta_s-\theta\|_2^2 &\leqslant \|\theta_{s-1}-\theta\|_2^2 - 2\gamma_s g_s^\top(\theta_{s-1}-\theta) + \gamma_s^2B^2 \\
\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big|\mathcal{F}_{s-1}\big) &\leqslant \|\theta_{s-1}-\theta\|_2^2 - 2\gamma_s F_s'(\theta_{s-1})^\top(\theta_{s-1}-\theta) + \gamma_s^2B^2 \\
&\leqslant \|\theta_{s-1}-\theta\|_2^2 - 2\gamma_s\big[F_s(\theta_{s-1}) - F_s(\theta) + \frac{\mu}{2}\|\theta_{s-1}-\theta\|_2^2\big] + \gamma_s^2B^2.
\end{aligned}$$

Taking full expectations and isolating function values, we get:

$$\mathbb{E}\big[F_s(\theta_{s-1}) - F_s(\theta)\big] \leqslant \big(\frac{1}{2\gamma_s} - \frac{\mu}{2}\big)E\big[\|\theta_{s-1}-\theta\|_2^2\big] - \frac{1}{2\gamma_s}\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big]\Big) + \frac{\gamma_s}{2}B^2.$$

We can then use the specific form of step-size to get

$$\mathbb{E}\big[F_s(\theta_{s-1}) - F_s(\theta)\big] \leqslant \frac{\mu}{2}(s-1)E\big[\|\theta_{s-1}-\theta\|_2^2\big] - \frac{\mu}{2}s\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big] + \frac{1}{2\mu s}B^2.$$

Then, summing between $s=1$ to $s=t$, we obtain, with a telescoping sum:

$$\frac{1}{t}\sum_{s=1}^t \mathbb{E}\big[F_s(\theta_{s-1})\big] - \frac{1}{t}\sum_{s=1}^t F_s(\theta) \leqslant \frac{1}{t}\sum_{s=1}^t \frac{1}{2\mu s}B^2 \leqslant \frac{1}{2\mu t}(1+\log t),$$

using the classical $\log(t)$ upper bound on the harmonic series. Note the appearance of $\log(t)$, which would not be the case if we had used the step-size $\gamma_s = \frac{2}{s+1}$ like in Exercise 5.29 (but which would require a different averaging scheme with weights proportional to $s$). For online learning, it turns out that the logarithmic term is unavoidable (Hazan and Kale, 2014).

### 11.1.3 Online mirror descent (♦)

In this section, we extend the online stochastic gradient descent recursion analysis from Section 11.1.1 to the *online mirror descent* framework, which will apply to the regular stochastic case as well.

**Mirror map.** We assume given a "mirror map" $\Phi : \mathcal{C}_\Phi \to \mathbb{R}$, which is differentiable and $\mu$-strongly convex (on the set $\mathcal{C} \subset \mathcal{C}_\Phi$) with respect to a norm $\|\cdot\|$, that is,

$$\Phi(\eta) \geqslant \Phi(\theta) + \Phi'(\theta)^\top(\eta-\theta) + \frac{\mu}{2}\|\eta-\theta\|^2, \quad \forall \eta,\theta \in \mathcal{C}.$$

We also assume that the gradient $\Phi'$ is a bijection from $\mathcal{C}_\Phi$ to $\mathbb{R}^d$. Classical examples are:

- Squared Euclidean norm: $\Phi(\theta) = \frac{1}{2}\|\theta\|_2^2$ with full domain, and norm $\|\cdot\| = \|\cdot\|_2$, with $\mu = 1$.

- Entropy: $\Phi(\theta) = \sum_{i=1}^d \theta_i \log \theta_i$ with domain $\mathcal{C}_\Phi = (\mathbb{R}_+^*)^d$, and norm $\|\cdot\| = \|\cdot\|_1$, with $\mu = 1$ (a result which is equivalent to Pinsker inequality[2]).
- Squared $\ell_p$-norms: $\Phi(\theta) = \frac{1}{2}\|\theta\|_p^2$ with full domain, for $p \in (1,2]$, and norm $\|\cdot\| = \|\cdot\|_p$, with $\mu = p-1$ (see proof of strong-convexity by Ball et al., 2002).

See also Exercise 13.2 in Chapter 13 for an example of mirror map for matrices.

**Online mirror descent.** We consider the same set-up as the beginning of Section 11.1.1, that is, we have convex Lipschitz-continuous functions $F_s$, for $s \geqslant 1$, and we access an unbiased (sub)gradient $g_s$, that is, if $\mathcal{F}_s$ denotes the information up to (and including) time $s$,

$$\mathbb{E}\big[g_s|\mathcal{F}_{s-1}\big] = F_s'(\theta_{s-1}).$$

The online mirror descent iteration is defined by:

$$\theta_t = \arg\min_{\theta\in\mathcal{C}} \ g_t^\top(\theta - \theta_{t-1}) + \frac{1}{\gamma} D_\Phi(\theta, \theta_{t-1}), \tag{11.3}$$

where $\mathcal{C}$ is a compact convex set, $D_\Phi(\theta,\eta) = \Phi(\theta) - \Phi(\eta) - \Phi'(\eta)^\top(\theta-\eta)$ is the Bregman divergence associated with the mirror map $\Phi$, and $\gamma$ is a step-size. If $\mathcal{C} \subset \mathcal{C}_\Phi$, then the update is simply defined by $\Phi'(\theta_t) = \Phi(\theta_{t-1}) - \gamma g_t$.

**Proposition 11.1** *Given the mirror descent recursion in Eq. (11.3), assume that each stochastic gradient has bounded expected squared norm $\mathbb{E}[\|g_s\|_*^2|\mathcal{F}_{s-1}] \leqslant B$, for all $s \geqslant 1$. Then, for every $\theta \in \mathcal{C}$, we have:*

$$\frac{1}{t}\sum_{s=1}^t \mathbb{E}\big[F_s(\theta_{s-1}) - F_s(\theta)\big] \leqslant \frac{1}{\gamma t} D_\Phi(\theta, \theta_0) + \frac{B^2\gamma}{2\mu}.$$

**Proof** The proof follows the same structure as for online stochastic gradient descent in Section 11.1.1. From the optimality conditions of the update in Eq. (11.3), we have $(\theta - \theta_t)^\top\big(\gamma g_t + \Phi'(\theta_t) - \Phi'(\theta_{t-1})\big) \geqslant 0$ for all $\theta \in \mathcal{C}$. Given $\theta \in \mathcal{C}$, we have:

$$\begin{aligned}
& D_\Phi(\theta,\theta_t) - D_\Phi(\theta,\theta_{t-1}) \\
= \ & \Phi(\theta_{t-1}) + \Phi'(\theta_{t-1})^\top(\theta - \theta_{t-1}) - \Phi(\theta_t) - \Phi'(\theta_t)^\top(\theta - \theta_t) \\
= \ & \Phi(\theta_{t-1}) - \Phi(\theta_t) + \Phi'(\theta_{t-1})^\top(\theta_t - \theta_{t-1}) + (\Phi'(\theta_{t-1}) - \Phi'(\theta_t))^\top(\theta - \theta_t) \\
\leqslant \ & \Phi(\theta_{t-1}) - \Phi(\theta_t) + \Phi'(\theta_{t-1})^\top(\theta_t - \theta_{t-1}) + \gamma g_t^\top(\theta - \theta_t) \\
= \ & -D_\Phi(\theta_t,\theta_{t-1}) - \gamma g_t^\top(\theta_{t-1} - \theta) - \gamma g_t^\top(\theta_t - \theta_{t-1}) \\
\leqslant \ & -\frac{\mu}{2}\|\theta_t - \theta_{t-1}\|^2 - \gamma g_t^\top(\theta_{t-1} - \theta) + \gamma\|g_t\|_* \cdot \|\theta_t - \theta_{t-1}\| \leqslant \frac{\|g_t\|_*^2\gamma^2}{2\mu} - \gamma g_t^\top(\theta_{t-1} - \theta),
\end{aligned}$$

using the strong-convexity of $\Phi$ and the bound on gradients. By taking conditional expectation, we get:

$$\mathbb{E}\big[D_\Phi(\theta,\theta_t) - D_\Phi(\theta,\theta_{t-1})\big|\mathcal{F}_{t-1}\big] \leqslant \frac{B^2\gamma^2}{2\mu} - \gamma F_t'(\theta_{t-1})^\top(\theta_{t-1} - \theta). \tag{11.4}$$

[2] see https://en.wikipedia.org/wiki/Pinsker%27s_inequality.

This leads to the desired result by using a telescoping sum and the convexity property $F_t(\theta_{t-1}) - F_t(\theta) \leqslant F_t'(\theta_{t-1})^\top(\theta_{t-1} - \theta)$. ■

We can make the following observations:

- We can optimize for the step-size when the optimization horizon $t$ is known. Indeed, for $D^2 = 2\sup_{\theta,\theta'\in\mathcal{C}} D_\Phi(\theta,\theta')$, and the choice $\gamma = D\sqrt{\mu}/(B\sqrt{t})$, this leads to the regret bound $DB/\sqrt{\mu t}$. Alternatively, decaying step-sizes can be used like for regular SGD (which corresponds precisely to the feature map $\Phi = \frac{1}{2}\|\cdot\|_2^2$).
- With the online-to-batch conversion, we also get the same bound when all $F_t$'s are equal for the averaged iterate.
- A classical application is for the simplex $\mathcal{C} = \{\theta \in \mathbb{R}_+^d, \sum_{j=1}^d \theta_j = 1\}$, and the entropy feature map. The update becomes $\theta_t \propto \theta_{t-1} \circ \exp(-\gamma g_t)$ (where $\circ$ denotes the component-wise product), with then a normalization step to sum to one, which is a *multiplicative* update, and the regret bound is equal to $B\sqrt{2\log(d)}/\sqrt{t}$. This regret bound would be of order $\sqrt{d}$ (instead of $\sqrt{\log d}$) if the Euclidean feature map was used.

**Exercise 11.2 (Stochastic mirror descent for $\ell_1$-regularization)** *In the context of Prop. 11.1, we consider equal functions $F_t = F$, and assume that $\mathbb{E}[\|g_s\|_\infty^2|\mathcal{F}_{s-1}] \leqslant B^2$ for all $s \geqslant 1$, and that $\theta_0 = 0$. Show that using mirror descent with the mirror map $\Phi(\theta) = \frac{1}{2}\|\theta\|_p^2$ for $p \in (1,2]$, we get, for the average iterate $\bar\theta_t = \frac{1}{t}\sum_{s=1}^{t-1}\theta_s$, the bound $\mathbb{E}\big[F(\bar\theta_t)\big] \leqslant F(\theta) + \frac{1}{2\gamma t}\|\theta\|_1^2 + \frac{B^2 d^{2-2/p}\gamma}{2(p-1)}$. For $d \geqslant 2$, show that with $p = 1 + \frac{1}{\log d}$, the last term is less than $2B^2\gamma\log d$, and, if $\theta_*$ is the minimizer of $F$, with an appropriate choice of $\gamma$, $\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta_*) \leqslant \frac{2B\|\theta_*\|_1\sqrt{\log d}}{\sqrt{t}}$.*

### 11.1.4 Lower bounds (♦♦)

In order to prove a lower bound, following Abernethy et al. (2008), we consider the set $\mathcal{C} = \{\theta \in \mathbb{R}^d,\ \|\theta\|_\infty \leqslant 1\}$, and the linear (hence convex) function $F_t^{(\varepsilon)} : \mathbb{R}^d \to \mathbb{R}$ defined as $F_t^{(\varepsilon)}(\theta) = \varepsilon_t^\top\theta$, for $\varepsilon_s \in \{-1,1\}^d$ for all $s \in \{1,\dots,t\}$. The gradient vectors $g_t$ are then simply equal to $\varepsilon_t$. We here have the exact deterministic gradient, with constants $B = \sqrt{d}$ and $\operatorname{diam}(\mathcal{C}) = 2\sqrt{d}$.

To obtain a lower bound of performance, it suffices to show that for any sequence $(\theta_s)$,

$$\sup_{\varepsilon\in\mathcal{E}} \frac{1}{t}\sum_{s=1}^t F_s^{(\varepsilon)}(\theta_{s-1}) - \inf_{\theta\in\mathcal{C}} \frac{1}{t}\sum_{s=1}^t F_s^{(\varepsilon)}(\theta)$$

is lower-bounded for $\mathcal{E}$ a well-chosen set. As already used in proving lower bounds in Section 3.7 and in Chapter 15, this is lower-bounded by the expectation for any distribution on $\mathcal{E}$, which we take to be all independent Rademacher random variables (note that the algorithm is deterministic, with no noise in the gradients, but the problem itself is random).

The regret of any algorithm is $\frac{1}{t}\sum_{s=1}^t \varepsilon_s^\top \theta_{s-1}$, which has zero expectation because $\theta_{s-1}$ does not use the information of $\varepsilon_s$. Moreover, using that the $\ell_1$-norm is dual to the $\ell_\infty$-norm:

$$\inf_{\theta\in\mathcal{C}} \frac{1}{t}\sum_{s=1}^t \varepsilon_s^\top\theta \quad = \quad -\Big\|\frac{1}{t}\sum_{s=1}^t \varepsilon_s\Big\|_1 = -d\Big|\frac{1}{t}\sum_{s=1}^t (\varepsilon_s)_1\Big|.$$

Therefore, from the following lemma, the regret is greater than $d\big|\frac{1}{t}\sum_{s=1}^t(\varepsilon_s)_1\big| \geqslant d/(8\sqrt{t})$, which is equal to $B\mathrm{diam}(\mathcal{C})/(16\sqrt{t})$, a lower bound that matches the upper-bound from SGD from Eq. (11.2), up to a constant factor.

**Lemma 11.1 (Khintchine's inequality)** *Let $\eta \in \{-1,1\}^t$ be a vector of independent Rademacher random variables (with equal probabilities for $-1$ and $+1$) and $x \in \mathbb{R}^d$. Let $p \in [0,\infty)$. Then*

$$\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \leqslant B_p\|x\|_2, \tag{11.5}$$

*with $B_p = \big(p2^{p/2}\Gamma(p/2)\big)^{1/2}$, where $\Gamma$ is the Gamma function.*[3] *The bound $B_p$ is less than $3\sqrt{p}$ for $p \geqslant 1$ and $\frac{3}{2}\sqrt{p}$ for $p \geqslant 2$. Moreover, if $p \geqslant 2$, $\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \geqslant \|x\|_2$, and if $p \leqslant 2$, we have:*

$$\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \geqslant B_{4-p}^{2-p/2}\|x\|_2. \tag{11.6}$$

*We also have when $p \geqslant 1$, with $1/p + 1/q = 1$, $\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \geqslant B_q^{-1}\|x\|_2$.*

**Proof** (♦) We have, for $s = x^\top\eta$, and $p > 0$:

$$\mathbb{E}[|s|^p] = p\int_0^{+\infty} \lambda^{p-1}\mathbb{P}(|s| \geqslant \lambda)d\lambda,$$

(which can be checked using Fubini's theorem). We then compute directly:

$$\mathbb{E}[e^{ts}] = \prod_{i=1}^d \Big(\frac{1}{2}e^{ts} + \frac{1}{2}e^{-ts}\Big) = \prod_{i=1}^d \cosh(tx_i) \leqslant \exp(t^2\|x\|_2^2/2),$$

using that $\cosh\alpha \leqslant \exp(\alpha^2/2)$ for any $\alpha \in \mathbb{R}$. Thus, for $\lambda \geqslant 0$,

$$\begin{aligned}\mathbb{P}(|s| \geqslant \lambda) \quad &\leqslant \quad 2\mathbb{P}(s \geqslant \lambda) = 2\mathbb{P}(e^{ts} \geqslant e^{t\lambda}) \leqslant 2\inf_{t\geqslant 0} e^{-\lambda t}\mathbb{E}[e^{ts}] \text{ using Markov's inequality,}\\ &\leqslant \quad 2\inf_{t\geqslant 0} e^{-\lambda t}\exp(t^2\|x\|_2^2/2) = 2\exp(-\lambda^2/(2\|x\|_2^2)), \text{ with } t = \lambda/\|x\|_2.\end{aligned}$$

Thus, through the change of variable $\mu = \lambda/\|x\|_2$:

$$\mathbb{E}[|s|^p] \leqslant 2p\int_0^{+\infty} \lambda^{p-1}\exp(-\lambda^2/(2\|x\|_2^2))d\lambda = \|x\|_2^p \times 2p\int_0^{+\infty} \mu^{p-1}\exp(-\mu^2/2)d\mu.$$

[3] See https://en.wikipedia.org/wiki/Gamma_function.

Thus, for Eq. (11.5), we can take $B_p^p = 2p\int_0^{+\infty}\lambda^{p-1}e^{-\lambda^2/2}d\lambda = p2^{p/2}\int_0^{+\infty}u^{p/2-1}e^{-u}du = p2^{p/2}\Gamma(p/2)$, with the change of variable $u = \lambda^2/2$. Through Stirling formula $\Gamma(p/2)^{1/p} \sim \sqrt{p/(2e)}$, and thus $B_p \sim \sqrt{p/e}$, and one can then check the bound $B_p \leqslant 3\sqrt{p}$ for $p \geqslant 1$, and $B_p \leqslant \frac{3}{2}\sqrt{p}$ for $p \geqslant 2$.

Assuming $\|x\|_2 = 1$ without loss of generality, we have, using Hölder's inequality:

$$1 = \mathbb{E}[|x^\top\eta|^2] \leqslant \big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/p}\big(\mathbb{E}[|x^\top\eta|^q]\big)^{1/q},$$

which leads to the last lower bound.

Moreover, for $p \geqslant 2$, we have directly $\|x\|_2 \leqslant \big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p}$, and to prove Eq. (11.6), for $p \in [0,2]$, we have by Cauchy-Schwarz inequality:

$$\begin{aligned} 1 &= \mathbb{E}[|x^\top\eta|^2] = \mathbb{E}[|x^\top\eta|^{p/2}|x^\top\eta|^{2-p/2}] \leqslant \big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/2}\big(\mathbb{E}[|x^\top\eta|^{4-p}]\big)^{1/2} \\ &\leqslant \big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/2}B_{4-p}^{2-p/2}. \end{aligned}$$

The optimal constant in Eq. (11.5) is $B_p = 1$ for $p \in (0,2]$, and $B_p = \sqrt{2}(\Gamma(p/2+1/2)/\sqrt{\pi})^{1/p}$ if $p > 2$, while the optimal constant in Eq. (11.6) is $A_p = 1$ if $p \geqslant 1$, and $A_p = 2^{1/2-1/p}$ if $p < 1.847$.[4] ■

Thus, with the notations of the lemma above:

$$\big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/p} \geqslant B_{4-p}^{1-4/p}\|x\|_2.$$

This leads to $\mathbb{E}|x^\top\eta| \geqslant \|x\|_2 B_3^{-3} \geqslant \|x\|_2(3\cdot 2^{3/2}\Gamma(3/2))^{-1} \geqslant \|x\|_2/8$ for the lower bound for online learning.

**Exercise 11.3** *(♦) What would upper and lower bounds be if the regret criterion is replaced by $\mathbb{E}\big[\sum_{s=1}^t \alpha_s F_s(\theta_{s-1})\big] - \inf_{\theta\in\mathcal{C}}\frac{1}{t}\sum_{s=1}^t \alpha_s F_s(\theta)$ for an arbitrary sequence $(\alpha_s)$ of positive numbers?*

## 11.2 Zero-th order convex optimization

In this section, we consider the task of unconstrained minimization of a convex function $F$, given only access to function values, which is typically referred to as *zero-th order optimization* (since the function value is the zero-th order derivative of $F$, while the gradient is the vector of first-order derivatives).

If the function values are accessible with no noise and the function is smooth, then one can get a gradient by finite differences by defining the following estimate:

$$\hat{F}'(\theta) = \sum_{i=1}^d \frac{1}{\delta}\big[F(\theta+\delta e_i) - F(\theta)\big]e_i \in \mathbb{R}^d, \tag{11.7}$$

[4] See https://en.wikipedia.org/wiki/Khintchine_inequality.

where $(e_i)_{i\in\{1,\dots,d\}}$ is the canonical orthonormal basis of $\mathbb{R}^d$, with arbitrary precision when $\delta$ tends to zero. Indeed, using the smoothness inequality from Eq. (5.10):

$$\|\hat{F}'(\theta) - F'(\theta)\|_2^2 = \frac{1}{\delta^2}\sum_{i=1}^d \big[F(\theta+\delta e_i) - F(\theta) - F'(\theta)^\top \delta e_i\big]^2 \leqslant \frac{d}{\delta^2}\big(L\delta^2/2\big)^2 = \frac{dL^2\delta^2}{4}.$$

Therefore, assuming for simplicity that algorithms have infinite numerical precision at the expense of $d+1$ noiseless function evaluations (one at $\theta$, and $d$ at each $\theta+\delta e_i$), we can compute the exact gradient, and use gradient descent. Note also that for many functions, the gradient can be computed easily with automatic differentiation techniques (see, e.g., Baydin et al., 2018, and references therein). The problem is more interesting with noisy evaluations.

In this section, we first consider for simplicity the case where $f$ is convex and smooth (that is, essentially with bounded second-order derivatives) but only accessible with a stochastic first-order oracle (unbiased, with variance $\sigma^2$), for which, in Eq. (11.7), the noise in the function values explodes when $\delta$ goes to zero.

That is, we will consider the iteration

$$\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{\delta}\big(F(\theta_{t-1}+\delta z_t) + \zeta_t - F(\theta_{t-1}) - \zeta_t'\big)z_t\Big],$$

where $\zeta_t$ and $\zeta_t'$ are zero-mean random variables with variance $\sigma^2$, corresponding to the additive noise on the two function evaluations. By writing $\varepsilon_t = \zeta_t - \zeta_t'$, we get:

$$\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{\delta}\big(F(\theta_{t-1}+\delta z_t) - F(\theta_{t-1}) + \varepsilon_t\big)z_t\Big], \tag{11.8}$$

where $\varepsilon_t$ corresponds to the noise with the two function evaluations at $\theta_{t-1}$ and $\theta_{t-1}+\delta z_t$, thus of variance $2\sigma^2$, and $z_t$ is sampled from a distribution so that the mean is $\mathbb{E}[z_t] = 0$ and the covariance matrix is $\mathbb{E}[z_t z_t^\top] = I$.

There are two natural candidates: (1) $z$ a signed canonical basis vectors selected uniformly at random (that is, $\pm\sqrt{d}e_i$, with $i$ selected uniformly at random in $\{1,\dots,d\}$, and a factor $\sqrt{d}$ to obtain an identity covariance matrix), which corresponds to a single coordinate change like in Eq. (11.7), or (2) $z$ standard Gaussian vector (with mean zero and identity covariance matrix). We consider the second option, as this will lead to an interesting property relating the stochastic gradient estimate to the gradient of a modified function.

Note that if $F$ is defined as an expectation $F(\theta) = \mathbb{E}_\xi\big[f(\theta,\xi)\big]$, the stochasticity at time $t$ comes from a sample $\xi_t$. We then compute the function values $f(\theta,\xi_t)$ at *two* different points with the same $\xi_t$, and we can get an improved bound (see the end of Section 11.2.1).

The key in analyzing the iteration in Eq. (11.8) is to study $g = \frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta)\big)z$, for a certain $\theta$ and $z$ and a standard Gaussian vector.

For $\delta$ small, a simple Taylor expansion around $\theta$ leads to

$$g = \frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta)\big)z = \frac{1}{\delta}\big(\delta z^\top F'(\theta) + O(\delta^2)\big)z = zz^\top F'(\theta) + O(\delta).$$

Thus, by taking an expectation with respect to $z$, we get $\mathbb{E}[g] = F'(\theta) + O(\delta)$, that is, we have an almost unbiased gradient (for $\delta$ small), and we can thus expect to use stochastic gradient techniques. It turns out that the analysis will be made even simpler through integration by parts and the property of the Gaussian distribution.

In terms of variance linked to noisy evaluations, the term $\frac{1}{\delta}\varepsilon_t z_t$ has zero mean, but its squared norm has expectation $\mathbb{E}\big[\big\|\frac{1}{\delta}\varepsilon_t z_t\big\|_2^2\big] = \frac{1}{\delta^2}2\sigma^2 d$. Thus, it explodes when $\delta$ goes to zero, thus leading to some trade-offs we now look at.

### 11.2.1 Smooth stochastic gradient descent

For simplicity, we consider an $L$-smooth function $F$ defined on $\mathbb{R}^d$ (see next section for the non-smooth version).

An important tool will be to consider the function $F_\delta : \mathbb{R}^d \to \mathbb{R}$ defined as

$$F_\delta(\theta) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[F(\theta+\delta z)], \tag{11.9}$$

which is the expectation of $F$ taken at point distributed as a Gaussian with mean $\theta$ and covariance matrix $\delta^2 I$.

**Approximation properties.** We can analyze the difference between $F$ and $F_\delta$ when $F$ is $L$-smooth:

$$\forall \theta \in \mathbb{R}^d,\ F_\delta(\theta) - F(\theta) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[F(\theta+\delta z) - F(\theta) - \delta F'(\theta)^\top z\big].$$

Thus, using Jensen's inequality, we get $F_\delta(\theta) \geqslant F(\theta)$ and using the smoothness bound from Eq. (5.10), we get:

$$\forall \theta \in \mathbb{R}^d,\ 0 \leqslant F_\delta(\theta) - F(\theta) \leqslant \frac{L\delta^2}{2}\mathbb{E}_{z\sim\mathcal{N}(0,I)}[\|z\|_2^2] = \frac{L}{2}\delta^2 d. \tag{11.10}$$

Moreover, we can compute the expectation of the squared norm of the gradient estimate

$$\begin{aligned}
& \mathbb{E}\Big[\Big\|\frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta)\big)z\Big\|_2^2\Big] \\
\leqslant\ & 2\mathbb{E}\Big[\Big\|\frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta) - \delta F'(\theta)^\top z\big)z\Big\|_2^2\Big] + 2\mathbb{E}\big[\|zz^\top F'(\theta)\|_2^2\big] \\
\leqslant\ & 2\mathbb{E}\Big[\frac{L^2\delta^2}{4}\|z\|_2^6\Big] + 2F'(\theta)^\top\mathbb{E}\big[\|z\|_2^2 zz^\top\big]F'(\theta) \text{ using smoothness,} \\
=\ & \frac{L^2\delta^2}{2}d(d+2)(d+4) + 2\|F'(\theta)\|_2^2\cdot 3d \leqslant \frac{L^2\delta^2}{2}15d^3 + 6d\|F'(\theta)\|_2^2,
\end{aligned} \tag{11.11}$$

where we have used that $\|z\|_2^2$ is a chi-squared random variable and that we get in closed form $\mathbb{E}[\|z\|_2^6] = d(d+2)(d+4)$ and $\mathbb{E}\big[\|z\|_2^2 zz^\top\big] = 3dI$ (see the exercise below).

**Exercise 11.4** *Show that for a standard Gaussian vector $z \in \mathbb{R}^d$ (with zero mean and covariance matrix identity), then $\mathbb{E}[\|z\|_2^6] = d(d+2)(d+4)$ and $\mathbb{E}\big[\|z\|_2^2 zz^\top\big] = 3dI$.*

**Stochastic gradient descent.** We can now analyze gradient descent and take conditional expectations given the information $\mathcal{F}_{s-1}$ up to time $s-1$, and use the standard manipulations from Chapter 5, starting from:

$$\theta_s - \theta_* = \theta_{s-1} - \theta_* - \gamma \frac{1}{\delta}\big(F(\theta_{s-1} + \delta z_s) - F(\theta_{s-1})\big) z_s - \frac{\gamma}{\delta} \varepsilon_s z_s,$$

to get, by expanding the squared norm:

$$\begin{aligned}
& \mathbb{E}\big[\|\theta_s - \theta_*\|_2^2 | \mathcal{F}_{s-1}\big] \\
\leqslant\ & \|\theta_{s-1} - \theta_*\|_2^2 - 2\gamma F_\delta'(\theta_{s-1})^\top(\theta_{s-1} - \theta_*) \\
& \qquad\qquad + 2\gamma^2 \mathbb{E}\big[\|\frac{1}{\delta}\big(F(\theta_{s-1} + \delta z_s) - F(\theta_{s-1})\big) z_s\|_2^2 | \mathcal{F}_{s-1}\big] + 2\frac{\gamma^2}{\delta^2}\mathbb{E}[\varepsilon_s^2 \|z_s\|_2^2] \\
\leqslant\ & \|\theta_{s-1} - \theta_*\|_2^2 - 2\gamma F_\delta'(\theta_{s-1})^\top(\theta_{s-1} - \theta_*) \\
& \qquad + 2\gamma^2 \cdot \big[\frac{L^2\delta^2}{2} 15 d^3 + 6d\|F'(\theta_{s-1})\|_2^2\big] + 2\frac{\gamma^2}{\delta^2} \cdot 2d\sigma^2 \text{ using Eq. (11.11),} \\
\leqslant\ & \|\theta_{s-1} - \theta_*\|_2^2 - 2\gamma\big[F_\delta(\theta_{s-1}) - F_\delta(\theta_*)\big] \\
& \qquad + 15\gamma^2 L^2 \delta^2 d^3 + 24L\gamma^2 d\big[F(\theta_{s-1}) - F(\theta_*)\big] + 4d\frac{\gamma^2}{\delta^2}\sigma^2 \text{ using co-coercivity,} \\
\leqslant\ & \|\theta_{s-1} - \theta_*\|_2^2 - 2\gamma\big[F(\theta_{s-1}) - F(\theta_*)\big] + 2\gamma \cdot \frac{L}{2}\delta^2 d \\
& \qquad + 15\gamma^2 L^2 \delta^2 d^3 + 24L\gamma^2 d\big[F(\theta_{s-1}) - F(\theta_*)\big] + 4d\frac{\gamma^2}{\delta^2}\sigma^2, \text{ using Eq. (11.10).}
\end{aligned}$$

Thus, if $\gamma \leqslant \frac{1}{24dL}$, we have $24L\gamma^2 d \leqslant \gamma$, and we get:

$$\begin{aligned}
\mathbb{E}\big[\|\theta_s - \theta_*\|_2^2 | \mathcal{F}_{s-1}\big] \ \leqslant\ & \|\theta_{s-1} - \theta_*\|_2^2 - \gamma\big[F(\theta_{s-1}) - F(\theta_*)\big] + \gamma L \delta^2 d \\
& \qquad + \frac{15}{24}\gamma L \delta^2 d^2 + 4d\frac{\gamma^2}{\delta^2}\sigma^2 \\
\leqslant\ & \|\theta_{s-1} - \theta_*\|_2^2 - \gamma\big[F(\theta_{s-1}) - F(\theta_*)\big] + 2\gamma L \delta^2 d^2 + 4d\frac{\gamma^2}{\delta^2}\sigma^2,
\end{aligned}$$

leading to, taking full expectations:

$$\mathbb{E}\big[F(\theta_{s-1})\big] - F(\theta_*) \ \leqslant\ \frac{1}{\gamma}\Big(\mathbb{E}\big[\|\theta_{s-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_s - \theta_*\|_2^2\big]\Big) + 2L\delta^2 d^2 + 4d\frac{\gamma}{\delta^2}\sigma^2.$$

Summing from $s = 1$ to $s = t$, we get

$$\frac{1}{t}\sum_{s=1}^t \mathbb{E}\big[F(\theta_{s-1})\big] - F(\theta_*) \leqslant \frac{1}{\gamma t}\|\theta_0 - \theta_*\|_2^2 + 2L\delta^2 d^2 + 4d\frac{\gamma}{\delta^2}\sigma^2. \tag{11.12}$$

We can now analyze various situations depending on the presence or absence of noise (see empirical illustration in Figure 11.1):

- If $\sigma = 0$, then we can take $\delta$ as close to zero as possible and get the rate, with $\gamma = \frac{1}{24dL}$, for the average iterate $\bar{\theta}_t = \frac{1}{t}\sum_{s=1}^t \theta_{s-1}$, and using Jensen's inequality:

$$\mathbb{E}\big[F(\bar{\theta}_t)\big] - F(\theta_*) \leqslant \frac{24Ld}{t}\|\theta_0 - \theta_*\|_2^2. \tag{11.13}$$

  As suggested at the beginning of Section 11.2, we only lose a factor of $d$ compared to regular gradient descent in Section 5.2.4.

- If $\sigma > 0$, we can optimize over $\delta$ to get (assuming $\sigma$ is known), with the choice $\delta^4 = 2\gamma\sigma^2 L^{-1} d^{-1}$, $\mathbb{E}\big[F(\bar{\theta}_t)\big] - F(\theta_*) \leqslant \frac{1}{\gamma t}\|\theta_0 - \theta_*\|_2^2 + 2\sqrt{2}\cdot\gamma^{1/2}L^{1/2}\sigma d^{3/2}$. With the maximal allowed step-size $\gamma = \frac{1}{24dL}$, this leads to

$$\mathbb{E}\big[F(\bar{\theta}_t)\big] - F(\theta_*) \quad\leqslant\quad \frac{24Ld}{t}\|\theta_0 - \theta_*\|_2^2 + \sigma d.$$

  There is convergence only up to the noise level with a limiting bound $\sigma d$. We can also use a step-size $\gamma$ that depends on the horizon $t$, by taking $\gamma = \frac{1}{24Ld}t^{-2/3}$, leading to:

$$\mathbb{E}\big[F(\bar{\theta}_t)\big] - F(\theta_*) \quad\leqslant\quad \frac{d}{t^{1/3}}\big[24L\|\theta_0 - \theta_*\|_2^2 + \sigma\big].$$

  We not only lose a factor of $d$ in the bound, but the dependence in $t$ is worsened from $1/t$ to $1/t^{1/3}$. Note that (a) the natural rate for convex stochastic first-order methods is $O(1/\sqrt{t})$, and (b) the dependence in $\sigma$ could be improved if the noise level were known.

**Extensions.** We can also consider the case where we can do two function evaluations, where one can check that we can essentially remove the variance term in $d\frac{\gamma^2}{\delta^2}\sigma^2$ due to two noisy evaluations, removing in Eq. (11.12) the last term, and thus with improved behavior. For related lower bounds, see Duchi et al. (2015).

**Exercise 11.5** *When two function evaluations are available, compute optimal values of $\delta$ and $\gamma$ and provide an improved convergence rate.*

### 11.2.2 Stochastic smoothing (♦)

In this section, we consider the case where $F$ may not be smooth, which leads to considering the nice effect of randomized smoothing. This randomized smoothing can simply be explained by seeing $F_\delta$ as the convolution of the function $F$ by the density of the Gaussian distribution with mean zero and covariance matrix $\delta^2 I$. Since this density is infinitely differentiable, a continuous function will be turned into an infinitely differentiable function. One particular instance of this phenomenon is shown precisely below.

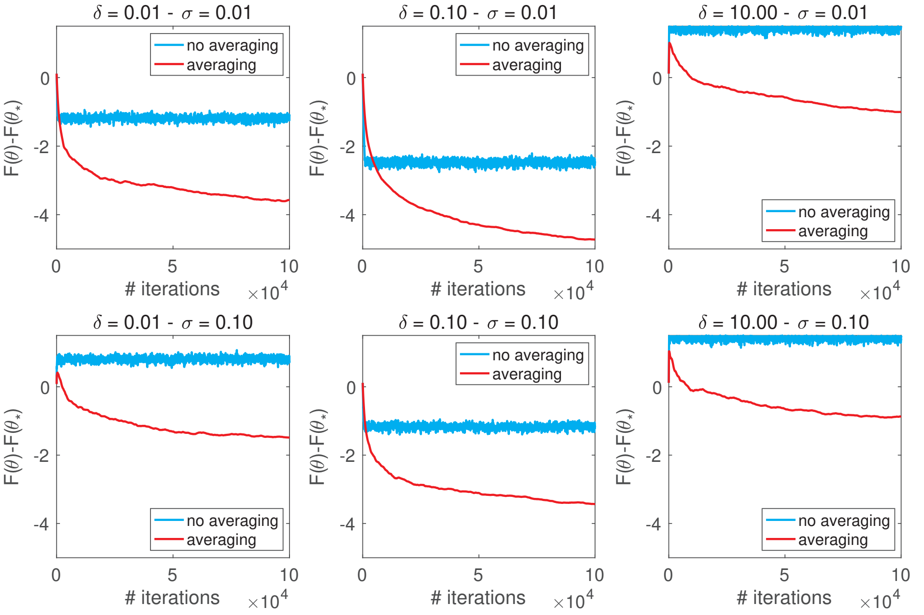
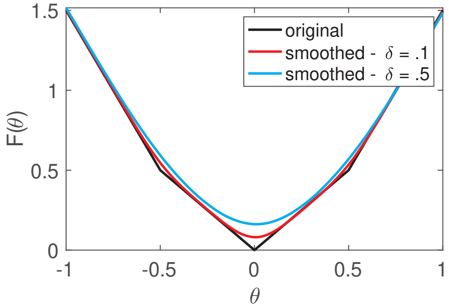


Figure 11.1: Zero-th order optimization with Gaussian smoothing on a quadratic function $F$ in dimension $d = 10$, with step-size $\gamma = 1/(4Ld)$: two different levels of noise added to the function values, $\sigma = 0.01$ (top), and $\sigma = 0.1$ (bottom), with three different smoothing constants, $\delta = 0.01$ (left), $\delta = 0.1$ (middle), and $\delta = 10$ (right). Performance improves with smaller noise variance $\sigma^2$, while $\delta$ should be chosen not too large (then too much bias) and not too small (too much variance).



**Proposition 11.2 (Randomized smoothing)** *Assume $F$ is $B$-Lipschitz-continuous. Then the function $F_\delta : \mathbb{R}^d \to \mathbb{R}$ defined in Eq. (11.9) is also $B$-Lipschitz-continuous. Morerover, it is $(\frac{\sqrt{d}}{\delta}B)$-smooth, with gradient equal to*

$$F'_\delta(\theta) = \frac{1}{\delta}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[F(\theta+\delta z)z\big] = \frac{1}{\delta}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[(F(\theta+\delta z)-F(\theta))z\big].$$

*Moreover, $\forall\theta\in\mathbb{R}^d,\ |F_\delta(\theta)-F(\theta)| \leqslant B\delta\sqrt{d}$.*

**Proof** If $F$ is $B$-Lipschitz-continuous, then for any $\theta,\theta'\in\mathbb{R}^d$, we have

$$\begin{aligned} |F_\delta(\theta)-F_\delta(\theta')| &= \big|\mathbb{E}\big[F(\theta+\delta z)-F(\theta'+\delta z)\big]\big| \leqslant \mathbb{E}\big[|F(\theta+\delta z)-F(\theta'+\delta z)|\big] \\ &\leqslant \mathbb{E}\big[B\|\theta-\theta'\|_2\big] = B\|\theta-\theta'\|_2, \end{aligned}$$

which shows Lipschitz-continuity of $F_\delta$. In terms of approximation, we have:

$$\forall\theta\in\mathbb{R}^d,\ |F_\delta(\theta)-F(\theta)| \leqslant \mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[|F(\theta+\delta z)-F(\theta)|\big] \leqslant B\delta\mathbb{E}_{z\sim\mathcal{N}(0,I)}[\|z\|_2] \leqslant B\delta\sqrt{d}.$$

We can now use the expression of the multivariate standard Gaussian density to get:

$$F_\delta(\theta) = \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} F(\theta+\delta\eta)\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)d\eta.$$

Then, assuming for simplicity that we can differentiate through the expectation, we get, by integration by parts:

$$\begin{aligned} F'_\delta(\theta) &= \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} F'(\theta+\delta\eta)\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)d\eta \\ &= \frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} \delta F'(\theta+\delta\eta)\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)d\eta \\ &= \frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} \frac{\partial F(\theta+\delta\eta)}{\partial\eta}\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)d\eta \\ &= -\frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} F(\theta+\delta\eta)\frac{\partial\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)}{\partial\eta}d\eta \text{ by integration by parts,} \\ &= -\frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} F(\theta+\delta\eta)\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)(-\eta)d\eta = \mathbb{E}\Big[\frac{1}{\delta}F(\theta+\delta z)z\Big]. \end{aligned}$$

That is, the gradient is equal to

$$F_\delta'(\theta) = \mathbb{E}\Big[\frac{1}{\delta}F(\theta+\delta z)z\Big] = \mathbb{E}\Big[\frac{1}{\delta}\big(F(\theta+\delta z)-F(\theta)\big)z\Big].$$

The function $F_\delta$ is $\big(\frac{\sqrt{d}}{\delta}B\big)$-smooth, since for $\theta,\theta'\in\mathbb{R}^d$,

$$\|F_\delta'(\theta)-F_\delta'(\theta')\|_2 \leqslant \frac{1}{\delta}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[|F(\theta+\delta z)-F(\theta'+\delta z)|\|z\|\big] \leqslant \frac{B}{\delta}\|\theta-\theta'\|_2\mathbb{E}_{z\sim\mathcal{N}(0,I)}[\|z\|_2].$$

■

In other words, the expectation of the gradient estimate happens to be exactly the gradient of a smoothed version $F_\delta$ of $F$. This will be used in the proof below. Moreover, the expression of $F_\delta'$ as an expectation leads naturally to the stochastic gradient $\hat{F}_\delta'(\theta) = \frac{1}{\delta}F(\theta+\delta z)z - \frac{1}{\delta}F(\theta)z$, for which we have: $\mathbb{E}[\hat{F}_\delta'(\theta)] = F_\delta'(\theta)$ and

$$\mathbb{E}\big[\|\hat{F}_\delta'(\theta)\|_2^2\big] \leqslant \mathbb{E}\big[B^2\|z\|_2^4\big] \leqslant 4B^2d^2.$$

**Stochastic gradient descent.** We have, for $\theta_*$ a minimizer of $F$ on $\mathbb{R}^d$, by expanding the square,

$$\begin{aligned}\|\theta_s-\theta_*\|_2^2 \;&=\; \|\theta_{s-1}-\theta_*\|_2^2 - 2\frac{\gamma}{\delta}\big(\big[F(\theta_{s-1}+\delta z_s)-F(\theta_{s-1})+\varepsilon_s\big]z_s\big)^\top(\theta_{s-1}-\theta_*)\\ &\qquad + \frac{\gamma^2}{\delta^2}\big\|\big[F(\theta_{s-1}+\delta z_s)-F(\theta_{s-1})+\varepsilon_s\big]z_s\big\|_2^2.\end{aligned}$$

We have, using the previous inequalities:

$$\mathbb{E}\big[\|\theta_s-\theta\|_2^2|\mathcal{F}_{s-1}\big] \;=\; \|\theta_{s-1}-\theta\|_2^2 - 2\gamma F_\delta'(\theta_{s-1})^\top(\theta_{s-1}-\theta) + 2\gamma^2\cdot 4B^2d^2 + 2\frac{\gamma^2}{\delta^2}\cdot\sigma^2 d,$$

leading to

$$\begin{aligned}F_\delta(\theta_{s-1})-F_\delta(\theta) \;&\leqslant\; \frac{1}{2\gamma}\Big(\mathbb{E}\big[\|\theta_{s-1}-\theta\|_2^2\big]-\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big]\Big) + 4\gamma B^2d^2 + \frac{\gamma}{\delta^2}\sigma^2 d\\ F(\theta_{s-1})-F(\theta) \;&\leqslant\; \frac{1}{2\gamma}\Big(\mathbb{E}\big[\|\theta_{s-1}-\theta\|_2^2\big]-\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big]\Big) + 4\gamma B^2d^2 + \frac{\gamma}{\delta^2}\sigma^2 d + 2B\delta\sqrt{d}.\end{aligned}$$

We thus get

$$\frac{1}{t}\sum_{s=1}^t F(\theta_{s-1})-F(\theta) \;\leqslant\; \frac{1}{2\gamma t}\|\theta_0-\theta\|_2^2 + 4\gamma B^2d^2 + \frac{\gamma}{\delta^2}\sigma^2 d + 2B\delta\sqrt{d}.$$

This leads to a similar discussion as for the smooth case in Section 11.2.1, for the choice of step-sizes:

- When $\sigma = 0$ (no noise in function evaluations), we can take $\delta$ as small as possible so that rounding errors do not perturb the finite differences, and we then only lose a factor of $d$ compared to the standard subgradient method studied in Section 5.3.

- When $\sigma > 0$, then we can optimize over $\delta$, with $\delta^3 = \gamma\sigma^2 B^{-1}\sqrt{d}$. We then get

$$\frac{1}{t}\sum_{s=1}^{t} F(\theta_{s-1}) - F(\theta) \quad \leqslant \quad \frac{1}{2\gamma t}\|\theta_0 - \theta\|_2^2 + 4\gamma B^2 d^2 + 3d^{2/3}\gamma^{1/3}\sigma^{2/3}B^{2/3}.$$

  To optimize the rate for large values of $t$, we can take $\gamma = \frac{1}{B^2 d^{1/2} t^{3/4}}$ for a final rate

$$\frac{d^{1/2}}{2t^{1/4}}\Big(B^2\|\theta_0 - \theta\|_2^2 + 6\sigma^{2/3}\Big) + 4\frac{d^{3/2}}{t^{3/4}}.$$

### 11.2.3 Extensions

In this section on zero-th order algorithms, we have focused on optimization algorithms with potentially stochastic noise, with a criterion which is the function values at the final time. This can be extended to online learning formulations with a different function $F_t$ at time $t$, then using the regret criterion in Eq. (11.1). Online zero-th order optimization is significantly more complicated, and in the next section, we will focus only on multi-armed bandits, which are optimization problems over finite sets and already lead to significant theoretical and practical developments. For more general cases, see Hazan (2022).

## 11.3 Multi-armed bandits

This section aims to provide the simplest results for multi-armed stochastic bandits. There is extensive and rich literature; see Bubeck and Cesa-Bianchi (2012); Lattimore and Szepesvári (2020); Slivkins (2019) for a more detailed account.

Multi-armed bandits are the simplest model of sequential decision problems where information is gathered as decisions are made and losses incurred, where the "exploration-exploitation" dilemma occurs. Beyond being a stepping stone for many more complex models, it directly applies to clinical trials or routing in networks.

We consider $k$ potential "arms" with associated means $\mu^{(1)}, \dots, \mu^{(k)} \in \mathbb{R}$. Every time we select the arm $i$, we receive a reward sampled independently of all other rewards and the previous arm choices from a sub-Gaussian distribution with mean $\mu^{(i)}$, and sub-Gaussian parameter $\sigma$. At time $s$, we select the arm $i_s$ based on the information $\mathcal{F}_{s-1}$ up to time $s-1$ (that is, the rewards received before time $s-1$) and receive the reward $r_s$. In this chapter, we focus on "plain" bandits, noting that many variations exist where limited feedback is given to the algorithm, in particular "contextual" bandits, where a feature vector is observed before each arm is selected and where rewards are unknown functions of the feature vectors.

**Criterion for reward maximization.** Our criterion is the expected regret (adapted to the *maximization* of rewards), equal to

$$R_t = t \cdot \max_{i\in\{1,\dots,k\}} \mu^{(i)} - \sum_{s=1}^{t} \mathbb{E}[r_s].$$

As opposed to online learning in the previous section, here we are not dividing the regret by $t$.

Denoting $\Delta^{(j)} = \max_{i\in\{1,\dots,k\}} \mu^{(i)} - \mu^{(j)} \geqslant$ the difference between the mean of the best arm and the mean of arm $j$, and $n_t^{(j)}$ the number of times the arm $j$ was selected in the first $t$ iterations, we can express the regret as

$$R_t = \sum_{j=1}^{k} \Delta^{(j)} \mathbb{E}\big[n_t^{(j)}\big]. \tag{11.14}$$

Thus, the regret is a direct function of the number of times each arm is selected. For all algorithms, we consider the natural unbiased estimate of the arm means at time $s$, that is,

$$\hat{\mu}_t^{(j)} = \frac{1}{n_t^{(j)}} \sum_{s=1}^{t} r_s 1_{i_s=j} = \frac{1}{n_t^{(j)}} \sum_{a=1}^{n_t^{(j)}} x_a^{(j)},$$

where we imagine we select rewards from a sequence of i.i.d. samples $x_a^{(i)}$ with mean $\mu^{(i)}$ from each arm. This implies that as we select some arms multiple times, we get a more accurate estimate of $\mu^{(i)}$ as the expected squared distance between $\hat{\mu}_t^{(j)}$ and $\mu^{(i)}$ is proportional to $1/n_t^{(j)}$. To simplify the exposition, we ignore the equality cases among the various estimated $\hat{\mu}_t^{(j)}$, which is safe as long as the distributions of the arm values are absolutely continuous with respect to the Lebesgue measure.

### 11.3.1 Need for an exploration-exploitation trade-off

We can now consider two extreme algorithms, highlighting the need to both "explore" and "exploit".

**Pure exploration.** If we select a random arm at each step, then, from Eq. (11.14) and $\mathbb{E}[n_t^{(j)}] = \frac{t}{k}$, the expected regret is $t \cdot \frac{1}{k}\sum_{j=1}^{k} \Delta^{(j)}$ and depends linearly in $t$, that is, we have a "linear regret". At time step $t$, we get a reasonable estimate of the best arm, but this incurs a strong loss along the iterations.

**Pure exploitation.** The previous strategy was ignoring the online estimates $\hat{\mu}_t^{(j)}$. The pure exploitation strategy does the opposite by only selecting the arm with the current largest estimate, assuming that the first $k$ steps are dedicated to selecting each arm only once. This has linear regret because there is a non-zero probability that the best arm will never be selected again.

**Exercise 11.6** *Provide a lower bound on the regret of the pure exploitation strategy.*

### 11.3.2 "Explore-then-commit"

If we consider $mk$ steps where we select exactly each arm $m$ times, we can build the $m$ estimates $\hat{\mu}^{(1)}, \dots, \hat{\mu}^{(k)}$, which are all independent random variables, with means $\mu^{(1)}, \dots, \mu^{(k)}$ and with sub-Gaussian parameters $\sigma^2/m$. Let $i_*$ be the optimal arm.

We then select the arm with maximal $\hat{\mu}_{mk}^{(j)}$ for all remaining $t - km$ steps. The regret for this algorithm is then equal to, using Eq. (11.14), for $t > mk$:

$$R_t \;=\; m \sum_{j=1}^{k} \Delta^{(j)} + (t - mk) \sum_{j=1}^{k} \Delta^{(j)} \mathbb{P}(\hat{\mu}_{mk}^{(j)} \geqslant \hat{\mu}_{mk}^{(i)}, \forall i \neq j),$$

where the first term corresponds to the first $m$ steps, for which this is the exact contribution of the regret, and the second term corresponds to the other $(t - mk)$ steps, where the arm $j$ is selected if $\hat{\mu}_{mk}^{(i)}$ is maximized for $i = j$.

We can now upper-bound the second term by only imposing that an arm $j$ is selected if $\hat{\mu}_{mk}^{(j)} \geqslant \hat{\mu}_{mk}^{(i_*)}$ (noting that $\Delta^{(i_*)} = 0$):

$$\begin{aligned} R_t &\leqslant m \sum_{j=1}^{k} \Delta^{(j)} + (t - mk) \sum_{j=1}^{k} \Delta^{(j)} \mathbb{P}(\hat{\mu}_{mk}^{(j)} \geqslant \hat{\mu}_{mk}^{(i_*)}) \\ &\leqslant m \sum_{j \neq i_*} \Delta^{(j)} + t \sum_{j \neq i_*} \Delta^{(j)} \exp\Big( - \frac{(\Delta^{(j)})^2 m}{4\sigma^2} \Big), \end{aligned}$$

by using sub-Gaussian tail bounds (see Section 1.2.1) on the difference of the $m$ arm values between $j$ and $i_*$ (that is, $\hat{\mu}_{mk}^{(i_*)} - \hat{\mu}_{mk}^{(j)}$ is a sub-Gaussian random variable with mean $\Delta^{(j)}$ and sub-Gaussianity parameter $2\sigma^2/m$).

**Two arms ($k = 2$).** For $k = 2$ arms, the upper-bound is, with $\Delta = \Delta^{(i)}$ for $i \neq i_*$:

$$m\Delta + t\Delta \exp\Big( - \frac{\Delta^2 m}{4\sigma^2} \Big),$$

and we can minimize approximately with respect to $m$ by taking the gradient with respect to $m$ (assuming for a moment it is not restricted to be an integer), leading to $\Delta = t \frac{\Delta^3}{4\sigma^2} \exp\big( - \frac{\Delta^2 m}{4\sigma^2} \big)$, that is, we consider the candidate $m^* = \big\lfloor \frac{4\sigma^2}{\Delta^2} \log \frac{\Delta^2 t}{4\sigma^2} \big\rfloor$.

If $t > \frac{4\sigma^2}{\Delta^2} \exp\big( \frac{4\sigma^2}{\Delta^2} \big)$, then $m^* \geqslant 1$, while it is always less than $t/2$. We then have a regret less than (using $\log \alpha \leqslant \alpha - 1$):

$$\begin{aligned} \frac{4\sigma^2}{\Delta} \log \frac{\Delta^2 t}{4\sigma^2} + t\Delta \exp\Big[ - \frac{\Delta^2}{4\sigma^2} \Big( \frac{4\sigma^2}{\Delta^2} \log \frac{\Delta^2 t}{4\sigma^2} - 1 \Big) \Big] &= \frac{4\sigma^2}{\Delta} \Big[ \exp(\Delta^2/(4\sigma^2)) + 2 \log \frac{\Delta\sqrt{t}}{2\sigma} \Big] \\ &\leqslant \frac{4\sigma^2}{\Delta} \Big( \exp(\Delta^2/(4\sigma^2)) - 2 + 2 \frac{\Delta\sqrt{t}}{2\sigma} \Big), \end{aligned}$$

which is less than a constant plus $4\sigma\sqrt{t}$. As shown below, this simple algorithm will achieve the lower bound (up to constant factors) for all possible algorithms. However, this requires knowing $\Delta$ and $t$ in advance to select $m^*$ appropriately.

**More than two arms ($k \geqslant 2$).** We consider the event $\mathcal{A} = \{\forall i \neq i_*, \hat{\mu}^{(i)} - \mu^{(i)} \leqslant \frac{r}{\sqrt{m}},\ \hat{\mu}^{(i_*)} - \mu^{(i_*)} \geqslant -\frac{r}{\sqrt{m}}\}$, where $r$ is a constant to be determined later. This event is true if suboptimal arms are not too overestimated while the optimal arm is not too underestimated. If the event $\mathcal{A}$ is true, then the loss in rewards for the last $t - mk$ steps is less than $2\frac{r}{\sqrt{m}}$ (since only arms with means that are less than $2\frac{r}{\sqrt{m}}$ away from the optimal one can be selected), while it is less than $\delta = \max_{i \neq i_*} \Delta^i$ otherwise. Moreover, using sub-Gaussian tail bounds and the union bound, $\mathbb{P}(\mathcal{A}^c) \leqslant k \exp(-\frac{r^2}{2\sigma^2})$.

Thus, the regret is less than

$$R_t \leqslant mk\delta + 2\frac{rt}{\sqrt{m}} + \delta kt \exp\Big(-\frac{r^2}{2\sigma^2}\Big),$$

where the first term corresponds to the explore phase and the last two terms to the commit phase.

With $m^{3/2} \approx rt/(k\delta)$, we can minimize the first two terms and get

$$R_t \leqslant 3(rt)^{2/3}(k\delta)^{1/3} + \delta kt \exp\Big(-\frac{r^2}{2\sigma^2}\Big).$$

With $r = \sigma\sqrt{2\log(kt)}$, we then get $R_t \leqslant \delta + 3t^{2/3}k^{1/3}\delta^{1/3}\sigma^{2/3}\big(2\log(kt)\big)^{1/3}$, which grows as $t^{2/3}$ and does not achieve the lower bound (see a better algorithm in Section 11.3.3).

**$\varepsilon$-greedy.** We can mix exploration and exploitation with the "$\varepsilon$-greedy" strategy, which will update estimates $\hat{\mu}^{(i)}$ but spread the exploration phase over iterations by selecting with some positive probability a random arm. The final regret is similar to explore-and-commit (Auer et al., 2002a).

### 11.3.3 Optimism in the face of uncertainty (♦)

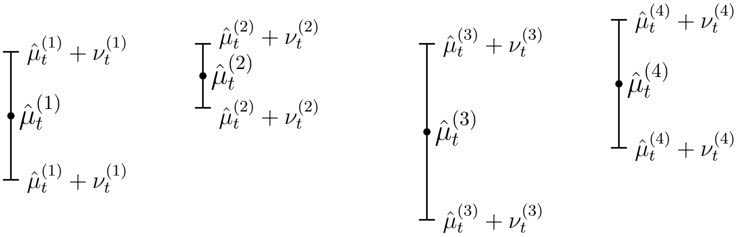
We consider the classical "upper confidence bound" (UCB) algorithm (Auer et al., 2002a), whose principle is simple. As arms are being selected, confidence intervals for the values of each arm are maintained as $[\hat{\mu}_t^{(i)} - \nu_t^{(i)}, \hat{\mu}^{(i)} + \nu_t^{(i)}]$. The arm that is selected is the one with maximal upper-confidence bound $\hat{\mu}_t^{(i)} + \nu_t^{(i)}$. This is one instance of the general principle of optimism in the face of uncertainty (Munos et al., 2014).

The precise algorithm is as follows (assuming that $\sigma$ is known):

- For the first $k$ rounds, select each arm exactly once, and form $\hat{\mu}_k^{(i)}$ as the reward received for arm $i$, with $\nu_k^{(i)} = \sqrt{2\rho\sigma^2\log(k)/n_k^{(i)}} = \sqrt{2\rho\sigma^2\log(k)}$, with $\rho > 0$ to be determined later.
- For all other $t > k$, select the arm $i_t$ which maximizes $\hat{\mu}_{t-1}^{(i)} + \nu_{t-1}^{(i)}$, receive the reward, and update, for all $i$, $\hat{\mu}_t^{(i)}$ as the average reward received for all arms $i \in \{1, \dots, k\}$, with the interval width $\nu_t^{(i)} = \sqrt{2\rho\sigma^2\log(t)/n_t^{(i)}}$.

The confidence interval lenght for arm $i$ is naturally proportional to $\sigma/\sqrt{n_t^{(i)}}$ with an extra factor that ensures sub-linear regret.

Thus, as illustrated below for $k=4$, we have $k$ confidence intervals, and we select the arm with the largest upper confidence bound (here $i=4$).


$\hat\mu_t^{(1)}+\nu_t^{(1)}$
$\hat\mu_t^{(1)}$
$\hat\mu_t^{(1)}+\nu_t^{(1)}$
$\hat\mu_t^{(2)}+\nu_t^{(2)}$
$\hat\mu_t^{(2)}$
$\hat\mu_t^{(2)}+\nu_t^{(2)}$
$\hat\mu_t^{(3)}+\nu_t^{(3)}$
$\hat\mu_t^{(3)}$
$\hat\mu_t^{(3)}+\nu_t^{(3)}$
$\hat\mu_t^{(4)}+\nu_t^{(4)}$
$\hat\mu_t^{(4)}$
$\hat\mu_t^{(4)}+\nu_t^{(4)}$

The analysis consists in upper-bounding $\mathbb{E}\big[n_t^{(i)}\big]$ for $i\neq i_*$ and using Eq. (11.14), that is, $R_t=\sum_{i\neq i_*}\Delta^{(i)}\mathbb{E}\big[n_t^{(i)}\big]$, to obtain the regret bound. We follow the proof technique from Garivier and Cappé (2011). For simplicity, we assume that there is a single arm $i_*$ with maximal mean.

The main idea of the proof is to compare the upper-confidence bounds to the optimal arm mean $\mu^{(i_*)}$. That is, for $i\neq i_*$, we have:

$$\begin{aligned}\mathbb{E}\big[n_t^{(i)}\big] &= \sum_{u=1}^t \mathbb{P}(i_u=i)\\ &= \sum_{u=1}^t \mathbb{P}(i_u=i,\hat\mu_{u-1}^{(i)}+\nu_{u-1}^{(i)}>\mu^{(i_*)})+\sum_{u=1}^t \mathbb{P}(i_u=i,\hat\mu_{u-1}^{(i)}+\nu_{u-1}^{(i)}\leqslant\mu^{(i_*)})\\ &\leqslant \sum_{u=1}^t \mathbb{P}\big(i_u=i,\hat\mu_{u-1}^{(i)}+\nu_{u-1}^{(i)}>\mu^{(i_*)}\big)+\sum_{u=1}^t \mathbb{P}\big(\hat\mu_{u-1}^{(i_*)}+\nu_{u-1}^{(i_*)}\leqslant\mu^{(i_*)}\big),\qquad(11.15)\end{aligned}$$

since if we select arm $i$ at time $u$ (i.e., $i_u=i$), then $\hat\mu_{u-1}^{(i_*)}+\nu_{u-1}^{(i_*)}\leqslant\hat\mu_{u-1}^{(i)}+\nu_{u-1}^{(i)}$.

In order to bound $\mathbb{P}\big(\hat\mu_{u-1}^{(i_*)}+\nu_{u-1}^{(i_*)}\leqslant\mu^{(i_*)}\big)$, it is tempting to apply a concentration inequality for the average of $n_{u-1}^{(i_*)}$ independent random variables distributed from the optimal arm distribution. However, these variables are not independent because the obtained reward and the choice of arms are not independent. Instead, we need to use our sequence of i.i.d. samples $x_a^{(i_*)}$, $a\geqslant 1$, with mean $\mu^{(i_*)}$, and bound the probability that *at least* one of these $u-1$ averages of i.i.d. random variables is less than the desired bound. Thus, we have, from sub-Gaussian tail bounds, for $u\in\{1,\dots,t\}$:

$$\mathbb{P}\big(\hat\mu_{u-1}^{(i_*)}+\nu_{u-1}^{(i_*)}\leqslant\mu^{(i_*)}\big)\leqslant\sum_{s=1}^{u-1}\exp(-\rho\log(s))\leqslant\frac{1}{u^{\rho-1}}\leqslant\frac{1}{t^{\rho-1}}.$$

Thus, the right term in Eq. (11.15) is less than $t^{2-\rho}$.

We now bound the left term in Eq. (11.15), as follows, for $t \geqslant k$:

$$\begin{aligned}
& \sum_{u=1}^{t} \mathbb{P}\big(i_u = i, \hat{\mu}^{(i)}_{u-1} + \nu^{(i)}_{u-1} > \mu^{(i_*)}\big) \\
= & \sum_{u=1}^{t} \mathbb{P}\Big(i_u = i, \hat{\mu}^{(i)}_{u-1} + \sqrt{2\rho\sigma^2 \log(u)/n^{(i)}_{u-1}} > \mu^{(i_*)}\Big) \\
\leqslant & \sum_{u=1}^{t} \mathbb{P}\Big(i_u = i, \hat{\mu}^{(i)}_{u-1} + \sqrt{2\rho\sigma^2 \log(t)/n^{(i)}_{u-1}} > \mu^{(i_*)}\Big) \text{ since } u \leqslant t, \\
= & \sum_{u=1}^{t} \sum_{s=1}^{u-1} \mathbb{P}\Big(i_u = i, n^{(i)}_{u-1} = s, \frac{1}{s}\sum_{a=1}^{s} x^{(i)}_a + \sqrt{2\rho\sigma^2 \log(t)/s} > \mu^{(i_*)}\Big).
\end{aligned}$$

We can now swap the two summations to get the bound

$$\begin{aligned}
& \sum_{s=1}^{t-1} \sum_{u=s+1}^{t} \mathbb{P}\Big(i_u = i, n^{(i)}_{u-1} = s, \frac{1}{s}\sum_{a=1}^{s} x^{(i)}_a + \sqrt{2\rho\sigma^2 \log(t)/s} > \mu^{(i_*)}\Big) \\
\leqslant & \sum_{s=1}^{t-1} \mathbb{P}\Big(\frac{1}{s}\sum_{a=1}^{s} x^{(i)}_a + \sqrt{2\rho\sigma^2 \log(t)/s} > \mu^{(i_*)}\Big)
\end{aligned}$$

because the events $\{i_u = i, n^{(i)}_{u-1} = s\}$, for $u \in \{s+1, \dots, t\}$ are mutually exclusive. We can then use sub-Gaussian tail bounds, which are non-trivial (that is, less than 1) as soon as $\Delta^{(i)} \geqslant \sqrt{2\rho\sigma^2 \log(t)/s}$, leading to a bound

$$\sum_{s=1}^{+\infty} \exp\big[-\big(\Delta^{(i)} - \sqrt{2\rho\sigma^2 \log(t)/s}\big)_+^2 s/(2\sigma^2)\big].$$

When $s \geqslant \frac{8\rho\sigma^2 \log(t)}{(\Delta^{(i)})^2}$, then the summand is less than $\exp\big[-\frac{s}{4}(\Delta^{(i)})^2/(2\sigma^2)\big]$, and that part of the sum is less than, with $\kappa = \frac{1}{4}(\Delta^{(i)})^2/(2\sigma^2)$:

$$\sum_{s=1}^{\infty} \exp(-s\kappa) = \frac{e^{-\kappa}}{1-e^{-\kappa}} = \frac{1}{e^{\kappa}-1} \leqslant \frac{1}{\kappa} = \frac{8\sigma^2}{(\Delta^{(i)})^2}.$$

Otherwise, we bound the probability by one, and get a term equal to $\frac{8\rho\sigma^2 \log(t)}{(\Delta^{(i)})^2}$. Thus, overall we get that

$$\mathbb{E}\big[n^{(i)}_t\big] \leqslant t^{2-\rho} + \frac{8\rho\sigma^2 \log(t)}{(\Delta^{(i)})^2} + \frac{8\sigma^2}{(\Delta^{(i)})^2}.$$

For $\rho = 2$, this leads to a regret bound

$$R_t \leqslant \sum_{i \neq i_*} \Delta^{(i)} \Big(1 + \frac{\sigma^2}{(\Delta^{(i)})^2}(16 \log t + 8)\Big),$$

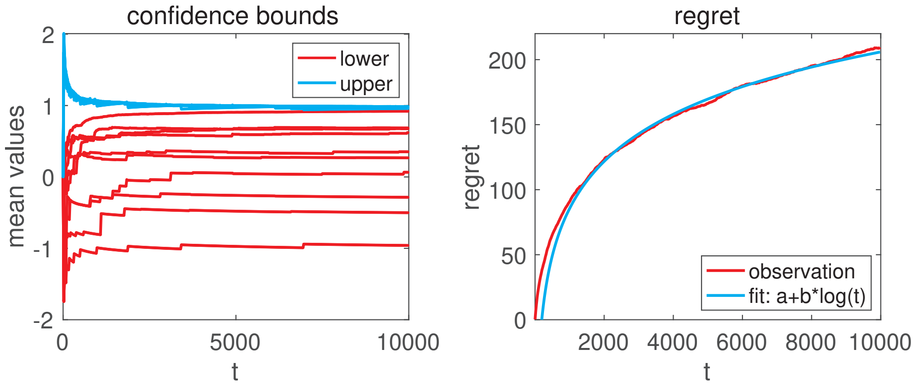


Figure 11.2: Upper-confidence bounds for $k = 10$ Bernoulli arms with random means: plot of upper and lower bounds as a function of time $t$ (left), and regret (right).

which happens to achieve the lower bound (up to constants, see below). We can also obtain a regret that does not blow up when $\Delta^{(i)}$ goes to zero. Indeed, we always have $\sum_{i=1}^k n_t^{(i)} \leqslant t$, leading to

$$\begin{aligned}
R_t &= \sum_{i,\ \Delta^{(i)}<\Delta} \Delta^{(i)}\mathbb{E}[n_t^{(i)}] + \sum_{i,\ \Delta^{(i)}\geqslant\Delta} \Delta^{(i)}\mathbb{E}[n_t^{(i)}] \text{ for a certain } \Delta, \\
&\leqslant t\Delta + \sum_{i,\ \Delta^{(i)}\geqslant\Delta} \Delta^{(i)}\Big(1 + \frac{\sigma^2}{(\Delta^{(i)})^2}(16\log t + 8)\Big) \\
&\leqslant t\Delta + \sum_i \Delta^{(i)} + k\frac{\sigma^2}{\Delta}(16\log t + 8) \leqslant \sum_i \Delta^{(i)} + 8\sigma\sqrt{kt(\log t + 2)},
\end{aligned}$$

by optimizing over $\Delta$, which is also optimal up to logarithmic terms (see below).

If $\rho \geqslant 2$, then we only pay an increase in the bound proportional to $\rho$, while if $\rho < 2$, the upper-bound on regret can start to be super-linear (and thus vacuous).

**Lower bounds.** It turns out that with $k$ arms, the best that can be achieved is a regret of order $\sigma\sqrt{kt}$, and for the instance-dependent problem, or order $\log(t)\sum_{i\neq i_*}\frac{\sigma^2}{\Delta^{(i)}}$ (see, e.g., Bubeck and Cesa-Bianchi, 2012).

**Illustration.** In Figure 11.2, we plot the performance of the UCB algorithm with $k = 10$ arms. In particular, the right plot highlights that the upper confidence bounds for all arms tend to be equal.

### 11.3.4 Adversarial bandits (♦)

We finish this section on multi-armed bandits by studying a non-stochastic set-up referred to as the adversarial set-up. We now have arbitrary reward vectors $\mu_t \in [0,1]^k$, $t \geqslant 1$, that may vary with time and are assumed deterministic, and at each time step, we choose an arm $i_t$, and we receive reward $\mu_t^{(i_t)}$. This context where the reward vectors are selected in advance (but arbitrary and unknown) is referred to as an "oblivious" adversary, as opposed to an "adaptive" adversary where the functions can depend on past information.

The regret is then

$$\max_{i \in \{1,\dots,k\}} \sum_{s=1}^t \mu_s^{(i)} - \sum_{s=1}^t \mu_s^{(i_s)}.$$

Note that in this set-up, there is no randomness in the environment and that we receive rewards that are elements of $[0,1]$ and not sampled in $\{0,1\}$ from that quantity. The stochastic setting can be seen as a particular subcase (but for which other algorithms, such as UCB, can be applied; see a comparison at the end of this section).

**Impossibility of deterministic policies.** If the choice of $i_t \in \{1,\dots,k\}$ is deterministic (and function of the past information), then there exists a reward sequence $(\mu_t)$ so that $\mu_t^{(i_t)} = 0$ and $\mu_t^{(i)} = 1$ for $i \neq i_t$. After $t$ steps, at least one arm has been chosen less than $t/k$ times. For that arm $\sum_{s=1}^t \mu_s^{(i)} \geqslant t - t/k$, and thus the regret is greater than $t(1-1/k)$ which is linear in $t$.

We, therefore, consider expectations from a randomized algorithm.

**Hedge algorithm (♦).** We start with the situation where a full reward vector $\mu_t \in [0,1]^k$ is observed at every iteration (after the choice of action, which is sampled from the probability vector $\pi_{t-1}$ in the simplex in $k$ dimensions). The expectation of the reward is then $\pi_{t-1}^\top \mu_t$. The Hedge algorithm (Freund and Schapire, 1997) consists in starting with $\pi_0$ uniform and updating $\pi_t$ as follows:

$$\forall i \in \{1,\dots,n\},\ \pi_t^{(i)} = \frac{\pi_{t-1}^{(i)} \exp(\gamma \mu_t^{(i)})}{\sum_{j=1}^k \pi_{t-1}^{(j)} \exp(\gamma \mu_t^{(j)})},$$

where $\gamma > 0$ is a free parameter. This happens to be exactly the online mirror descent algorithm from Section 11.1.3, with no randomness, applied to $F_t(\pi) = \mu_t^\top \pi$, with the entropy mirror map. We thus get immediately an expected normalized regret (with respect to the randomization of the algorithm) less than $\sqrt{2\log(k)}/\sqrt{t}$ for the choice $\gamma = \sqrt{2\log(k)}/\sqrt{t}$. We therefore get a (unnormalized) regret proportional to $\sqrt{t\log(k)}$.

**Exp3 algorithm (♦♦).** To tackle the bandit case with limited feedback, we follow the same strategy as the Hedge algorithm but with an unbiased estimator of the vector $\mu_t \in [0,1]^k$, from which we only observe the component $\mu_t^{(i_t)}$, where $i_t$ is sampled from $\pi_{t-1}$. The estimator suggested by Auer et al. (2002b) is an importance sampling estimator and leads to the "Exp3" algorithm. It is defined as $\hat{\mu}_t^{(i)} = \mu_t^{(i_t)} 1_{i \neq i_t} / \pi_{t-1}^{(i)}$; it thus has expectation $\mu_t$, and variance $\mathbb{E}\big[\|\hat{\mu}_t\|_\infty^2\big] \leqslant \mathbb{E}\big[\|\hat{\mu}_t\|_2^2\big] \leqslant \sum_{i=1}^k 1/\pi_{t-1}^{(i)}$, which is not enough to get a non-explosive bound. However, an improvement on Prop. 11.1 may be obtained for the simplex.

**Proposition 11.3** *The mirror descent recursion in Eq. (11.3), for $\mathcal{C}$ the simplex and $\Phi$ the entropy mirror map, is equal to $\theta_t^{(i)} = \theta_{t-1}^{(i)} \exp(-\gamma g_t^{(i)}) / \sum_{j=1}^d \theta_{t-1}^{(j)} \exp(-\gamma g_t^{(j)})$ for all $i \in \{1, \dots, d\}$. Then, assuming that $g_t$ has almost surely non-negative components and $\mathbb{E}[\sum_{i=1}^d \theta_{s-1}^{(i)} (g_s^{(i)})^2 | \mathcal{F}_{s-1}] \leqslant B^2$ almost surely for all $s \geqslant 1$, for every $\theta \in \mathcal{C}$, we have:*

$$\frac{1}{t} \sum_{s=1}^t \mathbb{E}\big[F_s(\theta_{s-1}) - F_s(\theta)\big] \leqslant \frac{1}{\gamma t} D_\Phi(\theta, \theta_0) + \frac{\gamma B^2}{2}.$$

**Proof** Following the proof of Prop. 11.1, we have $D_\Phi(\theta_t, \theta_{t-1}) + \gamma g_t^\top (\theta_t - \theta_{t-1}) = -\gamma \sum_{i=1}^d \theta_{t-1}^{(i)} g_t^{(i)} - \log \sum_{i=1}^d \exp(-\gamma g_t^{(i)})$, which, using Exercise 1.14, is greater than $-\frac{\gamma^2}{2} \sum_{i=1}^d \theta_{t-1}^{(i)} (g_t^{(i)})^2$. This leads to the desired result. ■

We can now provide a regret bound for the Exp3 algorithm by using Prop. 11.3 and noticing that we need to bound $\mathbb{E}[\sum_{i=1}^d \pi_{s-1}^{(i)} (\hat{\mu}_s^{(i)})^2 | \mathcal{F}_{s-1}] = \sum_{i=1}^d (\mu_t^{(i)})^2 \leqslant k$. This leads to, after optimizing with respect to the step-size, a non-normalized regret bound proportional to $\sqrt{kt \log k}$.

**From adversarial to stochastic.** In the stochastic set-up, the UCB algorithms provided an un-normalized regret of order $\sqrt{kt \log k}$, which is the same as the regret of the Exp3 algorithm, which is aimed at the adversarial setting. For an analysis of Exp3 in the stochastic case, see Seldin et al. (2013) for a similar regret bound.

## 11.4 Conclusion

In this chapter, we have provided extensions to the classical independent and identically distributed setting that is the book's main focus. In the convex case, algorithms and analysis were similar to the classical case, and seamlessly allowed arbitrary sequences of functions to be optimized. In the bandit setting, where only partial information was provided, a dedicated algorithmic framework was presented (optimism in front of uncertainty). There are multiple extensions, as described byShalev-Shwartz (2011); Bubeck and Cesa-Bianchi (2012); Hazan (2022); Slivkins (2019); Lattimore and Szepesvári (2020).

# Chapter 12

# Over-parameterized models

**Chapter summary**

- A model is said to be over-parameterized when it has sufficiently many parameters to fit the training data perfectly. While many overparameterized models can significantly overfit the data, the ones learned by (stochastic) gradient descent typically do not.
- Implicit regularization of gradient descent: for linear models, when there are several minimizers (typically for over-parameterized models), gradient descent techniques tend to converge to the one with minimum Euclidean norm.
- Double descent: for unregularized models learned with gradient descent techniques, when the number of parameters grows and when gradient descent is used to fit the model, the performance can exhibit a second descent after the test error blows up when the number of parameters goes beyond the number of observations.
- Global convergence of gradient descent for two-layer neural networks: in the infinite width (and thus strongly over-parameterized) limit, gradient descent exhibits some globally convergent behavior for a non-convex problem, which can be analyzed for simple architectures.

In this chapter, we will cover three recent topics within learning theory, all related to high-dimensional models (such as neural networks) in the "over-parameterized" regime, where the number of parameters is larger than the number of observations. When regularization is added to the estimation procedures, we have seen in Chapters 7, 8, and 9, that estimation can be numerically and statistically efficient by adding penalties to the empirical risk. In this section, we consider primarily non-penalized problems, with a regularization coming from the choice of optimization algorithm (here, gradient descent).

The number of parameters is not what generally characterizes the generalization capabilities of regularized learning methods. See Section 3.6 and Section 9.2.3.

## 12.1 Implicit bias of gradient descent

Given an optimization problem whose aim is to minimize some function $F(\theta)$ over with respect to a parameter $\theta \in \mathbb{R}^d$, if there is a unique global minimizer $\theta_*$, then the goal of optimization algorithms is to find this minimizer, that is, we want that the $t$-th iterate $\theta_t$ converges to $\theta_*$. When there are multiple minimizers (thus for a function which cannot be strongly convex), we showed only that $F(\theta_t) - \inf_{\theta \in \mathbb{R}^d} F(\theta)$ is converging to zero for convex functions $F$ (and only if a minimizer exists, see Chapter 5).

With some extra assumptions, it can be shown that the algorithm converges to one of the multiple minimizers of $F$ (Bolte et al., 2010) (note that when $F$ is convex, this set is also convex). But which one? This is referred to as the implicit regularization properties of optimization algorithms, and here, gradient descent and its variants.

This is interesting in machine learning because, when $F(\theta)$ is the empirical loss on $n$ observations, $d$ is much larger than $n$, and no regularization is used, there are multiple minimizers. An arbitrary empirical risk minimizer is not expected to work well on unseen data, and a classical solution is to use explicit regularization (e.g., $\ell_2$-norms like in Chapter 3 and Chapter 7, or $\ell_1$-norms like in Chapters 8 and 9). In this section, we show that optimization algorithms have a similar regularizing effect. In a nutshell, gradient descent usually leads to minimum $\ell_2$-norm solutions, in a similar way that boosting algorithms were related to $\ell_1$-norm regularization in Section 10.3. This shows that the chosen empirical risk minimizer is not arbitrary.

This will be explicitly shown for the quadratic loss and partially only for the logistic loss. These results will be used in subsequent sections.

### 12.1.1 Least-squares

We consider the least-squares objective function[1] $F(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$ from Chapter 3, with $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times d}$ such that $d > n$ and (for simplicity) $XX^\top \in \mathbb{R}^{n \times n}$ invertible (this is the kernel matrix). There are thus infinitely many (a whole affine subspace) solutions such that $y = X\theta$, since the column space of $X$ is the entire space $\mathbb{R}^n$ and $\theta$ has dimension $d > n$. We apply gradient descent with step-size $\gamma \leqslant \frac{1}{L} = \lambda_{\max}(\frac{1}{n}X^\top X)^{-1}$, which is equal to $\lambda_{\max}(\frac{1}{n}XX^\top)^{-1}$, starting from $\theta_0 = 0$, leading to $\theta_t = \theta_{t-1} - \frac{\gamma}{n}X^\top(X\theta_{t-1} - y)$, and

[1] We use $X$ as a notation for the design matrix to highlight that in this section we will consider predictions that are also linear in $x$.

thus we have

$$X\theta_t - y = X\theta_{t-1} - y - \frac{\gamma}{n}XX^\top(X\theta_{t-1} - y) = \big(I - \frac{\gamma}{n}XX^\top\big)(X\theta_{t-1} - y),$$

leading to, by recursion,

$$X\theta_t - y = \big(I - \frac{\gamma}{n}XX^\top\big)^t(X\theta_0 - y) = \big(I - \frac{\gamma}{n}XX^\top\big)^t(-y). \tag{12.1}$$

We thus get $\|X\theta_t - y\|_2^2 \leqslant \big(1 - \frac{\gamma}{n}\lambda_{\max}(XX^\top)\big)^{2t}\|y\|_2^2$ and hence linear convergence of $X\theta_t$ towards $y$, with a convergence rate depending on the condition number of the kernel matrix $XX^\top$.

Moreover, when started at $\theta_0 = 0$, gradient descent techniques (stochastic or not) will always have iterates $\theta_t$ that are linear combinations of rows of $X$, that is, of the form $\theta_t = X^\top\alpha_t$ for some $\alpha_t \in \mathbb{R}^n$. This is an alternative algorithmic version of the representer theorem from Chapter 7.

Since $X\theta_t$ converges to $y$, $X\theta_t = XX^\top\alpha_t$ converges to $y$. Since $K = XX^\top$ is invertible, this means that $\alpha_t$ is converges to $K^{-1}y$, and thus $\theta_t = X^\top\alpha_t$ converges to $X^\top K^{-1}y$. It turns out that this is exactly the minimum $\ell_2$-norm solution as by standard Lagrangian duality (Boyd and Vandenberghe, 2004):

$$\begin{aligned}
\inf_{\theta\in\mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \text{ such that } y = X\theta &= \inf_{\theta\in\mathbb{R}^d}\sup_{\alpha\in\mathbb{R}^n} \frac{1}{2}\|\theta\|_2^2 + \alpha^\top(y - X\theta)\\
&= \sup_{\alpha\in\mathbb{R}^n} \alpha^\top y - \frac{1}{2}\|X^\top\alpha\|_2^2 \text{ with } \theta = X^\top\alpha \text{ at optimum,}\\
&= \sup_{\alpha\in\mathbb{R}^n} \alpha^\top y - \frac{1}{2}\alpha^\top K\alpha.
\end{aligned}$$

The last problem is exactly solved for $\alpha = K^{-1}y$, with at optimum $\theta = X^\top\alpha$. Note that in Chapter 7, we used this formula for function interpolation to compare different RKHSs (see Prop. 7.2).

**Lojasiewicz's inequality (♦).** It turns out that the linear convergence obtained from Eq. (12.1) can be obtained directly for any $L$-smooth function, for which we have the so-called Lojasiewicz's inequality:

$$\forall\theta\in\mathbb{R}^d,\ F(\theta) - F(\theta_*) \leqslant \frac{1}{2\mu}\|F'(\theta)\|_2^2, \tag{12.2}$$

for some $\mu > 0$.

In Chapter 5, we have seen that this is a consequence of $\mu$-strong-convexity (Lemma 5.1), but this can be satisfied without strong convexity. For example, for the least-squares example, we have, for any minimizer $\theta_*$:

$$\begin{aligned}
\|F'(\theta)\|_2^2 &= \Big\|\frac{1}{n}X^\top X(\theta - \theta_*)\Big\|_2^2 = \frac{1}{n^2}(\theta - \theta_*)^\top X^\top XX^\top X(\theta - \theta_*)\\
&\geqslant \frac{\lambda_{\min}^+(XX^\top)}{n^2}(\theta - \theta_*)^\top X^\top X(\theta - \theta_*),
\end{aligned}$$

where $\lambda_{\min}^{+}(XX^\top) = \lambda_{\min}^{+}(X^\top X)$ is the smallest *non-zero* eigenvalue of $XX^\top$ (which is also the one of $X^\top X$). Thus, we have

$$\|F'(\theta)\|_2^2 \geqslant \frac{\lambda_{\min}^{+}(K)}{n^2}\|X(\theta-\theta_*)\|_2^2 = \frac{2\lambda_{\min}^{+}(K)}{n}[F(\theta)-F(\theta_*)].$$

Thus, Eq. (12.2) is satisfied with $\mu = \frac{1}{n}\lambda_{\min}^{+}(K)$. Note that this includes also the strongly-convex case since $\lambda_{\min}^{+}(X^\top X) \geqslant \lambda_{\min}(X^\top X)$.

When Eq. (12.2) is satisfied, we have for the $t$-th iterate of gradient descent with step-size $\gamma = 1/L$, following the analysis of Chapter 5 (Prop. 5.3):

$$F(\theta_t) - F(\theta_*) \leqslant F(\theta_{t-1}) - F(\theta_*) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2 \leqslant \big(1-\frac{\mu}{L}\big)\big[F(\theta_{t-1}) - F(\theta_*)\big].$$

Moreover, we can then show that the iterates $x_t$ are also converging to a minimizer of $F$ (see Bolte et al., 2010; Karimi et al., 2016, for more details).

### 12.1.2 Separable classification

We now consider logistic regression, that is, for $y_i \in \{-1,1\}$, $i=1,\dots,n$,

$$F(\theta) = \frac{1}{n}\sum_{i=1}^n \log(1+\exp(-y_i x_i^\top\theta)), \qquad (12.3)$$

with $X \in \mathbb{R}^{n\times d}$ the design matrix (with rows equal to the input vectors $x_1,\dots,x_n$) such that $d > n$ and the kernel matrix $XX^\top \in \mathbb{R}^{n\times n}$ is invertible. In the regression setting, interpolation corresponds to $X\theta = y$; in the classification setting, we predict perfectly if and only if $\mathrm{sign}(X\theta) = y$, which happens when $y \circ (X\theta)$ (where $\circ$ is the component-wise product) has strictly positive components. Such an interpolator always exists (for example, the one for regression on $y$).

**Maximum margin classifier.** Since $XX^\top$ is invertible, there exists $\eta \in \mathbb{R}^d$ of unit norm such that $\forall i \in \{1,\dots,n\}$, $y_i x_i^\top \eta > 0$ (e.g., as mentioned earlier, $y = X^\top(XX^\top)^{-1}y$). We denote by $\eta_*$ the one such that

$$\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$$

is maximal (and thus strictly positive). We denote by $\frac{1}{\rho} > 0$ its value. This $\eta_*$ solves the following problem, which can be rewritten as, using Lagrange duality:

$$\begin{aligned}\frac{1}{\rho} = \sup_{\|\eta\|_2\leqslant 1}\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta &= \sup_{\|\eta\|_2\leqslant 1, t\in\mathbb{R}} t \ \text{ such that } \ \forall i\in\{1,\dots,n\},\ y_i x_i^\top\eta \geqslant t \\ &= \inf_{\alpha\in\mathbb{R}_+^n}\sup_{\|\eta\|_2\leqslant 1, t\in\mathbb{R}} t + \sum_{i=1}^n \alpha_i(y_i x_i^\top\eta - t) \\ &= \inf_{\alpha\in\mathbb{R}_+^n}\Big\|\sum_{i=1}^n \alpha_i y_i x_i\Big\|_2 \ \text{ such that } \sum_{i=1}^n \alpha_i = 1, \qquad (12.4)\end{aligned}$$

with $\eta \propto \sum_{i=1}^n \alpha_i y_i x_i$ at optimum. Moreover, by complementary slackness, a non-negative $\alpha_i$ is non zero only for $i$ attaining the minimum in $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$.

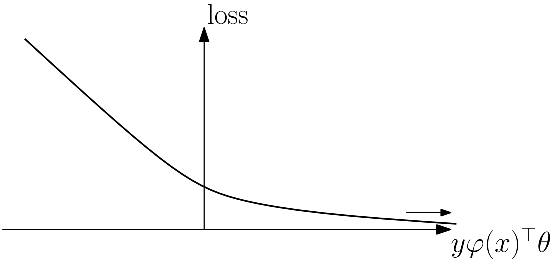
**Reformulation as an SVM.** Because of positive homogeneity, we aim to have the quantity $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$ large and $\|\eta\|_2$ small, and we can decide to constrain the first and minimize the second one. In other words, we can see $\eta_*$ as the unit-norm direction of the solution $\theta_*$ of the following optimization problem (with now non-negative Lagrange multipliers $\alpha_1, \dots, \alpha_n$):

$$\begin{aligned}\inf_{\theta\in\mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \text{ such that } \operatorname{Diag}(y)X\theta \geqslant 1_n &= \inf_{\theta\in\mathbb{R}^d} \sup_{\alpha\in\mathbb{R}_+^n} \frac{1}{2}\|\theta\|_2^2 + \alpha^\top(1_n - \operatorname{Diag}(y)X\theta) \\ &= \sup_{\alpha\in\mathbb{R}_+^n} \alpha^\top 1_n - \frac{1}{2}\|X^\top \operatorname{Diag}(y)\alpha\|_2^2 \\ &\qquad \text{with } \theta = X^\top \operatorname{Diag}(y)\alpha \text{ at optimum.}\end{aligned}$$

Note that above, $\operatorname{Diag}(y)X\theta \geqslant 1_n$ is the compact formulation of the inequality constraints $\forall i \in \{1,\dots,n\}$, $y_i x_i^\top \theta \geqslant 1$. Given $\eta$, $\theta$ is equal to $\eta / \min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$, so that the optimal value of the problem above is $\frac{1}{2}\rho^2$.

The vector $\theta_*$ above is the solution of the separable SVM from Section 4.1.2 with vanishing regularization parameter, that is, of $\frac{1}{2}\|\theta\|_2^2 + C\sum_{i=1}^n (1 - y_i x_i^\top \theta)_+$ for $C$ large enough. See Section 4.1.2 for an illustration.

**Divergence and convergence of directions.** Because the logistic loss plotted below is strictly positive and tends to zero at infinity, the function $F$ in Eq. (12.3) has an infimum equal to zero, which is not attained. However, for any sequence $\theta_t$ such that all $y_i x_i^\top \theta_t$, $i = 1, \dots, n$, tend to $+\infty$, we have $F(\theta_t) \to \inf_{\theta\in\mathbb{R}^d} F(\theta) = 0$.



In such a situation, gradient descent cannot converge to a point, and it has to diverge to achieve small values of $F$. It turns out that it diverges along a direction, that is, $\|\theta_t\|_2 \to +\infty$, with $\frac{1}{\|\theta_t\|_2}\theta_t \to \eta$ for some $\eta \in \mathbb{R}^d$ of unit $\ell_2$-norm. That direction $\eta$ has to lead to perfect classification (that is $y_i x_i^\top \eta > 0$ for all $i \in \{1,\dots,n\}$). Among all of them, the direction $\eta$ is exactly the maximum margin one, that is, which maximizes $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta > 0$. See Gunasekar et al. (2018) for a detailed proof. Here, we just give a simple argument on a slightly modified problem.

**Gradient flow on the exponential loss (♦).** We consider instead the logarithm of the empirical risk associated with the exponential loss $G(\theta) = \log \big[ \frac{1}{n} \sum_{i=1}^n \exp(-y_i x_i^\top \theta) \big]$, which is asymptotically equivalent to the logistic loss for $y_i x_i^\top \theta$ tending to infinity for all $i \in \{1, \dots, n\}$ (which is the case when gradient descent diverges). Moreover, we replace the gradient descent recursion $\theta_t = \theta_{t-1} - \gamma G'(\theta_{t-1})$ by the gradient flow

$$\xi'(\tau) = -G'(\xi(\tau)). \tag{12.5}$$

This ordinary differential equation approximates gradient descent for vanishing step-sizes, as $\xi(\gamma t) \approx \theta_t$ for $\gamma$ tending to zero. The use of gradient flows instead of gradient descent is a standard theoretical simplification that allows the use of differential calculus (see, e.g., Scieur et al., 2017, and references therein).

We have, for all $\theta \in \mathbb{R}^d$:

$$G'(\theta) = \frac{-\sum_{i=1}^n y_i x_i \exp(-y_i x_i^\top \theta)}{\sum_{i=1}^n \exp(-y_i x_i^\top \theta)} = -\sum_{i=1}^n \alpha_i y_i x_i,$$

for $\alpha_i = \frac{\exp(-y_i x_i^\top \theta)}{\sum_{j=1}^n \exp(-y_j x_j^\top \theta)}$, for $i \in \{1, \dots, n\}$, leading to $\alpha$ in the simplex (with non-negative components and summing to one). Thus, from Eq. (12.4) defining the maximum margin hyperplane, we get:

$$\|G'(\theta)\| \geqslant \frac{1}{\rho}. \tag{12.6}$$

Moreover, comparing maxima and soft-maxima,[2], we get:

$$-\log(n) - \min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta \leqslant G(\theta) \leqslant -\min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta.$$

We thus have a flow $\tau \mapsto \xi(\tau)$ that cannot converge as by Eq. (12.5) and Eq. (12.6), $\|\xi'(\tau)\|_2 \geqslant 1/\rho$. Moreover, it maximizes a function that is a constant away from the margin. Therefore, it has to diverge along a direction that maximizes this margin. We now make this reasoning precise.

By integration by part, using $\frac{d}{d\tau} G(\xi(\tau)) = G'(\xi(\tau))^\top \xi'(\tau) = -\|G'(\xi(\tau))\|_2^2$ and Eq. (12.6) twice:

$$\begin{aligned}
\min_{i \in \{1,\dots,n\}} y_i x_i^\top \xi(\tau) &\geqslant -G(\xi(\tau)) - \log(n) \\
&= -G(\xi(0)) + \int_0^\tau \|G'(\xi(u))\|_2^2 du - \log(n) \\
&\geqslant -G(\xi(0)) + \frac{1}{\rho} \int_0^\tau \|G'(\xi(u))\|_2 du - \log(n) \qquad (12.7) \\
&\geqslant -G(\xi(0)) + \frac{1}{\rho^2} \tau - \log(n) \qquad (12.8)
\end{aligned}$$

[2] We use $0 \geqslant \log \big( \frac{1}{n} \sum_{i=1}^n e^{z_i} \big) - \max_{i \in \{1,\dots,n\}} z_i = \log \big( \frac{1}{n} \sum_{i=1}^n e^{z_i - \max_{j \in \{1,\dots,n\}} z_j} \big) \geqslant \log(1/n)$.

(note that Eq. (12.7) is not needed to derive Eq. (12.8), but will be used later). Thus, from Eq. (12.8), for $\tau \geqslant \rho^2\big[\log(n) + G(\xi(0))\big]$, we have a non-negative lower bound. Moreover the derivative of $\tau \mapsto \|\xi(\tau)\|_2$ is $\tau \mapsto -G'(\xi(\tau))^\top\big(\frac{\xi(\tau)}{\|\xi(\tau)\|_2}\big)$, and its magnitude is less than $\|G'(\xi(\tau))\|_2$. This implies by integration that

$$\|\xi(\tau)\|_2 \leqslant \|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du.$$

We thus get from Eq. (12.7) and Eq. (12.6):

$$\begin{aligned}
\min_{i\in\{1,\dots,n\}} y_i x_i^\top\big(\tfrac{\xi(\tau)}{\|\xi(\tau)\|_2}\big) &\geqslant \frac{-G(\xi(0)) + \frac{1}{\rho}\int_0^\tau \|G'(\xi(u))\|_2 du - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du}\\
&= \frac{-G(\xi(0)) + \frac{1}{\rho}\big(\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du\big) - \frac{1}{\rho}\|\xi(0)\|_2 - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du}\\
&= \frac{1}{\rho} + \frac{-G(\xi(0)) - \frac{1}{\rho}\|\xi(0)\|_2 - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du}\\
&\geqslant \frac{1}{\rho} - \frac{G(\xi(0)) + \frac{1}{\rho}\|\xi(0)\|_2 + \log(n)}{\|\xi(0)\|_2 + \tau/\rho}, \text{ since } \int_0^\tau \|G'(\xi(u))\|_2 du \geqslant \frac{\tau}{\rho}.
\end{aligned}$$

The lower bound above tends to $\frac{1}{\rho}$ when $\tau$ tends to infinity, which is the maximal value. We thus get convergence to the maximum margin hyperplane.

**Alternative proof (♦).** We provide another informal derivation based on gradients. The gradient $F'(\theta)$ of the original objective function based on the logistic loss is equal to $F'(\theta) = -\frac{1}{n}\sum_{i=1}^n \frac{\exp(-y_i x_i^\top\theta)}{1+\exp(-y_i x_i^\top\theta)} y_i x_i$.

Asymptotically, $\theta_t$ will behave as $\|\theta_t\|\eta$, with $\|\theta_t\|_2$ tending to infinity. Thus, because we have a sum of exponentials with arguments that go to infinity, the dominant term in $F'(\theta_t)$ corresponds to the indices $i$ for which $-y_i x_i^\top\eta$ is the largest. Moreover, all of these values must be negative (indeed, we can only attain zero loss for well-classified training data). We denote by $I$ this set. Thus, asymptotically,

$$F'(\theta_t) \sim -\frac{1}{n}\sum_{i\in I} y_i \exp(-\|\theta_t\|_2 y_i x_i^\top\eta) x_i.$$

Moreover, if we admit for simplicity that $F'(\theta_t)$ diverges in the direction $-\eta$, thus $\eta$ has to be proportional to a vector $\sum_{i\in I}\alpha_i y_i x_i$, where $\alpha \geqslant 0$, and $\alpha_i = 0$ as soon as $i$ is not among the minimizers of $y_i x_i^\top\eta$. This is exactly the optimality condition for $\eta_*$ in Eq. (12.4). Thus $\eta = \eta_*$.

**Summary.** Overall, we obtain a classifier corresponding to a minimum $\ell_2$-norm separating hyperplane. See examples in two dimensions in Figure 12.1. Note that gradient descent on the logistic regression problem may not be the most efficient way to obtain a maximum margin hyperplane. See convergence rates by Soudry et al. (2018); Ji and Telgarsky (2018) and a simpler subgradient algorithm below.

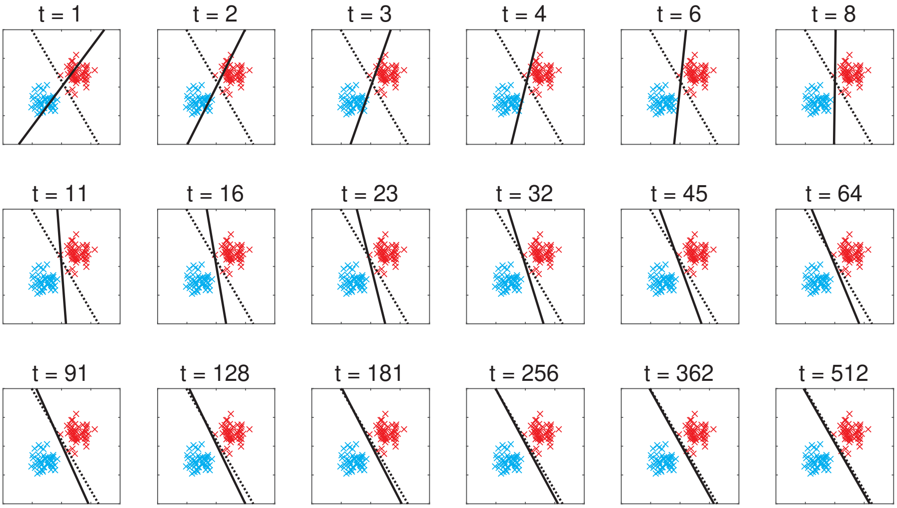


Figure 12.1: Logistic regression on separable data estimated with gradient descent on the unregularized empirical risk, at various numbers of iterations $t$. This is implemented by minimizing the logistic loss function with data $\binom{x_i}{1} \in \mathbb{R}^3$. The dotted line represents the maximum margin hyperplane, while the plain line represents the current classification hyperplane.

**Subgradient method for the hinge loss and perceptron (♦).** For linearly separable data, dedicated algorithms exist that are explicitly or implicitly based on the hinge loss. If we consider the "margin" $\rho > 0$ defined as

$$\rho^2 = \inf_{\theta \in \mathbb{R}^d} \ \|\theta\|_2^2 \ \text{ such that } \ \mathrm{Diag}(y) X \theta \geqslant 1_n. \tag{12.9}$$

To obtain a linear separator, one can use the subgradient method from Section 5.3 applied to the cost function

$$F(\theta) = \max_{i \in \{1,\dots,n\}} (1 - y_i x_i^\top \theta)_+.$$

The iteration is

$$\theta_t = \theta_{t-1} + \gamma 1_{y_{i_t} x_{i_t}^\top \theta_{t-1} < 1} y_{i_t} x_{i_t}, \tag{12.10}$$

where $i_t \in \arg\min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta_{t-1}$, and $\gamma$ is the step-size. With $\theta_*$ being the minimizer in Eq. (12.9), we have $F(\theta_*) = 0 = \min_{\theta \in \mathbb{R}^d} F(\theta)$, and after $t$ steps, following the analysis of Prop. 5.6, we get:

$$\min_{u \leqslant t} F(\theta_u) \leqslant \frac{\gamma R^2}{2} + \frac{\rho^2}{2\gamma t}.$$

The quantity above is less than $\varepsilon$ as soon as $\frac{\gamma R^2}{2} + \frac{\rho^2}{2\gamma t} \leqslant \varepsilon$, which can be achieved by $\gamma = \frac{\varepsilon}{R^2}$ and $t = \frac{\rho^2}{\gamma \varepsilon} = \frac{\rho^2 R^2}{\varepsilon^2}$, thus an objective function less than $\frac{\rho R}{\sqrt{t}}$. If $F(\theta_t) < 1$, then, following Section 4.1), we have linearly separated the data, which happens as soon as we have $t > (\rho R)^2$. The iteration in Eq. (12.10) is a variation on the perceptron algorithm (Rosenblatt, 1958; Novikoff, 1962), presented in the exercise below.

**Exercise 12.1** *Extend the analysis above to the stochastic gradient algorithm for the objective function $F(\theta) = \frac{1}{n}\sum_{i=1}^n (1 - y_i x_i^\top \theta)_+$. What can be concluded when the data are i.i.d. and a single pass over the data is made?*

**Exercise 12.2 (perceptron)** *In the same set-up as above, we consider the iteration, started at $\theta_0$, that at time $t$ looks for an $i_t \in \{1, \dots, n\}$ such that $y_{i_t} x_{i_t}^\top \theta_{t-1} \leqslant 0$ and, if found, implements the update $\theta_t = \theta_{t-1} + y_{i_t} x_{i_t}$. Show that all points are well-classified if the number of iterations is greater than $(\rho R)^2$.*

### 12.1.3 Beyond convex problems (♦)

The implicit bias of gradient descent can be observed and analyzed in various models, beyond linear ones. In this section, we focus on diagonal linear networks, where the analysis of gradient flows is reasonably simple. We highlight the potential difference in implicit biases depending on the chosen learning algorithm.

We consider our traditional least-squares model with design matrix $X \in \mathbb{R}^{n \times d}$ and response vector $y \in \mathbb{R}^n$, with the least-squares objective function: $F(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$, where $d > n$, and with an invertible kernel matrix $XX^\top \in \mathbb{R}^{n \times n}$, leading to infinitely many minimizers. We consider different learning dynamics, which we study in continuous time for simplicity.

**From gradient flow to mirror flow.** The gradient flow dynamics on $\theta$ is the ordinary differential equation (ODE):

$$\frac{d}{dt}\theta(t) = -F'(\theta(t)) = -\frac{1}{n}X^\top(X\theta(t) - y).$$

As shown in Section 12.1.1, $\theta(t)$ converges exponentially fast to the minimum $\ell_2$-norm interpolator. This can extended to the continuous-time limit of mirror descent presented in Section 11.1.3. If we consider a $\mu$-strongly convex mirror map $\Phi : \mathbb{R}^d \to \mathbb{R}$, the mirror descent recursion is defined as $\Phi'(\tilde{\theta}_{k+1}) = \Phi'(\tilde{\theta}_k) - \gamma F'(\tilde{\theta}_k)$, and we obtain the continuous-time limit by setting $\theta(\gamma k) = \tilde{\theta}_k$, leading to the ODE:

$$\frac{d}{dt}\Phi'(\theta(t)) = -F'(\theta(t)), \tag{12.11}$$

which is equivalent to $\Phi''(\theta(t))\frac{d}{dt}\theta(t) = -F'(\theta(t)) = -\frac{1}{n}X^\top(X\theta(t) - y)$. This leads to

$$\frac{d}{dt}\big[X\theta(t) - y\big] = -\frac{1}{n}X\Phi''(\theta(t))^{-1}X^\top\big[X\theta(t) - y\big],$$

which in turn leads to

$$\begin{aligned}\frac{d}{dt}\big[\|X\theta(t)-y\|_2^2\big] &= -\frac{2}{n}\big[X\theta(t)-y\big]^\top X\Phi''(\theta(t))^{-1}X^\top\big[X\theta(t)-y\big] \\ &\leqslant -\frac{2\lambda_{\min}(XX^\top)}{n\mu}\|X\theta(t)-y\|_2^2.\end{aligned}$$

Thus, like for the gradient flow dynamics, $X\theta(t)$ converges exponentially fast to $y$. Since from Eq. (12.11), $\Phi'(\theta)$ is of the form $\Phi'(\theta_0)+X^\top\alpha$ for some $\alpha\in\mathbb{R}^n$, the corresponding limit $\alpha_\infty$ (with the corresponding $\theta_\infty$ such that $\Phi'(\theta_\infty)=\Phi'(\theta_0)+X^\top\alpha_\infty$) of $\alpha(t)$ when $t$ tends to infinity is such $X\theta_\infty=y$ and $\Phi'(\theta_\infty)=\Phi'(\theta_0)+X^\top\alpha_\infty$, which is exactly the interpolator of the data with minimum value of the Bregman divergence $D_\Phi(\theta,\theta_0)=\Phi(\theta)-\Phi(\theta_0)-\Phi'(\theta_0)^\top(\theta-\theta_0)$.[3]

**Diagonal linear networks.** Following Woodworth et al. (2020), we consider "diagonal linear networks", which are simple hidden layer neural networks defining a prediction function of the form

$$f(x)=(u\circ v)^\top x=\sum_{j=1}^d u_jv_jx_j,$$

for $u,v\in\mathbb{R}^d$, and $u\circ v$ denoting the pointwise product. This is thus an alternative (non-linear) way of defining a linear model defined by $\theta=u\circ v$. We study the gradient flow dynamics for the objective function $G(u,v)=F(u\circ v)$, that is,

$$\begin{aligned}\frac{d}{dt}u(t) &= -\frac{\partial G}{\partial u}(u(t),v(t))=-F'(u(t)\circ v(t))\circ v(t) \\ \frac{d}{dt}v(t) &= -\frac{\partial G}{\partial v}(u(t),v(t))=-F'(u(t)\circ v(t))\circ u(t).\end{aligned}$$

We thus have $\frac{d}{dt}[u\circ u(t)-v\circ v(t)]=0$, and therefore $u\circ u-v\circ v$ is a constant function. If we initialize $v=0$ and $u$ the constant vector equal to $\alpha\in\mathbb{R}$, we have $u\circ u(t)-v\circ v(t)=\alpha^2 1_d$ for all $t\geqslant 0$. Thus, for $\theta(t)=u\circ v(t)$, we have:

$$\frac{d}{dt}\theta(t)=u(t)\circ\frac{d}{dt}v(t)+v(t)\circ\frac{d}{dt}u(t)=-F'(\theta(t))\circ(u\circ u(t)+v\circ v(t)).$$

Moreover, we have $\theta\circ\theta(t)=(u\circ u)\circ(v\circ v)(t)=\frac{1}{4}[u\circ u(t)+v\circ v(t)]^2-\frac{1}{4}[u\circ u(t)-v\circ v(t)]^2$. Thus, we obtain the following ODE for each component $\theta_j$ of $\theta$:

$$\frac{d}{dt}\theta_j(t)=-F'(\theta(t))_j\sqrt{4\theta_j(t)^2+\alpha^4}.$$

It can be exactly cast as a mirror flow with mirror map $\Phi(\theta)=\sum_{j=1}^d q(\theta_j)$ for $q$ a convex function such that $q''(\eta)=(4\eta^2+\alpha^4)^{-1/2}$. By integrating twice, one obtains,

[3] By duality, we have $\inf_{X\theta=y}\Phi(\theta)-\Phi'(\theta_0)^\top\theta=\sup_{\alpha\in\mathbb{R}^n}\inf_{\theta\in\mathbb{R}^d}\Phi(\theta)-\Phi'(\theta_0)^\top\theta+\alpha^\top(y-X\theta)$, which is equal to $\sup_{\alpha\in\mathbb{R}^n}\alpha^\top y-\Phi^*(\Psi'(\theta_0)+X^\top\alpha)$, with optimality conditions $\Phi'(\theta)=\Phi'(\theta_0)+X^\top\alpha$ and $X\theta=y$.

with imposing $q'(0) = 0$:

$$q'(\eta) = \frac{1}{2}\arg\sinh(2\eta/\alpha^2) = -\log\alpha + \frac{1}{2}\log\big[2\eta + \sqrt{\alpha^4 + 4\eta^2}\big],$$

and then, imposing $q(0) = 0$, we get an even function of $\eta$ defined as:

$$\begin{aligned} q(\eta) &= \frac{\eta}{2}\arg\sinh(2\eta/\alpha^2) + \frac{\alpha^2}{4}\big[1 - \sqrt{4\eta^2/\alpha^4 + 1}\big] \\ &= \frac{\eta}{2}\log\big[2\eta + \sqrt{\alpha^4 + 4\eta^2}\big] - \frac{1}{4}\sqrt{4\eta^2 + \alpha^4} + \frac{\alpha^2}{4} - \frac{\eta}{2}\log\alpha^2. \end{aligned}$$

When $\alpha$ tends to $+\infty$, we get $q(\eta) \sim \frac{\eta^2}{\alpha^2}$, and thus the implicit bias converges to the Euclidean norm, and we recover the traditional geometry of gradient descent directly on $\theta$.

However, when $\alpha$ tends to zero, we get $q(\eta) \sim |\eta| \log\frac{1}{\alpha}$, and thus the implicit bias corresponds to the $\ell_1$-norm, showing how non-convex optimization can lead to a different implicit bias. See more details by Woodworth et al. (2020), and an analysis of the extra regularizing effect of stochastic gradient descent by Pesme et al. (2021).[4] Note that the analysis above for diagonal networks explicitly shows quantitative global convergence of the gradient flow for a non-convex objective; we consider in Section 12.3 qualitative results which apply more generally.

**Beyond linear networks.** Characterizing the implicit bias of gradient descent can be done in more complex situations. For example, Chizat and Bach (2020) show that with a neural network with ReLu activations and infinitely many neurons estimated by gradient descent on the empirical logistic loss, then in the infinite width limit, we get a predictor that interpolates the data, with a minimum specific norm, for norms which are exactly the ones obtained in Section 9.3.[5]

## 12.2 Double descent

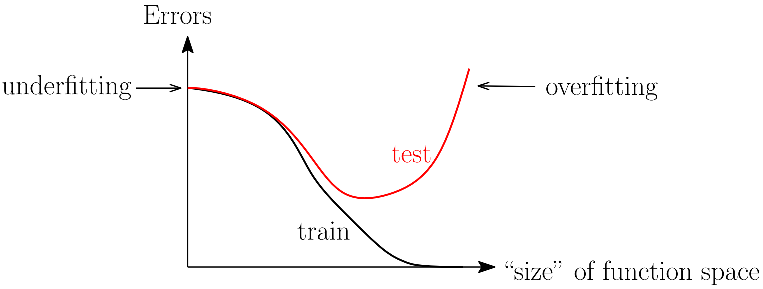
In this section, we consider a recent and interesting phenomenon described in several recent works (Belkin et al., 2019; Mei and Montanari, 2022; Geiger et al., 2020; Hastie et al., 2019), which shows a particular behavior for overparameterized models learned by gradient descent.

### 12.2.1 The double descent phenomenon

As seen in Chapter 2 and Chapter 4, typical learning curves look like the one below.

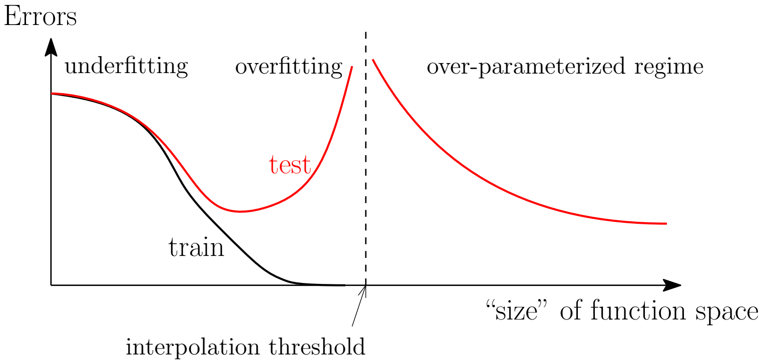
[4] See `https://francisbach.com/implicit-bias-sgd/`.

[5] See `https://www.di.ens.fr/~fbach/ltfp/wide_implicit_bias.html` for more details.



Typically, the “capacity” of the space of functions $\mathcal{H}$ used to estimate the prediction function is controlled either by the number of parameters or by some norms of its parameters. In particular, when there is zero training error at the extreme right of the curve, the testing error may be arbitrarily bad. The bound that we have used in Chapter 4, such as Rademacher averages for $\mathcal{H}$ controlled by the $\ell_2$-norm of some parameters (with a bound $D$), grows as $D/\sqrt{n}$, which can typically be quite large. These bounds were true for *all* empirical risk minimizers. In this section, we will focus on a particular one, namely the one obtained by unconstrained gradient descent.

When the model is over-parameterized (in other words, the capacity gets very large), that is, when the number of parameters is large, or the norm constraint allows for exact fitting, a new phenomenon occurs, where after the test error explodes as the capacity grows, it goes down again, as illustrated below.



This section aims to understand why, starting from empirical evidence.

There may be no double descent phenomenon if other empirical risk minimizers are used instead of the one obtained by (stochastic) gradient descent.

## 12.2.2 Empirical evidence

**Toy example with random features.** We consider a random feature models like in Chapter 7 and Chapter 9, with features $(v^\top x)_+$, for neurons $v \in \mathbb{R}^d$ sampled uniformly on the unit sphere. We consider $n = 200$, $d = 5$ with input data distributed uniformly on the unit sphere, and we consider outputs $y = \left(\frac{1}{4} + (v_*^\top x)^2\right)^{-1} + \mathcal{N}(0, \sigma^2)$, with $\sigma = 2$, for some random $v_*$.

We sample $m$ random features $v_1, \dots, v_m \in \mathbb{R}^d$ uniformly at random on the sphere, and we learn parameters $\theta \in \mathbb{R}^m$ by minimizing

$$\frac{1}{n} \sum_{i=1}^{n} \Big( y_i - \sum_{j=1}^{m} \theta_j (v_j^\top x_i)_+ \Big)^2 + \lambda \|\theta\|_2^2. \tag{12.12}$$

We report in Figure 12.2 train and test errors after learning with gradient descent until convergence: (left) varying $m$ with $\lambda = 0$, (right) varying $\lambda$ with $m = +\infty$ (we can perform estimation for $m = +\infty$ efficiently because we can compute the corresponding positive-definite kernel $k(x, x') = \mathbb{E}_v[(v^\top x)_+ (v^\top x')_+]$, see Section 9.5).

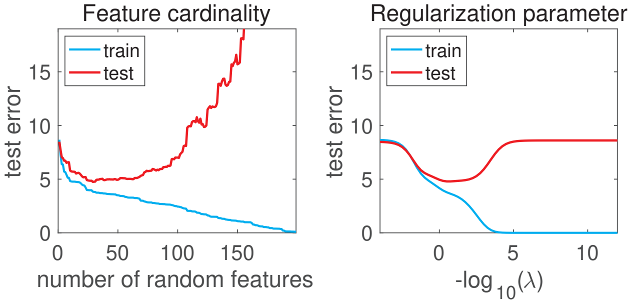


Figure 12.2: Classical learning curve: (left) train and test errors as functions of the number of random features, always less than the number of observations, (right) train and test errors for ridge regression with the same features (i.e., using $\ell_2$-regularization).

In the left plot of Figure 12.2, the number of random features $m$ is less than $n$ as the test error diverges. But, when this number $m$ is allowed to grow past $n$, we see the double descent phenomenon in Figure 12.3. Similar experiments are shown by Belkin et al. (2019); Geiger et al. (2020); Mei and Montanari (2022), in particular for neural networks.

**No phenomenon when using regularization.** When an extra regularizer is used, that is $\lambda \neq 0$ in Eq. (12.12), then the double descent phenomenon is reduced. In particular, if the regularization parameter $\lambda$ is adapted for each $m$, then the phenomenon totally disappears (see Mei and Montanari, 2022, for more details).

### 12.2.3 Linear regression with Gaussian projections (♦)

To provide some theoretical justification for the double descent phenomenon, we consider a linear regression model in the random design setting, with Gaussian inputs and Gaussian noise.

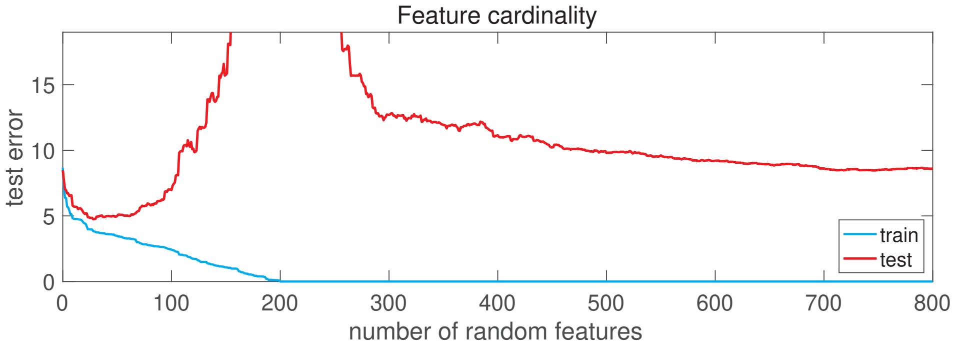


Figure 12.3: Double descent curve: train and test errors as functions of the number of random features. For $m \leqslant n = 200$, this is exactly the curves from Figure 12.2 (left).

That is, we consider a Gaussian random variable with mean 0 and covariance matrix $\Sigma$ the identity matrix, with $n$ observations $x_1, \dots, x_n$, and responses $y_i = x_i^\top \theta_* + \varepsilon_i$, with $\varepsilon_i$ normal with mean zero and variance $\sigma^2 I$.

To have a unique prediction problem with a varying number of features, we consider additional random projections, that is, like in Section 10.2.2,[6] we consider a random matrix $S \in \mathbb{R}^{d\times m}$ with independent components all sampled from a standard Gaussian distribution (mean 0 and variance 1). The main differences are that (a) we will perform an analysis in the random design setting, and (b) we will also need to tackle the overparameterized regime $m > n$.

We will compute the expectation of the risk of the minimum norm empirical risk minimizer (as detailed in Section 12.1.1), which is the one gradient descent converges to. See Bach (2023a) for further more precise asymptotic results using random matrix theory.

We denote by $X \in \mathbb{R}^{n\times d}$ the design matrix, and $\hat\Sigma = \frac{1}{n} X^\top X$ the non-centered covariance matrix, and by $K = XX^\top \in \mathbb{R}^{n\times n}$ the kernel matrix. We will need to compute expectations with respect to the data $X, \varepsilon$ *and* the random projection matrix $S$. The estimator $\hat\theta$ is equal to $S\hat\eta$ with $\hat\eta \in \mathbb{R}^m$ a minimizer of $\|y - XS\eta\|_2^2$.

The excess risk is denoted $\mathcal{R}(\hat\theta) = (\hat\theta - \theta_*)\Sigma(\hat\theta - \theta_*)$, and we now consider the two regimes $m > n$ (underparameterized) and $m > n$ (overparameterized). In both cases, as already seen in Chapter 3, the expectation of the excess risk will be composed of two terms: a (squared) "bias term" $\mathcal{R}^{(\text{bias})}(\hat\theta)$ corresponding to $\sigma = 0$, and a "variance term" $\mathcal{R}^{(\text{var})}(\hat\theta)$ corresponding to $\theta_* = 0$.

**Underparameterized regime.** In the under-parameterized regime where $n > d$, the minimum norm empirical risk minimizer is simply the ordinary least-squares estimator. We denote by $\eta_* = (S^\top \Sigma S)^{-1} S^\top \Sigma \theta_* \in \mathbb{R}^m$ the minimizer of $(\theta_* - S\eta)^\top \Sigma(\theta_* - S\eta)$. We have $S\eta_* = \Pi_S \theta_*$ with $\Pi_S = S(S^\top\Sigma S)^{-1} S^\top \Sigma \in \mathbb{R}^{d\times d}$, which is a projection matrix such that $\Pi_S S = S$, $\Pi_S^2 = \Pi_S$, and $\Pi_S^\top \Sigma \Pi_S = \Sigma \Pi_S$.

[6] For an analysis without random projections, see Hastie et al. (2019).

If $m \leqslant n$, the estimator is obtained from the normal equations $S^\top X^\top X S \hat\eta = S^\top X^\top y$, and can be expanded using $\theta_*$ as follows:

$$\begin{aligned}
\hat\theta &= S\hat\eta = S(S^\top X^\top XS)^{-1}S^\top X^\top y \\
&= S(S^\top X^\top XS)^{-1}S^\top X^\top X\theta_* + S(S^\top X^\top XS)^{-1}S^\top X^\top \varepsilon \ \text{ using } y = X\theta_* + \varepsilon, \\
&= N\theta_* + S(S^\top X^\top XS)^{-1}S^\top X^\top \varepsilon,
\end{aligned}$$

with $N = S(S^\top X^\top XS)^{-1}S^\top X^\top X$. Conditioned on $S$ and $X$, the expected excess risk is equal to:

$$\begin{aligned}
\mathbb{E}_\varepsilon\big[\mathcal{R}(\hat\theta)\big] &= \sigma^2 \operatorname{tr}\big(XS(S^\top X^\top XS)^{-1}S^\top \Sigma S(S^\top X^\top XS)^{-1}S^\top X^\top\big) + \big\|\Sigma^{1/2}(N\theta_* - \theta_*)\big\|_2^2 \\
&= \sigma^2 \operatorname{tr}\big(S^\top \Sigma S(S^\top X^\top XS)^{-1}\big) + \operatorname{tr}\big((N\theta_* - \theta_*)^\top \Sigma(N\theta_* - \theta_*)\big).
\end{aligned}$$

For the variance term (first term above), for $S$ fixed, since $X$ has a Gaussian distribution, the matrix $S^\top X^\top XS$ is distributed as a Wishart distribution with parameter $S^\top \Sigma S$ and $n$ degrees of freedom (see, e.g., Haff, 1979, for computations of moments of the Wishart distribution). Thus, if $n > m+1$, we have:

$$\mathbb{E}_X\big[(S^\top X^\top XS)^{-1}\big] = \frac{1}{n-m-1}(S^\top \Sigma S)^{-1},$$

which in turn implies $\mathbb{E}_{S,X,\varepsilon}\big[\mathcal{R}^{(\text{var})}(\hat\theta)\big] = \mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{var})}(\hat\theta)\big] = \dfrac{\sigma^2 m}{n-m-1}$, independently of the choice of the sketching matrix $S$.

For the bias term, the computation is more involved. We expand

$$\begin{aligned}
\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \mathbb{E}_X\big[\operatorname{tr}\big((N\theta_* - \theta_*)^\top \Sigma(N\theta_* - \theta_*)\big)\big] \\
&= \theta_*^\top \Sigma\theta_* + 2\theta_*^\top \Sigma S(S^\top X^\top XS)^{-1}S^\top X^\top X\theta_* \\
&\quad + \theta_*^\top X^\top XS(S^\top X^\top XS)^{-1}S^\top \Sigma S(S^\top X^\top XS)^{-1}S^\top X^\top X\theta_*.
\end{aligned}$$

To compute the expectation, we will first condition on $XS$ and use the Gaussian conditioning formulas from Section 1.1.3, which leads to, for any matrices $A$ and $B$ of appropriate sizes (proof left as an exercise):

$$\begin{aligned}
\mathbb{E}\big[X|XS\big] &= XS(S^\top \Sigma S)^{-1}S^\top \Sigma = X\Pi_S \\
\mathbb{E}\big[\operatorname{tr}(AX^\top BX)|XS\big] &= \operatorname{tr}\big(A\Pi_S^\top X^\top BX\Pi_S\big) + \operatorname{tr}(B)\operatorname{tr}(A\Sigma(I-\Pi_S)).
\end{aligned}$$

This leads to, with $S$ assumed fixed, using the identities above:

$$\begin{aligned}
\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top \Sigma\theta_* + \mathbb{E}_X\big[2\theta_*^\top \Sigma S(S^\top X^\top XS)^{-1}S^\top X^\top X\Pi_S\theta_*\big] \\
&+ \mathbb{E}_X\big[\theta_*^\top \Pi_S^\top X^\top XS(S^\top X^\top XS)^{-1}S^\top \Sigma S(S^\top X^\top XS)^{-1}S^\top X^\top X\Pi_S\theta_*\big] \\
&+ \mathbb{E}_X\big[\operatorname{tr}\big(XS(S^\top X^\top XS)^{-1}S^\top \Sigma S(S^\top X^\top XS)^{-1}S^\top X^\top\big)\cdot \operatorname{tr}\big(\theta_*\theta_*^\top \Sigma(I-\Pi_S)\big)\big].
\end{aligned}$$

Now, using properties of $\Pi_S$, we get:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{bias})}(\hat\theta)\big] &= \theta_*^\top \Sigma\theta_* + \mathbb{E}_X\big[2\theta_*^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\theta_*\big] \\ &\quad + \mathbb{E}_X\big[\theta_*^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\theta_*\big] \\ &\quad + \mathbb{E}_X\big[\operatorname{tr}\big(S^\top\Sigma S(S^\top X^\top XS)^{-1}\big)\cdot\operatorname{tr}\big(\theta_*\theta_*^\top\Sigma(I-S(S^\top\Sigma S)^{-1}S^\top\Sigma)\big)\big].\end{aligned}$$

We can now group some terms and use $\Pi_S^\top\Sigma\Pi_S = \Sigma\Pi_S$ to get:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\Big[1+\mathbb{E}_X\big[\operatorname{tr}\big(S^\top\Sigma S(S^\top X^\top XS)^{-1}\big)\big]\Big] \\ &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\Big(1+\operatorname{tr}\Big(\frac{1}{n-m-1}(S^\top\Sigma S)^{-1}S^\top\Sigma S\Big)\Big),\end{aligned}$$

using expectations of Wishart random variables. Overall, we get:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\frac{n-1}{n-m-1} \\ &= \theta_*^\top\big(\Sigma-\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\big)\theta_*\cdot\frac{n-1}{n-m-1}.\end{aligned}$$

We can further bound the bias term; we have, for $S$ Gaussian, following the same reasoning as in Section 10.2.2:

$$\begin{aligned} & \mathbb{E}_S\big[\theta_*^\top(I-\Pi_S)^\top\Sigma(I-\Pi_S)\theta_*\big] \\ = \ & \mathbb{E}_S\big[\min_{\eta\in\mathbb{R}^m}(\theta_*-S\eta)^\top\Sigma(\theta_*-S\eta)\big] \text{ by definition of } \Pi_S, \\ \leqslant \ & \mathbb{E}_S\big[\min_{\xi\in\mathbb{R}^d}(\theta_*-SS^\top\xi)^\top\Sigma(\theta_*-SS^\top\xi)\big] \text{ with } \eta \text{ constrained in the column space of } S^\top, \\ \leqslant \ & \min_{\xi\in\mathbb{R}^d}\mathbb{E}_S\big[(\theta_*-SS^\top\xi)^\top\Sigma(\theta_*-SS^\top\xi)\big] \text{ by swapping minimum and expectation}, \\ = \ & \min_{\xi\in\mathbb{R}^d}\Big(\theta_*^\top\Sigma\theta_* - 2\xi^\top\mathbb{E}[SS^\top]\Sigma\theta_* + \xi^\top\mathbb{E}[SS^\top\Sigma SS^\top]\xi\Big) \text{ by developing}, \\ = \ & \theta_*^\top\Big(\Sigma-\Sigma\mathbb{E}[SS^\top]\big(\mathbb{E}[SS^\top\Sigma SS^\top]\big)^{-1}\mathbb{E}[SS^\top]\Sigma\Big)\theta_* \text{ by minimizing in closed form}, \\ = \ & \theta_*^\top\Big(\Sigma-m\Sigma\big((m+1)\Sigma+\operatorname{tr}(\Sigma)I\big)^{-1}\Sigma\Big)\theta_* \text{ using Wishart expectations}, \\ = \ & \theta_*^\top(\Sigma+\operatorname{tr}(\Sigma)I)\big((m+1)\Sigma+\operatorname{tr}(\Sigma)I\big)^{-1}\Sigma\theta_* \\ \leqslant \ & \frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\theta_*^\top\Big(\Sigma+\frac{\operatorname{tr}(\Sigma)}{m+1}I\Big)^{-1}\Sigma\theta_* \leqslant \frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\|\theta_*\|_2^2, \text{ using } \Sigma\preccurlyeq\operatorname{tr}[\Sigma]\cdot I.\end{aligned}$$

Overall, for the underparameterized regime, we obtain an upper-bound equal to $\frac{1}{1-m/n}$ times $\frac{\sigma^2 m}{n}+\frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\|\theta_*\|_2^2$, which a similar excess risk bound as for ridge regression from Section 3.6 and Section 7.6.4, with $\operatorname{tr}(\Sigma)/m$ playing the role of the regularization parameter, but with the extra term $1/(1-m/n)$ due to the random design setting and the lack of regularization. This leads to a classical bias-variance trade-off with a U-shaped curve. See Bach (2023a) for sharper results.

**Overparameterized regime.** In the overparameterized regime, when $m \geqslant n$, then the kernel matrix is almost surely invertible, and the minimum $\ell_2$-norm interpolator $\hat\theta$ is equal to (using the formulas from Section 12.1.1)

$$\hat\theta = S\hat\eta = SS^\top X^\top (XSS^\top X^\top)^{-1}(X\theta_* + \varepsilon).$$

We can decompose the expectation with respect to $\varepsilon$ of the excess risk $\mathcal{R}(\hat\theta)$ as follows:

$$\begin{aligned}\mathbb{E}_\varepsilon\big[\mathcal{R}(\hat\theta)\big] &= \sigma^2 \operatorname{tr}\big((XSS^\top X^\top)^{-1} XSS^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1}\big) \\ &\qquad + \big\|\Sigma^{1/2}\big(SS^\top X^\top (XSS^\top X^\top)^{-1}X - I\big)\theta_*\|_2^2.\end{aligned}$$

We can now use the same reasoning as in the under-parameterized regime, but now taking expectation with respect to $S$ (with $X$ fixed). We have for any symmetric matrices $A$ and $B$ of compatible sizes (proof left as an exercise):

$$\begin{aligned}\mathbb{E}\big[\operatorname{tr}(ASBS^\top)|S^\top X^\top\big] &= \operatorname{tr}\big(X^\top (XX^\top)^{-1} XAX^\top (XX^\top)^{-1} XSBS^\top\big) \\ &\qquad + \operatorname{tr}(B)\operatorname{tr}\big[A(I - X^\top (XX^\top)^{-1}X)\big] \\ \mathbb{E}\big[S|S^\top X^\top\big] &= X^\top (XX^\top)^{-1} XS.\end{aligned}$$

Therefore, for the variance term proportional to $\sigma^2$, for which we take $A = \Sigma$ and $B = S^\top X^\top (XSS^\top X^\top)^{-2} XS$, we obtain two parts from the identities above. The second part of the variance term becomes:

$$\begin{aligned}& \operatorname{tr}\big[S^\top X^\top (XSS^\top X^\top)^{-2} XS\big] \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big] \\ =\ & \operatorname{tr}\big[(XSS^\top X^\top)^{-1}\big] \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big].\end{aligned}$$

The first part of the variance term is:

$$\begin{aligned}& \operatorname{tr}\big(X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} XSS^\top X^\top (XSS^\top X^\top)^{-2} XSS^\top\big) \\ =\ & \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big).\end{aligned}$$

Thus, using the expectation of the inverse Wishart distribution, the variance term is

$$\begin{aligned}\mathbb{E}_{\varepsilon,S}[\mathcal{R}^{(\text{var})}(\hat\theta)] &= \sigma^2 \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big) \\ &\qquad + \sigma^2 \frac{\operatorname{tr}\big((XX^\top)^{-1}\big)}{m - n - 1} \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big].\end{aligned}$$

For the bias term, we have:

$$\begin{aligned}& \big\|\Sigma^{1/2}\big(SS^\top X^\top (XSS^\top X^\top)^{-1}X - I\big)\theta_*\|_2^2 \\ =\ & \theta_*^\top \Sigma\theta_* + \theta_*^\top X^\top (XSS^\top X^\top)^{-1} XSS^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\ & \qquad - 2\theta_*^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\ =\ & \theta_*^\top \Sigma\theta_* + \operatorname{tr}(ASBS^\top) - 2\theta_*^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*,\end{aligned}$$

with $A = \Sigma$, and $B = S^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XS$, with conditional expectation given $XS$, simplifying the product $XSS^\top X^\top (XSS^\top X^\top)^{-1} = I$, and using $XSBS^\top X^\top = X\theta_*\theta_*^\top X^\top$:

$$\begin{aligned}
&\theta_*^\top \Sigma\theta_* \\
&+\mathrm{tr}\left(X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XSS^\top\right) \\
&+\mathrm{tr}(S^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XS) \cdot \mathrm{tr}\left[\Sigma(I - X^\top (XX^\top)^{-1} X)\right] \\
&-2\theta_*^\top \Sigma X^\top (XX^\top)^{-1} XSS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\
= \;& \theta_*^\top \Sigma\theta_* \\
&+\theta_*^\top X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} X\theta_* \\
&+\mathrm{tr}((XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top) \cdot \mathrm{tr}\left[\Sigma(I - X^\top (XX^\top)^{-1} X)\right] \\
&-2\theta_*^\top \Sigma X^\top (XX^\top)^{-1} X\theta_* \text{ by simplifying,} \\
= \;& \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma(I - X^\top (XX^\top)^{-1} X)\theta_* \\
&+\mathrm{tr}((XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top) \cdot \mathrm{tr}\left[\Sigma(I - X^\top (XX^\top)^{-1} X)\right],
\end{aligned}$$

by grouping terms. This leads to, by rearranging terms,

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}^{(\text{bias})}(\hat\theta)] \;=\;& \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma(I - X^\top (XX^\top)^{-1} X)\theta_* \\
&+\frac{1}{m-n-1}\theta_*^\top X^\top (XX^\top)^{-1} X\theta_* \cdot \mathrm{tr}\left[\Sigma(I - X^\top (XX^\top)^{-1} X)\right].
\end{aligned}$$

Pulling together bias and variance, when $m$ tends to infinity, we get the performance:

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}_\infty(\hat\theta)] \;=\;& \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma(I - X^\top (XX^\top)^{-1} X)\theta_* \\
&+\sigma^2\,\mathrm{tr}\left((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\right).
\end{aligned}$$

Overall, we get:

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}(\hat\theta)] \;=\;& \mathbb{E}_{\varepsilon,S}[\mathcal{R}_\infty(\hat\theta)] + \frac{1}{m-n-1}\,\mathrm{tr}\left[\Sigma(I - X^\top (XX^\top)^{-1} X)\right] \\
&\cdot\left(\theta_*^\top X^\top (XX^\top)^{-1} X\theta_* + \sigma^2\,\mathrm{tr}((XX^\top)^{-1})\right).
\end{aligned}$$

Thus, as a function of $m$, we get a descent curve on the right of $m = n$. See Bach (2023a) for a detailed expression obtained after taking the expectation with respect to $X$. While the limiting bias term typically has a better value than for the under-parameterized regime, for the variance term, the limit when $m$ tends to $+\infty$ does not always go to zero when $n$ tends to infinity. See Bartlett et al. (2020) for conditions under which the end of the double descent curve can lead to good performance when $\sigma^2 > 0$. See illustration in Figure 12.4.

## 12.3 Global convergence of gradient descent

In Section 9.2.1, we alluded to the property of gradient descent for overparameterized neural networks, which converges to a global minimum of the objective function despite being non-convex. We present more formal arguments in this section, with a general result without proof, as well as a detailed proof for linear neural networks.

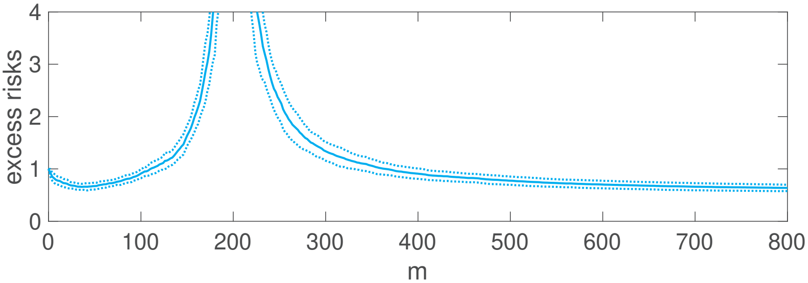


Figure 12.4: Example of a double descent curve, for linear regression with random projections with $n = 200$ observations, in dimension $d = 400$ and a non-isotropic covariance matrix. The data are normalized so that predicting zero leads to an excess risk of 1 and the noise so that the optimal expected risk is 1/4. The empirical estimate is obtained by sampling 20 datasets and 20 different random projections from the same distribution and averaging the corresponding excess risks

### 12.3.1 Mean field limits

In this section, we present results from Chizat and Bach (2018), following closely the exposition from Bach and Chizat (2022).[7] More precisely, we consider neural networks with one infinitely wide hidden layer, and we first rescale the prediction function by $1/m$ (which can be obtained by rescaling all output weights by $1/m$) and express it explicitly as an empirical average as

$$f(x) = \frac{1}{m} \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j)$$

where $\eta_j \in \mathbb{R}$ is the output weight associated to the $j$-th neuron, and $(w_j, b_j) \in \mathbb{R}^{d+1}$ the corresponding vector of input weights. The key observation is that the prediction function $h$ is the average of $m$ prediction functions $x \mapsto \eta_j \sigma(w_j^\top x + b_j)$, for $j = 1, \dots, m$, with *no sharing of the parameters* (which is not true if extra layers of hidden neurons are added).

To highlight this parameter separability, we define $v_j = \big[\eta_j, w_j^\top, b_j\big]^\top \in \mathbb{R}^{d+2}$ the set of weights associated to the hidden neuron $j \in \{1, \dots, m\}$, and we define the function $\Psi(v) = \Psi(\eta, w^\top, b) : x \mapsto \eta\sigma(x^\top w + b)$, so that the prediction function $h$ is parameterized by $v_1, \dots, v_m \in \mathbb{R}^{d+2}$, which is now

$$h = \frac{1}{m} \sum_{j=1}^{m} \Psi(v_j). \tag{12.13}$$

The expected risk is of the form

$$\mathcal{R}(h) = \mathbb{E}\big[\ell(y, h(x))\big],$$

[7] See also https://www.di.ens.fr/~fbach/ltfp/wide_convergence.html.

which is convex in $h$ for convex loss functions (which is the case throughout the book, even for neural networks, such as the logistic or square loss), but typically non convex in $V = (v_1, \dots, v_m)$. Note that the resulting problem of minimizing a convex function $\mathcal{R}(h)$ for $h = \frac{1}{m}\sum_{j=1}^m \Psi(v_j)$ applies beyond neural networks, for example, for sparse deconvolution (Chizat, 2021).

**Reformulation with probability measures.** We now define by $\mathcal{P}(\mathcal{V})$ the set of probability measures on $\mathcal{V} = \mathbb{R}^{d+2}$. We can rewrite Eq. (12.13) as

$$h = h(\cdot, v_1, \dots, v_m) = \int_{\mathcal{V}} \Psi(v) d\mu(v),$$

where $\mu = \frac{1}{m}\sum_{j=1}^m \delta_{v_j}$ is an average of Dirac measures at each $v_1, \dots, v_m \in \mathcal{V}$. Following a physics analogy, we will refer to each $w_j$ as a *particle*. When the number $m$ of particles grows, then the empirical measure $\frac{1}{m}\sum_{j=1}^m \delta_{w_j}$ may converge in distribution to a probability measure with a density, often referred to as a *mean field* limit. Our main reformulation will thus consider an optimization problem over probability measures.

The optimization problem we are faced with is equivalent to

$$\inf_{\mu \in \mathcal{P}(\mathcal{V})} \mathcal{R}\Big(\int_{\mathcal{V}} \Psi(v) d\mu(v)\Big), \tag{12.14}$$

with the constraint that $\mu$ is an average of $m$ Dirac measures. We now follow a long line of work in statistics and signal processing (Barron, 1993; Kurková and Sanguineti, 2001), and consider the optimization problem *without this constraint*, and relate optimization algorithms for finite but large $m$ (thus acting on $V = (v_1, \dots, v_m)$ in $\mathcal{V}^m$) to a well-defined algorithm in $\mathcal{P}(\mathcal{V})$, like we already did in Section 9.3.2.

Note that we now have a convex optimization problem, with a convex objective in $\mu$ over a convex set (all probability measures). However, it is still an infinite-dimensional space that requires dedicated finite-dimensional algorithms. In this section, we focus on gradient descent on $w$, corresponding to standard practice in neural networks (e.g., back-propagation). For algorithms based on classical convex optimization algorithms such as the Frank-Wolfe algorithm, see Section 9.3.6.

**From gradient descent to gradient flow.** Our general goal is to study the gradient descent recursion on $V = (v_1, \dots, v_m) \in \mathcal{V}^m$, defined as

$$V_k = V_{k-1} - \gamma m G'(V_{k-1}), \tag{12.15}$$

with

$$G(V) = \mathcal{R}\big(h(\cdot, v_1, \dots, v_m)\big) = \mathcal{R}\Big(\frac{1}{m}\sum_{j=1}^m \Psi(v_j)\Big).$$

In the context of neural networks, this is exactly the back-propagation algorithm. We include the factor $m$ in the step-size to obtain a well-defined limit when $m$ tends to infinity (see below).

For convenience in the analysis, we look at the limit when the step-size $\gamma$ goes to zero. If we consider a function $W : \mathbb{R} \to \mathcal{V}^m$, with values $W(k\gamma) = V_k$ at $t = k\gamma$, and we interpolate linearly between these points, then we obtain exactly the standard Euler discretization of the ordinary differential equation (ODE) (Suli and Mayers, 2003):

$$\dot{W} = -mG'(W). \tag{12.16}$$

This gradient flow will be our main focus in this paper. As highlighted above, and with extra regularity assumptions, it is the limit of the gradient recursion in Eq. (12.15) for vanishing step-sizes $\gamma$. Moreover, under appropriate conditions, stochastic gradient descent, where we only observe an unbiased noisy version of the gradient, also leads in the limit $\gamma \to 0$ to the same ODE (Kushner and Yin, 2003). This allows us to apply our results to probability distributions of the data $(x, y)$, which are not the observed empirical distribution but the unseen test distribution, where, for a single over the data, the stochastic gradients come from the gradient of the loss from a single observation.

**Wasserstein gradient flow.** Above, we have described a general framework where we want to minimize a function $F$ defined on probability measures:

$$F(\mu) = \mathcal{R}\Big(\int_{\mathcal{V}} \Psi(v) d\mu(v)\Big), \tag{12.17}$$

with an algorithm minimizing $G(v_1, \dots, v_m) = \mathcal{R}\big(\frac{1}{m}\sum_{j=1}^m \Psi(v_j)\big)$ through the gradient flow $\dot{V} = -mG'(V)$, with $V = (v_1, \dots, v_m)$.

As shown in a series of works concerned with the infinite width limit of two-layer neural networks (Nitanda and Suzuki, 2017; Chizat and Bach, 2018; Mei et al., 2018; Sirignano and Spiliopoulos, 2020; Rotskoff and Vanden-Eijnden, 2018), this converges to a well-defined mathematical object called a Wasserstein gradient flow (Ambrosio et al., 2008). This is a gradient flow derived from the Wasserstein metric on the set of probability measures, which is defined as (Santambrogio, 2015),

$$W_2(\mu, \nu)^2 = \inf_{\gamma \in \Pi(\mu,\nu)} \int \|v - w\|_2^2 d\gamma(v, w),$$

where $\Pi(\mu, \nu)$ is the set of probability measures on $\mathcal{V} \times \mathcal{V}$ with marginals $\mu$ and $\nu$. In a nutshell, the gradient flow is defined as the limit when $\gamma$ tends to zero of the extension of the following discrete-time dynamics:

$$\mu(t + \gamma) = \inf_{\nu \in \mathcal{P}(\mathcal{V})} F(\nu) + \frac{1}{2\gamma} W_2(\mu(t), \nu)^2.$$

When applying such a definition in a Euclidean space with the Euclidean metric, we recover the usual gradient flow $\dot{\mu} = -F'(\mu)$, but here with the Wasserstein metric, this defines a specific flow on the set of measures. When the initial measure is a weighted sum of Diracs, this is precisely asymptotically (when $\gamma \to 0$) equivalent to backpropagation. When initialized with an arbitrary probability measure, we obtain a partial differential equation (PDE), satisfied in the sense of distributions. Moreover, when the sum of Diracs converges in distribution to some measure, the flow converges to the solution of the PDE. More precisely, assuming $\Psi : \mathbb{R}^{d+1} \to \mathcal{H}$ a Hilbert space (in our neural network example, $\mathcal{H}$ is the space of square-integrable functions on $\mathbb{R}^d$), and $\mathcal{R}'(h) \in \mathcal{H}$ the gradient of $\mathcal{R}$, we consider the *mean potential*

$$J(v|\mu) = \Big\langle \Psi(v), \mathcal{R}'\Big( \int_{\mathcal{V}} \Psi(w) d\mu(w) \Big) \Big\rangle, \tag{12.18}$$

so that the gradient flow equation on each particle becomes $\dot{v}_j = -J'(v_j|\mu)$ (where $\mu$ is the time-dependent aggregation of all particles).

The PDE is then the continuity equation (see, e.g., Evans, 2022):

$$\partial_t \mu_t(v) = \operatorname{div}(\mu_t(v) J'(v|\mu_t)), \tag{12.19}$$

which is understood in the sense of distributions. The following result formalizes this behavior (see Chizat and Bach (2018) for details and a more general statement).

**Theorem 12.1** *Assume that $\mathcal{R} : \mathcal{H} \to [0, +\infty[$ and $\Psi : \mathcal{V} = \mathbb{R}^{d+2} \to \mathcal{H}$ are (Fréchet) differentiable with Lipschitz differentials, and that $R$ is Lipschitz on its sublevel sets. Consider a sequence of initial weights $(v_j(0))_{j\geq 1}$ contained in a compact subset of $\mathcal{V}$ and let $\mu_{t,m} = \frac{1}{m}\sum_{j=1}^m v_j(t)$ where $(v_1(t), \dots, v_m(t))$ solves the ODE (12.16). If $\mu_{0,m}$ weakly converges to some $\mu_0 \in \mathcal{P}(\mathcal{V})$ then $\mu_{t,m}$ weakly converges to $\mu_t$ where $(\mu_t)_{t\geq 0}$ is the unique weakly continuous solution to (12.19) initialized with $\mu_0$.*

In the following paragraph, we will study the solution of this PDE (i.e., the Wasserstein gradient flow), interpreting it as the limit of the gradient flow in Eq. (12.16) when the number of particles $m$ tends to infinity.

**Global convergence.** We consider the Wasserstein gradient flow defined above, which leads to the PDE in Eq. (12.19). We aim to understand when we can expect that when $t \to \infty$, $\mu_t$ converges to a global minimum of $F$ defined in Eq. (12.17). Obtaining a global convergence result is not out of the question because $F$ is a convex functional defined on the convex set of probability measures. However, it is nontrivial because, with our choice of the Wasserstein geometry on measures, which allows an approximation through particles, the flow has some stationary points that are not the global optimum.

We only consider an informal general result without technical assumptions before referring to Bach and Chizat (2022) for a formal simplified result and Chizat and Bach (2018) for the general result.

**Theorem 12.2 (Informal)** *If the support of the initial distribution includes all directions in $\mathbb{R}^{d+1}$, and if the function $\Psi$ is positively 2-homogeneous, then if the Wasserstein gradient flow weakly converges to a distribution, it can only be to a global optimum of $F$.*

Chizat and Bach (2018) present another version of this result that allows for *partial* homogeneity (e.g., with respect to a subset of variables) of degree 1 is proven, at the cost of a more technical assumption on the initialization. For neural networks, we have $\Psi(\eta, w^\top, b)(x) = m\eta\sigma(w^\top x + b)$, and this more general version applies. For the classical ReLU activation function $u \mapsto \max\{0, u\}$, we get a positively 2-homogeneous function, as required in the previous statement. A simple way to spread all directions is to initialize neural network weights from Gaussian distributions, which is standard in applications (Goodfellow et al., 2016).

**From qualitative to quantitative results?** Our result states that for infinitely many particles, we can only converge to a global optimum (note that we cannot show that the flow always converges). However, it is only a qualitative result in comparison with what is known for convex optimization problems in Chapter 5:

- This is only for $m = +\infty$, and we cannot provide an estimation of the number of particles needed to approximate the mean-field regime that is not exponential in $t$ (see such results, e.g., by Mei et al., 2019).
- We cannot provide an estimation of the performance as a function of time that would give an upper bound on the running time complexity.

Moreover, our result does not apply beyond a single hidden layer, and understanding the non-linear infinite width limits for deeper networks is an important research area (Nguyen and Pham, 2020; Araújo et al., 2019; Fang et al., 2021; Hanin and Nica, 2019; Sirignano and Spiliopoulos, 2021; E and Wojtowytsch, 2020; Yang and Hu, 2020).

In the remainder of this section, to present a simpler analysis, we focus on linear neural networks and first reformulate them as optimizing over positive definite matrices.

### 12.3.2 From linear networks to positive definite matrices

We consider "linear" neural networks, that is, neural networks with no activation function. For example, for $x \in \mathbb{R}^d$, we consider $f(x) = UV^\top x \in \mathbb{R}^k$, where $U \in \mathbb{R}^{k\times m}$ and $V \in \mathbb{R}^{d\times m}$. This is a linear function $f(x) = \Theta x$, with $\Theta$ of the form $\Theta = UV^\top \in \mathbb{R}^{k\times d}$. Assuming that we minimize some smooth convex risk function $G : \mathbb{R}^{k\times d} \to \mathbb{R}$, we aim to minimize $G(UV^\top)$.

It can be rewritten as the function $G$ applied to a linear projection of the matrix $\binom{U}{V}\binom{U}{V}^\top = \begin{pmatrix} UU^\top & UV^\top \\ VU^\top & VV^\top \end{pmatrix}$, which is of the form $WW^\top$ with $W = \binom{U}{V} \in \mathbb{R}^{(k+d)\times m}$. Thus, we can analyze instead the minimization of functions of the form $G(WW^\top)$ for $W \in \mathbb{R}^{d\times m}$, where $G$ is a smooth convex function defined on positive semidefinite matrices of size $d$.

The goal of this section is now to minimize a convex function $G$ over positive-semidefinite (PSD) matrices, using plain gradient descent techniques on a non-linear parameterization of such matrices. This is done to illustrate optimization for neural networks, noting that faster algorithms based on projected gradient descent presented in Chapter 5 could also be used. We already studied a special case in Section 12.1.3, where the ordinary differential equation could be integrated in closed form, while, here, we rely on more qualitative arguments.

### 12.3.3 Global convergence for positive definite matrices

We consider a twice continuously differentiable convex function $G : \mathbb{R}^{d\times d} \to \mathbb{R}$ (which only needs to be defined on symmetric matrices). We consider $m$ vectors $w_1, \dots, w_m \in \mathbb{R}^d$ put into a matrix $W = (w_1, \dots, w_m) \in \mathbb{R}^{d\times m}$, and the cost function $F(W) = G\big(WW^\top\big)$, where we have $WW^\top = \sum_{j=1}^m w_j w_j^\top$. This is thus an instance of the framework developed in Section 12.3.1, but, since the space of quadratic functions is finite-dimensional, there is no need to let $m$ tend to infinity, and we can keep $m \leqslant d$.

We consider the gradient flow $\dot{W} = -\frac{1}{2}F'(W)$, that is, $W'(t) = -\frac{1}{2}F'(W(t))$, where the factor $\frac{1}{2}$ was added so simplify formulas later. Since $F$ is twice differentiable, this ordinary differential equation (ODE) is defined for all $t \geqslant 0$. To compute the gradient of $F$, we perform an asymptotic expansion as follows:

$$\begin{aligned} F(W+\Delta) &= G\big(WW^\top + \Delta W^\top + W\Delta^\top + o(\|\Delta\|_2)\big) \\ &= F(W) + \operatorname{tr}\big[G'\big(WW^\top\big)\big(\Delta W^\top + W\Delta^\top\big)\big] + o(\|\Delta\|_2) \\ &= F(W) + 2\operatorname{tr}\big[\Delta^\top G'\big(WW^\top\big)W\big] + o(\|\Delta\|_2), \end{aligned}$$

so that $F'(W) = 2G'\big(WW^\top\big)W$, and the flow becomes $\dot{W} = -G'\big(WW^\top\big)W$. By projecting onto each of the $m$ columns of $W$, this leads to the following flow for each "particle" $w_j \in \mathbb{R}^d$:

$$\dot{w}_j = -G'\big(WW^\top\big)w_j,$$

which is a linear ODE, but with a time-dependent matrix $G'\big(WW^\top\big)$ which depends on the aggregation of all particles since $WW^\top = \sum_{j=1}^m w_j w_j^\top$.

We denote $M = WW^\top$ and $A = G'(M)$, which are functions of time defined for all time $t \geqslant 0$. We then have:

$$\dot{M} = \dot{W}W^\top + W\dot{W}^\top = -G'(M)M - MG'(M) = -AM - MA.$$

**Preservation of rank.** If at time zero $M = WW^\top$ has full rank, then the rank is preserved throughout the flow. This is a simple consequence of the ODE for $r(M) = \log\det(M)$, equal to

$$\dot{r} = \operatorname{tr}\big[M^{-1}\dot{M}\big] = \operatorname{tr}\big[M^{-1}(-AM - MA)\big] = -2\operatorname{tr}(A).$$

Thus, since $A$ is continuous for all positive times, the log determinant is finite for all times as soon as it exists at initialization, and we thus obtain a full rank matrix. If $m \geqslant d$, which corresponds to an over-parameterized situation, and the columns of $W$ are initialized randomly (e.g., from a standard Gaussian random variable), then $WW^\top$ indeed has full rank.

**Exercise 12.3** *(♦) Show that if at initialization, $M = WW^\top$ has rank $r \leqslant \min\{d, m\}$, then $M$ has rank $r$ at all times.*

**Global optimality conditions.** The problem of minimizing $G(M)$ over PSD matrices has the following optimality condition: (a) $\mathrm{tr}[MG'(M)] = 0$ and (b) $G'(M) \succcurlyeq 0$. Note that once (b) is satisfied, (a) is equivalent to $MG'(M) = 0$.[8]

- *Necessary conditions (no need for convexity).* If $M$ is optimal, then for all $\Delta$ such that $M + \Delta \succcurlyeq 0$, $G(M + \Delta) - G(M) \geqslant 0$. When $\Delta$ is small, this leads to $\mathrm{tr}[\Delta G'(M)] \geqslant 0$.

  Taking $\Delta$ small along $-M$ or $M$, we get: $\mathrm{tr}[MG'(M)] = 0$ as necessary condition.

  Taking $\Delta = uu^\top$ for all $u \in \mathbb{R}^d$, we get $G'(M) \succcurlyeq 0$ as a necessary condition.

- *Sufficient conditions.* If the conditions are met, then for any matrix $N \succcurlyeq 0$, we get from the subgradient inequality for the convex function $R$:

  $$G(N) \geqslant G(M) + \mathrm{tr}\big[G'(M)(N - M)\big].$$

  Using condition (a), we get : $\mathrm{tr}\big[G'(M)M\big] = 0$, while condition (b) ensures that $\mathrm{tr}\big[G'(M)N\big] \geqslant 0$. Thus, $M$ is a global optimum.

If $M$ is invertible, the optimality conditions simplify to $G'(M) = 0$.

**Global convergence. (♦)** If the flow in $M$ is initialized with a full-rank matrix and converges to some $M_\infty$,[9] we now show that it satisfies the two optimality conditions above (and thus, it has to be a global optimum). Note that while we know that $M$ is invertible for all time $t \geqslant 0$, it is often not the case for $M_\infty$ (see examples below).

The first condition (a) is a direct consequence of $-G'(M_\infty)M_\infty - M_\infty G'(M_\infty) = 0$ (by taking the trace), which is satisfied at convergence (this is the stationary condition, stating that all particles stop). The difficult part is to show the second condition (b), which can be interpreted as ensuring that no other potential particles could enter and increasing the rank of $M$ while reducing the cost function.

We now assume that $A_\infty = G'(M_\infty)$ is not PSD, that is, $\lambda_{\min}(A_\infty) < 0$. We choose $\eta > 0$ such that $\lambda_{\min}(A_\infty) < -\eta$, and $-\eta$ is not an eigenvalue of $A_\infty$ (which is possible because there are at most $d$ distinct eigenvalues). This implies that for $u$ such that $\|u\|_2 = 1$ and $u^\top A_\infty u = -\eta$,

$$\eta = -u^\top A_\infty u < \|u\|_2 \|A_\infty u\|_2 = \|A_\infty u\|_2$$

by Cauchy-Schwarz inequality and the impossibility of having $A_\infty u = -\eta u$ (which is the equality condition for Cauchy-Schwarz inequality). We denote by $\beta > \eta$ the minimal value of such $\|A_\infty u\|_2$ (for all $u$ that satisfies $\|u\|_2 = 1$ and $u^\top A_\infty u = -\eta$).

The idea is to show that sufficiently close to convergence, once a particle has a direction in

$$K = \big\{u \in \mathbb{R}^d,\ \|u\|_2 = 1,\ u^\top A_\infty u < -\eta\big\},$$

[8] For two PSD matrices $A$ and $B$ of the same sizes, $AB = 0 \Leftrightarrow \mathrm{tr}(AB) = 0$.

[9] It does under basic assumptions on $R$, such as piecewise analyticity, see Bolte et al. (2006).

its direction never gets out of $K$, and that it leads to a contradiction (the set $K$ is not empty because $\lambda_{\min}(A_\infty) < -\eta$).

We now introduce the time dependence explicitly.

**Choice of particle close to convergence (♦).** We have $M(t) \to M_\infty$. Thus there exists $t_0$ such that $\|A(t) - A_\infty\|_{\rm op} \leqslant \varepsilon$, for all $t \geqslant t_0$, with $\varepsilon$ well chosen (small enough).

Let $y_0 \in \mathbb{R}_+K$, $y_0 \neq 0$ (it exists since $K$ is not empty). Since $W(t_0) \in \mathbb{R}^{d\times m}$ has full rank equal to $d$, then there exists $\alpha_0 \in \mathbb{R}^m$ such that $y_0 = W(t_0)\alpha_0$.

We then consider a particle $z(t) = W(t)\alpha_0 \in \mathbb{R}^d$. By construction, $z'(t) = \dot{W}(t)\alpha_0 = -A(t)W(t)\alpha_0 = -A(t)z(t)$, and $z(t_0) = y_0 \in \mathbb{R}_+K$. We now show by contradiction that we must have $z(t) \in \mathbb{R}_+K$ for all $t \geqslant t_0$. If $t_1$ is the smallest $t \geqslant t_0$ such that $z(t) \notin \mathbb{R}_+K$ (which is assumed to exist by contradiction), then by continuity, $z(t_1) \in \mathbb{R}_+\partial K$, that is, $z(t_1)^\top A_\infty z(t_1) = -\eta z(t_1)^\top z(t_1)$. We then have, with $z_1 = z(t_1)$, and using that $z'(t_1) = -A(t_1)z(t_1)$:

$$\begin{aligned}
\frac{d}{dt}\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)}\Big|_{t=t_1} &= 2\frac{z(t_1)^\top A_\infty z'(t_1)}{z(t_1)^\top z(t_1)} - 2\frac{z(t_1)^\top A_\infty z(t_1)}{z(t_1)^\top z(t_1)}\frac{z'(t_1)^\top z(t_1)}{z(t_1)^\top z(t_1)} \\
&= -2\frac{z_1^\top A_\infty A(t_1) z_1}{z_1^\top z_1} + 2\frac{z_1^\top A_\infty z_1}{z_1^\top z_1}\frac{z_1^\top A(t_1) z_1}{z_1^\top z_1} \\
&= -2\frac{z_1^\top A_\infty^2 z_1}{z_1^\top z_1} + 2\frac{z_1^\top A_\infty(A_\infty - A(t_1))z_1}{z_1^\top z_1} + 2\frac{z_1^\top A_\infty z_1}{z_1^\top z_1}\frac{z_1^\top A(t_1) z_1}{z_1^\top z_1}.
\end{aligned}$$

Using that $\|A(t_1) - A_\infty\|_{\rm op} \leqslant \varepsilon$ and $z(t_1)^\top A_\infty z(t_1) = -\eta z(t_1)^\top z(t_1)$, this leads to

$$\begin{aligned}
\frac{d}{dt}\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)}\Big|_{t=t_1} &\leqslant -2\frac{z_1^\top A_\infty^2 z_1}{z_1^\top z_1} + 2\frac{\|A_\infty z_1\|_2\varepsilon}{\|z_1\|_2} + 2\eta^2 + 2\eta\varepsilon \\
&\leqslant -2\beta^2 + 2\eta^2 + 2\|A_\infty\|_{\rm op}\varepsilon + 2\eta\varepsilon,
\end{aligned}$$

which is strictly negative for $\varepsilon$ small enough, which is a contradiction because it would imply that for $t$ just above $t_1$, $\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)} < \frac{z(t_1)^\top A_\infty z(t_1)}{z(t_1)^\top z(t_1)} = -\eta$, and thus, $z(t) \in \mathbb{R}_+K$.

**Final contradiction. (♦)** We now have that the particule $z(t)$ is in $\mathbb{R}_+K$ for all $t \geqslant t_0$. We then have for all $t \geqslant t_0$,

$$\frac{d}{dt}z(t)^\top z(t) = -2z(t)^\top A(t)z(t) \geqslant 2\big(-z(t)^\top A_\infty z(t) - \|z(t)\|_2^2\varepsilon\big) \geqslant 2(\eta - \varepsilon)\|z\|_2^2,$$

leading to, after integration, $\|z(t)\|_2^2 \geqslant \|z(t_0)\|_2^2 \exp(2(\eta-\varepsilon)(t-t_0))$, and thus a divergence. This contradicts the convergence of $z(t) = W(t)\alpha_0$.

### 12.3.4 Special case of Oja flow

As an illustration of the convergence results above, we consider the function

$$G(M) = \frac{1}{2}\|M - C\|_F^2,$$

for a symmetric matrix $C \in \mathbb{R}^{d\times d}$, for which the flow can integrated in closed form. We have $G'(M) = M - C$, and thus the following gradient flows:

$$\dot{W} = -G'(WW^\top)W = CW - WW^\top W \quad \text{and} \quad \dot{M} = CM + MC - 2M^2.$$

If we initialize $W(0) = V \in \mathbb{R}^{d\times m}$, we obtain a solution in closed-form (as can be checked by taking derivatives and showing that $\dot{M} = CM + MC - 2M^2$), as

$$M = WW^\top = \exp(Ct)V\big(I + V^\top C^{-1}(\exp(2Ct) - 1)V\big)^{-1}V^\top \exp(Ct).$$

This is the Oja flow, up to a change of variable (Yan et al., 1994). It is interesting to note that if we use $m \leqslant d$ particles, the rank of $WW^\top$ is always less than $m \leqslant d$, and in fact, the same as the rank of the initialization. The global minimizer of $R$ on PSD matrices is the positive part of $C$, whose rank can be strictly smaller than $m$, and thus, it cannot converge to the global optimum of $R$. The minimum number of particles is the number of positive eigenvalues of $C$.

**Vanishing initialization.** If $V = \sqrt{\alpha}I \in \mathbb{R}^{d\times d}$, we get:

$$M = \alpha \exp(2Ct)\big(I + \alpha C^{-1}(\exp(2Ct) - 1)\big)^{-1}.$$

Then, $M$ is a spectral variant of $C$, thus with the same eigenvectors, and eigenvalues $m = \frac{\alpha e^{2ct}}{1+\alpha c^{-1}(e^{2ct}-1)} \approx \frac{c}{1+e^{-2ct}c/\alpha}$ for small $\alpha$, where $c$ is the corresponding eigenvalue of $c$.

Thus, when $\alpha$ tends to zero (and therefore closer to the stationary point), the eigenvalues $m$ stay at zero until they increase to the final positive values $c$, and this increase happens around $t = \frac{1}{2c}\log\frac{1}{\alpha}$. We thus observe incremental learning of each eigenvector, with each eigenvector corresponding to a positive eigenvalue $c$, which is a very different optimization dynamic from the one obtained from projected gradient descent, which corresponds to $m = c(1 - e^{-t})$ where all eigenvectors come in together. This incremental learning at different time scales is common in non-convex optimization, see Saxe et al. (2019); Gidel et al. (2019) for linear networks, and Berthier (2023); Pesme and Flammarion (2023) for diagonal linear networks, where precise statements can be made.

## 12.4 Lazy regime and neural tangent kernels (♦)

For over-parameterized one-hidden layer neural networks, with prediction functions of the form (note the rescaling by $1/m$):

$$f(x) = \frac{1}{m}\sum_{j=1}^m \eta_j\sigma(w_j^\top x + b_j) = \frac{1}{m}\sum_{j=1}^m \Psi(v_j),$$

with $\Psi$ defined in Section 12.3.1, we have seen two types of learning procedures in Chapter 9 when the number of neurons grows unbounded:

- Optimizing over all layers, leading to the mean-field limit presented in Section 12.3.1 and with the non-Hilbertian norm $\gamma_1$. This corresponds to all parameters being initialized with a scaling that does not depend on $m$.
- Optimizing over the last layer only, leading to the kernel regime, with the Hilbertian norm $\gamma_2$ defined in Section 9.5. The input weights $(w_j, b_j)$ are all sampled without scaling. For the output weights, $\eta_j$ can be $O(1)$ or $O(\sqrt{m})$ if initialized with zero-mean (so that the overall norm of the prediction function remains bounded in high probability). Since this is a convex optimization problem, scaling does not matter as much.

**Lazy training.** We now consider a third regime, which we refer to as the "lazy" regime, following Chizat et al. (2019). It corresponds to initializing each $\eta_j$ with a scaling proportional to $\sqrt{m}$. This is made possible by having zero mean initializations so that a mean of $m$ terms is of order $O(1/\sqrt{m})$ and not $O(1)$ (leading to an overall predictor that remains $O(1)$). We will formalize this training regime by seeing this model as a diverging constant $\alpha$ (here $\sqrt{m}$) times a classical model with a mean-field limit.

We thus consider the minimization of the objective function $G(V) = \mathcal{R}(h(V))$ with respect to $V = (v_1, \dots, v_m)$. In the lazy regime, we end up minimizing $\mathcal{R}(\alpha h(V))$ with a scaling factor $\alpha$ that tends to infinity, using a gradient flow on $V$, started at $V(0)$ such that $\alpha h(V(0))$ remains bounded. In our neural network example, $\alpha = \sqrt{m}$, and $h$ is the regular neural network.

We consider the gradient flow to minimize $G(V)$, with a step-size $1/\alpha^2$ (scaling adapted to have a non-trivial dynamic), that is,

$$\frac{d}{dt}V(t) = -\frac{1}{\alpha}Dh(V)^\top \mathcal{R}'(\alpha h(V(t))), \tag{12.20}$$

where $Dh(V)$ is the differential of $h$ at $V$ (that is, a linear function from $\mathbb{R}^{(d+2)\times m}$ to $\mathcal{H}$). For the predictor $\alpha h(V)$, we get:

$$\frac{d}{dt}[\alpha h(V(t))] = -Dh(V(t))Dh(V(t))^\top \mathcal{R}'(\alpha h(V(t))). \tag{12.21}$$

At initialization $t = 0$, $\alpha h(V(t))$ remains bounded by construction, and since the optimization will tend to make the predictor better and better, we can expect it to remain bounded. Thus, we can expect $\mathcal{R}(\alpha h(V(t)))$ and $\mathcal{R}'(\alpha h(V(t)))$ to be $O(1)$. Thus, from Eq. (12.20), we obtain that the parameters $V$ change at rate $O(1/\alpha)$, while from Eq. (12.21) the predictor changes at a rate which is independent of $\alpha$. Thus, in the limit of large $\alpha$, the parameters only move infinitesimally, while the predictor still makes significant progress, hence the name lazy training (see more formal arguments by Chizat et al., 2019).

**Equivalent linear model.** Since parameters move infinitesimally, the model $\alpha h(V)$ behaves like an affine model, $\alpha h(V(0)) + \alpha Dh(\alpha V(0))(V - V(0))$, and thus the corresponding cost function $\mathcal{R}(\alpha h(V))$ behaves like a convex function, hence leading to attractive global convergence results for neural network training (see, e.g. Du et al., 2018).

However, since we have an explicit linear model, the lazy regime also leads to a positive definite kernel, which we now define (with the same properties as traditional kernel methods in Chapter 7).

**Neural tangent kernel (♦).** If we assume that $h(V(0)) = 0$ (e.g., for neural networks, assuming that all initial neurons come by pairs with the same input weights and opposite output weights), then the affine model has a linear part proportional to $Dh(\alpha V(0))V$. We can thus associate to it a kernel, referred to as the neural tangent kernel (Jacot et al., 2018).

To make things concrete, for neural networks with one hidden layer, $h(x, v_1, \dots, v_m) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j)$, the corresponding features, for each $j \in \{1, \dots, m\}$, are

$$\begin{aligned}
\text{derivative with respect to } \eta_j \quad &: \quad \frac{1}{\sqrt{m}} \sigma(w_j(0)^\top x + b_j(0)) \\
\text{derivative with respect to } w_j \quad &: \quad \frac{1}{\sqrt{m}} \eta_j(0) \sigma'(w_j(0)^\top x + b_j(0)) x \\
\text{derivative with respect to } b_j \quad &: \quad \frac{1}{\sqrt{m}} \eta_j(0) \sigma'(w_j(0)^\top x + b_j(0)).
\end{aligned}$$

When the initialization of neuron weights is random, we get the equivalent kernel by the law of large numbers:

$$k(x, x') = \mathbb{E}\big[\sigma(w^\top x + b)\sigma(w^\top x' + b) + \mathbb{E}\big[\sigma'(w^\top x + b)\sigma'(w^\top x' + b)(x^\top x' + 1)\big],$$

whose first part is the traditional random feature kernel, but which has an additional part, which makes for a richer model but cannot correct the limitations of kernel methods (see, e.g., Bietti and Bach, 2021, and references therein).

## 12.5 Conclusion

In this chapter, we have presented a series of results related to over-parameterized models, confirming that some form of regularization is needed: as opposed to previous chapters, where (except for boosting procedures in Section 10.3) an explicit penalty was put on the model parameters, overfitting is avoided by “computational regularization”, that is, through the implicit bias of gradient descent techniques. This was formally shown for linear models, but this extends more generally (see, e.g., Lyu and Li, 2019; Chizat and Bach, 2020)

We also described how over-parameterization, while not detrimental to generalization performance, can be a blessing in terms of optimization, with qualitative results showing global convergence for infinitely overparameterized problems. Obtaining non-asymptotic results (in terms of convergence times of number of neurons) remains an active area of research.

# Chapter 13

# Structured prediction

**Chapter summary**

- With appropriate modifications, we can design convex surrogates for output spaces that are arbitrarily complex and with generic loss functions, starting with multi-category classification.
- Like for binary classification, these convex surrogates lead to efficient algorithms that predict optimally given infinite amounts of data (Fisher consistency).
- Quadratic surrogates that extend the square loss lead to simple, intuitive, and consistent estimation procedures with well-defined decoding steps once a score function has been learned. They can be extended to smooth surrogates.
- Non-smooth surrogates can be defined in the general structured prediction framework, that extends the support vector machine.

In most of this book on supervised learning, we have focused on regression or binary classification, which led to estimating real-valued prediction functions directly when predicting a real-valued output (least-squares regression) or indirectly through convex surrogates (support vector machine or logistic regression) where the binary output in $\{-1, 1\}$ was obtained by taking the sign function. As shown in Section 4.1, the use of convex surrogates comes with strong theoretical guarantees in terms of achieving the Bayes error (that is, the optimal performance on unseen data).

In this chapter, we tackle arbitrary output spaces $\mathcal{Y}$, with arbitrary loss functions, which are ubiquitous in practice (see examples in Section 13.2). Most developments from Section 4.1 will extend with appropriate modifications.

We start in Section 13.1 with the natural extension to multi-category classification with the 0-1 loss, which directly extends binary classification, before describing in Section 13.2 a more general class of problems, referred to as structured prediction. We then present surrogate methods in Section 13.3 and their desirable properties before describing the two main classes, that is, smooth surrogates in Section 13.4 and non-smooth surrogates in Section 13.5. We then present generalization bounds in Section 13.6 and experiments in Section 13.7.

## 13.1 Multi-category classification

We dealt with binary classification with $\mathcal{Y} = \{-1, 1\}$ in Section 4.1.1 by estimating real-valued prediction functions and taking their signs. Going from 2 to $k > 2$ classes requires multi-dimensional vector-space valued functions. To preserve symmetry among classes, we will consider $k$-dimensional outputs. That is, for $\mathcal{Y} = \{1, \dots, k\}$, we will estimate a function $g : \mathcal{X} \to \mathbb{R}^k$, and predict as $f(x) \in \arg\max_{j\in\{1,\dots,k\}} g_j(x) \subset \mathcal{Y}$.

When $k = 2$, we recover our traditional framework by mapping $\{1, 2\}$ to $\{-1, 1\}$ and taking the sign of $g_2(x) - g_1(x)$, highlighting the general fact (valid for all $k$) that predictions are invariant by the addition of a constant vector to $g(x) \in \mathbb{R}^k$.

In the binary case, the convex surrogates we considered were all of the form $\Phi(yg(x))$. In the multi-category case, there is significantly more diversity. In Section 13.1.1 below, we describe the most commonly used convex surrogates and, when possible, the corresponding optimal predictors and their relationship with the Bayes predictor for the 0-1 loss, equal to $\arg\max_{z\in\{1,\dots,k\}} \mathbb{P}(y = z|x)$. Generalization bounds will then be derived, first for stochastic gradient descent used on linear models in Section 13.1.2 because it does not require any new developments, and then using Rademacher complexities in Section 13.1.3. In later sections, we show how this can be generalized to general output spaces.

Throughout this section on multi-category classification, we will identify elements $y$ of $\{1, \dots, k\}$ with the corresponding canonical basis vector in $\mathbb{R}^k$, that is, the vector $\bar{y} \in \mathbb{R}^k$ with all zero components except a one at index $y$.

### 13.1.1 Extension of classical convex surrogates

All binary convex surrogates presented in Section 4.1.1 have natural extensions that we now present. We consider a label $y \in \{1, \dots, k\}$ (also identified to a canonical basis vector $\bar{y}$), and a vector-valued function $g : \mathcal{X} \to \mathbb{R}^k$. Our goal is to build a convex surrogate $S(y, g(x))$ (which is convex with respect to its second variable).

**Softmax loss.** We can extend the logistic loss and its relationship with maximum likelihood by considering the conditional model:

$$\mathbb{P}(y = j|x) = \frac{\exp(g_j(x))}{\sum_{i=1}^k \exp(g_i(x))} = \text{softmax}(g(x))_j,$$

by definition of the softmax function from $\mathbb{R}^k$ to the simplex in $\mathbb{R}^k$, defined as $\text{softmax}(u)_j = \exp(u_j)/\sum_{i=1}^k \exp(u_i)$.

The negative log-likelihood (often referred to as the cross-entropy loss) for this model is then equal to

$$\begin{aligned} S(y, g(x)) &= -\log \frac{\exp(g_y(x))}{\sum_{i=1}^k \exp(g_i(x))} = -g_y(x) + \log\Big(\sum_{j=1}^k \exp(g_j(x))\Big) \\ &= -\bar{y}^\top g(x) + \log\Big(\sum_{j=1}^k \exp(g_j(x))\Big). \end{aligned}$$

The minimizer of the surrogate expected risk $\mathbb{E}[S(y, g(x))]$ is then equal to $g^*(x)_j = \log \mathbb{P}(y = j|x) + c(x)$, for any function $c : \mathcal{X} \to \mathbb{R}$; thus, the predictor $\arg\max_{j\in\{1,\dots,k\}} g^*(x)_j$ is the Bayes predictor, which will lead to consistent estimation. A calibration function relating the excess surrogate risk and the 0-1 excess risk will be derived in Section 13.4 in the more general structured prediction case (with the same square root behavior as for logistic regression).

**Square loss.** The square loss has a natural extension $S(y, g(x)) = \|\bar{y} - g(x)\|_2^2$, with a minimizer of the surrogate expected risk equal to $g^*(x) = \mathbb{E}[\bar{y}|x]$. Again, the predictor $\arg\max_{j\in\{1,\dots,k\}} g^*(x)_j$ is the Bayes predictor (note that this is an instance of the “one vs. all” framework presented below), with a calibration function derived in Section 13.4, also with a square root behavior typical of smooth surrogates.

**Hinge loss.** The maximum-margin framework presented in Section 4.1.2 can be extended in several ways. We present the one that has natural extensions in structured prediction. The goal is to make $g_y(x)$ strictly larger than all others $g_j(x)$, for $j \neq y$, with potential slack, that is, we aim at finding the lowest $\xi \geqslant 0$, such that

$$\forall j \neq y, \;\; g_y(x) \geqslant g_j(x) + 1 - \xi.$$

The lowest such $\xi$ can obtained in closed form, leading to the surrogate:

$$S(y, g(x)) = \sup_{j\in\{1,\dots,k\}} \big\{ 1_{y\neq j} + g(x)_j - g(x)_y \big\}.$$

Finding the minimizer of the expected surrogate risk is not as easy. We have for a given $x \in \mathcal{X}$,

$$\mathbb{E}[S(y, g(x))|x] = \sum_{i=1}^k \mathbb{P}(y = i|x) \sup_{j\in\{1,\dots,k\}} \big\{ 1_{i\neq j} + g(x)_j - g(x)_i \big\}.$$

Assuming without loss of generality that posterior probabilities given $x$ are non-increasing, the global minimizer of the quantity above can be shown (proof left as an exercise) to be achieved for $g_1(x) \geqslant g_2(x) = \cdots = g_k(x)$, and we thus need to minimize

$$\mathbb{P}(y = 1|x)(1 + g_1(x) - g_2(x))_+ + (1 - \mathbb{P}(y = 1|x))(1 + g_2(x) - g_1(x))_+.$$

If $\mathbb{P}(y = 1|x) > 1/2$, then the optimal $g_1(x) - g_2(x)$ can be shown to be equal to 1, and the prediction is optimal. Otherwise, it is not, hence, we lose consistency in general, as for $k > 2$, it is not always the case that the posterior probability of the most likely class exceeds $1/2$. A consistent version of the non-smooth hinge loss will be discussed in Section 13.5.

**One vs. all (♦).** This class of techniques essentially solve $k$ different binary optimization problems, by solving *independently* for each $g_j(x)$ predicting $y_j \in \{0, 1\}$. Using convex surrogates for these problems and taking into account the mapping from $\{0, 1\}$ to $\{-1, 1\}$, the overall cost function is:

$$S(y, g(x)) = \sum_{j=1}^{k} \Phi((2y_j - 1)g_j(x)),$$

with $\Phi$ one of the convex surrogates from Section 4.1.1. For the square loss, we recover the multivariate square loss, but other losses can be used as well. The expected surrogate risk given $x \in \mathcal{X}$, is then equal to

$$\mathbb{E}[S(y, g(x))|x] = \sum_{j=1}^{k} \Big\{ \mathbb{P}(y = j|x)\Phi(g_j(x)) + (1 - \mathbb{P}(y = j|x))\Phi(-g_j(x)) \Big\}.$$

For a differentiable strictly convex function such that $\Phi(z) < \Phi(-z)$ for $z > 0$ (which excludes the hinge loss), minimizing it is done by setting $\mathbb{P}(y = j|x)\Phi'(g_j(x)) = (1 - \mathbb{P}(y = j|x))\Phi'(-g_j(x))$. One can then show that $g_j(x)$ is a strictly increasing function of $\mathbb{P}(y = j|x)$, and thus we get consistent predictions (in the population case), with also calibration functions (see Zhang, 2004b, Theorem 11).

**Beyond.** As reviewed by Zhang (2004b), there are many examples of convex surrogates to estimate the $k$ functions $g_1, \dots, g_k : \mathcal{X} \to \mathbb{R}$, based on several principles. Reductions to binary classification problems go beyond one vs. all approaches, for example, by considering several subsets $A$ of $\{1, \dots, k\}$ and solving the binary classification problems of deciding $y \in A$ vs. $y \notin A$. This approach based on error-correcting codes (Dietterich and Bakiri, 1994) will also be considered within the general surrogate framework.

**Exercise 13.1** *Consider the following surrogate $S(y, g(x)) = \sum_{i \neq y} \Phi(-g_i(x))$, with the additional constraint that $\sum_{i=1}^{k} g_i(x) = 0$, with a strictly convex decreasing function $\Phi$. Show that if $g^*$ is the minimizer of $\mathbb{E}[S(y, g(x)]$, then for all $i, j \in \{1, \dots, k\}$, $\mathbb{P}(y = i|x) > \mathbb{P}(y = j|x) \Rightarrow g_i^*(x) > g_j^*(x)$. Conclude on the consistency of this convex surrogate.*

### 13.1.2 Generalization bound I: stochastic gradient descent

In these following two sections, we will consider generalization bounds for Lipschitz-continuous losses (all losses in the section above are Lipschitz-continuous, potentially once restricted to a bounded set), like done in Chapter 4 with estimation errors controlled by Rademacher complexities, and Chapter 5 using single-pass stochastic gradient descent (SGD). We start with SGD because the analysis can be done without the need for new tools.

**Linear models.** Since we will leverage convergence results for convex optimization algorithms, we need to consider linear models. We thus assume that we have a feature vector $\varphi : \mathcal{X} \to \mathbb{R}^d$ almost surely bounded by $R$ in $\ell_2$-norm, and that the vector-valued function $g : \mathcal{X} \to \mathbb{R}^k$ is parameterized linearly as $g^{(\theta)}(x) = \theta^\top \varphi(x)$, with $\theta = (\theta_1, \dots, \theta_k) \in \mathbb{R}^{d \times k}$. We aim to estimate $\theta$ restricted to a ball of radius $D$ for a certain norm $\Omega$.

Several candidates are natural for the norm $\Omega$: the simplest one is the Frobenius norm defined through its square $\|\theta\|_{\mathrm{F}}^2 = \sum_{i=1}^k \|\theta_i\|_2^2 = \sum_{i=1}^k \sum_{j=1}^d \theta_{ji}^2$, which corresponds to the Euclidean norm for the matrix $\theta$ seen as a vector, and for which all results related to SGD will apply. Another classical norm is the nuclear norm (a.k.a. trace norm) defined as the sum of singular values of $\theta$, which will push for low-rank $\theta$ with similar properties as the $\ell_1$-norm in Section 8.3. Other norms, such as the $\ell_1$-norm or "grouped" norms, could also be considered for variable selection. For these norms, optimization tools such as stochastic mirror descent from Section 11.1.3 are needed (see Exercise 13.2 for the nuclear norm).

Note finally that we could use (positive definite) kernel methods with the kernel trick from Section 7.4.5 to deal with infinite-dimensional feature vectors.

**Sampling assumptions.** Following the rest of the book, we assume that we have $n$ i.i.d. pairs of observations $(x_i, y_i) \in \mathcal{X} \times \{1, \dots, k\}$, $i = 1, \dots, n$. Given the function $g^{(\theta)} : \mathcal{X} \to \mathbb{R}^k$ defined through the linear model above, we consider the expected risk $\mathcal{R}(f)$ for the predictor $f : \mathcal{X} \to \{1, \dots, k\}$ defined as $f(x) \in \arg\max_{i \in \{1,\dots,k\}} g^{(\theta)}(x)_i$ and the 0-1 loss. As already mentioned and shown in the subsequent section, the excess risk $\mathcal{R}(f) - \mathcal{R}^*$ (where $\mathcal{R}^*$ is the minimum risk overall measurable functions, not only the ones obtained from the model), will be bounded by an increasing function of the excess surrogate risk $\mathcal{R}_S(g) - \mathcal{R}_S^*$ (again $\mathcal{R}_S^*$ is the minimal value across all measurable functions). We thus focus on the excess surrogate risk.

Using the same decomposition as in Section 4.5.4, we consider an estimator $\hat{\theta} \in \mathbb{R}^{d \times k}$ that depends on the observations and subject to the constraint $\|\theta\|_{\mathrm{F}} \leqslant D$ (other norms could also be considered) for some real value $D$. A bound on the expected excess risk is obtained by the sum of the approximation error $\inf_{\|\theta\|_{\mathrm{F}} \leqslant D} \mathcal{R}_S(g^{(\theta)}) - \mathcal{R}_S^*$ and the estimation error $\mathcal{R}_S(g^{(\hat{\theta})}) - \inf_{\|\theta\|_{\mathrm{F}} \leqslant D} \mathcal{R}_S(g^{(\theta)})$. We focus on the latter quantity, a random quantity we bound through its expectation.

We assume that the convex surrogate $S : \mathcal{Y} \times \mathbb{R}^k \to \mathbb{R}$ is Lipschitz-continuous with respect to its second variable, that is, its subgradients $S'(y, u) \in \mathbb{R}^k$ are bounded by $G$ in $\ell_2$-norm.

**Single-pass SGD.** We consider the following SGD iteration, for $t \in \{1, \dots, n\}$, with an arbitrary subgradient of the surrogate $S$:

$$\theta_t = \Pi_D\big(\theta_{t-1} - \gamma_t \varphi(x_t) S'(y_t, \theta_{t-1}^\top \varphi(x_t))^\top\big).$$

The analysis of Section 5.4 exactly applies, and with the choice $\gamma_t = \frac{D}{RG\sqrt{n}}$, we obtain the following generalization bound:

$$\mathbb{E}\big[\mathcal{R}_S(g_{(\theta_n)}) - \inf_{\|\theta\|_{\mathrm{F}} \leqslant D} \mathcal{R}_S(g^{(\theta)})\big] \leqslant \frac{DRG}{\sqrt{n}}, \tag{13.1}$$

which is exactly the same as for real-valued predictions.

**Exercise 13.2 (Mirror descent for trace-norm penalty (♦))** *We consider the following mirror map on $\mathbb{R}^{d\times k}$, $\Psi(\theta) = \frac{1}{2}\|\sigma(\theta)\|_p^2$, where $\sigma(\theta)$ is the vector of singular values of $\theta$. Show that for $p \in (1,2)$ is $(p-1)$-strongly-convex with respect to the norm $\|\sigma(\cdot)\|_p$. Show how to apply stochastic mirror descent on a nuclear norm ball and provide a convergence rate.*

### 13.1.3 Generalization bound II: Rademacher complexities (♦)

Another approach we followed in this book is to assume we can compute a minimizer of the empirical risk beyond linear models (in particular for neural networks). We thus assume a generic space of functions $\mathcal{G}$ of functions from $\mathcal{X} \to \mathbb{R}^k$, and define $\hat{g} \in \mathcal{G}$ as the minimizer of the empirical surrogate risk $\widehat{\mathcal{R}}(g) = \frac{1}{n}\sum_{i=1}^n S(y_i, g(x_i))$ over $g \in \mathcal{G}$. Following Section 4.2 and Section 4.5, the expected estimation error $\mathcal{R}_S(\hat{g}) - \inf_{g\in\mathcal{G}} \mathcal{R}_S(g)$ is then less than twice the Rademacher complexity

$$\mathrm{R}_n(S, \mathcal{G}) = \mathbb{E}\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i S(y_i, g(x_i))\Big],$$

where the expectation is taken with respect to the data and the Rademacher random variables $\varepsilon_1, \dots, \varepsilon_n \in \{-1, 1\}$. In the real-valued case, we used a contraction principle (Prop. 4.3) that allows us to get rid of the surrogate cost $S$ as long as it is Lipschitz-continuous. Such a contraction principle also exists for vector-valued prediction functions (Maurer, 2016) and is presented in Prop. 13.1 below. Its application leads to

$$\mathrm{R}_n(S, \mathcal{G}) \leqslant \sqrt{2}G \cdot \mathbb{E}\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n (\varepsilon_i')^\top g(x_i)\Big], \tag{13.2}$$

where now each $\varepsilon_i' \in \{-1,1\}^k$, $i = 1, \dots, n$, is a *vector* of independent Rademacher random variables. This bound allows us to obtain a bound for all the function spaces that we considered in this book. We consider linear models below (and compare them to the SGD bound from above) and leave neural networks as an exercise.

**Linear models.** In the set-up of the previous section (linear models with Frobenius bound), we can further upper-bound in Eq. (13.2) as:

$$\begin{aligned}
\mathrm{R}_n(S,\mathcal{G}) &\leqslant \sqrt{2}G\cdot\mathbb{E}\Big[\sup_{g\in\mathcal{G}}\frac{1}{n}\sum_{i=1}^n(\varepsilon_i')^\top g(x_i)\Big] = \sqrt{2}G\cdot\mathbb{E}\Big[\sup_{\|\theta\|_{\mathrm{F}}\leqslant D}\frac{1}{n}\sum_{i=1}^n(\varepsilon_i')^\top\theta^\top\varphi(x_i)\Big]\\
&= \frac{D}{n}\sqrt{2}G\cdot\mathbb{E}\Big[\Big\|\sum_{i=1}^n\varphi(x_i)(\varepsilon_i')^\top\Big\|_{\mathrm{F}}\Big] \leqslant \frac{DG}{n}\sqrt{2}\cdot\bigg(\mathbb{E}\Big[\Big\|\sum_{i=1}^n\varphi(x_i)(\varepsilon_i')^\top\Big\|_{\mathrm{F}}^2\Big]\bigg)^{1/2}\\
&= \frac{DG}{n}\sqrt{2}\cdot\bigg(\mathbb{E}\Big[\sum_{i=1}^n\|\varphi(x_i)\|_2^2\|\varepsilon_i'\|_2^2\|_{\mathrm{F}}^2\Big]\bigg)^{1/2} \leqslant \frac{DGR\sqrt{2k}}{\sqrt{n}}.
\end{aligned}$$

We thus obtain the same bound as in Eq. (13.1), but with an extra factor of $\sqrt{k}$, showing that the Rademacher average technique, as opposed to the real-valued case, does not lead to the same result as SGD. However, it applies more generally to non-convex loss functions (as long as they are Lipschitz-continuous) and non-linear predictors.

**Contraction principle (♦).** We now provide a proof for the vector-valued contraction principle, taken from Maurer (2016). The proof follows the same structure as the proof of Prop. 4.3 for $k=1$, and we start from a key lemma.

**Lemma 13.1** *Given any functions $b:\Theta\to\mathbb{R}$, $a:\Theta\to\mathbb{R}^k$ (no assumption) and $c:\mathbb{R}^k\to\mathbb{R}$ any 1-Lipschitz-functions (with respect to the $\ell_2$-norm), we have, for $\varepsilon\in\{1,1\}$ a Rademacher random variable, and $\varepsilon'\in\mathbb{R}^n$ a vector of independent Rademacher random variables:*

$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\varepsilon c(a(\theta))\Big\}\Big] \leqslant \mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sqrt{2}\sum_{j=1}^k\varepsilon_j' a_j(\theta)\Big\}\Big].$$

**Proof** (♦♦) Writing down explicitly the expectation with respect to $\varepsilon$, we get:

$$\begin{aligned}
\mathbb{E}_\varepsilon\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\varepsilon c(a(\theta))\Big\}\Big] &= \frac{1}{2}\sup_{\theta\in\Theta}\Big\{b(\theta)+c(a(\theta))\Big\}+\frac{1}{2}\sup_{\theta\in\Theta}\Big\{b(\theta)-c(a(\theta))\Big\}\\
&= \sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{c(a(\theta))-c(a(\theta'))}{2}.
\end{aligned}$$

By taking the supremum over $(\theta,\theta')$ and $(\theta',\theta)$, and using Lipschitz-continuity of $c$, we further get the bound

$$\sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{|c(a(\theta))-c(a(\theta'))|}{2} \leqslant \sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{\|a(\theta)-a(\theta')\|_2}{2}.$$

In the proof of Prop. 4.3 (for $k=1$), we then applied the same set of equalities to obtain the desired result without the constant $\sqrt{2}$. Here, we can use Khintchine's inequality from Lemma 11.1, that is, for any vector $v\in\mathbb{R}^k$, $\|v\|_2\leqslant\sqrt{2}\cdot\mathbb{E}\big[\big|\sum_{j=1}^k\varepsilon_j'v_j\big|\big]$ for any vector $v \in \mathbb{R}^k$ and independent Rademacher random variables $\varepsilon'_1, \dots, \varepsilon'_n$. This leads to the bound:

$$\begin{aligned}
&\sup_{\theta,\theta'\in\Theta} \frac{b(\theta)+b(\theta')}{2} + \frac{\sqrt{2}}{2}\mathbb{E}\Big[\Big|\sum_{j=1}^k \varepsilon'_j(a_j(\theta)-a_j(\theta'))\Big|\Big] \\
&\leqslant \mathbb{E}\Big[\sup_{\theta,\theta'\in\Theta} \frac{b(\theta)+b(\theta')}{2} + \frac{\sqrt{2}}{2}\Big|\sum_{j=1}^k \varepsilon'_j(a_j(\theta)-a_j(\theta'))\Big|\Big] \\
&\qquad\qquad \text{using properties of expectations and suprema,} \\
&= \mathbb{E}\Big[\sup_{\theta,\theta'\in\Theta} \frac{b(\theta)+b(\theta')}{2} + \frac{\sqrt{2}}{2}\sum_{j=1}^k \varepsilon'_j(a_j(\theta)-a_j(\theta'))\Big] \text{ by symmetry,} \\
&\leqslant \mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sqrt{2}\sum_{j=1}^k \varepsilon'_j a_j(\theta)\Big\}\Big] \text{ which is the desired result.}
\end{aligned}$$

■

**Proposition 13.1 (Vector-valued contraction principle)** *Given any functions $b : \Theta \to \mathbb{R}$, $a_i : \Theta \to \mathbb{R}^k$ (no assumption) and $c_i : \mathbb{R}^k \to \mathbb{R}$ any 1-Lipschitz-functions (with respect to the $\ell_2$-norm), for $i = 1, \dots, n$, we have, for $\varepsilon \in \mathbb{R}^n$ a vector of independent Rademacher random variables, and $\varepsilon' \in \mathbb{R}^{n\times k}$ a matrix of independent Rademacher random variables:*

$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sum_{i=1}^n \varepsilon_i c_i(a_i(\theta))\Big\}\Big] \leqslant \mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sqrt{2}\sum_{i=1}^n\sum_{j=1}^k \varepsilon'_{ij}a_{ij}(\theta)\Big\}\Big].$$

**Proof** (♦♦) We consider a proof by induction on $n$. The case $n = 0$ is trivial, and we show how to go from $n \geqslant 0$ to $n+1$. We thus consider $\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sum_{i=1}^{n+1}\varepsilon_i c_i(a_i(\theta))\Big\}\Big]$ and apply Lemma 13.1 with fixed $\varepsilon_1, \dots, \varepsilon_n$, leading to the bound

$$\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n,\varepsilon'_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sum_{j=1}^k \varepsilon'_{n+1,j}a_{n+1,j}(\theta)+\sum_{i=1}^n \varepsilon_i c_i(a_i(\theta))\Big\}\Big],$$

and apply the induction hypothesis with $\varepsilon'_{n+1}$ fixed to obtain the desired result. ■

**Exercise 13.3 (Multi-cateagory classification with neural networks (♦))** *We consider the neural network with outputs in $\mathbb{R}^k$, that is, using the notations from Section 9.2, $f(x) = \sum_{j=1}^m \sigma(w_j^\top x + b_j)\eta_j$, with now $\eta_j \in \mathbb{R}^k$. Extend the estimation error analysis from Section 9.2.3 by imposing a constraint on $\sum_{j=1}^m \|\eta_j\|_2$.*

## 13.2 General set-up and examples

Now that the multi-category classification has been presented, we consider the same general set-up presented earlier in Section 2.2, that is, we want to predict a variable

$y \in \mathcal{Y}$ from some $x \in \mathcal{X}$, and given a prediction $z \in \mathcal{Y}$, we incur the loss $\ell(y, z)$, with the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Like in Section 2.2, given a test distribution $p$ on $\mathcal{X} \times \mathcal{Y}$, we can define the Bayes predictor

$$f^*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \int_{\mathcal{Y}} \ell(y, z) dp(y|x) \tag{13.3}$$

in the usual way. While it led to simple closed-form formulas for the 0–1 loss and binary classification, this will not always be the case. Nevertheless, our goal will still be to achieve its (optimal) performance at a reasonable computational cost.

### 13.2.1 Examples

We now consider classic examples with their applicative motivations in natural language processing, biology, or computer vision—see more examples by Nowak et al. (2019) and Ciliberto et al. (2020):

- **Multi-category classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i, j) = L_{ij}$. The usual 0-1 loss from Section 13.1 corresponds to $L_{ij} = 1_{i \neq j}$, but in most applications, errors do not have the same cost (for example, in spam prediction, classifying a legitimate email as spam costs much more than the opposite).
- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with $\ell(y, z) = \rho(y - z)$ and typically $\rho$ non convex. When $\rho$ is convex, such as $\rho(\delta) = |\delta|$ or $\rho(\delta) = \delta^2$, there is no need for a surrogate framework, but then regression may be non-robust to strong outlier perturbations. Having a non-convex $\rho$, such as, $\rho(\delta) = 1 - \exp(-\delta^2)$ leads to robust regression.
- **Ordinal regression**: this is a particular case of the situation above, where the loss matrix has a specific structure where the loss $L_{ij}$ is increasing in $|i - j|$. This is common when using a rating system with a few discrete levels. One possibility is to ignore the discrete structure of the loss and use least-squares regression together with rounding, but this does not lead to optimal predictions.
- **Multiple labels**: $\mathcal{Y} = \{-1, 1\}^k$, with cardinality $2^k$, with the traditional Hamming loss $\ell(y, z) = \frac{1}{2}\|y - z\|_1 = \frac{1}{4}\|y - x\|_2^2$, which counts the number of mistakes and will be a running example in this chapter. Other scores, such as precision/recall[1] or $F$-scores[2], are typically used (and may not be symmetric) and can be treated as well with the frameworks presented in this chapter. These are detailed by Nowak et al. (2019).

  Multiple label prediction is common in multimedia applications, where there are potentially $k$ objects in a document, and one wants to predict which ones are present.
- **Permutations**: $\mathcal{Y}$ is the set of permutations among $m$ elements, that is, $y$ a bijection from $\{1, \dots, m\}$ to $\{1, \dots, m\}$. We have then $|\mathcal{Y}| = m!$. A common loss function is the "pairwise disagreement", equal to $\ell(y, z) = \sum_{i,j=1}^m 1_{y(i) > y(j)} 1_{z(i) < z(j)}$,

[1] See https://en.wikipedia.org/wiki/Precision_and_recall.

[2] See https://en.wikipedia.org/wiki/F-score.

but other losses such as the discounted cumulative gain[3] can be used, or losses of the form $\sum_{i=1}^m \ell_i(a_{y(i)} - b_{(y(i)})$ for vectors $a, b \in \mathbb{R}^m$ and functions $\ell_i : \mathbb{R} \to \mathbb{R}$. Predicting permutations occurs in information retrieval and ranking problems where the permutation encodes a user's preferences over a set of $m$ items. See Nowak et al. (2019) and references therein for a review of classical losses used in practice.

- **Sequences**: $\mathcal{Y}$ is the set of sequences of potentially arbitrary lengths over some alphabet; this has applications in natural language processing (e.g., translation from one language to another), computational biology (DNA basis or amino-acid sequences), or econometrics/finance (prediction of time series, where the alphabet is usually not finite). The cardinality of $y$ is thus large (or infinite), and the Hamming loss is commonly used.
- **Trees, graphs**: $\mathcal{Y}$ is set of potentially labelled graphs over some vertices. Classic examples include the prediction of molecules (which can be represented as graphs) or the grammatical analysis of sentences in natural language processing.

**Why is it difficult?** Structured prediction is challenging for two reasons:

- Computationally: we need to predict large structured (often discrete) objects from real-valued outputs.
- Statistically: there is a potential curse of dimensionality in both $k$ (the underlying dimension of the problem, to be defined later precisely) *and* the input $d$, in addition to a complicated combinatorial structure.

Our goal is to obtain polynomial-time algorithms in $k$, $n$, and $d$ to attain the optimal prediction, that is, we aim to obtain:

1. Computational tractability by introducing convex surrogates (to use convex optimization) and efficient decoding steps (often dedicated algorithms).
2. Fisher consistency (excess risk goes to zero in the population case) and calibration (sub-optimality for the convex surrogate leads to sub-optimality for the true risk).

Following the rest of the book, we will always go through vector-space valued prediction functions. Thus, there will always be two components:

1. Learning some scores from data, implicitly and explicitly, in a Hilbert space $\mathcal{H}$ or $\mathbb{R}^k$, where $k$ is the (potentially implicit) "affine dimension" of $\mathcal{Y}$.
2. Decoding step to go from score functions to predictions (obvious and somewhat overlooked in the binary classification case, as this was simply the sign).

**From one learning framework per situation to a general framework.** The development of structured prediction methods has seen two streams of work: first, methods dedicated to specific instances (in particular, cost-sensitive multi-category classification, ranking, or learning with multiple labels), then generic frameworks that encompass all the particular cases. In this book, we focus on the latter set of techniques.

[3] See https://en.wikipedia.org/wiki/Discounted_cumulative_gain.

### 13.2.2 Structure encoding loss functions

To achieve guaranteed predictive performance, we will need to impose some low-dimension vectorial structure, which in turn imposes some specific structure within $\mathcal{Y}$, hence the name "structured prediction". More precisely, we will assume that we have two embeddings of the label space $\mathcal{Y}$ into the same Hilbert space $\mathcal{H}$, that is, two maps $\chi, \psi : \mathcal{Y} \to \mathcal{H}$ and a constant $c \in \mathbb{R}$, such that

$$\forall (y,z) \in \mathcal{Y} \times \mathcal{Y},\ \ \ell(y,z) = c + \langle \chi(z), \psi(y) \rangle. \tag{13.4}$$

This assumption is called "structure encoding loss function" (SELF) (Ciliberto et al., 2020). This can be an implicit or explicit embedding (see examples below). Note that the representation is not unique as given a pair $(\chi, \psi)$, any pair $(V\chi, V^{-*}\psi)$ is valid for any invertible operator.

⚠ There are *two* different embeddings of outputs in $\mathcal{Y}$, while typically one for the inputs in $\mathcal{X}$.

**Bayes predictor.** With the assumption in Eq. (13.4) above, we can now express the optimal predictor in Eq. (13.3) as:

$$f^*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \Big\langle \chi(z), \int_{\mathcal{Y}} \psi(y) dp(y|x) \Big\rangle. \tag{13.5}$$

Thus, to obtain Fisher consistency, it is sufficient to estimate well the conditional expectation $\int_{\mathcal{Y}} \psi(y)dp(y|x) \in \mathcal{H}$; this is what smooth surrogates will do in Section 13.4. However, what is only needed is, in fact, sufficient knowledge of this conditional expectation to perform the computation of $f^*(x)$ above. This will lead to non-smooth surrogates in Section 13.5.

**Examples.** We can now revisit the list of losses described in Section 13.2.1 to check if a SELF decomposition exists. In our analysis, we will need a bound on $R_\ell = \sup_{z \in \mathcal{Y}} \|\chi(z)\|$, which we also provide here.

- **Binary classification**, with $\mathcal{Y} \in \{-1, 1\}$ and the 0-1 loss: $\mathcal{H} = \mathbb{R}$, $\chi(y) = -y/2$ and $\psi(z) = z$, since $\ell(y,z) = 1_{y \neq z} = \frac{1}{2} - \frac{yz}{2}$, with $R_\ell = 1/2$.
- **Multi-category classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i,j) = L_{ij}$. This corresponds to the usual "one-hot" encoding of discrete distributions, where $\psi(i) \in \mathbb{R}^k$ is the $i$-th element of the canonical basis. We then have $\ell(i,j) = L_{ij} = \psi(i)^\top L \psi(j)$, that is, $\chi(j) = L\psi(j)$. For this case, we have $R_\ell = \sup_{j \in \{1,\dots,k\}} \|L(:,j)\|_2$. In particular, for the 0-1 loss, we have $L_{ij} = 1_{i \neq j}$, and we can write $\ell(i,j) = 1 - \psi(i)^\top \psi(j)$, and we can take $\chi(i) = -\psi(i)$, with $R_\ell = 1$.

  We can also choose to have a feature map $\psi$ with values in $\{-1, 1\}$ instead of $\{0, 1\}$, in particular for the general reduction to binary classification problems.
- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with the loss $\ell(y,z) = 1 - \exp\big[-(y-z)^2\big]$, which can be written as, using the Fourier transform, $\ell(y,z) = 1 - \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-\omega^2/4) \cos\omega(x-z)d\omega$, which leads to the existence of an infinite-dimensional $\mathcal{H}$.

Indeed, we can select $\mathcal{H}$ to be the set of square integrable functions from $\mathbb{R}$ to $\mathbb{R}^2$, with $\psi(y)(\omega) = e^{-\omega^2/8}\binom{\cos \omega y}{\sin \omega y}$, and $\chi(z)(\omega) = -\frac{1}{2\sqrt{\pi}} e^{-\omega^2/8}\binom{\cos \omega z}{\sin \omega z}$, leading to $R_\ell^2 = \frac{1}{4\pi}\int_{-\infty}^{\infty} \exp(-\omega^2/4) = \frac{1}{2\sqrt{\pi}}$.

- **Multiple labels**: for $\mathcal{Y} = \{-1,1\}^k$, the traditional Hamming loss can be rewritten as $\ell(y,z) = \frac{k}{2} - \frac{1}{2}y^\top z$. We then have, $\psi(y) = y$ and $\chi(z) = -z/2$, and $R_\ell = \sqrt{m}$.

  Note here that choosing the 0-1 loss is not advocated as it is non-zero as soon as a single mistake is made, and it requires a lot of observations in practice (as can be seen by having a constant $R_\ell$ that grows exponentially in $k$.

- **Permutations**: for the pairwise disagreement, we have directly $\mathcal{H} = \mathbb{R}^k$ with $k = m(m-1)$, with $\psi(y)_{ij} = 1_{y(i)>y(j)}$ and $\chi(z)_{ij} = 1_{z(i)<z(j)}$ for $i \neq j$, and $R_\ell \leqslant m$. For the loss $\ell(y,z) = \sum_{i=1}^m (a_{y(i)} - b_{(y(i))})^2$, we have $\psi(y) = y$ and $\chi(z) = -2 * z$, with $R_\ell \leqslant \sqrt{2(m+1)}$.

- **Sequences**: we consider binary sequences for simplicity, that is, $\mathcal{Y} = \{-1,1\}^m$, but it extends more generally to all factor graphs (Wainwright and Jordan, 2008) and types of labels. Using the Hamming loss like for multiple labels ignores the sequential structure and does not enforce any notion of consistency between two successive elements of the sequence. On top of the feature $y_1, \dots, y_m \in \{-1,1\}$, we can add the features $y_1y_2, y_2y_3, \dots, y_{m-1}y_m \in \{-1,1\}$, that allows to consider losses that encourages perfectly predicted sequence of size 2, for example, by considering the loss $\ell(y,z) = \sum_{j=1}^{m-1} 1_{y_j \neq z_j \text{ or } y_{j+1} \neq z_{j+1}} = \sum_{j=1}^{m-1} \big\{ y_j z_j + y_{j+1} z_{j+1} - y_j z_j y_{j+1} z_{j+1} \big\}$.

⚠ Like for binary classification or regression, the loss choice is independent of the function space considered (local averaging, kernels, neural nets).

**Reduction to binary problems.** We have encountered several examples above, where the feature map $\Psi$ has binary values in $\{-1,1\}^m$. We will see below that natural convex surrogates end up simply considering each of the $m$ labels independently (ignoring their potential dependency, i.e., in the ranking case, where components of $\Psi(y)$ are $1_{y(i)<y(j)}$, not all values are possible). This can be useful in structured cases like sequence models or ranking, but also in multi-category classification with the 0-1 loss.

## 13.3 Surrogate methods

In this section, our main concern will be to obtain consistent convex surrogates, convex so that we can run efficient algorithms from Chapter 5, consistent so that we are sure that given sufficient amounts of data and sufficiently flexible models, predictions are optimal. In particular, this will allow us to derive generalization bounds in Section 13.6.

### 13.3.1 Score functions and decoding step

**Binary classification.** In this book, we have performed binary classification by learning a real-valued function $g : \mathcal{X} \to \mathbb{R}$, and then predicting with $f(x) = \text{sign}(g(x)) \in \{-1, 1\}$. In the language of this chapter, we have learned a real-valued score function and applied a specific decoding step from $\mathbb{R}$ to $\{-1, 1\}$ (the sign function). We now present the general surrogate framework.

**General surrogate framework.** In this chapter, we will consider functions $f : \mathcal{X} \to \mathcal{Y}$ that can be written as:

$$f(x) = \text{dec} \circ g(x),$$

where

- $g : \mathcal{X} \to \mathcal{H}$ is a function with values in the vector space $\mathcal{H}$, referred to as a score function.[4]
- $\text{dec} : \mathcal{H} \to \mathcal{Y}$ is the decoding function.

We then need a surrogate loss $S : \mathcal{Y} \times \mathcal{H} \to \mathbb{R}$, that will be used to form empirical and expected surrogate risks:

$$\hat{\mathcal{R}}_S(g) = \frac{1}{n} \sum_{i=1}^n S(y_i, g(x_i)) \quad \text{and} \quad \mathcal{R}_S(g) = \mathbb{E}\big[S(y, g(x))\big].$$

For binary classification where $\mathcal{Y} = \{-1, 1\}$, we had $S(y, g(x)) = \Phi(yg(x))$ for $\Phi$ a convex function.

### 13.3.2 Fisher consistency and calibration functions

Following the same definition as in Section 4.1, we denote $\mathcal{R}_S^*$ the minimim $S$-risk, that is the infimum over all functions from $\mathcal{X}$ to $\mathcal{H}$ of $\mathcal{R}_S(g) = \mathbb{E}\big[S(y, g(x))\big]$. It is equal to:

$$\mathcal{R}_S^* = \mathbb{E}\Big[\inf_{h \in \mathcal{H}} \mathbb{E}\big[S(y, h)|x\big]\Big].$$

The loss is said “Fisher-consistent” if we can get an arbitrary small excess risk $\mathcal{R}(f) - \mathcal{R}^*$ for $f = \text{dec} \circ g$, as soon as the excess $S$-risk of $g$ is sufficiently small. In other words, perfectly minimizing the $S$-risk should lead to the Bayes predictor.

A stronger property that enables the transfer of convergence rates for the excess $S$-risk to the excess risk is the existence of a *calibration function*, that is, an increasing function $H : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\mathcal{R}(\text{dec} \circ g) - \mathcal{R}^* \leqslant H\big[\mathcal{R}_S(g) - \mathcal{R}_S^*\big]$. Note that, like in the binary classification case in Section 4.1.3, a more refined notion of consistency can be defined and studied, see, e.g., Long and Servedio (2013).

[4] In statistics, the score function often refers to the gradient of the log-density with respect to parameters. There is no link between these two definitions.

### 13.3.3 Main surrogate frameworks

As described in Section 4.1, for binary classification, we saw two classes of convex surrogates:

- Smooth surrogates, where the predictor minimizing the expected surrogate risk led to a complete description of the conditional distribution of $y$ given $x$, that is, since we had only two outcomes, knowledge of $\mathbb{E}[y|x]$. Classic examples were the square loss and the logistic loss. Then, when going from the excess surrogate risk to the true excess risk, the calibration function was the square root.
- Non-smooth surrogates, where the predictor minimizing the expected surrogate risk already provided a thresholded version, that is, $\text{sign}(\mathbb{E}[y|x])$. The calibration function, however, did not exhibit a square root behavior but rather a (better) linear behavior.

In this chapter, we will present extensions of these two sets of surrogates: (1) least-squares (or more generally smooth surrogates), (2) max-margin (non-smooth that estimates the discrete estimator directly), as they come with efficient algorithms and guarantees. But there are other related frameworks that we will not study (Osokin et al., 2017; Lee et al., 2004; Blondel et al., 2020). In particular, probabilistic graphical models in the form of conditional random fields are popular (Sutton and McCallum, 2012).

## 13.4 Smooth/quadratic surrogates

We first look at a class of techniques that extends the square and logistic losses beyond binary classification for the whole class of structure encoding loss functions. We first start with quadratic surrogates, following Ciliberto et al. (2020), where the analysis is the simplest and most elegant.

### 13.4.1 Quadratic surrogate

Given the SELF decomposition in Eq. (13.4), we consider estimating a score function $g : \mathcal{X} \to \mathcal{H}$ with the following surrogate function:

$$S(y, g(x)) = \|\psi(y) - g(x)\|^2,$$

for the Hilbert norm $\|\cdot\|$. In other words, we aim at directly estimating $\mathbb{E}\big[\psi(y)|x\big]$ for every $x \in \mathcal{X}$. The decoding function is then naturally

$$\text{dec}(s) \in \arg\min_{z \in \mathcal{Y}} \langle \chi(z), g(x) \rangle,$$

since, when $g(x) = \mathbb{E}\big[\psi(y)|x\big]$, it leads to $\arg\min_{z \in \mathcal{Y}} \mathbb{E}\big[\langle \chi(z), \psi(y) \rangle |x\big] = \arg\min_{z \in \mathcal{Y}} \mathbb{E}\big[\ell(y, z)|x\big]$, which is the optimal predictor.

For the binary classification case, it leads to the square loss framework from Section 4.1.1, but in the general case, it extends to the many situations alluded to earlier. The decoding steps will be described in Section 13.4.3.

When the loss function is induced by a positive-definite kernel, then this framework is also referred to as output kernel regression (see, e.g., Brouard et al., 2016), or kernel dependency estimation (Weston et al., 2002).

### 13.4.2 Theoretical guarantees

For the framework proposed above, we can prove a precise calibration result by leveraging the properties of the square loss. We first notice that

$$\mathcal{R}_S(g) - \mathcal{R}_S^* = \mathbb{E}\Big[\big\|g(x) - \mathbb{E}[\psi(y)|x]\big\|^2\Big]. \tag{13.6}$$

Moreover, by construction, the function defined by $g^*(x) = \mathbb{E}[\psi(y)|x]$ is the minimizer of the expected $S$-risk, and the Bayes predictor is indeed $f^* = \mathrm{dec} \circ g^*$.

We can then express the excess risk using the decomposition of the loss as:

$$\begin{aligned}
& \mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}^* \\
= \ & \mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}(\mathrm{dec} \circ g^*) \\
= \ & \mathbb{E}\Big[\mathbb{E}\big[\ell(y, \mathrm{dec} \circ g(x)) - \ell(y, \mathrm{dec} \circ g^*(x))\big|x\big]\Big] \\
= \ & \mathbb{E}\Big[\mathbb{E}\big[\big\langle\psi(y), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big|x\big]\Big] \text{ by the SELF decomposition,} \\
= \ & \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big], \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big] \text{ by moving expectations,} \\
= \ & \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big] \\
& \qquad\qquad + \mathbb{E}\Big[\big\langle g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big],
\end{aligned}$$

by adding and subtracting $g(x)$. The definition of the decoding function implies the negativity of the second term. Thus, we get:

$$\begin{aligned}
\mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}^* & \leqslant \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big] \\
& \leqslant 2 \sup_{z \in \mathcal{Y}} \|\chi(z)\| \cdot \mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|\Big] \text{ using Cauchy-Schwarz inequality,} \\
& \leqslant 2 \sup_{z \in \mathcal{Y}} \|\chi(z)\| \cdot \sqrt{\cdot \mathbb{E}\Big[\big\|\big\langle\mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|^2\Big]} \text{ using Jensen's inequality,} \\
& = 2R_\ell \cdot \sqrt{\mathcal{R}_S(g) - \mathcal{R}_S^*} \text{ because of Eq. (13.6),}
\end{aligned}$$

which is precisely a calibration function result. A key feature of this result is that the constant $R_\ell$ typically does not explode, even for sets $\mathcal{Y}$ with large cardinality (see examples in Section 13.2.2). To get a learning bound, we then need to use learning bounds for multivariate least-squares regression, which behave similarly to univariate least-squares regression (see Section 13.6). For example, if we assume that the target function $g^*(x) = \mathbb{E}[\psi(y)|x]$ from $\mathcal{X} \to \mathcal{H}$ is in the space of functions that we are using for learning, then penalized least-squares with the proper choice of regularization parameter will lead to explicit convergence rates. Otherwise, we need to let the parameter go to zero to obtain universal consistency. See Ciliberto et al. (2020) for more details.

### 13.4.3 Linear estimators and decoding steps

When the function $g$ is linear in the observations $\psi(y_i)$, $i = 1, \dots, n$ (e.g., local averaging methods from Section 6.2.1, or kernel methods from Section 7.6.1), that is,

$$g(x) = \sum_{i=1}^{n} w_i(x)\psi(y_i),$$

for well-defined functions $w_i : \mathcal{X} \to \mathbb{R}$, we see that the decoding step is

$$\text{dec}(s) \in \arg\min_{z \in \mathcal{Y}} \Big\langle \chi(z), \sum_{i=1}^{n} w_i(x)\psi(y_i) \Big\rangle = \arg\min_{z \in \mathcal{Y}} \sum_{i=1}^{n} w_i(x)\ell(y_i, z), \qquad (13.7)$$

that is, there is no need to know the loss decomposition to run the algorithm. This makes the decoding step even easier with the following examples:

- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with the loss $\ell(y,z) = 1 - \exp\big[-(y-z)^2\big]$. Eq. (13.7) then leads to
$$\arg\max_{z \in \mathbb{R}} \sum_{i=1}^{n} w_i(x) \exp\big[-(y_i - z)^2\big],$$
which is a one-dimensional optimization problem that can be solved by grid search.
- **Multi-category classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i,j) = L_{ij}$. Eq. (13.7) then leads to $\arg\max_{z \in \{1,\dots,k\}} \sum_{i=1}^{n} w_i(x) L_{iz}$.
- **Multiple labels**: $\mathcal{Y} = \{-1,1\}^k$ with $\ell(y,z) = \frac{k}{2} - \frac{1}{2} y^\top z$. Eq. (13.7) then leads to $\arg\max_{z \in \{-1,1\}^k} z^\top \sum_{i=1}^{n} w_i(x) y_i$, which leads to a closed-form formula for $z$.
- **Permutations**: for the pairwise disagreement, the optimization problem does not have a closed form anymore and is an instance of a hard combinatorial problem ("minimum weighted feedback arc set"), which can be solved for small $m$, and with simple approximation algorithms otherwise (see Ciliberto et al., 2020).
- **Sequences**: When using separable loss functions, we return to the classical multiple-label setups. However, when using losses over consecutive pairs, we need to minimize with respect to $z \in \{-1,1\}^m$ a function of the form $\sum_{j=1}^{m} u_j z_j + \sum_{j=1}^{m-1} v_j z_j z_{j+1}$, which can be done in time $O(m)$ by message passing algorithms (see, e.g., Murphy, 2012).

### 13.4.4 Smooth surrogates (♦)

Following Nowak-Vila et al. (2019) and as done in Section 4.1, we can also consider smooth surrogate functions of the form:

$$S(y, g(x)) = c(y) - 2\langle \psi(y), g(x) \rangle + 2a(g(x)),$$

where $a : \mathcal{H} \to \mathbb{R}$ is convex and $\beta$-smooth, that is, for any $h, h' \in \mathcal{H}$, $a(h') \leqslant a(h) + \langle a'(h), h' - h\rangle + \frac{\beta}{2}\|h - h'\|^2$. We also assume that $a(0) = 0$ and that the domain of its Fenchel conjugate includes all $\psi(y)$ for $y \in \mathcal{Y}$. The square loss corresponds to $a(h) = \frac{1}{2}\|h\|^2$ and $c(y) = \|\psi(y)\|^2$.

We consider the decoding function $\mathrm{dec} : \mathcal{H} \to \mathcal{Y}$ equal to

$$\mathrm{dec}(h) \in \arg\min_{z \in \mathcal{H}} \ \chi(z)^\top a'(h).$$

For the square loss, we recover exactly the quadratic surrogate.

**Examples.** Three examples are particularly interesting:

- **Softmax regression:** for multi-category classification, we can always take $\psi(y) = \bar{y} \in \mathbb{R}^k$ the "one-hot" encoding of $y \in \{1, \dots, k\}$. The convex hull of all $\psi(y)$ for $y \in \mathcal{Y}$ is then the simplex in $\mathbb{R}^k$. Softmax regression corresponds to $a(y) = \log\big(\sum_{j=1}^k \exp(x_j)\big)$.
- **Reduction to binary logistic regression:** when $\psi(y) \in \{-1, 1\}^m$, we can consider $a(h) = \sum_{i=1}^m \log \frac{1}{2}(\exp(h_i/2) - \exp(-h_i/2))$, leading to independent logistic regression
- **Graphical models (♦):** The examples above can be made more general using the graphical model framework. We consider sequences in $\{-1, 1\}^m$ for simplicity, but this extends to more general situations, that is, more complex graphical models (see, e.g., Murphy, 2012). For $\psi(y)$ composed of all $y_j$ and $y_j y_{j+1}$. To build the function $a$, we consider the convex hull of all $\psi(y)$, for $y \in \mathcal{Y}$, and for any elements of this convex hull (which corresponds to a probability distribution on $\mathcal{Y}$), we consider its negative entropy which is a convex function $b$. We then take $a$ to be the Fenchel conjugate of $b$.

  For $\psi(y) = y$, we recover independent logistic regression, while for the sequence models, we recover conditional random fields (Sutton and McCallum, 2012).
- **"Perturb-and-MAP":** In situations where one can efficiently maximize linear functions of $\psi(y)$ with respect to $y \in \mathcal{Y}$. In other words, we can compute the convex function $a_0(z) = \max_{y \in \mathcal{Y}} \psi(y)^\top z$. Then we can make it smooth using stochastic smoothing, as presented in Section 11.2, that is, define $a_\sigma = \mathbb{E}\big[a_0(z + \sigma u)\big]$ for a random variable $u \in \mathbb{R}^k$ (typically a Gaussian). When we use Gumbel distributions for $u$, and $\mathcal{Y} = \{1, \dots, k\}$ with $\psi(y) = \bar{y}$, we recover the softmax function, but the framework is mode generally applicable (see Berthet et al., 2020).

**Calibration function.** We then have, by definition of the Fenchel-conjugate $a^*(u) = \sup_{h \in \mathcal{H}} \langle u, h\rangle - a(h)$:

$$\begin{aligned}
\mathcal{R}_S(g) &= \mathbb{E}\Big[\mathbb{E}[c(y)|x] - 2\langle \mathbb{E}[\psi(y)|y], g(x)\rangle + 2a(g(x))\Big] \\
\mathcal{R}_S^* &= \mathbb{E}\Big[\mathbb{E}[c(y)|x] + \inf_{h \in \mathcal{H}} -2\langle \mathbb{E}[\psi(y)|x], h\rangle + 2a(h)\Big] \\
&= \mathbb{E}\big[\mathbb{E}[c(y)|x] - 2a^*(\mathbb{E}[\psi(y)|x])\big], \text{ by definition of } a^*,
\end{aligned}$$

leading to a compact expression of the excess $S$-risk and a lower bound:

$$\begin{aligned}\mathcal{R}_S(g) - \mathcal{R}_S^* &= \mathbb{E}\big[ -2\langle \mathbb{E}[\psi(y)|x], g(x)\rangle + 2a(g(x)) + 2a^*(\mathbb{E}[\psi(y)|x])\big] \\ &\geqslant \frac{1}{\beta}\mathbb{E}\big[\|a'(g(x)) - \mathbb{E}[\psi(y)|x]\|^2\big],\end{aligned}$$

where we have used the $(1/\beta)$-strong-convexity of $a^*$.

Moreover, like in the previous section, we can express the excess risk as:

$$\begin{aligned}\mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}^* &= \mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}(\mathrm{dec}\circ g^*) \\ &= \mathbb{E}\Big[\mathbb{E}\big[\ell(y, \mathrm{dec}\circ g(x)) - \ell(y, \mathrm{dec}\circ g^*(x))\big|x\big]\Big] \\ &= \mathbb{E}\Big[\mathbb{E}\big[\big\langle\psi(y), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big|x\big]\Big] \\ &= \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big], \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big] \\ &= \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - a'(g(x)), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big] \\ &\qquad\qquad + \mathbb{E}\Big[\big\langle a'(g(x)), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big].\end{aligned}$$

By definition of the decoding step, we get:

$$\begin{aligned}\mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}^* &\leqslant \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - a'(g(x)), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big] \\ &\leqslant 2\sup_{z\in\mathcal{Y}}\|\chi(z)\|\cdot\mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - a'(g(x))\big\|\Big] \\ &\leqslant 2\sup_{z\in\mathcal{Y}}\|\chi(z)\|\cdot\sqrt{\cdot\mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|^2\Big]} = 2\sqrt{\beta}R_\ell\cdot\sqrt{\mathcal{R}_S(g) - \mathcal{R}_S^*},\end{aligned}$$

We thus have the same calibration function as for the quadratic surrogate but with an extra factor of $\sqrt{\beta}$. For example, this applies to softmax regression.

**Comparison with quadratic surrogates.** The comparison between quadratic and smooth surrogates for structured prediction mimics the one for binary classification from Section 4.1. While both lead to consistent predictions and similar calibration functions, they differ in their Bayes predictors, with typically smooth surrogates leading to more natural assumptions (see, e.g., Section 13.7.2).

## 13.5 Max-margin formulations

Rather than extending the square or logistic loss from binary classification to structured prediction, we can also extend the hinge loss, leading to "max-margin formulations" in reference to the geometric interpretation from Section 4.1.2. In this section, we assume that for any $y \in \mathcal{Y}$, $z \mapsto \ell(z, y)$ is minimized at $y$, that is, the loss provides a measure of dissimilarity with $y$.

### 13.5.1 Structured SVM

Following Taskar et al. (2005); Tsochantaridis et al. (2005), we consider a traditional extension of the support vector machine with a simple interpretation.

We consider a score function, which is a function of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, with the decoder $\arg\max_{z \in \mathcal{Y}} h(x,z)$. The score $S(y, h(x,\cdot))$ is defined as the minimal $\xi \in \mathbb{R}_+$ such that for all $z \in \mathcal{Y}$,

$$h(x,y) \geqslant h(x,z) + \ell(z,y) - \ell(y,y) - \xi.$$

The intuition behind this definition is that we aim at making $h(x,y)$ larger for the observed $y$ than for the other $h(x,z)$, with a difference that is stronger when $y$ and $z$ are further apart, as measured by the loss.

If we take the particular form $h(x,z) = \langle \psi(z), g(x) \rangle$ for $g : \mathcal{X} \to \mathcal{H}$, then the constraint becomes

$$\langle \psi(y), g(x) \rangle \geqslant \langle \psi(z), g(x) \rangle + \langle \chi(y), \psi(z) \rangle - \langle \chi(y), \psi(y) \rangle - \xi,$$

which is equivalent to

$$\xi \geqslant \langle \psi(z) - \psi(y), \chi(y) + g(x) \rangle,$$

and thus the score function is:

$$S(y, g(x)) = \max_{z \in \mathcal{Y}} \langle \psi(z) - \psi(y), \chi(y) + g(x) \rangle. \tag{13.8}$$

For binary classification with the 0-1 loss, this recovers exactly the SVM. Moreover, this convex loss is computable as soon as we can maximize linear functions of $\chi(z)$; thus, this applies to many combinatorial problems, in particular those described earlier.

However, this approach is not consistent; that is, even in the population case where the test distribution is known, it does not lead to the optimal predictor in general; note that there are subcases, such as multi-category classification with the 0-1 loss and a "majority class", where the approach is consistent (Liu, 2007) (see exercise below).

**Exercise 13.4** *(♦) For the multi-category classification with the 0–1 loss, show that the structural SVM is Fisher-consistent if for all $x \in \mathcal{X}$, $\max_{j \in \{1,\dots,k\}} \mathbb{P}(y = j|x) > \frac{1}{2}$.*

### 13.5.2 Max-min formulations (♦♦)

Following Fathony et al. (2016); Nowak-Vila et al. (2020), we can provide a non-smooth surrogate, which is both consistent and comes with a calibration function that does not have a square root. In the binary case, the support vector machine led to a target surrogate function, which was exactly the Bayes predictor, in values in $\{-1, 1\}$. We will see that it is possible to reproduce in the general case. We still assume that for any $y \in \mathcal{Y}$, $z \mapsto \ell(z,y)$ is minimized at $y$, that is, the loss provides a measure of dissimilarity with $y$.

Given the expression of the Bayes predictor in Eq. (13.5), that is,

$$f^*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \big\langle \chi(z), \mathbb{E}[\psi(y)|x] \big\rangle,$$

we can define the function $g^*(x) = \chi(f^*(x)) \in \mathcal{H}$, which is defined as

$$g^*(x) \in \arg \min_{h \in \chi(\mathcal{Y})} \big\langle h, \mathbb{E}[\psi(y)|x] \big\rangle,$$

which happens to be a subgradient at $\mathbb{E}[\psi(y)|x]$ of the function

$$b : \mu \mapsto -\min_{h \in \chi(\mathcal{Y})} \langle h, \mu \rangle = -\min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle.$$

Thus, if we can design a surrogate function $S$ so that $\mathbb{E}[S(y, g(x))|x]$ has minimizer $g^*(x)$ defined above, we can consider the decoder function with our desired consistent behavior:

$$\text{dec} \circ g(x) \in \arg\min_{y' \in \mathcal{Y}} \ \psi(y')^\top g(x),$$

for which

$$\begin{aligned} \text{dec} \circ g^*(x) \ &\in \ \arg\min_{y' \in \mathcal{Y}} \ \psi(y')^\top g^*(x) = \arg\min_{y' \in \mathcal{Y}} \ \psi(y')^\top \chi(f^*(x)) \\ &= \ \arg\min_{y' \in \mathcal{Y}} \ell(y', f^*(x)) = f^*(x), \end{aligned}$$

because of our assumption on the loss $\ell(y, z)$ being minimizer with respect to $y$ at $z$.

To have a subgradient $g^*(x)$ of $b$ at $\mathbb{E}[\psi(y)|x]$, to be a minimizer of $\mathbb{E}[S(y, g(x))|x]$, it is sufficient to consider (this is not the only choice):

$$S(y, g(x)) = b^*(g(x)) - \langle g(x), \psi(y) \rangle,$$

where $b^*$ is the Fenchel conjugate of $b$ restricted to $\mathcal{M}(\psi) \subset \mathcal{H}$ the closure of the convex hull of all $\psi(z), z \in \mathcal{Y}$, that is,

$$b^*(h) = \max_{\mu \in \mathcal{M}(\psi)} \langle \mu, h \rangle - b(\mu) = \max_{\mu \in \mathcal{M}(\psi)} \langle \mu, h \rangle + \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle.$$

We thus have:

$$\begin{aligned} S(y, g(x)) \ &= \ \max_{\mu \in \mathcal{M}(\psi)} \ \langle g(x), \mu \rangle + \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle - \langle g(x), \psi(y) \rangle \\ &= \ \max_{\mu \in \mathcal{M}(\psi)} \min_{y' \in \mathcal{Y}} \ \langle g(x) + \chi(y'), \mu - \psi(y) \rangle + \ell(y, y'). \end{aligned} \tag{13.9}$$

Note the similarity with the maximum-margin SVM loss in Eq. (13.8), which considers $y' = y$ instead of the minimization with respect to $y' \in \mathcal{Y}$. This extra minimization makes the surrogate loss function more complicated to minimize (though it is still convex) but leads to a Fisher-consistent estimator.

**Fisher consistency.** We now prove that any minimizer $g^*$ of $\mathbb{E}\big[S(y, g(x))\big]$ over all measurable functions from $\mathcal{X}$ to $\mathcal{H}$ leads to the optimal prediction, with

$$\text{dec} \circ g^*(x) = \arg\max_{y' \in \mathcal{Y}} \ \psi(y')^\top g^*(x) = f^*(x).$$

As in Section 13.4.4, for $x \in \mathcal{X}$, any minimizer $g^*$ has a value $g^*(x)$ that minimizes

$$\mathbb{E}\big[S(y, g(x))|x\big] = a(g(x)) - \big\langle g(x), \mathbb{E}[\psi(y)|x]\big\rangle.$$

By definition of $a$, $g^*(x)$ is a minimizer of $h \mapsto \langle h, \mathbb{E}[\psi(y)|x]\rangle$ over $h \in \mathcal{M}(\chi)$. Thus, given the expression of the Bayes predictor in Eq. (13.5), we get $g^*(x) = \chi(f^*(x)) \in \mathcal{H}$. This leads to $\operatorname{dec} \circ g^*(x) = f^*(x)$ because of the assumption that $z \mapsto \ell(z, y)$ is minimized at $y$. We can also get a linear calibration function in generic situations; see Nowak-Vila et al. (2020) for details.

**Optimization algorithms.** In this book, we have focused on optimization methods based on subgradients. For the loss defined in Eq. (13.9), this requires to have an optimizer $\mu \in \mathcal{M}(\psi)$, which requires solving a min-max problem in general. Below, we consider the multi-category classification problem with 0-1 loss, where this can be achieved, and refer to Nowak-Vila et al. (2020) for algorithms based on primal-dual formulations.

**Binary classification with the 0-1 loss.** In this situation, we can compute $a^*(\mu) = \frac{1}{2}|\mu|$ with domain $[-1, 1]$, leading to $a(v) = (|v| - 1/2)_+$, and a formulation that is close to the binary SVM (but non-identical).

**Multi-category classification.** for $\mathcal{Y} = \{1 \ldots, k\}$ and the 0-1 loss, a short exercise shows that we obtain

$$S(y, g(x)) = \max_{A \subset \{1,\ldots,k\},\ A \neq \varnothing} \frac{\sum_{j \in A} g_j(x) + |A| - 1}{|A|},$$

where the maximizers in $A$ can be obtained in closed form (together with the subgradient).

## 13.6 Generalization bounds (♦)

In this section, we provide generalization bounds for the structured prediction problem with smooth convex surrogates defined in Section 13.4. For simplicity, we will assume a linear model with feature vector $\varphi : \mathcal{X} \to \mathbb{R}^d$, and that the feature vector is flexible enough so that the minimizer of the expected surrogate risk is indeed a linear function of $\varphi$, that is, $g^{(\theta)}(x) = \theta^\top \varphi(x)$. Taking care of an approximation error would lead to developments similar to Section 7.5.1.

For real-valued prediction functions and linear models, we looked at two frameworks to obtain generalization bounds: one based on Rademacher complexities and one based on stochastic gradient descent. For multi-category classification in Section 13.1, we considered both and only considered SGD in this section as it leads to better bounds. Moreover, we could use kernel methods when $\varphi$ is only known through the associated kernel function (using in particular Section 7.4.5), but we stick to explicit feature maps for simplicity. Finally, for least-squares, we could directly extend the ridge regression analysis from Section 7.6, which does not use Rademacher complexities. We instead focus on

Lipschitz-continuous losses, that is, we assume that $a$ has gradients bounded by $G$ in $\ell_2$-norm.

We thus assume that there exists $\theta_* \in \mathbb{R}^{d \times k}$ such that $a'(\mathbb{E}[\psi(y)|x)]) = \theta_*^\top \varphi(x)$, and thus $\mathcal{R}_S^* = \mathbb{E}\big[S(y, \theta_*^\top \varphi(x))\big]$. We assume i.i.d. observations $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, and the single pass gradient descent recursion, initialized at 0 and defined for $t \in \{1, \dots, n\}$, as

$$\theta_t = \theta_{t-1} - \gamma_t \varphi(x_t) S'(y_t, \theta_{t-1}^\top \varphi(x_t))^\top M,$$

for a positive definite matrix $M^{1/2}$, which will allow to include some penalty function of the form $\mathrm{tr}[\theta\theta^\top M^{-1}]$.

The analysis of Section 5.4 exactly applies, and with the choice $\gamma_t = \gamma$, we obtain the following generalization bound:

$$\mathbb{E}\big[\mathcal{R}_S(g_{(\theta_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{1}{2\gamma n} \mathrm{tr}[\theta_*^\top M^{-1} \theta_*] + \frac{\gamma G^2 R^2}{2}.$$

With the optimal choice of $\gamma$, we get:

$$\mathbb{E}\big[\mathcal{R}_S(g_{(\theta_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{GR\big(\mathrm{tr}[\theta_*^\top M^{-1} \theta_*]\big)^{1/2}}{\sqrt{n}}.$$

We can then use the calibration result to obtain a consistency result for the structured prediction problem with a bound in $n^{-1/4}$.

**Examples of structured regularization.** The prediction function is characterized by a matrix $\theta \in \mathbb{R}^{d \times k}$, and without further knowledge, it is natural to use the Frobenius norm or the nuclear norm as regularization or constraint. In particular, set-ups where the $k$ columns have a specific structure, some particular squared norm $\mathrm{tr}[\theta\theta^\top M^{-1}]$ can be natural.

For example, in the ranking problem with the pairwise disagreement loss, where $\psi(y)$ is indexed by two indices $i, j \in \{1, \dots, m\}$, it is natural to consider $\theta_{ij} = \eta_i - \eta_j$ for a matrix $\eta \in \mathbb{R}^{d \times m}$ (see Section 13.7.2 for more details).

## 13.7 Experiments

In this section, we present two experiments highlighting the benefits of structured prediction and illustrating the results from this chapter.

### 13.7.1 Robust regression

We consider a toy robust regression problem to illustrate the use of the quadratic surrogates presented in Section 13.4.1. We use a simple one-dimensional robust regression problem, where we compare the square loss and the loss $\ell(y, z) = 1 - \exp(-(y - z)^2)$. We

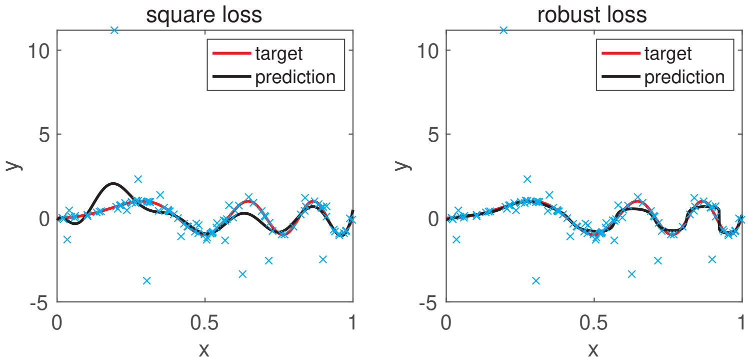


Figure 13.1: Robust regression in one dimension, with heavy-tail noise (fifth power of Gaussian noise): regular square loss (left), vs. robust loss (right).

generate data with heavy-tail additive noise and plot below the best performance for kernel ridge regression with the Gaussian kernel, with the optimal regularization parameter (selected for test performance).

Since we use a kernel method, we can use Section 13.4.3, that is, once the $n$ data-dependent weight functions $w_1(x), \dots, w_n(x)$ are estimated using ridge regression, we need at test time to compute $\arg\min_{z\in\mathbb{R}} \sum_{i=1}^n w_i(x)\ell(y_i, z) = \arg\min_{z\in\mathbb{R}} \sum_{i=1}^n w_i(x)(1-\exp(-(y_i - z)^2))$, which can be done by grid search. See results in Figure 13.1, where we see that the robust loss is indeed more robust to outliers.

### 13.7.2 Ranking

We illustrate structured prediction on a ranking problem, where $\mathcal{Y}$ is the set of permutations from $\{1, \dots, m\}$ to $\{1, \dots, m\}$. We consider two different loss functions:

- **Square loss:** $\ell(y, x) = \sum_{i=1}^m (y(i) - z(i))^2$, with $\psi(y) = y \in \{1, \dots, m\}^m$. For this loss, we only consider the square loss. We thus need to fit a function $h : \mathcal{X} \to \mathbb{R}^m$ using least-squares directly on $y$. The decoding step a for test point $x$ is then simply to sort the $m$ components of $h(x)$.

- **Pairwise disagreement:** $\ell(y, z) = \sum_{i,j=1}^m \big(1_{y(i)<y(j)} - 1_{z(i)<z(j)}\big)^2$, with the feature $\psi(y) \in \{-1, 1\}^{m(m-1)/2}$ defined as $\psi(y)_{ij} = 1_{y(i)<y(j)}$ for $i < j$. We thus need to learn a function $g : \mathcal{X} \to \mathbb{R}^{m(m-1)/2}$; we consider the square loss, where we minimize expectations of $\|\psi(y) - h(x)\|_2^2$, as well as the logistic loss, where we minimize the expectation of $\sum_{i<j} \log\big(1 + \exp(-\psi(y)_{ij} h_{ij}(x))\big)$.

  In terms of decoding, given the function $h$, at test point $x$, we need to maximize $\sum_{i<j} g_{ij}(x) 1_{z(i)<z(j)}$ with respect to a permutation $z$ when using the square loss, while we need to maximize $\sum_{i<j} \tanh(g_{ij}(x)) 1_{z(i)<z(j)}$ when using the logistic loss. This is an instance of the "minimum feedback arc set problem", which is an NP-hard problem with known approximation algorithms (Ailon et al., 2008). When the weights $g_{ij}(x)$ are of the form $u(h_j(x) - h_i(x))$ for a non-decreasing function $u$ and

a function $g : \mathcal{X} \to \mathbb{R}^m$, then it can be solved by sorting $h$. Thus, when either the square loss or the logistic loss is used, using a specific model $g_{ij} = h_i - h_j$ leads to simpler decoding. For the square loss, using this specific model leads exactly to performing least-squares on $y \in \mathbb{R}^m$.

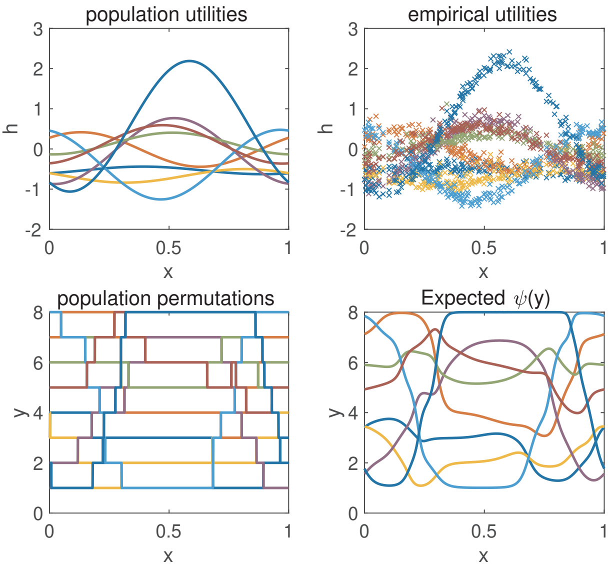


Figure 13.2: Top Left: utilities $h_1, \dots, h_m$, top right: empirical utilities $h_1 + \eta_1, \dots, h_m + \eta_m$. Bottom: population permutations $y^*(x) \in \{1, \dots, m\}$ (left) and conditional expectation $\mathbb{E}[y|x] \in [1, m]$ (right).

**Placket-Luce model.** We generate data from the Plackett-Luce model (Marden, 1996), that is, from $m$ functions $h_1(x), \dots, h_m(x)$, with the random permutation obtained by sorting the $m$ real values $h_1(x) + \eta_1, \dots, h_m(x) + \eta_m$ in ascending order, where each $\eta_i$, $i = 1, \dots, m$, is a Gumbel random variable. Our convention is that $y(i)$ is the position of item $i$, that is, $y(i) = 1$ if $h_i(x) + \eta_i$ is the lowest, and $y(i) = m$ is $h_i(x) + \eta_i$ is the largest.

If $\pi_i(x) = \text{softmax}(h(x))$, this happens to be equivalent to the model where $y(m)$ is selected with probability vector $\pi(x)$, and then $y(m-1)$ with probability vector proportional to $\pi(x)$ (but without the possibility of taking $y(1)$. In other words, the probability of selecting a permutation $z$ is equal to:

$$\pi(x)_{z(m)} \frac{\pi(x)_{z(m-1)}}{1 - \pi(x)_{z(m)}} \frac{\pi(x)_{z(m-2)}}{1 - \pi(x)_{z(m)} - \pi(x)_{z(m-1)}} \cdots \frac{\pi_{z(2)}}{\pi_{z(1)} + \pi(z(2))}.$$

Moreover, we also have $\mathbb{E}[1_{y(i)<y(j)}|x] = \frac{\pi_j(x)}{\pi_j(x)+\pi_i(x)} = \frac{\exp(h_j(x))}{\exp(h_i(x))+\exp(h_j(x))}$, so that a logistic regression model for predicting $1_{y(i)<y(j)}$ has a target function equal to $h_j - h_i$.

However, the target function for the square loss, $\mathbb{E}[y|x]$, can be expressed as products of softmax functions of subsets of $h_1, \dots, h_m$, and thus are not linear in these functions.

We consider $\mathcal{X} = [0,1]$ and consider functions $h_1, \dots, h_m$ which are linear combinations of cosine functions $\cos(2\pi kx)$ and sine function $\cos(2\pi kx)$ for $k \in \{0,1\}$ for the generating functions. See Figure 13.2 for an illustration.

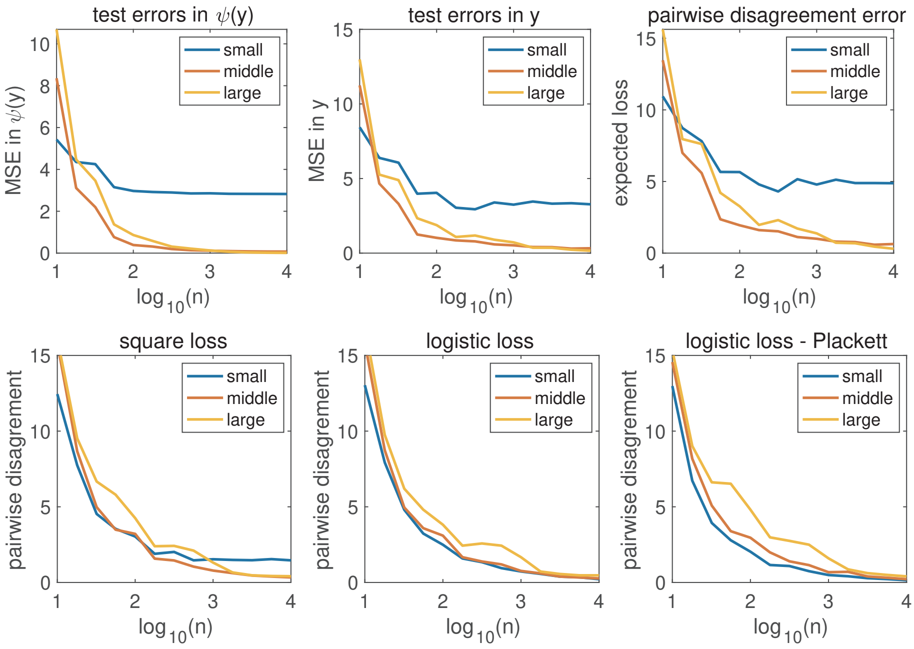


Figure 13.3: Top: using the square loss $\ell(y,z) = \|y - z\|_2^2$ with the square surrogate. Bottom: using the pairwise disagreement loss $\ell(y,z) = \|\psi(y) - \psi(z)\|_2^2$, with the square surrogate, and logistic surrogate.

We consider situations where the prediction model we use for the functions $g$ and $h$ includes the ones generating the data, hence a well-specified model for the logistic loss, but not for the square loss. In Figure 13.3, we provide learning curves where we vary the number $n$ of observations, for three classes of prediction functions based on sines and cosines: for the small model, we use exactly the same model class as for $h_1, \dots, h_m$, that is, $k \in \{0,1\}$, while for the middle model, $k \in \{0, \dots, 3\}$, and for the large model, $k \in \{0, \dots, 15\}$. We use a fixed regularization parameter proportional to $1/n$.

In the top-left plot of Figure 13.3, the quadratic surrogate with $\psi(y) = y$ is considered, and the small model is not well-specified. Thus, when $n$ grows, the testing error does not go down to zero, similarly for the bottom-left plot where the square loss is used with the pairwise disagreement loss function. For the logistic loss, however, where even the small model is well-specified, we obtain a learning curve that goes closer to zero.

## 13.8 Conclusion

In this chapter, we explored surrogate frameworks beyond binary classification, focusing on convex surrogates. These convex formulations can be used with any prediction functions (linear in the parameter, such as kernel methods, or not, such as neural networks) and come with guarantees for linear models. We presented several principles, e.g., margin-based techniques or frameworks with probabilistic interpretation through maximum likelihood.

# Chapter 14

# Probabilistic methods

**Chapter summary**

- Probabilistic models can lead to intuitive algorithmic formulations but sometimes misleading interpretations. In particular, maximum a posteriori (MAP) estimation does *not* work best when the parameters are generated from the prior distribution. Minimum mean-square estimation (MMSE) is preferable (for the square loss).
- Generative models (such as linear discriminant analysis) that explicitly try to model the input data with simple models can lead to biased but efficient estimators in large dimensions compared to their discriminative counterparts (such as logistic regression).
- Bayesian inference can be used naturally for model selection using the marginal likelihood, both for model selection among a finite number of choices or with Gaussian processes.
- PAC-Bayesian analysis: aggregating estimators provide natural statistically efficient estimators with an elegant link with Bayesian inference.

In this chapter, we first consider probabilistic modeling interpretations of several learning methods, focusing primarily on identifying losses and priors with log-densities but drawing clear distinctions between what this analogy brings and what it does not. We then show how Bayesian inference naturally leads to model selection criteria and end the chapter with a description of PAC-Bayes analysis.

## 14.1 From empirical risks to log-likelihoods

Many methods in machine learning may be given a probabilistic interpretation through maximum likelihood or "maximum a posteriori" (MAP ) estimation. For example, consider the regularized empirical risk as:

$$\hat{\mathcal{R}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \frac{\lambda}{n}\Omega(\theta),$$

multiply by $-n$ and take the exponential to get:

$$\begin{aligned} \exp(-n\hat{\mathcal{R}}(\theta)) &= \exp\Big(-\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) - \lambda\Omega(\theta)\Big) \\ &= \prod_{i=1}^{n} \exp\big[-\ell(y_i, f_\theta(x_i))\big] \cdot \exp\big[-\lambda\Omega(\theta)\big]. \end{aligned} \tag{14.1}$$

We can give a probabilistic interpretation by considering a *likelihood*, that is, a density (with respect to a well-defined base measure),

$$p(y_i|x_i, \theta) \propto \exp\big[-\ell(y_i, f_\theta(x_i))\big],$$

and a *prior* density

$$p(\theta) \propto \exp\big[-\lambda\Omega(\theta)\big],$$

so that we have:

$$\exp(-n\hat{\mathcal{R}}(\theta)) \quad \propto \quad \prod_{i=1}^{n} p(y_i|x_i, \theta) \cdot p(\theta),$$

which is precisely the (conditional) likelihood for the model where $\theta$ is a parameter and where, given $\theta$, all pairs $(x_i, y_i)$ are independent and identically distributed.

⚠ Overloading of notations for probability densities, where the symbol $p$ is used for all random variables.

⚠ Note the difference between conditional likelihood and likelihood.

⚠ There is more to probabilistic interpretation than simply taking the exponential, e.g., generative models, Bayesian inference for hyperparameter learning (as done in later sections), dealing with missing data through the expectation-maximization algorithm, etc.

⚠ We only scratch the surface here, and from a learning theory point of view. See Murphy (2012); Bishop (2006) for many more details.

In this section, we primarily focus on the formulation in Eq. (14.1) and now look at specific examples for data likelihoods and priors.

### 14.1.1 Conditional likelihoods

For logistic regression where $\mathcal{Y} \in \{-1, 1\}$, we can interpret the loss as the conditional log-likelihood of the model where

$$\mathbb{P}(y_i = 1|x_i) = \frac{1}{1 + \exp(-f_\theta(x_i))},$$

which can be put in a compact way as $p(y_i|x_i) = \frac{1}{1+\exp(-y_i f_\theta(x_i))} = \sigma(y_i f_\theta(x_i))$.

⚠ To apply logistic regression, there is no need to assume that the model is well-specified, that is, there exists a $\theta_*$ so that the data are actually generated from the model above. For the non-parametric analysis, this is often assumed.

For least-squares regression, we can interpret the loss $\frac{1}{2}(y_i - f_\theta(x_i))^2$ as a Gaussian model with mean $f_\theta(x_i)$ and variance 1. We can also estimate a more general variance parameter that is uniform across all $x$ (homoscedastic regression) or depends on $x$ (heteroscedastic regression).

⚠ No need to have Gaussian noise! Simply zero mean and bounded variance are enough for the analysis.

**Exercise 14.1** *Show that the negative log-density of the Gaussian distribution with mean $\mu$ and variance $\sigma^2$, that is, $-\log p(y|\mu, \sigma) = \frac{1}{2\sigma^2}(x-\mu)^2 + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log\sigma^2$ is not convex in $(\mu, \sigma^2)$, but is jointly convex in $(\mu/\sigma^2, \sigma^{-2})$.*

### 14.1.2 Classical priors

We can interpret classical regularizers that we have already encountered in previous chapters. For the squared $\ell_2$-norm with $\Omega(\theta) = \frac{\lambda}{2}\|\theta\|_2^2$, this corresponds to a Gaussian distribution with mean zero and covariance matrix $\lambda^{-1}I$.

For the $\ell_1$-norm with $\Omega(\theta) = \lambda\|\theta\|_1$, this is the so-called Laplace (or double exponential) prior:

$$p(\theta) = \prod_{j=1}^{d} \frac{\lambda}{2}\exp(-\lambda|\theta_j|).$$

**Exercise 14.2** *Show that the variance of a Laplace-distributed random variable is equal to $\frac{2}{\lambda^2}$.*

The interactions between regularization terms and priors can go both ways, and we can consider other classical priors. One that will be useful later in the Bayesian setting is the multivariate Student distribution (often used marginally for independent components):

$$p(\theta) \propto \left(\beta + \tfrac{1}{2}\|\theta\|_2^2\right)^{-\alpha-d/2},$$

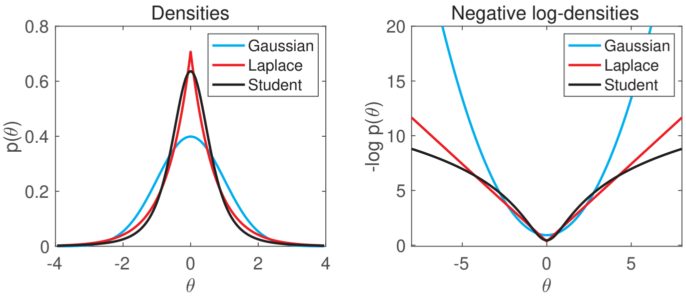


Figure 14.1: Classical priors in one dimension, all normalized to zero mean and unit variance: (left) densities, (right) negative log-densities.

leading to the regularizer $(\alpha + d/2)\log(\beta + \frac{1}{2}\|\theta\|_2^2)$, which is not convex in $\theta$. This will be used within sparse priors in the next section.

**Exercise 14.3** *(♦) We consider a random vector $\theta$ which is Gaussian with mean zero and covariance matrix $\eta I$, with $1/\eta$ being distributed as a Gamma random variable with parameters $\alpha$ and $\beta$, that is, $\eta$ with density $p(\eta) = \frac{\beta^\alpha}{\Gamma(\alpha)}(1/\eta)^{\alpha+1}\exp(-\beta/\eta)$. Show that the marginal density of $\theta$ is the Student distribution with density $p(\theta) = c(\beta + \frac{1}{2}\|\theta\|_2^2)^{-\alpha-d/2}$, with $c = \frac{1}{(2\pi)^{d/2}}\frac{\beta^\alpha\Gamma(\alpha+d/2)}{\Gamma(\alpha)}$, and that $\mathbb{E}\big[\theta\theta^\top\big] = \frac{\beta}{\alpha-1}I$ if $\alpha > 1$.*

This can be misleading as even when the target function is sampled from the prior, MAP estimation may not work. See Section 14.1.4.

The expression of regularizers as log-densities may lead to the impression that MAP estimation is particularly well suited when (1) the conditional model is well-specified, that is, there exists $\theta_*$ such that $p(y|x)$ is indeed proportional to $\exp(-\ell(y, f_{\theta_*}))$, and (2) the optimal $\theta_*$ is sampled from the prior distribution proportional to $\exp(-\lambda\Omega(\theta))$. As we now explain, this is not the case *at all*.

### 14.1.3 Sparse priors

As shown in the next section, the Laplace prior is not a good prior for sparse data. We consider the following ones instead. For each one-dimensional component, we consider:

- Generalized Gaussians: $p(\theta) = \dfrac{\alpha}{2}\dfrac{\lambda^{1/\alpha}}{\Gamma(1/\alpha)}\exp(-\lambda|\theta|^\alpha)$, with variance $\lambda^{-2/\alpha}\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}$.
- Student: $p(\theta) = \frac{1}{(2\pi)^{1/2}}\frac{\beta^\alpha\Gamma(\alpha+1/2)}{\Gamma(\alpha)}1(\beta + \frac{1}{2}\theta^2)^{-\alpha-1/2}$, with variance $\frac{\beta}{\alpha-1}$ if $\alpha > 1$.
- Mixture of two Gaussians: $p(\theta) = \alpha\mathcal{N}(\theta|0, \sigma_0^2) + (1-\alpha)\mathcal{N}(\theta|0, \tau^2)$, with variance $\alpha\sigma_0^2 + (1-\alpha)\tau^2$.

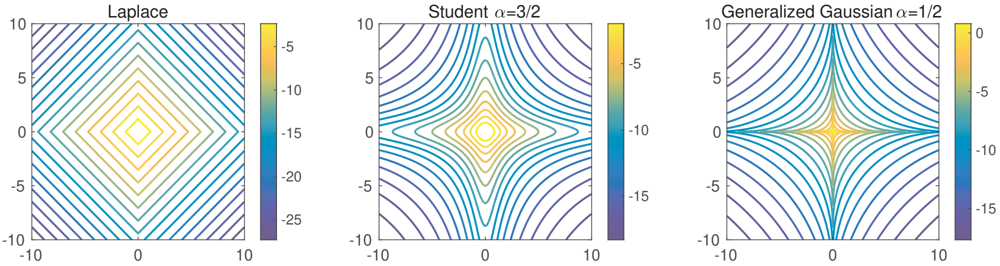


Figure 14.2: Sparse priors in two dimensions.

It turns out that all of these examples happen to be "scale mixtures of Gaussians", that is, they can be seen as the (potentially continuous) mixtures of Gaussian distributions with zero mean but different variances:

$$p(\theta) = \int_0^{+\infty} \frac{1}{\sqrt{2\pi\eta}} e^{-\frac{1}{2}\frac{\theta^2}{\eta}} dq(\eta),$$

where $q$ is a probability measure on $\mathbb{R}_+$. For the third example, this is straightforward, with $q$ being a weighted sum of two Diracs at $\sigma_0^2$ and $\tau^2$. For the Laplace distribution (generalized Gaussians with $\alpha = 1$), one can check by direct integration that we can take $q$ to be an exponential distribution, that is, with density $q(\eta) = \frac{\lambda^2}{2} \exp(-\eta\lambda^2/2)$, while for the Student distribution, $q$ has an inverse Gamma distribution, with density $q(\eta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \eta^{-\alpha-1} e^{-\beta/\eta}$ (see Exercise 14.3).

As we show in Section 14.3.2, this hierarchical model can be used with marginal likelihood maximization, leading to reweighted least-squares algorithms that are close to the "$\eta$-trick" from Section 8.3.1, and thus provides a Bayesian interpretation.

**Exercise 14.4** *A density $p(\theta)$ on $\mathbb{R}$ is said super-Gaussian if $\log p(\theta)$ is convex in $\theta^2$ and non-increasing. Show that scale mixtures of Gaussians are super-Gaussian.*[1]

### 14.1.4 On the relationship between MAP and MMSE (♦)

In this section, following Gribonval (2011), we consider a very simple conditional model of the form

$$y = \theta + \varepsilon, \tag{14.2}$$

where $\varepsilon$ is normal with zero mean and covariance matrix $\sigma^2 I$, assuming $\sigma^2$ is known. We have prior knowledge on $\theta$ in the form of a prior density $q(\theta)$. Given the observation of $y$, our goal is to obtain an estimator of $\theta$ with the most favorable properties, which we define here as the minimum squared error (this estimator will be generalized in Section 14.3).

That is, given an estimator $\hat{\theta}(y)$, we consider the criterion:

$$J(\hat{\theta}) = \int_{\mathbb{R}^d} q(\theta) \|\theta - \hat{\theta}(y)\|_2^2 d\theta.$$

[1] The converse is not true, see Palmer et al. (2005).

As shown in Section 2.2.3, the optimal estimator (i.e., function) $\hat{\theta} : \mathbb{R}^d \to \mathbb{R}^d$ is the *a posteriori mean*, that is

$$\hat{\theta}_{\text{MMSE}}(y) = \mathbb{E}\big[\theta|y\big],$$

assuming that $\theta$ is sampled according to $q(\theta)$ and $y$ follows the model in Eq. (14.2). Here, MMSE stands for "minimum mean-square error". We now want to compare it with the maximum a posteriori parameter

$$\hat{\theta}_{\text{MAP}}(y) \in \arg\max_{\theta \in \mathbb{R}^d} \ p(\theta|y) = \arg\max_{\theta \in \mathbb{R}^d} \ q(\theta)p(y|\theta).$$

**Gaussian prior.** When $q$ is a Gaussian distribution with mean zero and covariance matrix $\tau^2 I$, then $(\theta, y)$ is a Gaussian vector and from conditioning results presented in Section 1.1.3, we have

$$\hat{\theta}_{\text{MMSE}}(y) = \mathbb{E}\big[\theta|y\big] = \frac{\tau^2}{\tau^2 + \sigma^2} y,$$

while the MAP estimate is also equal to $\frac{\tau^2}{\tau^2+\sigma^2} y$ because, for Gaussians, the mean and the mode are the same, but, as we will show later, Gaussian priors are the only ones for which these two are equal.

**Simple expression of the MMSE.** We denote by $p(y)$ the density of $y$, that is,

$$\begin{aligned} p(y) &= \int_{\mathbb{R}^d} p(y, \theta) d\theta = \int_{\mathbb{R}^d} p(\theta) p(y|\theta) d\theta \\ &= \int_{\mathbb{R}^d} q(\theta) \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\big(-\frac{1}{2\sigma^2}\|\theta - y\|_2^2\big) d\theta, \end{aligned}$$

using the expression of the Gaussian density. We can now express the a posteriori mean as, introducing the gradient of the Gaussian density:

$$\begin{aligned} \hat{\theta}_{\text{MMSE}}(y) &= \mathbb{E}\big[\theta|y\big] = \int_{\mathbb{R}^d} \frac{p(\theta, y)}{p(y)} \theta d\theta \\ &= y + \sigma^2 \int_{\mathbb{R}^d} \frac{p(y|\theta)p(\theta)}{p(y)} \frac{1}{\sigma^2}(\theta - y) d\theta \\ &= y + \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q(\theta) \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\big(-\frac{1}{2\sigma^2}\|\theta - y\|_2^2\big) \frac{1}{\sigma^2}(\theta - y) d\theta \\ &= y - \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q(\theta) \frac{\partial}{\partial\theta}\Big[\exp\big(-\frac{1}{2\sigma^2}\|\theta - y\|_2^2\big)\Big] d\theta. \end{aligned}$$

Thus, using integration by parts, we get:

$$\begin{aligned} \hat{\theta}_{\text{MMSE}}(y) &= y + \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q'(\theta) \exp\big(-\frac{1}{2\sigma^2}\|\theta - y\|_2^2\big) d\theta \\ &= y + \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q'(y - \eta) \exp\big(-\frac{1}{2\sigma^2}\|\eta\|_2^2\big) d\eta \\ &= y + \frac{\sigma^2}{p(y)} p'(y) = y + \sigma^2 \frac{d}{dy}(\log p(y)). \end{aligned} \tag{14.3}$$

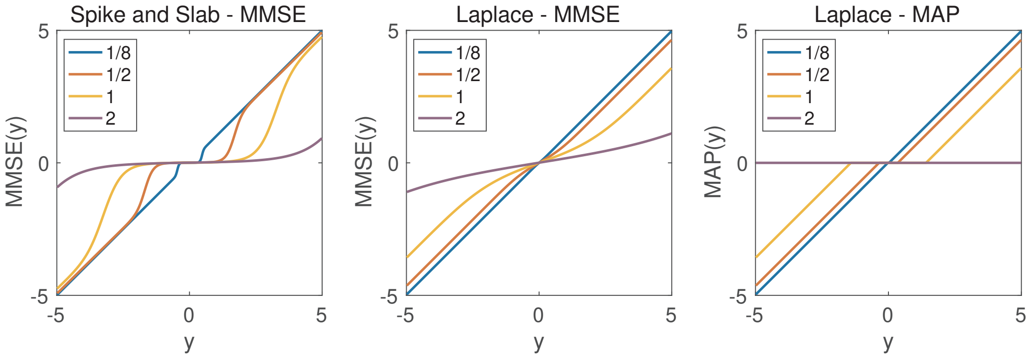


Figure 14.3: Comparison of MMSE and MAP for the spike-and-slab and Laplace priors: MMSE for the spike-and-slab prior (left), MMSE for the Laplace prior (middle), MAP for the Laplace prior (right).

We thus get an explicit expression of the minimum mean square error estimate. Note that for a Gaussian prior, $y$ is (marginally) normally distributed; hence, the gradient of $\log p(y)$ is a linear function, and the MMSE is affine in $y$ if and only if the prior is Gaussian.

**Exercise 14.5** *(♦) Show that the posterior covariance matrix can be expressed as* $\text{var}(\theta|y) = \sigma^2 I + \sigma^4 \dfrac{d^2}{dydy^\top}(\log p(y))$.

**Expression of the MAP estimate.** If $q(\theta) = \exp(-h(\theta))$, then the MAP estimate is

$$\hat{\theta}_{\text{MAP}}(y) \in \arg\max_{\theta \in \mathbb{R}^d} \frac{1}{2\sigma^2}\|\theta - y\|_2^2 + h(\theta),$$

with optimality condition, for differentiable $h$, $\theta - y - \sigma^2 \frac{d}{d\theta}(\log q(\theta)) = 0$; thus we have:

$$\hat{\theta}_{\text{MAP}}(y) = y + \sigma^2 \frac{d}{dy}(\log q)\big[\hat{\theta}_{\text{MAP}}(y)\big]. \tag{14.4}$$

**Exercise 14.6** *(♦♦) We denote by $f(y) = -\log p(y)$. Show that the MMSE estimator $\hat{\theta}_{\text{MMSE}}(y) = y - \sigma^2 f'(y)$ defined in Eq. (14.3) is the MAP estimator for the negative log-prior $g$ that satisfies $g(\hat{\theta}_{\text{MMSE}}(y)) = f(y) - \frac{\sigma^2}{2}\|f'(y)\|_2^2$ for all $y \in \mathbb{R}^d$.*

**Differences between MMSE and MAP.** Given the expressions in Eq. (14.3) and Eq. (14.4), we can now study how the two estimators differ for the various sparse priors that we have described above, where we consider the one-dimensional case for simplicity (which extends to independent marginal priors in the multi-dimensional case) (see plots in Figure 14.3):

- Spike-and-slab: this is the model essentially used in the analysis of the Lasso in Chapter 8, for which MAP with the Laplace prior (that is, the Lasso) is shown to have favorable properties. We consider the prior, which is the mixture of a Dirac at zero (with weight $\alpha$) and a Gaussian with mean zero and variance $\tau^2$. The variance is then equal to $(1-\alpha)\tau^2$, and $p(y)$ is the mixture of two Gaussian distributions, centered in zero, with variances $\sigma^2$ and $\sigma^2+\tau^2$.

  **Exercise 14.7** *Show that the marginal density $p(y)$ for the spike-and-slab prior is equal to $p(y) = \alpha \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\big(-\frac{y^2}{2\sigma^2}\big) + (1-\alpha)\frac{1}{(2\pi(\sigma^2+\tau^2))^{1/2}} \exp\big(-\frac{y^2}{2(\sigma^2+\tau^2)}\big)$. Provide an expression of $\hat{\theta}_{\rm MMSE}(y)$ and of $\hat{\theta}_{\rm MAP}(y)$.*

- Laplace: this is the model for which the MAP estimation leads to the Lasso method. For $q(\theta) = \frac{2}{\lambda}\exp(-\lambda|\theta|)$, the variance is equal to $2/\lambda^2$. We can compute the MMSE by explicitly computing $p(y)$ by integrating separately over positive and negative numbers (see the exercise below). We see in Figure 14.3 that the MMSE is very far from the soft-thresholding operator from Section 8.3.1. In other words, the Lasso is not adapted to signals that are sampled from the Laplace distribution but rather to signals sampled from the spike-and-slab prior.

  **Exercise 14.8** *Show that the marginal density $p(y)$ for the Laplace prior can be expressed using the Gauss error function ${\rm erf}(\alpha) = \frac{2}{\sqrt{\pi}}\int_0^\alpha \exp(-t^2)dt$, as: $p(y) = \frac{\lambda}{4}\exp\big(\frac{\lambda^2\sigma^2}{2} - \lambda y\big)\big[1-{\rm erf}\big(\frac{\lambda\sigma - \frac{y}{\sigma}}{\sqrt{2}}\big)\big] + \frac{\lambda}{4}\exp\big(\frac{\lambda^2\sigma^2}{2} + \lambda y\big)\big[1-{\rm erf}\big(\frac{\lambda\sigma + \frac{y}{\sigma}}{\sqrt{2}}\big)\big]$. Provide an expression of $\hat{\theta}_{\rm MMSE}(y)$ and of $\hat{\theta}_{\rm MAP}(y)$.*

**Exercise 14.9** *When $q$ is a Gaussian distribution with mean zero and covariance matrix $C$, provide an expression of the MMSE and MAP estimates.*

**Exercise 14.10** *(♦) Provide a closed-form expression for the marginal density $p(y)$ for the Student prior.*

## 14.2 Discriminative vs. generative models

We consider a traditional supervised learning set-up, with $(x,y) \in \mathcal{X}\times\mathcal{Y}$. The goal is for any $x\in\mathcal{X}$ to obtain a good conditional predictive model of $y$ given $x$, that is, to get a good model for $p(y|x)$.

We can first directly model $p(y|x)$ with a parameterized conditional model (like done for least-squares or logistic regression). This will be called the *discriminative* approach.

We can also consider a joint density $p(x,y)$, and obtain $p(y|x) = \frac{p(x,y)}{p(x)} \propto p(x,y)$ using Bayes rule. Most often (in particular for classification problems), the joint model is obtained by modeling $y$ and $x|y$, that is, the conditional model of the inputs given the outputs, with a particularly simple model. This will be called the *generative* approach.

### 14.2.1 Linear discriminant analysis and softmax regression

We consider a generative model with Gaussian class-conditional densities with a common covariance matrix, with $x \in \mathbb{R}^d$ and $y \in \{1, \dots, k\}$:

$$\begin{array}{rcl} y & \sim & \text{multinomial}(\pi) \\ x|y=i & \sim & \text{Gaussian}(\mu_i, \Sigma). \end{array}$$

We can then compute the distribution of $y$ given $x$ as (removing all parts that are independent of $i$):

$$\begin{array}{rcl} \mathbb{P}(y=i|x) & \propto & \mathbb{P}(y=i,x) = \pi_i \exp\big[-\frac{1}{2}(x-\mu_i)^\top\Sigma^{-1}(x-\mu_i)\big] \\ & \propto & \pi_i \exp\big[-\frac{1}{2}\mu_i^\top\Sigma^{-1}\mu_i\big]\exp(\mu_i^\top\Sigma^{-1}x). \end{array}$$

This implies that, defining the softmax function $\text{softmax} : \mathbb{R}^k \to \mathbb{R}^k$ through $\text{softmax}(v)_j = \frac{e^{v_j}}{e^{v_1}+\cdots+e^{v_k}}$:

$$\mathbb{P}(y=i|x) = \text{softmax}\big[(\mu_i^\top\Sigma^{-1}x + \log\pi_i - \tfrac{1}{2}\mu_i^\top\Sigma^{-1}\mu_i)_i\big] = \text{softmax}\big[(w_i^\top x + b_i)_i\big]_i,$$

that is, the conditional model is the softmax function of a linear model, which is precisely the definition of softmax regression from Section 13.1.1, with $w_i = \Sigma^{-1}\mu_i$, and $b_i = \log\pi_i - \frac{1}{2}\mu_i^\top\Sigma^{-1}\mu_i$. The availability of a generative model will lead to alternative parameter estimation algorithms (see below). Note that (a) for $k=2$, we recover logistic regression and that (b) we can apply the softmax regression model for any set of $k$ prediction functions $f_1, \dots, f_k$ beyond affine functions.

Note, finally, that the common covariance matrix is often restricted to be diagonal.

**Maximum likelihood estimation.** Given observations $(x_1, y_1), \dots, (x_n, y_n)$, the parameters of the model above can be estimated naturally by maximum likelihood. It turns out that for the particular case of multinomial and Gaussian random variables, this is equivalent to computing empirical moments (proof left as an exercise), that is, the estimator or $\pi$ is the vector of empirical proportions of each class. In contrast, the estimator of each mean $\mu_i$ is the empirical mean of observations with class $i$, and the joint covariance is a weighted combination of the empirical covariances of each class.

**Exercise 14.11 (Quadratic discriminant analysis)** *Assume that the class-conditional covariance matrices are different for each class. Show that the conditional model is still a softmax function, but now of "affine + quadratic" functions of $x$.*

### 14.2.2 Naive Bayes

We consider discrete data, that is $x \in \{1, \dots, m\}^d$ and $y \in \{1, \dots, k\}$, and the following generative model

$$\begin{array}{rcl} y & \sim & \text{multinomial}(\pi) \\ x|y=i & \sim & \displaystyle\prod_{j=1}^d \text{multinomial}(x_j|\theta_{ji}), \end{array}$$

where $\pi \in \mathbb{R}^k$, and each $\theta_{ji}$ is in the simplex in $\mathbb{R}^m$. In other words, given $y$, the $d$ components $x_1, \dots, x_d$ are independent.

Using the usual "one-hot" encoding of discrete distribution, we see each $x_j$ in $\mathbb{R}^m$ as one of the canonical basis vectors so that the probability of $x_j|y = i$ is equal to $\prod_{a=1}^m \theta_{jia}^{x_{ja}}$. We can then compute

$$\begin{aligned} \mathbb{P}(y = i|x) &\propto \mathbb{P}(y = i, x) = \prod_{i=1}^k \pi_i^{y_i} \prod_{i=1}^k \prod_{j=1}^d \prod_{a=1}^m \theta_{jia}^{x_{ja} y_i} \\ \log \mathbb{P}(y = i|x) &\propto \sum_{i=1}^n y_i \Big( \log \pi_i + \sum_{j=1}^d \sum_{a=1}^m (\log \theta_{jia}) x_{ja} \Big). \end{aligned}$$

Like for linear discriminant analysis, we thus also get a softmax model $\mathrm{softmax}\big[(w_i^\top x + b_i)_i\big]$, with $b_i = \log \pi_i$, and $w_i$ with components $\log \theta_{jia}$. Also, like for linear discriminant analysis, we can obtain maximum likelihood estimates for each parameter of multinomial variables using empirical proportions.

### 14.2.3 Maximum likelihood estimations

As shown above, for linear discriminant analysis and naive Bayes, we obtain conditional models corresponding to softmax regression, for which we can use optimization algorithms to get the relevant parameters (this is the discriminative approach followed in this book).

However, we can also use the generative models to estimate parameters in closed form. For example, for linear discriminant analysis, the maximum likelihood estimates for the class proportions are the empirical class proportions $\hat{\pi}_i$, the means are the empirical means, and $\widehat{\Sigma} = \sum_{i=1}^k \hat{\pi}_i \widehat{\Sigma}_i$, which allows us to compute $\hat{w}_i$ and $\hat{b}_i$, through the formula above, instead of having to solve a convex problem. The key question is: which one is better?

**Discriminative vs. generative learning.** When making an even simpler assumption of $\Sigma$ diagonal, we can study the potential benefits of the discriminative and the generative set-up, following Ng and Jordan (2001): the generative approach has a stronger bias but potentially a lower variance.

For both linear discriminant analysis (LDA[2]) in Section 14.2.1 and Naive Bayes in Section 14.2.2, if we use the conditional log-likelihood as a criterion, the discriminative approaches in the population case optimize the correct criterion directly, and thus must lead to better or equal performance. However, in the unregularized case, to approach the population case for logistic regression, we need a number of samples proportional to $d$ (e.g., by considering our bounds on Rademacher complexities in Section 4.5 with data with equal variance in all directions). For LDA or Naive Bayes, we need to estimate $d$ separate quantities simultaneously, and when using concentration inequalities and the

[2] Not to be mixed with Latent Dirichlet Allocation (Blei et al., 2003), which is a generative model for collections of text documents.

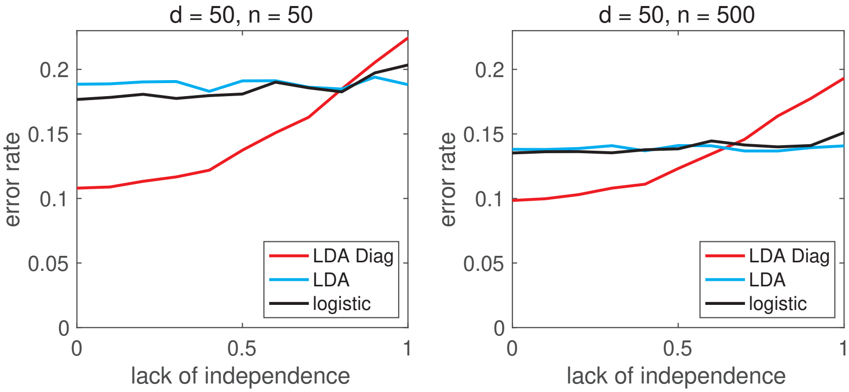


Figure 14.4: Comparison of LDA with full covariance matrix, LDA with diagonal covariance matrix, and logistic regression, on a well-specified binary classification problem (Gaussian class-conditional densities with same covariance matrix), with independent components and non-independent components (with a smooth transition, which is linear in the matrix logarithm). For independent components (left parts of the plots), linear discriminant analysis with the independence assumptions leads to better performance.

union bound, we should expect to have $n$ larger than a constant times $\log d$ to attain the population performance. We thus get a larger bias with generative approaches but significantly less variability. See the experiments in Figure 14.4, more details by Ng and Jordan (2001), and a similar approach to variable selection in regression (Fan and Lv, 2008).

## 14.3 Bayesian inference

For simplicity, in this section, we consider random observations $z \in \mathcal{Z}$ that could be the traditional pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in supervised learning, but we note that Bayesian inference applies much more generally. See more details by Robert (2007).

We assume that we have a set of probability distributions over $z$, with densities with respect to some base measure, which are parameterized by some vector $\theta \in \Theta$ (a subset of a vector space), and which we denote $p(z|\theta)$, and refer to as the *likelihood function*. We assume some *prior distribution* with density $q(\theta)$ with respect to the Lebesgue measure. In the Bayesian methodology, we assume that $\theta$ is sampled once from the prior distribution and that we obtain *i.i.d.* observations $z_1, \dots, z_n \in \mathcal{Z}$ sampled from $p(z|\theta)$.

By independence and identical distributions, the overall joint distribution of the data and $\theta$ is

$$p(z_1, \dots, z_n, \theta) = q(\theta) \prod_{i=1}^{n} p(z_i|\theta).$$

We can then obtain the *posterior distribution* of $\theta$ given the data $(z_1, \dots, z_n)$, which is proportional to $p(z_1, \dots, z_n, \theta)$, and with density:

$$p(\theta|z_1, \dots, z_n) = \frac{q(\theta) \prod_{i=1}^n p(z_i|\theta)}{\int_\Theta q(\eta) \prod_{i=1}^n p(z_i|\eta) d\eta}.$$

As already noted, the mode of the posterior distribution is the "maximum a posteriori" (MAP) estimate, which is rarely used within Bayesian inference (see some reasons in Section 14.1.4). Other estimates or estimation procedures are preferred, all using the posterior distribution as the main source. Thus, being able to characterize this posterior distribution is the computational tool (see below).

**Posterior mean.** A good summary of the posterior distribution is the posterior mean $\int_\Theta \theta p(\theta|z_1, \dots, z_n) d\theta$ and is traditionally associated with parameter estimation with the square loss. This was called the MMSE in Section 14.1.4.

**Bayesian model averaging.** Given the multiple models characterized by the posterior distribution, we can consider performing inference on unseen data through the mixture distribution

$$\int_\Theta p(z|\theta) p(\theta|z_1, \dots, z_n) d\theta.$$

Thus, overall, Bayesian inference naturally leads to parameter estimation procedures that can be studied both from a computational perspective (see Section 14.3.1) and a statistical perspective, as part of the "PAC-Bayes" framework described in Section 14.4. But it can also be used for model selection, as described in Section 14.3.2.

### 14.3.1 Computational handling of posterior distributions

This section gives only a brief account of algorithms to characterize posterior distributions. See many more details by Gelman et al. (1995); Robert (2007).

**Conjugate priors.** In rare instances, the posterior distribution has a simple form. Two classic examples are the Gaussian prior on the mean parameter of a Gaussian variable and the Dirichlet prior on the parameters of a multinomial distribution.

**Gaussian approximation (Laplace method).** When the number of observations gets large, then, the integral defining the normalizing factor of the posterior distribution can be written as:

$$\int_\Theta q(\eta) \prod_{i=1}^n p(z_i|\eta) d\eta = \int_\Theta \exp\Big[n \times \Big(\frac{1}{n} \log q(\eta) + \frac{1}{n} \sum_{i=1}^n \log p(z_i|\eta) d\eta\Big)\Big],$$

and thus as $\int_\Theta \exp(nh(\theta)) d\theta$ for a certain function $h$. The Laplace method is a traditional approximation technique for approximating integrals of that form when the function $h$

has a global maximum within the interior of $\Theta$.[3] This maximizer is exactly the MAP estimate $\hat{\theta}_{\mathrm{MAP}}$, and the approximation is exactly equivalent to modeling the posterior density as a Gaussian with mean $\hat{\theta}_{\mathrm{MAP}}$ and covariance matrix $\frac{1}{n}h''(\hat{\theta}_{\mathrm{MAP}})^{-1}$.

**Sampling.** Obtaining independent samples from the posterior distribution is often enough for inference purposes, and many algorithms exist, such as Markov chain Monte Carlo methods (Robert and Casella, 2005).

**Variational inference.** An alternative to sampling is to approximate the posterior distribution by a family of simple tractable distributions that are made to fit the posterior as closely as possible. See Blei et al. (2017) and references therein.

### 14.3.2 Model selection through marginal likelihood

Probabilistic models are often naturally defined hierarchically, with prior distributions that have themselves parameters (which we can call hyperparameters), themselves with their own prior distribution (often called hyperprior distribution). For example, using the above notations, the prior distribution is $q(\theta|\lambda)$ with a hyperprior $r(\lambda)$, with often a data distribution that depends on both $\theta$ and $\lambda$.

While we could still treat $\lambda$ as a random variable on which Bayesian inference is performed, it is common to perform maximum-likelihood estimation on $\lambda$, or more generally, maximum a posteriori estimation. This is sometimes called "type II maximum likelihood" or "empirical Bayes". This leads to a form of hyperparameter selection for $\lambda$. More precisely, we maximize

$$\begin{aligned} p(\lambda|z_1,\dots,z_n) & \propto & p(\lambda,z_1,\dots,z_n) = \int_\Theta p(\lambda,\theta,z_1,\dots,z_n)d\theta \\ & \propto & r(\lambda)\int_\Theta \prod_{i=1}^n p(z_i|\theta,\lambda)q(\theta|\lambda)d\theta. \end{aligned}$$

The quantity $\int_\Theta \prod_{i=1}^n p(z_i|\theta)q(\theta|\lambda)d\theta$ is referred to as the *marginal likelihood*, and its maximization is a generic tool for hyparameter selection, with many applications. We present briefly two of them below.

**Selection among finitely many models.** A classical application of marginal likelihood maximization is to consider $m$ different models, that is, $m$ different distribution $p_j(z|\theta_j)$, with potentially parameters $\theta_j \in \Theta_j$ living in different spaces, with prior distribution $q_j(\theta_j)$. With a uniform distribution on the models, model selection is performed by maximizing with respect to $j \in \{1,\dots,m\}$:

$$\int_{\Theta_j} \prod_{i=1}^n p_j(z_i|\theta_j)q_j(\theta_j)d\theta_j.$$

[3] See https://francisbach.com/laplace-method/ for details.

Let's consider the Gaussian approximation obtained from the Laplace approximation. One can show that we obtain penalized maximum log-likelihood with a penalty equal to $\frac{d_j}{2} \log n$, where $d_j$ is the dimension of $\Theta_j$, leading to the Bayesian information criterion (BIC) (see Robert, 2007, Chapter 7). See the discussion in Section 8.2.2.

**Sparsity with automatic relevance determination.** As mentioned in Section 14.1.3, we consider a prior distribution $q(\theta|\eta)$ which is Gaussian with mean zero and covariance matrix $\eta I$. Maximizing the penalized marginal likelihood ends up being similar to the "$\eta$-trick" from Section 8.3.1. Indeed, when we consider regression with Gaussian noise, that is, when $y$ given $\theta$ is normal with mean $\Phi\theta$ and covariance matrix $\sigma^2 I$, then $y$ given $\eta$ is Gaussian with mean $\Phi \operatorname{Diag}(\eta)\Phi^\top + \sigma^2 I$, and thus we can compute the log-likelihood in closed form, which leads to a natural (non-convex) cost function to estimate $\eta$.

**Gaussian processes.** The example above may be extended to kernel methods presented in Chapter 7. Indeed, it is possible to define a probabilistic model of random function from a set $\mathcal{X}$ to $\mathbb{R}$ such that the marginal distribution of $f(x_1), \dots, f(x_n)$ is Gaussian with mean zero and covariance matrix $K \in \mathbb{R}^{n \times n}$ where $K_{ij} = k(x_i, x_j)$, where $k$ is a positive definite kernel function. This allows us to combine Bayesian inference with non-parametric kernel learning. See more details by Rasmussen and Williams (2006).

## 14.4 PAC-Bayesian analysis

In this section, we briefly review a generic framework to obtain generalization guarantees for randomized or averaged predictors like those from Bayesian inference. For more details, see Alquier (2021) and the many references therein.

### 14.4.1 Set-up

We consider the classical supervised learning framework that we have been following throughout the book, namely, with $n$ pairs of i.i.d. observations $(x_i, y_i)$ from a distribution $p$ on $\mathcal{X} \times \mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$. We assume that we have a family of prediction functions $f_\theta : \mathcal{X} \to \mathbb{R}$, parameterized by $\theta \in \Theta$ (which is a subset of a vector space equipped with the Lebesgue measure).

We consider predictors that are not based on selecting a single $\theta \in \Theta$, but a probability distribution $\rho$ over $\theta$. Given that probability distribution, we can consider:

(a) a randomized predictor $f_\theta$, where $\theta$ is sampled from $\rho$. Then the generalization performance will be considered with this extra randomness (on top of the randomness of the training data),

(b) the posterior mean $x \mapsto \int_\Theta f_\theta(x) d\rho(\theta)$ which is a function from $\mathcal{X}$ to $\mathbb{R}$ and then only the randomness of the training data need to be considered. Note that in this situation, the final prediction function is not in the set of all $f_\theta$, $\theta \in \Theta$, and is often called an "aggregated predictor".

The generalization bounds that will be presented will be valid for *all* potential probability distributions $\rho$, including ones that depend on the data, which implies that we can then optimize the bounds over the distribution, leading to a candidate which is very close to the Bayesian posterior distribution (but with an added temperature). Like in Bayesian inference, we consider a fixed probability distribution $q$ on $\Theta$, which we will refer to as the prior.

We use the notation $\mathcal{R}(\theta) = \mathbb{E}\big[\ell(y, f_\theta(x)\big]$ for the expected risk (a deterministic function of $\theta$), and $\widehat{\mathcal{R}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f_\theta(x_i))$ for the empirical risk (which is a random function of with expectation $\mathcal{R}$).

### 14.4.2 Uniformly bounded loss functions

We assume that almost surely, for all $\theta \in \Theta$, we have: $\ell(y, f_\theta(x)) \in [0, \ell_\infty]$ (for example, with the 0-1 loss for binary classification, or with bounded predictors for regression). Following the exposition of Alquier (2021); Catoni (2003), in the proof of Hoeffding's inequality in Section 1.2.1, we saw that for all $\theta \in \Theta$ and $s \in \mathbb{R}_+$, we have:

$$\mathbb{E}\big[\exp\big(s(\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta))\big)\big] \leqslant \exp\Big(\frac{s^2\ell_\infty^2}{8n}\Big).$$

Integrating over $\theta$, we get

$$\int_\Theta \mathbb{E}\big[\exp\big(s(\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta))\big)\big]dq(\theta) \leqslant \exp\Big(\frac{s^2\ell_\infty^2}{8n}\Big).$$

We now use the variational formulation of the log-partition function (also known as the Donsker-Varadhan formula), with $h(\theta) = s(\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta))$.

$$\log \int_\Theta \exp(h(\theta))dq(\theta) = \sup_{\rho \in \mathcal{P}(\theta)} \int_\Theta h(\theta)d\rho(\theta) - D(\rho\|q),$$

with $\mathcal{P}(\theta)$ the set of probability distribution on $\Theta$ and $D(\rho\|q)$ the Kullback-Leibler divergence between $\rho$ and $q$, defined as (see also Section 15.1.3):

$$D(\rho\|q) = \int_\Theta \log\Big(\frac{d\rho}{dq}(\theta)\Big)d\rho(\theta).$$

This leads to

$$\mathbb{E}\Big[\exp\Big(\sup_{\rho \in \mathcal{P}(\theta)} \int_\Theta s(\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta))d\rho(\theta) - D(\rho\|q)\Big)\Big] \leqslant \exp\Big(\frac{s^2\ell_\infty^2}{8n}\Big). \tag{14.5}$$

Thus, using Chernoff bound,[4] we obtain that with probability greater than $1-\delta$,

$$\sup_{\rho \in \mathcal{P}(\theta)} \int_\Theta s(\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta))d\rho(\theta) - D(\rho\|q) \leqslant \frac{s^2\ell_\infty^2}{8n} + \log\frac{1}{\delta},$$

[4] See https://en.wikipedia.org/wiki/Chernoff_bound.

or, in other words, for all $\rho \in \mathcal{P}(\theta)$,

$$\int_\Theta \mathcal{R}(\theta)d\rho(\theta) \leqslant \int_\Theta \widehat{\mathcal{R}}(\theta)d\rho(\theta) + \frac{1}{s}D(\rho\|q) + \frac{1}{s}\log\frac{1}{\delta} + \frac{s\ell_\infty^2}{8n}.$$

We thus get a bound on the average generalization error based on the average empirical error. The bound can be empirically computed for any $\rho$ and minimized, with the optimal distribution being proportional to $\exp(-s\widehat{\mathcal{R}}(\theta))dq(\theta)$, which is often called the Gibbs posterior distribution. With $s = n$, this is exactly the Bayesian posterior distribution. Denoting $\hat{\rho}_s$ this distribution, we get with probability greater than $1-\delta$, that

$$\int_\Theta \mathcal{R}(\theta)d\hat{\rho}_s(\theta) \leqslant \inf_{\rho\in\mathcal{P}(\Theta)} \int_\Theta \widehat{\mathcal{R}}(\theta)d\rho(\theta) + \frac{1}{s}D(\rho\|q) + \frac{1}{s}\log\frac{1}{\delta} + \frac{s\ell_\infty^2}{8n}.$$

**Beyond integrated risks.** For convex loss functions, by Jensen's inequality, the risk of the posterior mean $x \mapsto \int_\Theta f_\theta(x)d\rho(\theta)$ is less than the integrated risk, so the bound applies.

Moreover, by applying Jensen's inequality to Eq. (14.5), we can get a bound in expectation as for all $\rho \in \mathcal{P}(\theta)$ (again $\rho$ may depend on the data):

$$\mathbb{E}\Big[\int_\Theta \mathcal{R}(\theta)d\rho(\theta)\Big] \leqslant \mathbb{E}\Big[\int_\Theta \widehat{\mathcal{R}}(\theta)d\rho(\theta) + \frac{1}{s}D(\rho\|q) + \frac{s\ell_\infty^2}{8n}\Big].$$

Moreover, for the Gibbs posterior, by applying Jensen's inequality, we get:

$$\mathbb{E}\Big[\int_\Theta \mathcal{R}(\theta)d\hat{\rho}_s(\theta)\Big] \leqslant \inf_{\rho\in\mathcal{P}(\Theta)} \int_\Theta \mathcal{R}(\theta)d\rho(\theta) + \frac{1}{s}D(\rho\|q) + \frac{s\ell_\infty^2}{8n}. \tag{14.6}$$

**Finite set of models.** We consider $m$ prediction functions $\hat{f}_1, \dots, \hat{f}_n$. By considering all Diracs in Eq. (14.6), we get that

$$\mathbb{E}\Big[\int_\Theta \mathcal{R}(\theta)d\hat{\rho}_s(\theta)\Big] \ \leqslant\ \inf_{\theta\in\Theta}\ \mathcal{R}(\theta) + \frac{1}{s}\log\frac{1}{q(\theta)} + \frac{s\ell_\infty^2}{8n}.$$

With $q(\theta) = 1/m$ and optimizing over $s$, we get the usual $\ell_\infty\sqrt{\frac{\log m}{n}}$ like we obtained for empirical risk minimization in Section 4.4.3.

**Lipschitz-continuous losses, linear predictions, and Gaussian priors.** See the tutorial from Alquier (2021) to recover rates similar to ones that can be obtained with Rademacher complexities in Chapter 4.

**Application to sparse regression.** PAC-Bayesian analysis can be considered in many settings, including the sparse linear regression problems as dealt with in Chapter 8. For example, Alquier and Lounici (2011); Rigollet and Tsybakov (2011) consider the combination of all least-squares predictors with supports restricted to a set $A \subset \{1, \dots, d\}$ for all such sets $A$. The combination is performed with exponential weights, and the estimator is shown to exhibit the same performance as the $\ell_0$-penalty from Section 8.2.2, but now requires sampling as an estimation algorithm instead of combinatorial optimization.

## 14.5 Conclusion

Probabilistic modeling is an important part of machine learning. In this chapter, we simply highlighted some topics related to learning theory, namely (1) the link between prior models and predictive performance, where we showed that maximum a-priori estimation may not correctly leverage the knowledge of the prior distribution, namely (2) the use of generative models to obtain alternatives to discriminative estimators, and (3) the link between Bayesian inference.

# Chapter 15

# Lower bounds on performance

**Chapter summary**

- Statistical lower bounds: for least-squares regression, the optimal performance of supervised learning with target functions that are linear in some feature vector or in Sobolev spaces on $\mathbb{R}^d$ happens to be achieved by several algorithms presented earlier in the book. The lower bounds can be obtained through information theory or Bayesian analysis.
- Optimization lower bounds: for the classical problem classes from Chapter 5, hard functions can be designed so that gradient-descent-based algorithms that linearly combine gradients are shown to be optimal.
- Lower bounds for stochastic gradient descent: The rates proportional to $O(1/\sqrt{n})$ for convex functions and $O(1/n\mu)$ for $\mu$-strongly convex problems are optimal.

In this textbook, we have shown various convergence rates for statistical procedures when the number of observations $n$ goes to infinity, and optimization methods, as the number of iterations $t$ goes to infinity. Most were non-asymptotic upper bounds on the error measures, with a precise dependence on the problem parameters (e.g., smoothness of the target function or the objective function).

In this chapter, we are looking at lower bounds on performance, that is, we aim to show that for a particular problem class and a specific class of algorithms, the error measures cannot go to zero too quickly. Lower bounds are useful, in particular when they match upper bounds up to constants (we can then claim that we have an "optimal" method). They sometimes provide hard problems (like for optimization), sometimes not (when they are based on information theory, such as for prediction performance).

> Lower bounds will be obtained in a "minimax" setting where we look at the worst-case performance over the entire problem class. As for upper bounds, looking at worst-case performance is, in essence, pessimistic, and algorithms often behave better than their bounds. The key is to identify classes of problems that are not too large (or the bounds will be very bad) but still contain interesting problems.

The chapter is naturally divided into three sections: Section 15.1 considers statistical lower bounds, Section 15.2 considers optimization lower bounds, while Section 15.3 considers lower bounds for stochastic gradient methods. All of these provide bounds related to the setups encountered in earlier chapters.

## 15.1 Statistical lower bounds

In this section, our goal is to obtain lower bounds for regression problems in $\mathbb{R}^d$ with the square loss when assuming the target function $f^* : \mathbb{R}^d \to \mathbb{R}$ (here the conditional expectation of $y$ given $x$) is in a particular set, such as:

- linear function of some $d$-dimensional features, that is, $f_*(x) = \langle \theta_*, \varphi(x) \rangle$, for $\theta_* \in \mathbb{R}^d$, potentially in a $\ell_2$-ball, and/or with less than $k$ non-zero elements,
- functions with all partial derivatives up to order $s$ bounded in $L_2$-norm (e.g., Sobolev spaces).

Since we are looking for lower bounds, we are free to make extra assumptions (that can only make the problem simpler) and lower the lower bounds. For example, we will focus on Gaussian noise with constant variance $\sigma^2$ that is independent of $x$.

We can either consider fixed design assumptions or random designs with the simplest input distributions (that can only make the problem simpler).

**Classification.** Lower bounds for classification problems are more delicate and out of scope (see, e.g., Yang, 1999). However, we can get lower bounds for the convex surrogates that are typically used (but note that this does not translate to lower bounds for the 0-1 loss), see for example Section 15.3 for Lipschitz-continuous loss functions.

### 15.1.1 Minimax lower bounds

We consider a set of probability distributions indexed by some set $\Theta$ (that can characterize input distributions and the smoothness of the target function). We consider some data $\mathcal{D}$, generated from this distribution, and we denote $\mathbb{E}_\theta$ expectations with respect to data coming from the distribution indexed by $\theta$.

We consider an estimator $\mathcal{A}(\mathcal{D})$ of $\theta \in \Theta$, with some squared distance $d^2$ between two elements of $\Theta$, so that $d(\theta, \theta')^2$ measures the performance of $\theta'$ when the true estimator is $\theta$. The performance of $\mathcal{A}$ when the data come from $\theta_*$ is

$$\mathbb{E}_{\theta_*}\big[d(\theta_*, \mathcal{A}(\mathcal{D}))^2\big].$$

The goal is to find an algorithm so that $\sup\limits_{\theta_*\in\Theta} \mathbb{E}_{\theta_*}\big[d(\theta_*, \mathcal{A}(\mathcal{D}))^2\big]$ is as small as possible, and the lower bound of performance is thus:

$$\inf_{\mathcal{A}} \sup_{\theta_*\in\Theta} \mathbb{E}_{\theta_*}\big[d(\theta_*, \mathcal{A}(\mathcal{D}))^2\big]. \tag{15.1}$$

This is often referred to as "minimax" lower bounds.

Since by Markov's inequality, $\mathbb{E}_{\theta_*}\big[d(\theta_*, \mathcal{A}(\mathcal{D}))^2\big] \geqslant A\,\mathbb{P}_{\theta_*}\big(d(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A\big)$, it is sufficient to lower bound

$$\inf_{\mathcal{A}} \sup_{\theta_*\in\Theta} \mathbb{P}_{\theta_*}\big(d(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A\big), \tag{15.2}$$

for some arbitrary $A > 0$. This will be useful for techniques based on information theory.

We will see two principles for obtaining statistical minimax lower bounds:

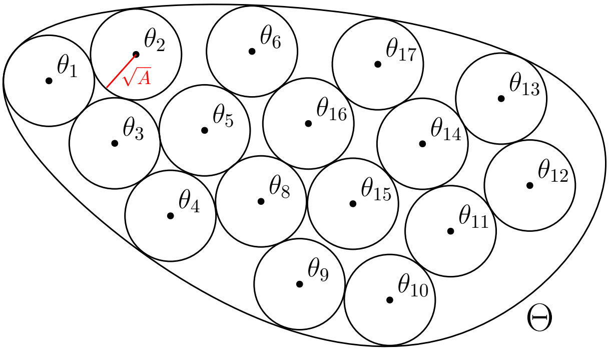
- **Reduction to a hypothesis test:** by selecting a finite subset $\{\theta_1, \dots, \theta_M\}$ of distributions $\Theta$ which is maximally spread, a good estimator leads to a good hypothesis test that can identify which $\theta_j$ (among the $M$ possibilities) was used to generate the data. We can then use information theory to lower-bound the probability of error of such a test. This versatile technique can handle most situations, from fixed to random design.
- **Bayesian analysis:** We can lower bound the supremum for all $\Theta$ by any expectation over a distribution supported on $\Theta$. Once we have an expectation, we can use the same decision-theoretic argument as the ones we used to compute the Bayes risk is Chapter 4, e.g., for Hilbertian or Euclidean performance measures, the optimal estimator is the conditional expectation $\mathbb{E}[\theta_*|\mathcal{D}]$. The key is choosing distributions so they can be computed in closed form. This approach is less flexible but the simplest in situations where it can be applied (fixed design regression on balls, with potentially sparse assumptions).

### 15.1.2 Reduction to a hypothesis test

The principle is simple: pack the set $\Theta$ with "balls" of some radius $4A$, that is, find $\theta_1, \dots, \theta_M \in \Theta$ such that

$$\forall i \neq j,\ d(\theta_i, \theta_j)^2 \geqslant 4A, \tag{15.3}$$

and transform the estimation problem into a hypothesis test, that is, an algorithm going from the data $\mathcal{D}$ to one out of $M$ potential outcomes (see illustration below).



Then, because we take the supremum over a smaller set:

$$\sup_{\theta_*\in\Theta} \mathbb{P}_{\theta_*}\big(d(\theta_*,\mathcal{A}(\mathcal{D}))^2 > A\big) \geqslant \max_{j\in\{1,\dots,M\}} \mathbb{P}_{\theta_j}\big(d(\theta_j,\mathcal{A}(\mathcal{D}))^2 > A\big). \tag{15.4}$$

Any algorithm $A(\mathcal{D}) \in \Theta$ gives a "test", that is, a function $g\circ\mathcal{A} : \mathcal{D} \to \{1,\dots,m\}$ defined as

$$g(\mathcal{A}(\mathcal{D})) = \arg\min_{j\in\{1,\dots,m\}} d(\theta_j,\mathcal{A}(\mathcal{D})) \in \{1,\dots,m\},$$

where ties are broken arbitrarily (e.g., by selecting the minimal index). Because of the packing condition in Eq. (15.3), the performance of $\mathcal{A}$ can be lower-bounded by the classification performance of $g\circ\mathcal{A}$ (with the 0-1 loss).

Indeed, if, for some $j \in \{1,\dots,M\}$, $g(\mathcal{A}(\mathcal{D})) \neq j$, there exists $k \neq j$, such that $d(\theta_k,\mathcal{A}(\mathcal{D})) < d(\theta_j,\mathcal{A}(\mathcal{D}))$. Moreover, using the triangle inequality for $d$, we get:

$$d(\theta_j,\theta_k)^2 \leqslant 2\big[d(\theta_j,\mathcal{A}(\mathcal{D}))^2 + d(\theta_k,\mathcal{A}(\mathcal{D}))^2\big],$$

then,

$$\begin{aligned} d(\theta_j,\mathcal{A}(\mathcal{D}))^2 &\geqslant \frac{1}{2}d(\theta_j,\theta_k)^2 - d(\theta_k,\mathcal{A}(\mathcal{D}))^2 \\ &> \frac{1}{2}d(\theta_j,\theta_k)^2 - d(\theta_j,\mathcal{A}(\mathcal{D}))^2 \text{ by the choice of } k, \end{aligned}$$

which implies $d(\theta_j,\mathcal{A}(\mathcal{D}))^2 > \frac{1}{4}d(\theta_j,\theta_k)^2 \geqslant A$. Thus, we have the following inequality for the probabilities of these two events:

$$\mathbb{P}_{\theta_j}\big(d(\theta_j,\mathcal{A}(\mathcal{D}))^2 > A\big) \geqslant \mathbb{P}_{\theta_j}\big(g(\mathcal{A}(\mathcal{D})) \neq j\big),$$

leading to, using Eq. (15.2) and Eq. (15.4),

$$\begin{aligned} \inf_{\mathcal{A}} \sup_{\theta_*\in\Theta} \mathbb{E}_{\theta_*}\big[d(\theta_*,\mathcal{A}(\mathcal{D}))^2\big] &\geqslant A\cdot\inf_h \max_{j\in\{1,\dots,M\}} \mathbb{P}_{\theta_j}\big(g(\mathcal{D}) \neq j\big) \\ &\geqslant A\cdot\inf_h \frac{1}{M}\sum_{j=1}^M \mathbb{P}_{\theta_j}\big(g(\mathcal{D}) \neq j\big), \end{aligned} \tag{15.5}$$

where $h$ is any function from $\mathcal{D}$ to $\{1, \dots, M\}$. We have thus lower-bounded the minimax statistical performance by the minimax performance of a hypothesis test $h : \mathcal{D} \to \{1, \dots, M\}$. Information theory can then be used to lower-bound this minimax error. We first provide a quick review of information theory (see Cover and Thomas, 1999, for more details).

### 15.1.3 Review of information theory

**Entropy.** Given a random variable $y$ taking finitely many values in $\mathcal{Y}$, its entropy is equal to

$$H(y) = -\sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \log \mathbb{P}(y = y').$$

Since $\mathbb{P}(y = y') \in [0, 1]$, the entropy is always non-negative. Moreover, using Jensen's inequality for the logarithm, we have $H(y) = \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \log \frac{1}{\mathbb{P}(y=y')} \leqslant \log \big( \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \frac{1}{\mathbb{P}(y=y')} \big) = \log |\mathcal{Y}|$.

The entropy $H(y)$ represents the uncertainty associated with the random variable $y$, going from $H(y) = 0$ if $y$ is deterministic (that is $\mathbb{P}(y = y') = 1$ for some $y' \in \mathcal{Y}$), to $\log |\mathcal{Y}|$ when $y$ has a uniform distribution.

**Joint and conditional entropies.** Given two random variables $x, y$ with finitely many values in $\mathcal{X}$ and $\mathcal{Y}$, we can define the joint entropy

$$H(x, y) = -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x', y = y') \log \mathbb{P}(x = x', y = y').$$

It can be decomposed as

$$\begin{aligned} H(x, y) &= -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \big[ \mathbb{P}(y = y' | x = x') \mathbb{P}(x = x') \big] \\ &= -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \mathbb{P}(y = y' | x = x') \\ &\qquad\qquad - \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \mathbb{P}(x = x') \\ &= \sum_{x' \in \mathcal{X}} \mathbb{P}(x = x') \log H(y | x = x') + H(x), \end{aligned}$$

where $H(y|x = x')$ is the entropy of the conditional distribution of $y$ given $x = x'$. By defining the conditional entropy $H(y|x)$ as $H(y|x) = \sum_{x' \in \mathcal{X}} \mathbb{P}(x = x') H(y|x = x')$, we exactly have:

$$H(x, y) = H(y|x) + H(x).$$

This leads to a first version of Fano's inequality, which lower bounds the probability that $y \neq \hat{y}$ from the conditional entropy $H(y|\hat{y})$; the main idea is that if $y$ remains very uncertain given $\hat{y}$, then the probability that they are equal cannot be too large.

**Proposition 15.1 (Fano's inequality)** *If the random variables $y$ and $\hat{y}$ have values in the same finite set $\mathcal{Y}$, then*

$$\mathbb{P}(\hat{y} \neq y) \geqslant \frac{H(y|\hat{y}) - \log 2}{\log |\mathcal{Y}|}.$$

**Proof** Let $e = 1_{y \neq \hat{y}} \in \{0, 1\}$ be the indicator function of errors; by decomposing the joint entropy through conditional and marginal entropies in the two different ways, we get:

$$H(e|\hat{y}) + H(y|e, \hat{y}) = H(e, y|\hat{y}) = H(y|\hat{y}) + H(e|y, \hat{y}).$$

We then have $H(e|y, \hat{y}) = 0$ (since $e$ is deterministic given $y$ and $\hat{y}$), $H(e|\hat{y}) \leqslant H(e) \leqslant \log 2$ (because $e \in \{0, 1\}$), and $H(y|e, \hat{y}) = \mathbb{P}(e = 1)H(y|\hat{y}, e = 1) + \mathbb{P}(e = 0)H(y|\hat{y}, e = 0) = \mathbb{P}(e = 1)H(y|\hat{y}, e = 1) + 0 \leqslant \mathbb{P}(\hat{y} \neq y) \log |\mathcal{Y}|$. Expressing $\mathbb{P}(\hat{y} \neq y)$ in function of the other quantities leads to the desired result. ■

**Data processing inequality.** A fundamental result in information theory allows to lower-bound conditional entropies where conditional independencies are present. That is, if we have three random variables $x, y, z$, such that $z$ and $x$ are conditionally independent given $y$, then $H(x|z) \geqslant H(x|y)$: in words, the uncertainty of $x$ given $z$ has to be larger than the uncertainty of $x|y$, which is "normal" because the statistical dependence between $x$ and $z$ is occurring through $y$. In other words, the sequence $x \to y \to x$ forms a Markov chain.

The data processing inequality is a simple application of the concavity of the entropy as a function of the probability mass function; indeed, using that by conditional independence $\mathbb{P}(x = x'|z = z') = \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x', y = y'|z = z') = \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x'|y = y')\mathbb{P}(y = y'|z = z')$, and Jensen's inequality for the function $t \mapsto -t \log t$, we have:

$$\begin{aligned} H(x|z) &= \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z')H(x|z = z') \\ &\geqslant \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z') \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y'|z = z')H(x|y = y') \\ &= \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y')H(x|y = y') = H(x|y). \end{aligned}$$

This leads immediately to the following full version of Fano's inequality:

**Proposition 15.2 (Fano's inequality)** *If the random variable $y$ and $\hat{y}$ have values in the same finite set $\mathcal{Y}$, and if we have a Markov chain $y \to z \to \hat{y}$, then*

$$\mathbb{P}(\hat{y} \neq y) \geqslant \frac{H(y|\hat{y}) - \log 2}{\log |\mathcal{Y}|} \geqslant \frac{H(y|z) - \log 2}{\log |\mathcal{Y}|}.$$

We need a last concept from information theory, namely mutual information and Kullback-Leibler divergence, both for discrete and continuous-valued random variables.

**Mutual information.** Given two random variables $x$ and $y$, then we can define their mutual information as

$$I(x,y) = H(x) - H(x|y) = H(x) + H(y) - H(x,y) = H(y) - H(y|x).$$

This can be seen as the uncertainty reduction in $x$ when observing $y$. It is symmetric, always less than $\log|\mathcal{X}|$ and $\log|\mathcal{Y}|$. Moreover, it can be written as:

$$\begin{aligned} I(x,y) &= H(x) + H(y) - H(x,y) \\ &= \sum_{x'\in\mathcal{X}}\sum_{y'\in\mathcal{Y}} \mathbb{P}(x=x',y=y') \log \frac{\mathbb{P}(x=x',y=y')}{\mathbb{P}(x=x')\mathbb{P}(y=y')}, \end{aligned}$$

which can be seen as the Kullback-Leibler (KL) divergence between the distribution of $(x,y)$ and the product of marginals of $x$ and $y$. Indeed, given two distributions on $\mathcal{Z}$, $p$ and $q$ (which are non-negative functions on $\mathcal{Z}$ that sum to one), then the KL divergence is defined as

$$D_{\mathrm{KL}}(p||q) = \sum_{z\in\mathcal{Z}} p(z) \log \frac{p(z)}{q(z)}.$$

The KL divergence is always non-negative by convexity of the function $t \mapsto t\log t$, and equal to zero, if and only if $p = q$. Moreover, the KL divergence is jointly convex in $(p,q)$. Thus, one can see the mutual information between the KL divergences between the joint distribution of $(x,y)$ and the corresponding product of marginals (which is thus non-negative).

**From discrete to continuous distributions.** Many of the information theory concepts can be extended to continuous random variables on $\mathbb{R}^d$ by replacing the probability mass function with the probability density with respect to some base measures. Then, many properties (which were obtained through convex arguments) extend. In particular, the data processing inequality and Fano's inequality when $z$ is continuous-valued (see more details by Cover and Thomas, 1999).

Moreover, the KL divergence between two distributions can be defined as

$$D_{\mathrm{KL}}(p||q) = \mathbb{E}_p\Big[\log\frac{dp}{dq}(x)\Big].$$

A short calculation shows that for two normal distributions of means $\mu_1,\mu_2$ and equal covariance matrices $\Sigma$, the KL divergence is equal to $\frac{1}{2}(\mu_1-\mu_2)^\top\Sigma^{-1}(\mu_1-\mu_2)$.

### 15.1.4 Lower-bound on hypothesis testing based on information theory

We consider a joint random variable $(y,\mathcal{D})$ distributed as $y$ uniform in $\{1,\dots,M\}$, and, given $y=j$, $\mathcal{D}$ distributed as the distribution associated with $\theta_j$. We consider $\hat{y} = h(\mathcal{D})$. This defines a Markov chain: $y \to \mathcal{D} \to h(\mathcal{D})$, that is, even for a randomized test $h$ (with extra randomization), $h(\mathcal{D})$ is independent of $y$ given $\mathcal{D}$. By construction, the last term in Eq. (15.5) is exactly the probability that $\hat{y} \neq y$. This is exactly what Fano's inequality from information theory gives us, leading to the following corollary.

**Corollary 15.1 (Fano's inequality for multiple hypothesis testing)** *Given $M$ probability distributions $p_j$ on $\mathcal{D}$, then*

$$\inf_h \frac{1}{M} \sum_{j=1}^{M} \mathbb{P}_{p_j}\big(h(\mathcal{D}) \neq j\big) \geqslant 1 - \frac{1}{M^2 \log M} \sum_{j,j'=1}^{M} D_{\mathrm{KL}}(p_j || p_{j'}) - \frac{\log 2}{\log M}. \tag{15.6}$$

**Proof** We consider a joint random variable $(y, \mathcal{D})$ distributed as $y$ uniform in $\{1, \dots, M\}$, and, given $y = j$, $\mathcal{D}$ distributed as the distribution $p_j$. We have:

$$\begin{aligned} H(y|z) &= H(y) - I(y,z) = \log M - \frac{1}{M} \sum_{j=1}^{M} D_{\mathrm{KL}}\Big(p_j \Big\| \frac{1}{M} \sum_{j'=1}^{M} p_{j'}\Big) \\ &\geqslant \log M - \frac{1}{M^2} \sum_{j,j'=1}^{M} D_{\mathrm{KL}}(p_j \| p_{j'}), \end{aligned}$$

by the convexity of the Kullback-Leibler divergence. We can then apply Prop. 15.2 and conclude. ■

**Using Gaussian noise to compute KL divergences.** For regression with Gaussian errors such as $y_i = f_\theta(x_i) + \varepsilon_i$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, then, for fixed designs (all $x_i$'s deterministic), we get exactly

$$D_{\mathrm{KL}}(p_{\theta_j} || p_{\theta_{j'}}) = \frac{1}{2\sigma^2} \sum_{i=1}^{n} \big[f_{\theta_j}(x_i) - f_{\theta_{j'}}(x_i)\big]^2 = \frac{n}{2\sigma^2} d(\theta_j, \theta_{j'})^2,$$

where $d(\theta, \theta')^2 = \frac{1}{n} \sum_{i=1}^{n} \big[f_\theta(x_i) - f_{\theta'}(x_i)\big]^2$ is the empirical mean squared difference between two models.

For random designs, we consider distributions on $(x_i, y_i)_{i=1,\dots,n}$. If we consider a common distribution $p$ for $x$, then

$$D_{\mathrm{KL}}(p_{\theta_j} || p_{\theta_{j'}}) = \frac{1}{2\sigma^2} \int_{\mathcal{X}} \big[f_{\theta_j}(x) - f_{\theta_{j'}}(x)\big]^2 dp(x) = \frac{1}{2\sigma^2} \| f_{\theta_j} - f_{\theta_{j'}} \|^2_{L_2(p)},$$

which we define to be $\frac{1}{2\sigma^2} d(\theta_j, \theta_{j'})^2$.

Overall, to obtain a lower bound with Gaussian noise, we need to find $\theta_1, \dots, \theta_M$ in $\Theta$ such that:

- $\displaystyle \frac{1}{M^2} \sum_{j,j'=1}^{M} \frac{n}{2\sigma^2} d(\theta_j, \theta_{j'})^2 \leqslant \log(M)/4.$ and $\log 2 / \log M \leqslant 1/4$ (that is $M \geqslant 16$), so that Eq. (15.6) leads to a lower bound of $A/2$.

- $\min_{j\neq k} d(\theta_j,\theta_k)^2 \geqslant 4A$, so that we can apply Eq. (15.5).

Then, the minimax lower bound is $A/2$. Thus, the lower bound is essentially the largest possible $A$ for a given $M$ such that we can find $M$ points in $\Theta$, which are all $2\sqrt{A}$ apart. There are two main tools to find such packings: (1) a direct volume argument and (2) using Varshamov-Gilbert's lemma. We present them before going over examples.

**Volume argument.** The following lemma provides the simplest argument.

**Lemma 15.1 (Packing $\ell_2$-balls)** *Let $M$ be the maximal number of elements of the Euclidean ball of radius 1, which are at least $2\varepsilon$-apart in $\ell_2$-norm. Then $(2\varepsilon)^{-d} \leqslant M \leqslant (1+\varepsilon^{-1})^d$.*

**Proof** Let $\theta_1,\dots,\theta_M$ be the corresponding $M$ points.

(a) All balls of center $\theta_j$ and radius $\varepsilon$ are disjoint and included in the ball of radius $1+\varepsilon$. Thus, the sum of the volumes of the small balls is smaller than the volume of the large balls, that is, $M\varepsilon^d \leqslant (1+\varepsilon)^d$.

(b) Since $M$ is maximal, for any $\theta$ such that $\|\theta\|_2 \leqslant 1$, there exists a $j \in \{1,\dots,M\}$ such that $\|\theta_j-\theta\|_2 \leqslant 2\varepsilon$ (otherwise, we can add a new point to $\{\theta_1,\dots,\theta_M\}$ and $M$ is not maximal). Thus, the ball of radius 1 is covered by the $M$ balls of radius $\theta_j$ and radius $2\varepsilon$. Thus, by using volumes, we get $1 \leqslant M(2\varepsilon)^d$. ■

**Packing with Varshamov-Gilbert lemma.** The maximal number of points in the hypercube $\{0,1\}^d$ that are at least $d/4$-apart in Hamming loss (i.e., $\ell_1$-distance) is greater than than $\exp(d/8)$, with a nice probabilistic argument.

**Lemma 15.2 (Varshamov-Gilbert's lemma)** *For any $\alpha \in (0,1)$, there exists a subset $\mathcal{B}$ of the hypercube $\{0,1\}^d$ such that*

*(a) for all $x,x' \in \mathcal{B}$ such that $x \neq x'$, $\|x-x'\|_1 \geqslant (1-\alpha)\frac{d}{2}$,*

*(b) $|\mathcal{B}| \geqslant \exp(d\alpha^2/2)$.*

**Proof** We consider the largest family satisfying (a). By maximality, the union of $\ell_1$-balls of radius $(1-\alpha)\frac{d}{2}$ includes all of $\{0,1\}^d$ . Therefore, by comparing cardinalities,

$$2^d \leqslant \sum_{x\in\mathcal{B}} \big|\big\{y \in \{0,1\}^d, \|y-x\|_1 \leqslant (1-\alpha)\frac{d}{2}\big\}\big|.$$

Consider a random variable $z$, which is binomial with parameters $d$ and $1/2$ (that is, the sum of $d$ independent uniform Bernoulli random variables). Then,

$$2^{-d}\big|\big\{y \in \{0,1\}^d, \|y-x\|_2^2 = \|y-x\|_1 \leqslant (1-\alpha)\frac{d}{2}\big\}\big| = \mathbb{P}\big(z \leqslant (1-\alpha)\frac{d}{2}\big) = \mathbb{P}\big(z \geqslant (1+\alpha)\frac{d}{2}\big).$$

Using Hoeffding's inequality (Prop. 1.1), we get $\mathbb{P}(z \geqslant (1+\alpha)\frac{d}{2}) = \mathbb{P}(\frac{z}{d} - \frac{\mathbb{E}[z]}{d} \geqslant \alpha\frac{d}{2}) \leqslant \exp(-2d(\alpha/2)^2) = \exp(-d\alpha^2/2)$. This leads to the result. ■

### 15.1.5 Examples

**Fixed design linear regression.** We consider linear regression with $\Phi \in \mathbb{R}^{n\times d}$ a design matrix with $\frac{1}{n}\Phi^\top \Phi = I$ (which imposes $n \geqslant d$). We consider the ball $\Theta = \{\theta \in \mathbb{R}^d, \|\theta\|_2 \leqslant D\}$. By rotational invariance of the Gaussian distribution of the noise variable $\varepsilon$, we can assume that the first $d$ rows of $\Phi$ are equal to $\sqrt{n}I$ and the rest of the rows are equal to zero. Thus we can assume the model $y = \theta_* + \frac{1}{\sqrt{n}}\varepsilon$, where $\varepsilon \in \mathbb{R}^d$ with normal distribution with mean zero and covariance $\sigma^2 I$, and $y \in \mathbb{R}^d$. We are thus in the situation where $d(\theta,\theta')^2 = \|\theta - \theta'\|_2^2$.

In order to find $M$ points in $\Theta = \{\theta \in \mathbb{R}^d,\ \|\theta\|_2 \leqslant D\}$, we consider the $M \geqslant \exp(d/8)$ elements $x_1,\dots,x_M$ of $\{0,1\}^d$ from Lemma 15.2 with $\alpha = 1/2$, and define $\theta_i = \beta(2x_i - 1_d) \in \{-\beta,\beta\}$. Thus $\|\theta_i\|_2^2 = \beta^2 d$, and, for $i \neq j$,

$$\|\theta_i - \theta_j\|_2^2 \leqslant 4\beta^2 d \leqslant 32\beta^2\log(M) \text{ and } \|\theta_i - \theta_j\|_2^2 \geqslant \beta^2 d.$$

We thus need, $\beta^2 d \leqslant D^2$, and $32\beta^2\log(M)\frac{n}{2\sigma^2} \leqslant \frac{\log M}{4}$, that is, $64\beta^2\frac{n}{\sigma^2} \leqslant 1$. Thus, the optimal rate is greater than

$$\frac{1}{8}\beta^2 d \geqslant \frac{1}{8}\min\Big\{D^2, \frac{\sigma^2 d}{64n}\Big\}.$$

Therefore, when $D^2 \geqslant \frac{\sigma^2 d}{64n}$, we get a lower bound of $\dfrac{\sigma^2 d}{512n}$, which is the upper-bound obtained in Chapter 3 (note that in Section 3.7 we provided a sharper lower-bound using similar tools as in Section 15.1.6).

The sparse regression setting could also be considered with the same tool, but the proof is simpler with the Bayesian arguments from Section 15.1.6. We now turn to the random design setting.

**Exercise 15.1** *Use Lemma 15.1 instead of Lemma 15.2 to obtain the same result.*

**Random design linear regression.** We consider the same model as above, but with $(x_i,y_i)$ sampled i.i.d. from a given distribution such that $\mathbb{E}[\varphi(x)\varphi(x)^\top] = I$, so that $d(\theta,\theta')^2 = \|\theta-\theta'\|_2^2$. Thus, the result above for fixed design regression also applies to the random design setting.

**Non-parametric estimation with Hilbert spaces (♦).** We consider random design regression with a fixed distribution for the inputs, with Gaussian independent noise and target functions in a certain ellipsoid of $L_2(p)$. That is, we assume that there exists a compact self-adjoint operator $T$ on $L_2(p)$ such that $\langle\theta, T^{-1}\theta\rangle_{L_2(p)} \leqslant D^2$. We denote by $(\lambda_m)_{m\geqslant 1}$ the non-increasing sequence of eigenvalues of $T$, with the associated eigenvectors $\psi_m$ in $L_2(p)$.

We consider a certain integer $K$, and $M \geqslant \exp(K/8)$ elements $x_1,\dots,x_M$ of $\{0,1\}^K$. We define $\theta_i = \beta\sum_{m=1}^K (2(x_i)_m - 1)\psi_m$. Then $\langle\theta, T^{-1}\theta\rangle_{L_2(p)} = \beta^2\sum_{m=1}^K \lambda_m^{-1} \leqslant K\beta^2\lambda_K^{-1}$, and, for $i \neq j$,

$$\|\theta_i - \theta_j\|_{L_2(p)}^2 \leqslant 4\beta^2 K \leqslant 32\beta^2 \log(M) \text{ and } \|\theta_i - \theta_j\|_{L_2(p)}^2 \geqslant \beta^2 K.$$

We thus need, $\beta^2 K \leqslant D^2 \lambda_K$, and $32\beta^2 \log(M) \frac{n}{2\sigma^2} \leqslant \frac{\log M}{4}$, that is, $64\beta^2 \frac{n}{\sigma^2} \leqslant 1$. Thus, the minimax lower bound is greater than

$$\frac{1}{8}\beta^2 K \geqslant \frac{1}{8} \min \Big\{ D^2 \lambda_K, \frac{\sigma^2 K}{64n} \Big\}.$$

We can now specialize to Sobolev spaces where it can be shown that for compact supports with piecewise smooth boundaries. The sum of all $L_2$-norms of partial derivatives corresponds to an operator for which $\lambda_K \geqslant C \cdot K^{-\alpha}$, with $\alpha = 2s/d$ when all $s$-th order derivatives are taken, for a constant $C$ (Adams and Fournier, 2003). The lower bound becomes

$$\max_{K \geqslant 1} \frac{1}{8} \min \Big\{ D^2 C K^{-\alpha}, \frac{\sigma^2 K}{64n} \Big\},$$

which can be balanced to obtain $K \propto \big(\frac{nD^2}{\sigma^2}\big)^{1/(1+\alpha)}$, leading to lower bound proportional to

$$D^{2/(1+\alpha)} \Big(\frac{\sigma^2}{n}\Big)^{\alpha/(1+\alpha)}.$$

For $\alpha = 2s/d$, we get $\alpha/(1+\alpha) = \frac{2s}{2s+d}$, and the lower matches the upper-bound obtained with kernel ridge regression in Chapter 7. It turns out that the lower bound on the minimax rate for Lipschitz-continuous functions is the same as for $s = 1$ (Tsybakov, 2008, Section 2.6).

### 15.1.6 Minimax lower bounds through Bayesian analysis

We can use a Bayesian analysis as outlined for least-squares in Section 3.7. We consider a particular probability distribution $p(\theta_*)$ whose support is included in $\Theta$. Then we have:

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*} \big[ d(\theta_*, \mathcal{A}(\mathcal{D}))^2 \big] \geqslant \inf_{\mathcal{A}} \mathbb{E}_{p(\theta_*)} \mathbb{E}_{\theta_*} \big[ d(\theta_*, \mathcal{A}(\mathcal{D}))^2 \big].$$

This reasoning is particularly simple when the optimal algorithm $\mathcal{A}$ is simple to estimate, which is the case in particular where $d$ is a Euclidean norm so that $\mathcal{A}^*(\mathcal{D}) = \mathbb{E}\big[\theta_* | \mathcal{D}\big]$. If the prior $p(\theta_*)$ and the likelihood $p(\mathcal{D}|\theta_*)$ are simple enough, then the conditional expectation can be computed in closed form. In Section 3.7, these were all Gaussians, which was possible for the prior distribution on $\Theta$ because $\Theta$ was unbounded. When dealing with bounded balls, we need to use different distributions, as used originally by Donoho and Johnstone (1994).

**Least-squares on a Euclidean ball.** We consider linear regression with a fixed design like in the previous section (with a bound $\|\theta_*\|_2 \leqslant D$), which corresponds to the model $y = \theta_* + \frac{1}{\sqrt{n}}\varepsilon$, where $\varepsilon \in \mathbb{R}^d$ with normal distribution with mean zero and covariance $\sigma^2 I$, and $y, \theta_* \in \mathbb{R}^d$.

We then consider a prior distribution on $\theta_*$ as $\theta_* = \beta x$, where $x \in \{-1,1\}^d$ are independent Rademacher random variables. We need $\beta^2 d \leqslant D^2$ to be in the correct set. We then need to compute $\mathbb{E}[\theta_*|y]$. The posterior probability of $\theta_*$ is supported on $\beta\{-1,1\}^n$. Moreover, given the independence by component, we can treat each separately. Then, by keeping only terms that depend on the posterior value, we get:

$$\mathbb{P}((\theta_*)_i = \pm\beta|y_i) \propto \exp(-\frac{n}{2\sigma^2}(y_i - \pm\beta)^2) \propto \exp(\pm\frac{n}{\sigma^2}y_i\beta).$$

Thus,

$$\mathbb{E}\big[(\theta_*)_i|y_i\big] = \beta\frac{\exp(\frac{n}{\sigma^2}y_i\beta) - \exp(\frac{-n}{\sigma^2}y_i\beta)}{\exp(\frac{n}{\sigma^2}y_i\beta) + \exp(\frac{-n}{\sigma^2}y_i\beta)} = \beta\frac{1-\exp(\frac{-2n}{\sigma^2}y_i\beta)}{1+\exp(\frac{-2n}{\sigma^2}y_i\beta)} = \beta\big[2\text{sigmoid}\big(\frac{2n}{\sigma^2}y_i\beta\big) - 1\big],$$

where $\text{sigmoid}(\alpha) = 1/(1+\exp(-\alpha))$.

The posterior variance for the $i$-th component is equal to

$$\begin{aligned}
\mathbb{E}\big[\big((\theta_*)_i - \mathbb{E}\big[(\theta_*)_i|y_i\big]\big)^2\big] &= \frac{1}{2}\mathbb{E}_{\varepsilon_i}\big(\beta - \beta\big[2\text{sigmoid}\big(2\frac{n}{\sigma^2}\beta(\beta + \varepsilon_i/\sqrt{n})\big) - 1\big]\big)^2 \\
&\quad + \frac{1}{2}\mathbb{E}_{\varepsilon_i}\big(-\beta - \beta\big[2\text{sigmoid}\big(2\frac{n}{\sigma^2}\beta(-\beta + \varepsilon_i/\sqrt{n})\big) - 1\big]\big)^2 \\
&= 4\beta^2\mathbb{E}_{\varepsilon_i\sim\mathcal{N}(0,\sigma^2)}\Big[\big(\text{sigmoid}\big(-2\frac{n}{\sigma^2}\beta^2 + 2\frac{\sqrt{n}}{\sigma^2}\beta\varepsilon_i\big)\big)^2\Big] \\
&= 4\beta^2\mathbb{E}_{\tilde{\varepsilon}_i\sim\mathcal{N}(0,1)}\Big[\big(\text{sigmoid}(-2\frac{n}{\sigma^2}\beta^2 + 2\frac{\beta\sqrt{n}}{\sigma}\tilde{\varepsilon}_i)\big)^2\Big].
\end{aligned}$$

We consider the function $\psi : \alpha \mapsto \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,1)}\big[\big(\text{sigmoid}(-2\alpha^2 + 2\alpha\varepsilon)\big)^2\big]$. We have $\psi(0) = 1/4$, and $\psi(\alpha) \to 0$ when $\alpha \to +\infty$, and $\psi(\alpha) \geqslant \frac{1}{4}\mathbb{P}_{\varepsilon\sim\mathcal{N}(0,1)}(\varepsilon > \alpha) \geqslant \frac{1}{8}\exp(-\alpha^2)$, by using simple Gaussian tail bounds (and since the sigmoid function is greater than 1/2 for positive numbers).

Thus, the total posterior variance $\mathbb{E}\big[\big\|\theta_* - \mathbb{E}\big[\theta_*|y\big]\big\|_2^2\big]$ is greater than

$$\frac{\beta^2 d}{2}\exp(-n\beta^2/\sigma^2) = \frac{\sigma^2 d}{n} \times \frac{\beta^2 n}{2\sigma^2}\exp(-n\beta^2/\sigma^2),$$

which is maximized for $\beta^2 \propto \sigma^2/n$, and thus if $\sigma^2 d/n$ is smaller than $D^2$, we obtain the usual $\sigma^2 d/n$, while if it is greater then $D^2$, we take $\beta_2^2 = D^2/d$, to obtain the lower bound

$$D^2\exp(-4nD^2/(\sigma^2 d)) \geqslant D^2\exp(-4),$$

which leads to the same bound as the previous section but with a more direct argument.

**Sparse case (♦).** To deal with the sparse case, we could consider a prior on $\theta_*$ that only selects $k$ non-zero elements out of $d$ and perform an analysis based on the posterior probability of $\theta_*$. Following Donoho and Johnstone (1994), it is easier to divide the set of $d$ variables into $k$ blocks of size $d/k$ (for simplicity, we assume that $d/k$ is an integer).

We then consider a prior probability defined independently on each of the $k$ blocks by selecting one of the $d/k$ variables uniformly at random and setting its value to $\beta$. In contrast, all others are set to zero.

To compute the posterior probability of $\theta_*$, we can treat each block independently and sum the posterior variances; we thus consider the first block, composed of $d/k$ variables, and compute the probability that the selected variable is the $j$-th one, which is proportional to

$$\exp(-n/(2\sigma^2)(y_j-\beta)^2)\prod_{i\neq j}\exp(-n/(2\sigma^2)(y_i)^2)\propto\exp(n\beta y_j/\sigma^2).$$

The conditional expectation of $\theta_*$ then satisfies

$$\mathbb{E}[(\theta_*)_i|y]=\beta\frac{\exp(n\beta y_i/\sigma^2)}{\sum_{j=1}^{d/k}\exp(n\beta y_j/\sigma^2)}.$$

To compute the posterior variance, we need to sample from the prior $\theta_*$. By symmetry, we may consider that $\theta_1=\beta$. If $y_1\leqslant\max_{j\neq 1}y_j$, then

$$\mathbb{E}[(\theta_*)_1|y]=\beta\frac{\exp(n\beta y_1/\sigma^2)}{\sum_{j=1}^{d/k}\exp(n\beta y_j/\sigma^2)}\leqslant\beta t\frac{\exp(n\beta y_1/\sigma^2)}{\exp(n\beta y_1/\sigma^2)+\exp(n\beta\max_{j\neq 1}y_j/\sigma^2)}\leqslant\beta/2,$$

and then the risk is at least $(\beta-\mathbb{E}[(\theta_*)_1|y])^2\geqslant\beta^2/4$.

In order to lower-bound the probability that $y_1\leqslant\max_{j\neq 1}y_j$, we can consider the events $\{y_1\leqslant\beta\}$ and $\{\beta\leqslant\max_{j\neq 1}y_j\}$. The probability that $y_1=\beta+\varepsilon_1$ is less than $\beta$ is greater than 1/2. Moreover, by independence of all $y_j$, $j\neq 1$,

$$\mathbb{P}\big(\{\beta\leqslant\max_{j\neq 1}y_j\}\big)\geqslant 1-\big(1-\mathbb{P}_{t\sim\mathcal{N}(0,1)}(t\geqslant\beta\sqrt{n}/\sigma)\big)^{d/k-1}.$$

Thus, the lower bound is greater than

$$k\frac{\beta^2}{8}\Big[1-\big(1-\mathbb{P}_{t\sim\mathcal{N}(0,1)}(t\geqslant\beta\sqrt{n}/\sigma)\big)^{d/k-1}\Big]\geqslant k\frac{\beta^2}{8}\Big[1-\Big(1-\frac{1}{2}\exp(-\beta^2n/\sigma^2)\Big)^{d/k-1}\Big],$$

using the Gaussian tail bound $\mathbb{P}_{t\sim\mathcal{N}(0,1)}(t\geqslant z)\geqslant\frac{1}{2}\exp(-z^2)$. We can then consider $\beta^2=\frac{\sigma^2}{n}\sqrt{2\log(d/k)}$, leading to a lower bound

$$\frac{\sigma^2k}{4n}\log(d/k)\big[(1-(1-\tfrac{1}{2}(k/d))^{d/k-1}\big],$$

which is greater than $\frac{\sigma^2k}{8n}\log(d/k)$ if $k\leqslant 2d$. We obtain the same lower-bound as the upper-bound for $\ell_0$-penalty-based methods in Chapter 8.

## 15.2 Optimization lower bounds

In this section, we consider ways of obtaining lower bounds of performance for optimization algorithms corresponding to upper-bounds derived in Chapter 5 for gradient-based algorithms. While the statistical lower bounds from the previous section were not obtained by explicitly building hard problems, the algorithmic lower bounds of this section will explicitly build such hard problems.

### 15.2.1 Convex optimization

To obtain computational lower bounds for convex optimization, which is notoriously hard in general in computer science, we will rely on a simple model of computation; that is, we will restrict ourselves to methods that access gradients of the objective function and combine them linearly to select a new query point.

We follow the results from Nesterov (2018, Section 2.1.2) and Bubeck (2015, Section 3.5), and assume that we want to minimize a convex function $F$ defined on $\mathbb{R}^d$. The algorithm starts from $\theta_0 = 0$ and can only query points in the span of the observed gradients or some sub-gradients of $F$ at the previously observed points.

The key is finding functions with the proper regularity properties, for which we know a few iterations provably lead to suboptimal performance. These functions will only reveal one new variable at each iteration and, after $k$ iterations, can only achieve the minimum on the first $k$ variables.

**Non-smooth functions.** We consider the following function, which will be dedicated to a given number of iterations $k$:

$$F(\theta) = \eta \max_{i \in \{1,\dots,k+1\}} \theta_i + \frac{\mu}{2}\|\theta\|_2^2,$$

for $k < d$, and $\eta, \mu$ positive parameters that will be set later.

The subdifferential of $F(\theta)$ is equal to

$$\mu\theta + \eta \cdot \mathrm{hull}\big(\big\{e_i,\ \theta_i = \max_{i' \in \{1,\dots,k+1\}} \theta_{i'}\big\}\big),$$

which is bounded in $\ell_2$-norm on the ball of radius $R$, by $\mu R + \eta$ (here $e_i$ denotes the $i$-th basis vector). We consider the oracle where the output gradient is $\mu\theta + \eta e_i$, where $i$ is the smallest index within maximizers of $\theta_{i'}$.

Starting from $\theta_0 = 0$, $\theta_1$ is supported on the first variable, and by recursion, after $k \leqslant d$ steps of subgradient descent, $\theta_k$ is supported on the first $k$ variables. Since $k < d$, then $(\theta_k)_{k+1} = 0$, so $F(\theta_k) \geqslant 0$. Minimizing over the span of the first $k+1$ variables leads to, by symmetry, $\theta_* = \kappa \sum_{i=1}^{k+1} e_i$, for a certain $\kappa$ which minimizes $\eta\kappa + \frac{(k+1)\mu}{2}\kappa^2$, so that $\kappa = -\frac{\eta}{\mu(k+1)}$, and thus $\theta_* = -\frac{\eta}{\mu(k+1)} \sum_{i=1}^{k+1} e_i$, with value $F(\theta_*) = -\frac{\eta^2}{2\mu(k+1)}$. Thus

$$F(\theta_k) - F(\theta_*) \geqslant 0 - F(\theta_*) = \frac{\eta^2}{2\mu(k+1)},$$

with $\|\theta_*\|_2^2 = \frac{\eta^2}{\mu^2(k+1)}$.

In order to build a $B$-Lipschitz-continuous function on a ball of center 0 and radius $D$, we can take $\eta = B/2$, and $D = B/(2\mu)$, and we get a lower bound of $\frac{B^2}{8\mu k}$.

With $\mu = \frac{B}{D}\frac{1}{1+\sqrt{k+1}}$ and $\eta = B\frac{\sqrt{k+1}}{1+\sqrt{k+1}}$, we also get a $B$-Lipschitz continuous function, and we get the lower bound $\frac{DB}{2(1+\sqrt{k+1})}$, which is valid as long as $k < d$.

⚠ The lower bounds are only valid for $k < d$ because there exist algorithms that are linearly convergent in this setting with a constant that depends on $d$, such as the ellipsoid method or the center of mass method (see Bubeck, 2015, for details).

**Smooth functions (♦).** We consider a sequence of quadratic functions on $\mathbb{R}^d$. We need that the gradient for iterates supported on the first $i$ components is supported on the first $i+1$ components, so that the $k$-th iterate starting from 0 only has its first $k$ coordinates that can be non-zero. We consider the example from Nesterov (2018, Section 2.1.2), and highlight the main arguments without proof:

$$F_k(\theta) = \frac{L}{4}\Big\{\frac{1}{2}\Big[\theta_1^2 + \theta_k^2 + \sum_{i=1}^{k-1}(\theta_i - \theta_{i+1})^2\Big] - \theta_1\Big\}.$$

The function $F_k$ is convex and smooth, with a smoothness constant less than $L$. Moreover, its global minimizer is attained at $\theta_*^{(k)}$ such that $(\theta_*^{(k)})_i = 1 - \frac{i}{k+1}$ for $i \in \{1,\dots,k\}$ and 0 otherwise, with an optimal value of $F_k(\theta_*^{(k)}) = \frac{L}{8}\frac{-k}{k+1}$, and with

$$\|\theta_*^{(k)}\|_2^2 = \sum_{i=1}^{k}\big(1 - \frac{i}{k+1}\big)^2 \leqslant \frac{k+1}{3}.$$

By construction, if $\theta$ is supported in the first $i$ components for $i < k$, then $F_k'(\theta)$ is supported on the first $i+1$ components. Thus, the $i$-th iterate is supported on the first $i$ components, and therefore the lowest attainable value is $F_i(\theta_*^{(i)})$.

Given this set of functions, for a given $k$ such that $k \leqslant \frac{d-1}{2}$, we consider $F_{2k+1}$, for which $\theta_*^{(2k+1)}$ is the global minimizer with value $\frac{L}{8}\frac{-2k-1}{2k+2}$, while after $k$ iterations, we can only achieve $F_k(\theta_*^{(k)}) = \frac{L}{8}\frac{-k}{k+1}$. Thus, we have:

$$\frac{F_{2k+1}(\theta_k) - F_{2k+1}^*}{\|\theta_0 - \theta_*\|_2^2} \geqslant \frac{L}{8}\frac{\frac{1}{k+1} - \frac{1}{2k+2}}{\frac{2k+2}{3}} \geqslant \frac{3L}{32}\frac{1}{(k+1)^2}.$$

We thus obtain the lower bounds corresponding to the upper bounds obtained from Nesterov acceleration.

⚠ The number of iterations has to be less than half the dimension for the lower bound to hold.

**Smooth strongly-convex functions (♦).** Following Nesterov (2018), we consider a function defined on the space $\ell_2$ of square-summable sequences as

$$F(\theta) = \frac{L-\mu}{4}\Big\{\frac{1}{2}\Big[\theta_1^2 + \sum_{i=1}^{\infty}(\theta_i-\theta_{i+1})^2\Big] - \theta_1\Big\} + \frac{\mu}{2}\|\theta\|_2^2.$$

This function is $L$-smooth and $\mu$-strongly convex. Its global minimizer is $\theta_*$ such that

$$(\theta_*)_k = \Big(\frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}\Big)^k = q^k,$$

with $\|\theta_*\|_2^2 = \sum_{k=1}^{\infty} q^{2k} = \frac{q^2}{1-q^2}$. Moreover, it can be shown that $\|\theta_k - \theta_*\|_2^2 \geqslant \sum_{i=k+1}^{\infty} q^{2i} = q^{2k}\|\theta_*\|_2^2$. This leads to $F(\theta_k) - F_* \geqslant \frac{\mu}{2}\|\theta_k - \theta_*\|_2^2 \geqslant q^{2k}\|\theta_0 - \theta_*\|_2^2$.

### 15.2.2 Non-convex optimization (♦)

While upper and lower bounds can have good behavior with respect to dimension in the convex case, this is not the case when removing the convexity assumption. In this section, we show that when optimizing a Lipschitz-continuous function on a compact subset of $\mathbb{R}^d$, we cannot hope to have guarantees which are not exponential in dimension.

This does not mean that all problem instances will require exponential time, but that in the worst-case sense, for any algorithm, there will always be a bad function.

We consider minimizing a function $F$ on a bounded subset $\Theta$ of $\mathbb{R}^d$, based only on function evaluations, a problem often referred to as zero-th order optimization or derivative-free optimization (see algorithms for convex functions in Section 11.2). No convexity is assumed in this section, so we should not expect fast rates and, again, no efficient algorithms that can provably find a global minimizer. Clearly, such algorithms are not made to be used to find millions of parameters for logistic regression or neural networks. Still, they are often used for hyperparameter tuning (regularization parameters, size of neural network layers, etc.). See, e.g., Snoek et al. (2012) for applications.

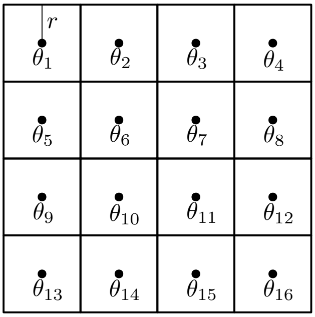
We will assume some regularity for the functions we want to minimize, typically bounded derivatives. We will thus assume that $f \in \mathcal{F}$, for a space $\mathcal{F}$ of functions from $\Theta$ to $\mathbb{R}$. We will take a worst-case approach, where we characterize convergence over all members of $\mathcal{F}$. That is, we want our guarantees to hold for *all* functions in $\mathcal{F}$. Note that this worst-case analysis may not predict well what is happening for a particular function; in particular, it is (by design) pessimistic.

An algorithm $\mathcal{A}$ will be characterized by (a) the choice of points $\theta_1, \dots, \theta_n \in \Theta$ to query the function, and (b) the algorithm to output a candidate $\hat{\theta} \in \Theta$ such that $F(\hat{\theta}) - \inf_{\theta\in\Theta} F(\theta)$ is small. The estimate $\hat{\theta}$ can only depend on $(\theta_i, F(\theta_i))$, for $i \in \{1, \dots, n\}$. In this section, the choice of points $\theta_1, \dots, \theta_n$ is made once (without seeing any function values).[1]

Given a selection of points and the algorithm $\mathcal{A}$, the rate of convergence is the supremum over all functions $F \in \mathcal{F}$ of the error $F(\hat{\theta}) - \inf_{\theta \in \Theta} F(\theta)$. This is a function $\varepsilon_n(\mathcal{A})$ of the number $n$ of sampled points (and of the class of functions $\mathcal{F}$). The optimal algorithm (minimizing $\varepsilon_n(\mathcal{A})$) will lead to a rate we denote $\varepsilon_n^{\text{opt}}$, and which we aim to characterize.

**Direct lower/upper bounds for Lipschitz-continuous functions.** The argument is particularly simple for a bounded metric space $\Theta$ with distance $\delta$, and $\mathcal{F}$ the class of $L$-Lipschitz-continuous functions, that is, such that for all $\theta, \theta' \in \Theta$, $|F(\theta) - F(\theta')| \leqslant L\delta(\theta, \theta')$. This is a very large set of functions, so we expect weak convergence rates.

Like in Section 4.4.4, we will need to cover the set $\Theta$ with balls of a given radius. The minimal radius $r$ of a cover of $\Theta$ by $n$ balls of radius $r$ is denoted $r_n(\Theta, \delta)$. This corresponds to $n$ ball centers $\theta_1, \dots, \theta_n$. See example below for the unit cube $\Theta = [0,1]^2$ and the metric obtained from the $\ell_\infty$-norm, with $n = 16$, and $r_n([0,1]^2, \ell_\infty) = 1/8$.



More generally, for the unit cube $\Theta = [0,1]^d$, we have $r_n([0,1]^d, \ell_\infty) \approx \frac{1}{2} n^{-1/d}$ (which is not an approximation when $n$ is the $d$-th power of an integer). For other normed metrics (since all norms are equivalent), the scaling as $r_n \sim \operatorname{diam}(\Theta) n^{-1/d}$ is the same on any bounded set in $\mathbb{R}^d$ (with an extra constant that depends on $d$).

**Naive algorithm.** Given the ball centers $\theta_1, \dots, \theta_n$, outputting the minimum of function values $F(\theta_i)$ for $i = 1, \dots, n$, leads to an error which is less than $Lr_n(\Theta, \delta)$, as the optimal $\theta_* \in \Theta$ is at most at distance $r_n(\Theta, \delta)$ from one of the cluster centers, let's say $\theta_k$, and thus $F(\theta_k) - F(\theta_*) \leqslant L\delta(\theta_k, \theta_*) \leqslant Lr_n(\Theta, \delta)$. This provides an upper bound on $\varepsilon_n^{\text{opt}}$. The algorithm we just described seems naive, but it turns out to be optimal for this class of problems.

**Lower bound.** Consider any optimization algorithm, with its first $n$ point queries and its estimate $\hat{\theta}$. By considering the functions that are zero in these $n+1$ points, the algorithm can only output an arbitrary fixed real number for the optimal value (let's say

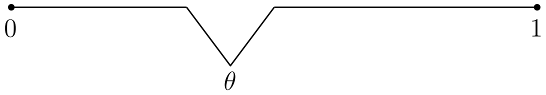
[1] It turns out that going *adaptive*, where the point $\theta_{i+1}$ is selected after seeing $(\theta_j, F(\theta_j))$ for all $j \leqslant i$, does not bring much (at least in the worst-case sense) (Novak, 2006).

zero). We now simply need to construct a function $F \in \mathcal{F}$ such that $F$ is zero at these points but maximally smaller than zero at a different point.

Given the $n+1$ points above, there is at least a point $\eta \in \Theta$ which is at a distance at most $r_{n+1}(\Theta, \delta)$ from all of them (otherwise, we obtain a cover of $\Theta$ with $n+1$ points). We can then construct the function

$$F(\theta) = \ -L\big(r_{n+1}(\Theta, \delta) - \delta(\theta, \eta)\big)_+ = -L \max\big\{r_{n+1}(\Theta, \delta) - d(\theta, \eta), 0\big\},$$

which is $L$-Lipschitz-continuous, equal to zero on all points of the algorithm and the output point $\hat{\theta}$, and with minimum value $-Lr_{n+1}(\Theta, \delta)$ attained at $\eta$. Thus, we must have $\varepsilon_n^{\rm opt} \geqslant 0 - (-Lr_{n+1}(\Theta, \delta)) = Lr_{n+1}(\Theta, \delta)$. This difficult function is plotted below in one dimension.



Thus, the performance of any algorithm from $n$ function values has to be larger than $Lr_{n+1}(\Theta, \delta)$. Thus, so far, we have shown that

$$Lr_{n+1}(\Theta, \delta) \leqslant \varepsilon_n^{\rm opt} \leqslant Lr_n(\Theta, \delta).$$

For $\Theta \subset \mathbb{R}^d$, $r_n(\Theta, \delta)$ is typically of order $\mathrm{diam}(\Theta) n^{-1/d}$, and thus the difference between $n$ and $n+1$ above is negligible. Note that the rate in $n^{-1/d}$ is *very* slow and symptomatic of the classical curse of dimensionality. The appearance of a covering number is not totally random here and comes from the equivalence in terms of worst-case guarantees between optimization and uniform approximation (Novak, 2006).

**Random search.** We can have a similar bound up to logarithmic terms for random search, that is, after selecting independently $n$ points $\theta_1, \dots, \theta_n$, uniformly at random in $\Theta$, and selecting the points with smallest function value $F(\theta_i)$. The performance can be shown to be proportional to $L\mathrm{diam}(\Theta)(\log n)^{1/d} n^{-1/d}$ in high probability, leading to an additional logarithmic term (the proof can be obtained with a simple covering argument, see exercise below). Therefore, random search is optimal up to logarithmic terms for optimizing this very large class of functions.

To go beyond Lipschitz-continuous functions, we can leverage smoothness like in supervised learning and hopefully avoid the dependence in $n^{-1/d}$. This can be done by a somewhat surprising equivalence between worst-case guarantees from optimization and worst-case guarantees for uniform approximation.[2]

**Exercise 15.2** *(♦) Consider sampling independently and uniformly in $\Theta$ $n$ points $\theta_1, \dots, \theta_n$.*

*(a) For a given $L$-Lipschitz-continuous function $F$, show that the worst-case performance of outputting the lower function value is less than $L \max_{\theta \in \Theta} \min_{i \in \{1,\dots,n\}} \delta(\theta, \theta_i)$.*

[2] See https://francisbach.com/optimization-is-as-hard-as-approximation/ for more details, as well as Novak (2006).

*(b) Considering an optimal cover with $m$ points and radius $r = r_m(\Theta, d)$, show that*

$$\mathbb{P}\Big(\max_{\theta\in\Theta}\min_{i\in\{1,\dots,n\}}\delta(\theta,\theta_i)\geqslant 2r\Big)\leqslant m(1-1/m)^n.$$

*(c) By the appropriate choice of $m$, show that when $r \sim m^{-1/d}\mathrm{diam}(\mathcal{X})$, we get an overall performance proportional to $L\big(\frac{\log n}{n}\big)^{1/d}$ with probability greater than $1-\frac{\log n}{n}$.*

## 15.3 Lower bounds for stochastic gradient descent (♦)

In this section, our goal is to show that the convergence rates for stochastic gradient descent (SGD) shown in Section 5.4 are "optimal", in a sense to be made precise. We consider a class $\mathcal{F}$ of functions, here the convex $B$-Lipschitz-continuous functions on the ball of center zero and radius $D$ (for the Euclidean norm). We consider the class $\mathcal{A}$ of algorithms that can sequentially access independent random, unbiased estimates of the gradients of a function $F$ in $\mathcal{F}$, with squared norm bounded by $B^2$. We denote $A_t(F)\in\mathbb{R}^d$ the output of algorithm $A$. Our goal is to find upper and lower bounds on

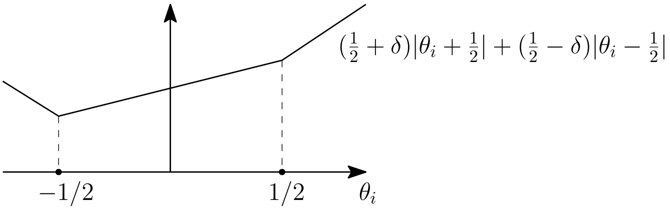
$$\varepsilon_t(\mathcal{A},\mathcal{F}) = \inf_{A\in\mathcal{A}}\ \sup_{f\in\mathcal{F}}\ \mathbb{E}\big[F(A_t(F)) - \inf_{\|\theta\|_2\leqslant D}F(\theta)\big].$$

SGD is an algorithm in $\mathcal{A}$ achieving a bound proportional to $BD/\sqrt{t}$, thus, up to a constant, $\varepsilon_t(\mathcal{A},\mathcal{F})\leqslant BD/\sqrt{t}$. We now prove a matching lower-bound by exhibiting a set of functions that will make any algorithm have this desired performance. Note that, as opposed to Section 15.2.1 on deterministic convex optimization, we make no assumption on the running-time complexity of algorithms in $\mathcal{A}$.

We follow the exposition from Agarwal et al. (2012) and consider a function

$$F_\alpha(\theta) = \frac{B}{2d}\sum_{i=1}^d\Big\{\big(\frac{1}{2}+\alpha_i\delta\big)\cdot\big|\theta_i+\frac{1}{2}\big| + \big(\frac{1}{2}-\alpha_i\delta\big)\cdot\big|\theta_i-\frac{1}{2}\big|\Big\}, \tag{15.7}$$

with $\alpha\in\{-1,1\}^d$ a well chosen vector and $\delta\in(0,1/4]$, and $B>0$. One element of the sum is plotted below.



The function $F_\alpha$ is convex and Lipschitz-continuous with gradients bounded in $L_2$-norm by $B/(2\sqrt{d})$. Moreover, the global minimizer of $F_\alpha$ is $\theta=-\frac{\alpha}{2}$, with an optimal value equal to $F_\alpha^* = \frac{B}{4}(1-2\delta)$. That is, minimizing $F_\alpha$ on $[-1/2,1/2]^d$ exactly corresponds to finding an element of the hypercube $\alpha$. Moreover, it turns out that minimizing it approximately also leads to identifying $\alpha$ among a set of $\alpha$'s, which are sufficiently different.

**Lemma 15.3** *If $\alpha, \beta \in \{-1,1\}^d$, and $F_\alpha(\theta) - F_\alpha^* \leqslant \varepsilon$, then $F_\beta(\theta) - F_\beta^* \geqslant \frac{B\delta}{2d}\|\alpha - \beta\|_1 - \varepsilon$.*

**Proof** (♦) We have: $F_\beta(\theta) - F_\beta^* = F_\beta(\theta) + F_\alpha(\theta) - F_\beta^* - F_\alpha^* + [F_\alpha^* - F_\alpha(\theta)]$. We then notice that for all $\theta \in \mathbb{R}^d$,

$$F_\beta(\theta) + F_\alpha(\theta) - F_\beta^* - F_\alpha^* \geqslant \frac{B}{2d} \sum_{i,\ \alpha_i \neq \beta_i} \big\{ \big|\theta_i + \frac{1}{2}\big| + \big|\theta_i - \frac{1}{2}\big| + 2\delta - 1 \big\} \geqslant \frac{B\delta}{2d}\|\alpha - \beta\|_1.$$

■

Thus, if we consider $M$ points $\alpha^{(1)}, \dots, \alpha^{(M)} \in \{-1,1\}^d$ such that $\|\alpha^{(i)} - \alpha^{(j)}\|_1 \geqslant \frac{d}{2}$ (with potentially $M \geqslant \exp(d/8)$ such points from Lemma 15.2), then, if $\varepsilon < \frac{B\delta}{8}$, because of Lemma 15.3, minimizing up to $\varepsilon$ exactly identifies which of the functions $F_{\alpha^{(i)}}$ was being minimized.

Moreover, if $\hat\theta$ is random then, denoting $\mathcal{A} = \{\alpha^{(1)}, \dots, \alpha^{(M)}\}$, following the same reasoning as in Section 15.1.2:

$$\sup_{\alpha \in \mathcal{A}} \mathbb{E}_\alpha\big[F_\alpha(\hat\theta) - F_\alpha^*\big] \geqslant \varepsilon \cdot \sup_{\alpha \in \mathcal{A}} \mathbb{P}_\alpha\big(F_\alpha(\hat\theta) - F_\alpha^* > \varepsilon\big) \geqslant \varepsilon \cdot \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \mathbb{P}_\alpha\big(F_\alpha(\hat\theta) - F_\alpha^* > \varepsilon\big).$$

From an estimate $\hat\theta$, we can build a test $g(\hat\theta) \in \mathcal{A}$ by selecting the (unique if $\varepsilon < \frac{B\delta}{8}$) $\alpha \in \mathcal{A}$ such that $F_\alpha(\hat\theta) - F_\alpha^* \leqslant \varepsilon$ if it exists, and uniformly at random in $\mathcal{A}$ otherwise. Therefore, the minimax performance is greater than $\varepsilon$ times the probability of a mistake in the best possible test.

We consider the following stochastic oracle:

(1) pick some coordinate $i \in \{1, \dots, d\}$ uniformly at random,

(2) draw a Bernoulli random variable $b_i \in \{0,1\}$ with parameter $\frac{1}{2} + \alpha_i\delta$,

(3) consider $\hat F(\theta) = b_i\big|\theta_i + \frac{1}{2}\big| + (1 - b_i)\big|\theta_i - \frac{1}{2}\big|$, with gradient with components

$$\hat{F}_\alpha'(\theta)_i = \frac{B}{2}\big[b_i \operatorname{sign}(\theta_i + 1/2) + (1 - b_i)\operatorname{sign}(\theta_i - 1/2)\big].$$

The stochastic gradients have an $\ell_2$-norm bounded by $B$ and are unbiased. Moreover, observation of the gradient for a $\theta \in [-1/2, 1/2]^d$ reveals the outcome of the Bernoulli random variable $b_i$.

Therefore, after $t$ steps, we can apply Fano's inequality to the following set-up: the random variable $\alpha \in \mathcal{A}$ is uniform, and given $\alpha$, we sample independently $t$ times, one variable $i$ in $\{1, \dots, d\}$ and observe (a potentially noisy version of) a Bernoulli random variable $b$, with parameter $\alpha_i$.

We then need to upper bound the mutual information between $\alpha$ and $(i, b)$ and multiply the result $t$ times because each of the $t$ gradients is sampled independently.

The mutual information can be decomposed as

$$I(\alpha, (i,b)) \quad = \quad I(\alpha, i) + I(\alpha, b|i) = 0 + \mathbb{E}_i \mathbb{E}_\alpha\big[D_{\mathrm{KL}}(p(b|i,\alpha)||p(b|i))\big],$$

where $p(b|i,\alpha)$ and $p(b|i)$ denotes the probability distribution of $b$. Thus, by convexity of the KL divergence,

$$\begin{aligned} I(\alpha,(i,b)) &= \mathbb{E}_i\mathbb{E}_\alpha\Big[D_{\mathrm{KL}}\Big(p(b|i,\alpha)\Big\|\frac{1}{|\mathcal{A}|}\sum_{\alpha'\in\mathcal{A}} p(b|i,\alpha')\Big)\Big] \\ &\leqslant \frac{1}{|\mathcal{A}|}\sum_{\alpha'\in\mathcal{A}} \mathbb{E}_i\mathbb{E}_\alpha\big[D_{\mathrm{KL}}(p(b|i,\alpha)||p(b|i,\alpha'))\big]. \end{aligned}$$

Since $p(b|i,\alpha)$ is Bernoulli random variable with parameter $\frac{1}{2}+\delta$ or $\frac{1}{2}-\delta$, the KL divergences above are bounded by the KL between two Bernoulli random variables with the two different parameters, that is,

$$\begin{aligned} I(\alpha,(i,b)) &\leqslant \big(\frac{1}{2}+\delta\big)\log\frac{\frac{1}{2}+\delta}{\frac{1}{2}-\delta}+\big(\frac{1}{2}-\delta\big)\log\frac{\frac{1}{2}-\delta}{\frac{1}{2}+\delta} = 2\delta\log\frac{1+2\delta}{1-2\delta} \\ &= 2\delta\log\Big(1+\frac{4\delta}{1-2\delta}\Big) \leqslant \frac{8\delta^2}{1-2\delta} \leqslant 16\delta^2 \text{ if } \delta\in[0,1/4]. \end{aligned}$$

Therefore, applying Corollary 15.1, the minimax lower bound is greater than

$$\varepsilon\Big(1-\frac{16t\delta^2-\log 2}{\log M}\Big) \geqslant \varepsilon\Big(1-\frac{16t\delta^2-\log 2}{d/8}\Big).$$

Thus, we need $256t\delta^2 \geqslant d$, and then $B\delta/4$ is the lower bound on the rate so that the lower bound is

$$\frac{1}{16}\sqrt{\frac{d}{t}},$$

which is the desired lower-bound (up to a constant) $\varepsilon_t(\mathcal{A},\mathcal{F}) \geqslant DB/\sqrt{t}$ where $D$ is the diameter of the set of $\theta$. The lower-bound is thus the same as the upper-bound achieved by stochastic gradient descent in Chapter 5. The result above can be extended to strongly-convex problems (Agarwal et al., 2012).

## 15.4 Conclusion

This chapter was entirely dedicated to lower bounds of performance associated with the upper bounds presented in the rest of the book. Statistical lower bounds are obtained by reducing the learning problem to a hypothesis test where information theory is brought to bear. In comparison, optimization lower bounds are obtained by designing functions that are explicitly hard to optimize for the proposed computational model of combining gradients linearly.

# Bibliography

J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 414–424, 2008. (cited on page 297)

R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. Elsevier, 2003. (cited on page 401)

A. Agarwal, M. J. Wainwright, P. L. Bartlett, and P. K. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, 2009. (cited on pages 130 and 133)

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012. (cited on pages 409 and 411)

N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5):1–27, 2008. (cited on page 369)

E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2020. (cited on page x)

E. Alpaydin. *Maschinelles Lernen*. de Gruyter, 2022. (cited on page x)

P. Alquier. User-friendly introduction to PAC-Bayes bounds. Technical Report 2110.11216, arXiv, 2021. (cited on pages 386, 387, and 388)

P. Alquier and K. Lounici. PAC-Bayesian bounds for sparse regression estimation with exponential weights. *Electronic Journal of Statistics*, 5:127–145, 2011. (cited on page 388)

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008. (cited on page 337)

L. Ambrosio, N. Gigli, and G. Savaré. Density of Lipschitz functions and equivalence of weak gradients in metric measure spaces. *Revista Matemática Iberoamericana*, 29(3): 969–996, 2013. (cited on page 164)

D. Araújo, R. I. Oliveira, and D. Yukimura. A mean-field limit for certain deep neural networks. *arXiv preprint arXiv:1906.00193*, 2019. (cited on page 339)

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. (cited on page 24)

L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives.

*Pacific Journal of mathematics*, 16(1):1–3, 1966. (cited on page 106)

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337 – 404, 1950. (cited on pages 173 and 174)

A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008. (cited on page 128)

A. d'Aspremont, D. Scieur, and A. Taylor. Acceleration methods. *Foundations and Trends in Optimization*, 5(1-2):1–245, 2021. (cited on page 120)

J.-Y. Audibert and A. B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007. (cited on pages 96 and 153)

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002a. (cited on page 310)

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b. (cited on page 314)

C.-A. Azencott. *Introduction au Machine Learning*. Dunod, 2019. (cited on page x)

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. Technical Report 1607.06450, arXiv, 2016. (cited on page 237)

F. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(Jun):1019–1048, 2008. (cited on page 230)

F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209. PMLR, 2013. (cited on page 204)

F. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *Journal of Machine Learning Research*, 15(1):595–627, 2014. (cited on page 138)

F. Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015. (cited on page 250)

F. Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017. (cited on pages 244, 250, 252, 253, 254, 255, and 256)

F. Bach. High-dimensional analysis of double descent for linear regression with random projections. Technical Report 2303.01372, arXiv, 2023a. (cited on pages 330, 332, and 334)

F. Bach. On the relationship between multivariate splines and infinitely-wide neural networks. Technical Report 2302.03459, arXiv, 2023b. (cited on pages 257 and 259)

F. Bach and L. Chizat. Gradient descent on infinitely wide neural networks: Global convergence and generalization. In *Proceedings of the International Congress of Mathematicians*, 2022. (cited on pages 237, 335, and 338)

F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. *Advances in Neural Information Processing Systems*, 20, 2007. (cited on page 97)

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems*, 2013. (cited on pages 133 and 138)

F. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research*, 7:1713–1741, 2006. (cited on page 26)

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012a. (cited on pages 219 and 230)

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012b. (cited on page 230)

K. Ball, E. A. Carlen, and E. H. Lieb. Sharp uniform convexity and smoothness inequalities for trace norms. *Inequalities: Selecta of Elliott H. Lieb*, pages 171–190, 2002. (cited on page 296)

N. Bansal and A. Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(1):1–32, 2019. (cited on page 118)

A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. (cited on pages 249 and 336)

A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14:115–133, 1994. (cited on page 249)

A. R. Barron and J. M. Klusowski. Approximation and estimation for high-dimensional deep learning networks. Technical Report 1809.03090, arXiv, 2018. (cited on page 244)

A. R. Barron, A. Cohen, W. Dahmen, and R. A. DeVore. Approximation and learning by greedy algorithms. *The Annals of statistics*, 36(1):64–94, 2008. (cited on page 288)

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. (cited on pages 86 and 87)

P. L. Bartlett and M. Traskin. Adaboost is consistent. *Journal of Machine Learning Research*, 8(10), 2007. (cited on page 285)

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005. (cited on page 93)

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. (cited on pages 75, 77, and 78)

P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. (cited on page 334)

A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18, 2018. (cited on page 300)

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. (cited on page 122)

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. (cited on pages 327 and 329)

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, volume 3. Springer, 2004. (cited on page 174)

Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with differentiable pertubed optimizers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. (cited on page 363)

R. Berthier. Incremental learning in diagonal linear networks. *Journal of Machine Learning Research*, 24(171):1–26, 2023. (cited on page 343)

R. Bhatia. *Positive Definite Matrices*, volume 24. Princeton University Press, 2009. (cited on page 111)

R. Bhatia. *Matrix Analysis*, volume 169. Springer Science & Business Media, 2013. (cited on page 7)

G. Biau and L. Devroye. *Lectures on the Nearest Neighbor Method*, volume 246. Springer, 2015. (cited on pages 153, 158, 160, and 166)

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016. (cited on page 271)

G. Biau, F. Cérou, and A. Guyader. On the rate of convergence of the bagged nearest neighbor estimate. *Journal of Machine Learning Research*, 11(2), 2010. (cited on pages 268 and 269)

A. Bietti and F. Bach. Deep equals shallow for ReLU networks in kernel regimes. In *International Conference on Learning Representations*, 2021. (cited on page 345)

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. (cited on pages 39 and 374)

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003. (cited on page 382)

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. (cited on page 385)

M. Blondel, A. F. T. Martins, and V. Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020. (cited on page 360)

T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009. (cited on page 216)

J. Bolte, A. Daniilidis, and A. Lewis. A nonsmooth Morse–Sard theorem for subanalytic functions. *Journal of mathematical analysis and applications*, 321(2):729–740, 2006.

(cited on page 341)

J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of lojasiewicz inequalities and applications. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010. (cited on pages 318 and 320)

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005. (cited on page 86)

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. (cited on page 9)

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002. (cited on pages 96 and 97)

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. (cited on pages 73, 78, 91, 111, 117, 121, 187, and 319)

H. Brass and K. Petras. *Quadrature theory: the theory of numerical integration on a compact interval*. Number 178 in Mathematical Surveys and Monographs. American Mathematical Society, 2011. (cited on page 18)

L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993. (cited on page 244)

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. (cited on page 271)

L. Breiman and D. Freedman. How many variables should be entered in a regression equation? *Journal of the American Statistical Association*, 78(381):131–136, 1983. (cited on page 61)

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. Technical Report 2104.13478, arXiv, 2021. (cited on page 261)

C. Brouard, M. Szafranski, and F. d'Alché Buc. Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17:1–48, 2016. (cited on page 361)

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015. (cited on pages 103, 126, 143, 404, and 405)

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. (cited on pages 292, 307, 313, and 315)

V. Cabannes, L. Pillaud-Vivien, F. Bach, and A. Rudi. Overcoming the curse of dimensionality with Laplacian regularization in semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. (cited on page 97)

O. Catoni. A PAC-Bayesian approach to adaptive classification. Technical Report 840, LPMA, 2003. (cited on page 387)

O. Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical*

*Learning*, volume 56. Institute of Mathematical Statistics, 2007. (cited on page 96)

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. (cited on page 40)

V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12:805–849, 2012. (cited on page 280)

O. Chapelle, B. Scholkopf, and A. e. Zien. *Semi-Supervised Learning*. MIT Press, 2010. (cited on page 39)

K. Chaudhuri and S. Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 3437–3445, 2014. (cited on page 153)

G. H. Chen and D. Shah. *Explaining the Success of Nearest Neighbor Methods in Prediction*. Now Publishers, 2018. (cited on pages 153 and 160)

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018. (cited on page 261)

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. (cited on page 279)

L. Chizat. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, pages 1–46, 2021. (cited on page 336)

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, pages 3036–3046, 2018. (cited on pages 237, 335, 337, and 338)

L. Chizat and F. Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Proceedings of the Conference on Learning Theory*, 2020. (cited on pages 327 and 345)

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019. (cited on pages 261 and 344)

Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, 2009. (cited on page 256)

A. Christmann and I. Steinwart. *Support Vector Machines*. Springer, 2008. (cited on pages x, xi, 25, and 169)

C. Ciliberto, L. Rosasco, and A. Rudi. A general framework for consistent structured prediction with implicit loss embeddings. *Journal of Machine Learning Research*, 21 (98):1–67, 2020. (cited on pages 355, 357, 360, and 362)

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. (cited on page 73)

T. M. Cover and J. A. Thomas. *Elements of information Theory*. John Wiley & Sons,

1999. (cited on pages 395 and 397)

M. Cuturi, K. Fukumizu, and J.-P. Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198, 2005. (cited on page 183)

P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Academic Press, 1984. (cited on page 18)

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 2014. (cited on page 138)

A. Défossez and F. Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2015. (cited on page 138)

A. Défossez, L. Bottou, F. Bach, and N. Usunier. A simple convergence proof of adam and adagrad. *Transactions on Machine Learning Research*, 2022. (cited on page 135)

R. A. DeVore and V. N. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Mathematics*, 5:173–187, 1996. (cited on page 282)

L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic Theory of Pattern Recognition*, volume 31. Springer Science & Business Media, 1996. (cited on pages xi, 36, 38, and 76)

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1994. (cited on page 350)

A. Dieuleveut and F. Bach. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399, 2016. (cited on page 138)

A. Dieuleveut, N. Flammarion, and F. Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *Journal of Machine Learning Research*, 18(1):3520–3570, 2017. (cited on page 138)

E. Dobriban and S. Liu. Asymptotics for sketching in least squares regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019. (cited on page 273)

D. L. Donoho and I. M. Johnstone. Minimax risk over $\ell_p$-balls for $\ell_q$-error. *Probability Theory and Related Fields*, 99(2):277–303, 1994. (cited on pages 401 and 402)

S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018. (cited on page 344)

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011. (cited on pages 135 and 237)

J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015. (cited on page 303)

W. E and S. Wojtowytsch. On the Banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics. *arXiv preprint arXiv:2007.15623*, 2020. (cited on page 339)

L. C. Evans. *Partial Differential Equations*, volume 19. American Mathematical Society, 2022. (cited on page 338)

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society, Series B*, 70(5):849–911, 2008. (cited on page 383)

C. Fang, J. Lee, P. Yang, and T. Zhang. Modeling from features: a mean-field framework for over-parameterized deep neural networks. In *Conference on Learning Theory*, pages 1887–1936. PMLR, 2021. (cited on page 339)

R. Fathony, A. Liu, K. Asif, and B. Ziebart. Adversarial multiclass classification: A risk minimization perspective. *Advances in Neural Information Processing Systems*, 29, 2016. (cited on page 365)

O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015. (cited on page 218)

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, 1996. (cited on pages 281 and 283)

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. (cited on page 314)

Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999. (cited on page 278)

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, 2009. (cited on page 150)

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. (cited on page 284)

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016. (cited on page 97)

A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the Conference on Learning Theory*, 2011. (cited on page 311)

M. Geiger, A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. d'Ascoli, G. Biroli, C. Hongler, and M. Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, 2020. (cited on pages 327 and 329)

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995. (cited on page 384)

G. Gidel, F. Bach, and S. Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 2019. (cited on page 343)

C. Giraud. *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC, 2014. (cited on pages 213, 220, 223, 226, 227, and 229)

C. Giraud, S. Huet, and N. Verzelen. High-dimensional regression with unknown variance. *Statistical Science*, 27(4):500–518, 2012. (cited on page 215)

A. A. Goldstein. Cauchy's method of minimization. *Numerische Mathematik*, 4(1):146–150, 1962. (cited on page 106)

G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, 1996. (cited on pages 7, 47, 64, 109, 122, and 185)

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016. (cited on pages 39, 261, and 339)

R. M. Gower, M. Schmidt, F. Bach, and P. Richtarik. Variance-reduced methods for machine learning. Technical Report 2010.00892, arXiv, 2020. (cited on page 141)

R. Gribonval. Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Transactions on Signal Processing*, 59(5):2405–2410, 2011. (cited on page 377)

S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. In *In International Conference on Machine Learning*, 2018. (cited on page 321)

D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997. (cited on page 184)

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution-free Theory of Nonparametric Regression*. Springer Science & Business Media, 2006. (cited on page 156)

L. R. Haff. An identity for the Wishart distribution with applications. *Journal of Multivariate Analysis*, 9(4):531–544, 1979. (cited on page 331)

T. Hamm and I. Steinwart. Adaptive learning rates for support vector machines working on data with low intrinsic dimension. *The Annals of Statistics*, 49(6):3153–3180, 2021. (cited on page 38)

B. Hanin and M. Nica. Finite depth and width corrections to the neural tangent kernel. In *International Conference on Learning Representations*, 2019. (cited on page 339)

Z. Harchaoui, F. Bach, and E. Moulines. Testing for homogeneity with kernel fisher discriminant analysis. Technical Report 0804.1026, arXiv, 2008. (cited on page 203)

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. Technical Report 903.08560, arXiv, 2019. (cited on pages 327 and 330)

E. Hazan. *Introduction to Online Convex Optimization*. MIT Press, 2022. (cited on pages 40, 292, 307, and 315)

E. Hazan and S. Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15(1): 2489–2512, 2014. (cited on page 295)

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. (cited on page 261)

M. Holtz. *Sparse grid quadrature in high dimensions with applications in finance and insurance*, volume 77. Springer Science & Business Media, 2010. (cited on page 18)

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. In *Conference on Learning Theory*, 2012. (cited on page 62)

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Willey ans Sons, 2001. (cited on page 39)

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. (cited on page 237)

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018. (cited on pages 261 and 345)

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, 2013. (cited on page 250)

Z. Ji and M. Telgarsky. Risk and parameter convergence of logistic regression. Technical Report 1803.07300, arXiv, 2018. (cited on page 323)

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, volume 26, 2013. (cited on page 138)

W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference on Modern Analysis and Probability*, volume 26, pages 189–206, 1984. (cited on page 277)

A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017. (cited on page 184)

A. Juditsky and A. Nemirovski. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods. *Optimization for Machine Learning*, pages 121–148, 2011a. (cited on page 126)

A. Juditsky and A. Nemirovski. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, 30 (9):149–183, 2011b. (cited on page 126)

A. Kabán. New bounds on compressive linear least squares regression. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2014. (cited on page 274)

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016. (cited on page 320)

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971. (cited on page 171)

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. Technical Report 1412.6980, arXiv, 2014. (cited on pages 135 and 237)

J. M. Klusowski and A. R. Barron. Approximation by combinations of relu and squared relu ridge functions with $\ell^1$ and $\ell^0$ controls. *IEEE Transactions on Information Theory*, 64(12):7649–7656, 2018. (cited on page 249)

J. M. Klusowski and J. W. Siegel. Sharp convergence rates for matching pursuit. Technical Report 2307.07679, arXiv, 2023. (cited on page 282)

V. Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems: École D'Été de Probabilités de Saint-Flour (2008)*, volume 2033. Springer Science & Business Media, 2011. (cited on page x)

V. Koltchinskii and O. Beznosova. Exponential convergence rates in classification. In *International Conference on Computational Learning Theory*, pages 295–307. Springer, 2005. (cited on page 96)

S. Kpotufe. k-NN regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems*, volume 24, 2011. (cited on page 38)

V. Kurková and M. Sanguineti. Bounds on rates of variable-basis and neural-network approximation. *IEEE Transactions on Information Theory*, 47(6):2659–2665, 2001. (cited on pages 243, 279, and 336)

H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, second edition, 2003. (cited on page 337)

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. (cited on pages 292, 307, and 315)

N. Le Roux and Y. Bengio. Continuous neural networks. In *Artificial Intelligence and Statistics*, pages 404–411, 2007. (cited on page 186)

G. Lecué and S. Mendelson. Performance of empirical risk minimization in linear aggregation. *Bernoulli*, 22(3):1520–1534, 2016. (cited on page 62)

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*, volume 23. Springer Science & Business Media, 1991. (cited on pages 89 and 90)

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004. (cited on page 360)

M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. (cited on page 242)

D. Liberzon. *Calculus of Variations and Optimal Control Theory: a Concise Introduction*. Princeton University Press, 2011. (cited on page 40)

A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön. *Machine Learning: a First Course for Engineers and Scientists*. Cambridge University Press, 2022. (cited on page x)

Y. Liu. Fisher consistency of multicategory support vector machines. In *Artificial Intelligence and Statistics*, pages 291–298, 2007. (cited on page 365)

P. Long and R. Servedio. Consistency versus realizable H-consistency for multiclass classification. In *International Conference on Machine Learning*, pages 801–809, 2013. (cited on pages 80 and 359)

J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep network approximation for smooth functions. Technical Report 2001.03040, arXiv, 2020. (cited on page 261)

G. Lugosi and N. Vayatis. On the Bayes-risk consistency of regularized boosting methods. *The Annals of statistics*, 32(1):30–55, 2004. (cited on page 288)

K. Lyu and J. Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2019. (cited on page 345)

C. Ma, S. Wojtowytsch, and L. Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. Technical Report 2009.10713, arXiv, 2020. (cited on page 261)

J. Mairal and B. Yu. Complexity analysis of the Lasso regularization path. In *Proceedings of the International Coference on International Conference on Machine Learning*, 2012. (cited on page 220)

J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014. (cited on pages 39 and 230)

S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. (cited on pages 281 and 282)

J. I. Marden. *Analyzing and Modeling Rank Data*. CRC Press, 1996. (cited on page 370)

P.-G. Martinsson and J. A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020. (cited on page 122)

A. Maurer. A vector-contraction inequality for Rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17, 2016. (cited on pages 352 and 353)

S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022. (cited on pages 327 and 329)

S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. (cited on page 337)

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pages 2388–2464. PMLR, 2019. (cited on page 339)

R. Meir and T. Zhang. Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research*, 4(Oct):839–860, 2003. (cited on page 89)

S. Minsker. On some extensions of Bernstein's inequality for self-adjoint operators. *Statistics & Probability Letters*, 127:111–119, 2017. (cited on page 19)

M. Mohri and A. Rostamizadeh. Stability bounds for stationary $\varphi$-mixing and $\beta$-mixing processes. *Journal of Machine Learning Research*, 11(2), 2010. (cited on page 97)

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018. (cited on page x)

J. Mourtada. Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. Technical Report 1912.10754, arXiv, 2019. (cited on pages 57, 61, and 62)

J. Mourtada and L. Rosasco. An elementary analysis of ridge regression with random design. *Comptes Rendus. Mathématique*, 360:1055–1063, 2022. (cited on page 198)

R. Munos et al. From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends in Machine Learning*, 7(1):1–129, 2014. (cited on page 310)

K. P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012. (cited on pages 39, 362, 363, and 374)

D. Nagaraj, P. Jain, and P. Netrapalli. Sgd without replacement: Sharper rates for general smooth convex functions. In *International Conference on Machine Learning*, pages 4703–4711, 2019. (cited on page 128)

R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995. (cited on page 256)

A. Nekvinda and L. Zajíček. A simple proof of the Rademacher theorem. *Časopis pro pěstování matematiky*, 113(4):337–341, 1988. (cited on page 123)

Y. Nesterov. *Introductory Lectures on Convex Optimization: a Basic Course*. Kluwer, 2004. (cited on page 120)

Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep*, 76, 2007. (cited on pages 120 and 122)

Y. Nesterov. *Lectures on Convex Optimization*, volume 137. Springer, 2018. (cited on pages 103, 113, 114, 143, 404, 405, and 406)

Y. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017. (cited on page 106)

B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015. (cited on page 241)

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, 2001. (cited on pages 382 and 383)

P.-M. Nguyen and H. T. Pham. A rigorous framework for the mean field limit of multilayer neural networks. Technical Report 2001.11443, arXiv, 2020. (cited on page 339)

H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992. (cited on page 18)

A. Nitanda and T. Suzuki. Stochastic particle gradient descent for infinite ensembles. Technical Report 1712.05438, arXiv, 2017. (cited on page 337)

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999. (cited on page 116)

E. Novak. *Deterministic and Stochastic Error Bounds in Numerical Analysis*. Springer, 2006. (cited on pages 407 and 408)

A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, 1962. (cited on page 325)

A. Nowak, F. Bach, and A. Rudi. Sharp analysis of learning with discrete losses. In *International Conference on Artificial Intelligence and Statistics*, pages 1920–1929, 2019. (cited on pages 355 and 356)

A. Nowak-Vila, F. Bach, and A. Rudi. A general theory for structured prediction with smooth convex surrogates. Technical Report 1902.01958, arXiv, 2019. (cited on page 362)

A. Nowak-Vila, F. Bach, and A. Rudi. Consistent structured prediction with Max-Min Margin Markov Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. (cited on pages 365 and 367)

R. I. Oliveira. The lower tail of random quadratic forms, with applications to ordinary least squares and restricted eigenvalue properties. Technical Report 1312.2903, arXiv, 2013. (cited on page 62)

M. R. Osborne, B. Presnell, and B. A. Turlach. On the Lasso and its dual. *Journal of Computational and Graphical statistics*, 9(2):319–337, 2000. (cited on page 220)

A. Osokin, F. Bach, and S. Lacoste-Julien. On structured prediction theory with calibrated convex surrogate losses. In *Advances in Neural Information Processing Systems*, volume 30, 2017. (cited on page 360)

D. Ostrovskii and F. Bach. Finite-sample analysis of M-estimators using self-concordance. *Electronic Journal of Statistics*, 15(1):326–391, 2021a. (cited on page 101)

D. M. Ostrovskii and F. Bach. Finite-sample analysis of M-estimators using self-concordance. *Electronic Journal of Statistics*, 15:326–391, 2021b. (cited on page 225)

J. Palmer, K. Kreutz-Delgado, B. Rao, and D. Wipf. Variational EM algorithms for non-Gaussian latent variable models. In *Advances in Neural Information Processing Systems*, volume 18, 2005. (cited on page 377)

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

(cited on page 213)

S. Pesme and N. Flammarion. Saddle-to-saddle dynamics in diagonal linear networks. *Advances in Neural Information Processing Systems*, 2023. (cited on page 343)

S. Pesme, L. Pillaud-Vivien, and N. Flammarion. Implicit bias of SGD for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34, 2021. (cited on page 327)

L. Pillaud-Vivien, A. Rudi, and F. Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. In *Advances in Neural Information Processing Systems*, pages 8114–8124, 2018. (cited on page 203)

J. Platt. Using analytic QP and sparseness to speed training of support vector machines. *Advances in Neural Information Processing Systems*, 11, 1998. (cited on page 74)

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008. (cited on pages 186 and 256)

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. (cited on pages 181, 182, and 386)

M. Reed and B. Simon. *Methods of Modern Mathematical Physics, Volume 2*. Academic press, 1978. (cited on page 180)

P. Rigollet and A. Tsybakov. Exponential screening and optimal rates of sparse estimation. *The Annals of Statistics*, 39(2):731–771, 2011. (cited on page 388)

P. Rigollet and A. B. Tsybakov. Linear and convex aggregation of density estimators. *Mathematical Methods of Statistics*, 16(3):260–280, 2007. (cited on page 209)

C. P. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*, volume 2. Springer, 2007. (cited on pages 383, 384, and 386)

C. P. Robert and G. Casella. *Monte Carlo statistical methods*, volume 2. Springer, 2005. (cited on page 385)

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997. (cited on pages 123 and 280)

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958. (cited on page 325)

S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5:941–973, 2004. (cited on page 287)

G. M. Rotskoff and E. Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pages 7146–7155, 2018. (cited on page 337)

A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017. (cited on pages 186, 198, and 201)

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015. (cited on pages 186, 198, and 201)

W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987. (cited on pages 243 and 247)

F. Santambrogio. *Optimal Transport for Applied Mathematicians*. Springer, 2015. (cited on page 337)

A. M. Saxe, J. L. McClelland, and S. Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019. (cited on page 343)

R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012. (cited on page 278)

M. Schmidt, N. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *Advances in Neural Information Processing Systems*, 24, 2011. (cited on page 128)

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017. (cited on page 138)

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001. (cited on pages 169 and 188)

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, 2001. (cited on page 171)

D. Scieur, V. Roulet, F. Bach, and A. d'Aspremont. Integration methods and optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 30, 2017. (cited on pages 117 and 322)

Y. Seldin, C. Szepesvári, P. Auer, and Y. Abbasi-Yadkori. Evaluation and analysis of the performance of the EXP3 algorithm in stochastic environments. In *European Workshop on Reinforcement Learning*, 2013. (cited on page 315)

B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. (cited on page 39)

G. Shakhnarovich, T. Darrell, and P. Indyk, editors. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2005. (cited on page 151)

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011. (cited on pages 292 and 315)

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. (cited on page x)

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. (cited on pages 169, 183, 184, and 188)

A. V. Sil'nichenko. Rate of convergence of greedy algorithms. *Mathematical Notes*, 76: 582–586, 2004. (cited on page 282)

B. W. Silverman. Algorithm AS 176: Kernel density estimation using the fast fourier transform. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31 (1):93–99, 1982. (cited on page 153)

J. Sirignano and K. Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020. (cited on page 337)

J. Sirignano and K. Spiliopoulos. Mean field analysis of deep neural networks. *Mathematics of Operations Research*, 2021. (cited on page 339)

A. Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019. (cited on pages 292, 307, and 315)

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012. (cited on page 406)

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(1): 2822–2878, 2018. (cited on page 323)

K. Sridharan, S. Shalev-Shwartz, and N. Srebro. Fast rates for regularized objectives. In *Advances in Neural Information Processing Systems*, 2009. (cited on page 93)

I. Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4(Nov):1071–1105, 2003. (cited on page 74)

G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990. (cited on pages 7 and 19)

C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, pages 595–620, 1977. (cited on page 164)

M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007. (cited on page 97)

E. Suli and D. F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003, 2003. (cited on page 337)

C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012. (cited on pages 360 and 363)

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2018. (cited on page 40)

B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the International Conference on Machine learning*, pages 896–903, 2005. (cited on page 365)

G.-A. Thanei, C. Heinze, and N. Meinshausen. Random projections for large-scale regression. In *Big and Complex Data Analysis*, pages 51–68. Springer, 2017. (cited on page 274)

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. (cited on page 216)

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. (cited on pages 19 and 62)

I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005. (cited on page 365)

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Science & Business Media, 2008. (cited on pages 167 and 401)

A. W. Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000. (cited on pages 99 and 100)

J. E. Van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. (cited on page 39)

V. N. Vapnik and A. Y. Chervonenkis. On a perceptron class. *Automation and Remote Control*, 25:112–120, 1964. (cited on page 72)

V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer, 2015. (cited on pages xi and 70)

S. R. S. Varadhan. *Probability Theory*. American Mathematical Society, 2001. (cited on page 180)

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47. Cambridge University Press, 2018. (cited on page 9)

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990. (cited on page 259)

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019. (cited on pages 85, 86, 226, and 227)

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. (cited on page 358)

S. Wang, A. Gittens, and M. W. Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. *Journal of Machine Learning Research*, 18:1–50, 2018. (cited on page 273)

L. Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006. (cited on page 166)

J. Weston, O. Chapelle, V. Vapnik, A. Elisseeff, and B. Schölkopf. Kernel dependency estimation. *Advances in Neural Information Processing Systems*, 2002. (cited on page 361)

B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673, 2020. (cited on pages 326 and 327)

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. *Advances in Neural Information Processing Systems*, 17, 2004. (cited on page 97)

W.-Y. Yan, U. Helmke, and J. B. Moore. Global analysis of Oja's flow for neural networks. *IEEE Transactions on Neural Networks*, 5(5):674–683, 1994. (cited on page 343)

G. Yang and E. J. Hu. Feature learning in infinite-width neural networks. Technical Report 2011.14522, arXiv, 2020. (cited on page 339)

G. Yang and E. J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737, 2021. (cited on page 261)

Y. Yang. Minimax nonparametric classification. i. rates of convergence. *IEEE Transactions on Information Theory*, 45(7):2271–2284, 1999. (cited on page 392)

J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007. (cited on page 184)

L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, volume 26, 2013. (cited on page 138)

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004a. (cited on page 77)

T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5(Oct):1225–1251, 2004b. (cited on page 350)

T. Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006. (cited on page 96)

T. Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10(3), 2009. (cited on page 213)

T. Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, 57(7):4689–4708, 2011. (cited on page 216)