Learning Theory from First Principles

May 26, 2025

Francis Bach

francis.bach@inria.fr

Contents

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, ranging 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 greater 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 also to understand how quickly (or slowly) they learn, with a particular eye toward adaptivity to specific structures that make learning faster (such as smoothness of the prediction functions or dependence on low-dimensional subspaces).

This book is geared toward 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 and 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 the readers have 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, chapter 5 is dedicated to that topic. For most learning methods, some simple **illustrative experiments** are presented, with accompanying code (MATLAB and Python for the moment, and Julia in the future) so students can see for themselves that the algorithms are simple and effective in synthetic experiments. Exercises currently come without solutions and are meant to help students understand the related material.

Finally, the third part of the book provides an in-depth discussion of **modern 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 and their relation with machine learning.

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

Book organization. The book comprises three main parts: an introduction, a core part, and special topics. Readers are encouraged to read the first two parts to understand the main concepts fully and can pick and choose among 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 the simulation experiments are available at `https://www.di.ens.fr/~fbach/ltfp/` as MATLAB and Python code. Many exercises are proposed and are embedded in the text with dedicated paragraphs, with a few mentioned within the text (e.g., as "proof left as an exercise"). These exercises are meant to deepen the understanding of the nearby material, by proposing extensions or applications.

Sections or more advanced exercises are denoted by ♦, ♦♦, or ♦♦♦, with the number of diamonds denoting the level of complexity. Comments or suggestions are most welcome and should be sent to `francis.bach@inria.fr`.

Many topics are not covered at all, and many others are not covered in depth. There are many good textbooks on learning theory that go deeper or wider (e.g., 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 base my 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 for 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 chapter 12 on overparameterized models and chapter 13 on structured prediction, which present modern topics in machine learning. More generally, the goal in the third part of the book (special topics) was, for each chapter, to introduce new concepts, while remaining a few steps away from the core material and using unified notations.

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 have any elegant and simple ideas I have overlooked.

Mathematical notations. Throughout the textbook, I provide unified notations:

- Random variables: given a set $\mathcal{X}$, we will use the lowercase notation for a random variable with values in $\mathcal{X}$, as well as for its observations. Probability distributions will be denoted μ 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 overformalizations. For a detailed treatment, see Devroye et al. (1996) and Christmann and Steinwart (2008).
- Norms on $\mathbb{R}^d$: we will consider the usual ℓ_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 semidefinite (i.e., all of its eigenvalues are nonnegative), and for two symmetric matrices A and B, $A \succcurlyeq B$ means that $A - B \succcurlyeq 0$. For a vector $\lambda \in \mathbb{R}^n$, $\mathrm{Diag}(\lambda)$ is the diagonal matrix with diagonal vector λ.

[1] http://www.mit.edu/~rakhlin/notes.html.

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

- 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 as $f''(x) \in \mathbb{R}^{d \times d}$.

How to use this book? The first nine 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 six 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., two-semester) class. The book is intended to be adapted to self-study, with the first nine chapters being read in sequence and the last six in random order. In all situations, chapter 1, on mathematical preliminaries, can be read quickly and studied in more detail when relevant notions are needed in subsequent chapters.

Acknowledgments. 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 extra passes during the classes I taught in the Spring 2021, Fall 2021, Fall 2022, and Fall 2023 semesters.

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 with chapter 9 on neural networks and for proofreading 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 chapter 5 on optimization, to Marc Lelarge for his help on overparameterized models, Olivier Cappé for his help on multiarmed bandits, and Lawrence Stewart for his help on neural network architectures. The notes from Philippe Rigollet have been a very precious help for chapter 8 on model selection. The careful readings of large portions of the text by Bertille Follain and Gabriel Stoltz have been very helpful. Feedback from the anonymous reviewers has also been useful.

Former and current collaborators also helped in the final stages by reading carefully, annotating, and commenting a chapter: Eloïse Berthier, Raphaël Berthier, Vivien Cabannes, Aymeric Dieuleveut, Nicolas Flammarion, Pierre Gaillard, Hadrien Hendrickx, David Holzmüller, Dmitrii Ostrovskii, Loucas Pillaud-Vivien, Alessandro Rudi, Kevin Scaman, and Adrien Taylor. This was greatly appreciated.

Typos and suggestions have been highlighted 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, Leon Rofagha, John Zarka, Liviu Aolaritei, Gaétan Marceau Caron, Ivan Barrientos, Thomas Boudou, Sebastian Gruber, Julien Stoehr, Jingxin Zhang, Sacha Braun, Noâm Boussouf, Abderrahmane Kasmi, Jacques Sun, Sebastiano Scardera, Mariem Aalabou, Pierre Cornilleau, Eric Moulines, Alexandre Olech, Nabil Kahalé, Aaron Mishkin, Patrik Wolf, Jan Quan (⚠ Add your name to the list by sending me typos and comments!).

I am grateful to Elizabeth Swayze, Matthew Valades, Susan McClung, Roger Wood, Emma Donovan, Jitendra Kumar, and everyone at MIT Press for their assistance in preparing and publishing this book.

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 the 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 nonelementary mathematical tools used throughout the book: first, linear algebra tricks, and then concentration inequalities. The chapter can be safely skipped for readers familiar with linear algebra and concentration inequalities since the 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 usually greatly simplifies 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 vector $b \in \mathbb{R}^n$, the minimization of quadratic forms with linear terms can be done in closed form:

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

with the 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 were not invertible (simply positive semidefinite) and b were 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 it 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 2×2 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 2×2 matrix. If $ad - bc \neq 0$, then we may invert it as follows:

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

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

1.1.3 Inverting Matrices Defined by Blocks, Matrix Inversion Lemma

The example given 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] 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 rows (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 block A and also assume that M/A is invertible.

[1] For a discussion of this term, 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.

We thus get by Gaussian elimination, referring to L_i, $i = 1, 2$ as the two lines of blocks, so for the first matrix $M = \binom{L_1}{L_2}$:

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

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 first setting the upper-right block to zero, and assuming that 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 equations (1.1) and (1.2), we obtain the following 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

$$(I + BB^\top)^{-1} = I - B(I + B^\top B)^{-1}B^\top, \tag{1.3}$$

and, once right-multiplied by B, this leads to the following compact formula (which is easier to rederive and remember than equation (1.3)):

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

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

Exercise 1.1 *For* $\alpha \in \mathbb{R}$ *such that* $\alpha \neq -1/n$ *and* $1_n \in \mathbb{R}^n$ *the vector of all* 1*s, show that we have* $(I + \alpha 1_n 1_n^\top)^{-1} = I - \frac{\alpha}{1+n\alpha} 1_n 1_n^\top$.

Exercise 1.2 (♦) *Show that we can block-diagonalize the matrices* M *and* M^{-1} *as*

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

Exercise 1.3 *Show that* $\det\left(\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)\right) = \det(M/A)\det(A) = \det(M/D)\det(D)$.

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

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

By writing it as the product of a function of x and a function of (x,y), we can get that x is Gaussian with mean μ_x and covariance matrix Σ_x, and that given $x = x'$, y is Gaussian with mean $\mu_{y|x'} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x' - \mu_x)$ (which depends on x') and covariance matrix $\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$ (which does not depend on x').

Exercise 1.4 (♦) *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 are an orthogonal matrix $U \in \mathbb{R}^{n\times n}$ (i.e., 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 ith 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 semidefinite if and only if all its eigenvalues are nonnegative.

Given a rectangular matrix $X \in \mathbb{R}^{n\times d}$, such that $n \geqslant d$, there are an orthogonal matrix $V \in \mathbb{R}^{d\times d}$ (i.e., such that $V^\top V = VV^\top = I$), a matrix $U \in \mathbb{R}^{n\times d}$ with orthonormal columns (i.e., such that $U^\top U = I$, but $UU^\top \neq I$ if $n > d$), and 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 ith 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 $\begin{pmatrix} 0 & X \\ X^\top & 0 \end{pmatrix}$ 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 1.5 *Express the eigenvectors of $XX^\top$ and $X^\top X$ using the singular vectors of X.*

Exercise 1.6 *Express the eigenvectors of $\begin{pmatrix} 0 & X \\ X^\top & 0 \end{pmatrix}$ using the singular vectors of X.*

1.1.5 Differential Calculus

Throughout this 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 $F'(\theta) = \frac{1}{2}(A + A^\top)\theta - b$ 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 1.7 *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 $\sigma(u) = (1 + e^{-u})^{-1}$ 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 $h_i = \sigma(y_i(X\theta)_i)\sigma(-y_i(X\theta)_i)$.*

Exercise 1.8 (Functions on matrices) *Let A be a symmetric matrix. Show that the gradient of the function $M \mapsto \mathrm{tr}(AM^{-1})$, defined on invertible symmetric matrices, is equal to $M \mapsto -M^{-1}AM^{-1}$. Show that the gradient of $M \mapsto \log\det(M)$ is $M \mapsto M^{-1}$.*

1.2 Concentration Inequalities

All the results presented in this textbook rely on the simple probabilistic assumption that data are independently and identically distributed (i.i.d.). The primary goal, then, is 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 $Z_1, \ldots, Z_n \in \mathbb{R}$ are i.i.d. with variance

$\sigma^2 = \mathbb{E}\big[(Z - \mathbb{E}[Z])^2\big]$, then the variance of the sum is the sum of the variances, and

$$\operatorname{var}\left[\frac{1}{n}\sum_{i=1}^{n} Z_i\right] = \frac{1}{n^2}\sum_{i=1}^{n}\operatorname{var}[Z_i] = \frac{\sigma^2}{n}.$$

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

The equality shown above can be interpreted as

$$\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbb{E}[Z]\right)^2\right] = \frac{\sigma^2}{n}, \tag{1.4}$$

which provides the simplest proof of the law of large numbers when variances exist and also highlights the convergence in the 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 inequality on the moments of a random variable, deviation bounds can be derived. Markov's inequality (see the proof in exercise 1.9 below) states that

$$\mathbb{P}(Y \geqslant \varepsilon) \leqslant \frac{1}{\varepsilon}\mathbb{E}[Y], \tag{1.5}$$

for all nonnegative random variables 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 the random variable X with finite mean $\mathbb{E}[X]$ and variance $\operatorname{var}[X]$, leading to

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

Thus, from the mean $\mathbb{E}[Z]$ and the variance $\frac{\sigma^2}{n}$ of the random variable $\frac{1}{n}\sum_{i=1}^{n} Z_i$, as computed in equation (1.4), we obtain the deviation bounds

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

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 $\mathbb{E}[Z]$

[4] See https://en.wikipedia.org/wiki/Convergence_of_random_variables for a discussion on convergence of random variables.

and variance σ^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 σ^2. Although it gives the correct scaling in n, in this textbook, we will look primarily at nonasymptotic results that quantify the deviation for any n.

In what follows, we will always provide versions of inequalities for *averages* of random variables (even though 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 other sources, such as Boucheron et al. (2013) and 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 θ. Typically, these units are encapsulated in the constants describing the problem (such as the noise standard deviation for y or bounds for x and θ).

Exercise 1.9 *Let Y be a nonnegative random variable with finite expectation, and $\varepsilon > 0$. Show that $\varepsilon 1_{Y\geqslant\varepsilon} \leqslant Y$ almost surely and prove Markov's inequality in equation (1.5).*

Exercise 1.10 (Chernoff bound) *Let X be a random variable. Show that for any $t \in \mathbb{R}$ and $s > 0$, we have $\mathbb{P}(X \geqslant t) \leqslant e^{-st}\mathbb{E}[e^{sX}]$.*

Exercise 1.11 *Let Y be a nonnegative random variable with finite expectation. Show that $\mathbb{E}[Y] = \int_0^\infty \mathbb{P}(Y \geqslant t)dt$.*

Jensen's inequality. Beyond the union bound, another key tool in probabilistic modeling is Jensen's inequality, which allows to obtain bounds for the expectation of convex functions of random variables (see extension to functions defined on $\mathbb{R}^d$ in section 5.2.2).

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

Proposition 1.1 (Jensen's inequality on $\mathbb{R}$) *If $F : \mathbb{R} \to \mathbb{R}$ is convex and X is a real-valued random variable, then*

$$F\big(\mathbb{E}[X]\big) \leqslant \mathbb{E}\big[F(X)\big]. \tag{1.6}$$

Stated in words: "The image of the average is smaller than the average of the images."

⚠ When using Jensen's inequality, be extra careful about the direction of the inequality.

1.2.1 Hoeffding's Inequality

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

Proposition 1.2 (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 $t \geqslant 0$,*

$$\mathbb{P}\bigg(\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i] \geqslant t\bigg) \leqslant \exp(-2nt^2). \tag{1.7}$$

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

(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 the function φ defined as $\varphi(s) = \log(\mathbb{E}\big[\exp(s(Z - \mathbb{E}[Z]))\big])$, which is a "log-sum-exp" function, often referred to as the "cumulant generating function." We can compute the derivatives of φ as

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

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 $z \mapsto e^{s(z-\mathbb{E}[Z])}/\mathbb{E}\big[e^{s(Z-\mathbb{E}[Z])}\big]$ with respect to 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, $\varphi(s) \leqslant \varphi(0) + \varphi'(0)s + \frac{1}{4} \cdot \frac{s^2}{2} = \frac{s^2}{8}$.

(2) For any $t \geqslant 0$, and denoting $\bar{Z} = \frac{1}{n}\sum_{i=1}^n Z_i$, we get:

$$\begin{aligned}
& \mathbb{P}\big(\bar{Z} - \mathbb{E}[\bar{Z}] \geqslant t\big) \\
= \ & \mathbb{P}\big(\exp(s(\bar{Z} - \mathbb{E}[\bar{Z}])) \geqslant \exp(st)\big) \text{ by monotonicity of the exponential,} \\
\leqslant \ & \exp(-st)\mathbb{E}\big[\exp(s(\bar{Z} - \mathbb{E}[\bar{Z}]))\big] \text{ using Markov's inequality (equation (1.5)).}
\end{aligned}$$

Then, using independence, we get

$$\begin{aligned}\mathbb{P}\big(\bar{Z}-\mathbb{E}[\bar{Z}]\geqslant t\big) &\leqslant \exp(-st)\prod_{i=1}^n \mathbb{E}\Big[\exp\Big(\frac{s}{n}(Z_i-\mathbb{E}[Z_i])\Big)\Big]\\ &\leqslant \exp(-st)\prod_{i=1}^n \exp\Big(\frac{s^2}{8n^2}\Big) = \exp\Big(-st+\frac{s^2}{8n}\Big),\end{aligned}$$

using the lemma at the beginning of the proof. This last bound 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 equation (1.7) 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 $\sigma^2 = \lim_{n\to+\infty}\frac{1}{n}\sum_{i=1}^n \mathrm{var}(Z_i)$ (see exercise 1.12). The central limit theorem is more precise (as it involves the variance of Z_i and not an almost sure bound) but is asymptotic. Bernstein's inequalities (see section 1.2.3) will be between the central limit theorem and Hoeffding's inequality, as they use both the variance and an almost sure bound.

Exercise 1.12 (♦) *For X a Gaussian random variable with mean 0 and variance 1, show that for $t \geqslant 0$, $\frac{1}{4}\exp(-t^2) \leqslant \mathbb{P}(X \geqslant t) \leqslant \exp(-t^2/2)$.*

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 $t \geqslant 0$,*

$$P\bigg(\bigg|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\bigg| \geqslant t\bigg) \leqslant 2\exp(-2nt^2). \tag{1.8}$$

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

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

- 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

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

Note the dependence in n is $1/\sqrt{n}$ while the dependence in δ is logarithmic (corresponding to the exponential tail bound in t).

Exercise 1.13 *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 exercise 1.14).

Exercise 1.14 (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*

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

- Hoeffding's inequality is often extended 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[7] of X holds:

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

which is exactly what we used in the proof of proposition 1.2. 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. For example, we have the usual two versions of the tail bound (see also exercise 1.16):

$$\forall t \geqslant 0,\ \mathbb{P}(|Z-\mathbb{E}[Z]| \geqslant t) \leqslant 2\exp\Big(-\frac{t^2}{2\tau^2}\Big) \tag{1.9}$$

$$\Leftrightarrow\quad \forall \delta \in (0,1],\ |Z-\mathbb{E}[Z]| \leqslant \tau\sqrt{2\log\Big(\frac{2}{\delta}\Big)} \text{ with probability } 1-\delta. \tag{1.10}$$

Exercise 1.15 *Show that a Gaussian random variable with variance* σ^2 *is sub-Gaussian with constant* σ^2.

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

[7] See https://en.wikipedia.org/wiki/Laplace_transform.

Exercise 1.16 *If $Z_1, \dots, Z_n$ are independent random variables which are sub-Gaussian with constant τ^2, show that $\mathbb{P}\big(\big|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\big| \geqslant t\big) \leqslant 2\exp(-\frac{nt^2}{2\tau^2})$ for any $t \geqslant 0$.*

- Sub-Gaussian random variables can be defined in several other ways, equivalent (up to constants) to the bound on the Laplace transform. See exercises 1.17 and 1.18.

Exercise 1.17 (♦) *Let Z be a random variable that is sub-Gaussian with constant τ^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 equation (1.9), show that for any positive integer q, $\mathbb{E}[(Z - \mathbb{E}[Z])^{2q}] \leqslant 2 \cdot q!(2\tau^2)^q$.*

Exercise 1.18 (♦♦) *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 1.19 *Assume that the random variable Z has almost surely nonnegative values and finite second-order moment. Show that for any $s \geqslant 0$, we have $\log\big(\mathbb{E}[e^{-sZ}]\big) \leqslant -s\mathbb{E}[Z] + \frac{s^2}{2}\mathbb{E}[Z^2]$.*

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 1.3 (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 $i \in \{1, \dots, n\}$, and all $z_1, \dots, z_n, z_i' \in \mathcal{Z}$, we have*

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

Then

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

Proof (♦) The proof generalizes the formulation of Hoeffding's inequality in equation (1.8), which corresponds to $f(z) = \frac{1}{n}\sum_{i=1}^n z_i$ and $c = \frac{1}{n}$. We will only consider the one-sided inequality:

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

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

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

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

with $V_1 = \mathbb{E}[f(Z_1, \dots, Z_n)|Z_1] - \mathbb{E}[f(Z_1, \dots, Z_n)]$. We have $\mathbb{E}[V_i|Z_1, \dots, Z_{i-1}] = 0$. Moreover, given $Z_1, \dots, Z_{i-1}$, 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 ith variable. Moreover, through a telescoping sum, we have $f(Z_1, \dots, Z_n) - \mathbb{E}[f(Z_1, \dots, Z_n)] = \sum_{i=1}^n V_i$. Using the same argument as in part (1) of the proof of Hoeffding's inequality (page 10), we get for any $s > 0$, $\mathbb{E}\big[e^{sV_i} | Z_1, \dots, Z_{i-1}\big] \leqslant e^{s^2c^2/8}$, and we can obtain a proof with the same steps as part (2) of the same proof (page 10) by being careful with conditioning, for any $s \geqslant 0$:

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

using the bound on $\mathbb{E}\big[e^{sV_n} | Z_1, \dots, Z_{n-1}\big]$ given above. Applying the same reasoning n times, we get a probability that is 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 1.20 (♦) *Use McDiarmid's inequality to prove a Hoeffding-type bound for vectors: 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*

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

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 uses (but only with an asymptotic result). Bernstein's inequality allows the use of variance to get a finer nonasymptotic result.

Proposition 1.4 (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 $t \geqslant 0$,*

$$\mathbb{P}\bigg(\bigg|\frac{1}{n}\sum_{i=1}^n Z_i\bigg| \geqslant t\bigg) \leqslant 2\exp\Big(-\frac{nt^2}{2\sigma^2 + 2ct/3}\Big), \tag{1.11}$$

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

$$\Big|\frac{1}{n}\sum_{i=1}^n Z_i\Big| \leqslant \sqrt{\frac{2\sigma^2\log(2/\delta)}{n}} + \frac{2c\log(2/\delta)}{3n}. \tag{1.12}$$

Proof The proof is also divided into two parts, first with 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

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

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

(b) With $\sigma_i^2 = \mathrm{var}(Z_i)$, we have the following 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 stated at the beginning of the proof. We now need to find an upper bound on the minimal value (with respect to s) of $-nst + \frac{n\sigma^2}{c^2}(e^{sc}-1-sc) = \frac{n\sigma^2}{c^2}(e^{sc}-1-sc-\alpha sc)$, 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 $u = \frac{\alpha}{1+\alpha/3}$ (which leads to a candidate $s = u/c$), we get $1 - u/3 = 1 - \frac{\alpha/3}{1+\alpha/3} = \frac{3}{\alpha+3}$, 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 equation (1.11).

To get equation (1.12) from the two-sided version of equation (1.11), we solve in t the equation $2\exp\left(\frac{-nt^2}{2\sigma^2+2ct/3}\right) = \delta \Leftrightarrow \log\frac{2}{\delta} = \frac{nt^2}{2\sigma^2+2ct/3}$. 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 equation (1.12).

■

Note here that we get the same dependence as for the central limit theorem for small deviations t (and a strict improvement on Hoeffding's inequality 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 equation (1.11) is replaced by

$$\mathbb{P}\bigg(\bigg|\frac{1}{n}\sum_{i=1}^n Z_i - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[Z_i]\bigg| \geqslant t\bigg) \leqslant 2\exp\Big(-\frac{nt^2}{2\sigma^2+2ct/3}\Big), \tag{1.13}$$

with a corresponding single-sided inequality.

Exercise 1.21 (♦) *Prove the inequality in equation (1.13).*

1.2.4 Expectation of the Maximum

Concentration inequalities bound the deviation from the expectation. Often, computing the expectation is tricky, particularly for the 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 to the expectation of a maximum and to the Gaussian decay of tail bounds.

⚠ The variables do not need to be independent.

Proposition 1.5 (Expectation of the maximum) *If $Z_1, \dots, Z_n$ are (potentially dependent) zero-mean real random variables that are sub-Gaussian with constant τ^2, then*

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

Proof We have

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

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 earlier in this discussion, 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 random variables $U_1,\dots,U_n$. In other words, the dependence in the probability δ as $\sqrt{\log(\frac{2}{\delta})}$ in equation (1.10) is directly related to the term $\sqrt{\log n}$ (see exercise 1.22). We will see a different dependence in n in section 8.1.2 for the maximum of the squared norms of Gaussians.

Exercise 1.22 *Assume that $Z_1,\dots,Z_n$ are random variables that are sub-Gaussian with constant τ^2 and have zero means. Show that $\mathbb{E}\big[\max\{|Z_1|,\dots,|Z_n|\}\big] \leqslant \sqrt{2\tau^2\log(2n)}$. Prove 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 1.23 (♦♦) *Assume that $Z_1,\dots,Z_n$ are independent Gaussian random variables with mean zero and variance σ^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 1.24 *Assume that $Z_1,\dots,Z_n$ are sub-Gaussian random variables with common sub-Gaussianity parameter τ, and potentially different means $\mu_1,\dots,\mu_n$. For a fixed set of nonnegative weights $\pi_1,\dots,\pi_n$ that sum to 1, 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 $\hat{\imath}\in\arg\min_{i\in\{1,\dots,n\}}\big\{z_i+\tau\sqrt{2\log(1/\pi_i)}\big\}$, show that with probability greater than $1-\delta$, $\mu_{\hat{\imath}}\leqslant\min_{i\in\{1,\dots,n\}}\big\{\mu_i+2\tau\sqrt{2\log(1/\pi_i)}\big\}+2\tau\sqrt{2\log(2/\delta)}$.*

Exercise 1.25 (♦♦) *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 Ω; 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*

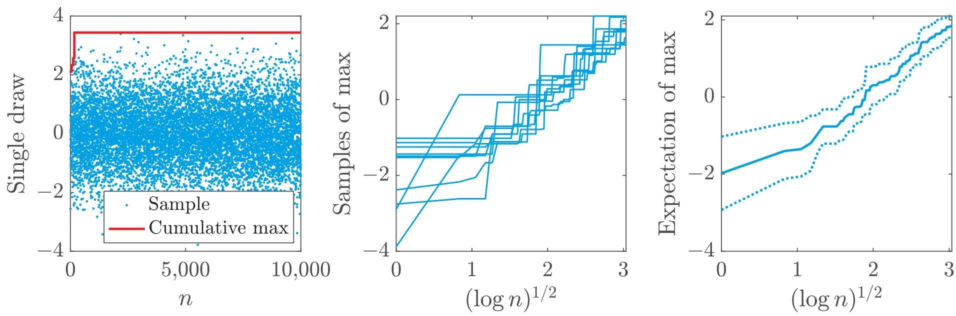


Figure 1.1. Expectation of the maximum of n independent standard Gaussian random variables. Left: illustration of the cumulative maximum $\max\{Z_1, \dots, Z_n\}$. Middle: 10 samples of the cumulative maximum as a function of $\sqrt{\log n}$. Right: mean and standard deviations from 1,000 replications. Notice the linear growth in $\sqrt{\log n}$, which is compatible with our bounds.

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* $\mathbb{E}[f(Z_1 + \cdots + Z_n)] \leqslant nL\frac{\sigma^2}{2}$.

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 i.i.d. samples, with a convergence rate of $O(1/\sqrt{n})$ from n samples (as shown, e.g., from Hoeffding's inequality).

In this section, we briefly present *quadrature* methods whose aim is to estimate the same expectation, but with potentially nonrandom observations. For simplicity, we consider a random variable X uniformly distributed in $[0, 1]$, and the task of computing the expectation of a function $f : [0, 1] \to \mathbb{R}$ (i.e., $I = \mathbb{E}[f(X)] = \int_0^1 f(x)dx$), 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, so-called "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, particularly in chapters 6 and 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

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

It corresponds to approximating f by its piecewise interpolant on $[x_{i-1}, x_i]$ based on values at $\{x_{i-1}, x_i\}$ (see proof in exercise 1.26).

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; see exercise 1.26):

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

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 exercises 1.27 and 1.28.

Exercise 1.26 *Consider a function $g : [0,1] \to \mathbb{R}$. Show that the piecewise interpolant based on values at $\{0,1\}$ equals $\tilde{g} : x \mapsto (1-x)g(0) + xg(1)$ and that its integral equals $\frac{1}{2}g(0) + \frac{1}{2}g(1)$. Assuming g is twice differentiable with second-derivative bounded in magnitude by L, show that for all $x \in [0,1]$, $|g(x) - \tilde{g}(x)| \leqslant \frac{L}{2} x(1-x)$.*

Exercise 1.27 *Show that the trapezoidal rule leads to an error in $O(1/n)$ if we assume only one bounded derivative.*

Exercise 1.28 (♦) *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 Random Matrices (♦♦)

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

Proposition 1.6 (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, with $\sigma^2 = \lambda_{\max}\big(\frac{1}{n}\sum_{i=1}^{n} C_i^2\big)$. Then for all $t \geqslant 0$,*

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

Proposition 1.7 (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, with $\sigma^2 = \lambda_{\max}\big(\frac{1}{n}\sum_{i=1}^n \mathbb{E}[M_i^2]\big)$. Then for all $t \geqslant 0$,*

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

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 also apply to rectangular matrices $M_i \in \mathbb{R}^{d_1\times d_2}$ by considering the symmetric matrices $\widetilde{M}_i = \left(\begin{smallmatrix} 0 & M_i \\ M_i^\top & 0 \end{smallmatrix}\right) \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$, 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 1.29 *Assume that the matrices $M_i \in \mathbb{R}^{d_1\times d_2}$ are independent, have zero mean, and $\|M_i\|_{\mathrm{op}} \leqslant c$ almost surely for all $i \in \{1,\dots,n\}$. Show that*

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

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

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

Infinite-dimensional covariance operators (♦♦). As used within chapter 7, we will need to extend the results given 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, equation (3.9)):

Proposition 1.8 (Matrix Bernstein bound–intrinsic dimension) *Given n independent random bounded self-adjoint operators M_i on a Hilbert space, such that for all $i \in \{1,\dots,n\}$, $\mathbb{E}[M_i]=0$, $\lambda_{\max}(M_i) \leqslant c$ almost surely, and $\frac{1}{n}\sum_{i=1}^n \mathbb{E}[M_i^2] \preccurlyeq V$. Then for all $t \geqslant 0$,*

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

for $\sigma^2 \geqslant \lambda_{\max}(V)$ and $d = \frac{\mathrm{tr}(V)}{\sigma^2}$. 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.

In this chapter, we present the supervised learning problem, which is the main object of study in this book. After a short introduction highlighting the main motivating practical examples in section 2.1, the decision-theoretic probabilistic framework set forth in section 2.2 provides the traditional mathematical formalization, with the notions of loss, risk, and optimal predictor. This will precisely define the goals and evaluation standards of machine learning that will be applied to the learning algorithms presented throughout this book. Section 2.3 then presents the two main classes of learning algorithms: local averaging techniques, and methods based on empirical risk minimization. Notions of statistical consistencies are described in section 2.4; studying the consistency of learning methods will be our main objective in this book: as shown in section 2.5 on "no free lunch" theorems, no method can perform uniformly well, and assumptions have to be made to obtain meaningful quantitative results, as shown in section 2.6. We conclude this introductory chapter by presenting in section 2.7 classical extensions to the basic supervised learning frameworks, and, in section 2.8, a summary and an outline of the subsequent chapters of this 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 exists in statistics). At the same time, the emphasis on models and their interpretability beyond their predictive performance is more prominent within statistics (but also exists in machine learning). See also the discussion in section 4.7.

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 documents, 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 some 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 (1) common topologies are encountered in many diverse areas (such as sequences or two-dimensional or three-dimensional objects), and thus common tools are used, and (2) learning these representations is an active area of research (see discussions in chapters 7 and 9).

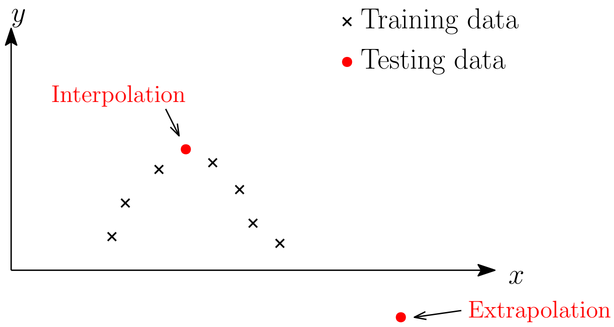
  In this textbook, we will primarily consider that inputs are d-dimensional vectors, with d potentially large (up to 10^6 or 10^9).

- **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 that we examine in most of the book. Note, however, that most of the concepts extend to the more general *structured prediction* setup, where more general structured outputs (e.g., graph prediction, visual scene analysis, source separation, ranking) 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 a random 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 ε, 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 (i.e., $y = f(x, z)$, with z random and not observed, which is common, e.g., in medical applications, where we need to predict a future occurrence of a disease based on limited information about patients).

- The prediction function f may be quite complex, highly nonlinear 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 the following diagram 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.

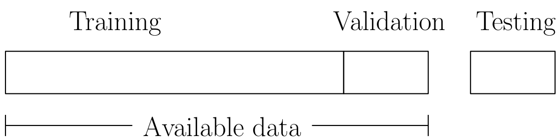
- The input space $\mathcal{X}$ may be very large (i.e., with high dimension when this is a vector space). This leads to both computational issues (scalability) and statistical issues (generalization to unseen data). One usually refers to this problem as the *curse of dimensionality.*
- There may be 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 book, 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 then 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 be used only 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 into k folds (typically $k = 5$ or 10), and all models are estimated k times, each time choosing a different fold as validation data (see the 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 with 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 hyperparameters (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 thus far. It is also possible to use penalization techniques based on generalization bounds. These two approaches are analyzed in section 4.6.

Random design versus fixed design. What we have described is often referred to as the "random design" setup in statistics, where both x and y are assumed to be 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" (i.e., 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 concepts 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 measurability issues on purpose. The interested reader can look at Christmann and Steinwart (2008) for a more formal presentation.

2.2.1 Supervised Learning Problems and 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 of the loss.

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

The loss function only concerns the output space $\mathcal{Y}$ independent of the input space $\mathcal{X}$. The main examples are as follows, each corresponding to a particular supervised learning problem (note that for each problem, different losses may be considered):

- **Binary classification:** $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$, or, less often, when seen as a subcase of the multicategory situation below, $\mathcal{Y} = \{1, 2\}$); the "0–1 loss" defined as $\ell(y, z) = 1_{y \neq z}$ is the most commonly used; 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 $\mathcal{Y} = \{-1, 1\}$: double-check which convention is used when using toolboxes.

- **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 three examples above, there are many practical problems where $\mathcal{Y}$ is more complicated, with associated algorithms and theoretical results. For example, when $\mathcal{Y} = \{0, 1\}^k$ (leading to multilabel 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 for a detailed treatment.

Throughout this 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 ℓ in this book.

⚠ While the loss function ℓ will be used to define the generalization performance in section 2.2.2, 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 error*, 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, 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 expected and empirical 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 (error rate), while the empirical risk is the proportion of mistakes on the training data.

  In practice, the *accuracy*, which is 1 minus the error rate, is often reported.

- **Multicategory 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 (error rate).

- **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'.

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

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

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 next.

⚠ 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 (1) the Bayes predictor is not always unique, but that all lead to the same Bayes risk (e.g., in binary classification when $\mathbb{P}(y = 1|x) = 1/2$); and (2) that the Bayes risk is usually nonzero (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 nonnegative).*

Therefore, machine learning could be seen trivial: *given* the distribution $y|x$ for any x, the optimal predictor is known and given by equation (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 as follows:

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

$$
\begin{aligned}
f_*(x') \in \mathop{\arg\min}_{z \in \{-1,1\}} \ \mathbb{P}(y \neq z|x = x') \quad &= \quad \mathop{\arg\min}_{z \in \{-1,1\}} \ 1 - \mathbb{P}(y = z|x = x') \\
&= \quad \mathop{\arg\max}_{z \in \{-1,1\}} \ \mathbb{P}(y = z|x = x').
\end{aligned}
$$

  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 $\eta(x') < 1/2$, $f_*(x') = -1$. What happens for $\eta(x') = 1/2$ is irrelevant, as the expected error is the same for the two potential predictions.

  The Bayes risk is then equal to $\mathcal{R}^* = \mathbb{E}\big[\min\{\eta(x), 1 - \eta(x)\}\big]$, which in general is 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 $\mathcal{Y} = \{1, \ldots, k\}$, for $k \geqslant 2$, where we have $f_*(x') \in \mathop{\arg\max}_{i \in \{1,\ldots,k\}} \mathbb{P}(y = i|x = x')$.

  ⚠ These Bayes predictors and risks are valid only 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 2.1 and chapter 13).

- **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 $f_*(x') = \mathbb{E}[y|x = x']$, with a Bayes risk equal to the expected conditional variance.

Exercise 2.1 *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 *We consider a learning problem on* $\mathcal{X} \times \mathcal{Y}$, *with* $\mathcal{Y} = \mathbb{R}$ *and the absolute loss defined as* $\ell(y, z) = |y - z|$. *Compute a Bayes predictor* $f_* : \mathcal{X} \to \mathbb{R}$.

Exercise 2.3 *We consider a learning problem on* $\mathcal{X} \times \mathcal{Y}$, *with* $\mathcal{Y} = \mathbb{R}$ *and the "pinball" loss* $\ell(y, z) = \alpha(y - z)_+ + (1 - \alpha)(z - y)_+$, *for* $\alpha \in (0, 1)$. *Compute a Bayes predictor* $f_* : \mathcal{X} \to \mathbb{R}$. *Provide an interpretation in terms of quantiles.*

Exercise 2.4 (♦) *Characterize Bayes predictors for regression with the "*ε*-insensitive" loss defined as* $\ell(y, z) = \max\{0, |y - z| - \varepsilon\}$. *If for each* x, y *is supported in an interval of length less than* 2ε, *what are the Bayes predictors?*

Exercise 2.5 (Inverting predictions) *Consider the binary classification problem with* $\mathcal{Y} = \{-1, 1\}$ *and the 0–1 loss. Relate the risk of a prediction* f *to that of its opposite* $-f$.

Exercise 2.6 ("Chance" predictions) *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 independent of input* x*? Address the same question with multiple categories.*

Exercise 2.7 (♦) *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 section 2.2, with notations summarized in table 2.1, 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.

[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.

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

$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space
p	Joint distribution on $\mathcal{X} \times \mathcal{Y}$
$(x_1, y_1, \ldots, x_n, y_n)$	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
$f_*(x') = \arg\min_{z\in\mathcal{Y}} \mathbb{E}[\ell(y, z)\|x = x']$	Bayes prediction at x'
$\mathcal{R}^* = \mathbb{E}_{x'\sim p} \inf_{z\in\mathcal{Y}} \mathbb{E}[\ell(y, z)\|x = x']$	Bayes risk

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, and 13).

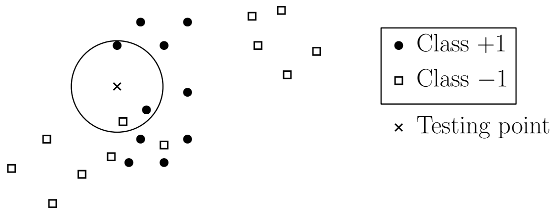
Note that there are prediction algorithms that do not fit precisely into one of these two categories, such as boosting or ensemble classifiers (which perform several empirical risk minimizations, in series or parallel, see chapter 10). Moreover, some situations do not fit the classical i.i.d. framework,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

2.3.1 Local Averaging

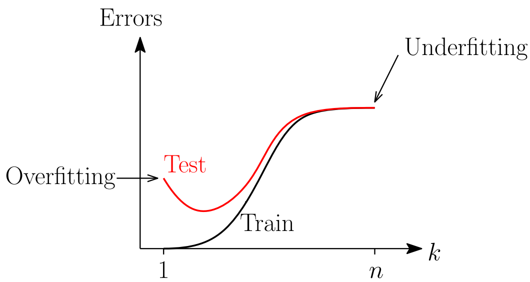
The goal here is to approximate/emulate the Bayes predictor (e.g., $f_*(x') = \mathbb{E}[y|x = x']$ for least-squares regression, or $f_*(x') = \arg\max_{z\in\mathcal{Y}} \mathbb{P}(y = z|x = x')$ for classification with the 0–1 loss) from empirical data. This is often done by explicit or implicit estimation of the conditional distribution by *local averaging* (k-nearest neighbors, which is used as the primary example for this chapter; Nadaraya-Watson estimators; or decision trees). We briefly outline here the main properties for one instance of these algorithms; see chapter 6 for details.

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

We consider the 3-nearest-neighbor classifier on a particular testing point (which will be predicted as 1):



- Pros: (1) no optimization or training, (2) often easy to implement, and (3) can get very good performance in low dimensions (in particular for nonlinear dependences between x and y).
- Cons: (1) slow at query time: must pass through all training data at each testing point (there are algorithmic tools to reduce complexity; see chapter 6); (2) bad for high-dimensional data (because of the curse of dimensionality; more on this in chapter 6); (3) the choice of local distance function is crucial; and (4) 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 2.8 *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). This class of learning methods aims at minimizing the empirical risk with respect to $\theta \in \Theta$:

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

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

The most classic example is linear least-squares regression (studied thoroughly in chapter 3), where we minimize $\frac{1}{n}\sum_{i=1}^{n}(y_i - \theta^\top \varphi(x_i))^2$, and 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: (1) can be relatively easy to optimize (e.g., least-squares with its simple derivation and numerical algebra; see chapter 3), many algorithms are available (primarily based on gradient descent; see chapter 5); and (2) can be applied in any dimension (if a suitable feature vector is available).
- Cons: (1) can be relatively hard to optimize when the optimization formulation is not convex (e.g., neural networks); (2) need a suitable feature vector for linear methods; (3) the dependence on parameters can be complex (e.g., neural networks); (4) need some capacity control to avoid overfitting; and (5) require 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.

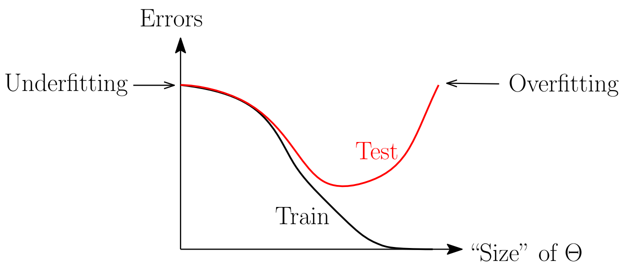
- Risk decomposition in estimation error + approximation error: given any $\hat{\theta} \in \Theta$, we can write the excess risk of $f_{\hat{\theta}}$ as

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

  The approximation error $\big\{ \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) - \mathcal{R}^* \big\}$ is always nonnegative, 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 Θ grows, the approximation error goes down to zero if arbitrary functions can be approximated arbitrarily well by functions f_θ. It is also independent of the number n of observations.

  The estimation error $\big\{ \mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} \mathcal{R}(f_{\theta'}) \big\}$ is also always nonnegative and is typically random because the function $f_{\hat{\theta}}$ is random. It typically decreases in n and increases when Θ 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 θ' a minimizer on Θ of the expected risk $\mathcal{R}(f_{\theta'})$:

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

where the empirical optimization error is $\sup_{\theta\in\Theta}\big\{\widehat{\mathcal{R}}(f_{\hat\theta}) - \widehat{\mathcal{R}}(f_\theta)\big\}$ (it is equal to zero for exact empirical risk minimizers, but it is not when using the optimization algorithms from chapter 5 in practice). The uniform deviation defined as $\sup_{\theta\in\Theta}\big|\widehat{\mathcal{R}}(f_\theta) - \mathcal{R}(f_\theta)\big|$ grows with the "size" of Θ (e.g., number or norm of parameters), 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 Θ): this leads to constrained optimization and still allows for risk decompositions as done previously.

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

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

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

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

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

There is a difference between parameters (e.g., θ) learned on the training data and hyperparameters (e.g., λ) 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 that are polynomials of order k, from $k = 0$ (constant functions) to $k = 14$ (this corresponds to linear regression with $f_\theta(x)$ of the form $\theta^\top\varphi(x)$, where $\varphi(x) = (1, x, \dots, x^k)^\top \in \mathbb{R}^{k+1}$). 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 figures 2.1 and 2.2, the training error monotonically decreases in k while the testing error goes down and then up. Note the strong overfitting when k is large (third row in figure 2.1).

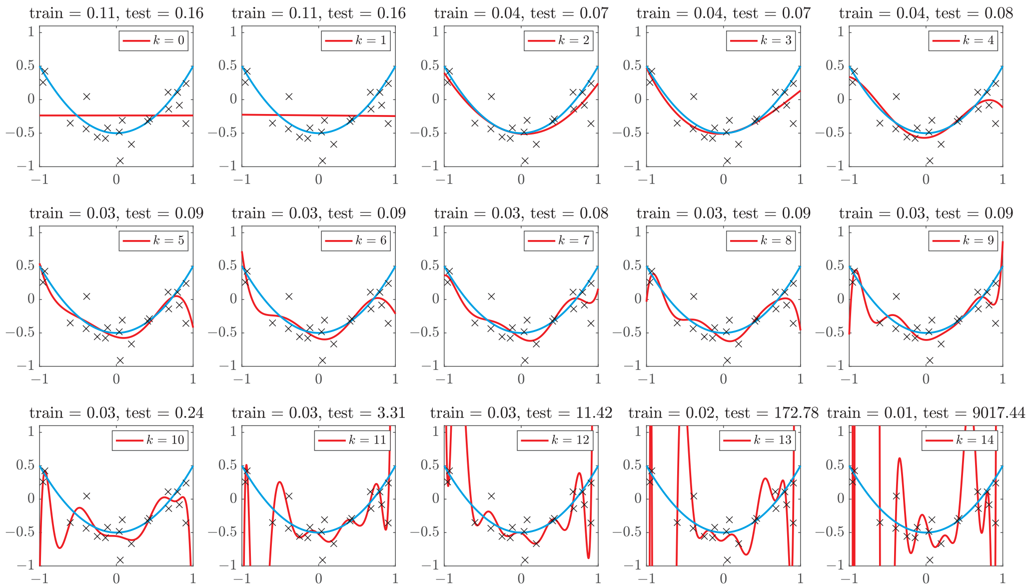


Figure 2.1. Polynomial regression with increasing orders k. Plots of estimated functions in red, with training and testing errors. The Bayes prediction function $f_*(x) = \mathbb{E}[y|x]$ is plotted in blue (it is the same for all plots).

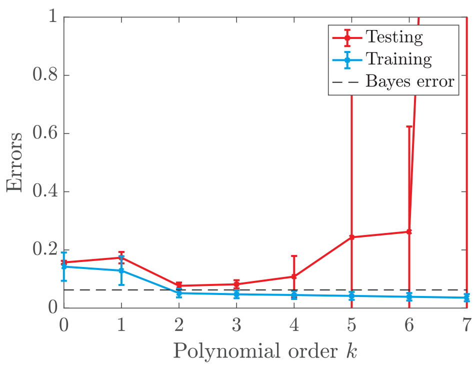


Figure 2.2. Polynomial regression with increasing orders. Plots of training and testing errors with error bars (computed as standard deviations obtained from 32 replications), together with the Bayes error. Note that the variance is increasing with the order k.

2.4 Statistical Learning Theory

The goal of learning theory is to provide some guarantees of performance on unseen data given some properties of the learning problem. A common assumption is that the data $\mathcal{D}_n(p) = \{(x_1, y_1), \dots, (x_n, y_n)\}$ are obtained as i.i.d. observations from some unknown distribution p from some family $\mathcal{P}$. The family $\mathcal{P}$ of probability distributions on (x, y) encapsulates the properties of the learning problem and one may consider conditions on the distributions of inputs or on the conditional distributions of outputs given inputs.

As seen earlier, 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 excess expected risk

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

is small, where $\mathcal{R}_p^*$ is the Bayes risk (which depends on the joint distribution p), assuming that $\mathcal{D}_n(p)$ 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 the randomness of the expected risk of the estimator to obtain a criterion:

- *Expected error*: we measure performance as

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

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

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

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

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

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

  The goal of learning theory in this framework is then to find an ε that is as small as possible (typically as a function of δ and n). The notion of PAC consistency corresponds, for any $\varepsilon > 0$, to have such an inequality for each n and a sequence δ_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), algorithm $\mathcal{A}$ is consistent in expectation for the distribution p.

Be careful with the order of quantifiers: the convergence speed of the excess risk toward zero will depend on p. See the "no free lunch" theorem in section 2.5 that highlights 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 has at most some complexity, e.g., linear in some feature vector or with a certain number of bounded derivatives). We thus aim at finding algorithm $\mathcal{A}$ such that

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

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

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

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

- Upper-bounding the optimal excess risk: one given algorithm with a convergence proof provides an upper bound. This is the main focus of this book.
- Lower-bounding the optimal excess risk: 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).

Nonasymptotic versus asymptotic analysis. Theoretical results in learning theory can be *nonasymptotic*, with an upper bound with explicit dependence on all quantities; the bound is then valid for all n, even if it is 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, which is common in random matrix theory, where dimension d of the features and number n of observations both tend to infinity, with a ratio tending to a constant (see, e.g., Potters and Bouchaud, 2020). See also the discussion in section 4.7.

The key aspect here is (arguably) how these rates depend on the problem. Specifically, the choice of in expectation versus in high probability, or asymptotic versus nonasymptotic, does not really matter 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 chapter 7 of Devroye et al. (1996) for more details.

Proposition 2.2 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 binary classification with 0–1 loss and $\mathcal{X}$ 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 $\mathcal{A}$,*

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

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 (1) 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 (2) to choose parameters of this distribution (the binary vector r defined next) with largest possible expected risk and compare this worst performance to the performance obtained by a random choice of 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 $j \in \{1, \ldots, k\}$; 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 $\hat{f}_{\mathcal{D}_n} = \mathcal{A}(\mathcal{D}_n(p))$ as the classifier, and $S(r) = \mathbb{E}\big[\mathcal{R}_p(\hat{f}_{\mathcal{D}_n})\big]$ as 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 $y = r_x$ almost surely. Note that we take expectations and probabilities with respect to $x_1, \ldots, x_n, x,$ *and* r (all being independent of each other).

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

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

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 $y = r_x$ has the same probability of being 0 or 1, i.e., a random guess, given that x 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 of proposition 2.2 is that the hard distribution used in the proof above may depend on n (from the proof, it takes k values, with k tending to infinity fast enough compared with n). The following proposition (theorem 7.2 from Devroye et al., 1996) is given without proof; it is much “stronger,” 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 $n \geqslant 1$:*

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

2.6 Quest for Adaptivity

As seen in section 2.5, 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 (as the required number n of observations to reach a given precision is exponential in d). No method can beat this–not k-nearest-neighbors, not kernel methods, and not even neural networks (see lower bounds on performance in chapter 15).

When the target function is smoother (i.e., 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 an r-dimensional linear projection, neural networks (if the optimization problem is solved correctly) will have the extra ability to lead to rates of the form $O(n^{-m/r})$ instead of $O(n^{-m/d})$. This is not the case for kernel methods (see chapter 9).

Note that another form of adaptivity, which is often considered, may apply 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. For more, see 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 chapters 7 and 9 for detailed results regarding adaptivity for kernel methods and neural networks.

2.7 Beyond Supervised Learning

This textbook focuses primarily on the traditional supervised learning paradigm, with i.i.d. 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. Next, we present briefly some of these extensions, 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 (PCA, studied in section 3.9), 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 in unsupervised learning; each method typically has an ad hoc empirical criterion, such as reconstruction of the data, full or partial (like in self-supervised learning); or 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).

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

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

Active learning. This is a similar setting to semisupervised learning, but the user can choose which unlabeled 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 unlabeled 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 to optimal control (Liberzon, 2011).

Generative modeling. A key task in computer vision or natural language processing is to generate images or text documents based on simple "prompts." Here, the goal is often not to give an output that minimizes some loss, but rather to sample from a distribution that reflects the natural variability of images and text, given the prompt. Sampling from such high-dimensional distributions is a practical and theoretical challenge, where diffusion models prove particularly useful (see, e.g., Chan, 2024, and references therein).

2.8 Summary–Book Outline

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

Part I: Preliminaries. Part I contains chapter 1 on mathematical preliminaries, this introductory chapter, and chapter 3, on linear least-squares regression. We start with least-squares, as it allows the introduction of the main concepts of the book, such as underfitting, overfitting, regularization, using only simple linear algebra, without the need for more advanced analytic or probabilistic tools.

Part II: Generalization bounds for learning algorithms. Part II 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, which 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 approximately minimize the empirical risk and, through stochastic gradient descent

(SGD), 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 "nonparametric" 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 adapting to the smoothness of the target prediction function.
- **Sparse methods**: While chapter 7 focused on Euclidean or Hilbertian regularization techniques for linear models, chapter 8 considers regularization by sparsity-inducing penalties such as the ℓ_1-norm or the ℓ_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, leading to nonconvex 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 nonlinear variable selection).

Part III: Special topics. Part III presents a series of chapters on special topics 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, such as in boosting methods.
- **From online learning to bandits**: Chapter 11 considers sequential decision problems within the regret framework, focusing first on online convex optimization, then on zeroth-order optimization (without access to gradients), and finally multiarmed bandits.
- **Overparameterized 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 (GD). We present the implicit bias of GD in linear models toward minimum Euclidean norm solutions and then the double descent phenomenon, before looking at implicit biases and global convergence for nonconvex optimization problems.
- **Structured prediction**: Chapter 13 goes beyond the traditional regression and binary classification frameworks by first considering multicategory 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–Bayesian analysis.
- **Lower bounds on generalization and optimization errors**: While most of the book is dedicated to obtaining upper bounds on the generalization or optimization errors of our algorithms, chapter 15 considers lower bounds on such errors, 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 $\sigma^2 d/n$, where σ^2 is the prediction noise variance.
- Ridge regression: With ℓ_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 generalization error: Under well-specification, the rate $\sigma^2 d/n$ cannot be improved.

3.1 Introduction

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

Why should we study linear least-squares regression? Has there not been any progress since 1805? Here are 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

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

dimension of the problem with no regularization, or on dimensionless quantities when regularization is added.

- Because of its simplicity, many results can be easily derived without the need for complicated mathematics, both in terms of algorithms and statistical analysis (simple linear algebra for the simplest results in the fixed design setting).
- Using nonlinear features, this approach can lead to arbitrary nonlinear predictions (see the discussion of kernel methods in chapter 7).

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

3.2 Least-Squares Framework

We recall the goal of supervised machine learning from chapter 2: we are given some training data composed of observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, which are pairs of inputs/outputs, sometimes referred to as features/responses. Given a new $x \in \mathcal{X}$, the goal is to 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 chapter 2 that the optimal predictor is $f_*(x) = \mathbb{E}[y|x]$ (see section 2.2.3).

In this chapter, we consider empirical risk minimization for regression problems. 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 to be *misspecified*.

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

Being linear in x or linear in θ is different!

While we assume linearity in parameter θ, nothing forces $f_\theta(x)$ to be linear in 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:

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

When $\mathcal{X} \subset \mathbb{R}^d$, we can make the extra assumption that f_θ is an affine function in x, which can be obtained through $\varphi(x) = \binom{x}{1} = (x^\top, 1)^\top \in \mathbb{R}^{d+1}$. Another classical assumption is to consider vectors $\varphi(x)$ composed of monomials (so that prediction functions are polynomials, as done in experiments in section 3.5.2). We will see in chapter 7 (kernel methods) that we can consider infinite-dimensional features.

Matrix notation. The cost function shown in equation (3.1) can be rewritten in matrix notation. Let $y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ 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 this notation, the empirical risk is

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

where $\|\alpha\|_2^2 = \sum_{j=1}^{d}\alpha_j^2$ is the squared ℓ_2-norm of α.

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

3.3 Ordinary Least-Squares Estimator

We assume that the matrix $\Phi \in \mathbb{R}^{n\times d}$ has full column rank (i.e., the rank of Φ is d). In particular, the problem is said to be "overdetermined," 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 Φ has full column rank, the minimizer of equation (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 through a linear system.

Proposition 3.1 *When Φ 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 noncentered[2] *empirical covariance matrix as* $\widehat{\Sigma} = \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d\times d}$*; we have* $\hat{\theta} = \frac{1}{n}\widehat{\Sigma}^{-1}\Phi^\top y$.

Proof Since the function $\widehat{\mathcal{R}}$ 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 $\widehat{\mathcal{R}}'(\hat{\theta}) = 0$ where $\widehat{\mathcal{R}}'(\theta) \in \mathbb{R}^d$ is the gradient of $\widehat{\mathcal{R}}$ at θ. For all $\theta \in \mathbb{R}^d$, we get, by expanding the square and computing the gradient:

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

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

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

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

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

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

3.3.2 Geometric Interpretation

The OLS estimator has a natural geometric interpretation.

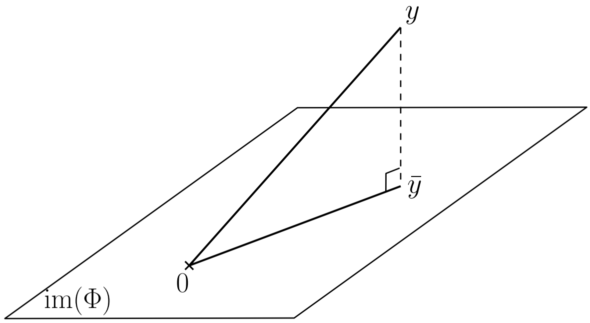
Proposition 3.2 *The vector of predictions* $\Phi\hat{\theta} = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top y$ *is the orthogonal projection of* $y \in \mathbb{R}^n$ *onto* $\operatorname{im}(\Phi) \subset \mathbb{R}^n$*, the column space of* Φ.

Proof Let us show that $\Pi = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top \in \mathbb{R}^{n\times n}$ is the orthogonal projection on $\operatorname{im}(\Phi)$. For any $a \in \mathbb{R}^d$, it holds $\Pi\Phi a = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top\Phi a = \Phi a$, so $\Pi u = u$ for all $u \in \operatorname{im}(\Phi)$. Also, since $\operatorname{im}(\Phi)^\perp = \operatorname{null}(\Phi^\top)$, then for all $u' \in \operatorname{im}(\Phi)^\perp$, $\Phi^\top u' = 0$, and hence $\Pi u' = \Phi(\Phi^\top\Phi)^{-1}(\Phi^\top u') = 0$. These properties characterize the orthogonal projection on $\operatorname{im}(\Phi)$. Alternatively, we directly have $\Phi\hat{\theta} = \arg\min_{z\in\operatorname{im}(\Phi)} \|y - z\|_2^2$. ■

We can thus interpret the OLS estimation as doing the following (see the following plot for an illustration):

1. Compute the projection $\bar{y}$ of y onto the image of Φ.
2. Solve the linear system $\Phi\theta = \bar{y}$, which has a unique solution.

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



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 Φ 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 the positive semidefinite matrix $\Phi^\top \Phi \in \mathbb{R}^d$. One then has, since R is invertible,

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

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 (GD). It consists in approximately minimizing $\widehat{\mathcal{R}}$ by taking an initial point $\theta_0 \in \mathbb{R}^d$ and iteratively going toward the minimizer by following the opposite of the gradient:

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

where $\gamma > 0$ is the step size. When these iterates converge, they do toward the OLS estimator since a fixed-point θ satisfies $\widehat{\mathcal{R}}'(\theta) = 0$. We will study such algorithms in chapter 5, with running-time complexities going down to linear in d, e.g., $O(nd)$.

3.4 Statistical Analysis of Ordinary Least-Squares

In this section, we provide guarantees on the predictive performance of the OLS estimator. There are two classical settings of analysis for least-squares regression:

- *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 chapter 2). 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 (but the output data $(y_1, \dots, y_n)$ are themselves 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 Φ is deterministic):

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

  This assumption allows a complete analysis with basic linear algebra. It is justified in some settings, such as 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, we want to estimate well a label vector y resampled from the same distribution as the observed y, and no attempts are made to generalize to unseen input points $x \in \mathcal{X}$. The risk in equation (3.3) is often called the *in-sample prediction error*, and the task can be seen as "denoising" the labels.

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

3.5 Fixed Design Setting

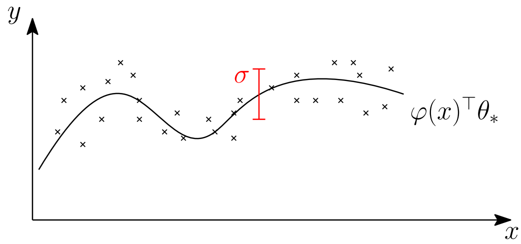
We thus assume that Φ is deterministic, and as before, that $\widehat{\Sigma} = \frac{1}{n}\Phi^\top\Phi$ is invertible. Any guarantee requires assumptions about how the data are generated. We assume the following:

- There is 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 ε_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 the later bound $\sigma^2 d/n$ to be an equality). Note that to prove upper bounds in generalization error, we could also only assume that $\mathbb{E}[\varepsilon_i^2] \leqslant \sigma^2$ for each $i \in \{1, \dots, n\}$. The noise variance σ^2 is the expected squared error between the observations y_i and the model $\varphi(x_i)^\top \theta_*$, as illustrated below:



⚠ In equation (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).

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 σ^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 θ_* coincides with the maximum likelihood estimator (where Φ is assumed to be fixed). Indeed, the density/likelihood of y is, using independence between the noise variables ε_i and the density of the Gaussian 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 given above, show that $\tilde{\sigma}^2$ the maximum likelihood estimator of σ^2 is equal to $\tilde{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \varphi(x_i)^\top \hat{\theta})^2$.*

Denoting by $\mathcal{R}^*$ the minimum value of $\mathcal{R}(\theta) = \mathbb{E}_y \big[\frac{1}{n} \|y - \Phi\theta\|_2^2 \big]$ over $\mathbb{R}^d$, proposition 3.3 shows that it is attained at θ_* and is equal to σ^2.

Proposition 3.3 (Risk decomposition for OLS–fixed design) *Under the linear model and fixed design assumptions made in this section, 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 θ_*), then*

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

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 θ_* is the unique global minimizer of $\mathcal{R}(\theta)$, and the minimum value $\mathcal{R}^*$ is equal to σ^2. This shows the first claim.

Now if θ is random, we perform the usual *bias/variance decomposition*:

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

(This is also a simple application of the law of total variance for vectors; that is, $\mathbb{E}\big[\|z - a\|_M^2\big] = \|\mathbb{E}[z] - a\|_M^2 + \mathbb{E}\big[\|z - \mathbb{E}[z]\|_M^2\big]$, applied 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 ε, and, to compute the expectation and variance of $\hat{\theta}$, we thus need to compute the expectation of linear and quadratic forms in ε.

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 Since $\mathbb{E}[y] = \Phi\theta_*$, we have directly $\mathbb{E}[\hat{\theta}] = (\Phi^\top\Phi)^{-1}\Phi^\top\Phi\theta_* = \theta_*$. Moreover, $\hat{\theta} - \theta_* = (\Phi^\top\Phi)^{-1}\Phi^\top(\Phi\theta_* + \varepsilon) - \theta_* = (\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon$. Thus, using $\mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I$, 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 expressions of the bias (equal to 0) and variance in the risk decomposition of proposition 3.3.

Proposition 3.5 (Risk of OLS) *The excess risk of the OLS estimator equals*

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

Proof Note here that the expectation is over ε 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

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

Thus, using proposition 3.4, we have, using the "trace trick,"

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

since the identity matrix is of size $d \times d$. We can also give a direct proof: Using the identity $\hat{\theta} - \theta_* = (\Phi^\top\Phi)^{-1}\Phi^\top\varepsilon$, we get

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

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

We can make the following observations:

- In the fixed design setting, OLS thus leads to unbiased estimation, with an excess risk of $\sigma^2 d/n$.
- ⚠ In the fixed design setting, the expectation over ε appears twice: (1) in the definition of the testing risk of some arbitrary θ in equation (3.3), and (2) when taking an expectation over the data in equation (3.5) to compute the expectation of the testing risk for the OLS estimator $\hat{\theta}$.

- In exercise 3.2, 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 regression, 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]

  Exercise 3.2 *Show that the expected empirical risk is equal to $\mathbb{E}[\widehat{\mathcal{R}}(\hat{\theta})] = \frac{n-d}{n}\sigma^2$. In particular, when $n > d$, deduce that an unbiased estimator of the noise variance σ^2 is given by $\frac{1}{n-d}\|y - \Phi\hat{\theta}\|_2^2$.*

- 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 σ^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 ℓ_1-norm in chapter 8) will come to the rescue.
- This is only for the fixed design setting. We consider the random design setting next, which is a bit more involved mathematically, primarily because of the presence of $\widehat{\Sigma}^{-1}$, which does not cancel any further, leading to the term $\widehat{\Sigma}^{-1}\Sigma$, where Σ is the population covariance matrix.

Exercise 3.3 (General noise) *Consider the fixed design regression model $y = \Phi\theta_* + \varepsilon$ with ε with zero mean and covariance matrix equal to $C \in \mathbb{R}^{n\times n}$ (not $\sigma^2 I$ anymore). Show that the expected excess risk of the OLS estimator is equal to $\frac{1}{n}\operatorname{tr}\left[\Phi(\Phi^\top\Phi)^{-1}\Phi^\top C\right]$.*

Exercise 3.4 (Multivariate regression (♦)) *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 ε has zero-mean with covariance matrix $S \in \mathbb{R}^{k\times k}$. 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 obtained from i.i.d. $\varepsilon_i \in \mathbb{R}^k$, $i = 1, \dots, n$, derive the OLS estimator minimizing $\frac{1}{n}\|Y - \Phi\theta\|_{\mathrm{F}}^2$ and its excess risk (where $\|M\|_{\mathrm{F}}$ denotes the Frobenius norm defined as the square root of the sum the squared components of M).*

3.5.2 Experiments

To illustrate the bound $\sigma^2 d/n$, we consider polynomial regression in one dimension, with $x \in \mathbb{R}$ and the feature vector $\varphi(x) = (1, x, x^2, \dots, x^k)^\top \in \mathbb{R}^{k+1}$, 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 OLS estimator is plotted in red for various values of n, from $n = 10$ to $n = 1,000$, for $k = 5$. We can observe in figure 3.1 that the testing error goes down when n increases.

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

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

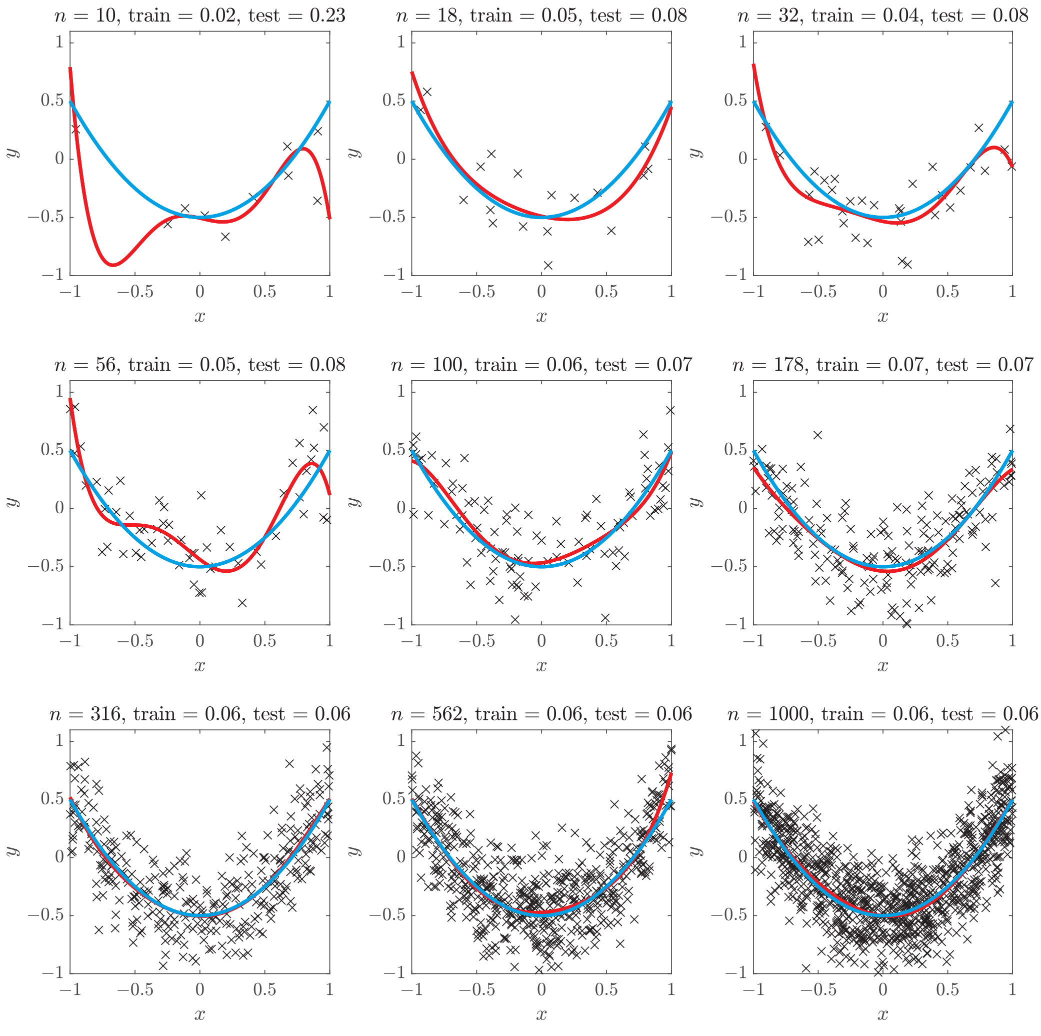


Figure 3.1. Polynomial regression in one dimension with a varying number of observations (from $n = 10$ to $n = 1,000$ with training and testing errors. Blue: Optimal prediction, red: estimated prediction by OLS with degree-5 polynomials.

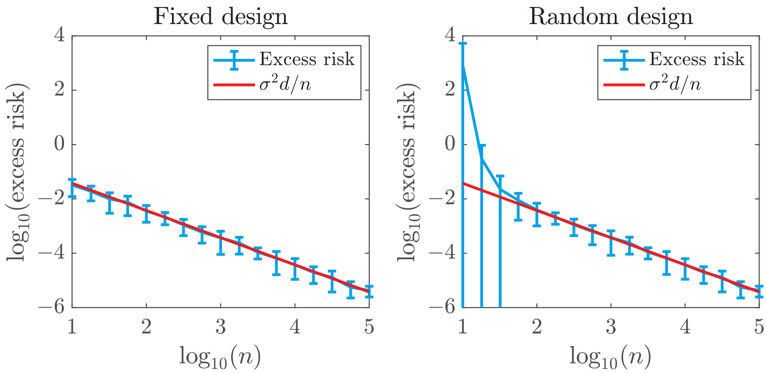


Figure 3.2. Convergence rate for polynomial regression with error bars (obtained from 32 replications by adding/subtracting standard deviations), plotted in logarithmic scale, with fixed design (left plot) and random design (right plot). The large error bars for small n in the right plot are due to the 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 valid only 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, e.g., when $d = n$ and Φ is a square invertible matrix, $\theta = \Phi^{-1}y$ leads to $y = \Phi\theta$; that is, OLS 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 vector of lower dimension, with a classical method being principal component analysis (PCA), presented in section 3.9, or random projections (presented in section 10.2.2). Regularization adds a term to the least-squares objective, typically either an ℓ_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 in chapter 7).

Definition 3.2 (Ridge least-squares regression estimator) *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. Note that we no longer require $\Phi^\top \Phi$ 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 with 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 (discussed in section 1.1.3), show that the ridge regression estimator given in proposition 3.6 can also be written as $\hat{\theta}_\lambda = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top y = \Phi^\top(\Phi\Phi^\top + n\lambda I)^{-1}y$. 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 that, using the fact that $\widehat{\Sigma}$ and $(\widehat{\Sigma} + \lambda I)^{-1}$ commute,

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

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 proof ends by summing the bias and variance terms. ■

We can make the following observations:

- The result given above is also a bias/variance decomposition with the bias equal to $B = \lambda^2\theta_*^\top(\widehat{\Sigma} + \lambda I)^{-2}\widehat{\Sigma}\theta_*$, and the variance 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.

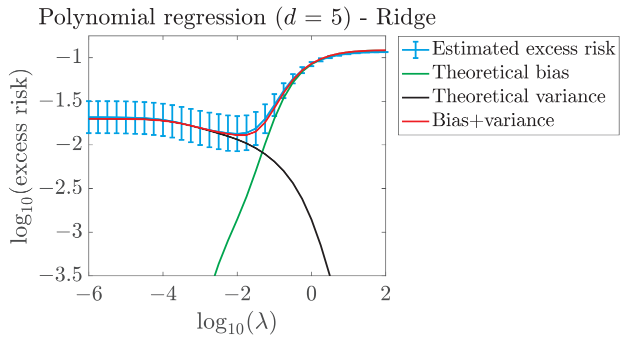


Figure 3.3. Polynomial regression (same setup as figure 3.2, with $n = 300$), with $k = 5$: bias/variance trade-offs for ridge regression as a function of λ. We can see the monotonicity of bias and variance with respect to λ and the presence of an optimal choice of λ.

- 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 λ and equal to zero for $\lambda = 0$ if $\widehat{\Sigma}$ is invertible, while when λ 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 λ and is equal to $\sigma^2 d/n$ for $\lambda = 0$ if $\widehat{\Sigma}$ is invertible, and converging to zero when λ 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 $\sum_{j=1}^d \lambda_j^2/(\lambda_j+\lambda)^2$, where $(\lambda_j)_{j\in\{1,\dots,d\}}$ are the eigenvalues of $\widehat{\Sigma}$. This quantity will be important in analyzing kernel methods in chapter 7. Since the function $\mu \mapsto \mu^2/(\mu+\lambda)^2$ is increasing from 0 to 1 (when μ goes from 0 to $+\infty$), close to zero if $\mu \ll \lambda$, and close to 1 if $\mu \gg \lambda$, the degrees of freedom provide a soft count of the number of eigenvalues that are larger than λ.
- 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 λ. Based on the expression for the risk, we can tune the regularization parameter λ 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* $\lambda_* = \dfrac{\sigma \operatorname{tr}(\widehat{\Sigma})^{1/2}}{\|\theta_*\|_2 \sqrt{n}}$, *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 gotten, 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 μ 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[4] $V = \dfrac{\sigma^2}{n} \operatorname{tr}\big[\widehat{\Sigma}^2 (\widehat{\Sigma} + \lambda I)^{-2}\big] = \dfrac{\sigma^2}{\lambda n} \operatorname{tr}\big[\widehat{\Sigma} \lambda \widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^{-2}\big] \leqslant \dfrac{\sigma^2 \operatorname{tr} \widehat{\Sigma}}{2 \lambda n}$. This leads to

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

The bound above is of the form $a\lambda + b/\lambda$ for $a, b > 0$, and is minimized for $\lambda = \sqrt{b/a}$ with optimal value $2\sqrt{ab}$. This is how λ_* is chosen (i.e., chosen to minimize the upper bound on $B + V$), which leads to the desired result. ■

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, 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). Note, however, that in practice, R often increases with dimension d.

  The number of parameters is usually not the best way to measure the generalization capabilities of a learning method; here, the maximal feature norm R is more informative and depends on how the data are normalized.

- 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 σ^2 to σ). The presence of a *fast* rate in $O(n^{-1})$ with a potentially

[4] Using the properties (proof using eigenvalue decompositions left as an exercise) that for any vector u, any symmetric matrix M, and any symmetric positive semidefinite matrix A, $u^\top M u \leqslant \|u\|_2^2 \cdot \lambda_{\max}(M)$ and $\operatorname{tr}(AM) \leqslant \operatorname{tr}(A) \cdot \lambda_{\max}(M)$.

large constant and of a *slow* rate $O(n^{-1/2})$ with a smaller constant will be explored several times in this book.

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

- The value of λ^* involves quantities that we typically do not know in practice (such as σ and $\|\theta_*\|_2$). This is still useful to highlight the existence of some λ 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 unit 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 σ^2). Moreover, we have $[y] = [\sigma] = [\varphi][\theta]$, leading to $[\lambda] = [\varphi]^2$. The value of λ suggested in proposition 3.8 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 λ in practice. The regularization λ is an example of a *hyperparameter*. 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 hyperparameters, 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 (♦) *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*

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

where $\operatorname{Diag}(h)$ denotes the diagonal matrix with h as its diagonal. Hint: use Woodbury matrix identities from section 1.1.3.

3.7 Lower Bound (♦)

In this section, our aim is to compute a lower bound on the excess risk for *any* estimator that is a function of Φ and y. This lower bound will turn out to be equal to the upper bound on the excess risk obtained from the OLS estimator, namely $\sigma^2 d/n$, showing that this estimator is optimal.

To show a lower bound in the fixed design setting, we will consider only Gaussian noise (to obtain lower bounds, we can specialize the problem as much as we want); that is, ε has a joint Gaussian distribution with mean zero and covariance matrix $\sigma^2 I$. We follow the elegant and simple proof technique outlined by Mourtada (2022).

The only unknown in the model is the location of θ_*. To make the dependence on θ_* explicit, we denote by $\mathcal{R}_{\theta_*}(\theta) - \mathcal{R}^*$ the excess risk (in chapter 2, 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 Φ). Indeed, algorithms take $y = \Phi\theta_* + \varepsilon \in \mathbb{R}^n$ as input and then 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 distribution on θ_*, called the "prior distribution" in Bayesian statistics. That is, we have, for any algorithm/estimator $\mathcal{A}$ (for a parameter $\lambda > 0$ that will be chosen to tend to zero later in the discussion),

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

Here, we choose the Gaussian 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 using $\sigma^2 = \mathcal{R}^*$), we thus get the lower bound

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

which we need to minimize with respect to $\mathcal{A}$. By making θ_* random, we now have a joint Gaussian distribution for (θ_*, ε). 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

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

Thus, for each y, the optimal $\mathcal{A}(y)$ has to minimize $\int_{\mathbb{R}^d} \|\mathcal{A}(y) - \theta_*\|_{\widehat{\Sigma}}^2 dp(\theta_*|y)$, which is exactly the posterior mean of θ_* 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 $p(\theta_*|y)$.

Since the joint distribution of (θ_*, y) is Gaussian with known parameters, we could use classical results about conditioning for Gaussian vectors (see section 1.1.3). We instead 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 θ_*. Up to constants and using independence of ε and θ_*, this log-likelihood is the sum of the log-likelihoods of ε and θ_*:

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

which is exactly (up to a sign and a constant) the ridge regression cost function in section 3.6. Thus, from proposition 3.6, we have $\mathcal{A}_*(y) = (\Phi^\top\Phi + n\lambda I)^{-1}\Phi^\top y$, 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 equation (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}^* \\
& \qquad\qquad\qquad \text{using equality of posterior mean and posterior mode}, \\
=\ & \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\qquad\qquad\qquad \text{by independence of } \varepsilon \text{ and } \theta_*, \\
=\ & \lambda^2 \mathbb{E}_{\theta_* \sim \mathcal{N}(0,\frac{\sigma^2}{\lambda n}I)}\big[\theta_*^\top(\widehat{\Sigma} + \lambda I)^{-1}\widehat{\Sigma}(\widehat{\Sigma} + \lambda I)^{-1}\theta_*\big] + \frac{1}{n^2}\mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\big[\varepsilon^\top\Phi\widehat{\Sigma}(\widehat{\Sigma} + \lambda I)^{-2}\Phi^\top\varepsilon\big] \\
=\ & \lambda^2\frac{\sigma^2}{n\lambda}\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)^{-2}(\lambda\widehat{\Sigma} + \widehat{\Sigma}^2)\big] = \frac{\sigma^2}{n}\operatorname{tr}\big[(\widehat{\Sigma} + \lambda I)^{-1}\widehat{\Sigma}\big].
\end{aligned}$$

When Φ (and thus $\widehat{\Sigma}$) has full rank, $\frac{\sigma^2}{n}\operatorname{tr}\big[(\widehat{\Sigma} + \lambda I)^{-1}\widehat{\Sigma}\big]$ tends to $\frac{\sigma^2}{n}\operatorname{tr}(I) = \frac{\sigma^2 d}{n}$ when λ tends to zero (otherwise, it tends to $\frac{\sigma^2}{n}\operatorname{rank}(\Phi)$). This 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 testing error, which exactly matches the upper bound obtained by OLS. Note that this lower bound is the infimum over all potential estimators of the worst-case error for all potential choices of θ_*; it thus shows that in this worst-case sense, ridge regression cannot improve on OLS (but it often does for particular choices of θ_*). To go beyond least-squares, such lower bounds are significantly harder to show. See the more general discussion in the dedicated 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 i.i.d. from a probability distribution p on $\mathcal{X} \times \mathbb{R}$. We aim to show that the bound on the excess risk that 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 is 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 the same assumption holds for all i, as observations are i.i.d.).

With the assumption made in this section, $\mathbb{E}[y_i|x_i] = \varphi(x_i)^\top \theta_*$, 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, given in proposition 3.9.

Proposition 3.9 (Excess risk for random design least-squares regression) *Under the random design linear model, 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 noncentered 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 ε_0 the corresponding noise variable,

$$\begin{aligned}
\mathcal{R}(\theta) &= \mathbb{E}\big[(y_0 - \theta^\top \varphi(x_0))^2\big] = \mathbb{E}\big[(\theta_*^\top \varphi(x_0) + \varepsilon_0 - \theta^\top \varphi(x_0))^2\big] \\
&= \mathbb{E}\big[((\theta_* - \theta)^\top \varphi(x_0))^2\big] + \mathbb{E}\big[\varepsilon_0^2\big] + \mathbb{E}\big[2\varepsilon_0(\theta_* - \theta)^\top \varphi(x_0)\big] \\
&= \mathbb{E}\big[(\theta_* - \theta)^\top \varphi(x_0)\varphi(x_0)^\top (\theta_* - \theta)\big] + \mathbb{E}\big[\varepsilon_0^2\big] + 0,
\end{aligned}$$

since ε_0 and x_0 are independent and $\mathbb{E}[\varepsilon_0] = 0$, leading to $\mathcal{R}(\theta) = (\theta - \theta_*)^\top \Sigma(\theta - \theta_*) + \sigma^2$, which leads to the desired result. ■

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

Proposition 3.10 *Under the random design linear model, assuming that $\widehat{\Sigma}$ 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, using proposition 3.9,

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

■

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 noninvertibility of $\widehat{\Sigma}$. Under simple assumptions (e.g., $\varphi(x)$ has a strictly positive 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., section 3 from Mourtada, 2022).

Exercise 3.8 *Show that for the random design setting with the same assumptions as proposition 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 that we assume that $\varphi(x)$ has a Gaussian distribution with mean 0 and covariance matrix Σ. In that case, we can directly compute the desired expectation by first considering $z = \Sigma^{-1/2}\varphi(x)$, which has a standard Gaussian distribution (i.e., with mean zero and identity covariance matrix), with the corresponding normalized design matrix $Z \in \mathbb{R}^{n\times d}$ such that $\Phi = Z\Sigma^{1/2}$, 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 function $M \mapsto \operatorname{tr}(M^{-1})$ on the cone of positive-definite matrices, using Jensen's inequality, we see that $\mathbb{E}\big[\operatorname{tr}((Z^\top Z)^{-1})\big] \geqslant \operatorname{tr}\big(\mathbb{E}[Z^\top Z])^{-1}\big) = \frac{d}{n}$ (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

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

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 $n > d+1$, leading to the expected excess risk of

$$\frac{\sigma^2 d}{n-d-1} = \frac{\sigma^2 d}{n} \frac{1}{1-(d+1)/n}. \tag{3.8}$$

See Breiman and Freedman (1983) for further details. Note here that for Gaussian designs, the expected risk is precisely equal to the expression in equation (3.8) and that later in this book, we will only consider upper bounds. See also a further analysis in section 12.2.3 in the context of double descent.

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

3.8.2 General Designs (♦♦)

This last, more technical subsection highlights how the Gaussian assumption can be avoided. The main idea is to show that with high probability, the lowest eigenvalue of $\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2}$ is larger than some $1-t$ for some $t \in (0,1)$. Since the excess risk is the expectation of $\frac{\sigma^2}{n}\operatorname{tr}(\Sigma\widehat{\Sigma}^{-1})$, 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, concentration inequalities for matrices are needed next, such as described by Tropp (2012), Hsu et al. (2012), Oliveira (2016), and Lecué and Mendelson (2016). Also, see complementary results by Mourtada (2022).

Matrix concentration inequality. We will use the matrix Bernstein bound, adapted from theorem 1.4 of Tropp (2012), 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*

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

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

Application to rescaled covariance matrices. We can now prove proposition 3.12, which will give the desired high-probability bound for the excess risk with one extra assumption. Next, we will use the partial order between symmetric matrices, defined as $A \succcurlyeq B \Leftrightarrow B \preccurlyeq A \Leftrightarrow A - B$ is positive semidefinite.

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 $\rho > 0$,*

$$\mathbb{E}\Big[\varphi(x)^\top\Sigma^{-1}\varphi(x)\varphi(x)\varphi(x)^\top\Big] \preccurlyeq \rho d\Sigma. \tag{3.9}$$

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

$$\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \succcurlyeq \frac{1}{4}I. \tag{3.10}$$

Before giving the proof, note that from the discussion earlier, the bound in equation (3.10) leads to an excess risk that is less than $\frac{\sigma^2 d}{n}\frac{1}{1-t} = 4\frac{\sigma^2 d}{n}$ for $t = 3/4$. Moreover, which is no surprise, the bound is nonvacuous 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 equation (3.9) 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 equation (3.9) is then equivalent to

$$\lambda_{\max}\big(\mathbb{E}\big[\|z\|^2 zz^\top\big]\big) \leqslant \rho d. \tag{3.11}$$

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, we always have $\rho \geqslant 1$,[6] and, 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 Consider the random symmetric matrix $M_i = I - z_i z_i^\top$, which satisfies $\mathbb{E}[M_i] = 0$, $\lambda_{\max}(M_i) \leqslant 1$ almost surely, and $\mathbb{E}[M_i^2] = \mathbb{E}\big[\|z_i\|^2 z_i z_i^\top\big] - I$ with the largest eigenvalue less than ρd (by equation (3.11)). We thus have for any $t \geqslant 0$, using proposition 3.11:

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

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 $I - \frac{1}{n}Z^\top Z = I - \Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \preccurlyeq tI$; that is, the desired result $\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \succcurlyeq (1-t)I$.

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

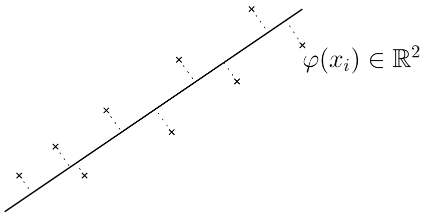
3.9 Principal Component Analysis (♦)

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 that is 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 the following

[6] From equation (3.11), we have $d^2 = (\mathbb{E}[\|z\|_2^2])^2 \leqslant \mathbb{E}[\|z\|_2^4] = \mathbb{E}[\|z\|_2^2\operatorname{tr}(zz^\top)] = \operatorname{tr}(\mathbb{E}[\|z\|_2^2 zz^\top]) \leqslant d\lambda_{\max}(\mathbb{E}[\|z\|_2^2\, zz^\top]) \leqslant \rho d^2$.

illustration for $d = 2$ and $k = 1$, where the goal is to minimize the sum of squares of all dotted segments).



In the formulation presented in this subsection, we consider a *linear* subspace (which contains 0), but it is common in practice to look for the optimal *affine* subspace (which 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 (nonuniquely) the subspace 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 $\Phi VV^\top$, and the optimal V is found by minimizing

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

Thus, minimizing $\|\Phi - \Phi VV^\top\|_{\mathrm{F}}^2$ is equivalent to maximizing $\operatorname{tr}\left[V^\top\Phi^\top\Phi V\right]$ with respect to a matrix $V \in \mathbb{R}^{d\times k}$ with orthonormal columns. Given an eigenvalue decomposition of the noncentered 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 λ a vector with nonincreasing 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 alternating 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 PCA. Using the singular value decomposition of Φ, 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 of D; compute the corresponding updates.*

Exercise 3.10 (K-means clustering) *Given $\Phi \in \mathbb{R}^{n\times d}$, we consider minimizing the objective $\|\Phi - AD\|_{\mathrm{F}}^2$ 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 1. Compute the updates of an alternating 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 regression (as shown in section 3.6), performing PCA and then unregularized OLS 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 $y = \Phi\theta_* + \varepsilon$ as in section 3.5, we can compute the excess risk of the estimator based on PCA as follows (using $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I$):

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

using that $\widehat{\Sigma} = \frac{1}{n}\Phi^\top\Phi$. We can then use that the columns of V are eigenvectors of $\widehat{\Sigma}$ so that $\widehat{\Sigma}V = VD$ and $V^\top\widehat{\Sigma}V = D$, for a diagonal matrix $D \in \mathbb{R}^{k\times k}$, leading to

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

Since V is composed of the eigenvectors of $\widehat{\Sigma}$ with the k largest eigenvalues, the matrix $(I - VV^\top)\widehat{\Sigma}(I - VV^\top)$ has all of its eigenvalues less than λ_{k+1}, where λ_{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 the 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 equation (3.6) (for ridge regression). A good value of k is then the closest integer to $\|\theta_*\|_2 \cdot (\mathrm{tr}[\widehat{\Sigma}])^{1/2}\sqrt{n}/\sigma$, leading, up to constants, to the same excess risk than for ridge regression, with the identification $k \sim \frac{1}{\lambda}\,\mathrm{tr}[\widehat{\Sigma}]$.

3.10 Conclusion

In this chapter, we have considered the simplest machine learning setup; that is, square loss and prediction functions linearly parameterized by a finite-dimensional parameter. This simple 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 error (which characterizes the modeling assumptions made by the chosen class of functions) and the estimation error (which characterizes the effect of having a finite number of observations).
- Rademacher complexity: To study estimation errors and compute expected uniform deviations of real-valued outputs, Rademacher complexities, also referred to as Rademacher averages, are a very flexible and powerful tool that allows obtaining uniform deviation bounds. This leads to dimension-independent upper bounds on estimation errors for constrained or penalized linear predictors.

As outlined in chapter 2, given a joint distribution p on $\mathcal{X} \times \mathcal{Y}$, and n independent and identically distributed observations (i.i.d.) 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, with a focus on statistical analysis (i.e., generalization to unseen data); optimization algorithms to efficiently find approximate minimizers will be studied in chapter 5. Before looking at the necessary probabilistic tools, we will 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 setup (see chapter 13).

As our goal is to estimate a binary-valued function, the first idea that comes to 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 it. Classical penalty-based regularization techniques can then be used for theoretical analysis (this chapter) and gradient-based methods for efficient 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, we will use instead the generic tool of Rademacher complexities (presented in section 4.5).

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

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

There are several conventions to define a prediction $f(x) \in \{-1, 1\}$ when $g(x) = 0$; a common one is to always choose one of the two labels. In this book, to preserve symmetry between -1 and 1 and to make sure that the loss function can be expressed as a function of $yg(x)$, we consider random predictions; that is, when $g(x) = 0$, the classifier $f(x)$ is sampled uniformly at random in $\{-1, 1\}$, independently from all other random quantities. When computing the loss incurred by the prediction $f(x)$, we will always take the expectation with respect to this random choice. Note that in practice, having $g(x)$ exactly equal to 0 occurs rarely, so the choice of convention makes no visible difference.

The 0–1 risk of function $f = \text{sign} \circ g$, still denoted as $\mathcal{R}(g)$ (⚠ note the slight overloading of notations $\mathcal{R}(g) = \mathcal{R}(\text{sign} \circ g)$), is then equal to, separating between situations where $g(x) = 0$ or not,

$$\begin{aligned} \mathcal{R}(g) &= \mathbb{P}(f(x) \neq y) = \mathbb{E}[1_{g(x)\neq 0} 1_{f(x)\neq y}] + \mathbb{E}[1_{g(x)=0} 1_{f(x)\neq y}] \\ &= \mathbb{E}[1_{yg(x)<0}] + \frac{1}{2}\mathbb{E}[1_{g(x)=0}] = \mathbb{E}\big[\Phi_{0-1}(yg(x))\big], \end{aligned}$$

where $\Phi_{0-1} : \mathbb{R} \to \mathbb{R}$ is defined as

$$\Phi_{0-1}(u) = \begin{cases} 1 & \text{if } u < 0 \\ \frac{1}{2} & \text{if } u = 0 \\ 0 & \text{if } u > 0, \end{cases} \tag{4.1}$$

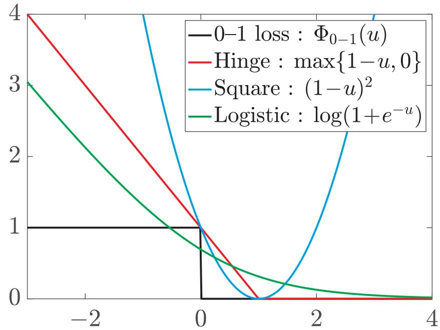


Figure 4.1. Classical convex surrogates for binary classification with the 0–1 loss, with Φ_{0-1} defined in equation (4.1).

and is called the "margin-based" 0–1 loss function or simply the 0–1 loss function.

⚠ Note the slightly overloaded notation 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 Φ_{0-1} is not continuous (and thus also nonconvex) and leads to difficult optimization problems.

4.1.1 Convex Surrogates

A key concept in machine learning is the use of *convex surrogates*, where we replace Φ_{0-1} by another function Φ with better numerical properties (mostly convexity). See the classic examples discussed next and plotted in figure 4.1.

Instead of minimizing the classical risk $\mathcal{R}(g)$ or its empirical version, one then minimizes the Φ-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 tackled 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 regression, 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 discussed next (which are nonincreasing).

- **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.

- **Hinge loss:** $\Phi(u) = \max(1-u, 0)$. With linear predictors, this leads to the support vector machine (SVM), and $yg(x)$ is often called the "margin" in this context. This loss has a geometric interpretation (see section 4.1.2).[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).

Section 4.1 analyzes precisely how replacing the 0–1 loss with convex surrogates still leads to optimal predictions. This allows us to focus only on *real-valued prediction functions* in the rest of this book. We will consider loss functions $\ell(y, f(x))$, which will be the square loss $(y - f(x))^2$ for regression, and any of the ones mentioned previously for binary classification; that is, $\Phi(yf(x))$. We will consider alternatives and extensions in chapter 13 (on structured prediction).

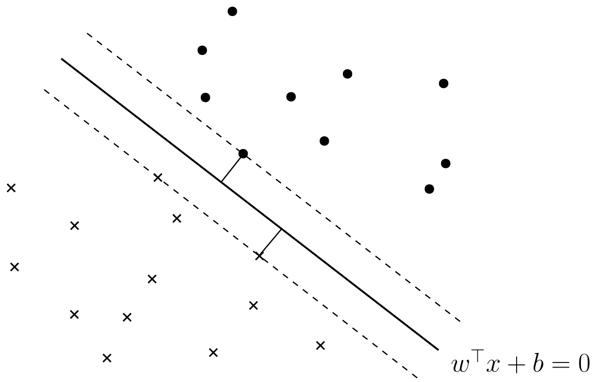
4.1.2 Geometric Interpretation of the Support Vector Machine (♦)

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 are $w \in \mathbb{R}^d$ 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 maximizing the distance to the closest points, as illustrated:

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

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



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, the minimal distance is

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

and we thus aim at maximizing this quantity. Because of the invariance by rescaling (i.e., 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:

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

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 $C > 0$):

$$\min_{w \in \mathbb{R}^d,\ b \in \mathbb{R},\ \xi \in \mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \xi_i \ \text{ such that } \ \forall i \in \{1,\dots,n\},\ y_i(w^\top x_i + b) \geqslant 1 - \xi_i,\ \xi_i \geqslant 0. \tag{4.3}$$

We can minimize in closed form with respect to each ξ_i through $\xi_i = (1 - y_i(w^\top x_i + b))_+$. With $\lambda = \frac{1}{nC}$, the problem in equation (4.3) is thus 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 ℓ_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 equation (4.3) is a linearly constrained convex optimization problem that can be analyzed using Lagrangian duality (see, e.g., Boyd and Vandenberghe, 2004). We consider nonnegative Lagrange multipliers α_i and β_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

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

As we will show in chapter 7 for all ℓ_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 α_i's equal 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 α_i's can be employed computationally (Platt, 1998), but statistically, given that in high dimensions, the number of support vectors is typically 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 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 so-called "margin bounds" where the 0–1 risk is upper-bounded by the Φ-risk. When the Φ-risk equals zero, which can occur only for problems with deterministic labels, this leads to a guarantee that the resulting classifier is the optimal one. In nondeterministic 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 that the resulting classifier is close to leading to optimal predictions.

We now study the tools dedicated to obtaining such guarantees, with, in section 4.1.3, the concept of classification calibration (making sure that minimizing the Φ-risk also leads to a minimizer of the 0–1 risk), and, in section 4.1.4, a quantitative relation between the two excess risks.

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 section 2.2.3, the Bayes risk (the best possible 0–1 expected risk) is equal to

$$\mathcal{R}^* = \mathbb{E}[\min(\eta(x), 1 - \eta(x))] = \mathbb{E}\big[\tfrac{1}{2} - \tfrac{1}{2}|\mathbb{E}[y|x]|\big],$$

and one optimal classifier is $f_*(x) = \text{sign}(2\eta(x) - 1) = \text{sign}(\mathbb{E}[y|x])$–noting that when $\eta(x) = 1/2$, our convention of random choice for the sign function is compatible with the fact that in this situation all predictions are optimal.

A key remark is that there are *many* potential other functions $g(x)$ than $2\eta(x) - 1$ so that $f_*(x) = \text{sign}(g(x))$ is optimal, namely, all functions g such that $g(x)$ has the same sign as $2\eta(x) - 1$, which leads to many possibilities beyond $2\eta(x) - 1$. In this section, we will mostly focus on functions of the form $g(x) = b(2\eta(x) - 1)$, where $b : \mathbb{R} \to \mathbb{R}$ is sign-preserving; that is, $b(u) \geqslant 0$ if $u \geqslant 0$, and $b(u) \leqslant 0$ if $u \leqslant 0$.

This section aims to ensure that the minimizers of the expected Φ-risk lead to optimal predictions by ensuring that its minimizer $g(x)$ has the same sign as $2\eta(x) - 1$.

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 optimal predictions in the population case.

General losses. To study the impact of using the Φ-risk beyond the square loss, 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). Then, knowledge of the probability $\eta(x) = \mathbb{P}(y = 1|x)$ is sufficient to characterize the optimal prediction at that x, and the ensuing excess risk.

Definition 4.1 (Conditional Φ-risk and 0–1 risk) *For $\xi \in [0, 1]$ and $u \in \mathbb{R}$, we define the conditional Φ-risk as*

$$C_\xi^\Phi(u) = \xi\Phi(u) + (1 - \xi)\Phi(-u),$$

and the conditional 0–1 risk as, with Φ_{0-1} defined in equation (4.1),

$$C_\xi(u) = \xi\Phi_{0-1}(u) + (1 - \xi)\Phi_{0-1}(-u).$$

With these definitions, we can compute the Φ-risk and 0–1 risk of a function $g : \mathcal{X} \to \mathbb{R}$ as follows:

$$\mathcal{R}_\Phi(g) = \mathbb{E}\big[C^\Phi_{\eta(x)}(g(x))\big] \quad \text{and} \quad \mathcal{R}(g) = \mathbb{E}\big[C_{\eta(x)}(g(x))\big].$$

Note that with our convention $\Phi_{0-1}(0) = \frac{1}{2}$, we have $C_{1/2}(u) = \frac{1}{2}$ for all $u \in \mathbb{R}$.

The least that 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 Φ-risk $C^\Phi_{\eta(x)}$ exactly leads to the same prediction as the Bayes predictor, which minimizes $C_{\eta(x)}$

(at least when this prediction is unique, i.e., $\eta(x) \neq \frac{1}{2}$). We thus need that for $\xi \neq \frac{1}{2}$, the minimizers of the function C_ξ^Φ are also minimizers of C_ξ.

Since the set minimizers of C_ξ is $\mathbb{R}_+^*$ when $\xi > 1/2$ (i.e., when $\eta(x) > \frac{1}{2}$, the optimal prediction at this x is $+1$), and $\mathbb{R}_-^*$ when $\xi < \frac{1}{2}$ (i.e., when $\eta(x) < \frac{1}{2}$, the optimal prediction at this x is -1), we want that for any $\xi \in [0,1] \backslash \{\frac{1}{2}\}$ (with $\mathbb{R}_+^*$ the set of strictly positive numbers, and a similar notation $\mathbb{R}_-^*$ for the set of strictly negative numbers):

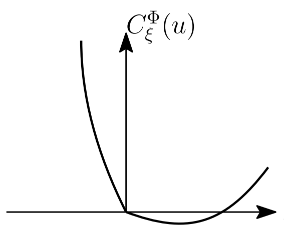
$$\text{(Positive optimal prediction)} \qquad \xi > \frac{1}{2} \quad \Leftrightarrow \quad \arg\min_{u \in \mathbb{R}} C_\xi^\Phi(u) \subset \mathbb{R}_+^* \tag{4.4}$$

$$\text{(Negative optimal prediction)} \qquad \xi < \frac{1}{2} \quad \Leftrightarrow \quad \arg\min_{u \in \mathbb{R}} C_\xi^\Phi(u) \subset \mathbb{R}_-^* \tag{4.5}$$

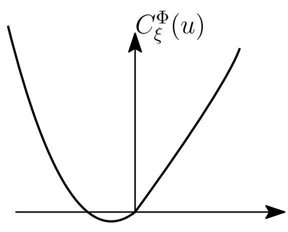
(in this discussion, we assume for simplicity that the argmins above are non-empty; degenerate cases are left as an exercise). A function Φ that satisfies these two statements is said to be *classification-calibrated*, or simply *calibrated*. The resulting binary classification method is then said "Fisher consistent." It turns out that when Φ is convex, a simple sufficient and necessary condition is available, as described in proposition 4.1.

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

Proof Since Φ is convex, so is C_ξ for any $\xi \in [0,1]$, and thus we simply consider left and right derivatives at zero to obtain conditions about the location of minimizers, with the two possibilities (a) and (b) shown next (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):



(a) $(C_\xi^\Phi)'(0_+) = \xi\Phi'(0_+) - (1-\xi)\Phi'(0_-) < 0 \;\Leftrightarrow\; \arg\min_{u\in\mathbb{R}} C_\xi^\Phi(u) \subset \mathbb{R}_+^*$



(b) $(C_\xi^\Phi)'(0_-) = \xi\Phi'(0_-) - (1-\xi)\Phi'(0_+) > 0 \;\Leftrightarrow\; \arg\min_{u\in\mathbb{R}} C_\xi^\Phi(u) \subset \mathbb{R}_-^*$

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

For the other direction of the equivalence, assume that Φ is differentiable at 0 and

$\Phi'(0) < 0$. Then $(C_\xi^\Phi)'(0) = (2\xi - 1)\Phi'(0)$; equations (4.4) and (4.5) are then direct consequences of equations (a) and (b). ■

Note that proposition 4.1 excludes the convex surrogate $u \mapsto (-u)+ = \max\{-u, 0\}$, which is not differentiable at zero. Moreover, all examples from section 4.1.1 are calibrated.

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

In the context of classification with probabilistic models, where a model for $\mathbb{P}(y = 1|x)$ is learned, calibration may also refer to the accuracy of the estimate of this probability. See Silva Filho et al. (2023) and references therein.

4.1.4 Relation 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 $\text{sign}(g(x))$, we would like to make sure that an explicit control of the excess Φ-risk (which we aim to accomplish 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 $\mathcal{R}(g) - \mathcal{R}^* \leqslant H\big[\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*\big]$, where $\mathcal{R}_\Phi^*$ is the minimum possible Φ-risk. Function H is often called the *calibration function*. This section shows that this calibration is the identity for the hinge loss (corresponding to the SVM), while it can be the square root for smooth convex surrogates such as the square and logistic losses. We will in fact look for the function $G = H^{-1}$ so that $G\big[\mathcal{R}(g) - \mathcal{R}^*\big] \leqslant \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*$, following the general frameworks of Zhang (2004b) and Bartlett et al. (2006).

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*, or, in this book, the *surrogate expected risk*.

In terms of conditional risks introduced in definition 4.1, classification-calibration meant that for all $\xi \in [0,1]\backslash\{\frac{1}{2}\}$, $\arg\min C_\xi^\Phi \subset \arg\min C_\xi$. The validity of the calibration function for the expected risk will be a consequence of the identity

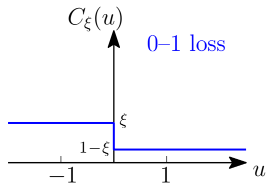
$$\forall u \in \mathbb{R},\ G\Big[C_\xi(u) - \inf_{u' \in \mathbb{R}} C_\xi(u')\Big] \ \leqslant\ C_\xi^\Phi(u) - \inf_{u' \in \mathbb{R}} C_\xi^\Phi(u'), \tag{4.6}$$

which relates the excess conditional Φ-risk and the excess 0–1 risk. Indeed, if equation (4.6) is satisfied and G is convex, then by Jensen's inequality,

$$\begin{aligned}
G\big[\mathcal{R}(g)-\mathcal{R}^*\big] &= G\Big(\mathbb{E}\Big[C_{\eta(x)}(g(x)) - \inf_{u'\in\mathbb{R}} C_{\eta(x)}(u')\Big]\Big) \\
&\leqslant \mathbb{E}\Big[G\Big(C_{\eta(x)}(g(x)) - \inf_{u'\in\mathbb{R}} C_{\eta(x)}(u')\Big)\Big] \\
&\leqslant \mathbb{E}\Big[C^{\Phi}_{\eta(x)}(g(x)) - \inf_{u'\in\mathbb{R}} C^{\Phi}_{\eta(x)}(u')\Big] = \mathcal{R}_\Phi(g) - \mathcal{R}^*_\Phi,
\end{aligned}$$

which is the desired calibration inequality.

Expression for the excess conditional 0–1 risk. For $\xi = \frac{1}{2}$, the function C_ξ is constant equal to $\frac{1}{2}$, so the corresponding excess risk is equal to zero. If $\xi > \frac{1}{2}$, then, as illustrated below, $\inf_{u'\in\mathbb{R}} C_{\eta(x)}(u') = 1-\xi$, and is attained on $\mathbb{R}^*_+$:



The excess 0–1 risk $C_\xi(u) - \inf_{u'\in\mathbb{R}} C_\xi(u')$ is equal to $2\xi-1$ if $u<0$, 0 if $u>0$, and $\xi - \frac{1}{2}$ if $u=0$, leading to

$$\forall u \in \mathbb{R},\ C_\xi(u) - \inf_{u'\in\mathbb{R}} C_\xi(u') = (2\xi-1)\Phi_{0-1}(u) \leqslant (2\xi-1)1_{u\leqslant 0}.$$

If $\xi < \frac{1}{2}$, the same reasoning leads to the quantity $(1-2\xi)\Phi_{0-1}(-u) \leqslant (1-2\xi)1_{-u\leqslant 0}$, which we can combine into, for any $\xi \in [0,1]$,

$$\forall u \in \mathbb{R},\ C_\xi(u) - \inf_{u'\in\mathbb{R}} C_\xi(u') = |2\xi-1|\cdot\Phi_{0-1}((2\xi-1)u) \leqslant |2\xi-1|\cdot 1_{(2\xi-1)u\leqslant 0}.$$

We can also obtain the more practical bound

$$\forall u \in \mathbb{R},\ C_\xi(u) - \inf_{u'\in\mathbb{R}} C_\xi(u') \leqslant |2\xi-1-b(u)|\cdot 1_{(2\xi-1)u\leqslant 0} \leqslant |2\xi-1-b(u)|, \tag{4.7}$$

for any sign-preserving function b (the inequality is true for $\xi = \frac{1}{2}$, and for $\xi > \frac{1}{2}$, it is implied by $(2\xi-1)\cdot 1_{u\leqslant 0} \leqslant (2\xi-1-b(u))\cdot 1_{u\leqslant 0}$, which is true as soon as b is sign-preserving).

Quadratic loss. For the square loss $\Phi(v) = (v-1)^2$, we have $C^\Phi_\xi(u) - \inf_{u'\in\mathbb{R}} C^\Phi_\xi(u') = (2\xi-1-u)^2$; thus, equation (4.7) with $b(u) = u$ directly leads to equation (4.6) with $G(\sigma) = \sigma^2$. Therefore,

$$\mathcal{R}(g) - \mathcal{R}(g_*) \leqslant \big(\mathcal{R}_\Phi(g) - \mathcal{R}^*_\Phi\big)^{1/2}, \tag{4.8}$$

which is a calibration result that we extend next to smooth surrogates.

Smooth surrogates. We consider smooth losses of the form (up to additive and multiplicative constants) $\Phi(u) = a(u) - u$, where $a(u) = \frac{1}{2}u^2$ for the quadratic loss, $a(u) = 2\log(e^{u/2} + e^{-u/2})$ for the logistic loss. We assume that a is even and β-smooth with $\beta > 0$ (i.e., as will be defined in chapter 5, $a''(u) \leqslant \beta$ for all $u \in \mathbb{R}$). This implies[3] that for all $u \in \mathbb{R}$ and $\alpha \in \mathbb{R}$,

$$a(u) - \alpha u - \inf_{u' \in \mathbb{R}} \big\{ a(u') - \alpha u' \big\} \geqslant \frac{1}{2\beta} |\alpha - a'(u)|^2, \tag{4.9}$$

leading to $C_\xi^\Phi(u) = \xi\Phi(u) + (1-\xi)\Phi(-u) = a(u) - (2\xi - 1)u$ and thus,

$$\begin{aligned} C_\xi^\Phi(u) - \inf_{u' \in \mathbb{R}} C_\xi^\Phi(u') &= a(u) - (2\xi-1)u - \inf_{u' \in \mathbb{R}} \big\{ a(u') - (2\xi - 1)u' \big\} \\ &\geqslant \frac{1}{2\beta} \big(2\xi - 1 - a'(u) \big)^2 \text{ by equation (4.9)}, \\ &\geqslant \frac{1}{2\beta} \Big[C_\xi(u) - \inf_{u' \in \mathbb{R}} C_\xi(u') \Big]^2 \text{ from equation (4.7)}. \end{aligned}$$

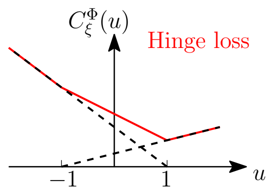
This leads to equation (4.6) with $G(\sigma) = \frac{\sigma^2}{2\beta}$, which implies

$$\mathcal{R}(g) - \mathcal{R}^* \leqslant \sqrt{2\beta} \big(\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^* \big)^{1/2}.$$

This leads to the calibration function $H(\sigma) = \sqrt{\sigma}$ for the square loss and $H(\sigma) = \sqrt{2\sigma}$ for the logistic loss (with the normalization from section 4.1.1).

Exercise 4.1 (♦) *On top of the assumptions made in this section, assume that $a(0) = 0$. Show that if a^* is the Fenchel conjugate of a, then for any function $g : \mathcal{X} \to \mathbb{R}$, we have $a^*\big(\mathcal{R}(g) - \mathcal{R}^*\big) \leqslant \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*$.*

Hinge loss. For the hinge loss, for all $\xi \in [0, 1]$, the function C_ξ^Φ is continuous piecewise affine, with kinks at -1 and $+1$. For $\xi > \frac{1}{2}$, as illustrated below, it is minimized for $u = 1$, and the excess risk is piecewise affine.



[3] Using the Fenchel conjugate $a^* : \mathbb{R} \to \mathbb{R}$, which is $(1/\beta)$-strongly convex (see chapter 5), we have $a(u) - \alpha u - \inf_{u' \in \mathbb{R}} \big\{ a(u') - \alpha u' \big\} = a^*(\alpha) - u\alpha - \inf_{\alpha' \in \mathbb{R}} \big\{ a^*(\alpha') - u\alpha' \big\} \geqslant \frac{1}{2\beta}|\alpha - \alpha'|^2$, where $\alpha' = a'(u)$ is the minimizer of $\inf_{\alpha' \in \mathbb{R}} \big\{ a^*(\alpha') - u\alpha' \big\}$ (Boyd and Vandenberghe, 2004). A direct proof may be obtained as follows: define the function $b : u \mapsto a(u) - \alpha u$, and take u' a minimizer of b; by smoothness, $b(u') \leqslant b(u - \frac{1}{\beta} b'(u)) \leqslant b(u) + b'(u)\big(- \frac{1}{\beta} b'(u)\big) + \frac{\beta}{2} \big| - \frac{1}{\beta} b'(u) \big|^2 = b(u) - \frac{1}{2\beta} b'(u)^2$, which exactly leads to equation (4.9). See also exercise 5.11.

For $\xi > \frac{1}{2}$, we can compare the excess conditional Φ-risk with the excess risk for the 0–1 conditional risk: these two are equal for $u = 0_-$ and 1, while the excess hinge risk is always larger. By symmetry, it is also true for $\xi < \frac{1}{2}$, and this is trivially true at $\xi = \frac{1}{2}$ (since the excess 0–1 risk is 0). Thus, we have shown that equation (4.6) is true for G the identity function (i.e., $H(\sigma) = \sigma$). In other words, for the hinge loss, we have $\mathcal{R}(g) - \mathcal{R}^* \leqslant \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*$; that is, the excess Φ-risk directly controls the excess 0–1 risk.

Note that only when the Bayes risk is zero (i.e., $\eta(x) \in \{0, 1\}$ almost surely), then 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 Φ-risk (indeed, the two optimal risks $\mathcal{R}^*$ and $\mathcal{R}_\Phi^*$ are equal to zero), but this is not the case otherwise.

Exercise 4.2 (♦♦) *Consider a convex function $\Phi : \mathbb{R} \to \mathbb{R}$, which is differentiable at zero with $\Phi'(0) < 0$. Define $G(z) = \Phi(0) - \inf_{u \in \mathbb{R}} \left\{ \frac{1+z}{2} \Phi(u) + \frac{1-z}{2} \Phi(-u) \right\}$. 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 nonsmooth hinge loss, the calibration function is identity, so if the excess Φ-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 Φ-risk to the excess risk (i.e., after thresholding function g at zero), the observed rates may be worse. However, as will be shown in chapter 5, smooth losses can be easier to optimize, and, for the square loss, better generalization bounds can be obtained (see section 7.6). Moreover, as shown next, the choice of surrogate also impacts the approximation error. There are, thus, multiple trade-offs between these two types of losses, and no clear superiority of one over the other.
- 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 exercise 4.3).

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 $\Phi(v) = a(v) - v$ as in this section, 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 $g : \mathcal{X} \to \mathbb{R}$.*

Impact on approximation errors (♦). For the same binary classification problem, several convex surrogates can be used. While the Bayes classifier is always the same (i.e., $f_*(x) = \operatorname{sign}(2\eta(x) - 1)$), the minimizer of the population Φ-risk will be different. For example, for the hinge loss, the minimizer $g(x)$ is exactly $\operatorname{sign}(2\eta(x) - 1)$, 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 $g(x) = 2 \operatorname{atanh}(2\eta(x) - 1)$ (with atanh the hyperbolic arc tangent; proof left as an

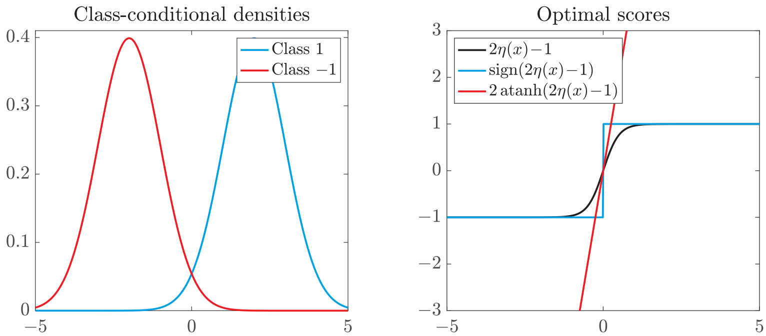


Figure 4.2. Optimal score functions for Gaussian class-conditional densities in one dimension. Left: conditional densities; right: optimal score functions for the square loss ($g_*(x) = 2\eta(x) - 1$), the hinge loss ($g_*(x) = \mathrm{sign}(2\eta(x) - 1)$), and the logistic loss ($g_*(x) = 2\,\mathrm{atanh}(2\eta(x)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

exercise). See the examples in figure 4.2, with $\mathcal{X} = \mathbb{R}$ and Gaussian class-conditional densities, showing that optimal scores (right plot) may be very different for different convex surrogates.

The choice of surrogates will have an impact since to attain the minimal Φ-risk, different assumptions are needed on the class of functions used for empirical risk minimization; that is, $\mathrm{sign}(2\eta(x)-1)$ has to be in the class of functions that we use (for the hinge loss), $2\eta(x) - 1$ for the square loss, or $2\,\mathrm{atanh}(2\eta(x) - 1)$ for the logistic loss. If these functions are not in the class of models, they have to be well-approximated, which could be harder for the hinge loss since $\mathrm{sign}(2\eta(x) - 1)$ may be less regular than $2\eta(x) - 1$ or $2\,\mathrm{atanh}(2\eta(x) - 1)$ (see also exercise 4.4 and section 14.2).

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 (LDA)." Provide an extension to the multicategory 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 that tends to zero at zero. When the space of functions chosen for g is flexible enough to reach the minimizer of $\mathcal{R}_\Phi$, such as 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 Φ-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 now 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 into two terms as follows:

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

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 ℓ_2-norm $\|\theta\|_2$, but other norms can be considered as well (such as the ℓ_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 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 (with respect to the second variable), assuming that θ_* is the minimizer of $\mathcal{R}(f_\theta)$ over $\theta \in \mathbb{R}^d$, which is assumed to exist (typically, θ_* does not belong to Θ). 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_θ. 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 nonnegative on $\mathbb{R}^d$ and can be typically upper-bounded by a specific norm (or its square) $\Omega(\theta - \theta_*)$, and we can see the first term above $\inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \inf_{\theta' \in \mathbb{R}^d} \mathcal{R}(f_{\theta'})$ as a notion of "distance" between θ_* and Θ.

  For example, if the loss that 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 first part of the approximation error is upper-bounded by G times the distance between f_{θ_*} and $\mathcal{F} = \{f_\theta,\ \theta \in \Theta\}$, for a particular pseudodistance $(\theta, \theta') \mapsto \mathbb{E}\big[|f_\theta(x) - f_{\theta'}(x)|\big]$ (missing the property of $\theta = \theta'$ being the only possibility to be zero).

  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 $\Theta = \{\theta \in \mathbb{R}^d,\ \|\theta\|_1 \leqslant D\}$ (ℓ_1-norm instead of the ℓ_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 ℓ_2-norm by D and G-Lipschitz-losses. See further applications in chapters 7 (kernel methods) and 9 (neural networks).

The estimation error is often decomposed using $g_\mathcal{F} \in \arg\min_{g \in \mathcal{F}} \mathcal{R}(g)$ as the minimizer of the expected risk for our class of models and $\hat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{\mathcal{R}}(f)$ as 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 θ proportional to θ_* and optimizing with respect to the proportionality constant.

the empirical risk:

$$\begin{aligned}
\mathcal{R}(\hat{f}) - \inf_{f\in\mathcal{F}} \mathcal{R}(f) &= \mathcal{R}(\hat{f}) - \mathcal{R}(g_{\mathcal{F}}) \\
&= \big\{\mathcal{R}(\hat{f}) - \widehat{\mathcal{R}}(\hat{f})\big\} + \big\{\widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(g_{\mathcal{F}})\big\} + \big\{\widehat{\mathcal{R}}(g_{\mathcal{F}}) - \mathcal{R}(g_{\mathcal{F}})\big\} \\
&\leqslant \sup_{f\in\mathcal{F}} \big\{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big\} + \big\{\widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(g_{\mathcal{F}})\big\} + \sup_{f\in\mathcal{F}} \big\{\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big\} \\
&\leqslant \sup_{f\in\mathcal{F}} \big\{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big\} + 0 + \sup_{f\in\mathcal{F}} \big\{\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big\} \text{ by definition of } \hat{f}. \quad (4.10)
\end{aligned}$$

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. This will also be used in section 8.1.1 on sparse methods for square loss.
- 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* ε has to be added to the estimation error considered in this section for the empirical risk minimizer (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 that follow.
- 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 gets 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 subsection, 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 i.i.d., and $\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i))$. We assume that the loss functions for all (x, y) in the support of the data generating distribution and $f \in \mathcal{F}$ are between 0 and some ℓ_∞ (for most loss functions, this is a consequence of having bounded prediction functions).[5]

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

[5]For extensions to sub-Gaussian distributions rather than distributions with bounded support, see theorem 3 in Meir and Zhang (2003).

$\delta \in (0,1)$, with probability greater than $1-\delta$,

$$\mathcal{R}(f) - \widehat{\mathcal{R}}(f) \leqslant \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}.$$

Such 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 $\frac{1}{n}\ell_\infty$.[6] Thus, applying McDiarmid's inequality (see section 1.2.2), with probability greater than $1-\delta$, we have

$$H(z_1,\dots,z_n) - \mathbb{E}[H(z_1,\dots,z_n)] \leqslant \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}.$$

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 $\frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{2}{\delta}}$, to ensure a high-probability bound.[7]

We now provide a series of bounds for 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 ℓ_2-ball constraint. We remember that in this case, $\ell(y,\theta^\top\varphi(x)) = (y-\theta^\top\varphi(x))^2$, leading to

$$\begin{aligned}\widehat{\mathcal{R}}(f) - \mathcal{R}(f) \;=\;& \theta^\top\Big(\frac{1}{n}\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top - \mathbb{E}\big[\varphi(x)\varphi(x)^\top\big]\Big)\theta \\ & -2\theta^\top\Big(\frac{1}{n}\sum_{i=1}^n y_i\varphi(x_i) - \mathbb{E}\big[y\varphi(x)\big]\Big) + \Big(\frac{1}{n}\sum_{i=1}^n y_i^2 - \mathbb{E}\big[y^2\big]\Big).\end{aligned}$$

Hence, the supremum can be upper-bounded in closed form as

$$\begin{aligned}\sup_{\|\theta\|_2\leqslant D} |\mathcal{R}(f)-\widehat{\mathcal{R}}(f)| \;\leqslant\;& D^2\Big\|\frac{1}{n}\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top - \mathbb{E}\big[\varphi(x)\varphi(x)^\top\big]\Big\|_{\rm op} \\ & +2D\Big\|\frac{1}{n}\sum_{i=1}^n y_i\varphi(x_i) - \mathbb{E}\big[y\varphi(x)\big]\Big\|_2 + \Big|\frac{1}{n}\sum_{i=1}^n y_i^2 - \mathbb{E}\big[y^2\big]\Big|,\end{aligned}$$

where $\|M\|_{\rm op}$ is the operator norm of 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 *nonuniform* expectations of deviations, and therefore of order $O(1/\sqrt{n})$, and we get an overall uniform

[6]For a fixed function $f\in\mathcal{F}$, only one term in the average is changed, with value in $[0,\ell_\infty]$, and thus a deviation of at most $\frac{1}{n}\ell_\infty$. This can be extended to the supremum by a simple computation left as an exercise.

[7]When combining two bounds in probability, the union bound leads to the term $2/\delta$ instead of $1/\delta$; for more details, see section 1.2.1.

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.

Exercise 4.6 (♦) *Provide an explicit bound on* $\sup_{\|\theta\|_2 \leqslant D} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$*, 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 following sections, 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 ℓ_∞. We can then upper-bound the uniform deviation using the union bound (as detailed in section 1.2) to get

$$\mathbb{P}\Big(\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\Big) \leqslant \sum_{f\in\mathcal{F}} \mathbb{P}\Big(\big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\Big).$$

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 (as done in section 4.4.1) to bound each $\mathbb{P}\big(\big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\big)$, leading to

$$\mathbb{P}\Big(\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \geqslant t\Big) \leqslant \sum_{f\in\mathcal{F}} 2\exp(-2nt^2/\ell_\infty^2) = 2|\mathcal{F}|\exp(-2nt^2/\ell_\infty^2).$$

Thus, by setting $\delta = 2|\mathcal{F}|\exp(-2nt^2\ell_\infty^2)$ and finding the corresponding t, with probability greater than $1-\delta$, we get (using $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$):

$$\begin{aligned}\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| &\leqslant t = \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{2|\mathcal{F}|}{\delta}} = \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log(2|\mathcal{F}|) + \log\frac{1}{\delta}} \\ &\leqslant \ell_\infty\sqrt{\frac{\log(2|\mathcal{F}|)}{2n}} + \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}},\end{aligned}$$

which is an upper bound on uniform deviations.

Exercise 4.7 (♦) *In terms of expectation, show the following (using the proof of the max of random variables from section 1.2.4, which applies because bounded random variables are sub-Gaussian):*

$$\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big|\Big] \leqslant \ell_\infty\sqrt{\frac{\log(2|\mathcal{F}|)}{2n}}.$$

Thus, according to the bound, learning is possible when the logarithm $\log(|\mathcal{F}|)$ of the number of models is significantly smaller than 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 the following subsections.

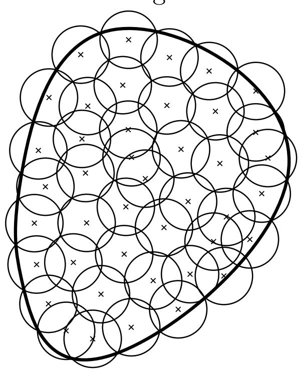
4.4.4 Beyond Finitely Many Models through Covering Numbers (♦)

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 "ε-net argument."

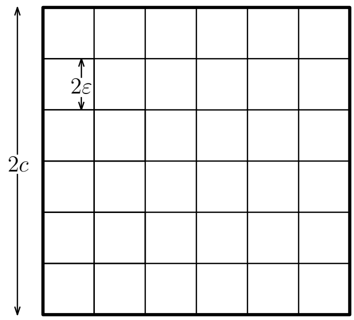
For simplicity, we assume that the loss functions are regular; for example, that they are G-Lipschitz-continuous with respect to their second argument. Then, as already seen in section 4.3, we have, for any $f, f' \in \mathcal{F}$,

$$\big|\mathcal{R}(f) - \mathcal{R}(f')\big| \leqslant G \cdot \mathbb{E}\big[|f(x) - f'(x)|\big] = G \cdot \Delta(f, f'). \tag{4.11}$$

Covering numbers. We assume that there are $m = m(\varepsilon)$ elements $f_1, \dots, f_m$ such that for any $f \in \mathcal{F}$, $\exists i \in \{1, \dots, m\}$ such that $\Delta(f, f_i) \leqslant \varepsilon$ for Δ defined in equation (4.11). The minimal possible number $m(\varepsilon)$ is the *covering number* of $\mathcal{F}$ at precision ε. See the example here in two dimensions of a covering with Euclidean balls:



The covering number $m(\varepsilon)$ is a nonincreasing function of ε. Typically, $m(\varepsilon)$ grows with ε as a power ε^{-d} when $\varepsilon \to 0$, where d is the underlying dimension. Indeed, for the ℓ_∞-metric, if (in a certain parameterization) $\mathcal{F}$ is included in a ball of radius c in the ℓ_∞-ball of dimension d, it can be easily covered by $(c/\varepsilon)^d$ cubes of length 2ε (if $c \geqslant \varepsilon$), as illustrated:



Given that all norms are equivalent in dimension d, we get the same dependence in ε^{-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 ε tends to zero. This dependence in dimension generalizes to all norms (see exercise 4.8).

Exercise 4.8 *Let $m(\varepsilon)$ be the covering number of a unit ball of $\mathbb{R}^d$ by balls of radius ε for an arbitrary norm. Using comparisons of volumes, show that $\left(\frac{1}{\varepsilon}\right)^d \leqslant m(\varepsilon) \leqslant \left(1+\frac{2}{\varepsilon}\right)^d$.*

For some sets (e.g., all Lipschitz-continuous functions with bounded Lipschitz-constant in d dimensions), $\log m(\varepsilon)$ grows faster, such as ε^{-d}. See, for instance, Wainwright (2019).

ε-net argument. Given a cover of $\mathcal{F}$, for all $f \in \mathcal{F}$, and with $(f_i)_{i\in\{1,\dots,m(\varepsilon)\}}$ being the associated cover elements, using that both $\widehat{\mathcal{R}}$ and $\mathcal{R}$ are G-Lipschitz-continuous with respect to the distance Δ, we have, for any $i \in \{1,\dots,m(\varepsilon)\}$,

$$\begin{aligned}
\big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| &\leqslant \big|\widehat{\mathcal{R}}(f) - \widehat{\mathcal{R}}(f_i)\big| + \big|\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)\big| + \big|\mathcal{R}(f_i) - \mathcal{R}(f)\big| \\
&\leqslant 2G\cdot\Delta(f,f_i) + \big|\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)\big| \\
&\leqslant 2G\cdot\Delta(f,f_i) + \sup_{j\in\{1,\dots,m(\varepsilon)\}} \big|\widehat{\mathcal{R}}(f_j) - \mathcal{R}(f_j)\big|.
\end{aligned}$$

Taking the minimum with respect to i, and using the cover property, we get

$$\big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \leqslant 2G\varepsilon + \sup_{j\in\{1,\dots,m(\varepsilon)\}} \big|\widehat{\mathcal{R}}(f_j) - \mathcal{R}(f_j)\big|$$

This implies, using section 4.4.3 that with probability greater than $1-\delta$,

$$\sup_{f\in\mathcal{F}} \big|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big| \leqslant 2G\varepsilon + \ell_\infty\sqrt{\frac{\log(2m(\varepsilon)))}{2n}} + \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}.$$

Therefore, if $m(\varepsilon) \sim \varepsilon^{-d}$, ignoring constants, we need to upper-bound the quantity $\varepsilon + \sqrt{d\log(1/\varepsilon)/n}$. The choice $\varepsilon \propto 1/\sqrt{n}$ leads 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 nonoptimal dependence on dimension and/or number of observations (see, e.g., Wainwright, 2019, for examples of these refinements).

Two powerful tools that allow sharp bounds at a reasonable cost are Rademacher complexity (Boucheron et al., 2005) and Gaussian complexity (Bartlett and Mendelson, 2002). In this book, we will focus on Rademacher complexity, which we now present.

4.5 Rademacher Complexity

We consider n i.i.d. 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 $\sup_{f\in\mathcal{F}} \{\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\}$, which is equal to

$$\sup_{h\in\mathcal{H}} \left\{ \mathbb{E}[h(z)] - \frac{1}{n}\sum_{i=1}^{n} h(z_i) \right\},$$

where $\mathbb{E}[h(z)]$ denotes the expectation with respect to a variable having the same distribution as all z_i's.

We denote the data $\mathcal{D} = \{z_1, \dots, z_n\}$, and define the *Rademacher complexity* of the class of functions $\mathcal{H}$ from $\mathcal{Z}$ to $\mathbb{R}$ as follows:

$$\mathrm{R}_n(\mathcal{H}) = \mathbb{E}_{\varepsilon,\mathcal{D}}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i h(z_i) \right], \tag{4.12}$$

where $\varepsilon \in \mathbb{R}^n$ is a vector of independent Rademacher random variables (i.e., taking values -1 or 1 with equal probability), which is also independent of $\mathcal{D}$. It is a deterministic quantity that depends only on n, $\mathcal{H}$, and the common distribution of all z_i's.

Stated in words, the Rademacher complexity is equal to the expectation of the maximal dot product between values of 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 4.9 *Show the following properties of Rademacher complexities (see Bartlett and Mendelson, 2002, for more details):*

- *If* $\mathcal{H} \subset \mathcal{H}'$, *then* $\mathrm{R}_n(\mathcal{H}) \leqslant \mathrm{R}_n(\mathcal{H}')$.
- $\mathrm{R}_n(\mathcal{H} + \mathcal{H}') = \mathrm{R}_n(\mathcal{H}) + \mathrm{R}_n(\mathcal{H}')$.
- *If* $\alpha \in \mathbb{R}$, $\mathrm{R}_n(\alpha\mathcal{H}) = |\alpha| \cdot \mathrm{R}_n(\mathcal{H})$.

- *If* $h_0 : \mathcal{Z} \to \mathbb{R}$, $\mathrm{R}_n(\mathcal{H} + \{h_0\}) = \mathrm{R}_n(\mathcal{H})$.
- $\mathrm{R}_n(\mathcal{H}) = \mathrm{R}_n(\text{convex hull}(\mathcal{H}))$.

Exercise 4.10 (Massart's lemma) *Assume that* $\mathcal{H} = \{h_1, \dots, h_m\}$, *and almost surely we have the bound* $\frac{1}{n}\sum_{i=1}^n h_j(x_i)^2 \leqslant R^2$ *for all* $j \in \{1, \dots, m\}$. *Show that 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 equation (4.12), 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 $\{1, \dots, n\}$, $\mathbb{E}[h(z'_i)|\mathcal{D}] = \mathbb{E}[h(z)]$ and $\mathbb{E}[h(z_i)|\mathcal{D}] = h(z_i)$, 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 ε_i and $h(z_i') - h(z_i)$, to get

$$\begin{aligned}
& \mathbb{E}\Big[\sup_{h\in\mathcal{H}}\Big\{\mathbb{E}[h(z)] - \frac{1}{n}\sum_{i=1}^n h(z_i)\Big\}\Big] \\
\leqslant\ & \mathbb{E}\Big[\sup_{h\in\mathcal{H}}\Big\{\frac{1}{n}\sum_{i=1}^n \varepsilon_i\big(h(z_i') - h(z_i)\big)\Big\}\Big] \\
\leqslant\ & \mathbb{E}\Big[\sup_{h\in\mathcal{H}}\Big\{\frac{1}{n}\sum_{i=1}^n \varepsilon_i\big(h(z_i)\big)\Big\}\Big] + \mathbb{E}\Big[\sup_{h\in\mathcal{H}}\Big\{\frac{1}{n}\sum_{i=1}^n \varepsilon_i\big(-h(z_i)\big)\Big\}\Big] \\
=\ & 2\mathbb{E}\Big[\sup_{h\in\mathcal{H}}\Big\{\frac{1}{n}\sum_{i=1}^n \varepsilon_i h(z_i)\Big\}\Big] = 2\mathrm{R}_n(\mathcal{H}).
\end{aligned}$$

The reasoning is identical for $\mathbb{E}\left[\sup_{h\in\mathcal{H}}\left\{\frac{1}{n}\sum_{i=1}^n h(z_i) - \mathbb{E}[h(z)]\right\}\right] \leqslant 2\mathrm{R}_n(\mathcal{H})$. ■

Proposition 4.2 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 4.11 (♦) *The Gaussian complexity of a class of functions $\mathcal{H}$ from $\mathcal{Z}$ to $\mathbb{R}$ is defined as $\mathrm{G}_n(\mathcal{H}) = \mathbb{E}_{\varepsilon,\mathcal{D}}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i h(z_i)\right]$, where $\varepsilon \in \mathbb{R}^n$ is a vector of independent Gaussian variables with mean zero and variance 1. Show that (1) $\mathrm{R}_n(\mathcal{H}) \leqslant \sqrt{\frac{\pi}{2}} \cdot \mathrm{G}_n(\mathcal{H})$ and (2) $\mathrm{G}_n(\mathcal{H}) \leqslant \sqrt{2\log(2n)} \cdot \mathrm{R}_n(\mathcal{H})$.*

Empirical Rademacher complexities (♦). The Rademacher complexity $\mathrm{R}_n(\mathcal{H})$ defined in equation (4.12) is a *deterministic* quantity that depends on the distribution of inputs. When using bounds in high probability through McDiarmid's inequality in section 4.4.1, we obtained that if $h(z) \in [0, \ell_\infty]$ for all $h \in \mathcal{H}$, then with probability greater than $1 - \delta$, for all $h \in \mathcal{H}$,

$$\mathbb{E}[h(z)] \leqslant \frac{1}{n}\sum_{i=1}^n h(z_i) + 2\mathrm{R}_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}.$$

While we provide 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,

$$\hat{\mathrm{R}}_n(\mathcal{H}) = \mathbb{E}_\varepsilon\Big[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i h(z_i)\Big], \tag{4.13}$$

which is now a random quantity that is computable from the training data and the class of functions. We can also use McDiarmid'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 before, that is,

$$\mathbb{E}[h(z)] \leqslant \frac{1}{n}\sum_{i=1}^{n} h(z_i) + 2\hat{\mathrm{R}}_n(\mathcal{H}) + 3\frac{\ell_\infty}{\sqrt{2n}}\sqrt{\log\frac{2}{\delta}}, \tag{4.14}$$

which is now computable (if one can compute the empirical Rademacher complexity). Note that the factor of $3 = 1 + 2$ comes from applying McDiarmid's inequality twice, once for $\sup_{h\in\mathcal{H}}\left\{\mathcal{R}(h) - \widehat{\mathcal{R}}(f)\right\}$ and once for $\hat{\mathrm{R}}_n(\mathcal{H})$ (with then an extra factor of 2 since it appears as $2\hat{\mathrm{R}}_n(\mathcal{H})$). Empirical Rademacher complexities are data-dependent complexity estimates that can be used for model selection (see section 4.6).

4.5.2 Lipschitz-Continuous Losses

A particularly appealing property in this context is shown in proposition 4.3 and is sometimes called the "contraction principle," using a simple proof from lemma 5 in Meir and Zhang (2003); see also section 4.5 in Ledoux and Talagrand (1991). See proposition 4.4 for a similar result for the Rademacher complexity defined with absolute values (and then with an extra factor of 2), and section 13.1.3 for extensions to vector-valued prediction functions.

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 $i = 1, \ldots, n$, we have, for $\varepsilon \in \mathbb{R}^n$, a vector of independent Rademacher random variables:*

$$\mathbb{E}_\varepsilon\left[\sup_{\theta\in\Theta}\left\{b(\theta) + \sum_{i=1}^{n}\varepsilon_i\varphi_i(a_i(\theta))\right\}\right] \leqslant \mathbb{E}_\varepsilon\left[\sup_{\theta\in\Theta}\left\{b(\theta) + \sum_{i=1}^{n}\varepsilon_i a_i(\theta)\right\}\right].$$

Proof (♦) 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 $\mathbb{E}_{\varepsilon_1,\ldots,\varepsilon_{n+1}}\left[\sup_{\theta\in\Theta}\left\{b(\theta)+\sum_{i=1}^{n+1}\varepsilon_i\varphi_i(a_i(\theta))\right\}\right]$ and compute the expectation with respect to ε_{n+1} explicitly, by considering the two potential values with probability 1/2:

$$\begin{aligned}
&\mathbb{E}_{\varepsilon_1,\ldots,\varepsilon_{n+1}}\left[\sup_{\theta\in\Theta}\left\{b(\theta) + \sum_{i=1}^{n+1}\varepsilon_i\varphi_i(a_i(\theta))\right\}\right] \\
&= \frac{1}{2}\mathbb{E}_{\varepsilon_1,\ldots,\varepsilon_n}\left[\sup_{\theta\in\Theta}\left\{b(\theta)+\sum_{i=1}^{n}\varepsilon_i\varphi_i(a_i(\theta)) + \varphi_{n+1}(a_{n+1}(\theta))\right\}\right] \\
&\qquad + \frac{1}{2}\mathbb{E}_{\varepsilon_1,\ldots,\varepsilon_n}\left[\sup_{\theta\in\Theta}\left\{b(\theta)+\sum_{i=1}^{n}\varepsilon_i\varphi_i(a_i(\theta)) - \varphi_{n+1}(a_{n+1}(\theta))\right\}\right],
\end{aligned}$$

which is equal to

$$\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\bigg[\sup_{\theta,\theta'\in\Theta}\bigg\{\frac{b(\theta)+b(\theta')}{2} + \sum_{i=1}^n \varepsilon_i \frac{\varphi_i(a_i(\theta))+\varphi_i(a_i(\theta'))}{2} + \frac{\varphi_{n+1}(a_{n+1}(\theta))-\varphi_{n+1}(a_{n+1}(\theta'))}{2}\bigg\}\bigg],$$

by assembling the terms. By taking the supremum over (θ,θ') and (θ',θ), we get

$$\begin{aligned}
&\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\bigg[\sup_{\theta,\theta'\in\Theta}\bigg\{\frac{b(\theta)+b(\theta')}{2} + \sum_{i=1}^n \varepsilon_i \frac{\varphi_i(a_i(\theta))+\varphi_i(a_i(\theta'))}{2} + \frac{|\varphi_{n+1}(a_{n+1}(\theta))-\varphi_{n+1}(a_{n+1}(\theta'))|}{2}\bigg\}\bigg] \\
&\leqslant \mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\bigg[\sup_{\theta,\theta'\in\Theta}\bigg\{\frac{b(\theta)+b(\theta')}{2} + \sum_{i=1}^n \varepsilon_i \frac{\varphi_i(a_i(\theta))+\varphi_i(a_i(\theta'))}{2} + \frac{|a_{n+1}(\theta)-a_{n+1}(\theta')|}{2}\bigg\}\bigg],
\end{aligned}$$

using Lipschitz continuity of φ_{n+1}. We can redo the same sequence of *equalities* with φ_{n+1} being the identity to obtain that the last expression is equal to

$$\begin{aligned}
&\mathbb{E}_{\varepsilon_{n+1}}\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n}\bigg[\sup_{\theta\in\Theta}\bigg\{b(\theta)+\varepsilon_{n+1}a_{n+1}(\theta)+\sum_{i=1}^n\varepsilon_i\varphi_i(a_i(\theta))\bigg\}\bigg] \\
\leqslant\ &\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n,\varepsilon_{n+1}}\bigg[\sup_{\theta\in\Theta}\bigg\{b(\theta)+\varepsilon_{n+1}a_{n+1}(\theta)+\sum_{i=1}^n\varepsilon_i a_i(\theta)\bigg\}\bigg] \text{ by the induction hypothesis,}
\end{aligned}$$

which leads to the desired result. ■

We can apply this contraction principle to our 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 $a_i(\theta)=\theta_i$, $\varphi_i(u_i)=\ell(y_i,u_i)$),

$$\mathbb{E}_\varepsilon\bigg[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\ell(y_i,f(x_i))\ \bigg|\ \mathcal{D}\bigg] \quad\leqslant\quad G\cdot\mathbb{E}_\varepsilon\bigg[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i f(x_i)\ \bigg|\ \mathcal{D}\bigg],$$

which leads to

$$\mathrm{R}_n(\mathcal{H}) \leqslant G\cdot\mathrm{R}_n(\mathcal{F}). \tag{4.15}$$

Thus, the Rademacher complexity of the class of prediction functions controls the uniform deviations of the empirical risk. We consider simple examples in section 4.5.3 but give before, without proof, a contraction result that we will need in section 9.2.3 (see proof of theorem 4.12 in Ledoux and Talagrand, 1991), with an extra factor of 2.

Proposition 4.4 (Contraction principle–absolute values) *Given any functions $a_i : \Theta \to \mathbb{R}$ (without further assumption) and any 1-Lipschitz-functions $\varphi_i : \mathbb{R} \to \mathbb{R}$ 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,*

$$\mathbb{E}_\varepsilon \bigg[\sup_{\theta \in \Theta} \bigg| \sum_{i=1}^n \varepsilon_i \varphi_i(a_i(\theta)) \bigg| \bigg] \leqslant 2\, \mathbb{E}_\varepsilon \bigg[\sup_{\theta \in \Theta} \bigg| \sum_{i=1}^n \varepsilon_i a_i(\theta) \bigg| \bigg].$$

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 Ω is a norm on $\mathbb{R}^d$. We denote the design matrix by $\Phi \in \mathbb{R}^{n \times d}$. We have (with expectations with respect to both ε 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 Ω. For example, when Ω is the ℓ_p-norm, with $p \in [1, \infty]$, then Ω^* is the ℓ_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

$$\begin{aligned} \mathrm{R}_n(\mathcal{F}) &= \frac{D}{n} \mathbb{E}\left[\|\Phi^\top \varepsilon\|_2 \right] \leqslant \frac{D}{n} \sqrt{\mathbb{E}\left[\|\Phi^\top \varepsilon\|_2^2 \right]} \text{ by Jensen's inequality,} \\ &\leqslant \frac{D}{n} \sqrt{\mathbb{E}\left[\mathrm{tr}[\Phi^\top \varepsilon \varepsilon^\top \Phi] \right]} = \frac{D}{n} \sqrt{\mathbb{E}\left[\mathrm{tr}[\Phi^\top \Phi] \right]}, \text{ using that } \mathbb{E}[\varepsilon \varepsilon^\top] = I, \\ &= \frac{D}{n} \sqrt{\sum_{i=1}^n \mathbb{E}\big[(\Phi \Phi^\top)_i \big]} = \frac{D}{n} \sqrt{\sum_{i=1}^n \mathbb{E}\big[\|\varphi(x_i)\|_2^2 \big]} = \frac{D}{\sqrt{n}} \sqrt{\mathbb{E}\big[\|\varphi(x)\|_2^2 \big]}. \end{aligned} \tag{4.16}$$

We thus obtain a *dimension-independent* Rademacher complexity that we will use in the summary in section 4.5.4. While $\mathbb{E}\big[\|\varphi(x)\|_2^2 \big]$ can be quite large in practice, the lack of explicit dependence in dimension makes it possible to consider infinite-dimensional feature vectors, so long as this quantity is controlled.

Exercise 4.12 (ℓ_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 $RD\big(\frac{2 \log(2d)}{n}\big)^{1/2}$.*

Exercise 4.13 (♦) *Let $p \in (1, 2]$, 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 1.25). Recover the result of exercise 4.12 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 section 4.5.3 to obtain a bound on the estimation error. We then look at the approximation error.

Estimation error. With all the elements discussed previously, 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–linear predictions) *Assume that the loss function is G-Lipschitz-continuous, with a set of linear prediction functions $\mathcal{F} = \{f_\theta(x) = \theta^\top\varphi(x),\ \|\theta\|_2 \leqslant D\}$, 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 proposition 4.2 to relate the uniform deviation to the Rademacher average, equation (4.15) to take care of the Lipschitz-continuous loss, and equation (4.16) to account for the ℓ_2-norm constraint, we get the desired result. Note that the factor of 4 comes from symmetrization (proposition 4.2, which leads to a factor of 2), and equation (4.10) in section 4.4 (which leads to another factor of 2). ■

Approximation error. If we assume that there is a minimizer θ_* of $\mathcal{R}(f_\theta)$ over $\mathbb{R}^d$, the approximation error (of using a ball of θ rather than the whole $\mathbb{R}^d$) is upper-bounded by, following derivations from section 4.3 and using Cauchy-Schwarz and Jensen's inequalities,

$$\begin{aligned}\inf_{\|\theta\|_2 \leqslant D} \mathcal{R}(f_\theta) - \mathcal{R}(f_{\theta_*}) &\leqslant G \inf_{\|\theta\|_2 \leqslant D} \mathbb{E}\big[|f_\theta(x) - f_{\theta_*}(x)|\big] = G \inf_{\|\theta\|_2 \leqslant D} \mathbb{E}\big[|\varphi(x)^\top(\theta - \theta_*)|\big] \\ &\leqslant G \inf_{\|\theta\|_2 \leqslant D} \|\theta - \theta_*\|_2 \cdot \mathbb{E}\big[\|\varphi(x)\|_2\big] \leqslant GR \inf_{\|\theta\|_2 \leqslant D} \|\theta - \theta_*\|_2.\end{aligned}$$

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 knowing $\|\theta_*\|_2$, which is not possible in practice (see section 4.5.5 for penalized problems, which do not have this issue). 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 4.14 *Consider a learning problem with 1-Lipschitz-continuous loss (with respect to the second variable), a function class $f_\theta(x) = \theta^\top \varphi(x)$, $\|\theta\|_1 \leqslant D$, 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 $\mathbb{E}\big[\mathcal{R}(f_{\hat{\theta}})\big] \leqslant \inf_{\|\theta\|_1 \leqslant D} \mathcal{R}(f_\theta) + 4RD\sqrt{2\log(2d)/n}$, for the constrained empirical risk minimizer $f_{\hat{\theta}}$.*

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. We first consider the squared ℓ_2-norm in this section. Moreover, we use the (overloaded) notation $\mathcal{R}(\theta) = \mathcal{R}(f_\theta)$ and $\widehat{\mathcal{R}}(\theta) = \widehat{\mathcal{R}}(f_\theta)$ as we only consider the same linear predictors $f_\theta = \varphi(\cdot)^\top \theta$, for $\theta \in \mathbb{R}^d$.

We now denote $\hat{\theta}_\lambda$ as a minimizer of

$$\widehat{\mathcal{R}}(\theta) + \frac{\lambda}{2}\|\theta\|_2^2 \;=\; \widehat{\mathcal{R}}(f_\theta) + \frac{\lambda}{2}\|\theta\|_2^2. \tag{4.17}$$

If the loss function is always nonnegative, then $\frac{\lambda}{2}\|\hat{\theta}_\lambda\|_2^2 \leqslant \widehat{\mathcal{R}}(\hat{\theta}_\lambda) + \frac{\lambda}{2}\|\hat{\theta}_\lambda\|_2^2 \leqslant \widehat{\mathcal{R}}(0)$, leading to a bound $\|\hat{\theta}_\lambda\|_2 = O(1/\sqrt{\lambda})$. Thus, with $D = O(1/\sqrt{\lambda})$ in the bound of proposition 4.5, this leads to an excess risk in $O(1/\sqrt{\lambda n})$, which is not optimal.

We now give a stronger result using the strong convexity of the squared ℓ_2-norm (with now a convex loss), adapted from Sridharan et al. (2009) and Bartlett et al. (2005).

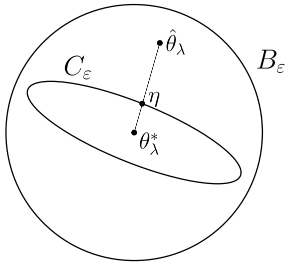
Proposition 4.6 (Fast rates for regularized objectives) *Assume the loss function is G-Lipschitz-continuous and convex in the second argument, with linear prediction functions $x \mapsto \theta^\top \varphi(x)$ for $\theta \in \mathbb{R}^d$, where $\|\varphi(x)\|_2 \leqslant R$ almost surely. Let $\hat{\theta}_\lambda \in \mathbb{R}^d$ be the minimizer of the regularized empirical risk in equation (4.17); then*

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_\lambda)\big] \leqslant \inf_{\theta \in \mathbb{R}^d} \Big\{\mathcal{R}(\theta) + \frac{\lambda}{2}\|\theta\|_2^2\Big\} + \frac{24G^2R^2}{\lambda n}.$$

Proof (♦♦) For this proof, we use the notation $\mathcal{R}_\lambda(\theta) = \mathcal{R}(\theta) + \frac{\lambda}{2}\|\theta\|_2^2$, with minimum value $\mathcal{R}_\lambda^*$ attained at θ_λ^* (which is unique by strong convexity). We also use the notation $\widehat{\mathcal{R}}_\lambda(\theta) = \widehat{\mathcal{R}}(\theta) + \frac{\lambda}{2}\|\theta\|_2^2$.

We consider the convex set $C_\varepsilon = \{\theta \in \mathbb{R}^d,\ \mathcal{R}_\lambda(\theta) - \mathcal{R}_\lambda^* \leqslant \varepsilon\}$ of ε-optimal predictors, for an $\varepsilon > 0$ to be chosen later. By strong convexity (see section 5.2.3 for more details), we have $\mathcal{R}_\lambda(\theta) - \mathcal{R}_\lambda^* \geqslant \frac{\lambda}{2}\|\theta - \theta_\lambda^*\|_2^2$ for all $\theta \in \mathbb{R}^d$, and thus C_ε is included in the ℓ_2-ball B_ε with center θ_λ^* and radius $\sqrt{2\varepsilon/\lambda}$.

The proof works as follows: if $\hat{\theta}_\lambda \notin C_\varepsilon$, then we can find η in the segment joining θ^*_λ and $\hat{\theta}_\lambda$, which is in the boundary of C_ε (i.e., such that $\mathcal{R}_\lambda(\eta) - \mathcal{R}^*_\lambda = \varepsilon$). Because $\eta \in C_\varepsilon \subset B_\varepsilon$, it cannot be too far from θ^*_λ, as $\|\eta - \theta^*_\lambda\|_2 \leqslant \sqrt{2\varepsilon/\lambda}$.



However, using uniform deviation bounds, uniformly on B_ε, $|\widehat{\mathcal{R}}_\lambda - \mathcal{R}_\lambda|$ will be, with high probability, less than a constant times $GR\sqrt{\frac{\varepsilon}{\lambda n}}$, leading to a contradiction if this quantity is greater than a constant times ε, which occurs when $\varepsilon \propto \frac{G^2R^2}{\lambda n}$, which is the desired scaling. We now make this reasoning precise.

On the segment $[\theta^*_\lambda, \hat{\theta}_\lambda]$ we thus choose η exactly on the boundary of C_ε, that is, such that $\mathcal{R}_\lambda(\eta) - \mathcal{R}^*_\lambda = \varepsilon$. Moreover, by convexity of the empirical risk $\widehat{\mathcal{R}}$, we have $\widehat{\mathcal{R}}_\lambda(\eta) \leqslant \max\big\{\widehat{\mathcal{R}}_\lambda(\theta^*_\lambda), \widehat{\mathcal{R}}_\lambda(\hat{\theta}_\lambda)\big\} = \widehat{\mathcal{R}}_\lambda(\theta^*_\lambda)$. This implies that

$$\mathcal{R}_\lambda(\eta) - \widehat{\mathcal{R}}_\lambda(\eta) + \widehat{\mathcal{R}}_\lambda(\theta^*_\lambda) - \mathcal{R}_\lambda(\theta^*_\lambda) = \big\{\mathcal{R}_\lambda(\eta) - \mathcal{R}_\lambda(\theta^*_\lambda)\big\} + \big\{\widehat{\mathcal{R}}_\lambda(\theta^*_\lambda) - \widehat{\mathcal{R}}_\lambda(\eta)\big\} \geqslant \varepsilon. \quad (4.18)$$

Using the uniform deviation bound on an ℓ_2-ball of radius $\sqrt{2\varepsilon/\lambda}$, derived in section 4.5.3 based on Rademacher averages,

$$\mathbb{E}\Big[\sup_{\xi \in B_\varepsilon} \big\{\mathcal{R}(\xi) - \widehat{\mathcal{R}}(\xi)\big\}\Big] \leqslant 2 \cdot \frac{GR}{\sqrt{n}} \cdot \sqrt{2\varepsilon/\lambda}\,.$$

This implies for $A = \sup_{\xi \in B_\varepsilon} \big\{\mathcal{R}_\lambda(\xi) - \mathcal{R}_\lambda(\theta^*_\lambda) - \big[\widehat{\mathcal{R}}_\lambda(\xi) - \widehat{\mathcal{R}}_\lambda(\theta^*_\lambda)\big]\big\}$, that its expectation satisfies $\mathbb{E}[A] \leqslant 2\frac{GR}{\sqrt{n}}\sqrt{2\varepsilon/\lambda} + \mathbb{E}[\widehat{\mathcal{R}}_\lambda(\theta^*_\lambda) - \mathcal{R}_\lambda(\theta^*_\lambda)] = 2\frac{GR}{\sqrt{n}}\sqrt{2\varepsilon/\lambda}$. We can now apply McDiarmid's inequality (as in section 4.4.1) since changing a single data point leads to an absolute difference $\frac{2}{n}GR\sqrt{2\varepsilon/\lambda}$, leading to, with probability greater than $1-\delta$,

$$A \leqslant \mathbb{E}[A] + \frac{2}{n}GR\sqrt{2\varepsilon/\lambda}\sqrt{\frac{n}{2}}\sqrt{\log\frac{1}{\delta}} \leqslant \frac{GR\sqrt{2\varepsilon/\lambda}}{\sqrt{n}}\Big(2 + \sqrt{2\log\frac{1}{\delta}}\Big).$$

We thus get a contradiction when $\varepsilon > A$ (since $\eta \in C_\varepsilon$ implies that $A \geqslant \varepsilon$ because of equation (4.18)), that is, we can take, in the limit $\varepsilon \to A$ (which leads to an equation in ε that can be solved):

$$\varepsilon = \bigg[\frac{GR\sqrt{2/\lambda}}{\sqrt{n}}\Big(2 + \sqrt{2\log\frac{1}{\delta}}\Big)\bigg]^2,$$

leading to the bound which is valid with probability greater than $1-\delta$:

$$\mathcal{R}(\hat{\theta}_\lambda) - \mathcal{R}^*_\lambda \leqslant \bigg[\frac{GR\sqrt{2/\lambda}}{\sqrt{n}}\Big(2 + \sqrt{2\log\frac{1}{\delta}}\Big)\bigg]^2 \leqslant \frac{2G^2R^2}{\lambda n}\Big(8 + 4\log\frac{1}{\delta}\Big).$$

By integration of the bound,[8] we obtain the desired result. ■

Note that we obtain a fast rate in $O(R^2/(\lambda n))$, which has a better dependence in n but depends on λ, which can be very small in practice. One classical choice of λ 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 section 3.6 for ridge (least-squares) regression, but now for all Lipschitz-continuous losses. Note that the amount of regularization to get the result discussed here still depends on the unknown quantity $\|\theta_*\|_2$. Next, 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 4.15 (♦♦) *Extend the result in proposition 4.6 to features that are almost surely bounded in the ℓ_p-norm by R, and a regularizer ψ that is strongly convex with respect to the ℓ_p-norm; that is, such that for all $\theta, \eta \in \mathbb{R}^d$, $\psi(\theta) \geqslant \psi(\eta) + \psi'(\eta)^\top(\theta-\eta) + \frac{\mu}{2}\|\theta-\eta\|_p^2$, for some $\mu > 0$, where $\psi'(\eta)$ is a subgradient of ψ at η. Hint: use exercise 4.13.*

Norm-penalized estimation (♦♦). While proposition 4.6 considered squared ℓ_2-norm penalization and relied on specific properties of the ℓ_2-norm, we now consider penalization by *any* nonsquared norm. That is, we now focus on the following objective function:

$$\widehat{\mathcal{R}}_\lambda(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, \theta^\top \varphi(x_i)) + \lambda\Omega(\theta) \;=\; \widehat{\mathcal{R}}(\theta) + \lambda\Omega(\theta), \tag{4.19}$$

for any norm Ω on $\mathbb{R}^d$, with Ω^* denoting the dual norm. Proposition 4.7 provides an estimation rate in $O(1/\sqrt{n})$, with a proof that is similar to the one of proposition 4.6.

Proposition 4.7 (Norm-penalized estimation) *Assume that the unregularized risk $\mathcal{R}(\theta) = \mathbb{E}\big[\ell(y, \theta^\top\varphi(x))\big]$ is minimized at some $\theta_* \in \mathbb{R}^d$, the function $\theta \mapsto \ell(y, \theta^\top\varphi(x))$ is convex and GR-Lipschitz continuous on the set $\{\theta \in \mathbb{R}^d,\ \Omega(\theta) \leqslant 2\Omega(\theta_*)\}$, and $\Omega^*(\varphi(x)) \leqslant R$ 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 the regularization parameter $\lambda = \frac{4GR}{\sqrt{n}}\big(\rho_\Omega + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\big)$, with probability at least $1-\delta$, any minimizer $\hat{\theta}_\lambda$ of equation (4.19) satisfies:*

$$\mathcal{R}(\hat{\theta}_\lambda) \leqslant \mathcal{R}(\theta_*) + \Omega(\theta_*)\frac{8GR}{\sqrt{n}}\bigg(\rho_\Omega + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\bigg).$$

Proof (♦♦) Let θ_λ^* be a minimizer of the population regularized risk $\mathcal{R}_\lambda(\theta)$ defined as $\mathcal{R}_\lambda(\theta) = \mathcal{R}(\theta) + \lambda\Omega(\theta)$. As in the proof of proposition 4.6, we consider the set $C_\varepsilon =$

[8] We use the following lemma: if Z is a nonnegative random variable such that $Z \leqslant u + v\log\frac{1}{\delta}$ with probability at least $1-\delta$ for all $\delta \in (0,1)$, then $\mathbb{E}[Z] \leqslant u + v$.

$\{\theta \in \mathbb{R}^d,\ \mathcal{R}_\lambda(\theta) - \mathcal{R}_\lambda(\theta^*_\lambda) \leqslant \varepsilon\}$ of ε-optimal predictors, with an ε to be chosen later. We first show that C_ε is included in the set $B_\varepsilon = \{\theta \in \mathbb{R}^d,\ \Omega(\theta) \leqslant \Omega(\theta_*) + \varepsilon/\lambda\}$. This is a consequence of the following series of inequalities that are using the optimality of θ_* for $\mathcal{R}$ and of θ^*_λ for $\mathcal{R}_\lambda$: if $\theta \in C_\varepsilon$,

$$\mathcal{R}(\theta) + \lambda\Omega(\theta) \ \leqslant\ \mathcal{R}(\theta^*_\lambda) + \lambda\Omega(\theta^*_\lambda) + \varepsilon \ \leqslant\ \mathcal{R}(\theta_*) + \lambda\Omega(\theta_*) + \varepsilon \ \leqslant\ \mathcal{R}(\theta) + \lambda\Omega(\theta_*) + \varepsilon,$$

which implies that $\theta \in B_\varepsilon$. We are thus in the same setup as the proof of proposition 4.6 (and thus the same illustrative plot), but with a set B_ε that only imposes that θ is bounded, not that θ is closed to θ^*_λ. We set $\varepsilon = \lambda\Omega(\theta_*)$ so that we have $B_\varepsilon = \{\theta \in \mathbb{R}^d,\ \Omega(\theta) \leqslant 2\Omega(\theta_*)\}$, with λ to be determined next.

We now show that with high probability, we must have $\hat\theta_\lambda \in C_\varepsilon$. If $\hat\theta_\lambda \notin C_\varepsilon$, since $\theta^*_\lambda \in C_\varepsilon$, there is an element η in the segment $[\theta^*_\lambda, \hat\theta_\lambda]$ which is in the boundary of C_ε; that is, so that $\mathcal{R}_\lambda(\eta) - \mathcal{R}_\lambda(\theta^*_\lambda) = \varepsilon$. Since the empirical risk is convex, we have $\widehat{\mathcal{R}}_\lambda(\eta) \leqslant \max\{\widehat{\mathcal{R}}_\lambda(\theta^*_\lambda), \widehat{\mathcal{R}}_\lambda(\hat\theta_\lambda)\} = \widehat{\mathcal{R}}_\lambda(\theta^*_\lambda)$. Thus,

$$\begin{aligned}\widehat{\mathcal{R}}(\theta^*_\lambda) - \widehat{\mathcal{R}}(\eta) - \mathcal{R}(\theta^*_\lambda) + \mathcal{R}(\eta) &= \{\widehat{\mathcal{R}}_\lambda(\theta^*_\lambda) - \widehat{\mathcal{R}}_\lambda(\eta)\} + \{\mathcal{R}_\lambda(\eta) - \mathcal{R}_\lambda(\theta^*_\lambda)\} \\ &\geqslant \mathcal{R}_\lambda(\eta) - \mathcal{R}_\lambda(\theta^*_\lambda) = \varepsilon. \end{aligned} \tag{4.20}$$

In order to derive uniform deviation bounds, we notice that ρ_Ω is such that the Rademacher complexity of the set of loss functions for linear predictors such that $\Omega(\theta) \leqslant 2\Omega(\theta_*)$, is less than $\frac{1}{\sqrt{n}}\rho_\Omega GR \cdot 2\Omega(\theta_*)$ (see section 4.5.3). Thus, using McDiarmid's inequality as in section 4.4.1, with probability greater than $1-\delta$, for all θ such that $\Omega(\theta) \leqslant 2\Omega(\theta_*)$,

$$\mathcal{R}(\theta) - \mathcal{R}(\theta^*_\lambda) \leqslant \widehat{\mathcal{R}}(\theta) - \widehat{\mathcal{R}}(\theta^*_\lambda) + \frac{4\rho_\Omega GR\Omega(\theta_*)}{\sqrt{n}} + \frac{2GR\Omega(\theta_*)}{\sqrt{n}}\sqrt{2\log\frac{1}{\delta}}.$$

Thus, if we choose λ so that $\varepsilon = \lambda\Omega(\theta_*) > \frac{4\rho_\Omega GR\Omega(\theta_*)}{\sqrt{n}} + \frac{2GR\Omega(\theta_*)}{\sqrt{n}}\sqrt{2\log\frac{1}{\delta}}$, we obtain a contradiction to equation (4.20) for $\theta = \eta$. Thus, with such a λ, with probability at least $1-\delta$, we have $\hat\theta_\lambda \in C_\varepsilon$; that is,

$$\mathcal{R}_\lambda(\hat\theta_\lambda) - \mathcal{R}_\lambda(\theta^*_\lambda) \leqslant \varepsilon = \lambda\Omega(\theta_*). \tag{4.21}$$

Then, by taking the limiting $\lambda = \frac{4GR}{\sqrt{n}}\big(\rho_\Omega + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\big)$, we get:

$$\begin{aligned}\mathcal{R}(\hat\theta_\lambda) &\leqslant \mathcal{R}_\lambda(\hat\theta_\lambda) \ \leqslant\ \mathcal{R}_\lambda(\theta^*_\lambda) + \lambda\Omega(\theta_*) \text{ from equation (4.21)}, \\ &\leqslant \mathcal{R}_\lambda(\theta_*) + \lambda\Omega(\theta_*) \ =\ \mathcal{R}(\theta_*) + 2\lambda\Omega(\theta_*) \\ &\leqslant \mathcal{R}(\theta_*) + \Omega(\theta_*)\frac{8GR}{\sqrt{n}}\Big(\rho_\Omega + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\Big),\end{aligned}$$

which is the desired result. Note that the key element in this proposition is that the value of λ does not depend on $\Omega(\theta_*)$. ■

Proposition 4.7 can be applied to most of the losses and norms we consider in this book. For example, for the ℓ_2-norm, we have $\rho_\Omega = 1$, while for the ℓ_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 $G = \sigma + 3R\Omega(\theta^*)$ (proof left as an exercise). See section 8.3.6 for an alternative proof framework for the square loss.

4.5.6 Extensions and Improvements

In this chapter, we have focused on the simplest situations for empirical risk minimization: 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 out 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 explored in exercise 4.3 (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 4.16).

  Exercise 4.16 (♦) *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 $f, g : \mathcal{X} \to \mathcal{Y}$ of the learning algorithm on datasets of n observations that differ by a single observation, $|\ell(y, f(x)) - \ell(y, g(x))| \leqslant \beta_n$, 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 β_n. With the same assumptions as in proposition 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 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).
- **Semisupervised learning:** In many applications, many unlabeled observations are available (i.e., only with the input x being available). To take advantage of the abundance of unlabeled data, some assumptions are typically made to show an improvement in 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 $[0, \ell_\infty]$, we have obtained in section 4.4.1, together with the Rademacher complexities in section 4.5, a bound of the form

$$\sup_{f \in \mathcal{F}} \big|\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big| \leqslant 2\mathrm{R}_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{2}{\delta}}, \tag{4.22}$$

with probability greater than $1 - \delta$, with the Rademacher complexity $\mathrm{R}_n(\mathcal{H})$ of the class of functions $\mathcal{H} = \{(x, y) \mapsto \ell(y, f(x)),\ f \in \mathcal{F}\}$. This leads to, with probability greater than $1 - \delta$,

$$\mathcal{R}(\hat{f}) \leqslant \widehat{\mathcal{R}}(\hat{f}) + 2\mathrm{R}_n(\mathcal{H}) + \frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{2}{\delta}}, \tag{4.23}$$

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 also use the empirical Rademacher complexity, which can be more easily computed (with fewer assumptions), and with a similar bound in equation (4.14).

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. For these two methods, we assume that we can enumerate all m models and minimize the appropriate criterion. This is not tractable when m is large; see chapter 8 for efficient methods in the context of variable selection.

In both cases, we consider a set of positive weights $\pi_1, \dots, \pi_m$ that sum to 1. 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 (even potentially infinite) 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,

$$\hat{\imath} = \underset{i \in \{1, \dots, m\}}{\arg\min} \Big\{ \widehat{\mathcal{R}}(\hat{f}_i) + 2\mathrm{R}_n(\mathcal{H}_i) + \frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{1}{\pi_i}} \Big\}. \tag{4.24}$$

We can then use equation (4.23) for each of the m models, with $\pi_i \delta$ instead of δ, and use the union bound (so equation (4.22) is satisfied for all models) to get, with probability greater than $1 - \delta$,

$$\begin{aligned}
\mathcal{R}(\hat{f}_{\hat{\imath}}) &\leqslant \widehat{\mathcal{R}}(\hat{f}_{\hat{\imath}}) + 2\mathrm{R}_n(\mathcal{H}_{\hat{\imath}}) + \frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{2}{\delta \pi_{\hat{\imath}}}} \text{ using equation (4.22)}, \\
&\leqslant \min_{i \in \{1, \dots, m\}} \Big\{ \widehat{\mathcal{R}}(\hat{f}_i) + 2\mathrm{R}_n(\mathcal{H}_i) + \frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{1}{\pi_i}} \Big\} + \frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{2}{\delta}} \\
&\qquad\qquad \text{by definition of } \hat{\imath} \text{ in equation (4.24)}, \\
&\leqslant \min_{i \in \{1, \dots, m\}} \Big\{ \inf_{f_i \in \mathcal{F}_i} \mathcal{R}(f_i) + 4\mathrm{R}_n(\mathcal{H}_i) + 2\frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{1}{\pi_i}} \Big\} + 2\frac{\ell_\infty}{\sqrt{2n}} \sqrt{\log \frac{2}{\delta}}, \quad (4.25)
\end{aligned}$$

by reusing equation (4.22) for all $i \in \{1, \dots, m\}$. For example, when $\pi_1 = \cdots = \pi_m = 1/m$, we thus find that the model selection procedure pays an extra price of $\sqrt{\log(m)/n}$ on top of the individual generalization bounds (with slightly worse constants).

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 ρn is an integer). We then have an empirical risk based on $(1-\rho)n$ observations, we which now denote as $\widehat{\mathcal{R}}^{(\text{training})}_{(1-\rho)n}$, and a validation empirical risk denoted as $\widehat{\mathcal{R}}^{(\text{validation})}_{\rho n}$. 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:

$$\widehat{\mathcal{R}}^{(\text{validation})}_{\rho n}(\hat{f}_i) + \frac{\ell_\infty}{\sqrt{2\rho n}}\sqrt{\log\frac{1}{\pi_i}}$$

(for uniform weights $\pi_1 = \cdots = \pi_m = 1/m$, this is simply the minimizer of the validation risk). We can then use Hoeffding's inequality (with respect to the randomness of the validation set, for which each $\hat{f}_i$ is deterministic) and the union bound to get the generalization bound (as in section 4.6.1), with probability greater than $1-\delta$,

$$\mathcal{R}(\hat{f}_{\hat{\imath}}) \quad \leqslant \quad \min_{i \in \{1,\dots,m\}} \mathcal{R}(\hat{f}_i) + \frac{\ell_\infty}{\sqrt{2\rho n}}\sqrt{\log\frac{1}{\pi_i}} + \frac{\ell_\infty}{\sqrt{2\rho n}}\sqrt{\log\frac{2}{\delta}},$$

which shows an extra price proportional to $\sqrt{\log(m)/(\rho n)}$ (for uniform weights), 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 section 4.6.1 by using the same generalization bounds based on Rademacher averages; that is, with probability greater than $1-2\delta$,

$$\begin{aligned}\mathcal{R}(\hat{f}_{\hat{\imath}}) \quad \leqslant \quad & \min_{i \in \{1,\dots,m\}} \Big\{ \inf_{f_i \in \mathcal{F}_i} \mathcal{R}(f_i) + 2\mathrm{R}_{(1-\rho)n}(\mathcal{H}_i) + \frac{\ell_\infty}{\sqrt{2n}}\Big(\frac{1}{\sqrt{\rho}} + \frac{1}{\sqrt{1-\rho}}\Big)\sqrt{\log\frac{1}{\pi_i}}\Big\} \\ & + \frac{\ell_\infty}{\sqrt{2n}}\Big(\frac{1}{\sqrt{\rho}} + \frac{1}{\sqrt{1-\rho}}\Big)\sqrt{\log\frac{2}{\delta}}.\end{aligned}$$

Suppose that ρ is bounded away from 0 and 1 (which the bound pushes to impose). In that case, we obtain a bound that is similar to equation (4.25), 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 Relation with Asymptotic Statistics (♦)

In this last section, we relate the nonasymptotic analysis presented in this chapter to results from asymptotic statistics (see the comprehensive book by van der Vaart (2000), which presents this large body of literature).

To make this concrete, we consider 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 theorems 5.21 or 5.41 on "M-estimation" (which cover empirical risk minimization) from van der Vaart (2000) apply. 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 semidefinite, as θ_* is a minimizer; we assume invertibility on top of it).

We let $\hat{\theta}_n$ denote a minimizer of $\widehat{\mathcal{R}}$. Since $\widehat{\mathcal{R}}'(\theta_*) = \frac{1}{n}\sum_{i=1}^n \frac{\partial \ell(y_i, f_\theta(x_i))}{\partial\theta}\big|_{\theta=\theta_*}$, 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 θ_* (all these statements can be made rigorous; see van der Vaart, 2000).

Then, a Taylor expansion of $\widehat{\mathcal{R}}'$ around θ_* 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 of 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 $\frac{1}{n}G(\theta_*) = \frac{1}{n}\mathbb{E}\big[\big(\frac{\partial \ell(y, f_\theta(x))}{\partial\theta}\big)\big(\frac{\partial \ell(y, f_\theta(x))}{\partial\theta}\big)^\top\big|_{\theta=\theta_*}\big]$, where $G(\theta_*)$ is referred to as the Fisher information matrix. Therefore, we intuitively find that $\hat{\theta}_n$ is asymptotically Gaussian with mean θ_* and covariance matrix $\frac{1}{n}H(\theta_*)^{-1}G(\theta_*)H(\theta_*)^{-1}$.

This asymptotic result has the nice consequence that

$$\begin{aligned}
\mathbb{E}\big[\|\hat{\theta}_n - \theta_*\|_2^2\big] &\sim \frac{1}{n}\operatorname{tr}\big[H(\theta_*)^{-1}G(\theta_*)H(\theta_*)^{-1}\big] \\
\mathbb{E}\big[\mathcal{R}(\hat{\theta}_n) - \mathcal{R}(\theta_*)\big] &\sim \frac{1}{2n}\operatorname{tr}\big[H(\theta_*)^{-1}G(\theta_*)\big].
\end{aligned}$$

For example, for well-specified linear regression (as 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 earlier, 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 also gives 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 potential looseness of nonasymptotic bounds that often rely on crude approximations (in particular the ones leading to excess risk in $1/\sqrt{n}$), and, while they are valid even for small n and still 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, and 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 here can be done in several ways. See, for example, Ostrovskii and Bach (2021) and references therein for finite-dimensional models, and chapter 7 for nonasymptotic results similar to $\sigma^2 d/n$ when the dimension of the feature space gets infinite. See also examples in Christmann and Steinwart (2008). Another line of work considers asymptotic analyses where several quantities (e.g., dimension d of the features and number n of observations) tend to infinity, with a ratio tending to a constant (see, e.g., Potters and Bouchaud, 2020).

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 detail in section 14.4, often leading to tighter bounds.

While this chapter was dedicated to the statistical analysis of empirical risk minimizers, the next chapter is dedicated to optimization algorithms aimed at approximating such minimizers, notably stochastic gradient descent (SGD), which also naturally exhibits good generalization performance.

Chapter 5

Optimization for Machine Learning

Chapter Summary

- Gradient descent (GD): 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 SGD: 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 GD and analyze their performance, mainly on convex objective functions. We will consider generic algorithms that have applications beyond machine learning as well as algorithms dedicated to machine learning (such as SGD methods). See Nesterov (2018); Bubeck (2015) for further details.

5.1 Optimization in Machine Learning

In supervised machine learning, we are given n independent and identically distributed (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 (since it still allows using arbitrarily complex prediction functions).

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 leads to the minimization of the following objective 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 as discussed 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 rather the error on unseen data.

Then, what accuracy is necessary in machine learning? If the algorithm returns $\hat{\theta}$ and we define $\theta_* \in \arg\min_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta)$, from section 2.3.2, the excess risk is the sum of the approximation error (which characterizes the error due to the use of a specific set of models $\{f_\theta\}$) and the estimation error, which can be decomposed as follows:

$$\begin{aligned}
\mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) &= \big\{\mathcal{R}(f_{\hat{\theta}}) - \widehat{\mathcal{R}}(f_{\hat{\theta}})\big\} + \big\{\widehat{\mathcal{R}}(f_{\hat{\theta}}) - \widehat{\mathcal{R}}(f_{\theta_*})\big\} + \big\{\widehat{\mathcal{R}}(f_{\theta_*}) - \mathcal{R}(f_{\theta_*})\big\}, \\
&\leqslant \underbrace{\big\{\mathcal{R}(f_{\hat{\theta}}) - \widehat{\mathcal{R}}(f_{\hat{\theta}})\big\}}_{\text{deviation - I}} + \underbrace{\big\{\widehat{\mathcal{R}}(f_{\hat{\theta}}) - \inf_{\theta \in \mathbb{R}^d} \widehat{\mathcal{R}}(f_\theta)\big\}}_{\text{optimization error}} + \underbrace{\big\{\widehat{\mathcal{R}}(f_{\theta_*}) - \mathcal{R}(f_{\theta_*})\big\}}_{\text{deviation - II}}.
\end{aligned}$$

On top of the two parts based on the deviation between the expected and empirical risks, we added the second term, the *optimization error*, which will equal zero if $\hat{\theta}$ is the minimizer of $\widehat{\mathcal{R}}$. It is thus sufficient to reach an optimization accuracy of the order of the deviation error (usually of the order $O(1/\sqrt{n})$ or $O(1/n)$; see chapters 3 and 4). Note that for machine learning, the optimization error defined here corresponds to characterizing approximate solutions through function values. While this will be one central focal point in this chapter, we will also 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 nonsmooth objective functions. We will then look at stochastic gradient descent (SGD) 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 θ_* may typically 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 η_* for the minimizer of empirical (potentially regularized) risk (i.e., when we look at optimization problems), and θ_* for the minimizer of the expected risk (i.e., 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 GD methods for convex optimization problems, which, in learning terms, correspond to predictors that are linear in their parameters (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, which do not use the finite sum structure of the objective function in equation (5.1) before moving on to stochastic gradient methods, which do take into account this structure for enhanced computational efficiency.

As for bounds on the estimation error in section 4.5.4, most of the convergence bounds in this section do not have any explicit dependence on the underlying dimension d. They thus apply in infinite-dimensional Hilbert spaces and can be made practically implementable in finite dimension using the "kernel trick" (see section 7.4).

5.2 Gradient Descent

Suppose that we want to solve, for the 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 *kth-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 the oracles. For example, for least-squares regression 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 function F is the regularized empirical risk, but this situation will be our motivating example throughout. First, we will study gradient descent, a first-order algorithm.

Algorithm 5.1 (Gradient descent) *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 γ_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 (such as backtracking or Armijo line search). 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 of this book). Note that the differences between convergence rates with and without line searches are generally not significant (see exercise 5.2, about quadratic functions), with sometimes some differences when exact line search is used (see, e.g., Bolte and Pauwels, 2022). 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 (OLS; see chapter 3 for the statistical analysis of this estimator). 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 η_* 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 avoid a factor of 2 for 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 $H = \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d\times d}$ for the Hessian matrix (equal for all θ, denoted as $\widehat{\Sigma}$ in chapter 3), minimizers η_* are characterized by $F'(\eta_*) = 0$; that is,

$$H\eta_* = \frac{1}{n}\Phi^\top y.$$

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. However, all minimizers η_* have the same function value $F(\eta_*)$, and we have, from a simple exact Taylor expansion (and using $F'(\eta_*) = 0$),

$$F(\theta) - F(\eta_*) = F'(\eta_*)^\top(\theta - \eta_*) + \frac{1}{2}(\theta - \eta_*)^\top H(\theta - \eta_*) = \frac{1}{2}(\theta - \eta_*)^\top H(\theta - \eta_*).$$

[1] See, e.g., https://en.wikipedia.org/wiki/Line_search.

Two quantities will be important in the following developments: the largest eigenvalue L and the smallest eigenvalue μ of the Hessian matrix H. As a consequence of the convexity of the objective, we have $0 \leqslant \mu \leqslant L$. We denote as $\kappa = \frac{L}{\mu} \geqslant 1$ the *condition number*.

Note that for least-squares regression, μ is the lowest eigenvalue of the noncentered empirical covariance matrix, and it is zero as soon as $d > n$, and, in most practical cases, it is *very* small. When adding a regularizer $\frac{\lambda}{2}\|\theta\|_2^2$ (as in ridge regression), then $\mu \geqslant \lambda$ (but then λ typically decreases with n, often between $1/\sqrt{n}$ and $1/n$; see section 7.6.4 in chapter 7 for more details).

Closed-form expression. GD iterates with fixed step size $\gamma_t = \gamma$ can be computed as follows:

$$\theta_t = \theta_{t-1} - \gamma F'(\theta_{t-1}) = \theta_{t-1} - \gamma\Big[\frac{1}{n}\Phi^\top(\Phi\theta_{t-1} - y)\Big] = \theta_{t-1} - \gamma H(\theta_{t-1} - \eta_*),$$

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:

$$\begin{aligned}
\|\theta_t - \eta_*\|_2^2 &= (\theta_0 - \eta_*)^\top (I - \gamma H)^{2t}(\theta_0 - \eta_*) \\
F(\theta_t) - F(\eta_*) &= \frac{1}{2}(\theta_0 - \eta_*)^\top (I - \gamma H)^t H (I - \gamma H)^t(\theta_0 - \eta_*) \\
&= \frac{1}{2}(\theta_0 - \eta_*)^\top (I - \gamma H)^{2t} H (\theta_0 - \eta_*), \text{ since matrices commute.}
\end{aligned}$$

The two optimization performance measures differ by the presence of the Hessian matrix H in the measure based on function values.

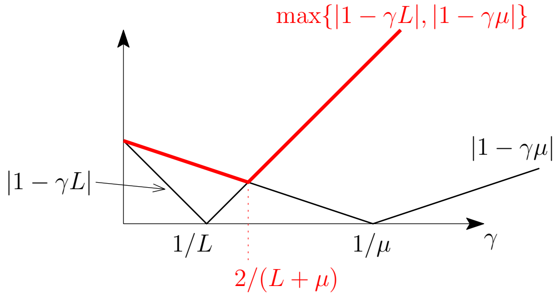
Convergence in distance to the minimizer. If we hope to have $\|\theta_t - \eta_*\|_2^2$ going to zero, we need to have a single minimizer η_*, and thus H has to be invertible; that is, $\mu > 0$. Given the form of $\|\theta_t - \eta_*\|_2^2$, we simply need to bound the eigenvalues of $(I - \gamma H)^{2t}$ (since for a positive semidefinite 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 λ being an eigenvalue of H (all of them are in the interval $[\mu, L]$). Thus, all the eigenvalues of $(I - \gamma H)^{2t}$ have a 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 γ:

- 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 the following geometric proof:



- Choice independent of μ: 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 term "linear" in this context 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 κ. 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 ε, we need at most $t = \frac{\kappa}{2} \log \frac{1}{\varepsilon}$ iterations.

For $\kappa = +\infty$ (i.e., $\mu = 0$), 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 before, and using the upper bound on eigenvalues of $(I - \gamma H)^{2t}$ (which are all less than $(1 - 1/\kappa)^{2t}$), we get

$$F(\theta_t) - F(\eta_*) \leqslant \Big(1 - \frac{1}{\kappa}\Big)^{2t} [F(\theta_0) - F(\eta_*)] \leqslant \exp(-2t/\kappa)[F(\theta_0) - F(\eta_*)]. \quad (5.4)$$

When $\kappa < \infty$ (i.e., $\mu > 0$), we also obtain linear convergence for this criterion, but when $\kappa = \infty$, this is noninformative.

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 λ of H that are close to zero, $(1-\gamma\lambda)^{2t}$ does not have a strong contracting effect, but the eigenvalues count less as they are multiplied by λ in the bound.

We can make this trade-off precise, for $\gamma \leqslant 1/L$, as

$$\begin{aligned} \left|\lambda(1-\gamma\lambda)^{2t}\right| &\leqslant \lambda \exp(-\gamma\lambda)^{2t} = \lambda\exp(-2t\gamma\lambda) \\ &= \frac{1}{2t\gamma} 2t\gamma\lambda\exp(-2t\gamma\lambda) \leqslant \frac{1}{2t\gamma}\sup_{\alpha\geqslant 0}\alpha\exp(-\alpha) = \frac{1}{2et\gamma} \leqslant \frac{1}{4t\gamma}, \end{aligned}$$

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 equation (5.4) for invertible Hessians, or $1/t$ in general in equation (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, when the step-size γ is strictly less than $2/L$, the lowest eigenvalue μ 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 λ 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 nonzero eigenvalue of H kicks in. See the simulations in figure 5.1, exercise 5.1, and section 12.1.1 for more details.

  Exercise 5.1 *Let μ_+ be the smallest nonzero eigenvalue of H. Show that GD is linearly convergent with a convergence rate proportional to $(1-\mu_+/L)^{2t}$ after t iterations.*

- From errors to numbers of iterations: As already mentioned, the bound in equation (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 ε.

- Can an algorithm having the same access to oracles of F do better?

  If we have access to matrix-vector products with matrix Φ, 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

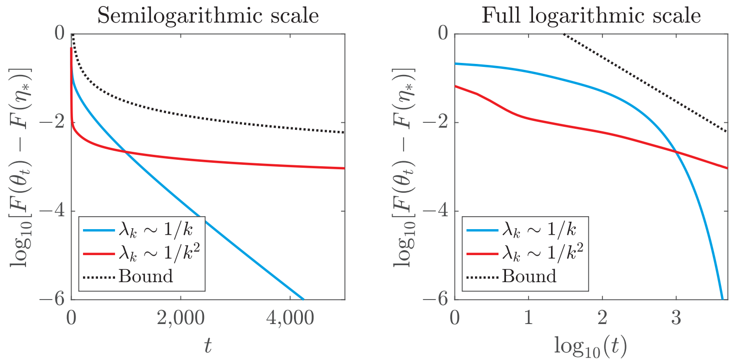


Figure 5.1. GD on two least-squares problems with step size $\gamma = 1/L$, and two different sets of eigenvalues $(\lambda_k)_{k\in\{1,\dots,d\}}$ of the Hessian, together with the bound from equation (5.5). Left: semilogarithmic scale. Right: joint logarithmic scale.

weaker), Nesterov acceleration (see section 5.2.5) will lead to the same convergence rates as conjugate gradient, which are then optimal (in a sense that will be defined later in this chapter and described in more detail in chapter 15).

- Can we extend beyond least-squares regression? The convergence results given here will generalize to convex functions (see section 5.2.2), but with less direct proofs. Nonconvex objectives are discussed in section 5.2.6.

Experiments. Here, we consider two quadratic optimization problems in dimension $d = 1,000$, with two different decays of eigenvalues $(\lambda_k)_{k\in\{1,\dots,d\}}$ for the Hessian matrix H: one as $1/k$ (in blue) and one in $1/k^2$ (in red), and for which we plot in figure 5.1 the optimization error for function values, both in semilogarithmic plots (left) and full-logarithmic plots (right). For slow decays (blue), we see the linear convergence kicking in (line in the left “semilog” 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 equation (5.5) is very pessimistic and does not lead to the observed power of t (which, as can be checked as an exercise, should be $1/\sqrt{t}$ for t small enough compared to d).

Exercise 5.2 (Exact line search (♦)) *For the quadratic objective in equation (5.3), show that the optimal step size γ_t in equation (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, we have $F(\theta_t) - F(\eta_*) \leqslant \big(\frac{\kappa-1}{\kappa+1}\big)^2\big[F(\theta_{t-1}) - F(\eta_*)\big]$, and compare the rate with constant step size GD. Hint: prove and use the Kantorovich inequality* $\sup_{\|z\|_2=1} z^\top H z z^\top H^{-1} z = \frac{(L+\mu)^2}{4\mu L}$.

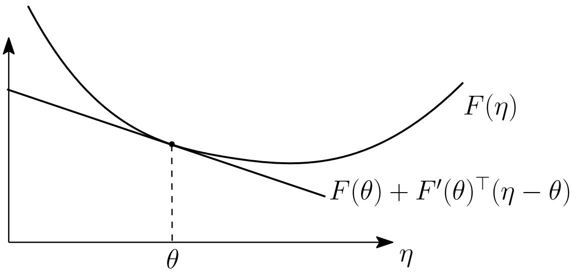
5.2.2 Convex Functions and Their Properties

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. In this section, we present the main properties of convex functions that we will need in this book.

Definition 5.1 (Convex function) *A differentiable function $F : \mathbb{R}^d \to \mathbb{R}$ is said to be 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 θ, as illustrated here:



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.

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, as in proposition 5.1.[2]

Proposition 5.1 (Jensen's inequality) *If $F : \mathbb{R}^d \to \mathbb{R}$ is convex and μ is a probability measure on $\mathbb{R}^d$, then*

$$F\Big(\int_{\mathbb{R}^d} \theta d\mu(\theta)\Big) \leqslant \int_{\mathbb{R}^d} F(\theta) d\mu(\theta). \tag{5.8}$$

Stated in words: "The image of the average is smaller than the average of the images."

⚠ When using Jensen's inequality, be extra careful about the direction of the inequality.

[2]See also section 1.2 and several applications in https://francisbach.com/jensen-inequality/.

Exercise 5.3 *Assume that 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 equation (5.8) if and only if the random variable θ is almost surely constant.*

The class of convex functions satisfies the following stability properties (proofs left as an exercise); for more properties, see Boyd and Vandenberghe (2004):

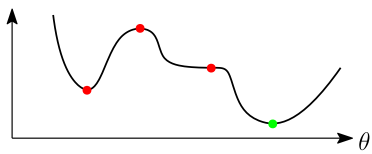
- If $(F_j)_{j\in\{1,\dots,m\}}$ are convex and $(\alpha_j)_{j\in\{1,\dots,m\}}$ are nonnegative, then $\sum_{j=1}^m \alpha_j F_j$ and $\max_{j\in\{1,\dots,m\}} F_j$ are convex.
- If $F : \mathbb{R}^d \to \mathbb{R}$ is convex and $A : \mathbb{R}^{d'} \to \mathbb{R}^d$ is affine then $F \circ A : \mathbb{R}^{d'} \to \mathbb{R}$ 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 equation (5.1) are convex if the loss ℓ is convex in the second variable, $f_\theta(x)$ is linear in θ, and Ω is convex. These thus correspond to linear models (in their parameters), and not to nonlinear models such as neural networks, which are studied in chapter 9.

Global optimality from local information. It is also worth emphasizing the property expressed in proposition 5.2 (immediate from equation (5.7)).

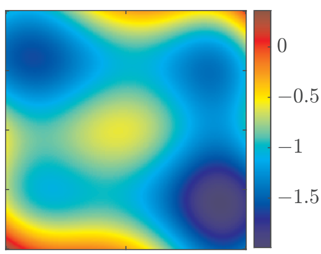
Proposition 5.2 *Assume that $F : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable. Then $\eta_* \in \mathbb{R}^d$ is a global minimizer of F if and only if $F'(\eta_*) = 0$.*

Thus, for convex functions, we only need to look for stationary points. This is *not* the case for potentially nonconvex functions. For example, in one dimension below, all red points are stationary points that are not the global minimum (shown 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 here:*



5.2.3 Analysis of Gradient Descent for Strongly Convex and Smooth Functions

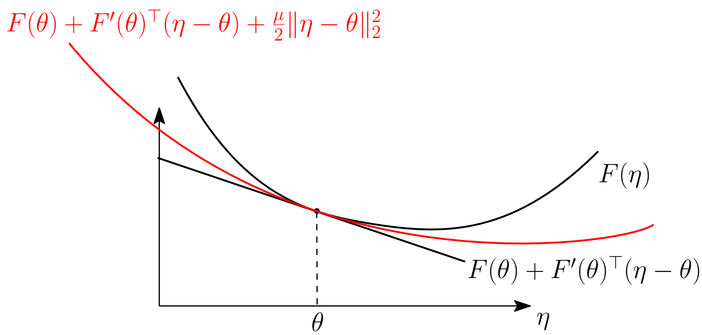
The analysis of optimization algorithms requires assumptions on the objective functions, like the ones introduced in this section. From these assumptions, additional properties are derived (typically inequalities), and then most convergence proofs look for a nonnegative "Lyapunov function" (sometimes 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 to a minimizer of V (which is usually chosen to be a minimizer of F). The task 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 to be μ-strongly-convex, with $\mu > 0$, if and only if*

$$F(\eta) \geqslant F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|_2^2, \quad \forall \eta, \theta \in \mathbb{R}^d. \tag{5.9}$$

Function F is strongly convex if and only if 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, as shown here:



For twice-differentiable functions, this is equivalent to $F''(\theta) \succcurlyeq \mu I$ for all $\theta \in \mathbb{R}^d$; that is, all eigenvalues of $F''(\theta)$ are greater than or equal to μ (see Nesterov, 2018), but nonsmooth functions can be strongly convex, since, as a consequence of exercise 5.5, we can add $\frac{\mu}{2}\|\cdot\|_2^2$ to any potentially nonsmooth convex function to make it μ-strongly-convex.

Exercise 5.5 *Show that function $F : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly-convex if and only if function $\theta \mapsto F(\theta) - \frac{\mu}{2}\|\theta\|_2^2$ is convex.*

Exercise 5.6 *Show that if function $F : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly-convex, then it has a unique minimizer.*

Exercise 5.7 (♦) *Show that the differentiable convex function $F : \mathbb{R}^d \to \mathbb{R}$ is μ-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 μ-strongly convex (see exercise 5.5). In practice, in machine learning problems with linear models, such that the empirical risk is convex, strong convexity most often comes from the regularizer (and thus μ 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's inequality. Strong convexity implies that F admits a unique minimizer η_*, 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's inequality) *If F is differentiable and μ-strongly convex with unique minimizer η_*, then we have*

$$\|F'(\theta)\|_2^2 \geqslant 2\mu(F(\theta) - F(\eta_*)), \quad \forall \theta \in \mathbb{R}^d.$$

Proof The right side of equation (5.9) is strongly convex in η and minimized with $\tilde{\eta} = \theta - \frac{1}{\mu}F'(\theta)$. Plugging this value into the bound and minimizing the left side by taking $\eta = \eta_*$, 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's inequality, it is not necessary, and it may lead to exponential convergence without strong convexity (see, e.g., section 12.1.1).

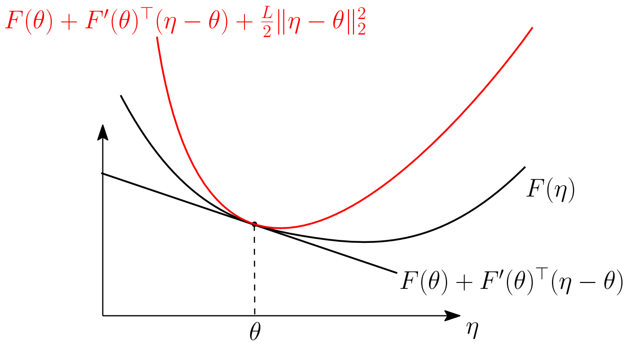
To obtain exponential convergence rates, strong convexity is typically associated with smoothness, which we now define.

Definition 5.3 (Smoothness) *A differentiable function F is said to be L-smooth if and only if*

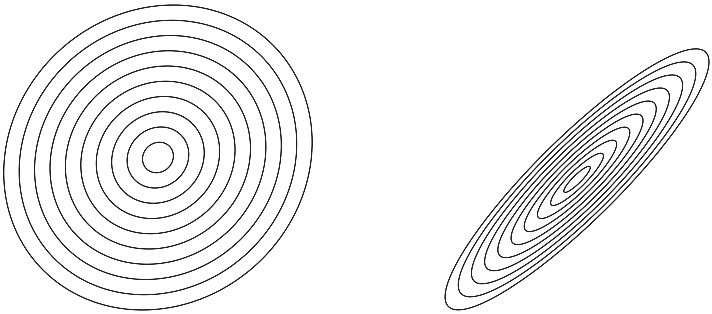
$$|F(\eta) - F(\theta) - F'(\theta)^\top(\eta - \theta)| \leqslant \frac{L}{2}\|\theta - \eta\|_2^2, \quad \forall \theta, \eta \in \mathbb{R}^d. \tag{5.10}$$

This is equivalent to F having a L-Lipschitz-continuous gradient with respect to the ℓ_2-norm; that is, $\|F'(\theta) - F'(\eta)\|_2 \leqslant L\|\theta - \eta\|_2$, $\forall \theta, \eta \in \mathbb{R}^d$ (proof left as an exercise). For twice-differentiable functions, this is equivalent to $-LI \preccurlyeq F''(\theta) \preccurlyeq LI$ (see Nesterov, 2018). If the function is also μ-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 μ), as shown here:

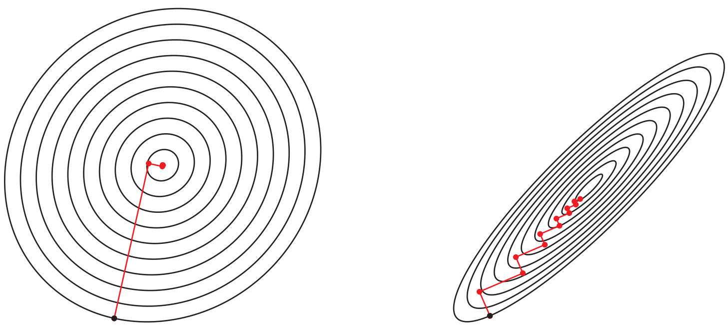


When a function is both smooth and strongly convex, we denote by $\kappa = L/\mu \geqslant 1$ its condition number (for quadratic functions, the Hessian is the same at all points, and we recover the definition from section 5.2.1). See the following examples of level sets of functions with varying condition numbers; the condition number affects the shapes of the level sets:



(Small $\kappa = L/\mu$) (Large $\kappa = L/\mu$)

The performance of GD will depend on this condition number. In the following plot, the steps of the "steepest descent" method (i.e., GD with exact line search) are plotted; with a small condition number (left), we get fast convergence, while for a large condition number (right), we get oscillations.



(Small $\kappa = L/\mu$) (Large $\kappa = L/\mu$)

Exercise 5.8 (♦) *Consider angle α 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 μ-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 proposition 5.4.)*

For machine learning problems, such as linear predictions and smooth losses (square or logistic), we have smooth problems. If we use a squared ℓ_2-regularizer $\frac{\mu}{2}\|\cdot\|_2^2$, we get a μ-strongly convex problem. Note that when using regularization, as explained in chapters 3 and 4, the value of μ decays with n, typically between $1/n$ and $1/\sqrt{n}$ (see also section 7.6.4), leading to condition numbers between $\sqrt{n}$ and n.

In this context, GD on the empirical risk is often called a "batch" technique because all the data points are accessed at every iteration. In proposition 5.3, we show that GD 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 μ-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 equation (5.10) applied to θ_{t-1} and the next iterate $\theta_t = \theta_{t-1} - F'(\theta_{t-1})/L$, 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 that

$$F(\theta_t) - F(\eta_*) \leqslant (1 - \mu/L)(F(\theta_{t-1}) - F(\eta_*)) \leqslant \exp(-\mu/L)(F(\theta_{t-1}) - F(\eta_*)).$$

We conclude by recursion on t 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 under 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, as well as the end of section 5.2.5).

- If we only assume that the function is smooth and convex (not strongly convex), then GD with the 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.
- Choosing the step size only requires an upper bound L on the smoothness constant (if it is overestimated, 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, and chapter 12 where this analogy is studied further for nonconvex problems).
- For this class of functions (convex and smooth), first-order methods exist that achieve a faster rate, showing that GD is not optimal. However, these improved algorithms also have drawbacks (lack of adaptivity, instability to noise, etc.). See section 5.2.5.

Exercise 5.9 *Compute all constants for ℓ_2-regularized logistic regression and for ridge regression.*

Adaptivity. Note that GD 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 η_*, which can be significantly larger than μ (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 find, 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 that 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 nonconvex 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 (Fenchel-Young inequality) *Let F be an L-smooth convex function on $\mathbb{R}^d$ and F^* be 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 $z = F'(\theta)$. (♦) Show in addition that we have the lower bound $F(\theta) + F^*(z) - z^\top\theta \geqslant \frac{1}{2L}\|z - F'(\theta)\|_2^2$.*

5.2.4 Analysis of Gradient Descent for Convex and Smooth Functions (♦)

To obtain the $1/t$ convergence rate without strong convexity (as we found 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 that 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*

$$\frac{1}{L}\|F'(\theta) - F'(\eta)\|_2^2 \leqslant \big[F'(\theta) - F'(\eta)\big]^\top (\theta - \eta). \tag{5.11}$$

Moreover, we have

$$F(\theta) \geqslant F(\eta) + F'(\eta)^\top(\theta - \eta) + \frac{1}{2L}\|F'(\theta) - F'(\eta)\|_2^2. \tag{5.12}$$

Proof We will prove equation (5.12), which implies equation (5.11), by applying it twice with η and θ swapped and summing them. Using convexity (to obtain the left inequality) and smoothness (to obtain the right inequality), we have, for any $\xi \in \mathbb{R}^d$,

$$F(\eta) + F'(\eta)^\top(\xi - \eta) \ \leqslant\ F(\xi) \ \leqslant\ F(\theta) + F'(\theta)^\top(\xi - \theta) + \frac{L}{2}\|\theta - \xi\|_2^2. \tag{5.13}$$

We can find the ξ minimizing the difference between the rightmost and leftmost terms in equation (5.13) by setting the gradient of the difference with respect to ξ to zero, leading to $F'(\eta) - F'(\theta) - L(\xi - \theta) = 0$. Putting this value of ξ back in equation (5.13) and rearranging terms lead to equation (5.12). ■

We can now state the following convergence result for GD with potentially no strong convexity. Up to constants, we obtain the same rate for quadratic functions in equation (5.5).

Proposition 5.5 (Convergence of GD for smooth convex functions) *Assume that F is L-smooth and convex, with a global minimizer η_*. 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 do the following:

- We use $F(\theta_t) - F(\theta_{t-1}) \leqslant -\frac{1}{2L}\|F'(\theta_{t-1})\|_2^2$ as in the proof of proposition 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 θ_{t-1}), as in equation (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 $\gamma = 1/L$,

$$\begin{aligned} V_t(\theta_t) - V_{t-1}(\theta_{t-1}) \ & \leqslant \ t\Big[-\frac{1}{2L}\|F'(\theta_{t-1})\|_2^2\Big] + F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*) \\ & \qquad -L\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + \frac{L\gamma^2}{2}\|F'(\theta_{t-1})\|_2^2 \\ & = \ -\frac{t-1}{2L}\|F'(\theta_{t-1})\|_2^2 \ \leqslant 0, \end{aligned}$$

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 $F(\theta_t) - F(\eta_*) \leqslant \frac{L}{2t}\|\theta_0 - \eta_*\|_2^2$. ■

This proof is mysterious on purpose: 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/`, and see exercises 5.12 and 5.13 for simpler more direct proofs.

Exercise 5.12 (Alternative convergence proof - I) *Consider an L-smooth convex function with a global minimizer η_*, and GD with step size $\gamma_t = 1/L$:*

- *Using proposition 5.4, show that $\|\theta_t - \eta_*\|_2^2 \leqslant \|\theta_{t-1} - \eta_*\|_2^2 - \frac{1}{L}F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*)$.*
- *Show that $F(\theta_t) \leqslant F(\theta_{t-1})$.*
- *Using a telescoping sum, show that $F(\theta_t) - F(\eta_*) \leqslant \frac{L}{t+1}\|\theta_0 - \eta_*\|_2^2$.*

Exercise 5.13 (Alternative convergence proof - II (♦)) *Consider an L-smooth convex function with a global minimizer η_*, and GD with step size $\gamma_t = 1/L$:*

- *Show that* $\|\theta_t - \eta_*\|_2^2 \leqslant \|\theta_{t-1} - \eta_*\|_2^2$ *for all* $t \geqslant 1$.
- *Show that* $F(\theta_t) \leqslant F(\theta_{t-1}) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2$ *for all* $t \geqslant 1$.
- *Denoting* $\Delta_t = F(\theta_t) - F(\eta_*)$, *show that* $\Delta_t \leqslant \Delta_{t-1} - \frac{1}{2L\|\theta_0 - \eta_*\|_2^2}\Delta_{t-1}^2$ *for all* $t \geqslant 1$. *Conclude that* $\Delta_t \leqslant \frac{2L}{t+4}\|\theta_0 - \eta_*\|_2^2$.

Early-stopping for machine learning (♦♦). An inspection of the proof of proposition 5.5 shows that throughout, a minimizer η_* can be replaced by *any* $\eta \in \mathbb{R}^d$, leading to (for the step size $\gamma = 1/L$):

$$t[F(\theta_t) - F(\eta)] + \frac{L}{2}\|\theta_t - \eta\|_2^2 \leqslant \frac{L}{2}\|\theta_0 - \eta\|_2^2. \tag{5.14}$$

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 ℓ_2-norms smaller than R, we have $L \leqslant G_2 R^2$. Moreover, if the loss is nonnegative, 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,

$$\mathbb{E}\Big[\sup_{\|\theta\|_2 \leqslant D} \big\{\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta)\big\}\Big] \leqslant \frac{2G_1 R D}{\sqrt{n}}. \tag{5.15}$$

Then from equation (5.14), assuming that we initialize with $\theta_0 = 0$, we get for $\eta = 0$, $\frac{G_2R^2}{2}\|\theta_t\|_2^2 \leqslant tG_0$; that is, $\|\theta_t\|_2 \leqslant \big(\frac{2G_0}{G_2R^2}t\big)^{1/2}$. This leads to for any $\eta \in \mathbb{R}^d$, using equation (5.15):

$$\begin{aligned}\mathbb{E}\big[\mathcal{R}(\theta_t)\big] &\leqslant \mathcal{R}(\eta) + \mathbb{E}\big[\widehat{\mathcal{R}}(\theta_t) - \widehat{\mathcal{R}}(\eta)\big] + \frac{2G_1R}{\sqrt{n}}\Big(\frac{2G_0}{G_2R^2}t\Big)^{1/2} \\ &\leqslant \mathcal{R}(\eta) + \frac{G_2R^2}{2t}\|\eta\|_2^2 + \frac{2\sqrt{2}\,G_1G_0^{1/2}}{G_2^{1/2}}\frac{\sqrt{t}}{\sqrt{n}},\end{aligned}$$

showing that if $t = o(n)$ (we do not take too many steps), the testing error is controlled. In particular, if θ_* is a minimizer of the expected risk $\mathcal{R}$, then with $t = \sqrt{n}$, we obtain $\mathbb{E}\big[\mathcal{R}(\theta_t)\big] - \mathcal{R}(\theta_*) = O(n^{-1/4})$. This bound with early-stopping of GD is not as good as $O(n^{-1/2})$, which we will obtain through explicit regularization at the end of section 5.3. Note that early-stopped SGD will also lead to a bound in $O(n^{-1/2})$ (at a much cheaper computational cost). We will revisit early-stopping of batch algorithms when we describe boosting procedures in section 10.3.

5.2.5 Beyond Gradient Descent (♦)

While GD is the simplest algorithm with a simple analysis, there are multiple extensions that we will only briefly mention here (see more details by Nesterov, 2004, 2018).

Nesterov acceleration. For strongly convex functions, a simple modification of GD allows for obtaining better convergence rates. The algorithm is as follows and is based on updating the following two iterates:

$$\theta_t = \eta_{t-1} - \frac{1}{L} g'(\eta_{t-1}) \tag{5.16}$$

$$\eta_t = \theta_t + \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}(\theta_t - \theta_{t-1}). \tag{5.17}$$

The convergence rate is then $F(\theta_t) - F(\eta_*) \leqslant L\|\theta_0 - \eta_*\|_2^2(1-\sqrt{\mu/L})^t$, which is equal to $L\|\theta_0 - \eta_*\|_2^2(1-1/\sqrt{\kappa})^t$; that is, the characteristic time to convergence goes from κ to $\sqrt{\kappa}$. If κ 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:

$$\theta_t = \eta_{t-1} - \frac{1}{L} F'(\eta_{t-1}) \tag{5.18}$$

$$\eta_t = \theta_t + \frac{t-1}{t+2}(\theta_t - \theta_{t-1}). \tag{5.19}$$

This simple modification dates back to Nesterov (1983) and leads to the following convergence rate $F(\theta_t) - F(\eta_*) \leqslant \frac{2L\|\theta_0 - \eta_*\|_2^2}{(t+1)^2}$. See exercises 5.14 and 5.15, as well as 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 (2013) and chapter 15 for more details.

Exercise 5.14 (♦♦) *For the updates in equations (5.16) and (5.17), show that for the Lyapunov function $V(\theta, \eta) = f(\theta) - f(\eta_*) + \frac{\mu}{2}\big\|\theta - \eta_* + (1+\sqrt{L/\mu})(\eta - \theta)\big\|_2^2$, then we have $V(\theta_t, \eta_t) \leqslant (1-\sqrt{\mu/L})V(\theta_{t-1}, \eta_{t-1})$. Show that this implies a convergence rate proportional to $(1-\sqrt{\mu/L})^t$.*

Exercise 5.15 (♦♦) *For the updates in equations (5.18) and (5.19), show that for the Lyapunov function $V_t(\theta, \eta) = \big(\frac{t+1}{2}\big)^2\big[f(\theta) - f(\eta_*)\big] + \frac{L}{2}\big\|\eta - \eta_* + \frac{t}{2}(\eta - \theta)\big\|_2^2$, then we have $V_t(\theta_t, \eta_t) \leqslant V_{t-1}(\theta_{t-1}, \eta_{t-1})$. Show that this implies a convergence rate proportional to $1/t^2$.*

Newton method. Given θ_{t-1}, the Newton method minimizes the second-order Taylor expansion around θ_{t-1} (or, equivalently, finds a zero of F' by using a first-order Taylor expansion of F' around θ_{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 $F'(\theta_{t-1}) + F''(\theta_{t-1})^\top(\theta - \theta_{t-1})$, 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 for solving the linear system. It leads to local quadratic convergence: if $\|\theta_{t-1} - \theta_*\|_2$ small enough, for some constant C, one can show $(C\|\theta_t - \theta_*\|_2) \leqslant (C\|\theta_{t-1} - \theta_*\|_2)^2$. 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 term "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 5.16 (♦) *Assume that function F is μ-strongly convex, twice-differentiable, and such that the Hessian is Lipschitz-continuous: $\|f''(\theta) - f''(\eta)\|_{\rm op} \leqslant M\|\theta - \eta\|_2$. Using Taylor's formula with an 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 to be "composite"; that is, the objective function F is the sum of a smooth function G and a nonsmooth function H (such as a norm). It turns out that a simple modification of GD allows us to benefit from the fast convergence rates of smooth optimization (compared to the slower rates for nonsmooth optimization that we would obtain from the subgradient method described in section 5.3).

For this, we need to first see GD as a *proximal method*. Indeed, one may see the iteration $\theta_t = \theta_{t-1} - \frac{1}{L}G'(\theta_{t-1})$ as

$$\theta_t = \arg\min_{\theta\in\mathbb{R}^d} \ G(\theta_{t-1}) + (\theta - \theta_{t-1})^\top G'(\theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2,$$

where, for a L-smooth function G, the objective function given here is an upper bound of $G(\theta)$ that is tight at θ_{t-1} (see equation (5.10)).

While this reformulation does not bring much for GD, we can extend this to the composite problem and consider the following iteration, where H is left *as is*:

$$\theta_t = \arg\min_{\theta\in\mathbb{R}^d} \ G(\theta_{t-1}) + (\theta - \theta_{t-1})^\top G'(\theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2 + H(\theta). \tag{5.20}$$

It turns out that the convergence rates for $G + H$ are the same as smooth optimization, with potential acceleration (Nesterov, 2013; Beck and Teboulle, 2009); see a proof in exercise 5.17.

The crux is to be able to compute the proximal update in equation (5.20); that is, minimize with respect to θ 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 GD. When H is the ℓ_1-norm (i.e., $H = \lambda\|\cdot\|_1$), this can be shown to be a soft-thresholding step, as for each coordinate $\theta_i = (|\eta_i| - \lambda/L)_+ \frac{\eta_i}{|\eta_i|}$ (the proof is left as an exercise). See applications to model selection and sparsity-inducing norms in chapter 8.

Exercise 5.17 (Convergence of proximal gradient method) *Consider a convex L-smooth function G and a convex function H defined on $\mathbb{R}^d$. We consider the update in equation (5.20) and a minimizer η_* of $G + H$.*

- *Show that $G(\theta_t) \leqslant G(\theta_{t-1}) + G'(\theta_{t-1})^\top(\theta_t - \theta_{t-1}) + \frac{L}{2}\|\theta_t - \theta_{t-1}\|_2^2$.*
- *Show that $G(\theta_{t-1}) \leqslant G(\eta_*) + G'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*)$.*
- *Show that $H(\theta_t) \leqslant H(\eta_*) + (L\theta_{t-1} - L\theta_t - G'(\theta_{t-1}))^\top(\theta_t - \eta_*)$.*
- *Deduce that $G(\theta_t) + H(\theta_t) \leqslant G(\eta_*) + H(\eta_*) + \frac{L}{2}\|\theta_{t-1} - \eta_*\|_2^2 - \frac{L}{2}\|\theta_t - \eta_*\|_2^2$.*
- *Conclude that for $t \geqslant 1$, $G(\theta_t) + H(\theta_t) - \big[G(\eta_*) + H(\eta_*)\big] \leqslant \frac{L}{2t}\|\theta_0 - \eta_*\|_2^2$.*

Preconditioning (♦). The convergence rate of GD depends crucially on the condition number κ, which is not invariant under 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 GD on function G can be written as $\tilde\theta_t = \tilde\theta_{t-1} - \gamma G'(\tilde\theta) = \tilde\theta_{t-1} - \gamma A^\top F'(A\tilde\theta_{t-1})$, which can be rewritten as $\theta_t = \theta_{t-1} - \gamma AA^\top F'(\theta_{t-1})$ with the change of variable $\theta = A\tilde\theta$. This is thus equivalent to premultiplying 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 preconditioned 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 (for functions with varying Hessians), 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 nonsmooth situations (see section 5.4.2 in the context of SGD).

5.2.6 Nonconvex Objective Functions (♦)

For smooth, potentially nonconvex objective functions, the best that one can hope for is to converge to a stationary point θ such that $F'(\theta) = 0$. The proof that follows provides the weaker result that at least one iterate has a small gradient. Indeed, using the same Taylor expansion as in the convex case (which is still valid), we get, using the L-smoothness of F,

$$F(\theta_t) \quad \leqslant \quad F(\theta_{t-1}) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2,$$

leading to, summing these inequalities for all iterations between 1 and t,

$$\frac{1}{2Lt}\sum_{s=1}^{t}\|F'(\theta_{s-1})\|_2^2 \leqslant \frac{F(\theta_0)-F(\theta_t)}{t} \leqslant \frac{F(\theta_0)-\inf_{\eta\in\mathbb{R}^d}F(\eta)}{t}.$$

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. See an extension of this proof for SGD in exercise 5.30.

5.3 Gradient Methods on Nonsmooth Problems

We now relax our assumptions and only require Lipschitz continuity in addition to convexity. The rates will be slower, but extending to stochastic gradients will be easier.

Definition 5.4 (Lipschitz-continuous function) *Function $F:\mathbb{R}^d \to \mathbb{R}$ is said to be B-Lipschitz-continuous if and only if*

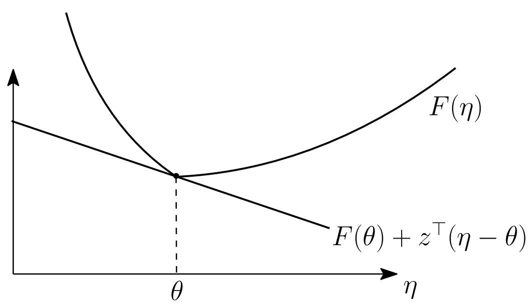
$$|F(\eta)-F(\theta)| \leqslant B\|\eta-\theta\|_2, \qquad \forall \theta,\eta\in\mathbb{R}^d.$$

This setting is usually referred to as *nonsmooth* optimization.

Exercise 5.18 *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 nonsmooth 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 the illustration that follows):

$$\partial F(\theta) = \big\{z\in\mathbb{R}^d,\ \forall\eta\in\mathbb{R}^d,\ F(\eta)\geqslant F(\theta)+z^\top(\eta-\theta)\big\}.$$



For a convex function defined on $\mathbb{R}^d$, the subdifferential happens to be a nonempty convex set at all points θ. 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 in Rockafellar (1997).

The GD 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 often referred to as the "subgradient method" (it is not a descent method anymore, i.e., the function values may increase occasionally).

Exercise 5.19 *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 GD algorithm, now with a decaying step size and a slower rate than for smooth functions.

⚠ As with SGD in the next section, and as opposed to GD for smooth functions in section 5.2, the objective function for the subgradient method for nonsmooth functions may not decrease at every iteration.

Proposition 5.6 (Convergence of the subgradient method) *Assume that F is convex and B-Lipschitz-continuous, and admits a minimizer η_* that satisfies $\|\eta_* - \theta_0\|_2 \leqslant D$. By choosing $\gamma_t = \frac{D}{B\sqrt{t}}$, the iterates $(\theta_t)_{t\geqslant 0}$ of GD on F satisfy*

$$\min_{0\leqslant s\leqslant t-1} \big\{F(\theta_s) - F(\eta_*)\big\} \leqslant DB\frac{2+\log(t)}{2\sqrt{t}}. \tag{5.21}$$

Proof We look at how θ_t approaches η_*; 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 equation (5.7), using the boundedness of the gradients (i.e., $\|F'(\theta_{t-1})\|_2^2 \leqslant B^2$), it follows that

$$\|\theta_t - \eta_*\|_2^2 \leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma_t[F(\theta_{t-1}) - F(\eta_*)] + \gamma_t^2 B^2.$$

We are in a situation where the Lyapunov function $\theta \mapsto \|\theta - \eta_*\|_2^2$ is not decreasing along iterations because of the term $\gamma_t^2 B^2$. 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

$$\gamma_t(F(\theta_{t-1}) - F(\eta_*)) \leqslant \frac{1}{2}\Big(\|\theta_{t-1} - \eta_*\|_2^2 - \|\theta_t - \eta_*\|_2^2\Big) + \frac{1}{2}\gamma_t^2 B^2. \tag{5.22}$$

It is sufficient to sum these inequalities to get (in fact, for any $\eta_* \in \mathbb{R}^d$, 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 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) and $\gamma_t \to 0$ (to converge to the global optimum). Let us choose $\gamma_s = \tau/\sqrt{s}$ for some $\tau > 0$. By using the series-integral comparisons that follow, we get the bound

$$\min_{0 \leqslant s \leqslant t-1} \left\{F(\theta_s) - F(\eta_*)\right\} \leqslant \frac{1}{2\sqrt{t}}\Big(\frac{D^2}{\tau} + \tau B^2(1 + \log(t))\Big).$$

We choose $\tau = D/B$ (which is suggested by optimizing the previous bound without the logarithmic term), which leads to the result. In the proof, we used the inequality $\sum_{s=1}^t \frac{1}{\sqrt{s}} \geqslant \sum_{s=1}^t \frac{1}{\sqrt{t}} = \sqrt{t}$, and the following series-integral comparisons for decreasing functions: $\sum_{s=1}^t \frac{1}{s} = 1 + \sum_{s=2}^t \frac{1}{s} \leqslant 1 + \int_1^t \frac{ds}{s} = 1 + \log(t)$. ■

The proof scheme given here 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 exercise 5.20.

  Exercise 5.20 *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 $\gamma_t' = D/\sqrt{t}$ (independent of B), we get the following 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 exercise 5.21).

  Exercise 5.21 *Let $K \subset \mathbb{R}^d$ be a convex closed set, and denote as $\Pi_K(\theta)$ the orthogonal projection of θ onto K, defined as $\Pi_K(\theta) = \arg\min_{\eta \in K} \|\eta - \theta\|_2^2$. Show that function Π_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 with η_* being a minimizer of F on K, show that the guarantee of proposition 5.6 still holds.*

- The algorithm applies to nondifferentiable convex and Lipschitz objective functions (using subgradients; i.e., any vector satisfying equation (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 and stochastic cases in section 11.1.3.

Exercise 5.22 (♦) *Let $F : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function, and $\psi : \mathbb{R}^d \to \mathbb{R}$ a strictly convex function.*

- *Show that the minimizer of $F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{1}{2\gamma}\|\eta - \theta\|_2^2$ is equal to $\eta = \theta - \gamma F'(\theta)$.*
- *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 nonnegative and equal to zero if and only if $\eta = \theta$.*
- *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 ψ is only defined on an open convex set $K \subset \mathbb{R}^d$, and the gradient ψ' is a bijection from K to $\mathbb{R}^d$.*
- *Provide an explicit form of the resulting algorithm when $\psi(\theta) = \sum_{i=1}^d \theta_i \log \theta_i$.*

- Often, the uniformly averaged iterate is used, such 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 5.23 (♦) *Consider the same assumptions as exercise 5.21 and the same algorithm with orthogonal projections. With D being 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 γ_t is an "anytime" algorithm; that is, it can be stopped at any time t, and the bound in equation (5.21) then applies. Computations are often easier when considering a constant step size γ that depends on the number of iterations T that the user wishes to perform, with T being usually referred to as the "horizon." Starting from equation (5.22), we get the bound

$$\frac{1}{T}\sum_{t=1}^{T} F(\theta_{t-1}) - F(\theta_*) \leqslant \frac{D^2}{2\gamma T} + \frac{\gamma B^2}{2}, \tag{5.23}$$

where the optimal γ can be obtained as $\gamma = D/(B\sqrt{T})$ and there is an optimized rate of $DB/\sqrt{T}$. We gain on the logarithmic factor, but we no longer have an anytime algorithm (since the bound applies only at time T). This also applies 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 undesirable practical behavior as the algorithm makes substantial changes at each iteration that is a power of 2 (see exercise 5.24).

- Stochastic gradients can be used, as presented in section 5.4 (one interpretation is that the subgradient method is so slow that it is robust to noisy gradients).

- The proof technique used for proposition 5.6 will be used several times in this book, for SGD in section 5.4 as well as for online learning in section 11.1.

Exercise 5.24 (Doubling trick for subgradient method) *Consider an algorithm that successively applies the SGD iteration with step size $\gamma = D/(B\sqrt{2^k})$ during 2^k iterations, for $k = 0, 1, \dots$. Show that after t subgradient iterations, the observed best expected value of F is less than a constant times $DB/\sqrt{t}$.*

Exercise 5.25 *Compute all constants for ℓ_2-regularized logistic regression and the support vector machine (SVM) 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 deviations between empirical means and expectations (which correspond to the estimation error for the minimizer of the empirical risk). For convex Lipschitz-continuous losses (with constant G) such as the logistic loss or the hinge loss, for linear predictions with feature ℓ_2-norms smaller than R, a parameter bounded in the ℓ_2-norm by D, we showed in section 4.5.4 that the estimation error for the empirical risk minimizer was upper-bounded by a constant times $GRD/\sqrt{n}$. From equation (5.23), the optimization error after t iterations of the subgradient method is upper-bounded by a constant times $GRD/\sqrt{t}$ 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 full gradient computation requires computing n gradients 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 SGD can turn this number to n with the same upper bound on the generalization error.

5.4 Stochastic Gradient Descent

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 GD algorithm requires computing a "full" gradient $F'(\theta_{t-1})$, which could be costly, as it requires accessing the entire dataset (all n pairs of observations). An alternative is to instead only compute *unbiased* stochastic estimations of the gradient $g_t(\theta_{t-1})$; that is, such that

$$\mathbb{E}[g_t(\theta_{t-1})|\theta_{t-1}] = F'(\theta_{t-1}), \tag{5.24}$$

which could be much faster to compute, in particular by accessing fewer observations.

⚠ Note that we need to condition over θ_{t-1} because θ_{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 get convergence only if the noise magnitudes go to zero (see, e.g., d'Aspremont, 2008; Schmidt et al., 2011 and references therein).

This leads to algorithm 5.2.

Algorithm 5.2 (Stochastic gradient descent) *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 equation (5.24).*

SGD in machine learning. There are two ways to use SGD for supervised machine learning:

- **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 are "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 5.27). We then converge to a *minimizer* η_* *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).

- **Expected risk minimization**: If $F(\theta) = \mathbb{E}[\ell(y, f_\theta(x))]$ is the expected nonobservable 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 unbiasedness. Note here that to preserve unbiasedness, only a single pass is allowed (otherwise, this would create dependencies that would break it) and the randomness comes from the observations (x_t, y_t) themselves.

  Here, we *directly minimize the generalization risk.* The counterpart is that if we have only n samples, then we can run only n SGD iterations, and when n grows, the iterates will converge to a *minimizer* θ_* *of the expected risk.*

  Note that in practice, multiple passes over the data (i.e., using each observation multiple times) often 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 these two situations 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 θ_* to refer to the global minimizer).

SGD 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 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; see exercise 5.28 for SGD for smooth functions). Assumption (H-1) is crucial and is often obtained by considering independent gradient functions g_t, for which we have $\mathbb{E}[g_t(\cdot)] = F'(\cdot)$.

Proposition 5.7 (Convergence of SGD) *Assume that F is convex, is B-Lipschitz, and admits a minimizer θ_* that satisfies $\|\theta_* - \theta_0\|_2 \leqslant D$. Further, assume that the stochastic gradients satisfy assumptions (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 (proposition 5.6), 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 the last inequality with the convexity inequality from equation (5.7) (i.e., $F(\theta_{t-1}) - F(\theta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)$), we get

$$\gamma_t \mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] \leqslant \frac{1}{2}\Big(\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\Big) + \frac{1}{2}\gamma_t^2 B^2. \tag{5.25}$$

Except for the expectations, this is the same bound as equation (5.22), so we can conclude as in the proof of proposition 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 θ_0. The bound is then exactly the same.
- As with the subgradient method in equation (5.23), we can consider a constant step size γ 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 $\gamma = D/(B\sqrt{T})$, which depends on the horizon T, and for the uniformly averaged iterate $\bar{\theta}_t$.
- The result that we obtain, when applied to a single-pass SGD, is a generalization bound; that is, after n iterations, we have an excess risk proportional to $1/\sqrt{n}$, corresponding to the excess risk compared to the best predictor f_θ.

  This is to be compared to using results from chapter 4 (uniform deviation bounds) and nonstochastic GD. 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). Still, we need to add the estimation error of the empirical risk minimizer 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 GD 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 by Agarwal et al. (2012), 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 5.28).
- 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 where feature vectors have ℓ_2-norms bounded by R and a G-Lipschitz-continuous loss, the gradient $g_t(\theta_{t-1})$ is the gradient of function $\theta \mapsto \ell(y_t, \varphi(x_t)^\top\theta)$ taken at θ_{t-1}, and thus its squared norm is less

than $G^2 \cdot \|\varphi(x_t)\|_2^2$. An almost sure bound, therefore, is G^2R^2, while a bound in expectation is $G^2 \cdot \mathbb{E}\big[\|\varphi(x_t)\|_2^2\big]$, which is stronger.

- In section 11.1, we will extend the analysis of this section to *online learning*, where the function that is optimized can change at every iteration, leading to guarantees that are more robust to nonstationary problems.
- We can obtain a result in high probability, using an extension of Hoeffding's inequality to "differences of martingales," as shown in exercise 5.26 below.

Exercise 5.26 (High-probability bound for SGD (♦)) *Using the same assumptions and notations as in proposition 5.7, we consider the projected SGD iteration:* $\theta_t = \Pi_D(\theta_{t-1} - \gamma_t g_t)$*, where* Π_D *is the orthogonal projection on the* ℓ_2*-ball with 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*

$$\gamma_t[F(\theta_{t-1}) - F(\theta_*)] \leqslant \frac{1}{2}\Big(\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\Big) + \frac{1}{2}\gamma_t^2 B^2 + z_t.$$

Using Azuma's inequality (see exercise 1.14), show that with probability at least $1-\delta$*, then, for the weighted average* $\bar{\theta}_t$ *defined in proposition 5.7, for any step sizes* γ_t*:*

$$F(\bar{\theta}_t) - F(\theta_*) \leqslant \frac{2D^2}{\sum_{s=1}^t \gamma_s} + B^2 \frac{\sum_{s=1}^t \gamma_s^2}{2\sum_{s=1}^t \gamma_s} + 4BD\frac{\big(\sum_{s=1}^t \gamma_s^2\big)^{1/2}}{\sum_{s=1}^t \gamma_s}\sqrt{2\log\frac{1}{\delta}},$$

and for a constant step size, $\gamma_t = \gamma$*,* $F(\bar{\theta}_t) - F(\theta_*) \leqslant \frac{2D^2}{\gamma T} + \frac{\gamma B^2}{2} + \frac{4DB}{\sqrt{t}}\sqrt{2\log\frac{1}{\delta}}$ *(for the uniformly averaged iterate).*

SGD or GD on the empirical risk? As seen previously, the number of iterations to reach a given precision will be larger for SGD than for smooth deterministic GD, 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 GD could be preferred. However, for low precision and large n, SGD is the method of choice (see also recent improvements that allow exponential convergence with cheap iteration costs, described in section 5.4.4).

In particular, as mentioned earlier, for the linear prediction case described at the end of section 5.3, we obtain the exact same rate in proposition 5.7 as for nonstochastic GD on the empirical risk. If sampling from the n observations with replacement, after $t = n$ steps, the sum of the optimization error and the estimation error of the empirical risk minimizer is of the same order $O(GRD/\sqrt{n})$, with 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*, proposition 5.7 is *directly* a generalization performance result with the same rate.

Exercise 5.27 (Minibatch SGD) *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 θ_{t-1}. Show that the convergence result of proposition 5.7 still holds.* ($\blacklozenge$) *Which assumption on gradients would improve the convergence rate?*

Exercise 5.28 (SGD for smooth functions ($\blacklozenge$)) *Consider independent and identically distributed (i.i.d.) convex L-smooth random functions $f_t : \mathbb{R}^d \to \mathbb{R}$, $t \geqslant 1$, with expectation $F : \mathbb{R}^d \to \mathbb{R}$, which has a minimizer $\theta_* \in \mathbb{R}^d$. Consider the SGD recursion $\theta_t = \theta_{t-1} - \gamma_t f_t'(\theta_{t-1})$, with γ_t being a deterministic step-size sequence. Using co-coercivity (proposition 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 proposition 5.7 to obtain an explicit rate in $O(1/\sqrt{t})$. ($\blacklozenge$) *Show that the minibatch version leads to an improvement in the rate (as opposed to the nonsmooth case in exercise 5.27).*

Exercise 5.29 (Nonuniform sampling ($\blacklozenge$)) *Consider the function $F : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$, which is convex with respect to the first variable, with a subgradient $F'(\theta, z)$ with respect to the first variable that is bounded in the ℓ_2-norm by a constant $B(z)$ that depends on z. Consider a distribution p on $\mathcal{Z}$. 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 i.i.d. random z_t, $t \geqslant 1$. 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 SVM when applying SGD to the empirical risk.*

Exercise 5.30 (SGD for nonconvex functions) *Consider an L-smooth potentially nonconvex function F, and the SGD recursion with constant step size γ, with unbiased and bounded gradient estimates (e.g., assumptions (H-1) and (H-2)).*

- *Show that* $\mathbb{E}\big[F(\theta_t)\big] \leqslant \mathbb{E}\big[F(\theta_{t-1})\big] - \gamma\mathbb{E}\big[\|F'(\theta_{t-1})\|_2^2\big] + \frac{LB^2\gamma^2}{2}$.
- *Show that* $\frac{1}{t}\sum_{s=1}^t \mathbb{E}\big[\|F'(\theta_{s-1})\|_2^2\big] \leqslant \frac{1}{\gamma t}\big[F(\theta_0) - \inf_{\eta\in\mathbb{R}^d} F(\eta)\big] + \frac{LB^2\gamma}{2}$.

5.4.1 Strongly Convex Problems ($\blacklozenge$)

We consider the regularized problem $G(\theta) = F(\theta) + \frac{\mu}{2}\|\theta\|_2^2$, with the same assumptions (H-1) and (H-2) as before, and started at $\theta_0 = 0$. The SGD iteration is then, with $g_t(\theta_{t-1})$ a stochastic subgradient of F at θ_{t-1}:

$$\theta_t = \theta_{t-1} - \gamma_t\big[g_t(\theta_{t-1}) + \mu\theta_{t-1}\big]. \tag{5.26}$$

We then have an improved convergence rate in $O(1/t)$ with a different decay for the step size γ_t, in $1/t$ rather than $1/\sqrt{t}$.

Proposition 5.8 (Convergence of SGD for strongly convex problems) *Assume that F is convex, is B-Lipschitz, and that $F + \frac{\mu}{2}\|\cdot\|_2^2$ admits a necessarily unique minimizer θ_*. Assume that the stochastic gradient g satisfies assumptions (H-1) and (H-2).*

Then, choosing $\gamma_t = 1/(\mu t)$, the iterates $(\theta_t)_{t\geqslant 0}$ of the SGD recursion from equation (5.26) 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 this proof is essentially the same as for convex problems, leading to (with the new terms in blue):

$$\begin{aligned}\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big] &= \mathbb{E}\big[\|\theta_{t-1} - \gamma_t(g_t(\theta_{t-1}) + \mu\theta_{t-1}) - \theta_*\|_2^2\big] \\ &= \mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - 2\gamma_t\mathbb{E}\big[(g_t(\theta_{t-1}) + \mu\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big] \\ &\qquad + \gamma_t^2\mathbb{E}\big[\|g_t(\theta_{t-1}) + \mu\theta_{t-1}\|_2^2\big].\end{aligned}$$

From the iterations in equation (5.26), 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, since all gradients are bounded in norm by B, $\|g_t(\theta_{t-1}) + \mu\theta_{t-1}\|_2^2$ is always less than $4B^2$. Therefore,

$$\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big] \leqslant \mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - 2\gamma_t\mathbb{E}\big[G'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\big] + 4\gamma_t^2B^2.$$

Therefore, combining with the inequality coming from strong convexity (see equation (5.9)) $G(\theta_{t-1}) - G(\theta_*) + \frac{\mu}{2}\|\theta_{t-1} - \theta_*\|_2^2 \leqslant G'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)$, it follows

$$\gamma_t\mathbb{E}[G(\theta_{t-1}) - G(\theta_*)] \leqslant \frac{1}{2}\big((1-\gamma_t\mu)\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\big) + 2\gamma_t^2B^2,$$

and thus, now using the specific step-size choice $\gamma_t = 1/(\mu t)$:

$$\begin{aligned}\mathbb{E}[G(\theta_{t-1}) - G(\theta_*)] &\leqslant \frac{1}{2}\big((\gamma_t^{-1} - \mu)\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \gamma_t^{-1}\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\big) + 2\gamma_t B^2, \\ &= \frac{1}{2}\big(\mu(t-1)\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mu t\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\big) + \frac{2B^2}{\mu t}.\end{aligned}$$

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 exercise 5.31.

  Exercise 5.31 (♦) *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), and the recursion $\theta_t = \theta_{t-1} - \gamma[F'(\theta_{t-1}) + \varepsilon_t]$, where all ε_t's are independent, with zero mean and covariance matrix equal to C. Compute explicitly $\mathbb{E}\big[F(\theta_t) - F(\theta_*)\big]$, 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 by Agarwal et al. (2012), 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 μ is very small.

  Note also the loss of adaptivity: the step size now depends on the problem's difficulty (this was different for deterministic GD). See the experiments that follow for illustrations.
- When applied in a single pass over the data, the resulting rate in $O(B^2/\mu n)$ is the same, up to logarithmic terms, as the generalization bound for the minimizer of the regularized empirical risk in proposition 4.6.

Exercise 5.32 *With the same assumptions as proposition 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* $\mathbb{E}\big[G(\bar\theta_t) - G(\theta_*)\big] \leqslant \frac{8B^2}{\mu(t+1)}$.

Exercise 5.33 *Consider the minimization of* $F(\theta) = \mathbb{E}\big[\|\theta - z\|_2^2/2\big]$ *from i.i.d. observations* $z_1, \dots, z_t$. *Show that the* t*-th iterate of SGD equals* $\frac{1}{t}(z_1 + \cdots + z_t)$.

Experiments. Here, 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 observe features with the ℓ_2-norm bounded by R. We consider the hinge loss with a squared ℓ_2-regularizer $\frac{\mu}{2}\|\cdot\|_2^2$ (i.e., the SVM presented in section 4.1.2). We measure the excess training objective. We consider two values of μ 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 the larger value of μ (top plot), the strongly convex step size is better. This is not the case for small μ (bottom plot). Note the strong variability for the step size $\gamma_t = 1/(\mu t)$ in early iterations.

These experiments highlight the danger of a step size equal to $1/(\mu t)$. In practice, it is often preferable to use $\gamma_t = 1/(B^2\sqrt{t} + \mu t)$, as shown in exercise 5.34.

Exercise 5.34 (♦♦) *With the same assumptions as in proposition 5.8, with step size* $\gamma_t = 1/(B^2\sqrt{t} + \mu t)$, *provide a convergence rate for the averaged iterate.*

5.4.2 Adaptive Methods (♦)

The discussion on preconditioning for GD on smooth functions at the end of section 5.2.5 can be adapted to stochastic gradient methods for nonsmooth 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 the ℓ_2-norm by R, and a convex G-Lipschitz-continuous loss function, in the setting of proposition 5.7. For a constant

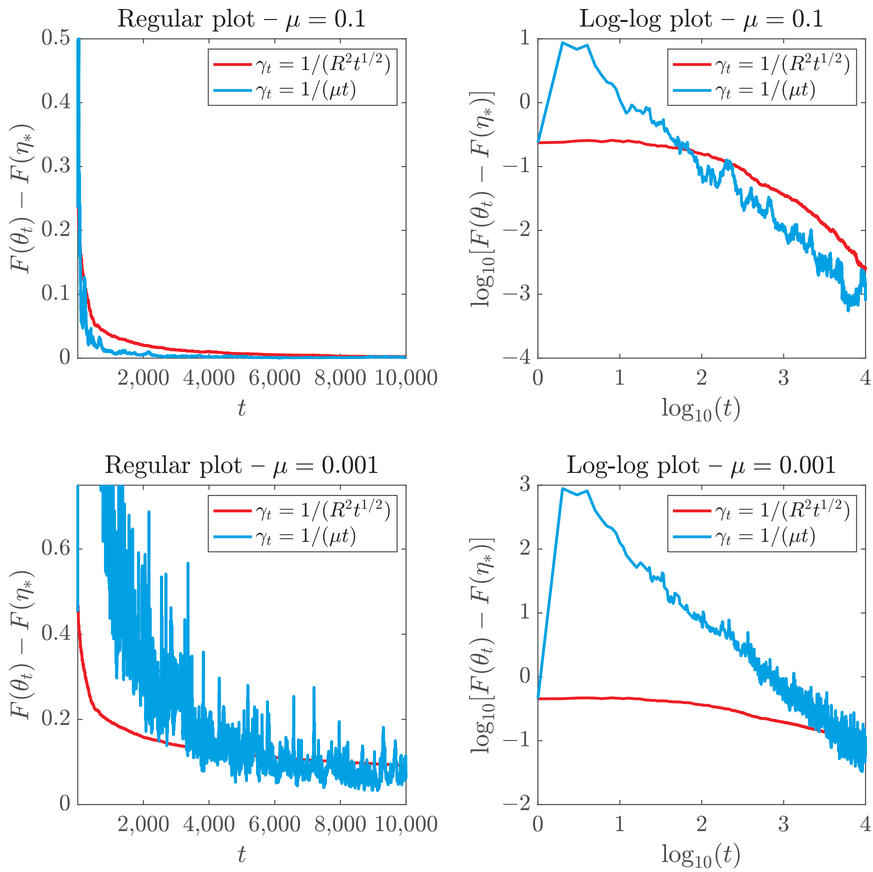


Figure 5.2. Comparison of step sizes for SGD for the SVM, for two values of the regularization parameter μ (top: large $\mu = 10^{-1}$; bottom: small $\mu = 10^{-3}$). The performance is measured with a single run (hence the variability) on the excess training objective (left: regular plot; right: log-log plot).

step size γ, in the proof of proposition 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 γ leads to the overall rate of $\frac{G\|\theta_*\|_2}{\sqrt{t}}\sqrt{\operatorname{tr}[\Sigma]}$.

As done at the end of section 5.2.5, premultiplying 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 φ by $A^\top\varphi$, and θ_* by $A^{-1}\theta_*$. The complexity bound then becomes

$$\frac{1}{2\gamma t}\theta_*^\top(AA^\top)^{-1}\theta_* + \frac{\gamma G^2}{2}\operatorname{tr}[\Sigma AA^\top].$$

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 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 $\frac{G\|\theta_*\|_\infty}{\sqrt{t}}\sqrt{d\sum_{j=1}^d \Sigma_{jj}}$; this is in particular the case when the Σ_{jj}'s have heterogeneous values.

In practice, before running the learning algorithm, we can estimate the required elements of Σ, the noncentered covariance matrix of the features, and, more generally, the covariance of the gradients. These quantities can be estimated online, leading to the Adagrad (Duchi et al., 2011), or Adam (Kingma and Ba, 2014) algorithms, 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 is 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 ε_i has mean zero and variance σ^2, and is independent of x_i. The goal of this section is to relate the performance of single-pass SGD to the regularized empirical risk minimization studied in sections 3.3 and 3.6, and to study the impact of noise on 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

$$\theta_t - \theta_* = (I - \gamma\varphi(x_t)\varphi(x_t)^\top)(\theta_{t-1} - \theta_*) + \gamma\varepsilon_t\varphi(x_t). \tag{5.27}$$

Thus, as 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 equation (5.27),

$$\begin{aligned}\|\theta_t - \theta_*\|_2^2 \;=\; & \|\theta_{t-1} - \theta_*\|_2^2 + \|\gamma\varphi(x_t)\varphi(x_t)^\top(\theta_{t-1} - \theta_*)\|_2^2 \\ & -2\gamma(\theta_{t-1} - \theta_*)^\top\varphi(x_t)\varphi(x_t)^\top(\theta_{t-1} - \theta_*) + \|\gamma\varepsilon_t\varphi(x_t)\|_2^2 \\ & +2\gamma\varepsilon_t\varphi(x_t)^\top(I - \gamma\varphi(x_t)\varphi(x_t)^\top)(\theta_{t-1} - \theta_*),\end{aligned}$$

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 $\mathbb{E}\big[\|\varphi(x_t)\|_2^2\varphi(x_t)\varphi(x_t)^\top\big] \preccurlyeq R^2\Sigma$, for $\Sigma = \mathbb{E}\big[\varphi(x_t)\varphi(x_t)^\top\big]$:

$$\mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big|\mathcal{F}_{t-1}\big] \;\leqslant\; \|\theta_{t-1} - \theta_*\|_2^2 + (\gamma^2R^2 - 2\gamma)(\theta_{t-1} - \theta_*)^\top\Sigma(\theta_{t-1} - \theta_*) + \gamma^2\sigma^2R^2.$$

This leads to, with $F(\theta) - F(\theta_*) = \frac{1}{2}(\theta - \theta_*)^\top\Sigma(\theta - \theta_*)$, for $\gamma \leqslant 1/R^2$,

$$\mathbb{E}\big[F(\theta_{t-1}) - F(\theta_*)\big] \leqslant \frac{1}{2\gamma}\Big(\mathbb{E}\big[\|\theta_{t-1} - \theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t - \theta_*\|_2^2\big]\Big) + \frac{\gamma\sigma^2R^2}{2},$$

and thus, for the average $\bar\theta_t = \frac{1}{t}\sum_{s=1}^t \theta_{s-1}$, using Jensen's inequality,

$$\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta_*) \leqslant \frac{1}{2\gamma t}\|\theta_0 - \theta_*\|_2^2 + \frac{\gamma\sigma^2R^2}{2}, \tag{5.28}$$

which is a similar result to the nonsmooth case but with an explicit bias/variance decomposition where the noise variance σ^2 explicitly appears, as well as the norm of θ_*. Note that it requires the step size to depend on the number of total iterations to obtain convergence. When applied in a single pass over the data, we end up with a generalization bound which is similar to the one for ridge regression in section 3.6, but now with an efficient algorithm.

Exercise 5.35 (Weaker assumptions) *Consider a joint distribution on $(x, y) \in \mathcal{X}\times\mathbb{R}$, and a feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$ bounded by R in the ℓ_2-norm. Denoting θ_* a minimizer of $\mathbb{E}\big[(y - \varphi(x)^\top\theta)^2\big]$ with respect to θ, show that the bound in equation (5.28) applies with $\sigma^2 = \mathbb{E}\big[(y - \varphi(x)^\top\theta_*)^2\big]$.*

However, for least-mean-squares, a finer analysis can be performed, explicitly allowing constant step sizes and a clear relationship with generalization bounds for least-squares regression outlined in chapter 3, which we present next.

Finer analysis of the LMS recursion (♦♦). A detailed analysis of the LMS recursion in equation (5.27) is out of the scope of this book. However, a simplified recursion with essentially the same behavior can be analyzed with simple linear algebra tools. To obtain this simplified recursion, we rewrite equation (5.27) 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 θ_{t-1}) and has a constant nonzero 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 θ_*. The simplified recursion ignores that term, and we now study the recursion (started at $\eta_0 = \theta_0$):

$$\eta_t - \theta_* = (I - \gamma\Sigma)(\eta_{t-1} - \theta_*) + \gamma\varepsilon_t\varphi(x_t), \tag{5.29}$$

which also corresponds to replacing $\varphi(x_t)\varphi(x_t)^\top$ in equation (5.27) by its expectation Σ.

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 that only depends on the initialization (i.e., $(I - \gamma\Sigma)^t(\eta_0 - \theta_*)$), which is precisely the deterministic recursion from section 5.2.1, and which we call the "bias," and a part that depends on the noise variables ε_u, $u = 1, \dots, t$, which we refer to as the "variance." Assuming that these noise variables are independent of x, the two parts can be considered totally independently when looking at expectations.

We then have, for the averaged iterates, using $\sum_{v=0}^{t-1}(I - \gamma\Sigma)^v = (\gamma\Sigma)^{-1}\big[I - (I - \gamma\Sigma)^t\big]$ several times,

$$\begin{aligned}
\bar\eta_t^{(\text{bias})} - \theta_* &= \frac{1}{t}\sum_{v=0}^{t-1}(I - \gamma\Sigma)^v(\eta_0 - \theta_*) = \frac{1}{t}(\gamma\Sigma)^{-1}\big[I - (I - \gamma\Sigma)^t\big](\eta_0 - \theta_*) \\
\bar\eta_t^{(\text{var})} - \theta_* &= \frac{1}{t}\sum_{v=1}^{t-1}\sum_{u=1}^{v}\gamma\varepsilon_u(I - \gamma\Sigma)^{v-u}\varphi(x_u) = \frac{\gamma}{t}\sum_{u=1}^{t-1}\sum_{v=u}^{t-1}(I - \gamma\Sigma)^{v-u}\varepsilon_u\varphi(x_u) \\
&= \frac{1}{t}\sum_{u=1}^{t-1}\Sigma^{-1}\big[I - (I - \gamma\Sigma)^{t-u}\big]\varepsilon_u\varphi(x_u),
\end{aligned}$$

leading to, using $\gamma \leqslant \frac{1}{R^2}$ (which implies $I - \gamma\Sigma \succcurlyeq 0$) for $t > 0$:

$$\begin{aligned}
\big\|\bar\eta_t^{(\text{bias})} - \theta_*\big\|_\Sigma^2 &= \frac{1}{t^2}(\eta_0 - \theta_*)^\top(\gamma\Sigma)^{-2}\big[I - (I - \gamma\Sigma)^t\big]^2\Sigma(\eta_0 - \theta_*) \\
&\leqslant \frac{1}{\gamma^2 t^2}(\eta_0 - \theta_*)^\top\Sigma^{-1}(\eta_0 - \theta_*), \\
\mathbb{E}\Big[\big\|\bar\eta_t^{(\text{var})} - \theta_*\big\|_\Sigma^2\Big] &= \frac{\sigma^2}{t^2}\sum_{u=1}^{t-1}\operatorname{tr}\Big[\Sigma^2\Sigma^{-2}\big[I - (I - \gamma\Sigma)^{t-u}\big]^2\Big] \leqslant \frac{\sigma^2 d}{t}.
\end{aligned}$$

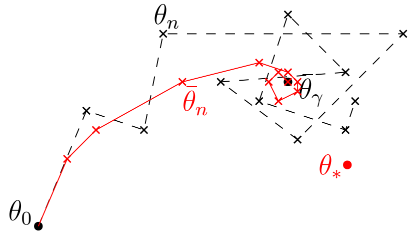


Figure 5.3. The iterates of SGD form a Markov chain, which is homogeneous when the step size γ is constant. It typically converges to a stationary distribution with expectation $\bar{\theta}_\gamma$, which happens to be the global minimum θ_* for quadratic costs (and with a deviation of γ^2 in general). The nonaveraged iterates go from the initial condition θ_0 to the vicinity of $\bar{\theta}_\gamma$, while the averaged iterates converge to that expectation $\bar{\theta}_\gamma$.

We thus obtain two terms, the variance in $\frac{\sigma^2 d}{t}$, which exists because the optimal prediction is not equal to the response, and the bias 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 nonaveraged iterates: the bias 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 Σ is strictly positive. The variance term is only of order $\gamma\sigma^2 \operatorname{tr}[\Sigma]$ (thus, with no convergence). This is illustrated in figure 5.3; note that averaged SGD with constant step size converges to the global optimum only for the quadratic loss (see Bach and Moulines, 2013, for an extension to the logistic loss).

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 is the same as $\sigma^2 d/n = O(1/n)$, while the bias is in $O(1/n^2)$ and seems smaller in the dependence on 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), in order 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 now consider a finite sum $F(\theta) = \frac{1}{n}\sum_{i=1}^n f_i(\theta)$, where each f_i is R^2-smooth (e.g., logistic regression with features bounded by R in the ℓ_2-norm), and which is such that F is μ-strongly convex (e.g., by adding $\frac{\mu}{2}\|\theta\|_2^2$ to each f_i, or by benefiting from the strong convexity of the sum). We denote by $\kappa = R^2/\mu$ the condition number of the problem (note that it is larger than L/μ, 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 past gradients in memory. 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

$$\theta_t = \theta_{t-1} - \gamma \bigg[f'_{i(t)}(\theta_{t-1}) + \frac{1}{n} \sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)} \bigg],$$

with $z_{i(t)}^{(t)} = f'_{i(t)}(\theta_{t-1})$ and all others $z_i^{(t)}$ left unchanged (i.e., the same as $z_i^{(t-1)}$). Stated 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*, as 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* θ_t converges to η_* *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 $\gamma = \frac{1}{4R^2}$,*

$$\mathbb{E}\big[\|\theta_t - \eta_*\|_2^2\big] \leqslant \Big(1 - \min\Big\{\frac{1}{3n}, \frac{3\mu}{16R^2}\Big\}\Big)^t \big(1 + \frac{n}{4}\big)\|\theta_0 - \eta_*\|_2^2. \tag{5.30}$$

Proof (♦♦) As with all proofs in this chapter, 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 $\omega_t = \big[f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\big]$:

$$\begin{aligned}
\|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top \omega_t + \gamma^2 \big\|\omega_t\big\|_2^2 \text{ by expanding the square,} \\
\mathbb{E}_{i(t)}\big[\|\theta_t - \eta_*\|_2^2\big] &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \\
&\quad + \gamma^2 \mathbb{E}_{i(t)}\bigg[\bigg\| f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)} \bigg\|_2^2\bigg],
\end{aligned}$$

using the unbiasedness of the stochastic gradient. We further get

$$\begin{aligned}\mathbb{E}_{i(t)}\big[\|\theta_t-\eta_*\|_2^2\big] &\leqslant \|\theta_{t-1}-\eta_*\|_2^2 - 2\gamma(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) \\ &\quad +2\gamma^2\mathbb{E}_{i(t)}\Big[\big\|f'_{i(t)}(\theta_{t-1})-f'_{i(t)}(\eta_*)\big\|_2^2\Big] + 2\gamma^2\mathbb{E}_{i(t)}\Big[\big\|f'_{i(t)}(\eta_*)-z_{i(t)}^{(t-1)}+\frac{1}{n}\sum_{i=1}^n z_i^{(t-1)}\big\|_2^2\Big],\end{aligned}$$

using $\|a+b\|_2^2 \leqslant 2\|a\|_2^2+2\|b\|_2^2$. To bound $\mathbb{E}_{i(t)}\Big[\big\|f'_{i(t)}(\theta_{t-1})-f'_{i(t)}(\eta_*)\big\|_2^2\Big]$, we use co-coercivity of all functions f_i (see proposition 5.4), to get

$$\begin{aligned}\mathbb{E}_{i(t)}\Big[\big\|f'_{i(t)}(\theta_{t-1})-f'_{i(t)}(\eta_*)\big\|_2^2\Big] &= \frac{1}{n}\sum_{i=1}^n \big\|f'_i(\theta_{t-1})-f'_i(\eta_*)\big\|_2^2 \\ &\leqslant \frac{1}{n}\sum_{i=1}^n R^2[f'_i(\theta_{t-1})-f'_i(\eta_*)]^\top(\theta_{t-1}-\eta_*) \\ &= R^2F'(\theta_{t-1})^\top(\theta_{t-1}-\eta_*) \text{ since } \sum_{i=1}^n f'_i(\eta_*)=0. \quad (5.31)\end{aligned}$$

To bound $\mathbb{E}_{i(t)}\Big[\big\|f'_{i(t)}(\eta_*)-z_{i(t)}^{(t-1)}+\frac{1}{n}\sum_{i=1}^n z_i^{(t-1)}\big\|_2^2\Big]$, we use the inequality between variance and second moment $\mathbb{E}_{i(t)}\big[\|Z-\mathbb{E}_{i(t)}Z\|_2^2\big] \leqslant \mathbb{E}_{i(t)}\big[\|Z\|_2^2\big]$. We thus get

$$\begin{aligned}\mathbb{E}_{i(t)}\big[\|\theta_t-\eta_*\|_2^2\big] &\leqslant \|\theta_{t-1}-\eta_*\|_2^2 - 2\gamma(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) + 2\gamma^2R^2(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) \\ &\quad +2\gamma^2\frac{1}{n}\sum_{i=1}^n \big\|f'_i(\eta_*)-z_i^{(t-1)}\big\|_2^2, \\ &= \|\theta_{t-1}-\eta_*\|_2^2 - 2\gamma(1-\gamma R^2)(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) \\ &\quad +2\frac{\gamma^2}{n}\sum_{i=1}^n \big\|f'_i(\eta_*)-z_i^{(t-1)}\big\|_2^2.\end{aligned}$$

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 $z_i^{(t)}$,

$$\begin{aligned}\sum_{i=1}^n \big\|f'_i(\eta_*)-z_i^{(t)}\big\|_2^2 &= \sum_{i=1}^n \big\|f'_i(\eta_*)-z_i^{(t-1)}\big\|_2^2 \\ &\quad -\big\|f'_{i(t)}(\eta_*)-z_{i(t)}^{(t-1)}\big\|_2^2 + \big\|f'_{i(t)}(\eta_*)-f'_{i(t)}(\theta_{t-1})\big\|_2^2.\end{aligned}$$

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 equation (5.31). Thus, for a positive real number Δ to be chosen later,

$$\begin{aligned}
&\mathbb{E}_{i(t)}\Big[\|\theta_t-\eta_*\|_2^2+\Delta\sum_{i=1}^n\big\|f_i'(\eta_*)-z_i^{(t)}\big\|_2^2\Big]\\
\leqslant\ & \|\theta_{t-1}-\eta_*\|_2^2-2\gamma(1-\gamma R^2-\frac{R^2\Delta}{2\gamma})(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1})\\
&\qquad\qquad+\big[2\frac{\gamma^2}{n\Delta}+(1-1/n)\big]\Delta\sum_{i=1}^n\big\|f_i'(\eta_*)-z_i^{(t-1)}\big\|_2^2.
\end{aligned}$$

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

$$\begin{aligned}
&\mathbb{E}_{i(t)}\Big[\|\theta_t-\eta_*\|_2^2+\Delta\sum_{i=1}^n\big\|f_i'(\eta_*)-z_i^{(t)}\big\|_2^2\Big]\\
\leqslant\ & \Big(1-\min\Big\{\frac{1}{3n},\frac{3\mu}{16R^2}\Big\}\Big)\Big[\|\theta_{t-1}-\eta_*\|_2^2+\Delta\sum_{i=1}^n\big\|f_i'(\eta_*)-z_i^{(t-1)}\big\|_2^2\Big].
\end{aligned}$$

Thus, by applying this inequality at all times from 1 to t, we get:

$$\mathbb{E}\big[\|\theta_t-\eta_*\|_2^2\big]\leqslant\Big(1-\min\Big\{\frac{1}{3n},\frac{3\mu}{16R^2}\Big\}\Big)^t\Big[\|\theta_0-\eta_*\|_2^2+\frac{3}{16R^4}\sum_{i=1}^n\big\|f_i'(\eta_*)-z_i^{(0)}\big\|_2^2\Big].$$

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 equation (5.30). ■

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 (i.e., 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 the functions (while some will be seen several times).

  To have a contracting effect of ε (i.e., 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 having at least this number of iterations $t\geqslant\max\big\{3n,\frac{16R^2}{3\mu}\big\}\log\frac{2n}{\varepsilon}$. 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 GD with step size $\gamma=1/R^2$ (which is the simplest step size that can be computed easily), whose complexity is $dn\dfrac{R^2}{\mu}\log\dfrac{1}{\varepsilon}$.

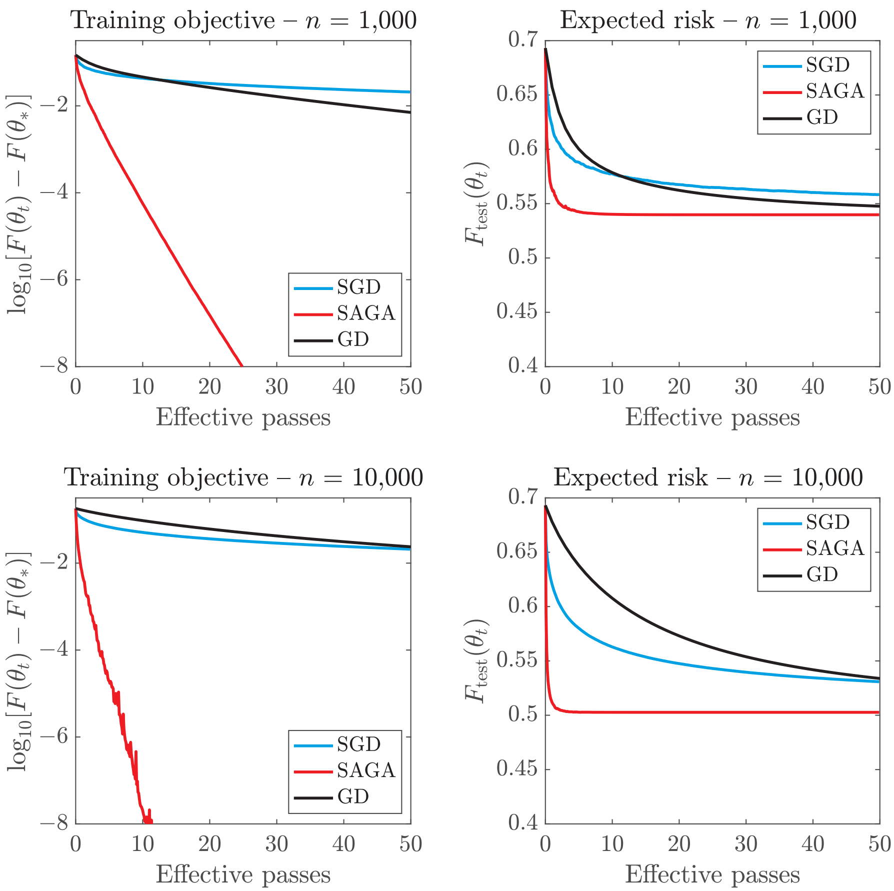


Figure 5.4. Comparison of stochastic gradient algorithms for logistic regression. Top: $n = 1,000$; bottom: $n = 10,000$. Left: training objective in semilog plot; right: expected risk estimated with n (independent) test points.

We replace the product of n and condition number $\kappa = \frac{R^2}{\mu}$ by a sum, which is significant where κ 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).
- Note that these fast algorithms allow very small optimization errors and the best testing risks will typically be obtained after a few (10 to 100) passes.

Experiments. Here, we consider ℓ_2-regularized logistic regression and 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 the regularization parameter $\mu = R^2/n$. See figure 5.4 (top: small n, bottom: 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 in figure 5.4 (right), where SGD always dominates GD. SAGA gets to machine precision after 50 effective passes over the data in these 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, with the main rates that we have seen in this chapter (and some that we have not seen) for convex objective functions. We distinguish between convex and strongly convex, and between smooth and nonsmooth, as well as between deterministic and stochastic methods. In the following table, L is the smoothness constant, μ the strong convexity constant, and B the Lipschitz constant (below, we ignore multiplicative factors that involve the initial distance to optimum in ℓ_2-norm or function values, to focus on the dependence in t and the conditioning of the problem for the strongly convex case):

	Convex	Strongly Convex
Nonsmooth	Deterministic: $1/\sqrt{t}$ Stochastic: $1/\sqrt{t}$	Deterministic: $B^2/(t\mu)$ Stochastic: $B^2/(t\mu)$
Smooth	Deterministic: $1/t^2$ Stochastic: $1/\sqrt{t}$ Finite sum: n/t	Deterministic: $\exp(-t\sqrt{\mu/L})$ Stochastic: $L/(t\mu)$ 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 ε. This corresponds to inverting each formula for ε as a function of t to a formula for t as a function of ε. This leads to the following table:

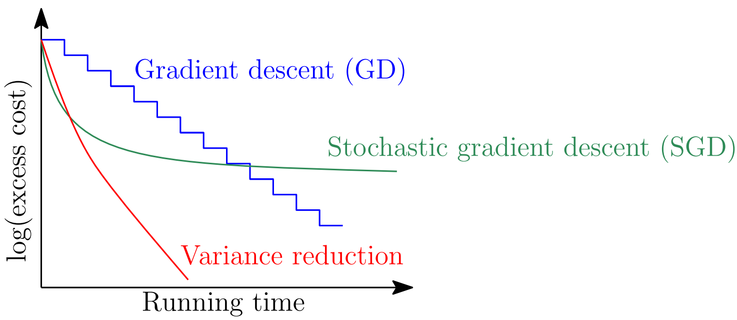
	Convex	Strongly Convex
Nonsmooth	Deterministic: $1/\varepsilon^2$ Stochastic: $1/\varepsilon^2$	Deterministic: $B^2/(\varepsilon\mu)$ Stochastic: $B^2/(\varepsilon\mu)$
Smooth	Deterministic: $1/\sqrt{\varepsilon}$ Stochastic: $1/\varepsilon^2$ Finite sum: n/ε	Deterministic: $\sqrt{L/\mu}\log(1/\varepsilon)$ Stochastic: $L/(\varepsilon\mu)$ Finite sum: $\max\{n, L/\mu\}\log(1/\varepsilon)$

As in the rest of the book, where we obtain explicit convergence rates, the homogeneity of all quantities can be checked (see exercise 5.36). In the context of optimization, this ensures that algorithms are invariant under a change of variable $\theta \to \alpha\theta$ for $\alpha \neq 0$.

Exercise 5.36 *Check the homogeneity of all quantities of this section (step size and convergence rates).*

Note that many important themes in first-order optimization have not been discussed here, such as Frank-Wolfe methods (presented in chapter 9), coordinate descent, or duality. See Nesterov (2018) and Bubeck (2015) for further details. See also chapters 7 and 9 for optimization methods for kernel methods and neural networks.

For strongly convex smooth problems, the following illustration also provides a good summary, with GD being along a line in a semilog plot (i.e., 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 available only through dot products $\varphi(x)^\top \varphi(x')$. In section 10.3, we also consider infinite-dimensional sets of predictors optimized through boosting procedures.

Beyond convex problems. Complexity bounds can be obtained beyond convex problems, as shown briefly in section 5.2.6 and exercise 5.30. However, they certify only 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 nonconvex objective functions, which we consider in chapter 9.

Generalization bounds: Rademacher or SGD? In chapter 4, we showed 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 (but 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

- Nonparametric estimation: This is the book's first chapter discussing nonparametric methods, which 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, these estimators are nonlinear in the input variables).
- Partitioning estimates: The input space is cut into nonoverlapping cells, and the predictor is piecewise constant.
- Nadaraya-Watson estimators (aka 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, with a majority vote among the corresponding labels.
- Consistency: All these methods can provably learn complex Lipschitz-continuous nonlinear 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

As in previous chapters, we consider the supervised learning setup, 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 are assumed to be 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 when 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 computed with respect to the distribution p.

⚠ As 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 $\hat{f}$ that we will use depends on the training data, 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: $\mathcal{Y} = \{-1, 1\}$ (or often $\mathcal{Y} = \{0, 1\}$), 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 (1) the Bayes predictor is not unique, but all Bayes predictors lead to the same Bayes risk, and (2) the Bayes risk is usually nonzero (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 $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ *and* the loss ℓ, with the goal of minimizing the risk or the excess risk $\mathcal{R}(f) - \mathcal{R}^*$. We have the following special cases to consider:

- For binary classification: $\mathcal{Y} = \{-1, 1\}$ 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 multicategory classification with the Bayes predictor $f_*(x) \in \underset{i \in \{1,\ldots,k\}}{\arg\max}\, \mathbb{P}(y = i|x)$.

  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 $\mathcal{R}(f) - \mathcal{R}^* = \int_{\mathcal{X}} (f(x) - f_*(x))^2 dp(x) = \|f - f_*\|^2_{L_2(p)}$.

In chapters 3 and 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. Later in this book, we will study kernel methods (chapter 7), neural networks (chapter 9), and boosting methods (section 10.3).

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: $\hat{f}(x) \in \underset{j\in\{1,\dots,k\}}{\arg\max}\, \widehat{\mathbb{P}}(y=j|x)$.
- 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 δ_{y_i} is the Dirac probability distribution at y_i (putting a unit mass at y_i), and the weight functions $\hat{w}_i : \mathcal{X} \to \mathbb{R}$, $i = 1,\dots,n$ depend on the input data only (for simplicity) and satisfies (almost surely in x):

$$\forall x \in \mathcal{X}, \quad \forall i \in \{1,\dots,n\},\ \hat{w}_i(x) \geqslant 0,\ \text{and} \ \sum_{i=1}^{n} \hat{w}_i(x) = 1. \tag{6.1}$$

These conditions ensure that for all $x \in \mathcal{X}$, $\widehat{p}(\cdot|x)$ is a probability distribution.

⚠ Some references allow the weights not to sum to 1.

For our running examples, this leads to the following predictions:

- For classification: $\hat{f}(x) \in \underset{j\in\{1,\dots,k\}}{\arg\max} \sum_{i=1}^{n} \hat{w}_i(x) 1_{y_i=j}$; 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: as a function of the response vector in $\mathbb{R}^n$, the estimator is linear (note that this is also the case for kernel ridge regression in chapter 7; see section 7.6.1). 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 $H_{ij} = \hat{w}_j(x_i)$. From equation (6.1), the smoothing matrix H is stochastic; that is, with nonnegative elements and rows that sum to one.

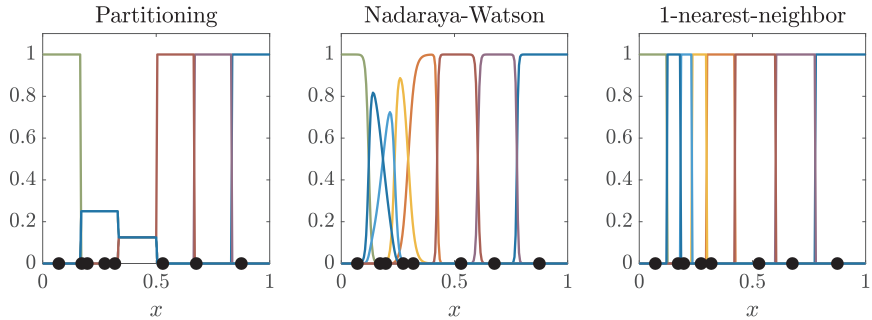


Figure 6.1. Weights of linear estimators in dimension $d = 1$ for the three types of local averaging estimators. The $n = 8$ weight functions $x \mapsto \hat{w}_i(x)$, $i = 1, \ldots, n$, are plotted with the observations in black.

⚠ For $\mathcal{X} = \mathbb{R}^d$, linear estimators typically do not lead to prediction functions that are linear in their inputs.

Note that in addition to being a linear estimator, the estimator satisfies additional properties: if the same constant is added to all outputs, the exact same constant 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 $i \in \{1, \ldots, n\}$, 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 large for training points x_i that are close to x, and small otherwise. We now show three classical ways of building them: (1) partition estimators, (2) nearest-neighbors, and (3) Nadaraya-Watson estimators (aka kernel regression). See the examples shown 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 to be finite for simplicity, equal to $\{1, \ldots, |J|\}$), then we can consider for any $x \in \mathcal{X}$ the corresponding element $A(x)$ of the partition (i.e., $A(x)$ is the unique A_j, $j \in J$, such that $x \in A_j$), and define

$$\hat{w}_i(x) = \frac{1_{x_i \in A(x)}}{\sum_{i'=1}^{n} 1_{x_{i'} \in A(x)}}, \tag{6.2}$$

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 $i \in \{1, \ldots, n\}$. This implies that each $\hat{w}_i$ is piecewise constant with respect to the partition; that is, for any nonempty cell A_j (i.e., such that at least one observation falls in A_j), for any $x \in A_j$, the vectors $(\hat{w}_i(x))_{i \in \{1,\ldots,n\}}$ have weights equal to $1/n_{A_j}$ for $i \in A_j$, where n_{A_j} is the number of training points in set A_j, and 0 otherwise.

Equivalence with least-squares regression. When applied to regression where the estimator is $\hat{f}(x) = \sum_{i=1}^n \hat{w}_i(x) y_i$, 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 $(x_1, y_1), \ldots, (x_n, y_n)$, 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 multidimensional normal equation

$$\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top \theta = \sum_{i=1}^n (y_i - \bar{y})\varphi(x_i).$$

It turns out that matrix $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n \varphi(x_i)\varphi(x_i)^\top$ is diagonal where for each $j \in J$, $n\widehat{\Sigma}_{jj}$ is equal to the number n_{A_j} of data points lying in cell A_j. This implies that for a nonempty cell A_j, θ_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 equation (6.2). For empty cells, θ_j is not determined by the normal equation given 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 (whether empirical or not) noncentered covariance matrix makes it attractive for theoretical purposes: as shown in section 6.3.1, we can essentially import results from chapter 3; moreover, partitioning estimators provide particularly simple examples of least-squares estimator since the inversion of the population and expected covariance matrices can 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 $|J| = h^{-d}$, as illustrated in dimension $d = 2$ with $|J| = 25$:

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}

  Note here that the computation time for each $x \in \mathcal{X}$ is not necessarily proportional to $|J|$ but rather to n (by simply considering the bins where the data lie). This estimator is sometimes called a "regressogram." We need then to choose bandwidth h (see analysis in section 6.3.1). See figure 6.2 for an illustration in one dimension.

- **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 the number of cells in the partition (see, e.g., section 9.2 from Friedman et al., 2009). Note here that the partition depends on the labels (so the analysis given here does not apply unless the partitioning is learned on data different from the one used for the estimation).

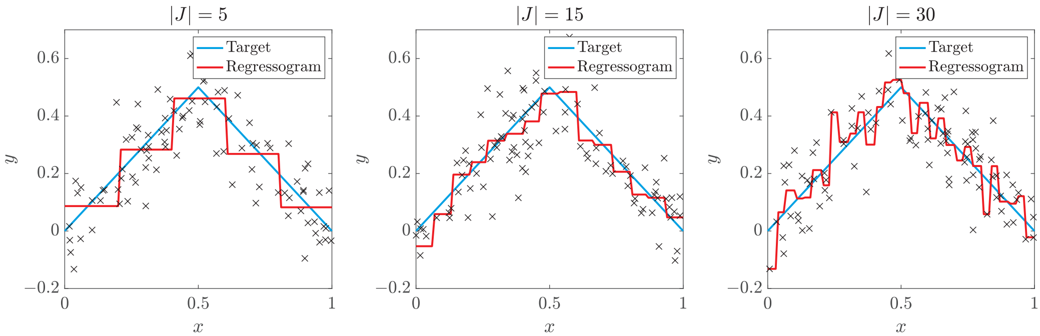


Figure 6.2. Regressograms in dimension $d = 1$, with three values of $|J|$ (the number of sets in the partition). Here, $n = 100$ input data points are distributed uniformly on $[0, 1]$, and, for $i \in \{1, \dots, n\}$, the outputs y_i equal $\frac{1}{2} - |x_i - \frac{1}{2}| + \varepsilon_i$, where ε_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 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).

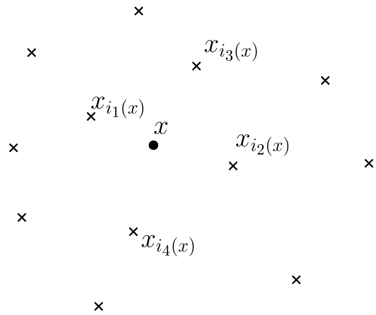
6.2.3 Nearest-Neighbors

Given an integer $k \geqslant 1$, and a distance Δ on $\mathcal{X}$, for any $x \in \mathcal{X}$, we can order the n observations so that

$$\Delta(x_{i_1(x)}, x) \leqslant \Delta(x_{i_2(x)}, x) \leqslant \cdots \leqslant \Delta(x_{i_n(x)}, x),$$

where $\{i_1(x), \dots, i_n(x)\} = \{1, \dots, n\}$ and ties are broken randomly[2] (i.e., for all $x \in \mathcal{X}$, the indices that come first are sampled randomly). See the illustration below:

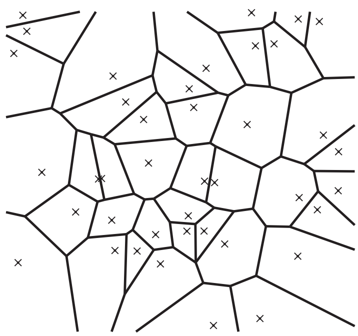
[1]See more details in https://en.wikipedia.org/wiki/Decision_tree_learning.
[2]Other conventions share the weights among all ties.



We then define

$$\hat{w}_i(x) = 1/k \ \text{ if } \ i \in \{i_1(x), \dots, i_k(x)\}, \text{ and 0 otherwise.}$$

Given a new input $x \in \mathbb{R}^d$, the nearest neighbor predictor looks at the k nearest points x_i in the dataset $\{(x_1, y_1), \dots, (x_n, y_n)\}$ 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. A one-dimensional example is illustrated 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, with all regions displayed[3]:



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 terms of both time and memory. Indexing techniques exist for (potentially approximate) nearest-neighbor search, such as "k-d trees,"[4] with typically a logarithmic complexity in n (but with some additional compiling time), and a memory footprint that can grow exponentially in dimension (see, e.g., Shakhnarovich et al., 2005).

Exercise 6.1 *For k-nearest-neighbors and partitioning estimates, what is the pattern of nonzeros in the smoothing matrix $H \in \mathbb{R}^{n \times n}$?*

[3] See more details about Voronoi diagrams in `https://en.wikipedia.org/wiki/Voronoi_diagram`.

[4] See `https://en.wikipedia.org/wiki/K-d_tree`.

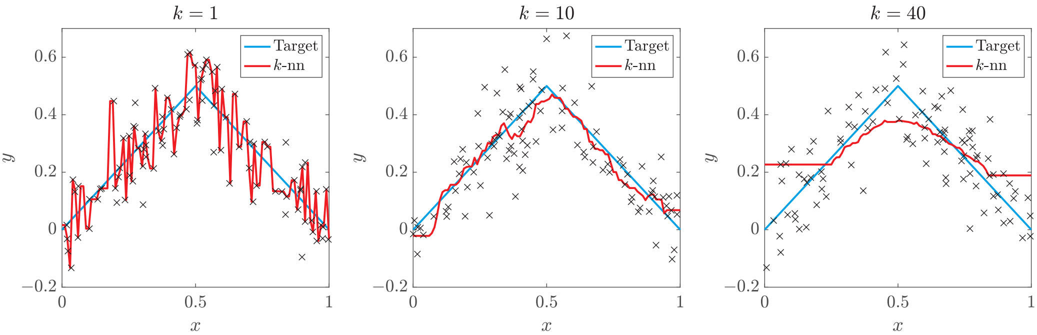


Figure 6.3. k-nearest neighbor regression in dimension $d = 1$, with three values of k (the number of neighbors), with the same data as figure 6.2. We can observe both underfitting (k too large) and overfitting (k too small).

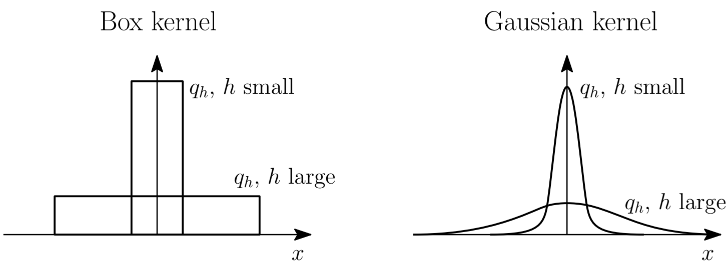
6.2.4 Nadaraya-Watson Estimator (aka Kernel Regression) (♦)

Given a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, which is pointwise nonnegative, we define

$$\hat{w}_i(x) = \frac{k(x, x_i)}{\sum_{i'=1}^n k(x, x_{i'})},$$

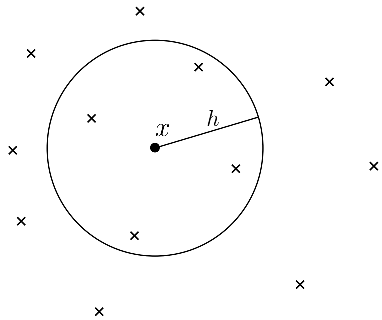
with the convention that if $k(x, x_i) = 0$ for all $i \in \{1, \dots, n\}$, then $\hat{w}_i(x)$ is equal to $1/n$ for each i (which is the same convention used for estimators based on partitions in section 6.2.2).

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, with $h > 0$ as a bandwidth parameter to be selected (see the analysis in section 6.3.3). If we assume that q is integrable with an integral equal to 1, then $k(\cdot, x')$ is a probability density with mass around x', which gets more concentrated as h goes to zero. See the following illustration 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 function $\hat{w}_i$ with many zeros. See the following for an illustration of this point in dimension $d = 2$:



- Gaussian kernel $q(x) \propto e^{-\|x\|_2^2/2}$, where we use the fact that it is nonnegative *pointwise*, as opposed to positive-definiteness (discussed in chapter 7).[5] 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 (i.e., with a complexity proportional to n). The same techniques used for an 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).

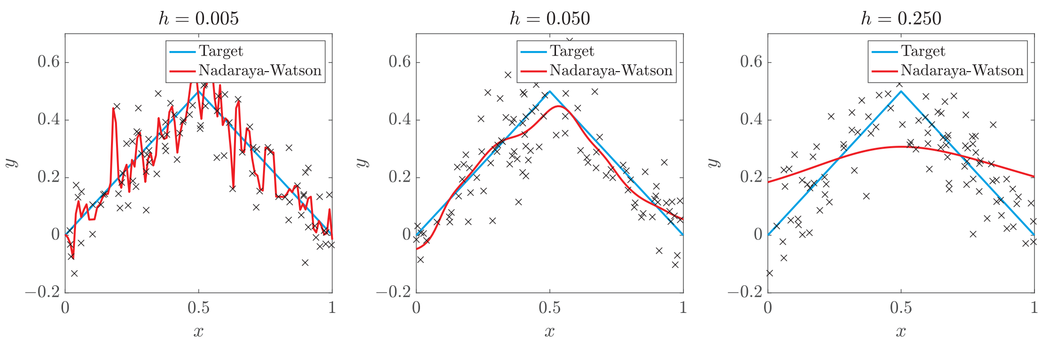


Figure 6.4. Nadaraya-Watson regression in dimension $d = 1$, with three values of h (the bandwidth), for the Gaussian kernel, with the same data as figure 6.2. We can observe both underfitting (h too large), and overfitting (h too small).

6.3 Generic Simplest Consistency Analysis

For simplicity, here we only consider 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 a square root calibration function on top of the least-squares excess risk; see exercise 6.2). Still, better rates can be obtained directly (see, e.g., Audibert and Tsybakov, 2007; Chaudhuri and Dasgupta, 2014).

We make the following generic simplifying assumptions (weaker ones could be considered with more involved proofs):

[5] See also https://francisbach.com/cursed-kernels/.

(H-1) Bounded noise: There is $\sigma \geqslant 0$ 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 σ^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 Δ. 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 the fact that the weights $\hat{w}_i(x)$ sum to 1),

$$
\begin{aligned}
\hat{f}(x) - f_*(x) &= \sum_{i=1}^n y_i \hat{w}_i(x) - \mathbb{E}[y|x] \\
&= \sum_{i=1}^n \hat{w}_i(x)\big[y_i - \mathbb{E}[y_i|x_i]\big] + \sum_{i=1}^n \hat{w}_i(x)\big[\mathbb{E}[y_i|x_i] - \mathbb{E}[y|x]\big] \\
&= \sum_{i=1}^n \hat{w}_i(x)\big[y_i - \mathbb{E}[y_i|x_i]\big] + \sum_{i=1}^n \hat{w}_i(x)\big[f_*(x_i) - f_*(x)\big].
\end{aligned}
$$

Given $x_1, \dots, x_n$ (and *because we have assumed that 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 being fixed (taking expectations uniquely with respect to labels $y_1, \dots, y_n$):

$$
\begin{aligned}
& \mathbb{E}\big[(\hat{f}(x) - f_*(x))^2 \big| x_1, \dots, x_n\big] \\
= \ & \big(\mathbb{E}[\hat{f}(x)\big|x_1, \dots, x_n] - f_*(x)\big)^2 + \operatorname{var}\big[\hat{f}(x)\big|x_1, \dots, x_n\big] \\
= \ & \Big[\sum_{i=1}^n \hat{w}_i(x)\big(f_*(x_i) - f_*(x)\big)\Big]^2 + \sum_{i=1}^n \hat{w}_i(x)^2 \mathbb{E}\Big[\big(y_i - \mathbb{E}[y_i|x_i]\big)^2\Big|x_i\Big] \\
= \ & \qquad\qquad \text{bias} \qquad\qquad + \qquad\qquad \text{variance},
\end{aligned}
$$

with a "bias" term that is zero if f_* is constant,[6] and a "variance" term that 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

$$
\begin{aligned}
& \mathbb{E}\big[(\hat{f}(x) - f_*(x))^2 \big| x_1, \dots, x_n\big] \\
\leqslant \ & \Big[\sum_{i=1}^n \hat{w}_i(x)\big|f_*(x_i) - f_*(x)\big|\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H-1),} \qquad (6.3) \\
\leqslant \ & \Big[\sum_{i=1}^n \hat{w}_i(x) B\Delta(x_i, x)\Big]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H-2),} \\
\leqslant \ & B^2 \sum_{i=1}^n \hat{w}_i(x)\Delta(x_i, x)^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using Jensen's inequality.} \qquad (6.4)
\end{aligned}
$$

[6] What we call "bias" in this book is sometimes referred to as the "squared bias."

Note that in the last inequality in equation (6.4), having the weight vector $\hat{w}(x)$ in the simplex is crucial. We then have for the expected excess risk this generic bound, which we will use for all three cases (partitions, k-nearest-neighbor, and Nadaraya-Watson):

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x)-f_*(x))^2]dp(x) \leqslant B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i,x)^2\Big]dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2]dp(x). \tag{6.5}$$

⚠ 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 $\sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2]dp(x)$, which depends on the noise on top of the optimal predictions. Since the weights sum to 1, we can write $\sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] = \sum_{i=1}^n \mathbb{E}[(\hat{w}_i(x)-1/n)^2]+1/n$; that is, up to a vanishing constant, the variance term measures the deviation from uniform weights.
- A bias term $B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i,x)^2\Big]dp(x)$, 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 their mass on observations x_i that are close to x.

This leads to two conditions: *both* variance and bias have to go to zero when n grows, corresponding to two explicit 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 (which is usually different for different testing points), 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 to $\mathcal{X}$ a subset of $\mathbb{R}^d$, with a distribution with a density with some minor regularity properties (all will have compact support, i.e., $\mathcal{X}$ is 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 relax 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*

$$\Big(B^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i,x)^2\Big]dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2]dp(x)\Big)^{1/2}.$$

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 a bounded noise (assumption (H-1)), a Lipschitz-continuous target function (assumption (H-2)),*

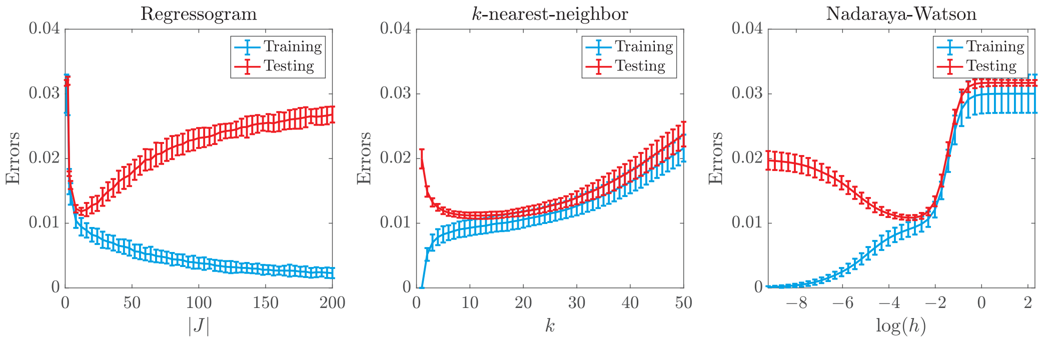


Figure 6.5. Learning curves for all three local averaging methods as a function of the corresponding hyperparameter. Left: regressogram (hyperparameter = number $|J|$ of sets in the partition); middle: k-nearest-neighbor (hyperparameter = number of neighbors k); right: Nadaraya-Watson (hyperparameter = bandwidth h). In all three cases, we see a trade-off between underfitting and overfitting.

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

$$\int_{\mathcal{X}} \mathbb{E}\big[(\hat{f}(x) - f_*(x))^2\big]\,dp(x) \leqslant \Big(8\sigma^2 + \frac{B^2}{2}\mathrm{diam}(\mathcal{X})^2\Big)\frac{|J|}{n} + B^2 \max_{j\in J} \mathrm{diam}(A_j)^2. \qquad (6.6)$$

Optimal trade-off between bias and variance. Before we look at the proof (which is based on equation (6.5)), we can look at the consequence of the bound in equation (6.6). We need to balance the terms (up to constants) $\max_{j\in J}\mathrm{diam}(A_j)^2$ and $\frac{|J|}{n}$. In the simplest 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 $\max_{j\in J}\mathrm{diam}(A_j)^2 = h^2$, 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 also 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 use the smoothness of the target function to get significantly improved bounds (local averaging cannot take strong advantage of such smoothness). In chapters 8 and 9, we will use dependence on a small number of variables.

Experiments. For the problem shown in section 6.2, we plot in figure 6.5 (left) 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 nonempty 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 equation (6.5), the variance term is bounded from above by σ^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 $n \times \frac{1}{n^2} = \frac{1}{n}$ 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 from section 3.4, as this is essentially equivalent to least-squares regression with $|J|$ features $(1_{x \in A_j})_{j \in J}$. We now make this precise.

We consider the decomposition of the variance term

$$\begin{aligned}\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^{n} \hat{w}_i(x)^2\Big] dp(x) &\leqslant \sum_{j \in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{1_{1 \leqslant n_{A_j} \leqslant \frac{n}{2}\mathbb{P}(A_j)}}{n_{A_j}} + \frac{1_{n_{A_j} > \frac{n}{2}\mathbb{P}(A_j)}}{n_{A_j}} + \frac{1_{n_{A_j}=0}}{n}\Big] \qquad (6.7)\\ &\leqslant \sum_{j \in J} \mathbb{P}(A_j) \Big[\mathbb{P}\Big(\frac{n_{A_j}}{n} \leqslant \frac{\mathbb{P}(A_j)}{2}\Big) + \frac{2}{n\mathbb{P}(A_j)} + \frac{1}{n}\mathbb{P}(n_{A_j}=0)\Big].\end{aligned}$$

We can then estimate the required probabilities: $\mathbb{P}(n_{A_j} = 0) = (1 - \mathbb{P}(A_j))^n$, and, using Bernstein's inequality (single-sided version of equation (1.13) in 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 get:

$$\begin{aligned}\mathbb{P}\Big(\frac{n_{A_j}}{n} \leqslant \frac{1}{2}\mathbb{P}(A_j)\Big) &= \mathbb{P}\Big(\frac{n_{A_j}}{n} \leqslant \mathbb{P}(A_j) - \frac{1}{2}\mathbb{P}(A_j)\Big) \qquad (6.8)\\ &\leqslant \exp\Big(-\frac{n\mathbb{P}(A_j)^2/4}{2\mathbb{P}(A_j) + 2(\mathbb{P}(A_j)/2)/3}\Big) \leqslant \exp(-n\mathbb{P}(A_j)/10) \leqslant \frac{5}{n\mathbb{P}(A_j)},\end{aligned}$$

where we have used $\alpha e^{-\alpha} \leqslant 1/2$ for any $\alpha \geqslant 0$. This leads to the bound

$$\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^{n} \hat{w}_i(x)^2\Big] dp(x) \leqslant \sum_{j \in J} \mathbb{P}(A_j)\, \mathbb{E}\Big[\frac{5}{n\mathbb{P}(A_j)} + \frac{2}{n\mathbb{P}(A_j)} + \frac{1}{n\mathbb{P}(A_j)}\Big] \leqslant \frac{8|J|}{n}.$$

Bias. We have, for $x \in A_j$ and a nonempty cell,

$$\sum_{i=1}^n \hat{w}_i(x)\Delta(x,x_i)^2 \leqslant \mathrm{diam}(A_j)^2,$$

with $\sum_{i=1}^n \hat{w}_i(x)\Delta(x,x_i)^2 = \frac{1}{n}\sum_{i=1}^n \Delta(x,x_i)^2 \leqslant \mathrm{diam}(\mathcal{X})^2$ for empty cells. Thus, separating the cases $n_{A_j}=0$ and $n_{A_j}>0$:

$$\begin{aligned}\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x,x_i)^2\Big] dp(x) &\leqslant \sum_{j\in J} \mathbb{P}(A_j)\,\mathbb{E}\Big[\mathrm{diam}(A_j)^2 1_{n_{A_j}>0} + \mathrm{diam}(\mathcal{X})^2 1_{n_{A_j}=0}\Big]\\ &\leqslant \sum_{j\in J} \mathbb{P}(A_j)\Big[\mathrm{diam}(A_j)^2 + (1-\mathbb{P}(A_j))^n \mathrm{diam}(\mathcal{X})^2\Big]\\ &= \sum_{j\in J}\mathbb{P}(A_j)\mathrm{diam}(A_j)^2 + \sum_{j\in J}\mathbb{P}(A_j)(1-\mathbb{P}(A_j))^n\ \mathrm{diam}(\mathcal{X})^2.\end{aligned}$$

Using that $u(1-u)^n \leqslant ue^{-nu} \leqslant 1/(2n)$ for $u\in[0,1]$, we get

$$\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x,x_i)^2\Big] dp(x) \leqslant \sum_{j\in J}\mathbb{P}(A_j)\mathrm{diam}(A_j)^2 + \frac{1}{2}\frac{|J|}{n}\times \mathrm{diam}(\mathcal{X})^2,$$

which leads to the desired term. ■

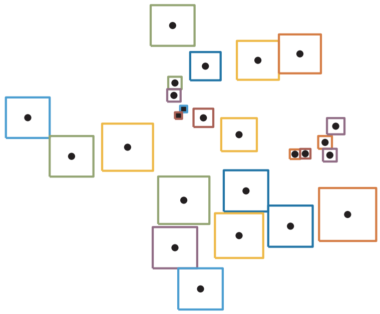
6.3.2 k-nearest Neighbor

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 $\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i,x)^2$ 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 kth-nearest neighbor $x_{i_k(x)}$, for which lemmas 6.1 and 6.2, taken from theorem 2.4 by Biau and Devroye (2015), give an estimate for the ℓ_∞-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 $\mathcal{X} \subset \mathbb{R}^d$, and $n+1$ points $x_1,\dots,x_n,x_{n+1}$ sampled i.i.d. from $\mathcal{X}$. Then the expected squared ℓ_∞-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, for simplicity, assume $R_i>0$ 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 following illustration in two dimensions, with squares representing sets B_i centered at x_i (represented as dots):



Moreover, their union has a 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 $\mathcal{X} \subset \mathbb{R}^d$, and $n+1$ points $x_1, \dots, x_n, x_{n+1}$ sampled i.i.d. from $\mathcal{X}$. Then the expected squared ℓ_∞-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 that $2k \leqslant n$ (otherwise, the proposed bounds are trivial). We can then divide randomly (and independently) the n first points into $2k$ sets of a size of approximately $\frac{n}{2k}$. We denote $x_{(k)}^j$ a 1-nearest neighbor of x_{n+1} within the jth set. The squared distance from x_{n+1} to the k-nearest neighbor among all first n points is less than the kth 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 kth smallest distance is less than $\frac{1}{k}\sum_{j=1}^{2k}\|x_{n+1} - x_{(k)}^j\|_\infty^2$ (it is a general fact that the k-smallest element among nonnegative p elements is less than their sum divided by $p - k$, applied here for $p = 2k$).

Thus, using lemma 6.1 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 $d = 1$ (proof left as an exercise). ■

Putting things together, we get the following result for the consistency of k-nearest neighbor regression.

Proposition 6.2 (Convergence rate for k-nearest neighbors) *Assume a bounded noise (assumption (H-1)), a Lipschitz-continuous target function (assumption (H-2)), and an input distribution with bounded support $\mathcal{X}$. Then for the k-nearest-neighbor estimate $\hat{f}$ with the ℓ_∞-norm, we have, for $d \geqslant 2$,*

$$\int_{\mathcal{X}} \mathbb{E}\big[(\hat{f}(x) - f_*(x))^2\big] dp(x) \leqslant \frac{\sigma^2}{k} + 8B^2 \mathrm{diam}(\mathcal{X})^2 \Big(\frac{2k}{n}\Big)^{2/d}. \tag{6.9}$$

Balancing the two terms in equation (6.9) 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 in Chen and Shah (2018) and Biau and Devroye (2015).

Exercise 6.3 *Show that if the Bayes rate is* 0 *(i.e.,* $\sigma = 0$*), then the* 1*-nearest-neighbor predictor is consistent.*

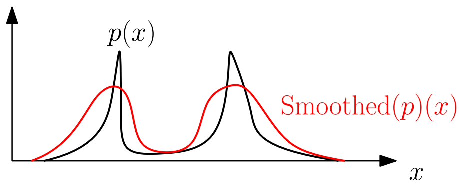
Experiments. For the problem shown in section 6.2, we plot in figure 6.5 (middle) training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of k.

6.3.3 Kernel Regression (Nadaraya-Watson) (♦)

In this section, we assume that $\mathcal{X} = \mathbb{R}^d$, and for simplicity, that the distribution of the inputs has a density (also denoted as 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 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 as

$$\hat{w}_i(x) = \frac{q_h(x - x_i)}{\sum_{j=1}^n q_h(x - x_j)}.$$

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 to $\displaystyle\int_{\mathbb{R}^d} q_h(x - z) g(z) p(z) dz$ almost surely, which is exactly the convolution between function q_h and function pg, which we can denote as $[(pg) * q_h](x)$. 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 function pg by averaging values at range h. Therefore, when h goes to zero, it converges to the function pg itself. See the following example for $g = 1$:



We can now look at the generalization bound from equation (6.5), and see how it applies to kernel regression. We now consider the ℓ_2-distance for simplicity and 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 with a formal result.

Variance term. We have, for 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 previously discussed, this sum $n\sum_{i=1}^n \hat{w}_i(x)^2$ is converging almost surely to

$$\frac{\int_{\mathbb{R}^d} q_h(x-z)^2 p(z)dz}{\big(\int_{\mathbb{R}^d} q_h(x-z)p(z)dz\big)^2} = \frac{[q_h^2 * p](x)}{[q_h * p](x)^2}. \tag{6.10}$$

When h tends to zero, then the denominator $[q_h * p](x)^2$ in equation (6.10) tends to $p(x)^2$ because the bandwidth of the smoothing goes to zero. The numerator in equation (6.10) 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 it 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,

$$\int_{\mathcal{X}} \Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] p(x)dx \sim \frac{1}{nh^d}\mathrm{vol}(\mathrm{supp}(p))\int_{\mathbb{R}^d} q(u)^2 du,$$

where $\mathrm{vol}(\mathrm{supp}(p))$ 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 to be bounded.

Bias. With the same intuitive reasoning, we get when n tends to infinity (for Δ the ℓ_2-norm distance):

$$\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i, x)^2 \quad \to \quad \frac{\int_{\mathbb{R}^d} q_h(x-z)\|x-z\|_2^2\, p(z)dz}{\int_{\mathbb{R}^d} q_h(x-z)p(z)dz}.$$

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

$$\int_{\mathcal{X}} \Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i,x)^2\Big]p(x)dx \sim h^2 \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.$$

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 this informal reasoning more formal using concentration inequalities, leading to nonasymptotic bounds of the same nature (simply more complicated) that make explicit the joint dependence on n and h. We will prove the result given in proposition 6.3.

Proposition 6.3 (Convergence rate for Nadaraya-Watson estimation) *Assume a bounded noise (assumption (H-1)), a Lipschitz-continuous target function (assumption (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*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x)-f_*(x))^2]dp(x) \leqslant \Big[\frac{8\|q\|_\infty}{nh^d}\Big(\sigma^2 + \frac{B}{2}\mathrm{diam}(\mathcal{X})^2\Big) + 2Bh^2\|p\|_\infty c\Big]\cdot C_h, \tag{6.11}$$

where $c = \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du$ 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 6.4). Before giving the proof for this proposition, 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.

As opposed to positive-definite kernel methods in chapter 7, for local averaging, consistency is only achieved if the bandwidth tends to zero when the number n of observations tends to infinity.

Proof of proposition 6.3 (♦♦) As for the proof for partitioning estimates (equation (6.7)), to deal with the denominator in the definition of the weights, we can first use Bernstein's inequality (single-sided version of equation (1.13) in 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 $\mathcal{A}(x) = \big\{\frac{1}{n}\sum_{i=1}^n q_h(x-x_i) \leqslant \frac{1}{2}\mathbb{E}[q_h(x-z)]\big\}$ (which corresponds to equation (6.8) in the proof of proposition 6.1),

$$\begin{aligned}\mathbb{P}\big(\mathcal{A}(x)\big) &\leqslant \exp\Big(\frac{-\frac{n}{4}(\mathbb{E}[q_h(x-z)])^2}{2\mathbb{E}[q_h^2(x-z)] + \mathbb{E}[q_h(x-z)]h^{-d}\|q\|_\infty/3}\Big) \\ &\leqslant \exp\Big(\frac{-\frac{n}{4}\mathbb{E}[q_h(x-z)]}{(7/3)h^{-d}\|q\|_\infty}\Big) \leqslant \frac{\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}\cdot\frac{1}{e}\frac{28}{3} \leqslant \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}, \quad (6.12)\end{aligned}$$

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, since $\hat{w}_i(x) = q\big(\frac{1}{h}(x-x_i)\big)/\sum_{j=1}^n q\big(\frac{1}{h}(x-x_j)\big)$,

$$\begin{aligned}\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big] &= \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)^2\Big] \\ &\leqslant \mathbb{P}(\mathcal{A}(x)) + \frac{4}{\big(n\mathbb{E}[q_h(x-z)]\big)^2}\mathbb{E}\Big[\sum_{i=1}^n q\Big(\frac{1}{h}(x-x_i)\Big)^2\Big] \\ &\leqslant \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]} + \frac{4\mathbb{E}\big[q_h(x-z)^2\big]}{n\big[\mathbb{E}q_h(x-z)\big]^2} \leqslant \frac{8\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}.\end{aligned}$$

Moreover, we have $\mathbb{E}[q_h(x-z)] = \int_{\mathbb{R}^d} p(x-hu)q(u)du = [p*q_h](x)$. This leads to an overall bound on the variance term as $\sigma^2\int_{\mathcal{X}}\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)^2\Big]p(x)dx \leqslant \frac{8\|q\|_\infty\sigma^2}{nh^d}\int_{\mathcal{X}}\frac{p(x)}{[p*q_h](x)}dx.$

Bias. We have a similar reasoning for the bias term. Indeed, we get for a given $x \in \mathcal{X}$, using the bound in equation (6.12),

$$\begin{aligned}&\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] \\ = \;& \mathbb{E}\Big[1_{\mathcal{A}(x)}\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] + \mathbb{E}\Big[1_{\mathcal{A}(x)^c}\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big] \\ \leqslant \;& \mathbb{P}(\mathcal{A}(x))\cdot\mathrm{diam}(\mathcal{X})^2 + \frac{2}{n\mathbb{E}[q_h(x-z)]}\cdot n\mathbb{E}[q_h(x-z)\|x-z\|_2^2] \\ \leqslant \;& \frac{4\|q\|_\infty}{nh^d[q_h*p](x)}\cdot\mathrm{diam}(\mathcal{X})^2 + \frac{2h^2}{[q_h*p](x)}\cdot\int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du \\ \leqslant \;& \frac{4\|q\|_\infty}{nh^d[q_h*p](x)}\cdot\mathrm{diam}(\mathcal{X})^2 + \frac{2h^2\|p\|_\infty}{[q_h*p](x)}\cdot\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.\end{aligned}$$

This leads to an overall bound on the bias term equal to $B^2\int_{\mathcal{X}}\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\Big]p(x)dx$, which is less than $B^2\int_{\mathcal{X}}\frac{p(x)}{[q_h*p](x)}dx\cdot\Big[\frac{4\|q\|_\infty}{nh^d}\mathrm{diam}(\mathcal{X})^2 + 2h^2\|p\|_\infty\Big(\int_{\mathbb{R}^d} q(u)\|u\|_2^2du\Big)\Big].$

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

$$\Big[\frac{8\|q\|_\infty}{nh^d}\big(\sigma^2 + \frac{B}{2}\mathrm{diam}(\mathcal{X})^2\big) + 2Bh^2\|p\|_\infty\Big(\int_{\mathbb{R}^d} q(u)\|u\|_2^2 du\Big)\Big] \cdot \int_{\mathcal{X}} \frac{p(x)}{[q_h * p](x)} dx,$$

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 $C_h = \int_{\mathcal{X}} \frac{p(x)}{[q_h * p](x)} dx \leqslant 2\mathrm{vol}(\mathcal{X})$.*

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 (♦)

Earlier in this chapter, we have required the following conditions on the weights:

- $\displaystyle\int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x_i, x)^2\Big] dp(x) \to 0$ when n tends to infinity, to ensure that the bias goes to zero.
- $\displaystyle\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] dp(x) \to 0$ 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 (i.e., consistency for any square-integrable functions), we need an extra technical assumption, which was first outlined by Stone (1977): there is $c > 0$ such that for any nonnegative integrable function $h : \mathcal{X} \to \mathbb{R}_+$,

$$\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E}\big[\hat{w}_i(x) h(x_i)\big] dp(x) \leqslant c \cdot \int_{\mathcal{X}} h(x) dp(x). \tag{6.13}$$

⚠ In this discussion, as in the rest of this chapter, 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.

If equation (6.13) is satisfied, for any $\varepsilon > 0$, and for any target function $f_* \in L_2(p)$, we can find function g, which is $B(\varepsilon)$-Lipschitz-continuous and such that $\|f_* - g\|_{L_2(p)} \leqslant \varepsilon$, because the set of Lipschitz-continuous functions is dense in $L_2(p)$ (see, e.g., Ambrosio et al., 2013).

Then we have, for a given $x \in \mathcal{X}$,

$$\begin{aligned}
& \mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\big[f_*(x_i) - f_*(x)\big]\Big)^2\Big] \\
\leqslant\ & \mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\Big(\big|f_*(x_i) - g(x_i)\big| + \big|g(x_i) - g(x)\big| + \big|g(x) - f_*(x)\big|\Big)^2\Big] \\
\leqslant\ & 3\mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\big|f_*(x_i) - g(x_i)\big|\Big)^2\Big] + 3\mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\big|g(x_i) - g(x)\big|\Big)^2\Big] \\
& +3\mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\big|g(x) - f_*(x)\big|\Big)^2\Big] \text{ using the inequality } (a+b+c)^2 \leqslant 3a^2+3b^2+3c^2, \\
\leqslant\ & 3\mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\big|f_*(x_i) - g(x_i)\big|\Big)^2\Big] + 3\mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)B(\varepsilon)\Delta(x, x_i)\Big)^2\Big] \\
& +3\mathbb{E}\Big[\big|g(x) - f_*(x)\big|^2\Big] \text{ since weights sum to 1, and } g \text{ is Lipschitz-continuous.}
\end{aligned}$$

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

$$\begin{aligned}
& 3\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\big|f_*(x_i) - g(x_i)\big|^2\Big] + 3B(\varepsilon)^2\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x, x_i)^2\Big] \\
& +3\mathbb{E}\Big[\big|g(x) - f_*(x)\big|^2\Big] \text{ using Jensen's inequality on the second term,} \\
\leqslant\ & 3c \cdot \mathbb{E}\big[\big|f_*(x) - g(x)\big|^2\big] + 3B(\varepsilon)^2\mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x, x_i)^2\Big] + 3\mathbb{E}\Big[\big|g(x) - f_*(x)\big|^2\Big],
\end{aligned}$$

using equation (6.13). We can now integrate with respect to x to get

$$\begin{aligned}
& \int_{\mathcal{X}} \mathbb{E}\Big[\Big(\sum_{i=1}^n \hat{w}_i(x)\big[f_*(x_i) - f_*(x)\big]\Big)^2\Big] dp(x) \\
& \qquad \leqslant 3c \cdot \varepsilon^2 + 3B(\varepsilon)^2 \int_{\mathcal{X}} \mathbb{E}\Big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x, x_i)^2\Big] dp(x) + 3\varepsilon^2. \quad (6.14)
\end{aligned}$$

Proving universal consistency. We can then combine the bound in equation (6.14) (which gives a bound on the bias) with equation (6.3), 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 equation (6.14), we can use the same tools for consistency as for equation (6.5).

To prove universal consistency, we fix a certain $\varepsilon > 0$, from which we obtain a Lipschitz constant $B(\varepsilon)$. For such $B(\varepsilon)$, we know how to make the (squared) bias term $B(\varepsilon)^2 \int_{\mathcal{X}} \mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x)\Delta(x,x_i)^2\big]dp(x) + \sigma^2 \int_{\mathcal{X}} \mathbb{E}\big[\sum_{i=1}^n \hat{w}_i(x)^2\big]dp(x)$ less than ε, by choosing an appropriate hyperparameter and a number of observations n (see previous sections). Thus, if the extra condition in equation (6.13) is satisfied, these three methods are universally consistent. Note that, in general, n has to grow unbounded when ε tends to zero without any a priori bound (since $B(\varepsilon)$ cannot be bounded without assumptions on the target function).

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 sections 6.2.2 and 6.3.1, we have for any fixed $x \in A_j$, $j \in J$, and f a nonnegative function:

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E}\big[\hat{w}_i(x) f(x_i)\big] &= \mathbb{E}\bigg[1_{n_{A_j}>0}\frac{1}{n_{A_j}} \sum_{i \text{ s.t. } x_i \in A_j} f(x_i) \ + 1_{n_{A_j}=0}\frac{1}{n}\sum_{i=1}^n f(x_i)\bigg] \\
&= \mathbb{E}\bigg[\sum_{i=1}^n \frac{1_{x_i\in A_j} f(x_i)}{1+\sum_{i'\neq i} 1_{x_{i'}\in A_j}} + 1_{n_{A_j}=0}\frac{1}{n}\sum_{i=1}^n f(x_i)\bigg] \\
&\leqslant \sum_{i=1}^n \mathbb{E}\big[1_{x_i\in A_j} f(x_i)\big]\cdot \mathbb{E}\bigg[\frac{1}{1+\sum_{i'\neq i} 1_{x_{i'}\in A_j}}\bigg] + \mathbb{E}[f(z)] \\
&\qquad\qquad \text{by independence of } x_1,\dots,x_n, \\
&\leqslant n\mathbb{E}\big[1_{x_i\in A_j} f(x_i)\big]\cdot\frac{1}{n\mathbb{P}(A_j)} + \mathbb{E}[f(z)] \text{ using exercise 6.5}, \\
&= \mathbb{E}[f(z)|z\in A_j] \ + \mathbb{E}[f(z)],
\end{aligned}$$

  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 equation (6.13) with $c = 2$.

  Exercise 6.5 *If $Z_1,\dots,Z_m$ are i.i.d. Bernoulli random variables with parameter $\rho \in (0,1]$. Show that $\mathbb{E}\big[\frac{1}{1+Z_1+\cdots+Z_m}\big] \leqslant \frac{1}{(m+1)\rho}$.*

- 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 equation (6.13) is not easy to show and is often referred to as "Stone's lemma." See lemma 10.7 from Biau and Devroye (2015).

6.5 Adaptivity (♦♦)

As shown earlier in this chapter, all local averaging techniques achieve the same performance on Lipschitz-continuous functions, which is an unavoidable bad performance when d grows (the curse of dimensionality). One extra order of smoothness (i.e., on $\mathbb{R}^d$, two bounded derivatives) can be used to lead to a convergence rate proportional to $n^{-4/(4+d)}$ (see section 5.4 in Wasserman, 2006). However, the higher smoothness of the target function does not seem to be easy to use; 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 used in the proofs; when f_* is twice-differentiable, by a Taylor expansion, $f_*(x_i) - f_*(x) = (x_i - x)^\top (f_*)'(x) + O(\|x_i - x\|^2)$, and we can choose weights such 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 6.6); 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 possible only 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 nonnegative 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 take advantage of smoothness. Among local averaging techniques, however, there are ways to do it. For example, using locally linear regression, where one solves for any test point x, the following local least-squares regression problem with an affine function:

$$\inf_{\beta_1 \in \mathbb{R}^d,\ \beta_0 \in \mathbb{R}} \sum_{i=1}^n \hat{w}_i(x)(y_i - \beta_1^\top x_i - \beta_0)^2.$$

(note that the regular regressogram corresponds to setting $\beta_1 = 0$). 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, improves over plain local averaging (Tsybakov, 2008). See an example with the regressogram weights in figure 6.6. In order to be adaptive to higher degrees of smoothness, local polynomial regression can be used at an increased computational cost (see, e.g., Fan et al., 1997, and references therein).

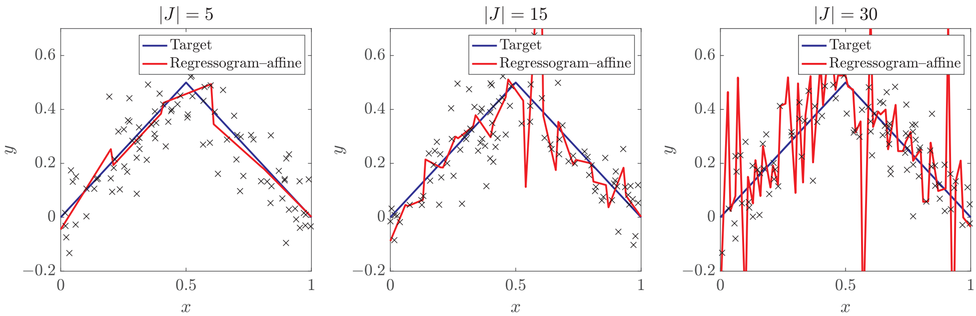


Figure 6.6. Locally affine regression based on the regressogram, on the same data as figure 6.2, for three values of the number $|J|$ of sets within in the partition. Notice the difference between this and figure 6.2.

Exercise 6.6 (♦) *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 employ the explicit formula for the Bayes predictor and explicitly aim at approximating it without the need for optimization (as opposed to all the other methods presented in this book). While they can potentially adapt to complex prediction functions, they suffer from the curse of dimensionality (i.e., the number of observations has to be exponential in dimension to create 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 being only a linear projection of the inputs (this will be possible only for neural networks). A key feature of these methods is that they will *not* look at local interactions with characteristic distance tending to zero when the number of observations goes to infinity (as local averaging does to reach statistical consistency).

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 an amount of time that depends on the number of observations, using a positive-definite kernel function.
- Kernels on $\mathbb{R}^d$: Such models include polynomials and classical smooth Sobolev spaces (functions with square-integrable partial derivatives of order greater than $d/2$).
- 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 situation, 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 $f_\theta : \mathcal{X} \to \mathbb{R}$ that are *linear in their parameters* θ (i.e., 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 doing so is not ambiguous.

The key differences between this chapter and chapter 3 on least-squares estimation are 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 allow the handling of very expressive models, embedded in the framework of linear models.
- 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 from chapter 6). 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 now set by neural networks presented in chapter 9). They can also be applied to many numerical analysis tasks (Schaback and Wendland, 2006).
- 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) and overparameterized models (see chapter 12).

The type of kernel considered here is different from the ones in chapter 6. The ones here are *positive-definite*, while the ones from chapter 6 are *nonnegative*. See more details in `https://francisbach.com/cursed-kernels/`.

[1] More precisely, this is a vector space that is endowed with an inner product and is complete for the associated normed space topology. See `https://en.wikipedia.org/wiki/Hilbert_space` for more details.

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.

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 that the loss function ℓ 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 section 4.1). Here, $\varphi : \mathcal{X} \to \mathcal{H}$ is the feature map, and dot products and norms are taken with respect to the Hilbertian structure of $\mathcal{H}$

The key property of the objective function in equation (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 in proposition 7.1.

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*

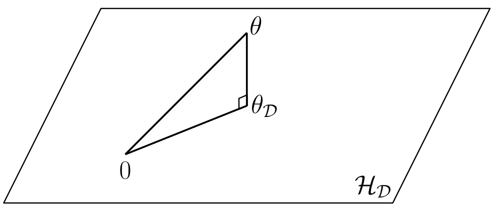
$$\Psi(\langle \theta, \varphi(x_1) \rangle, \dots, \langle \theta, \varphi(x_n) \rangle, \|\theta\|^2)$$

can be obtained by restricting to a vector θ in the span of $\varphi(x_1), \dots, \varphi(x_n)$; that is, 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 $\mathcal{H}_{\mathcal{D}} = \left\{ \sum_{i=1}^{n} \alpha_i \varphi(x_i), \ \alpha \in \mathbb{R}^n \right\} \subset \mathcal{H}$, 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 that 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 Ψ 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 equation (7.1) can be found among the vectors of the form $\theta = \sum_{i=1}^n \alpha_i\varphi(x_i)$, and thus as a *finite-dimensional* optimization problem (of dimension n).

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 vector θ 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 ℓ. 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 α'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* $k : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$, which is a symmetric function equal to the dot product between feature vectors:

$$k(x,x') = \langle\varphi(x),\varphi(x')\rangle.$$

We then have, if $\theta = \sum_{i=1}^n \alpha_i\varphi(x_i)$,

$$\forall j\in\{1,\dots,n\},\ \langle\theta,\varphi(x_j)\rangle = \sum_{i=1}^n \alpha_i k(x_i,x_j) = (K\alpha)_j,$$

where $K\in\mathbb{R}^{n\times n}$ is the *kernel matrix* (Gram matrix of the feature vectors), 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 the optimization problem solely as a function of y, K, and α:

$$\inf_{\theta\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \ell(y_i,\langle\theta,\varphi(x_i)\rangle) + \frac{\lambda}{2}\|\theta\|^2 \ = \ \inf_{\alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^n \ell(y_i,(K\alpha)_i) + \frac{\lambda}{2}\alpha^\top K\alpha. \tag{7.2}$$

Note that for any test point $x\in\mathcal{X}$, we have defined the prediction function as

$$f(x) = \langle\theta,\varphi(x)\rangle = \sum_{i=1}^n \alpha_i\langle\varphi(x_i),\varphi(x)\rangle = \sum_{i=1}^n \alpha_i k(x,x_i).$$

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, as we solely need dot products. This is the *kernel trick*, which allows one to do the following:

- Replace the search space $\mathcal{H}$ by $\mathbb{R}^n$; this is interesting computationally when the dimension of $\mathcal{H}$ is infinite or 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 7.2).

Proposition 7.2 *Given $x_1, \dots, x_n \in \mathcal{X}$, and $y \in \mathbb{R}^n$ such that there is 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 $\alpha \in \mathbb{R}^n$ such that $y = K\alpha$. (This system must then have a solution.)*

7.3 Kernels

In section 7.2, 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 between pairs of points (i.e., the Gram matrix of feature vectors) and is thus symmetric positive semidefinite (see the proof of proposition 7.3); that is, all its eigenvalues are nonnegative, or, equivalently, $\forall \alpha \in \mathbb{R}^n$, $\alpha^\top K \alpha \geqslant 0$. Reciprocally, it turns out that this simple property is enough to ensure 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 the ith 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 noncentered 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 resulting from this kernel function are symmetric positive semidefinite.*

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) *The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 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, for the kernel matrix K associated with these points,

$$\alpha^\top K \alpha = \sum_{i,j=1}^n \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}} = \Big\| \sum_{i=1}^n \alpha_i \varphi(x_i) \Big\|_{\mathcal{H}}^2 \geqslant 0.$$

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 $\mathcal{H}' \subset \mathbb{R}^{\mathcal{X}}$ 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

$$\left\langle \sum_{i=1}^{n} \alpha_i k(\cdot, x_i), \sum_{j=1}^{m} \beta_j k(\cdot, x'_j) \right\rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, x'_j). \tag{7.3}$$

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 in equation (7.3) is equal to $\sum_{j=1}^{m} \beta_j f(x'_j)$ and thus depends only on the values of f, not on its representation (and it is similar for the function on the right of the dot product).

This dot product is bi-linear and always nonnegative when applied to the same function (i.e., in equation (7.3), when $\alpha = \beta$ and the points (x_i) and (x'_j) are the same, we get a nonnegative number because k is positive-definite). Moreover, this dot product satisfies the two properties for any $f \in \mathcal{H}'$, $x, x' \in \mathcal{X}$:

$$\langle k(\cdot, x), f \rangle = f(x) \quad \text{and} \quad \langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x').$$

These are called "reproducing properties" and correspond to an explicit construction of the feature map $\varphi(x) = k(\cdot, x)$.

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 $x \in \mathcal{X}$, 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 $\langle f, f \rangle = 0$.

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) and Berlinet and Thomas-Agnan (2004) for more details. ■

We can make the following observations:

- $\mathcal{H}$ is called the "feature space," and φ the "feature map," which goes from the "input space" $\mathcal{X}$ to $\mathcal{H}$. Note that the existence of a feature map is needed only for the analysis (e.g., in sections 7.5 and 7.6), since algorithms will only use the kernel function values (see section 7.4).
- 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. For any positive-definite kernel k, the particular space of functions that we built is called the *reproducing kernel Hilbert space (RKHS)* associated with k, for which $\varphi(x) = k(\cdot, x)$.

[2] The Cauchy-Schwarz inequality applies to bi-linear forms that are symmetric positive semidefinite, but may not be positive-definite; that is, $\langle f, f \rangle = 0$ may not imply that $f = 0$.

[3] See https://en.wikipedia.org/wiki/Complete_metric_space for definitions.

- A classical intuitive interpretation of the reproducing property 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 exercise 7.1). This implies that not all Hilbert spaces of real-valued functions on $\mathcal{X}$ are RKHSs. Indeed, for example, if $L_2(\mathbb{R}^d)$ (the space of square-integrable functions with respect to the Lebesgue measure) was an RKHS, this would mean that there is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $f(x) = \langle f, k(\cdot, x)\rangle_{L_2(\mathbb{R}^d)} = \int_{\mathbb{R}^d} k(x, x')f(x')dx'$. 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 in the subsequent discussion 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 with some feature map φ such that $k(x, y) = \langle \varphi(x), \varphi(y)\rangle_{\mathcal{H}}$, but also with a *space of functions on $\mathcal{X}$ with a given norm*, either directly through the RKHS or by looking at all functions f_θ 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 we will use the notation $\langle \theta, \varphi(x)\rangle_{\mathcal{H}}$ instead of $f(x)$.
- In this chapter, following the same route as the rest of the book, we will adopt the decision-theoretic approach set forth in section 2.2, with notions of loss functions and risks. Positive-definite kernels and their associated function spaces can also be studied using Bayesian inference through Gaussian processes, as briefly outlined in section 14.3. See more details in Rasmussen and Williams (2006) and explicit algorithmic and theoretical connections with results from this chapter in Kanagawa et al. (2018).

Exercise 7.1 (♦♦) *Let $\mathcal{H}$ be a Hilbert space of real-valued functions on $\mathcal{X}$ endowed with a dot product $\langle \cdot, \cdot\rangle_{\mathcal{H}}$, such that for any $x \in \mathcal{X}$, the linear form $f \mapsto f(x)$ is bounded (i.e., $\sup_{f\in\mathcal{H},\ \|f\|_{\mathcal{H}}\leqslant 1} |f(x)|$ is finite). Using the Riesz representation theorem, show that this is an RKHS.*

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 $x \mapsto \binom{\lambda_1^{1/2}\varphi_1(x)}{\lambda_2^{1/2}\varphi_2(x)} \in \mathcal{H}_1 \times \mathcal{H}_2$ is a feature map for $\lambda_1 k_1 + \lambda_2 k_2$ (note the alternative proof using that the sum of two positive semidefinite matrices is positive semidefinite).

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 other times it is not (but in practice, we never need it). In the next subsections, 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 this discussion with the most obvious kernels on $\mathcal{X} = \mathbb{R}^d$, for which feature maps are easily found.

Linear kernel. We define $k(x, x') = x^\top x'$. This kernel corresponds to a function space composed of linear functions $f_\theta(x) = \theta^\top x$, with an ℓ_2-penalty $\|\theta\|_2^2$. The kernel trick can be useful when the input data have huge dimension d but are quite sparse (i.e., with 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 positive integer, kernel $k(x, x') = (x^\top x')^s$ is positive-definite as an integer power of a kernel and can be explicitly expanded as follows (with the binomial theorem)[4]:

$$k(x, x') = \Big(\sum_{i=1}^d x_i x_i'\Big)^s = \sum_{\alpha_1 + \cdots + \alpha_d = s} \binom{s}{\alpha_1, \dots, \alpha_d} \underbrace{(x_1 x_1')^{\alpha_1} \cdots (x_d x_d')^{\alpha_d}}_{(x_1^{\alpha_1} \cdots x_d^{\alpha_d})((x_1')^{\alpha_1} \cdots (x_d')^{\alpha_d})},$$

where the sum is over all nonnegative 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 $\binom{d+s-1}{s}$.

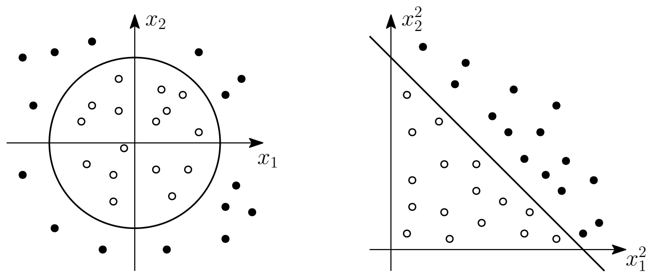
[4] See https://en.wikipedia.org/wiki/Binomial_theorem.

[5] A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be homogeneous if there is $s \in \mathbb{R}_+$ such that for all $x \in \mathbb{R}^d$, and $\lambda \in \mathbb{R}_+$, $f(\lambda x) = \lambda^s f(x)$.

When d and s grow, the feature space dimension grows as d^s and an explicit representation is not desirable; the kernel trick can then 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 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 $\alpha_1 + \cdots + \alpha_d \leqslant s$. Also, 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 following illustration:



7.3.2 Translation-Invariant Kernels on $[0, 1]$

We now consider $\mathcal{X} = [0, 1]$ and kernels of the form $k(x, x') = q(x - x')$ with function $q : [0, 1] \to \mathbb{R}$, which is assumed to be 1-periodic. We will show how they emerge from penalties on the Fourier coefficients of functions, which we quickly review here.[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 correspond 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 function that is 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. 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,

[6]See https://en.wikipedia.org/wiki/Fourier_series for more details.

we start from the squared norm, while in section 7.3.3, we start from the kernel.

Given a function $f \in L_2([0,1])$ decomposed into its Fourier series as

$$f(x) = \sum_{m \in \mathbb{Z}} e^{2im\pi x} \hat{f}_m,$$

we consider the weighted norm

$$\|f\|_c^2 = \sum_{m \in \mathbb{Z}} c_m |\hat{f}_m|^2,$$

with $c \in \mathbb{R}_+^{\mathbb{Z}}$; this penalty can be interpreted through a feature map and its associated dot product. Indeed, consider the Hilbert space $\ell_2(\mathbb{Z})$ of complex-valued square-summable sequences endowed with the dot product $\langle a, b \rangle = \sum_{m \in \mathbb{Z}} a_m b_m^*$ for $a, b \in \ell_2(\mathbb{Z})$. Then, take the feature vector $\varphi(x)_m = e^{-2im\pi x}/\sqrt{c_m}$, and $\theta \in \ell_2(\mathbb{Z})$, such that $\theta_m = \hat{f}_m \sqrt{c_m}$, so $f(x) = \langle \theta, \varphi(x) \rangle$ and $\|\theta\|_{\ell_2(\mathbb{Z})}^2 = \sum_{m \in \mathbb{Z}} |\theta_m|^2$ is equal to the norm $\|f\|_c^2 = \sum_{m \in \mathbb{Z}} c_m |\hat{f}_m|^2$.

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 takes the form $q(x - x')$ for a 1-periodic function q with Fourier series $\hat{q}_m = \frac{1}{c_m}$ for all $m \in \mathbb{Z}$.

What we showed here 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 finite. The kernel function then takes the form $k(x, x') = q(x - x')$, with q being 1-periodic, and such that the Fourier series has nonnegative real values $\hat{q}_m = c_m^{-1}$. In the other direction, all such kernels are positive-definite (see the extension to $\mathbb{R}^d$ discussed in section 7.3.3).

Penalization of derivatives. For power-law penalties based on the sequence $(c_m)_{m \in \mathbb{Z}}$, there is a natural link with penalties on derivatives, as, if f is s-times differentiable[7] with a square-integrable derivative, we have, by differentiating the Fourier series representation,

$$f^{(s)}(x) = \sum_{m \in \mathbb{Z}} (2im\pi)^s e^{2im\pi x} \hat{f}_m.$$

Thus, from Parseval's theorem, we get:

$$\int_0^1 |f^{(s)}(x)|^2 dx = (2\pi)^{2s} \sum_{m \in \mathbb{Z}} m^{2s} |\hat{f}_m|^2.$$

In this chapter, we will consider penalizing such derivatives, leading to Sobolev spaces on $[0,1]$ (see extensions in section 7.3.3). The following examples are often considered:

[7] More precisely, f is 1-periodic and almost everywhere s-times differentiable with $\int_0^1 (f^{(s)}(x))^2 dx$ bounded.

- **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

$$k(x,x') = \sum_{m \in \mathbb{Z}} c_m^{-1} e^{2im\pi(x-x')} = 1 + \sum_{m \geqslant 1} \frac{2\cos[2\pi m(x-x')]}{m^{2s}} = q(x-x').$$

To have an expression for q (and thus k) in closed form, we notice that if we take the $2s$th-order derivative of q, we get

$$q^{(2s)}(t) = 2(-1)^s (2\pi)^{2s} \sum_{m \geqslant 1} \cos[2\pi m t] = (2\pi)^{2s}(-1)^s \sum_{m \in \mathbb{Z}} \exp[2i\pi m t] - (2\pi)^{2s}(-1)^s,$$

which is equal to $(2\pi)^{2s}(-1)^{s-1}$ for $t \notin \mathbb{Z}$. Thus, if we define $\{t\} = t - \lfloor t \rfloor \in [0,1)$, the fractional part of t, the function q should be a polynomial in $\{t\}$ of degree $2s$, with largest coefficient $(-1)^{s-1} \frac{(2\pi)^{2s}}{(2s)!}$.

To compute the exact polynomial for $s = 1$, we can check (by computing the Fourier series coefficients by integration) that $\{t\} = \frac{1}{2} - \frac{1}{2\pi} \sum_{m \geqslant 1} \frac{2\sin[2\pi m t]}{m}$, and by integrating between 0 and t, that $\frac{1}{2}\{t\}^2 = \frac{\{t\}}{2} + \frac{1}{(2\pi)^2} \sum_{m \geqslant 1} \frac{2(\cos[2\pi m t]-1)}{m^2}$. Using that $\sum_{m \geqslant 1} \frac{1}{m^2} = \frac{\pi^2}{6}$, this leads to $q(t) = 2\pi^2\{t\}^2 - 2\pi^2\{t\} + \pi^2/3 + 1$. which is plotted in figure 7.1 (left).

For $s \geqslant 1$, it turns out we have $k(x,x') = 1 + (-1)^{s-1} \frac{(2\pi)^{2s}}{(2s)!} B_{2s}(\{x-x'\})$, where B_{2s} the $(2s)$th Bernoulli polynomial,[8] from which we can confirm the computation above for $s = 1$ 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 $q(t) = 1 - \frac{(2\pi)^4}{24}\big(\{t\}^4 - 2\{t\}^3 + \{t\}^2 - \frac{1}{30}\big)$.*

- **Periodic exponential kernel**: Here, we can consider $c_m = 1 + \alpha^2 |m|^2$, for which we also have a closed-form formula, with the norm $\|f\|_{\mathcal{H}}^2 = \frac{\alpha^2}{(2\pi)^2} \int_0^1 |f'(x)|^2 dx + \int_0^1 |f(x)|^2 dx$.

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.*[9]

[8] See https://en.wikipedia.org/wiki/Bernoulli_polynomials.
[9] See https://francisbach.com/cauchy-residue-formula/.

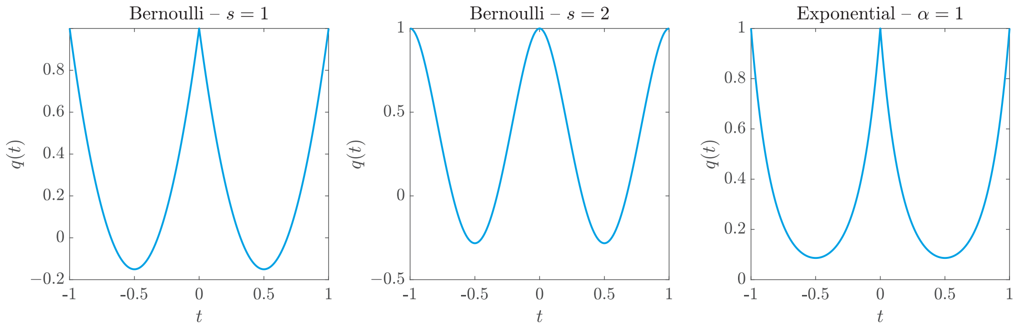


Figure 7.1. Translation-invariant kernels on $[0, 1]$, of the form $k(x, x') = q(x - x')$, with q 1-periodic, for the kernels based on Bernoulli polynomials (left and middle), and for the periodic exponential kernel (right). Kernels are normalized so $k(x, x) = 1$.

These kernels are mainly used for their simplicity and explicit feature map, which are simpler than the kernels described next, which are most used in practice (with similar links to Sobolev spaces). Note also that for inputs uniformly distributed 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). Note that the link between orthonormal bases and positive-definite kernels is more general; see exercises 7.6 and 7.7, as well as Steinwart and Scovel (2012).

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).

Exercise 7.6 (Mercer kernels) *Consider a probability distribution p on a set $\mathcal{X}$, an orthonormal basis $(\varphi_i)_{i\in I}$ of the Hilbert space $L_2(p)$ of square-integrable functions (with I countable), and a summable positive sequence $(\lambda_i)_{i\in I}$. Show that the function defined as $k(x, x') = \sum_{i\in I} \lambda_i \varphi_i(x)\varphi_i(x')$ is a positive-definite kernel and describe an associated feature space.*

Exercise 7.7 (Mercer decomposition (♦♦)) *Consider a probability distribution p on a set $\mathcal{X}$, a positive-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and the operator T defined on $L_2(p)$ as $Tf(y) = \int_{\mathcal{X}} k(x, y)f(x)dp(x)$.*

- *Show that if $\int_{\mathcal{X}}\int_{\mathcal{X}} k(x, y)^2 dp(x)dp(y)$ is finite, then the operator T is bounded (it is an instance of Hilbert-Schmidt integral operator[10]).*
- *Given an orthonormal basis $(e_i)_{i\in I}$ of $L_2(p)$ composed of eigenvectors for T (which is assumed to exist), show that the corresponding eigenvalues $(\lambda_i)_{i\in I}$ are nonnegative and $k(x, x') = \sum_{i\in I} \lambda_i \varphi_i(x)\varphi_i(x')$ (convergence meant in the norm $L_2(p)$).*

[10] See https://en.wikipedia.org/wiki/Hilbert-Schmidt_integral_operator.

7.3.3 Translation-Invariant Kernels on $\mathbb{R}^d$

We now consider $\mathcal{X} = \mathbb{R}^d$ and a kernel of the form $k(x, x') = q(x - x')$, with function $q : \mathbb{R}^d \to \mathbb{R}$, which we refer to as "translation-invariant," as it is invariant under the addition of the same constant to both arguments. We start with a short review of Fourier transforms.[11]

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

$$\hat{f}(\omega) = \int_{\mathbb{R}^d} f(x) e^{-i\omega^\top x} dx,$$

which is then a continuous function of ω. It can naturally be extended to an operator on all square-integrable functions, and under appropriate conditions on f (e.g., both f and $\hat{f}$ integrable), 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 $\displaystyle\int_{\mathbb{R}^d} |f(x)|^2 dx = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\hat{f}(\omega)|^2 d\omega.$

Translation-invariant kernels. Proposition 7.4 gives conditions under which we obtain a positive-definite kernel.

Proposition 7.4 (Bochner's theorem) *A translation-invariant kernel k defined as $k(x, x') = q(x - x')$ is positive-definite if and only if q is the inverse Fourier transform of a nonnegative Borel measure.*

Partial proof Here, we are just giving the proof of the direction we need for the purposes of this discussion. Assume that

$$q(x - x') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{i(x-x')^\top \omega} d\mu(\omega)$$

for a nonnegative measure μ. Let $x_1, \dots, x_n \in \mathbb{R}^d$ and $\alpha_1, \dots, \alpha_n \in \mathbb{R}$. We have

$$\begin{aligned}
\sum_{s,j=1}^n \alpha_s \alpha_j k(x_s, x_j) = \sum_{s,j=1}^n \alpha_s \alpha_j q(x_s - x_j) &= \frac{1}{(2\pi)^d} \sum_{s,j=1}^n \alpha_s \alpha_j \int_{\mathbb{R}^d} e^{i\omega^\top (x_s - x_j)} d\mu(\omega) \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \Big(\sum_{s,j=1}^n \alpha_s \alpha_j e^{i\omega^\top x_s} (e^{i\omega^\top x_j})^* \Big) d\mu(\omega) \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \Big| \sum_{s=1}^n \alpha_s e^{i\omega^\top x_s} \Big|^2 d\mu(\omega) \geqslant 0,
\end{aligned}$$

[11] See https://en.wikipedia.org/wiki/Fourier_transform for more details.

which shows the positive-definiteness. See Reed and Simon (1978) and theorem 2.7 from Varadhan (2001) for a proof of the other direction. ■

In practice, when q and $\hat{q}$ are both integrable, the associated kernel is positive-definite if and only if $\forall \omega \in \mathbb{R}^d$, $\hat{q}(\omega) \geqslant 0$.

Construction of the associated norm. We first give an intuitive nonrigorous reasoning: 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 takes the form $\langle \varphi(x), \varphi(x') \rangle$, with $\varphi(x)_\omega = \frac{1}{(2\pi)^{d/2}} \sqrt{\hat{q}(\omega)} e^{i\omega^\top x}$ (it is nonrigorous because the index ω belongs to $\mathbb{R}^d$, which is not countable). If we consider function f defined as $f(x) = \int_{\mathbb{R}^d} \varphi(x)_\omega \theta_\omega d\omega = \langle \varphi(x), \theta \rangle$, then we need to have $\theta_\omega = \frac{1}{(2\pi)^{d/2}} \hat{f}(\omega)/\sqrt{\hat{q}(\omega)}$. The squared norm of θ is then 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

$$\|f\|^2_{\mathcal{H}} = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(w)|^2}{\hat{q}(\omega)} d\omega. \tag{7.4}$$

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 (proof left as an exercise). Note the similarity with the penalty for the kernel on $[0, 1]$ (see more similarity next with links with derivatives).

Link with derivatives. When f has partial derivatives, the Fourier transform of $\frac{\partial f}{\partial x_j}$ equals $i\omega_j$ multiplied by 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 as follows:

$$\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\omega_1^{j_1} \cdots \omega_d^{j_d}|^2 |\hat{f}(w)|^2 d\omega = \int_{\mathbb{R}^d} \left| \frac{\partial^j f}{\partial x_1^{j_1} \cdots \partial x_d^{j_d}}(x) \right|^2 dx, \tag{7.5}$$

for a vector $j \in \mathbb{N}^d$. This will allow us to find corresponding norms by expanding $\hat{q}(\omega)^{-1}$ as a sum 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" (with unit 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, p. 84). Thus, for d being odd, $\hat{q}(\omega)^{-1}$ is a sum of monomials, and looking at their orders, we see that the corresponding RKHS norm (i.e., the norm on the space of functions on $\mathbb{R}^d$ that our kernel defines) is penalizing all derivatives up to the total order $(d+1)/2$; that is, in equation (7.5), for all $j \in \mathbb{N}^d$ such that $j_1 + \cdots + j_d \leqslant (d+1)/2$, which is a Sobolev space. For d even, we get a fractional Sobolev space.[12]

In particular, for $d = 1$, we have $\hat{q}(\omega) = \frac{2r}{1+r^2\omega^2}$, and thus

$$\begin{aligned}
\|f\|_{\mathcal{H}}^2 &= \frac{1}{2\pi}\int_{\mathbb{R}} \frac{|\hat{f}(w)|^2}{\hat{q}(\omega)} d\omega = \frac{1}{2r}\frac{1}{2\pi}\int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega + \frac{r}{2}\frac{1}{2\pi}\int_{\mathbb{R}} |\omega \hat{f}(\omega)|^2 d\omega \\
&= \frac{1}{2r}\int_{\mathbb{R}} |f(x)|^2 dx + \frac{r}{2}\int_{\mathbb{R}} |f'(x)|^2 dx,
\end{aligned}$$

and we recover the Sobolev space of functions with square-integrable derivatives.

The constant r is homogeneous (in terms of unit) to input x, while the constant R will be homogeneous to features $\varphi(x)$ (i.e., 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 that is penalizing all derivatives of all orders. Note that all members of this RKHS (the associated function space) are infinitely differentiable and, therefore, much smoother than functions coming from the exponential kernel (the RKHS is smaller); see figure 7.2.

Matern kernels and Sobolev spaces. More generally, one can define a series of kernels such that $\hat{q}(\omega)$ is proportional to $r^d(1 + r^2\|\omega\|_2^2)^{-s}$ 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 section 2.10 in Stein (2012). 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 ($s = 1$) leads to an RKHS only for $d = 1$, and having $s = 0$ (i.e., for $L_2(\mathbb{R}^d)$) never does.

For $s = \frac{d+1}{2}$, we obtain the exponential kernel $k(x, x') = \exp(-\|x - x'\|_2/r)$. 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); see Rasmussen and Williams (2006, p. 84).

Density in $L_2(\mathbb{R}^d)$. For all the kernels discussed here, 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 L_2-norm) by a function in $\mathcal{H}$. This is made quantitative in section 7.5.2.

[12]See `https://en.wikipedia.org/wiki/Sobolev_space`.

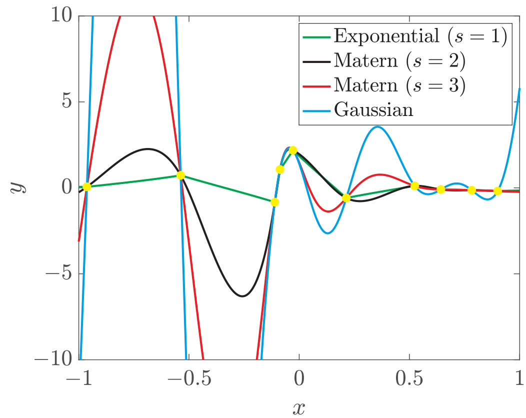


Figure 7.2. Examples of functions in the RKHS for several kernels. All functions are the minimum norm interpolators of the yellow points for the corresponding RKHS.

⚠ In this chapter, we will later consider two spaces of integrable functions, with respect to the Lebesgue measure (which is not a probability measure), which we denote as $L_2(\mathbb{R}^d)$, and with respect to the probability measure of the input data, which we denote as $L_2(p)$. If p has a density with respect to the Lebesgue measure and this density $\frac{dp}{dx}$ is uniformly bounded, then $L_2(\mathbb{R}^d) \subset L_2(p)$; more precisely, $\|f\|_{L_2(p)} \leqslant \left\|\frac{dp}{dx}\right\|_\infty^{1/2} \|f\|_{L_2(\mathbb{R}^d)}$. However, the converse is not true, simply because being an element of $L_2(\mathbb{R}^d)$ imposes zero limit at infinity, which being an element of $L_2(p)$ does not impose; moreover, nonzero constants are in $L_2(p)$ but not in $L_2(\mathbb{R}^d)$. Note, moreover, that $\left\|\frac{dp}{dx}\right\|_\infty$ 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. Here, we sampled $n = 10$ random points in $[-1, 1]$ with 10 random responses $y_1, \ldots, y_n$, and we look for the function $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for all $i \in \{1 \ldots, n\}$ 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 proposition 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). Note that each RKHS implicitly imposes an a priori on the function reconstruction by penalizing by an induced norm: see how smoothness requirements between Gaussian and exponential kernel lead to different interpolations.

7.3.4 Beyond Vectorial Input Spaces (♦)

While our theoretical analysis of kernel methods focuses 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 a number of classic examples in this discussion (see more details by Shawe-Taylor and Cristianini, 2004):

- Subsets of a given set V: For example, 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").[13]
- Point clouds: A point cloud in $\mathbb{R}^d$ is a finite subset of $\mathbb{R}^d$, in no particular order. Such clouds 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 (which ignores the ordering of the points) and then use a kernel on these averages. Other kernels may be obtained as functions of the concatenation of the two point clouds (see more details in 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 each word in the corresponding document. This gives a typically high-dimensional feature vector $\varphi(x)$ (with the vocabulary size as the dimension). 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 sources like Joulin et al. (2017) for examples.
- Sequences: Given some finite alphabet $\mathcal{A}$, we consider the set $\mathcal{X}$ of finite sequences in $\mathcal{A}$ of 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 if y is a subsequence of x (we could also count the number of times that 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 to compute the kernel function (there also are much faster algorithms; see Gusfield, 1997). These kernels have many applications in bioinformatics (Schölkopf et al., 2004).

  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 2010s with the use of large amounts of data, several kernels were designed for images, with often a bag-of-words assumption that provides invariance by translation. The key is what to consider as "words"; that is, 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.

[13] See https://en.wikipedia.org/wiki/Jaccard_index.

7.4 Algorithms

In this section, we describe algorithms aimed at solving

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2, \tag{7.6}$$

for ℓ being convex with respect to its second variable. We assume that features are bounded; that is, for all $i \in \{1, \dots, n\}$, $k(x_i, x_i) = \|\varphi(x_i)\|_{\mathcal{H}}^2 \leqslant R^2$.

7.4.1 Representer Theorem

We can directly apply the representer theorem, as done in equation (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 ℓ 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

$$\alpha = (K + n\lambda I)^{-1} y, \tag{7.7}$$

which is not unique when 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 in section 7.4.4). When the loss is smooth, the Hessians are equal to $\frac{1}{n} K \operatorname{Diag}(h) K + \lambda K$, where $h \in \mathbb{R}^n$ is a vector of second-order derivatives of ℓ, so that the Hessians are ill conditioned (i.e., with a large condition number).[14]

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 is the rank of K, and solve

$$\min_{\beta \in \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, (\Phi\beta)_i) + \frac{\lambda}{2} \|\beta\|_2^2, \tag{7.8}$$

with optimality condition $\frac{1}{n} \Phi^\top g + \lambda\beta = 0$, where $g \in \mathbb{R}^n$ is the vector of gradients defined as $g_i = \ell'(y_i, (\Phi\beta)_i)$ for all $i \in \{1, \dots, n\}$ (derivative with respect to the second variable). We can then obtain $\alpha \in \mathbb{R}^n$ as $\alpha = -\frac{1}{\lambda n} g$, so that $\beta = \Phi^\top \alpha$.

[14] For example, for the square loss, where $\operatorname{Diag}(h) = I$, the condition number of the $\frac{1}{n} K^2 + \lambda K$ is at least the one of K.

Note that this corresponds to an explicit feature space representation (i.e., the rows of Φ 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 λ and is thus directly controlled by regularization.

Computing a square root can be done in several ways through Cholesky decomposition or singular value decomposition (SVD), in running time $O(m^2 n)$ (see Golub and Loan, 1996).

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 submatrix 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 the following for an illustration when $I = \{1,\dots,m\}$ and a partition of the kernel matrix:

$$
\begin{array}{|c|c|}
\hline
K(I,I) & K(I,J) \\
\hline
K(J,I) & K(J,J) \\
\hline
\end{array}
$$

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 section 7.4.5), and is thus linear in n.

This approximation technique is standard in linear algebra (see, e.g., Mahoney and Drineas, 2009; Martinsson and Tropp, 2020) and is often called "Nyström approximation" in the context of machine learning (Williams and Seeger, 2000). It proves to be particularly useful when columns are chosen randomly (see the theoretical analysis by Rudi et al., 2015).

Exercise 7.8 (♦) *Show that column sampling corresponds to approximating optimally each $\varphi(x_j)$, $j \notin I$, by a linear combination of $\varphi(x_i)$, $i \in I$.*

Exercise 7.9 *Show that the matrix $K - K(V,I)K(I,I)^{-1}K(I,V)$ is positive-definite. If $\|M\|_*$ denotes the nuclear norm (sum of absolute values of eigenvalues of symmetric matrix M), show that the approximation error $\|K - K(V,I)K(I,I)^{-1}K(I,V)\|_*$ can be computed without the need to compute the entire matrix K.*

7.4.3 Random Features

Some kernels have a special form that leads to specific approximation schemes; that is,

$$k(x, x') = \int_{\mathcal{V}} \varphi(x, v)\varphi(x', v) d\tau(v) = \langle \varphi(x, \cdot), \varphi(x', \cdot)\rangle_{L_2(\tau)},$$

where τ is a probability distribution on a 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 independently and identically distributed (i.i.d.) from τ. We can thus use an explicit feature representation $\hat{\varphi}(x) = \big(\frac{1}{\sqrt{m}}\varphi(x, v_j)\big)_{j\in\{1,\ldots,m\}}$ and solve

$$\min_{\beta\in\mathbb{R}^m} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \hat{\varphi}(x_i)^\top \beta) + \frac{\lambda}{2} \|\beta\|_2^2,$$

with a predictor $x \mapsto \beta^\top \hat{\varphi}(x)$, with any algorithm from chapter 5.

For this scheme to make sense, the number m of random features has to be significantly smaller than n (otherwise, it is as efficient to use the square root of K as in equation (7.8), with no approximation), which is often sufficient in practice (see an analysis by Rudi and Rosasco, 2017).

⚠ Note that here, dimension reduction is performed independent 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** (section 7.3.3): For these kernels, we have $k(x, x') = q(x - x') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{q}(\omega) e^{i\omega^\top(x-x')} d\omega$, for which we can take complex-valued features $\varphi(x, \omega) = \sqrt{q(0)} e^{i\omega^\top x} \in \mathbb{C}$, where ω 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 (ReLU; i.e., $\sigma(\alpha) = \max\{0, \alpha\}$), and for v sampled uniformly on the sphere, we have $k(x, x') = \frac{\|x\|_2 \|x'\|_2}{2(d+1)\pi} \big[(\pi - \eta)\cos\eta + \sin\eta\big]$, where $\cos\eta = \frac{x^\top x'}{\|x\|_2 \cdot \|x'\|_2}$ (Le Roux and Bengio, 2007). (The proof is left as an exercise.) 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 of this point in chapter 9 (section 9.5).

Exercise 7.10 *In the setup of exercise 7.6, provide a random feature expansion of Mercer kernels.*

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 that 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 this 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 λ added on top of the regular multiplier α for convenience):

$$\begin{aligned} & \max_{\alpha \in \mathbb{R}^n} \min_{\theta \in \mathcal{H}, \; u \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, u_i) + \frac{\lambda}{2} \|\theta\|^2 + \lambda \sum_{i=1}^n \alpha_i \big(u_i - \langle \varphi(x_i), \theta \rangle \big) \\ = \; & \max_{\alpha \in \mathbb{R}^n} \Big\{ \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \big\{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \big\} + \min_{\theta \in \mathcal{H}} \Big\{ \frac{\lambda}{2} \|\theta\|^2 - \lambda \sum_{i=1}^n \alpha_i \langle \varphi(x_i), \theta \rangle \Big\} \Big\} \end{aligned}$$

by reordering terms. We can then optimize in closed form with respect to θ, as:

$$\begin{aligned} & \max_{\alpha \in \mathbb{R}^n} \Big\{ \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \big\{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \big\} - \frac{\lambda}{2} \Big\| \sum_{i=1}^n \alpha_i \varphi(x_i) \Big\|^2 \Big\} \text{ with } \theta = \sum_{i=1}^n \alpha_i \varphi(x_i), \\ = \; & \max_{\alpha \in \mathbb{R}^n} \Big\{ \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \big\{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \big\} - \frac{\lambda}{2} \alpha^\top K \alpha \Big\}, \end{aligned} \tag{7.9}$$

with $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$ at optimum. Since the functions $\alpha_i \mapsto \min_{u_i \in \mathbb{R}} \big\{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \big\}$ 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, and another is a maximization problem). Moreover, when the loss is smooth, one can show that the function $\alpha_i \mapsto \min_{u_i \in \mathbb{R}} \big\{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \big\}$ is a strongly concave function,[15] and

[15] More precisely, if $u_i \mapsto \ell_i(y_i, u_i)$ is L-smooth, then the function $\alpha_i \mapsto -\min_{u_i \in \mathbb{R}} \big\{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \big\}$ is $(n\lambda/L)$-strongly-convex (proof left as an exercise).

thus relatively easy to optimize (in other words, the associated condition numbers of dual problems are smaller than when using the representer theorem). See exercises 7.11 and 7.12.

Exercise 7.11 *(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* $(\Phi\Phi^\top + n\lambda I)^{-1}\Phi = \Phi(\Phi^\top\Phi + n\lambda I)^{-1}$.

Exercise 7.12 *Write down the dual problem in equation (7.9) for the logistic loss and the for the hinge loss (compare the results to section 4.1.2).*

Exercise 7.13 (Unregularized constant term) *Consider the minimization problem* $\min_{\theta\in\mathcal{H},c\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^n \ell(y_i, \langle\varphi(x_i),\theta\rangle + c) + \frac{\lambda}{2}\|\theta\|^2$. *If the loss function is convex with respect to the second variable, show that the dual problem is the one in equation (7.9) with the additional constraint that* $\sum_{i=1}^n \alpha_i = 0$. *Without any assumption on the loss function, show that we can restrict the search space for* θ *to all combinations* $\sum_{i=1}^n \alpha_i\varphi(x_i)$ *with the same constraint that* $\sum_{i=1}^n \alpha_i = 0$.

Exercise 7.14 (Limit of Gaussian kernel for infinite bandwidth) *Consider the minimization problem* $\min_{\theta\in\mathcal{H},c\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^n \ell(y_i, \langle\varphi(x_i),\theta\rangle + c) + \frac{\lambda}{2}\|\theta\|^2$ *from exercise 7.13. For the Gaussian kernel* $k(x,x') = \exp(-\|x-x'\|_2^2/r^2)$, *show that when* r *tends to infinity, the resulting prediction function is the same as the one obtained by the linear kernel* $k(x,x') = x^\top x'$ *with the regularization parameter* $\lambda r^2/2$.

Exercise 7.15 (Optimization of the kernel) *Show that for convex loss functions, the maximal value in equation (7.9) is a convex function of the kernel matrix* K. *For the square loss, show that it is equal to* $\frac{\lambda}{2} y^\top(K + n\lambda I)^{-1}y$.

7.4.5 Stochastic Gradient Descent (♦)

When minimizing an expectation

$$\min_{\theta\in\mathcal{H}} \Big\{\mathbb{E}\big[\ell(y, \langle\varphi(x),\theta\rangle)\big] + \frac{\lambda}{2}\|\theta\|^2\Big\}$$

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 ℓ' is the derivative with respect to the second variable.

When initializing at $\theta_0 = 0$, θ_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 α 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 $F_\lambda(\theta) = \mathbb{E}\big[\ell(y, \langle\varphi(x), \theta\rangle)\big] + \frac{\lambda}{2}\|\theta\|^2$, we have, for the averaged iterate θ_t (from proposition 5.8),

$$\mathbb{E}\big[F_\lambda(\bar\theta_t)\big] - \inf_{\theta\in\mathcal{H}} F_\lambda(\theta) \leqslant \frac{2G^2R^2(1+\log t)}{\lambda t}.$$

When doing a single pass with $t = n$, then $F_\lambda(\theta)$ is the regularized *expected* risk, and we obtain a generalization bound (i.e., on unseen data) for the expected risk $F(\theta) = \mathbb{E}\big[\ell(y, \langle\varphi(x), \theta\rangle)\big]$, leading to $\mathbb{E}\big[F(\bar\theta_n)\big] \leqslant \frac{2G^2R^2(1+\log n)}{\lambda n} + \inf_{\theta\in\mathcal{H}}\big\{F(\theta) + \frac{\lambda}{2}\|\theta\|_{\mathcal{H}}^2\big\}$. These bounds are similar to the ones in section 7.5 (which assume that a regularized empirical risk minimizer is available).

Exercise 7.16 (♦) *Consider the minimization of $F(\theta) = \mathbb{E}\big[\ell(y, \langle\varphi(x), \theta\rangle)\big]$ using constant step-size SGD for a convex G-Lipschitz-continuous loss and features almost surely bounded by R. Show that after t steps (initialized at $\theta_0 = 0$ and with step size γ), the averaged iterate $\bar\theta_t$ satisfies $\mathbb{E}\big[F(\bar\theta_t)\big] \leqslant \inf_{\theta\in\mathcal{H}}\big\{F(\theta) + \frac{\|\theta\|_{\mathcal{H}}^2}{2\gamma t}\big\} + \frac{\gamma G^2R^2}{2}$.*

7.4.6 Kernelization of Linear Algorithms

Beyond supervised learning, many unsupervised learning algorithms can be "kernelized," such as principal component analysis (PCA, as presented in section 3.9), K-means, or canonical correlation analysis.[16] 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 as well as exercises 7.17 and 7.18.

Exercise 7.17 (Kernel PCA) *We consider n observations $x_1, \dots, x_n$ in a set $\mathcal{X}$ equipped with a positive-definite kernel and feature map φ from $\mathcal{X}$ to $\mathcal{H}$. Show that the largest eigenvector of the empirical noncentered 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 kernel k, relate this eigenvalue problem to the maximizer of $\frac{1}{n}\sum_{i=1}^n f(x_i)^2$ subject to $\|f\|_{\mathcal{H}} = 1$.*

[16] See https://en.wikipedia.org/wiki/Canonical_correlation.

Exercise 7.18 (Kernel K-means) *Show that the K-means clustering algorithm*[17] *can be expressed only using dot products.*

Exercise 7.19 (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 that is linear in φ, 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 side with the kernel function and show how to minimize it 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 $\mathbb{E}\Big[\big\| \int_{\mathcal{X}} \varphi(x)dp(x) - \sum_{i=1}^n \alpha_i \varphi(x_i)\big\|^2\Big] \leqslant \frac{1}{n}\mathbb{E}[k(x,x)]$.

Exercise 7.20 *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 μ_+ (μ_-) be the mean of all feature vectors for positive (negative) labels. We consider the classification rule that predicts 1 if $\|\varphi(x) - \mu_+\|^2_{\mathcal{H}} < \|\varphi(x) - \mu_-\|^2_{\mathcal{H}}$ 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 a minimizer $\hat{f}_D^{(c)}$ of the constrained problem

$$\min_{f \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) \text{ such that } \|f\|_{\mathcal{H}} \leqslant D, \tag{7.10}$$

as well as the unique minimizer $\hat{f}_\lambda^{(r)}$ of the regularized problem

$$\min_{f \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}. \tag{7.11}$$

We denote as $\mathcal{R}(f) = \mathbb{E}\big[\ell(y, f(x))\big]$ the expected risk, and as f_* one of its minimizers (which we assume to be square-integrable). For the square loss, see section 7.6.

[17] See https://en.wikipedia.org/wiki/K-means_clustering.

As in section 4.3, we can first relate the excess risk to the L_2-norm of $f - f_*$ as (using Jensen's inequality)

$$\begin{aligned} \mathcal{R}(f) - \mathcal{R}(f_*) &\leqslant \mathbb{E}\big[|\ell(y, f(x)) - \ell(y, f_*(x))|\big] \leqslant G\,\mathbb{E}\big[|f(x) - f_*(x)|\big] \\ &\leqslant G\sqrt{\mathbb{E}\big[|f(x) - f_*(x)|^2\big]} = G\|f - f_*\|_{L_2(p)}; \end{aligned}$$

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 in section 7.3.3 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 $x \in \mathcal{X}$. Note that for translation-invariant kernels of the form $k(x, x') = q(x - x')$, this is always true with $R^2 = q(0) = k(x, x)$ for all $x \in \mathcal{X}$.

Constrained problem. Dimension-free results from chapter 4 (proposition 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

$$\mathbb{E}\big[\mathcal{R}(\hat{f}_D^{(c)})\big] - \mathcal{R}(f_*) \leqslant \frac{4GDR}{\sqrt{n}} + G \inf_{\|f\|_{\mathcal{H}}\leqslant D} \|f - f_*\|_{L_2(p)}. \tag{7.12}$$

(The first term is the estimation error of using the empirical risk minimizer constrained to the ball of the 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 $|a| + |b| \leqslant \sqrt{2(a^2 + b^2)}$)

$$\begin{aligned} \inf_{D\geqslant 0} \frac{4GRD}{\sqrt{n}} + G \inf_{\|f\|_{\mathcal{H}}\leqslant D} \|f - f_*\|_{L_2(p)} &= \inf_{f\in\mathcal{H}} \Big\{\frac{4GR\|f\|_{\mathcal{H}}}{\sqrt{n}} + G\|f - f_*\|_{L_2(p)}\Big\} \\ &\leqslant G\sqrt{2 \inf_{f\in\mathcal{H}} \Big\{\|f - f_*\|_{L_2(p)}^2 + \frac{16R^2}{n}\|f\|_{\mathcal{H}}^2\Big\}}. \end{aligned} \tag{7.13}$$

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 equation (7.13) (proof left as an exercise).

Overall, we need to understand how the deterministic quantity

$$A(\mu, f_*) = \inf_{f\in\mathcal{H}} \Big\{\|f - f_*\|_{L_2(p)}^2 + \mu\|f\|_{\mathcal{H}}^2\Big\} \tag{7.14}$$

goes to zero when μ goes to zero. (Note that we define $A(\mu, f_*)$ through a regularized estimation problem to study trade-offs between estimation and approximation errors, and 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 $\mathcal{H}$ (a well-specified problem), then we have $A(\mu, f_*) \leqslant \mu\|f_*\|_{\mathcal{H}}^2$, and thus it tends to zero as $O(\mu)$ when μ tends to zero. This is the best-case scenario, and it 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 $\mu = 16R^2/n$, the overall excess risk in equation (7.12) goes to zero as $G\sqrt{2A(\mu, f_*)} \leqslant 4\sqrt{2}GR\|f_*\|_{\mathcal{H}}/\sqrt{n} = O(1/\sqrt{n})$. Moreover, the suggested value of D not surprisingly is exactly $\|f_*\|_{\mathcal{H}}$.
- The target function f_* is not in $\mathcal{H}$ (a misspecified problem), but it can be approached arbitrarily closely in the $L_2(p)$-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 μ 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 as $\Pi_{\bar{\mathcal{H}}}(f_*)$ the orthogonal projection in $L_2(p)$ of f_* on the closure of $\mathcal{H}$, by the Pythagorean theorem, we have $A(\mu, f_*) = A(\mu, \Pi_{\bar{\mathcal{H}}}(f_*)) + \|f_* - \Pi_{\bar{\mathcal{H}}}(f_*)\|_{L_2(p)}^2$; that is, there is an incompressible error due to a choice of function space that is not large enough.

Note that we will use the same reasoning based on equation (7.14) for neural networks in section 9.4.

Regularized problem (♦). For the regularized problem, we can use the bound from chapter 4 (proposition 4.6):

$$\mathbb{E}\big[\mathcal{R}(\hat{f}_\lambda^{(r)})\big] - \mathcal{R}(f_*) \quad \leqslant \quad \frac{24G^2R^2}{\lambda n} + \inf_{f\in\mathcal{H}} \Big\{ G\|f - f_*\|_{L_2(p)} + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \Big\}.$$

We can now minimize the bound with respect to λ, leading to a joint optimization problem over (λ, f). With f fixed, the optimal λ is $\lambda = \frac{4\sqrt{3}RG}{\|f\|_{\mathcal{H}}\sqrt{n}}$ and we obtain the bound:

$$G \inf_{f\in\mathcal{H}} \Big\{ \|f - f_*\|_{L_2(p)} + \frac{4\sqrt{3}R}{\sqrt{n}}\|f\|_{\mathcal{H}} \Big\} \leqslant G\sqrt{2 \inf_{f\in\mathcal{H}} \Big\{ \|f - f_*\|_{L_2(p)}^2 + \frac{48R^2}{n}\|f\|_{\mathcal{H}}^2 \Big\}},$$

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 $\lambda = \frac{4\sqrt{3}RG}{\|f_*\|_{\mathcal{H}}\sqrt{n}}$ (similar value as in section 4.5.5).

7.5.2 Approximation Error for Translation-Invariant Kernels on $\mathbb{R}^d$

We start by analyzing kernel methods' approximation error for translation-invariant kernels. Given a distribution p of inputs, the goal is to compute

$$A(\mu, f_*) = \inf_{f \in \mathcal{H}} \Big\{ \|f - f_*\|^2_{L_2(p)} + \mu \|f\|^2_{\mathcal{H}} \Big\},$$

where f_* is the target function (e.g., the minimizer of the test risk), which we assume is square-integrable (i.e., in $L_2(p)$). If $A(\mu, f_*)$ tends to zero when μ 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 the constant r to preserve the homogeneity of units. 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

$$\widetilde{A}(\mu, f_*) = \inf_{f \in \mathcal{H}} \Big\{ \tfrac{1}{r^d} \|f - f_*\|^2_{L_2(\mathbb{R}^d)} + \mu \|f\|^2_{\mathcal{H}} \Big\},$$

keeping in mind that $A(\mu, f_*) \leqslant (C/r^d) \widetilde{A}(\mu r^d / C, f_*)$. Remember from section 7.5.1 that if $f_* \in \mathcal{H}$ (the 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 the translation-invariant kernels defined in section 7.3.3, an explicit formulation of the norm $\|\cdot\|_{\mathcal{H}}$ as $\|f\|^2_{\mathcal{H}} = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} d\omega$ (see equation (7.4)), and thus

$$\widetilde{A}(\mu, f_*) \quad = \quad \inf_{\hat{f} \in L_2(\mathbb{R}^d)} \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \Big[\tfrac{1}{r^d} |\hat{f}(\omega) - \hat{f}_*(\omega)|^2 + \mu \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} \Big] d\omega.$$

This is an optimization problem in infinite dimension, but like for the computation of the Bayes risk in section 2.2.3, the optimization with respect to $\hat{f}$ can be performed independently for each ω, which is a quadratic problem in $\hat{f}(\omega)$. 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 $\hat{f}(\omega) = \frac{\hat{f}_*(\omega)}{1 + \mu r^d \hat{q}(\omega)^{-1}}$. In terms of the objective function, we get

$$\widetilde{A}(\mu, f_*) = \frac{1}{(2\pi r)^d} \int_{\mathbb{R}^d} |\hat{f}_*(\omega)|^2 \Big(1 - \frac{1}{1 + \mu r^d \hat{q}(\omega)^{-1}}\Big) d\omega = \frac{1}{(2\pi r)^d} \int_{\mathbb{R}^d} |\hat{f}_*(\omega)|^2 \frac{\mu r^d}{\hat{q}(\omega) + \mu r^d} d\omega.$$

When μ goes to zero, we see that for each ω, $\hat{f}_\mu(\omega)$ tends to $\hat{f}_*(\omega)$. By the dominated convergence theorem, $\widetilde{A}(\mu, f_*)$ goes to zero when μ 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 (♦). Assume that f_* belongs to the Sobolev space of order t; that is,

$$\frac{1}{(2\pi r)^d} \int_{\mathbb{R}^d} (1 + r^2 \|\omega\|_2^2)^t |\hat{f}_*(\omega)|^2 d\omega < +\infty \tag{7.15}$$

for some $t > 0$ (i.e., f_* with square-integrable partial derivatives up to order t). Then we can further bound $\widetilde{A}(\mu, f_*)$ as follows:

$$\widetilde{A}(\mu, f_*) \quad \leqslant \quad \frac{1}{(2\pi r)^d} \int_{\mathbb{R}^d} (1 + r^2 \|\omega\|_2^2)^t |\hat{f}_*(\omega)|^2 d\omega \times \sup_{\omega \in \mathbb{R}^d} \Big\{ \frac{\mu r^d}{\hat{q}(\omega) + \mu r^d} \frac{1}{(1 + r^2\|\omega\|_2^2)^t} \Big\}.$$

If we now assume $\hat{q}(\omega) \propto r^d (1 + r^2 \|\omega\|_2^2)^{-s}$ (Matern kernels; see section 7.3.3), with $s > d/2$ to get an RKHS. We have two cases:

- When $t \geqslant s$, $f_* \in \mathcal{H}$, and $\widetilde{A}(\mu, f_*) \leqslant \mu \|f_*\|_{\mathcal{H}}^2$.
- When $t < s$ (i.e., the function is not inside the RKHS $\mathcal{H}$), then we get a bound proportional to, using $a + b \geqslant \frac{t}{s} a + (1 - \frac{t}{s}) b \geqslant a^{t/s} b^{1-t/s}$) (proof using Jensen's inequality for the logarithm),

$$\begin{aligned} \widetilde{A}(\mu, f_*) \quad &= \quad O\Big(\sup_{\omega \in \mathbb{R}^d} \Big\{ \frac{\mu r^d}{\hat{q}(\omega) + \mu r^d} \frac{1}{(1 + r^2\|\omega\|_2^2)^t} \Big\} \Big) \\ &= \quad O\Big(\sup_{\omega \in \mathbb{R}^d} \Big\{ \frac{\mu r^d}{\hat{q}(\omega)^{t/s} (\mu r^d)^{1-t/s}} \frac{1}{(1 + r^2\|\omega\|_2^2)^t} \Big\} = O(\mu^{t/s}) \Big). \end{aligned}$$

Exercise 7.21 (♦) *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 constraints: $t \geqslant 0$ for the target function, and $s > d/2$ for the kernel.

Putting things together. For Lipschitz-continuous losses and target functions that satisfy equation (7.15), we get from equations (7.12) and (7.13) 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 $t = 1$ (i.e., 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 a Lipschitz-loss setup, 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 if f is only assumed to be differentiable).

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 $f_* \in \mathcal{H}$ (i.e., $t \geqslant s$), then we obtain a dimension-independent rate of $1/\sqrt{n}$. In intermediate scenarios $t \in [1, s] = [1, d/2 + 1/2]$, the rates $O(n^{-t/(d+1)})$ fall in between. This is why kernel methods are said to be *adaptive to the smoothness* of the target function: for a fixed kernel defined by order s, we get adaptivity to a whole range of regularity order t of the target function.

When we say that in the smooth case, the bounds are independent of the underlying dimension d, we refer to the dependency in terms of powers of n. Yet, the constants in front of these decaying terms may still behave badly in d (e.g., exponential).

Approximation bounds (♦). In some analysis setups (such as those explored in section 9.3.5), it is required to approximate some f_* up to ε with the minimum possible RKHS norm. This can be done as follows.

A bound on the quantity $A(\mu, f_*) = \inf_{f \in \mathcal{H}} \big\{ \|f - f_*\|^2_{L_2(p)} + \mu \|f\|^2_{\mathcal{H}} \big\}$ of the form $c\mu^\alpha$ for $\alpha \in (0, 1)$ leads to the following bound:

$$\begin{aligned}
& \inf_{f \in \mathcal{H}} \|f\|^2_{\mathcal{H}} \text{ such that } \|f - f_*\|_{L_2(p)} \leqslant \varepsilon \\
= \; & \inf_{f \in \mathcal{H}} \sup_{\mu \geqslant 0} \Big\{ \|f\|^2_{\mathcal{H}} + \mu (\|f - f_*\|^2_{L_2(p)} - \varepsilon^2) \Big\} \text{ using Lagrangian duality,} \\
= \; & \sup_{\mu \geqslant 0} \big\{ \mu A(\mu^{-1}, f_*) - \mu \varepsilon^2 \big\} \; \leqslant \; \sup_{\mu \geqslant 0} \big\{ \mu c \mu^{-\alpha} - \mu \varepsilon^2 \big\}.
\end{aligned}$$

The optimal μ 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 $\alpha = t/s$ as before, this leads to an RKHS norm proportional to $\varepsilon^{-(1-\alpha)/\alpha} = \varepsilon^{1-s/t}$ to get an error less than $\|f - f_*\|_{L_2(\mathbb{R}^d)} \leqslant \varepsilon$. 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 (♦♦). In chapter 6, on local averaging methods, as well as in chapter 9, on neural networks, we consider Lipschitz-continuous functions on a subset of $\mathbb{R}^d$, which we take here to be the ball with center 0 and radius r. To apply results from this chapter, we need to extend them to a function g on $\mathbb{R}^d$ with a controlled squared Sobolev norm with order $t = 1$; that is, $\frac{1}{r^d} \int_{\mathbb{R}^d} \big(|g(x)|^2 + r^2 \|g'(x)\|_2^2 \big) dx$. Then, the estimation rates for Sobolev space of order t (i.e., $O(n^{-1/(1+d)})$) applies to Lipschitz-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 $|f(0)| \leqslant rD$ (with a dependence in r that ensures unit homogeneity), and f is D-Lipschitz-continuous on the ball with center 0 and radius r. We now show that we can extend it to function g with a squared Sobolev norm that is less than a constant c_d (which depends on d) times $r^2 D^2$.

We define 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, function g goes linearly from $f(y)$ to 0. Function g is continuous and has bounded derivatives almost everywhere. 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, $\frac{1}{r^d}\int_{\mathbb{R}^d}\big(|g(x)|^2 + r^2\|g'(x)\|_2^2\big)dx \leqslant 9r^2D^2 2^d \frac{\pi^{d/2}}{\Gamma(1+d/2)}$, since the volume of the Euclidean unit ball is equal to $\frac{\pi^{d/2}}{\Gamma(1+d/2)}$. Thus, 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 (i.e., square loss and penalization by squared norm) used within kernel methods. Compared to the analysis performed in section 3.6, there are three difficulties:

- We go from fixed design to random design: This will require finer probabilistic arguments to relate population and empirical covariance operators.
- We need to go infinite-dimensional: In terms of notation, this will mean not using transposes of matrices but rather dot products and tensor products, which is a minor modification.
- 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.

⚠ In this section, since we consider two different Hilbert spaces $L_2(p)$ and $\mathcal{H}$, we will use the notations $\|\cdot\|_{L_2(p)}$ and $\|\cdot\|_{\mathcal{H}}$ for their norms (and similarly for their associated dot products).

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 to minimize, for $\lambda > 0$,

$$\frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

Like the local averaging methods described in chapter 6, the ridge regression estimator happens to be a *linear* estimator that depends linearly on the response vector (but, of course, nonlinearly in x in general). Indeed, using the representer theorem from equation (7.2), the estimator is $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$, with $\alpha \in \mathbb{R}^n$ defined in equation (7.7) 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 (as defined in section 6.2.1) is then symmetric equal to $H = K(K + n\lambda I)^{-1}$.

The key differences with local averaging are that (1) the weights do not sum to 1 (i.e., $\sum_{i=1}^n \hat{w}_i(x)$ may be different from 1); and (2) the weights are not constrained to be nonnegative. While the first difference can be removed using centering (see exercise 7.22), the second is more fundamental: allowing the weights to be negative will enable the adaptivity to smoothness, which local averaging methods missed (see section 6.5). See also section 13.4.3 for the use of ridge regression and linear estimators in the context of structured prediction.

Exercise 7.22 *Consider the optimization problem* $\min_{\theta,\eta} \frac{1}{2n}\|y - \Phi\theta - \eta 1_n\|_2^2 + \frac{\lambda}{2}\|\theta\|_2^2$ *in the variables* $\theta \in \mathbb{R}^d$ *and* $\eta \in \mathbb{R}$*, where* $\Phi \in \mathbb{R}^{n\times d}$ *is the design matrix obtained from feature map* φ *and data points* $x_1, \dots, x_n$*,* $y \in \mathbb{R}^n$*, and* $1_n \in \mathbb{R}^n$ *is the vector of all* 1*s. Show that the optimal values of* θ *and* η *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* $f(x) = \varphi(x)^\top\theta + \eta$ *takes the form* $\sum_{i=1}^n \hat{w}_i(x) y_i$ *with weights that sum to* 1*.*

Exercise 7.23 (♦) *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 to be deterministic) and a finite-dimensional feature space $\mathcal{H}$, and obtained in proposition 3.7 the following *exact* expression of the excess risk of the ridge regression estimator $\hat{\theta}_\lambda$, assuming that $y_i = \langle \theta_*, \varphi(x_i)\rangle + \varepsilon_i$, with ε_i independent of x_i and where $\mathbb{E}[\varepsilon_i] = 0$, $\mathbb{E}[\varepsilon_i^2] = \sigma^2$:

$$\mathbb{E}\big[(\hat{\theta}_\lambda - \theta_*)^\top \widehat{\Sigma}(\hat{\theta}_\lambda - \theta_*)\big] = \lambda^2 \theta_*^\top (\widehat{\Sigma} + \lambda I)^{-2}\widehat{\Sigma}\theta_* + \frac{\sigma^2}{n}\operatorname{tr}\big[\widehat{\Sigma}^2(\widehat{\Sigma} + \lambda I)^{-2}\big]. \quad (7.16)$$

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 (in other words, the minimizer is only in the closure of $\mathcal{H}$).

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 given 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 following optimization problem:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2, \tag{7.17}$$

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 and commonly used in algorithms in section 7.4.

We have, with $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} \varphi(x_i) \otimes \varphi(x_i)$,[18] a self-adjoint operator from $\mathcal{H}$ to $\mathcal{H}$ (the empirical covariance operator), a quadratic cost function in equation (7.17) 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 equation (7.17), 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 $\mathbb{E}\big[\|\hat{f}_\lambda - f_*\|_{L_2(p)}^2\big]$ as (using that $\mathbb{E}[\varepsilon_i|x_i] = 0$)

$$\begin{aligned} & \mathbb{E}\big[\|\hat{f}_\lambda - f_*\|_{L_2(p)}^2\big] \\ = \ & \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i \varphi(x_i)\Big\|_{L_2(p)}^2\Big] + \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^{n} f_*(x_i)\varphi(x_i) - f_*\Big\|_{L_2(p)}^2\Big]. \end{aligned}$$

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 noncentered 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 $\mathcal{H}$),

$$\Sigma = \mathbb{E}\big[\varphi(x) \otimes \varphi(x)\big],$$

[18]For $a, b \in \mathcal{H}$, $a \otimes b$ is the operator from $\mathcal{H}$ to $\mathcal{H}$ defined through $(a \otimes b)f = \langle b, f \rangle_{\mathcal{H}} a$.

for the corresponding distribution of the x_i's. A key property relates the $L_2(p)$-norm and the RKHS norm; that is, for $g \in \mathcal{H}$,

$$\begin{aligned}\|g\|^2_{L_2(p)} &= \int_{\mathcal{X}} g(x)^2 dp(x) = \int_{\mathcal{X}} \langle g, \varphi(x)\rangle^2 dp(x) = \int_{\mathcal{X}} \langle g, \varphi(x) \otimes \varphi(x) g\rangle dp(x) \\ &= \langle g, \Sigma g\rangle = \|\Sigma^{1/2} g\|^2_{\mathcal{H}}. \end{aligned} \tag{7.18}$$

More generally, we have $\int_{\mathcal{X}} f(x)g(x)dp(x) = \langle f \Sigma g\rangle_{\mathcal{H}}$ for all $f, g \in \mathcal{H}$.

Variance term. The variance term can be upper-bounded as follows (first using independence and zero means of the variables ε_i), then using 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 equation (7.18)):

$$\begin{aligned}\text{variance} &= \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i)\Big\|^2_{L_2(p)}\Big] \\ &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}\Big[\mathrm{tr}\,\big((\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} \varepsilon_i^2 \varphi(x_i) \otimes \varphi(x_i)\big)\Big] \\ &\leqslant \frac{\sigma^2}{n} \mathbb{E}\big[\mathrm{tr}\,\big((\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma}\big)\big] \text{ using } \mathbb{E}[\varepsilon_i^2 | x_i] \leqslant \sigma^2, \\ &\leqslant \frac{\sigma^2}{n} \mathbb{E}\Big[\mathrm{tr}\,\big[(\widehat{\Sigma} + \lambda I)^{-1} \Sigma\big]\Big] \text{ using } (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} \preccurlyeq I. \end{aligned} \tag{7.19}$$

This will be the main expression that we will bound later in proposition 7.5.

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

$$\text{bias} = \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n f_*(x_i) \varphi(x_i) - f_*\Big\|^2_{L_2(p)}\Big] \tag{7.20}$$

$$= \mathbb{E}\Big[\Big\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \langle f_*, \varphi(x_i)\rangle \varphi(x_i) - f_*\Big\|^2_{L_2(p)}\Big] \tag{7.21}$$

$$= \mathbb{E}\big[\|(\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} f_* - f_*\|^2_{L_2(p)}\big] \text{ using the expression of } \widehat{\Sigma},$$

$$= \mathbb{E}\Big[\|\lambda \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} f_*\|^2_{\mathcal{H}}\Big] = \lambda^2 \mathbb{E}\Big[\langle f_*, (\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} f_*\rangle\Big], \tag{7.22}$$

where we have used equation (7.18) above to reintroduce the operator Σ. This will be the main expression we will bound in proposition 7.5.

Upper bound on excess risk. Combining equations (7.19) and (7.22), we have thus proved proposition 7.5.

Proposition 7.5 (Upper bound on expected risk–well-specified problem) *When $f_* \in \mathcal{H}$, the excess risk of the ridge regression estimator is upper-bounded by*

$$\mathbb{E}\big[\|\hat{f}_\lambda - f_*\|^2_{L_2(p)}\big] \leqslant \frac{\sigma^2}{n}\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\Big] + \lambda^2\mathbb{E}\Big[\langle f_*, (\widehat{\Sigma}+\lambda I)^{-1}\Sigma(\widehat{\Sigma}+\lambda I)^{-1} f_*\rangle\Big]. \tag{7.23}$$

Given the expression of the expected variance in equation (7.19) and the expected bias in equation (7.22), 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

Following Mourtada and Rosasco (2022), we derive simple relationships between the empirical covariance operator $\widehat{\Sigma}$ and the population operator Σ by showing lemma 7.1, dealing with expectations; for high probability bounds, see, for example, Rudi et al. (2015), Rudi and Rosasco (2017), and the end of section 7.6.4 of this book.

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 $g \in \mathcal{H}$,*

$$\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\Big] \leqslant \Big(1+\frac{R^2}{\lambda n}\Big)\operatorname{tr}\big((\Sigma+\lambda I)^{-1}\Sigma\big) \tag{7.24}$$

$$\mathbb{E}\Big[\big\langle g, (\widehat{\Sigma}+\lambda I)^{-1}\Sigma(\widehat{\Sigma}+\lambda I)^{-1} g\big\rangle\Big] \preccurlyeq \lambda^{-1}\Big(1+\frac{R^2}{\lambda n}\Big)^2\langle g, (\Sigma+\lambda I)^{-1}\Sigma g\rangle. \tag{7.25}$$

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 $C = \sum_{i=1}^{n+1} \varphi(x_i) \otimes \varphi(x_i) = n\widehat{\Sigma} + \varphi(x_{n+1}) \otimes \varphi(x_{n+1})$, and using the matrix inversion lemma (section 1.1.3), we have

$$\begin{aligned}(C+n\lambda I)^{-1}\varphi(x_{n+1}) &= \big(n\widehat{\Sigma} + n\lambda I + \varphi(x_{n+1}) \otimes \varphi(x_{n+1})\big)^{-1}\varphi(x_{n+1}) \\ &= \frac{(n\widehat{\Sigma}+n\lambda I)^{-1}\varphi(x_{n+1})}{1+\langle\varphi(x_{n+1}), (n\widehat{\Sigma}+n\lambda I)^{-1}\varphi(x_{n+1})\rangle}.\end{aligned} \tag{7.26}$$

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 equation (7.24), we use equation (7.26) to express $(\widehat{\Sigma}+\lambda I)^{-1}\varphi(x_{n+1})$ as $n(1+c)(C+n\lambda I)^{-1}\varphi(x_{n+1})$:

$$\begin{aligned}
\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\Big] &= \mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\varphi(x_{n+1})\otimes\varphi(x_{n+1})\big)\Big] \\
&= \mathbb{E}\Big[\big\langle\varphi(x_{n+1}),(\widehat{\Sigma}+\lambda I)^{-1}\varphi(x_{n+1})\big\rangle\Big] \\
&= n\mathbb{E}\Big[(1+c)\big\langle\varphi(x_{n+1}),(C+n\lambda I)^{-1}\varphi(x_{n+1})\big\rangle\Big],
\end{aligned}$$

which leads to $\mathbb{E}\big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\big]\leqslant n\big(1+\frac{R^2}{\lambda n}\big)\mathbb{E}\big[\big\langle\varphi(x_{n+1}),(C+n\lambda I)^{-1}\varphi(x_{n+1})\big\rangle\big]$. Thus, using that the variables $(x_1,\dots,x_{n+1})$ are exchangeable, we get:

$$\begin{aligned}
&\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\Big] \\
\leqslant\ & \Big(1+\frac{R^2}{\lambda n}\Big)n\times\frac{1}{n+1}\sum_{i=1}^{n+1}\mathbb{E}\big[\big\langle\varphi(x_i),\big(C+n\lambda I\big)^{-1}\varphi(x_i)\big\rangle\big] \\
=\ & \Big(1+\frac{R^2}{\lambda n}\Big)\frac{n}{n+1}\mathbb{E}\big[\operatorname{tr}\big(C(C+n\lambda I)^{-1}\big)\big] \text{ since } C=\sum_{i=1}^{n+1}\varphi(x_i)\otimes\varphi(x_i).
\end{aligned}$$

Using Jensen's inequality with the convex function $C\mapsto\operatorname{tr}[C(C+n\lambda I)^{-1}]$, we get

$$\begin{aligned}
&\mathbb{E}\Big[\operatorname{tr}\big((\widehat{\Sigma}+\lambda I)^{-1}\Sigma\big)\Big] \\
\leqslant\ & \Big(1+\frac{R^2}{\lambda n}\Big)\frac{n}{n+1}\big[\operatorname{tr}\big(\mathbb{E}[C](\mathbb{E}[C]+n\lambda I)^{-1}\big)\big] \\
=\ & \Big(1+\frac{R^2}{\lambda n}\Big)\frac{n}{n+1}\operatorname{tr}\big((n+1)\Sigma((n+1)\Sigma+n\lambda I)^{-1}\big) \text{ by definition of } \Sigma, \\
\leqslant\ & \Big(1+\frac{R^2}{\lambda n}\Big)\operatorname{tr}\big(\Sigma(\Sigma+\lambda I)^{-1}\big), \text{ which is exactly as in equation (7.24).}
\end{aligned}$$

To prove equation (7.25), we use the same technique; that is,

$$\begin{aligned}
&\mathbb{E}\big[(\widehat{\Sigma}+\lambda I)^{-1}\Sigma(\widehat{\Sigma}+\lambda I)^{-1}\big] = \mathbb{E}\Big[(\widehat{\Sigma}+\lambda I)^{-1}\varphi(x_{n+1})\otimes\varphi(x_{n+1})(\widehat{\Sigma}+\lambda I)^{-1}\Big] \\
=\ & n^2(1+c)^2\mathbb{E}\Big[\big[(C+n\lambda I)^{-1}\varphi(x_{n+1})\big]\otimes\big[(C+n\lambda I)^{-1}\varphi(x_{n+1})\big]\Big].
\end{aligned}$$

This leads to

$$
\begin{aligned}
& \mathbb{E}\big[\langle g, (\widehat{\Sigma}+\lambda I)^{-1}\Sigma(\widehat{\Sigma}+\lambda I)^{-1} g\rangle\big] \\
= \ & n^2 \mathbb{E}\Big[(1+c)^2 \langle (C+n\lambda I)^{-1}\varphi(x_{n+1}), g\rangle^2\Big] \\
\leqslant \ & n^2 \Big(1+\frac{R^2}{\lambda n}\Big)^2 \mathbb{E}\Big[\langle (C+n\lambda I)^{-1}\varphi(x_{n+1}), g\rangle^2\Big] \\
= \ & \frac{n^2}{n+1}\Big(1+\frac{R^2}{\lambda n}\Big)^2 \mathbb{E}\Big[\langle g, (C+n\lambda I)^{-1} C (C+n\lambda I)^{-1} g\rangle\Big] \text{ by exchangeability,} \\
\leqslant \ & \frac{1}{\lambda}\frac{n}{n+1}\Big(1+\frac{R^2}{\lambda n}\Big)^2 \mathbb{E}\Big[\langle g, C(C+n\lambda I)^{-1} g\rangle\Big] \text{ using } (C+n\lambda I)^{-2}C \preccurlyeq \frac{1}{n\lambda}(C+n\lambda I)^{-1}C, \\
\leqslant \ & \frac{1}{\lambda}\frac{n}{n+1}\Big(1+\frac{R^2}{\lambda n}\Big)^2 \langle g, \mathbb{E}[C](\mathbb{E}[C]+n\lambda I)^{-1} g\rangle \text{ by Jensen's inequality,} \\
= \ & \frac{1}{\lambda} n \Big(1+\frac{R^2}{\lambda n}\Big)^2 \langle g, \Sigma((n+1)\Sigma + n\lambda I)^{-1} g\rangle \leqslant \lambda^{-1}\Big(1+\frac{R^2}{\lambda n}\Big)^2 \langle g, (\Sigma+\lambda I)^{-1}\Sigma g\rangle.
\end{aligned}
$$

■

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 equation (7.23).

Proposition 7.6 (Kernel ridge regression–well-specified model) *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 $\mathbb{E}[\varepsilon_i|x_i] = 0$, $\mathbb{E}[\varepsilon_i^2|x_i] \leqslant \sigma^2$, and $f_* \in \mathcal{H}$. Assume that $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ almost surely. We have*

$$
\mathbb{E}\big[\|\hat{f}_\lambda - f_*\|^2_{L_2(p)}\big] \leqslant \frac{\sigma^2}{n}\Big(1+\frac{R^2}{\lambda n}\Big) \operatorname{tr}\big[(\Sigma+\lambda I)^{-1}\Sigma\big] + \lambda\Big(1+\frac{R^2}{\lambda n}\Big)^2 \langle f_*, \Sigma(\Sigma+\lambda I)^{-1} f_*\rangle. \tag{7.27}
$$

This is to be contrasted with equation (7.16): we obtain a similar result with $\widehat{\Sigma}$ being replaced by Σ, but with some extra multiplicative constants that are close to 1 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 proposition 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 $(\sigma^2 + R^2\|f_*\|^2_{\mathcal{H}})/\sqrt{n}$, with a behavior similar to Lipschitz-continuous losses in section 7.5.

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 λ of order R^2/n and an upper bound of the excess risk proportional to $\sigma^2 d/n + R^2\|f_*\|^2_{\mathcal{H}}/n$, with an improved behavior compared to

Lipschitz-continuous losses. Going beyond finite dimension requires the notion of "degrees of freedom" introduced in section 7.6.6.

These 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 thumb within kernel methods. Note, however, that this corresponds only to well-specified models, and that in practice, in particular for kernels corresponding to very smooth functions (such as the Gaussian kernel), smaller regularization parameters are required; see more details in section 7.6.6.

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 here for the simplest bound; see more refined bounds by Rudi et al. (2015) and Rudi and Rosasco (2017). Note that they do not rely on Rademacher averages but on direct probabilistic arguments that can be applied only 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 $\mathbb{E}[\varepsilon_i|x_i] = 0$, $\varepsilon_i^2 \leqslant \sigma^2$ almost surely, and $f_* \in \mathcal{H}$. Assume that $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ almost surely and $n \geqslant \big(\frac{4}{3} + \frac{8R^2}{\lambda}\big) \log \frac{14R^2}{\lambda\delta}$. We have, with a probability greater than $1 - \delta$,*

$$\|\hat{f}_\lambda - f_*\|^2_{L_2(p)} \leqslant \frac{8\sigma^2 R^2}{\lambda n} + 4\lambda \|f_*\|^2_{\mathcal{H}} + \frac{16\sigma^2 R^2}{\lambda n} \log \frac{2}{\delta}. \tag{7.28}$$

Proof We first apply proposition 1.8 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 $t = \frac{1}{2}$, 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 $1 - 7\frac{R^2}{\lambda} \exp\big[-\frac{n}{4/3 + 8R^2/\lambda}\big]$, 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 $n \geqslant (4/3 + 8R^2/\lambda) \log \frac{14R^2}{\lambda\delta}$.

If this event is true, 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's 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 $\widehat{\mathcal{R}}_\lambda(f_*) - \widehat{\mathcal{R}}_\lambda(\hat{f}_\lambda) = \frac{1}{2}\langle f_* - \hat{f}_\lambda, (\widehat{\Sigma} + \lambda I)(f_* - \hat{f}_\lambda)\rangle \geqslant \frac{1}{4}\langle f_* - \hat{f}_\lambda, (\Sigma + \lambda I)(f_* - \hat{f}_\lambda)\rangle \geqslant \frac{1}{4}\|\hat{f}_\lambda - f_*\|^2_{L_2(p)}$, 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 follows (see exercise 1.20):

$$\Big\| \frac{1}{n}\sum_{i=1}^n \varepsilon_i \varphi(x_i)\Big\|_{\mathcal{H}} \leqslant \frac{R\sigma}{\sqrt{n}}\Big(1+\sqrt{2\log\frac{2}{\delta}}\Big).$$

This leads to the desired result. ■

Before analyzing proposition 7.7 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 equation (7.27), 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 simple lemma 7.2 is the key to extending to all functions f_* in the closure of $\mathcal{H}$.

Lemma 7.2 *Given the covariance operator Σ and any function $f_* \in \mathcal{H}$, then*

$$\lambda\langle f_*, (\Sigma+\lambda I)^{-1}\Sigma f_*\rangle = \inf_{f\in\mathcal{H}} \Big\{ \|f-f_*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}}\Big\}. \tag{7.29}$$

Proof The optimization problem in equation (7.29) can be written using equation (7.18) as $\inf_{f\in\mathcal{H}}\big\{\|\Sigma^{1/2}(f-f_*)\|^2_{\mathcal{H}} + \lambda\|f\|^2_{\mathcal{H}}\big\}$, 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 proposition 7.6 to the general case of $f_* \in L_2(p)$ 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 a bias term less than

$$\Big(1+\frac{R^2}{\lambda n}\Big)^2 \inf_{f\in\mathcal{H}} \Big\{ \|f-f_*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}}\Big\}. \tag{7.30}$$

For translation-invariant kernels in $\mathbb{R}^d$ (which are dense in $L_2(\mathbb{R}^d)$), this allows for estimating any target function.

Final result. Using lemma 7.2 and equation (7.30) with proposition 7.6, we can now show the upper bound for kernel ridge regression in the potentially misspecified case.

Proposition 7.8 (Kernel ridge regression–misspecified model) *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 that $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ almost surely and f_* in the closure of $\mathcal{H}$ in*

$L_2(p)$. *We have*

$$\mathbb{E}\big[\|\hat{f}_\lambda - f_*\|^2_{L_2(p)}\big] \leqslant \frac{\sigma^2}{n}\Big(1+\frac{R^2}{\lambda n}\Big) \operatorname{tr}\big((\Sigma+\lambda I)^{-1}\Sigma\big) + \Big(1+\frac{R^2}{\lambda n}\Big)^2 \inf_{f\in\mathcal{H}} \Big\{\|f-f_*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}}\Big\}. \tag{7.31}$$

Be careful with the unit homogeneity of formulas; for example, $\frac{R^2}{\lambda n}$ 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 obtained from proposition 7.8:

$$\frac{\sigma^2}{n}\Big(1+\frac{R^2}{\lambda n}\Big) \operatorname{tr}\big((\Sigma+\lambda I)^{-1}\Sigma\big) + \Big(1+\frac{R^2}{\lambda n}\Big)^2 \inf_{f\in\mathcal{H}} \Big\{\|f-f_*\|^2_{L_2(p)} + \lambda\|f\|^2_{\mathcal{H}}\Big\}.$$

For this section, we will assume that $\mathcal{X} = \mathbb{R}^d$ and 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 $s \geqslant t$ (which we now assume). For the variance term, we need to study the so-called "degrees of freedom," associated to the covariance operator Σ.

Eigendecomposition of the covariance operator. The covariance operator defined as $\Sigma = \mathbb{E}\big[\varphi(x)\otimes\varphi(x)\big]$ is a linear operator from $\mathcal{H}$ to $\mathcal{H}$. When $\|\varphi(x)\|_{\mathcal{H}} \leqslant R$ almost surely, it is said to be "trace-class"[19] as its trace can be defined and bounded by $\mathbb{E}[\|\varphi(x))\|^2_{\mathcal{H}}] \leqslant R^2$. There thus exists a sequence of eigenfunctions $(f_m)_{m\geqslant 0}$ in $\mathcal{H}$ and a summable nonincreasing sequence of nonnegative eigenvalues $(\lambda_m)_{m\geqslant 0}$ such that

$$\Sigma = \sum_{m\geqslant 0} \lambda_m f_m \otimes f_m.$$

For kernels such as translation-invariant kernels on $[0,1]$ from section 7.3.2, where $k(x,x')$ is of the form $k(x,x') = \sum_{m\geqslant 0}\lambda_m g_m(x) g_m(x')$ for an orthonormal basis $(g_m)_{m\geqslant 0}$ of $L_2(p)$, the two sequences of eigenvalues coincide, and the eigenfunctions are equal up to normalization constants (as normalized eigenvectors in different spaces). This is true more generally and can be shown using the integral operator defined in exercise 7.7; see more details in Cucker and Smale (2002) and Steinwart and Scovel (2012).

Degrees of freedom. This is the quantity $\operatorname{tr}\big[\Sigma(\Sigma+\lambda I)^{-1}\big]$, which is decreasing in λ, 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

$$\lambda_m \leqslant C(m+1)^{-\alpha},$$

[19] See https://en.wikipedia.org/wiki/Trace_class.

for $\alpha > 1$, then one has, with the change of variable $u = \lambda C^{-1} t^\alpha$,

$$\begin{aligned}\operatorname{tr}\left[\Sigma(\Sigma+\lambda I)^{-1}\right] &= \sum_{m\geqslant 0}\frac{\lambda_m}{\lambda_m+\lambda} \leqslant \sum_{m\geqslant 0}\frac{1}{1+\lambda C^{-1}(m+1)^\alpha} \leqslant \int_0^\infty \frac{dt}{1+\lambda C^{-1}t^\alpha} \\ &\leqslant \int_0^\infty \lambda^{-1/\alpha} C^{1/\alpha}\frac{1}{\alpha}u^{1/\alpha-1}\frac{du}{1+u} = O(\lambda^{-1/\alpha}).\end{aligned}$$

For periodic Sobolev spaces of order s defined in section 7.3.2, the eigenvalues are exactly proportional to m^{-2s} since the eigenvalue decomposition is explicit. 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 bias $\lambda^{t/s}$ with variance $\frac{1}{n}\lambda^{-d/(2s)}$, leading to an optimal λ proportional to $n^{-(d/(2s)+t/s)^{-1}} = n^{-2s/(2t+d)}$, and a rate proportional to $n^{-2t/(2t+d)}$. This rate is achievable only through our analysis when $\frac{R^2}{n\lambda}$ remains bounded (i.e., essentially $\lambda \geqslant R^2/n$), thus, $2s/(2t+d) \leqslant 1$. On top of the constraint that $d/2 < s \leqslant t$ that we assumed earlier, we get the rate $n^{-2t/(2t+d)}$ so 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 λ does not depend on the kernel.
- We obtain some form of adaptivity (i.e., the rate improves with the regularity of the target function): we get the slow rate $n^{-2/(2+d)}$ when $t = 1$ (recovering the same rate as for local averaging methods[20] in chapter 6), and that can be achieved only when $s \leqslant d/2+1$ (e.g., with the exponential kernel, and then with a regularization parameter smaller than $1/\sqrt{n}$). At the same time, we can get the rate $n^{-2s/(2s+d)} = n^{-2t/(2t+d)}$ when $t = s$ (well-specified model); then, the rate is always better than $1/\sqrt{n}$, because of the constraint $s > d/2$, and can be as good as $1/n$ when t (and thus s) is large. The rate for the square loss is then significantly better than the rate $1/\sqrt{n}$ we obtained for Lipschitz-continuous losses.
- To allow for regularization parameters λ that are less than $1/n$ (and then even more adaptivity, as kernels with fast decay of eigenvalues, such as the Gaussian kernel, lead to good estimation rates for most target functions), further assumptions are needed. See, for instance, Pillaud-Vivien et al. (2018) and references therein.

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 nonsmooth target, and three kernels: an exponential kernel corresponding to the Sobolev space of order 1 (top of

[20] 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 shown at the end of section 7.5.2.

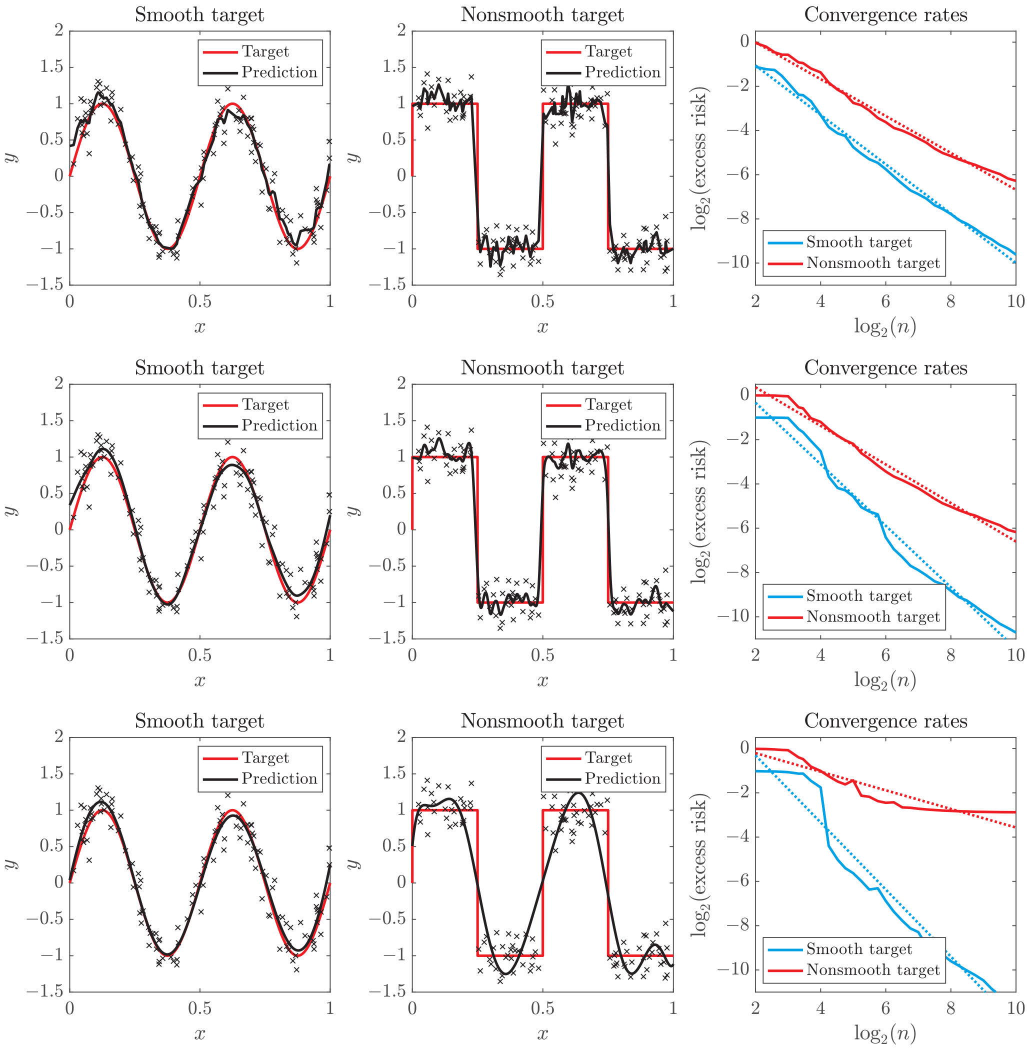


Figure 7.3. Comparison of three kernels; Sobolev space of order 1 (top), Matern kernel corresponding to the Sobolev space of order 3 (middle), and Gaussian kernel (bottom). We consider two different target functions and represent on the right plots the excess risks in logarithmic scale.

figure 7.3), a Matern kernel corresponding to the Sobolev space of order 3 (middle), and a 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 the results in figure 7.3.

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 kernels), the performance on the nonsmooth target function is worse than for the large feature space (exponential kernel). As highlighted by Bach (2013), this drop in performance for the Gaussian kernel 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 the 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 their adaptivity to the smoothness of the prediction function. What is still missing is adaptivity to problems where the optimal prediction function depends only 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 being able to solve nonconvex optimization problems.

Chapter 8

Sparse Methods

Chapter Summary

- Model selection through regularization: Model selection can be performed by adding a specific sparsity-inducing penalty on top of the empirical risk.
- ℓ_0-penalty: For fixed design linear regression, if the optimal predictor has k nonzeros, then we can replace the rate $\frac{\sigma^2 d}{n}$ by $\frac{\sigma^2 k \log d}{n}$ with an ℓ_0-penalty on the square loss (which is computationally hard).
- ℓ_1-penalty: With few assumptions, we can get a slow rate proportional to $k\sqrt{\frac{\log d}{n}}$ with an ℓ_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 ℓ_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 chapters 3 and 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 variables to be selected is not known in advance.

In practice, variable selection is mainly used in two ways:

- The original set of features is already large (e.g., 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 (see experiments in section 8.4).

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

$$\mathbb{E}\big[\ell(y, \varphi(x)^\top \theta)\big]$$

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 $\|\theta\|_0$, which is the number of nonzeros in θ (often miscalled "ℓ_0-norm"), and the ℓ_1-norm. See extensions to more structured situations in section 8.5.

Nonlinear 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 nonlinear 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 to be deterministic and ε_i has zero mean and variance σ^2 (we also assume independence from x_i and sometimes stronger regularity, such as almost-sure boundedness or Gaussian distributions). 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 (OLS) 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 θ_*. In this chapter, we assume that θ_* is sparse; that is, only a few of its components are nonzero, 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 as $\hat\theta$ a minimizer of $\frac{1}{n}\|y - \Phi\theta\|_2^2$ with the constraint that $\theta \in \Theta$, for some subset Θ 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 factor out $\|\Phi(\hat\theta - \theta_*)\|_2$ and write it as

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 2\|\Phi(\hat\theta - \theta_*)\|_2 \cdot \varepsilon^\top\left(\frac{\Phi(\hat\theta - \theta_*)}{\|\Phi(\hat\theta - \theta_*)\|_2}\right).$$

This reformulation is difficult to deal with because $\hat\theta$ also appears on the right side of the equation. As 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

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 2\|\Phi(\hat\theta - \theta_*)\|_2 \cdot \sup_{\theta\in\Theta} \varepsilon^\top\left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\right), \tag{8.1}$$

where $\hat\theta$ has disappeared from the right side. Finally, isolating $\|\Phi(\hat\theta - \theta_*)\|_2^2$, we get

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 4 \sup_{\theta\in\Theta}\left[\varepsilon^\top\left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\right)\right]^2. \tag{8.2}$$

This inequality is true almost surely, and we can take expectation (with respect to ε) to obtain bounds. Therefore, in this chapter, we will compute expectations of maxima of

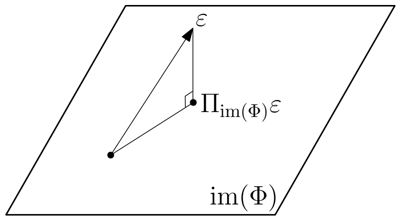
[1] This is taken from Philippe Rigollet's lecture notes; see https://math.mit.edu/~rigollet/. See also Rigollet and Tsybakov (2007) for an example of application.

quadratic forms in ε. Note that a key feature is that the set of $\frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}$ for $\theta \in \Theta$ is included in the unit ℓ_2-sphere.

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 (shown here),



we have

$$\sup_{z\in \mathrm{im}(\Phi), \|z\|_2=1} \big[\varepsilon^\top z\big]^2 = \sup_{z\in \mathrm{im}(\Phi), \|z\|_2=1} \big[(\Pi_\Phi\varepsilon)^\top z\big]^2 = \|\Pi_\Phi \varepsilon\|_2^2,$$

leading to

$$\mathbb{E}\big[\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \leqslant 4\mathbb{E}\big[\|\Pi_\Phi\varepsilon\|_2^2\big] = 4\,\mathrm{tr}\left(\Pi_\Phi^2 \mathbb{E}[\varepsilon\varepsilon^\top]\right) = 4\sigma^2\mathbb{E}\,\mathrm{tr}(\Pi_\Phi^2) = 4\sigma^2\mathrm{rank}(\Phi).$$

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 (proposition 3.5) but allows extensions to more complex situations.

This reasoning also allows getting high-probability bounds by adding assumptions to the noise ε. Finally, this also extends to penalized problems (see section 8.2.2).

8.1.2 Probabilistic and Combinatorial Lemmas

In the proof technique described in section 8.1.1, 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 σ), and $s < 1/2$ (using independence among the components of z),

$$\begin{aligned}
\mathbb{E}\big[e^{s\|z\|_2^2}\big] &= \mathbb{E}\big[e^{s\sum_{i=1}^n z_i^2}\big] = \prod_{i=1}^n \mathbb{E}\big[e^{s z_i^2}\big] = \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^n \int_{-\infty}^{\infty} e^{(s-\frac{1}{2})z_i^2} dz_i \\
&= \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^n \sqrt{2\pi}(1-2s)^{-1/2} = (1-2s)^{-n/2}.
\end{aligned}$$

■

Lemma 8.2 *Let $u_1, \dots, u_m$ be m random variables that are potentially dependent, and $s > 0$. Then* $\mathbb{E}\big[\max\{u_1,\dots,u_m\}\big] \leqslant \frac{1}{s}\log\big(\sum_{i=1}^m \mathbb{E}\big[e^{su_i}\big]\big)$.

Proof Following the reasoning from section 1.2.4 in chapter 1, for any $s > 0$,

$$\mathbb{E}\big[\max\{u_1,\dots,u_m\}\big] \leqslant \frac{1}{s}\log\big(\mathbb{E}\big[e^{s\max\{u_1,\dots,u_m\}}\big]\big) = \frac{1}{s}\log\big(\mathbb{E}\big[\max\{e^{su_1},\dots,e^{su_m}\}\big]\big),$$

which is thus less than $\frac{1}{s}\log\big(\sum_{i=1}^m \mathbb{E}\big[e^{su_i}\big]\big)$. ■

Lemmas 8.1 and 8.2 can be combined to upper-bound the expectated squared norms of Gaussian random variables: if $z_1,\dots,z_m \in \mathbb{R}^n$ are centered (i.e., zero mean) Gaussian random vectors that are potentially dependent, but for which the covariance matrix of z_i has eigenvalues less than σ^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 entries $\sigma_{ij}^2 \leqslant \sigma^2$ (for $i \in \{1,\dots,m\}$ and $j \in \{1,\dots,n\}$). Then we have from lemma 8.1,

$$\mathbb{E}[e^{s\|z_i\|_2^2}] = \prod_{j=1}^n \mathbb{E}[e^{sz_{ij}^2}] = \prod_{j=1}^n (1-2\sigma_{ij}^2 s)^{-1/2} \leqslant (1-2\sigma^2 s)^{-n/2}.$$

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,

$$\mathbb{E}\big[\max\{\|z_1\|_2^2,\dots,\|z_m\|_2^2\}\big] \leqslant 4\sigma^2\log(m2^{n/2}) = 2n\sigma^2\log(2) + 4\sigma^2\log(m),$$

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 ℓ_0-penalty. The term in d^k comes from lemma 8.3.

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 used for $\alpha > 0$, $(1+\frac{1}{\alpha})^\alpha = \exp(\alpha\log(1+1/\alpha)) \leqslant \exp(1) = e$. ■

Exercise 8.1 (Concentration of chi-squared variables) *Consider n independent standard Gaussian 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 $\varepsilon \in (0,1)$, $\mathbb{P}(y \leqslant n(1-\varepsilon)) \leqslant \big(\frac{1-\varepsilon}{\exp(-\varepsilon)}\big)^{n/2}$.*

We now consider two types of variable selection frameworks, one based on ℓ_0-penalties and one based on ℓ_1-penalties.

8.2 Variable Selection by the ℓ_0-penalty

In this section, we assume that the target vector θ_* has at most k nonzero components (i.e., $\|\theta_*\|_0 \leqslant k$). We denote by $A = \text{supp}(\theta_*)$ the "support" of θ_*; that is, the subset of $\{1, \ldots, d\}$ composed of j such that $(\theta_*)_j \neq 0$. We have $|A| \leqslant k$.

Price of adaptivity. If we knew set A, then we could simply perform least-squares regression with the design matrix $\Phi_A \in \mathbb{R}^{n\times|A|}$, where Φ_B denotes the submatrix of Φ 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, so long as k is small compared to n, we can estimate θ_* correctly, regardless of the potentially large value of d.

However, we do not know A in advance, and we would still like to have a convergence rate of the order $\sigma^2 k/n$, which is a form of adaptivity to potentially sparse predictors. 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.

Note that we could also apply the general model selection framework of structural risk minimization from section 4.6.1, which would be adapted to Lipschitz-continuous losses. As studied in exercise 8.3, model selection leads to an extra term in $\sqrt{k\log(d)/n}$.

8.2.1 Assuming That k Is Known

We start by assuming that the maximal cardinality k is known in advance, and we consider Gaussian noise for simplicity (this extends to sub-Gaussian noise as well; see the note below the proof of proposition 8.1).

Proposition 8.1 (Model selection–known k) *Assume that $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ being a vector with independent Gaussian components of zero mean and variance σ^2, with $\|\theta_*\|_0 \leqslant k$, for $k < d/2$. 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 θ 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,

$$\begin{aligned}
\|\Phi(\hat\theta - \theta_*)\|_2^2 &\leqslant 4 \sup_{\theta\in\mathbb{R}^d, \|\theta\|_0\leqslant k}\Big[\varepsilon^\top\Big(\frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}\Big)\Big]^2 \text{ from equation (8.2)},\\
&\leqslant 4 \sup_{\theta\in\mathbb{R}^d, \|\theta-\theta_*\|_0\leqslant 2k}\Big[\varepsilon^\top\Big(\frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}\Big)\Big]^2 \text{ from the discussion above},\\
&= 4 \sup_{B\subset\{1,\ldots,d\},\ |B|\leqslant 2k}\ \sup_{\text{supp}(\theta-\theta_*)\subset B}\Big[\varepsilon^\top\Big(\frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}\Big)\Big]^2
\end{aligned}$$

by separating by the supports. Thus, using the same argument as in section 8.1.1,

$$\begin{aligned}\|\Phi(\hat{\theta}-\theta_*)\|_2^2 &\leqslant 4 \sup_{B\subset\{1,\dots,d\},\ |B|\leqslant 2k}\ \sup_{z\in \mathrm{im}(\Phi_B),\|z\|_2=1}\big[\varepsilon^\top z\big]^2 \\ &\leqslant 4 \sup_{B\subset\{1,\dots,d\},\ |B|\leqslant 2k} \|\Pi_{\Phi_B}\varepsilon\|_2^2 \leqslant 4 \sup_{B\subset\{1,\dots,d\},\ |B|=2k} \|\Pi_{\Phi_B}\varepsilon\|_2^2\end{aligned}$$

because $\|\Pi_{\Phi_B}\varepsilon\|_2^2$ is nondecreasing in B.

The random variable $\|\Pi_{\Phi_B}\varepsilon\|_2^2$ has expectation at most $2k$. Given that there are $\binom{d}{2k} \leqslant \big(\frac{ed}{2k}\big)^{2k}$ sets B of cardinality $2k$ (bound from lemma 8.3), we should expect, with concentration inequalities from section 8.1.2, to pay a price of $\log\big[\big(\frac{ed}{2k}\big)^{2k}\big] \approx k\log\frac{d}{k}$. We will make this reasoning formal here.

Indeed, $\Pi_{\Phi_B}\varepsilon$ has a Gaussian distribution with an isotropic covariance matrix of dimension $|B| \leqslant 2k$, and thus we have for $s\sigma^2 < 1/2$, from lemma 8.1,

$$\mathbb{E}\big[e^{s\|\Pi_{\Phi_B}\varepsilon\|_2^2}\big] \leqslant (1-2\sigma^2 s)^{-k}.$$

Therefore, with $s = 1/(4\sigma^2)$, for which $(1-2\sigma^2 s)^{-k} = 2^k$, we get, from lemma 8.2,

$$\begin{aligned}\mathbb{E}\big[\|\Phi(\hat{\theta}-\theta_*)\|_2^2\big] &\leqslant 16\sigma^2\log\bigg(\binom{d}{2k}2^k\bigg) \\ &\leqslant 16\sigma^2\log\bigg(\Big(\frac{ed}{2k}\Big)^{2k}2^k\bigg) = 16\sigma^2\Big(2k\log\Big(\frac{d}{k}\Big)+(2-\log 2)k\Big).\end{aligned}$$

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, in particular for all sub-Gaussian ε_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 constrained minimization of the empirical risk is done only approximately. See exercise 8.2.

  Exercise 8.2 *Assume that* $\hat{\theta}\in\Theta$ *is such that* $\frac{1}{n}\|y-\Phi\hat{\theta}\|_2^2 \leqslant \inf_{\theta\in\Theta}\frac{1}{n}\|y-\Phi\theta\|_2^2+\rho$. *Show that* $\|\Phi(\hat{\theta}-\theta_*)\|_2^2 \leqslant 4\sup_{\theta\in\Theta}\Big[\varepsilon^\top\Big(\frac{\Phi(\theta-\theta_*)}{\|\Phi(\theta-\theta_*)\|_2}\Big)\Big]^2 + 2n\rho$ *(with notations from section 8.1.1).*

- This result is not improvable by any algorithm (polynomial time or not); see, for example, theorem 2.3 from Giraud (2014) 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 come with guarantees only when such fast rates are available for ℓ_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 $\frac{1}{n}\|y - \Phi\theta\|_2^2$ happens to be separable (i.e., a sum of functions of each component of θ). It can then be easily minimized with respect to θ such that $\|\theta\|_0 \leqslant k$ to obtain θ_t. This is done (with the proof left as an exercise) by computing the unconstrained minimizer $\theta_{t-1} - \frac{1}{L}\frac{1}{n}\Phi^\top(\Phi\theta_{t-1} - y)$ and selecting the k largest components.

Exercise 8.3 (♦) *Consider a linear model $f(x) = \theta^\top\varphi(x)$ with a G-Lipschitz-continuous loss function and features almost surely bounded in ℓ_∞-norm by R. Using section 4.6.1, show that the minimizer of the empirical risk over all $\theta \in \mathbb{R}^d$, such that $\|\theta\|_0 \leqslant k$ and $\|\theta\|_2 \leqslant D$, has an expected risk less than the minimum expected risk over this same set with an additive term proportional to $GRD\sqrt{k\log(d)/n}$.*

8.2.2 Sparsity-Adaptive Estimation (Unknown k) (♦)

In practice, regardless of the computational cost, one does not know k in advance. A classical idea is to consider penalized least-squares regression 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 λ, this almost leads to the same performance as if k were known.

Proposition 8.2 (Model selection–ℓ_0-penalty) *Assume that $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ being a vector with independent Gaussian components of zero mean and variance σ^2, with $\|\theta_*\|_0 \leqslant k$. Let $\hat\theta$ be a minimizer of equation (8.3). Then, for $\lambda = \frac{8\sigma^2}{n}\log(\sqrt{2}d)$, we have*
$$\mathbb{E}\Big[\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2\Big] \leqslant \frac{16k\sigma^2}{n}\big[1 + \log(d)\big] + \frac{16\sigma^2}{n}.$$

Proof (♦♦) We follow the same proof technique as 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 equation (8.1),

$$\begin{aligned}
\|\Phi(\hat\theta - \theta_*)\|_2^2 &\leqslant 2\|\Phi(\hat\theta - \theta_*)\|_2 \cdot \varepsilon^\top \Big(\frac{\Phi(\hat\theta - \theta_*)}{\|\Phi(\hat\theta - \theta_*)\|_2}\Big) + n\lambda\|\theta_*\|_0 - n\lambda\|\hat\theta\|_0 \\
&\leqslant 2\bigg(\varepsilon^\top\Big(\frac{\Phi(\hat\theta - \theta_*)}{\|\Phi(\hat\theta - \theta_*)\|_2}\Big)\bigg)^2 + \frac{1}{2}\|\Phi(\hat\theta - \theta_*)\|_2^2 + n\lambda\|\theta_*\|_0 - n\lambda\|\hat\theta\|_0,
\end{aligned}$$

leading to, by taking the supremum over $\theta \in \mathbb{R}^d$,

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant \sup_{\theta\in\mathbb{R}^d} \bigg\{4\bigg(\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\bigg)^2 + 2n\lambda\|\theta_*\|_0 - 2n\lambda\|\theta\|_0\bigg\}.$$

We then take the supremum by layers, as $\sup\limits_{\theta\in\mathbb{R}^d} = \sup\limits_{k'\in\{1,\dots,d\}} \sup\limits_{|B|=k'} \sup\limits_{\mathrm{supp}(\theta)\subset B}$; that is, using the same derivations as for proposition 8.1 (with A the support of θ_*):

$$\begin{aligned}
&\mathbb{E}\big[\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \\
\leqslant\ & \mathbb{E}\bigg[\sup_{k'\in\{1,\dots,d\}} \sup_{|B|=k'} \sup_{\mathrm{supp}(\theta)\subset B} \bigg\{4\bigg(\varepsilon^\top\Big(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}\Big)\bigg)^2 + 2n\lambda\|\theta_*\|_0 - 2n\lambda k'\bigg\}\bigg] \\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + 4\mathbb{E}\bigg[\sup_{k'\in\{1,\dots,d\}} \sup_{|B|=k'} \Big\{\|\Pi_{\Phi_{A\cup B}}\varepsilon\|_2^2 - \frac{n\lambda}{2}k'\Big\}\bigg].
\end{aligned}$$

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,

$$\begin{aligned}
&\mathbb{E}\big[\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + \frac{4}{s}\log\bigg(\sum_{k'\in\{1,\dots,d\}}\sum_{|B|=k'} \mathbb{E}\big[\exp(s\|\Pi_{\Phi_{A\cup B}}\varepsilon\|_2^2)\big]\exp\Big(-\frac{n\lambda k' s}{2}\Big)\bigg) \\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + 16\sigma^2\log\bigg(\sum_{k'=1}^d \binom{d}{k'} 2^{(k'+\|\theta_*\|_0)/2}\exp\Big(-\frac{n\lambda k'}{8\sigma^2}\Big)\bigg) \qquad (8.4) \\
\leqslant\ & 2n\lambda\|\theta_*\|_0 + 8\sigma^2\|\theta_*\|_0\log(2) + 16\sigma^2\log\bigg(\sum_{k'=1}^d\binom{d}{k'}\exp\Big(k'\Big(\frac{\log(2)}{2} - \frac{n\lambda}{8\sigma^2}\Big)\Big)\bigg) \\
\leqslant\ & (2n\lambda + 8\log(2)\sigma^2)\|\theta_*\|_0 + 16\sigma^2 d\log\Big(1 + \exp\Big(\log(\sqrt{2}) - \frac{n\lambda}{8\sigma^2}\Big)\Big) \\
& \text{using the binomial theorem,} \\
\leqslant\ & (2n\lambda + 8\log(2)\sigma^2)\|\theta_*\|_0 + 16\sigma^2 d\exp\Big(\log(\sqrt{2}) - \frac{n\lambda}{8\sigma^2}\Big). \qquad (8.5)
\end{aligned}$$

To find a good regularization parameter, we can then approximately minimize the bound in equation (8.5) with respect to λ. We obtain a good balance of the two terms by having $\log(\sqrt{2}) - \frac{n\lambda}{8\sigma^2} = -\log d$ (i.e., $\lambda = \frac{8\sigma^2}{n}\log(\sqrt{2}d)$), for which we get

$$\mathbb{E}\big[\|\Phi(\hat\theta - \theta_*)\|_2^2\big] \leqslant (2n\lambda + 8\log(2)\sigma^2)\|\theta_*\|_0 + 16\sigma^2 \leqslant 16\sigma^2\big((\log(d)+1)\|\theta_*\|_0 + 1\big),$$

and obtain 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) gives a penalty proportional to $\|\theta\|_0 \log n$ (which is often a smaller penalty than proposed here).
- Note that we need to know σ^2 in advance to compute λ, 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 the bound does not require any assumption on the design matrix Φ, we observe a positive high-dimensional phenomenon, where d only appears as $\frac{\log d}{n}$, but only exponential-time algorithms are possible for solving the problem with guarantees (see the algorithms that follow).

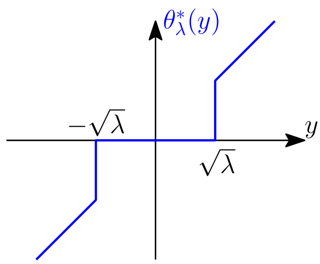
Exercise 8.4 (♦♦) *With a penalty proportional to $\|\theta\|_0 \log \frac{d}{\|\theta\|_0}$, show the same bound as for k known.*

Algorithms. We can extend the two algorithms from the end of section 8.2.1 for the penalized case:

- **Forward-backward algorithm to minimize a function of 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 perform a step only if it decreases the overall cost function. See an analysis of this point 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, as shown in the following diagram:



This is referred to as "iterative hard thresholding" (while for the ℓ_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 ℓ_1-regularization

We now consider a computationally efficient alternative to ℓ_0-penalties (namely, using ℓ_1-penalties), by minimizing, for the square loss,

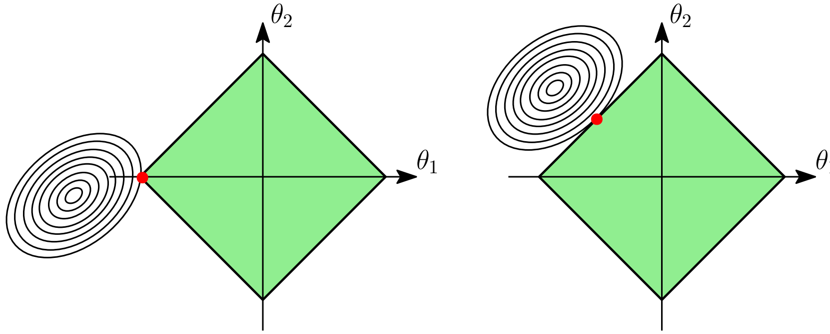
$$\frac{1}{2n}\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1. \tag{8.6}$$

This is a convex optimization problem to which algorithms from chapter 5 can be applied (see instances in section 8.3.1). It is often called the "Lasso" problem, for "least absolute shrinkage and selection operator" (Tibshirani, 1996).

We present algorithms dedicated to solving the optimization problem, and 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 ℓ_2-norm used in ridge regression, the ℓ_1-norm is nondifferentiable, and its nondifferentiability is not limited to $\theta = 0$. Rather, it occurs in many other points. To see this, we can look at the ℓ_1-ball and its varying geometry compared to the ℓ_2-ball. This is directly relevant to situations where we constrain the value of the norm instead of penalizing it:



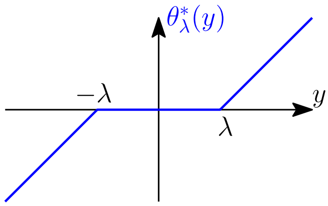
As shown here, where we represent the level set of a potential loss function, the solution of minimizing the loss subject to the ℓ_1-constraint (in green) is obtained when level sets are "tangent" to the constraint set. In the right part, this is obtained at a point away from the axes, but in the left part, this is achieved at one of the corners of the ℓ_1-ball, which is a point where one of the components of θ is equal to zero. Such corners are "attractive"; that is, minimizers tend to be precisely at these corners, which exactly leads to sparse solutions.

The ℓ_1-norm is also often introduced as the convex relaxation of the ℓ_0-penalty. Indeed, the ℓ_1-norm is the convex envelope (the largest convex function that is a lower bound) of the ℓ_0-penalty on the set $[-1,1]^d$ (the proof is left as an exercise). While this provides some intuition about the ℓ_1-norm and its potential generalization to other sparse situations, this does not directly justify its good behavior in 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 the left and right derivatives at zero (the proof is left 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 together as $\theta^*_\lambda(y) = \max\{|y| - \lambda, 0\} \operatorname{sign}(y)$, which is depicted here. This is referred to as "iterative soft thresholding" (this will be useful for the proximal methods discussed next).



Note that the minimizer is either set to zero or shrunk toward zero.

Optimization algorithms. We can adapt algorithms from chapter 5 to the problem in equation (8.6).

- **Iterative soft-thresholding:** We can apply proximal methods (section 5.2.5) 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 $F'(\theta) = \frac{1}{n}\Phi^\top(\Phi\theta - y)$. The plain (i.e., nonaccelerated) 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 $(\theta_t)_j = \max\{|(\eta_t)_j| - \lambda/L, 0\} \operatorname{sign}((\eta_t)_j)$, 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 ℓ_1-norm is a nondifferentiable function, coordinate descent can be applied (because the ℓ_1-norm is "separable," i.e., a sum of terms that depend on single variables). 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).

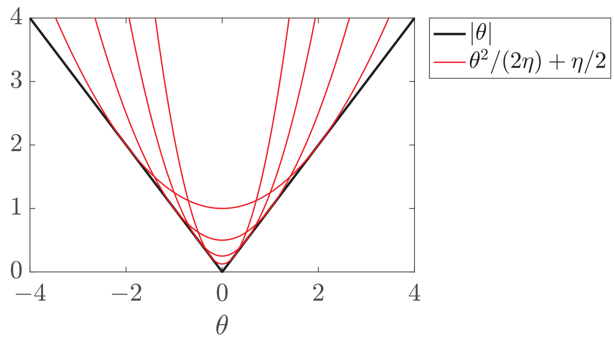
Exercise 8.5 *Provide a closed-form expression for the iteration of the coordinate descent algorithm described just above.*

- **Stochastic gradient descent**: To leverage the ℓ_1-norm geometry, we will need the mirror descent extension of gradient descent presented in section 11.1.3.

η-trick. The nondifferentiability of the ℓ_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 $\eta_j = |\theta_j|$. An example in one dimension, with $|\theta|$ and several quadratic upper bounds, is shown in the following illustration:



This leads to the reformulation of equation (8.6) 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: minimizing with respect to η when θ is fixed can be done in closed form as $\eta_j = |\theta_j|$, while minimizing with respect to θ when η is fixed is a quadratic optimization problem that can be solved by a linear system. This leads to the class of "reweighted ℓ_2-minimization" algorithms.[2]

Optimality conditions (♦). To study the estimator defined by equation (8.6), it is often necessary to characterize when a certain θ 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 zero. The gradient looks only at d directions (along

[2]See more details in `https://www.di.ens.fr/~fbach/ltfp/etatrick.html` and in section 5 of Bach et al. (2012a).

the coordinate axes), while, in the nonsmooth 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 nonnegative. That is, we need to go up in all directions. When H is differentiable at θ, then $\partial H(\theta, \Delta) = H'(\theta)^\top \Delta$, and the positivity for all Δ is equivalent to $H'(\theta) = 0$.

For $H(\theta) = F(\theta) + \lambda\|\theta\|_1$, we have

$$\partial H(\theta, \Delta) \;=\; F'(\theta)^\top \Delta + \lambda \sum_{j,\ \theta_j \neq 0} \operatorname{sign}(\theta_j)\Delta_j + \lambda \sum_{j,\ \theta_j = 0} |\Delta_j|.$$

It is separable in Δ_j, $j = 1, \dots, d$, and it is nonnegative for all j, if and only if all components that depend on Δ_j are nonnegative.

When $\theta_j \neq 0$, then this requires $F'(\theta)_j + \lambda \operatorname{sign}(\theta_j) = 0$, while when $\theta_j = 0$, we need $F'(\theta)_j \Delta_j + \lambda|\Delta_j| \geqslant 0$ for all Δ_j, which is equivalent to $|F'(\theta)_j| \leqslant \lambda$. This leads to the following 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 (the derivations are left as an exercise).

Homotopy method (♦♦). We assume for simplicity that $\Phi^\top\Phi$ is invertible such that the minimizer $\theta(\lambda)$ is unique. Given a certain sign pattern for θ, optimality conditions are all convex in λ and thus define an interval in λ where the sign is preserved. Given the sign, then the solution $\theta(\lambda)$ is affine in λ,[3] leading to a piecewise affine function in λ (see an example of a regularization path in figure 8.1).

Exercise 8.6 *Assume that $\lambda \geqslant \big\|\frac{1}{n}\Phi^\top y\big\|_\infty$. Show that $\theta = 0$ is a minimizer of the Lasso objective function in equation (8.6).*

If we know the breakpoints in λ and the associated signs, we can compute all the solutions for all λ. This is the source of the homotopy algorithm for equation (8.6), which starts with large λ and builds the path of solutions by computing the breakpoints one by one. See more details by Osborne et al. (2000) and 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

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

[3]If $A \subset \{1, \dots, k\}$ is the support associated with the sign vector s, we have $0 = F'(\theta)_A + \lambda s_A = \frac{1}{n}\Phi_A^\top(\Phi_A\theta_A - y) + \lambda s_A$, which leads to $\theta_A = (\Phi_A^\top\Phi_A)^{-1}\Phi_A^\top y - n\lambda(\Phi_A^\top\Phi_A)^{-1}s_A$.

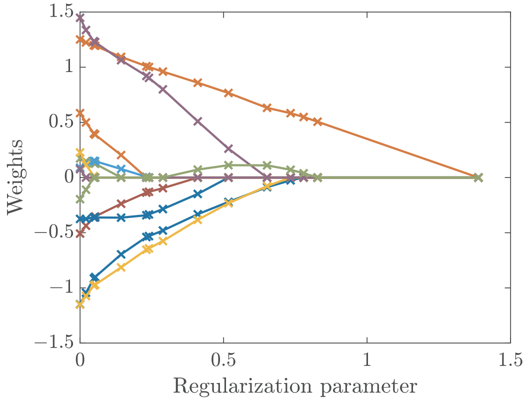


Figure 8.1. Regularization path for a Lasso problem in dimension $d = 32$ and $n = 32$ input observations sampled from a standard Gaussian distribution with 4 nonzero weights equal to -1 or $+1$, and outputs generated with additive Gaussian noise with unit variance. The random seed was chosen so that at least one weight comes in and out of the regularization path.

with ℓ having the 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 the penalized formulation using proposition 4.7 in 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 ℓ_1-norm, which we can compute as

$$\mathrm{R}_n(\mathcal{F}_D) = \mathbb{E}\bigg[\sup_{\|\theta\|_1 \leqslant D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i^{\mathrm{R}} \varphi(x_i)^\top \theta\bigg] = D\mathbb{E}\bigg[\bigg\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i^{\mathrm{R}} \varphi(x_i)\bigg\|_\infty\bigg],$$

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^2 (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). We can then use proposition 1.5 to find that the maximum of the $2d$ sub-Gaussian variables is less than $\big(2R^2\log(2d)/n\big)^{1/2}$. This leads to

$$\mathbb{E}\big[\mathcal{R}(\hat{\theta}_D)\big] \leqslant \inf_{\|\theta\|_1 \leqslant D} \mathcal{R}(\theta) + \frac{4GRD\sqrt{2\log(2d)}}{\sqrt{n}}.$$

When D is large enough (e.g., $D = \|\theta_*\|_1$), then we get an excess risk bounded by $4GRD\sqrt{2\log(2d)}/\sqrt{n}$. If θ_* has only k nonzeros, its ℓ_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, so 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 Φ. We see that we can deal with high-dimensional inference problems where d can be large, but it will be with rates in $1/\sqrt{n}$ (like in section 8.3.2) and not $1/n$ (hence the denomination "slow").

As for proposition 4.7 in section 4.5.5, we study the penalization by a general norm $\Omega : \mathbb{R}^d \to \mathbb{R}$ with a dual norm Ω^* defined as $\Omega^*(z) = \sup_{\Omega(\theta)\leqslant 1} z^\top\theta$ (see exercise 8.7 for classical examples). We thus denote by $\hat\theta$ a minimizer of

$$\frac{1}{2n}\|y - \Phi\theta\|_2^2 + \lambda\Omega(\theta). \tag{8.7}$$

We start with lemma 8.4, which characterizes the excess risk in two situations: (1) where λ is large enough and (2) in the general case.

Lemma 8.4 *Let $\hat\theta$ be a minimizer of equation (8.7).*

(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, following the same reasoning as in section 8.1.1, by optimality of $\hat\theta$ for equation (8.7):

$$\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant 2\varepsilon^\top\Phi(\hat\theta - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat\theta).$$

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 8.7 *For $p \in [1, \infty]$, show that the dual of the ℓ_p-norm is the ℓ_q-norm for $\frac{1}{p} + \frac{1}{q} = 1$.*

We can now use lemma 8.4 to compute the excess risk of the Lasso problem, for which $\Omega = \|\cdot\|_1$ and $\Omega^*(\Phi^\top \varepsilon) = \|\Phi^\top \varepsilon\|_\infty$.[4] 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 lemma 8.4 when λ is larger than $\sqrt{\log(d)/n}$. We denote by $\|\widehat{\Sigma}\|_\infty$ the largest element of matrix $\widehat{\Sigma}$ in absolute value.

Proposition 8.3 (Lasso–slow rate) *Assume that $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ being a vector with independent Gaussian components of zero mean and variance σ^2. Let $\hat{\theta}$ be the minimizer of equation (8.6). 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, $\mathbb{P}(|z| \geqslant t) \leqslant \exp(-t^2/2)$:[5]

$$\mathbb{P}\Big(\|\Phi^\top \varepsilon\|_\infty > \frac{n\lambda}{2}\Big) \leqslant \sum_{j=1}^d \mathbb{P}\Big(|\Phi^\top \varepsilon|_j > \frac{n\lambda}{2}\Big) \leqslant \sum_{j=1}^d \exp\Big(\frac{-n\lambda^2}{8\sigma^2 \widehat{\Sigma}_{jj}}\Big) \leqslant d \exp\Big(\frac{-n\lambda^2}{8\sigma^2 \|\widehat{\Sigma}\|_\infty}\Big) = \delta,$$

because of our choice of λ. Thus, with a 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 exercise 8.8. ■

We already observe a 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 θ_* is sparse such 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 for avoiding that dependence on σ (see section 5.4 from Giraud, 2014) by minimizing $\frac{1}{\sqrt{n}}\|y - \Phi\theta\|_2 + \lambda\|\theta\|_1$.

Proposition 8.3 suggests a regularization parameter λ proportional to $1/\sqrt{n}$, which does enable estimation in high-dimensional situations but can also add a significant bias because all nonzero components of $\hat{\theta}$ are shrunk toward zero. See section 8.5 for methods to alleviate this effect.

[4] Developments similar to proposition 4.7 in section 4.5.5 for general norms could also be carried out.

[5] We have for $t \geqslant 0$, $e^{t^2/2}\mathbb{P}(|z| \geqslant t) = \frac{2}{\sqrt{2\pi}} \int_t^{+\infty} e^{t^2/2 - s^2/2} ds \leqslant \frac{2}{\sqrt{2\pi}} \int_t^{+\infty} e^{-(s-t)^2/2} ds = 1$.

Exercise 8.8 (♦) *With the same assumptions as proposition 8.3, and with the choice of the regularization parameter* $\lambda = 4\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat{\Sigma}\|_\infty}$, *use lemma 8.4 to provide an upper bound of* $\mathbb{E}\big[\frac{1}{n}\|\Phi(\hat\theta-\theta_*)\|_2^2\big]$.

8.3.4 Fast Rates–Fixed Design (♦)

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 the ℓ_0-penalty but with tractable algorithms. This will come with additional strong conditions on the design matrix Φ.

We start with a simple (but crucial) lemma, characterizing the solution of equation (8.6) in terms of the support A of θ_*.

Lemma 8.5 *Let* $\hat\theta$ *be a minimizer of equation (8.6). Assume that* $\|\Phi^\top\varepsilon\|_\infty \leqslant \frac{n\lambda}{2}$. *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, as in previous proofs (e.g., lemma 8.4), with $\Delta = \hat\theta - \theta_*$ and A being the support of θ_*,

$$\|\Phi\Delta\|_2^2 \leqslant 2\varepsilon^\top\Phi\Delta + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat\theta\|_1.$$

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 get, by using the decomposability of the ℓ_1-norm and the triangle inequality,

$$\|\theta_*\|_1 - \|\hat\theta\|_1 = \|(\theta_*)_A\|_1 - \|\theta_*+\Delta\|_1 = \|(\theta_*)_A\|_1 - \|(\theta_*+\Delta)_A\|_1 - \|\Delta_{A^c}\|_1 \leqslant \|\Delta_A\|_1 - \|\Delta_{A^c}\|_1,$$

leading to

$$\begin{aligned}
\|\Phi\Delta\|_2^2 &\leqslant n\lambda\|\Delta\|_1 + 2n\lambda(\|\theta_*\|_1 - \|\hat\theta\|_1) \leqslant n\lambda\|\Delta\|_1 + 2n\lambda(\|\Delta_A\|_1 - \|\Delta_{A^c}\|_1)\\
&\leqslant n\lambda(\|\Delta_A\|_1 + \|\Delta_{A^c}\|_1) + 2n\lambda(\|\Delta_A\|_1 - \|\Delta_{A^c}\|_1) = 3n\lambda\|\Delta_A\|_1 - n\lambda\|\Delta_{A^c}\|_1.
\end{aligned}$$

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 is a constant $\kappa > 0$ such that

$$\frac{1}{n}\|\Phi\Delta\|_2^2 \geqslant \kappa\|\Delta_A\|_2^2 \tag{8.8}$$

for all Δ that satisfies the condition $\|\Delta_{A^c}\|_1 \leqslant 3\|\Delta_A\|_1$. This is called the "restricted eigenvalue property" because if the smallest eigenvalue of $\frac{1}{n}\Phi^\top\Phi$ is greater than κ, the condition is satisfied (but this is possible only if $n \geqslant d$). The relevance of this assumption is discussed in section 8.3.5. This leads to the proposition 8.4.

Proposition 8.4 (Lasso–fast rate) *Assume that $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ being a vector with independent Gaussian components of zero mean and variance σ^2. Denote as $A = \operatorname{supp}(\theta_*)$ the support of θ_*. Let $\hat\theta$ be the minimizer of equation (8.6). Then, for $\lambda = \frac{2\sigma}{\sqrt{n}}\sqrt{2\|\widehat\Sigma\|_\infty}\sqrt{\log(2d) + \log\frac{1}{\delta}}$, we have, if equation (8.8) is satisfied, and with probability greater than $1-\delta$,*

$$\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2 \leqslant \frac{72|A|\sigma^2}{n}\frac{\|\widehat\Sigma\|_\infty}{\kappa}\Big(\log(2d) + \log\frac{1}{\delta}\Big).$$

Proof (♦) We have, when λ is large enough, by application of lemma 8.5 and using equation (8.8):

$$\|\Delta_A\|_1 \leqslant |A|^{1/2}\|\Delta_A\|_2 \leqslant \frac{|A|^{1/2}}{\sqrt{n\kappa}}\|\Phi\Delta\|_2 \leqslant \frac{|A|^{1/2}}{\sqrt{n\kappa}}\sqrt{3n\lambda\|\Delta_A\|_1},$$

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 (namely that the ratio $\|\widehat\Sigma\|_\infty/\kappa$ is finite and not too large). Such results can be extended beyond the square loss using the notion of self-concordance (see, e.g., Ostrovskii and Bach, 2021, and references therein).

Exercise 8.9 (♦♦) *With the same assumptions as proposition 8.4, with the choice of the regularization parameter $\lambda = 4\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat\Sigma\|_\infty}$, provide an upper bound on the expectation of the excess risk $\mathbb{E}\big[\frac{1}{n}\|\Phi(\hat\theta - \theta_*)\|_2^2\big]$.*

8.3.5 Zoo of Conditions (♦♦)

Conditions to obtain fast rates through equation (8.8) 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. The most direct condition is the so-called restricted eigenvalue property (REP), which is exactly equation (8.8) with the supremum taken over the unknown set A of a cardinality less than k,

$$\inf_{|A|\leqslant k}\ \inf_{\|\Delta_{A^c}\|_1 \leqslant 3\|\Delta_A\|_1} \frac{\|\Phi\Delta\|_2^2}{n\|\Delta_A\|_2^2} \geqslant \kappa > 0. \tag{8.9}$$

Mutual incoherence condition. A simpler one to check, but stronger, is the mutual incoherence condition:

$$\sup_{i\neq j}|\widehat\Sigma_{ij}| \leqslant \frac{\min_{j\in\{1,\dots,d\}}\widehat\Sigma_{jj}}{14k}, \tag{8.10}$$

which states that all cross-correlation coefficients are small (pure decorrelation would set them to zero).

This is weaker than the REP condition. 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, with $\mathrm{Diag}(\mathrm{diag}(\widehat{\Sigma}_{AA}))$ the diagonal matrix with diagonal the diagonal of $\widehat{\Sigma}_{AA}$,

$$\begin{aligned}\Delta_A^\top\widehat{\Sigma}_{AA}\Delta_A &= \Delta_A^\top \mathrm{Diag}(\mathrm{diag}(\widehat{\Sigma}_{AA}))\Delta_A + \Delta_A^\top(\widehat{\Sigma}_{AA} - \mathrm{Diag}(\mathrm{diag}(\widehat{\Sigma}_{AA}))\Delta_A \\ &\geqslant \min_{j\in\{1,\dots,d\}} \widehat{\Sigma}_{jj}\Big(\|\Delta_A\|_2^2 - \frac{1}{14k}\|\Delta_A\|_1^2\Big),\end{aligned}$$

and

$$\big|\Delta_A^\top\Phi_A^\top\Phi_{A^c}\Delta_{A^c}\big| \leqslant \frac{\min_{j\in\{1,\dots,d\}}\widehat{\Sigma}_{jj}}{14k}\|\Delta_{A^c}\|_1\|\Delta_A\|_1 \leqslant \frac{3\min_{j\in\{1,\dots,d\}}\widehat{\Sigma}_{jj}}{14k}\|\Delta_A\|_1^2.$$

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 equation (8.9).

Restricted isometry property. One of the earlier conditions was the restricted isometry property: all eigenvalues of submatrices of $\widehat{\Sigma}$ of a size less than $2k$ are between $1-\delta$ and $1+\delta$ for values of δ that are small enough. See Giraud (2014) and Wainwright (2019) for details.

Gaussian designs (♦♦). It is not obvious that the conditions given here are nontrivial (i.e., there may be no matrix with good sizes d and n for k large enough). For our results to be nontrivial, we need $k\frac{\log d}{n}$ to be small (so the bound $\sigma^2 k\log(d)/n$ is small compared to σ^2) but not too small (so the guarantee is already satisfied for moderate n). Here, we show, without proof, that when sampling from Gaussian distributions, the REP assumption above is satisfied. This is a first step toward a random design assumption.

Proposition 8.5 (Theorem 7.16 from Wainwright, 2019) *If sampling $\varphi(x)$ from a Gaussian with mean zero and covariance matrix Σ, then with probability greater than $1 - \frac{e^{-n/32}}{1-e^{-n/32}}$, the REP is satisfied with $\kappa = \frac{1}{16}\lambda_{\min}(\Sigma)$ as soon as $k\frac{\log d}{n} \leqslant \frac{1}{12800}\frac{\lambda_{\min}(\Sigma)}{\|\Sigma\|_\infty}$.*

Proposition 8.5 is hard to prove; exercise 8.10 proposes to establish a weaker result, showing that the guarantees for the maximal cardinality k of the support have to be smaller.

Exercise 8.10 (♦♦♦) *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 equation (8.10) is satisfied.*

Model selection and irrepresentable condition (♦). Given that the Lasso problem aims at performing variable selection, it is natural to study its capacity to find the support of θ_*; that is, the set of nonzero variables. It turns out that it also depends on some conditions on the design matrix, which are stronger than the REP conditions and are called the "irrepresentable conditions" and also are valid for Gaussian random matrices with similar scalings between n, d, and k. See Giraud (2014) and Wainwright (2019) for details.

Algorithmic and theoretical tools are similar to "compressed sensing" (see, e.g., Eldar and Kutyniok, 2012), 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 Fast Rates–Random Design (♦)

In this section, we study the Lasso problem 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 consider the well-specified Lasso case, where the expected risk is then equal to $\mathcal{R}(\theta) = \frac{\sigma^2}{2} + \frac{1}{2}(\theta - \theta_*)^\top \Sigma(\theta - \theta_*)$. We assume that $\lambda_{\min}(\Sigma) \geqslant \mu > 0$; that is, the *expected* risk is μ-strongly-convex (which cannot be the case for the empirical risk as soon as $d > n$). We derive in proposition 8.6 a convergence rate for the excess risk that is of the form $\sigma^2 k \log(d)/n$ multiplied by the quantity R^2/μ, which we can identify as a condition number for the problem (which is thus close to one when the features have low correlation between them).

Proposition 8.6 (Lasso–fast rate for random design) *Assume given i.i.d. observations (x_i, y_i), $i = 1, \dots, n$, from the model $y = \varphi(x)^\top \theta_* + \varepsilon$, with ε having a sub-Gaussian distribution with parameter σ^2 and $\|\varphi(x)\|_\infty \leqslant R$ almost surely. Assume that $\delta \in (0, 1)$, $\lambda_{\min}(\Sigma) \geqslant \mu > 0$, and $\lambda = \frac{2\sigma R}{\sqrt{n}}\sqrt{2 \log \frac{4d^2}{\delta}}$. Then, if $n \geqslant 2\big[32R^2|A|\log(4d^2/\delta)/\mu\big]^2$, where A is the support of θ_*, we have, with probability at least $1 - \delta$,*

$$\mathcal{R}(\hat{\theta}_\lambda) - \mathcal{R}(\theta^*) \leqslant 2304 \cdot \frac{R^2}{\mu} \frac{\sigma^2 |A|}{n} \log \frac{4d^2}{\delta}.$$

Proof (♦♦) Denoting as $\Phi \in \mathbb{R}^{n \times d}$ the design matrix, and $\varepsilon \in \mathbb{R}^n$ the noise vector, we have

$$\widehat{\mathcal{R}}(\theta) = \frac{1}{2n}\|\Phi(\theta - \theta_*) - \varepsilon\|_2^2 = \frac{1}{2}(\theta - \theta_*)^\top \widehat{\Sigma}(\theta - \theta_*) - (\theta - \theta_*)^\top \Big(\frac{1}{n}\Phi^\top \varepsilon\Big) + \frac{1}{2n}\|\varepsilon\|_2^2, \quad (8.11)$$

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 noncentered covariance matrix (i.e., the second-moment matrix).

We will need $\left\|\frac{1}{n}\Phi^\top\varepsilon\right\|_\infty = \left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i\varphi(x_i)\right\|_\infty$ to be small enough, as well as a small error in the covariance matrix $\left\|\widehat{\Sigma}-\Sigma\right\|_\infty$. Since ε is sub-Gaussian with constant σ^2, and $\|\varphi(x)\|_\infty \leqslant R$ almost surely, we get that, using results from section 1.2.1 (Hoeffding's inequality and union bound), for any $t>0$,

$$\mathbb{P}\Big(\Big\|\frac{1}{n}\Phi^\top\varepsilon\Big\|_\infty \geqslant \frac{\sigma R t}{\sqrt{n}}\Big) \leqslant 2de^{-t^2/2} \text{ and } \mathbb{P}\Big(\big\|\widehat{\Sigma}-\Sigma\big\|_\infty \geqslant \frac{R^2 t}{\sqrt{n}}\Big) \leqslant 2\frac{d(d+1)}{2}e^{-t^2/2}.$$

Thus, the probability that at least one of these inequalities will be satisfied is less than $d(d+3)\exp(-t^2/2) \leqslant 4d^2\exp(-t^2/2)=\delta$, leading to $t=\sqrt{2\log\frac{4d^2}{\delta}}$.

We now assume that $\left\|\frac{1}{n}\Phi^\top\varepsilon\right\|_\infty \leqslant \frac{\sigma R t}{\sqrt{n}}$ and $\left\|\widehat{\Sigma}-\Sigma\right\|_\infty \leqslant \frac{R^2 t}{\sqrt{n}}$, which happens with a probability of at least $1-\delta$. From lemma 8.5, we know that if we have $\lambda \geqslant 2\left\|\frac{1}{n}\Phi^\top\varepsilon\right\|_\infty$ (which is satisfied with our choice $\lambda=\frac{2\sigma R t}{\sqrt{n}}$), then we have, with $\hat{\Delta}=\hat{\theta}_\lambda-\theta_*$,

$$\|\hat{\Delta}_{A^c}\|_1 \leqslant 3\|\hat{\Delta}_A\|_1 \text{ and } \|\hat{\theta}_\lambda\|_1 \leqslant 3\|\theta_*\|_1.$$

Let $v=\mathcal{R}(\hat{\theta}_\lambda)-\mathcal{R}(\theta^*)$ be the excess risk. We have, denoting $\widehat{\mathcal{R}}$ the empirical risk and $\widehat{\mathcal{R}}_\lambda$ its regularized version,

$$\begin{aligned}
v &\leqslant \mathcal{R}(\hat{\theta}_\lambda)-\mathcal{R}(\theta^*)-\widehat{\mathcal{R}}_\lambda(\hat{\theta}_\lambda)+\widehat{\mathcal{R}}_\lambda(\theta^*) \text{ since } \hat{\theta}_\lambda \text{ minimizes } \widehat{\mathcal{R}}_\lambda,\\
&= \mathcal{R}(\hat{\theta}_\lambda)-\mathcal{R}(\theta^*)-\widehat{\mathcal{R}}(\hat{\theta}_\lambda)+\widehat{\mathcal{R}}(\theta^*)+\lambda\|\theta_*\|_1-\lambda\|\hat{\theta}_\lambda\|_1 \text{ by definition of } \widehat{\mathcal{R}}_\lambda,\\
&= \frac{1}{2}\hat{\Delta}^\top(\Sigma-\widehat{\Sigma})\hat{\Delta}+\hat{\Delta}^\top\Big(\frac{1}{n}\Phi^\top\varepsilon\Big)+\lambda\|\theta_*\|_1-\lambda\|\hat{\theta}_\lambda\|_1 \text{ using equation (8.11) },\\
&\leqslant \frac{1}{2}\big\|\widehat{\Sigma}-\Sigma\big\|_\infty\cdot\|\hat{\Delta}\|_1^2+\Big\|\frac{1}{n}\Phi^\top\varepsilon\Big\|_\infty\cdot\|\hat{\Delta}\|_1+\lambda\|\hat{\Delta}\|_1 \text{ using norm inequalities},\\
&\leqslant \frac{R^2 t}{2\sqrt{n}}\cdot\|\hat{\Delta}\|_1^2+\frac{\sigma R t}{\sqrt{n}}\cdot\|\hat{\Delta}\|_1+\lambda\|\hat{\Delta}\|_1 \text{ using our assumptions on } \widehat{\Sigma} \text{ and } \Phi^\top\varepsilon.
\end{aligned}$$

Moreover, since $\lambda_{\min}(\Sigma)\geqslant\mu$, the function $\mathcal{R}$ is μ-strongly-convex; thus, we have

$$v=\mathcal{R}(\hat{\theta}_\lambda)-\mathcal{R}(\theta^*)\geqslant\frac{\mu}{2}\|\hat{\Delta}\|_2^2\geqslant\frac{\mu}{2|A|}\|\hat{\Delta}_A\|_1^2,$$

which leads 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 the following two inequalities:

$$v\leqslant\frac{3\sigma R t}{\sqrt{n}}\cdot\|\hat{\Delta}\|_1+\frac{R^2 t}{2\sqrt{n}}\cdot\|\hat{\Delta}\|_1^2 \quad\text{and}\quad \|\hat{\Delta}\|_1\leqslant 4\sqrt{\frac{2|A|v}{\mu}}. \tag{8.12}$$

Since $1\geqslant\frac{32R^2t}{\sqrt{n}}\frac{|A|}{\mu}$ (given our assumption on n), the last term in the first inequality in equation (8.12) 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, $v\leqslant 48^2\cdot\frac{R^2}{\mu}\frac{\sigma^2|A|}{n}\log\frac{4d^2}{\delta}$, which is the desired result. ■

Exercise 8.11 *With the notations of section 8.3.6, show that if* $\mu = 0$*, from equation (8.12), 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 θ_* 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 signal-to-noise ratio equal to one. 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 $\lambda\|\theta\|_2^2$.
- Lasso regression: Penalty by $\lambda\|\theta\|_1$.
- Orthogonal matching pursuit (OMP), which is a greedy forward method, with hyperparameter k (the number of included variables).

We compare two situations: (1) nonrotated data (exactly the model of data described above), and (2) rotated data, where we replace Φ with ΦS and θ_* by $S^\top \theta_*$, where S is a random rotation matrix. For the nonrotated data (left plot), we can observe that sparse methods (Lasso and OMP) are superior to ridge regression, with a sharp dependence in $\log(d)$ for Lasso. For the rotated data (right plot), we do not expect sparse solutions; hence, sparse methods are not expected to work better than ridge regression, with Lasso not much worse than ridge regression (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 ℓ_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 ℓ_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 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 especially used when output y is multidimensional (such as in multivariate regression or multicategory classification) to select variables that are relevant to all outputs. See, for example, Giraud (2014) for details.

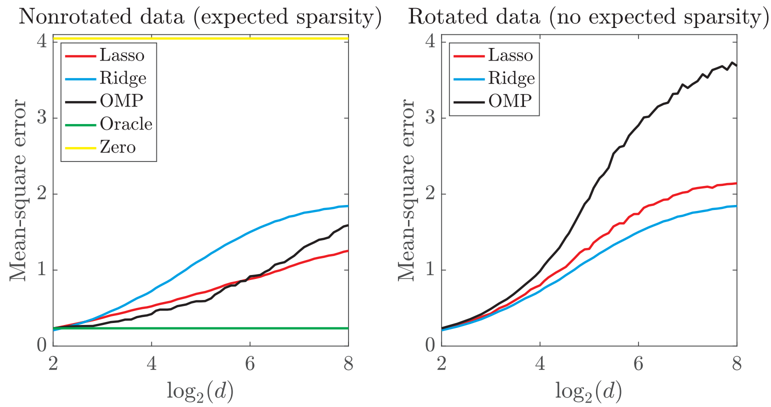


Figure 8.2. Comparison of estimators on least-squares regression: problem with sparse optimal predictor (left), and nonsparse optimal predictor (right).

Exercise 8.12 *Assuming that the design matrix Φ is orthogonal, compute the minimizer 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, or graphs, when such prior knowledge is needed. See Bach et al. (2012b) for details.

Exercise 8.13 *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 nonzeros 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, as well as applications in chapter 13, in the context of multicategory classification.

Exercise 8.14 *Compute the minimizer of $\frac{1}{2n}\|Y - \Theta\|_F^2 + \lambda\|\Theta\|_*$, where $\|M\|_F$ is the Frobenius norm and $\|M\|_*$ is the nuclear norm.*

Exercise 8.15 *Show that $\|M\|_*$ is the minimum of $\frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2$ over all decompositions of $M = UV^\top$.*

- **Multiple kernel learning:** The group penalty can be extended when the groups have an infinite dimension and ℓ_2-norms are replaced by reproducing kernel Hilbert space (RKHS) norms defined in chapter 7. This becomes a tool for learning the kernel matrix from data. See section 1.5 in Bach et al. (2012a), as well as Gönen and Alpaydın (2011), for details.

Exercise 8.16 (♦) *Consider m feature vectors $\varphi_j : \mathcal{X} \to \mathcal{H}_j$, associated with kernels $k_j : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for $j \in \{1, \dots, m\}$. Show that*

$$\inf_{\theta_1,\dots,\theta_m} \frac{1}{n} \sum_{i=1}^n \ell\big(y_i, \langle \theta_1, \varphi_1(x_i)\rangle + \cdots + \langle \theta_m, \varphi_m(x_i)\rangle\big) + \frac{\lambda}{2}\big(\|\theta_1\| + \cdots + \|\theta_m\|\big)^2$$

is equivalent to $\inf_{\eta \in \Delta_m} \inf_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K(\eta)\alpha)_i) + \frac{\lambda}{2}\alpha^\top K(\eta)\alpha$, *where $K(\eta) \in \mathbb{R}^{n\times n}$ is the kernel matrix associated with the kernel $\eta_1 k_1 + \cdots + \eta_m k_m$ and Δ_m is the simplex in dimension m.*

- **Elastic net:** Often, when both effects of the ℓ_1-norm (sparsity) and the squared ℓ_2-norm (with 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 ℓ_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 problem, 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 nonconvex optimization problem, where iterative weighted ℓ_1-minimization provides natural algorithms (see Mairal et al., 2014, and references therein).

Exercise 8.17 *Show that for $\alpha \in (0,1)$, $\frac{1}{\alpha}u^\alpha = \inf_{\eta>0} \frac{u}{\eta} + \big(\frac{1}{\alpha} - 1\big)\eta^{\alpha/(1-\alpha)}$, and derive both a reweighted ℓ_1-minimization and a reweighted ℓ_2-minimization algorithm for the penalty $\sum_{i=1}^d |\theta_i|^\alpha$.*

8.6 Conclusion

In this chapter, we have considered sparse methods based on the penalization by the ℓ_0- or ℓ_1-penalties of the weight vector of a linear model. For the square loss, ℓ_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, ℓ_1-norm penalization can be solved efficiently with appropriate convex optimization algorithms (such as proximal methods), but it 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 be obtained only with stronger assumptions on the covariance matrix of the features.

This chapter was limited to linear models. In chapter 9, on neural networks, we will see how models that are nonlinear in their parameters can lead to nonlinear variable selection, still exhibiting a high-dimensional phenomenon but at the expense of harder optimization. This will be obtained by an ℓ_1-norm on an infinite-dimensional space, and studied further in the context of gradient boosting in section 10.3.

Chapter 9

Neural Networks

Chapter Summary

- Neural networks are flexible models for nonlinear 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 nonlinearities.
- Optimization error: As the prediction functions are nonlinearly dependent on their parameters, we obtain nonconvex optimization problems with guaranteed convergence only 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 rectified linear unit (ReLU) activation function, the universal approximation properties can be characterized and are superior to those of kernel methods because they are adaptive to linear latent variables. In particular, neural networks can efficiently perform nonlinear variable selection.

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 Φ is convex; for example, $\Phi(u) = \max\{1-u, 0\}$ (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 that we have considered so far was as follows (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 regression) and chapter 4 (for Lipschitz-continuous losses).

  – *Pros:* They are simple to implement, as they lead to convex optimization with gradient descent (GD) algorithms, with running time complexity in $O(nd)$, as shown in chapter 5. They come with theoretical guarantees that are not necessarily scaling badly with dimension d if regularizers are used (ℓ_2- or ℓ_1-norm).
  – *Cons:* They only apply to linear functions on explicit (and fixed feature spaces), so they can underfit the data. Moreover, the feature vector φ is not learned from 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:* They are nonlinear, flexible predictions, simple to implement, and can be used with convex optimization algorithms with strong guarantees. They provide adaptivity to the regularity of the target function, allowing higher-dimensional applications than local averaging methods from chapter 6.
  – *Cons:* The running-time complexity goes up to $O(n^2)$ with algorithms from section 7.4 (but this scaling can be improved with appropriate techniques 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.

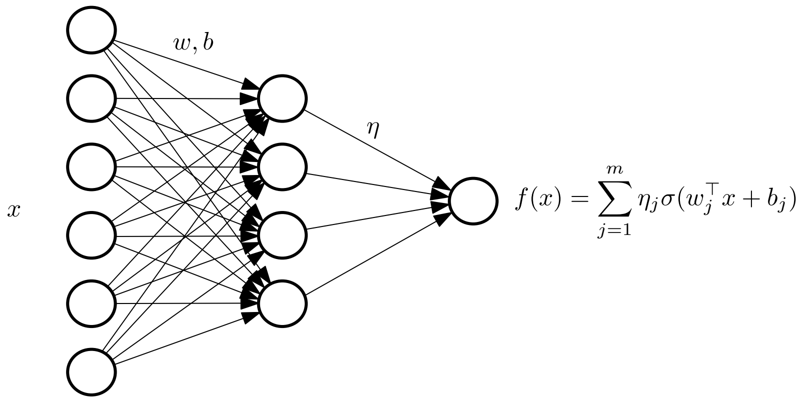
This chapter aims to explore another class of functions for nonlinear predictions– namely, neural networks, which come with additional benefits, such as more adaptivity to linear latent variables, but also have some potential drawbacks, such as a harder optimization problem.

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

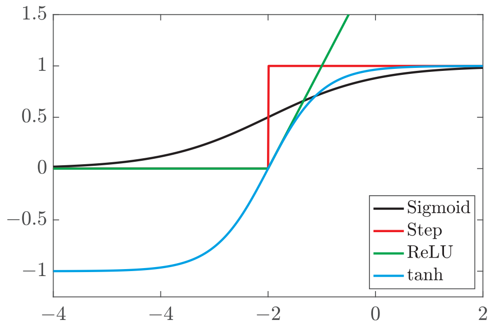
$$f(x) = \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j), \tag{9.1}$$

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 σ is an *activation function*. This is often represented as the following graph. The same architecture can also be considered with $\eta_j \in \mathbb{R}^k$, for $k > 1$ to deal with multicategory classification (see section 13.1).



The activation function is typically chosen from one of the following examples (see the following plot):

- Sigmoid $\sigma(u) = \frac{1}{1+e^{-u}}$.
- Step function $\sigma(u) = 1_{u>0}$, which is not continuous and has zero derivatives everywhere (and thus is not amenable to gradient-based optimization).
- Rectified linear unit (ReLU) $\sigma(u) = (u)_+ = \max\{u, 0\}$, which will be the main focus of this chapter.
- Hyperbolic tangent $\sigma(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$.



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] If the input weights are fixed, we obtain a linear model with the m hidden neurons as features. A key benefit of neural networks is that they perform feature learning by optimizing with respect to input weights.

The constant terms b_j are sometimes referred to as "biases," which is unfortunate in a statistical context, as that word already has a precise meaning within the bias/variance trade-off (see chapter 3 and section 7.3).

Do not be 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 nonlinearity 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

$$-1_{y=1} \log g(x) - 1_{y=-1} \log(1-g(x)) = 1_{y=1} \log\left(1+e^{-f(x)}\right) + 1_{y=-1} \log\left(1+e^{f(x)}\right)$$

which is exactly the logistic loss $\log\left(1+e^{-yf(x)}\right)$ defined in section 4.1.1 applied to prediction function $f(x)$. Practitioners sometimes refer to the cross-entropy loss without mentioning that a sigmoid is applied beforehand (they, in fact, mean the logistic loss). Such a discussion applies as well to multicategory classification and the softmax loss (see section 13.1.1).

Theoretical analysis of neural networks. As with any method based on empirical risk minimization, we have to study the three classical aspects: (1) optimization error (convergence properties of algorithms for minimizing the risk), (2) estimation error (the 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).

[1] See `https://playground.tensorflow.org/` for a nice interactive illustration of this architecture.

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:

$$\min_{\theta \in \mathbb{R}^{m(d+2)}} \ \frac{1}{n} \sum_{i=1}^{n} \ell\Big(y_i, \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x_i + b_j)\Big), \tag{9.2}$$

with potentially additional regularization (often the squared ℓ_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 in equation (9.2) is just a means to an end.

This is a nonconvex optimization problem where the GD algorithms from chapter 5 can be applied without a strong guarantee beyond obtaining a vector with a small gradient norm (section 5.2.6). See the following discussion for recent results when providing qualitative global convergence guarantees when m is large.

While stochastic gradient descent (SGD) remains an algorithm of choice (also with 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 as presented in section 5.4.2 (Duchi et al., 2011), momentum (Kingma and Ba, 2014), batch normalization (Ioffe and Szegedy, 2015), and layer normalization (Ba et al., 2016) to make the optimization better behaved. However, overall, the objective function is nonconvex, and it remains challenging to understand precisely why gradient-based methods perform well in practice, particularly with deeper networks (some elements are presented next and in chapter 12). See also boosting procedures in section 10.3 and chapter 12, which learn neuron weights incrementally.

Global convergence of GD for infinite widths (♦). It turns out that global convergence can be shown for this nonconvex optimization problem (Chizat and Bach, 2018; Bach and Chizat, 2022), with tools that go beyond the scope of this book and are partially described in chapter 12.[2]

We simply show some experimental evidence for a simple one-dimensional setup, where we compare several runs of SGD when observations are seen only 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) equal to zero. We show in figure 9.1 the estimated predictors and the corresponding testing errors with 20 different initializations. We can observe that small errors are never achieved when $m = 5$ (which is sufficient to have 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 overparameterization (see more details in section 12.3).

[2] See also https://francisbach.com/gradient-descent-neural-networks-global-convergence/ for more details.

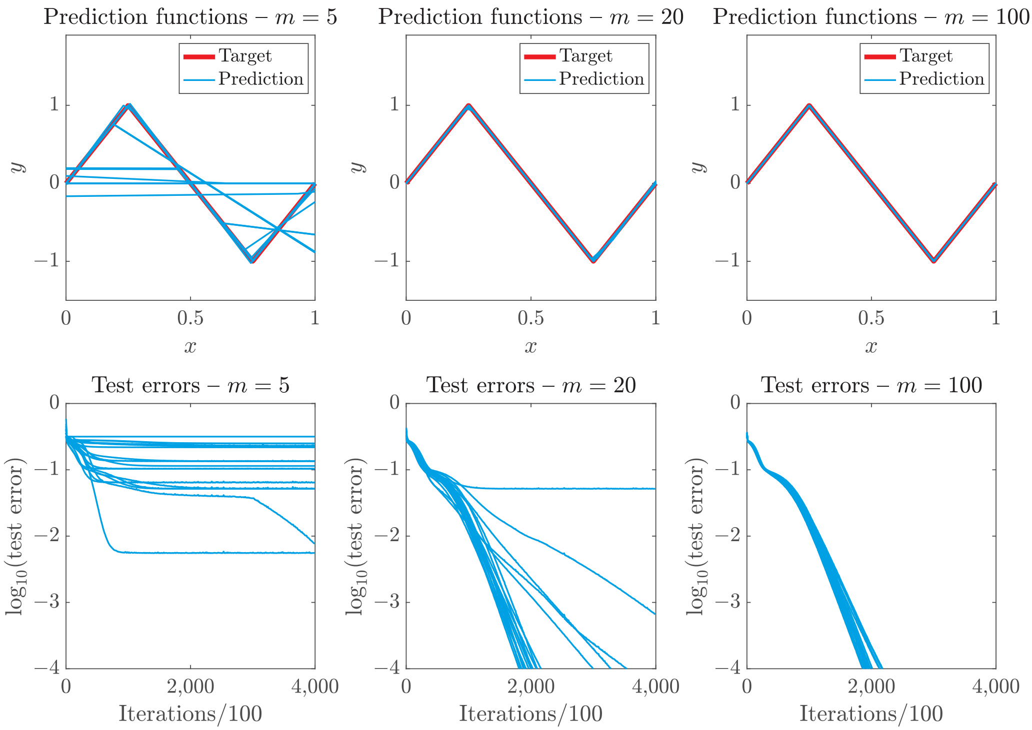


Figure 9.1. Comparison of optimization behavior for different numbers m of neurons for ReLU activations (left: $m = 5$; middle: $m = 20$; right: $m = 100$). To generate the data, we also used a neural network with ReLU activations and 3 hidden neurons. Top: examples of final prediction functions at convergence; bottom: plot of test errors versus the number of iterations.

9.2.2 Rectified Linear Units and Homogeneity

From now on, we will mostly focus on the ReLU activation $\sigma(u) = u_+$. The main property that we will employ 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 α_j and divide $(w_j, b_j) \in \mathbb{R}^{d+1}$ by the same α_j without changing the prediction function, since then $\eta_j(w_j^\top x + b_j)_+ = (\alpha_j \eta_j)\big(\big(\frac{w_j}{\alpha_j}\big)^\top x + \frac{b_j}{\alpha_j}\big)_+$.

This has a particular effect when using a squared ℓ_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 to the constant term for unit homogeneity reasons between the slope w_j and the constant term b_j (R will be a bound on the ℓ_2-norm of input data). Dealing with unit homogeneity between η_j and $(w_j, b_j/R)$ does not matter because of the invariance by rescaling described next.

Optimizing with respect to a scaling factor α_j (which affects 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 $\alpha_j^2 = (\|w_j\|_2^2 + b_j^2/R^2)^{1/2}/|\eta_j|$ as a minimizer and with the optimal value of the penalty equal to $2|\eta_j|(\|w_j\|_2^2 + b_j^2/R^2)^{1/2}$ (note that this leads to an ℓ_1-norm penalty, thus with potentially sparsifying effects (setting some of the output weights η_j to zero), and robustness to large number of neurons (as shown in section 9.2.3); for other relationship between ℓ_2-regularization in neural networks and sparse estimation, see section 12.1.3.

Therefore, for the theoretical analysis (study of the approximation and estimation errors), because of homogeneity, we can 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 ℓ_1-norm penalty on η; that is, $\|\eta\|_1$ (we will consider other normalizations for the input weights, either to ease the exposition or to induce another behavior; e.g., by using ℓ_1-norms on the w_j's). We focus on this choice of regularization in the following sections.

In this chapter, R denotes an almost sure upper bound on x directly, 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, $\|w_j\|_2^2 + b_j^2/R^2 = 1$ for each $j \in \{1, \dots, m\}$ and $\|\eta\|_1 \leqslant D$. This defines a set Θ of allowed parameters $\theta = \{(\eta_j), (w_j), (b_j)\}$.

Defining the class $\mathcal{F}$ of neural network models f_θ with parameters $\theta \in \Theta$, we can compute its Rademacher complexity 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 the ℓ_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 $f \in \mathcal{F}$. Note that following section 4.5.3, we consider a constraint on $\|\eta\|_1$, but we could also penalize, which is more common 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 to be independent and identically distributed (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 estimation error, as, using symmetrization from proposition 4.2 and equation (4.10) from section 4.4, when $\hat{f}$ is a minimizer of the empirical risk over $\mathcal{F}$, we have

$$\mathbb{E}\Big[\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f)\Big] \leqslant 4\mathrm{R}_n(\mathcal{G}).$$

We can now use the properties of Rademacher complexities presented in section 4.5, particularly their nice handling of nonlinearities. Assuming that the loss is G-Lipschitz-continuous with respect to the second variable, using proposition 4.3 from chapter 4, which allows getting rid of the loss, we get the following bound:

$$\mathrm{R}_n(\mathcal{G}) \leqslant G \cdot \mathbb{E}\Big[\sup_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i f_\theta(x_i)\Big] = G \cdot \mathbb{E}\Big[\sup_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \eta_j \varepsilon_i (w_j^\top x_i + b_j)_+\Big].$$

Using the ℓ_1-constraint on η and $\sup_{\|\eta\|_1 \leqslant D} z^\top \eta = D\|z\|_\infty$, we can directly maximize with respect to $\eta \in \mathbb{R}^m$, leading to (note that another ℓ_p-constraint on η, with $p \neq 1$, would be harder to deal with):

$$\mathrm{R}_n(\mathcal{G}) \leqslant G \cdot \mathbb{E}\Big[\sup_{j \in \{1,\dots,m\}} \sup_{\|w_j\|_2^2 + b_j^2/R^2 = 1} D\Big|\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i (w_j^\top x_i + b_j)_+\Big|\Big].$$

Notice now that all optimization problems for $j \in \{1, \dots, m\}$ are the same. Thus, we get

$$\mathrm{R}_n(\mathcal{G}) \leqslant G \cdot \mathbb{E}\Big[\sup_{\|w\|_2^2 + b^2/R^2 = 1} D\Big|\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i (w^\top x_i + b)_+\Big|\Big].$$

Since the ReLU activation function is 1-Lipschitz continuous and satisfies $(0)_+ = 0$, we get, this time using the extension of proposition 4.3 from chapter 4 to Rademacher complexities defined with an absolute value (i.e., proposition 4.4), which adds an extra factor of 2,

$$\mathrm{R}_n(\mathcal{G}) \leqslant 2GD \cdot \mathbb{E}\Big[\sup_{\|w\|_2^2 + b^2/R^2 = 1} \Big|w^\top \Big(\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i x_i\Big) + b\Big(\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i\Big)\Big|\Big].$$

We can now perform the optimization with respect to (w, b) in closed form,[3] 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 $\varepsilon_1, \dots, \varepsilon_n$:

$$\begin{aligned}\mathrm{R}_n(\mathcal{G}) &\leqslant 2GD\Big(\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_2^2 + R^2\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big]\Big)^{1/2} \qquad (9.3)\\ &= 2GD\Big(\frac{1}{n}\mathbb{E}[\|x\|_2^2] + \frac{R^2}{n}\Big)^{1/2} \leqslant \frac{2GDR\sqrt{2}}{\sqrt{n}} \leqslant \frac{4GDR}{\sqrt{n}}.\end{aligned}$$

Thus, we get proposition 9.1, with a bound proportional to $1/\sqrt{n}$ *with no explicit dependence in the number of parameters.*

Proposition 9.1 (Estimation error) *Let $\mathcal{F}$ be the class of neural networks defined in equation (9.1), with the constraint that $\|\eta\|_1 \leqslant D$ and $\|w_j\|_2^2 + b_j^2/R^2 = 1$ for all $j \in \{1, \dots, m\}$, with the ReLU activation function. If the loss function is G-Lipschitz-continuous, then, for $\hat{f}$ a minimizer of the empirical risk over $\mathcal{F}$,*

$$\mathbb{E}\Big[\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f)\Big] \leqslant \frac{16GDR}{\sqrt{n}}.$$

Proposition 9.1 will be combined with a study of the approximation properties in section 9.3, with a summary provided in section 9.4. We will see in chapter 12 some recent results showing how optimization algorithms add an implicit regularization that leads to provable generalization in overparameterized neural networks (i.e., networks with many hidden units).

For the estimation error, the number of parameters is irrelevant!
What counts is the overall norm of the weights.

Exercise 9.1 (♦) *Provide a bound similar to proposition 9.1 for the alternative constraint $\|w_j\|_1 + |b_j|/R = 1$, 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 given here for computing the Rademacher complexity can be extended by recursion to deeper networks and other activation functions, as exercise 9.2 shows (see, e.g., Neyshabur et al., 2015, for further results).

[3] Using $\sup_{\|w\|_2^2+b^2/R^2=1} z^\top w + t^\top b = \sup_{\|w\|_2^2+c^2=1} |z^\top w + (Rt)^\top c| = (\|z\|_2^2 + R^2t^2)^{1/2}$, from the Cauchy-Schwarz inequality.

Exercise 9.2 (♦) *We consider a* 1*-Lipschitz-continuous activation function* σ *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* $i = 1, \ldots, M$*,* $\mathcal{F}_i = \{x \mapsto \sum_{j=1}^{m_i} \theta_j \sigma(f_j(x)), f_j \in \mathcal{F}_{i-1}, \|\theta\|_1 \leqslant D_i\}$*, 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 in section 9.2.3, the estimation error for constrained output weights grows as $\|\eta\|_1/\sqrt{n}$, where η is the vector of output weights and is independent of the number m of neurons. Several 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 ℓ_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?

To do this, we need to understand the space of functions that neural networks span and how they relate to the smoothness properties of the function (as we did for kernel methods in chapter 7).

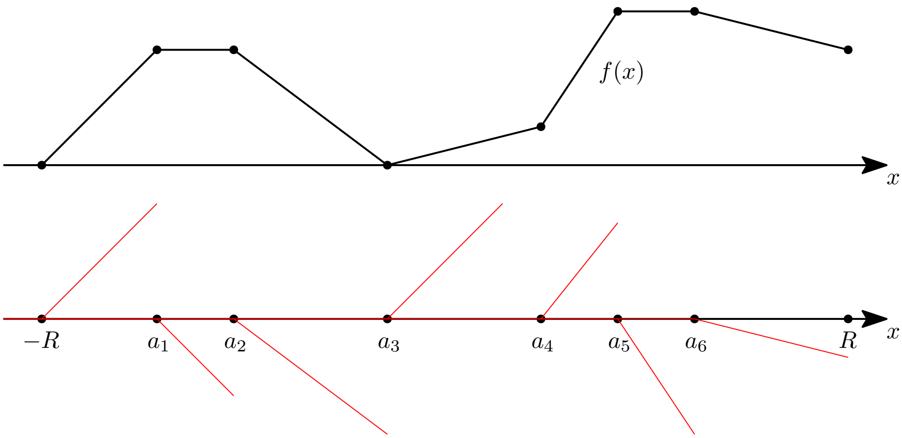
In this section, as in the previous section, we focus on the ReLU activation function, noting that universal approximation results exist as soon as the activation function is not a polynomial (Leshno et al., 1993). We start with a simple nonquantitative argument to show universality in one dimension (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 a number of simple, nonquantitative arguments.

Approximation of continuous piecewise affine functions. Since each individual function $x \mapsto \eta_j(w_j x + b_j)_+$ is continuous piecewise affine, the output of a neural network has to be continuous piecewise affine as well. It turns out that all continuous 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 here 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 for the change in slope of f. By pursuing this 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 using m neurons. This argument will be made more quantitative in section 9.3.3 by looking at the slopes of the piecewise affine function.

Universal approximation properties. Now that we can represent precisely all continuous 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 used to characterize the approximation. For example, continuous functions can be approximated by piecewise affine functions with arbitrary precision in the L_∞-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 ReLUs. 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 ℓ_1-norm of output weights, we first need to define the limit when the number of neurons diverges.

9.3.2 Infinitely Many Neurons and the Variation Norm

In this section, we consider neural networks of the form $f(x) = \sum_{j=1}^m \eta_j (w_j^\top x + b_j)_+$, 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 ℓ_2-sphere (but we will consider a slightly different set at the end of this section). Our goal is to define the set of functions that can be approximated by neural networks, while defining a norm on them that extends the ℓ_1-norm of the output weights. We consider $\mathcal{X}$ the d-dimensional ℓ_2-ball of radius R and center 0 (but the construction applies to any compact subset of $\mathbb{R}^d$).

Formulation through measures. We can write a neural network with finitely many neurons $f(x) = \sum_{j=1}^m \eta_j (w_j^\top x + b_j)_+$ as the integral

$$f(x) = \int_K (w^\top x + b)_+ d\nu(w, b), \tag{9.4}$$

for ν being the signed 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 ν.[4]

Since we want to have a norm $\|\eta\|_1$ which is as small as possible, among all representations of f as in equation (9.4), we look for the one for which $\int_K |d\nu(w, b)|$ is the smallest; that is, for $f \in \widetilde{\mathcal{F}}_1$ the set of neural networks with arbitrary (finite) width, we define

$$\tilde{\gamma}_1(f) = \inf_{\nu \in \widetilde{\mathcal{M}}(K)} \int_K |d\nu(w, b)| \text{ such that } \forall x \in \mathcal{X},\ f(x) = \int_K (w^\top x + b)_+ d\nu(w, b),$$

where $\widetilde{\mathcal{M}}(K)$ is the set of signed measures on K with *finite* support. This happens to define a norm on $\widetilde{\mathcal{F}}_1$. In order to extend beyond the set $\widetilde{\mathcal{F}}_1$ (which is equal to the set of continuous piecewise affine functions for $d = 1$), we simply relax the constraint of finite support for the measure ν. That is, for $f : \mathcal{X} \to \mathbb{R}$, we define

$$\gamma_1(f) = \inf_{\nu \in \mathcal{M}(K)} \int_K |d\nu(w, b)| \text{ such that } \forall x \in \mathcal{X},\ f(x) = \int_K (w^\top x + b)_+ d\nu(w, b), \tag{9.5}$$

where $\mathcal{M}(K)$ is the set of signed measures on K with finite total variation, with the convention that if no measure can be found to represent f, then $\gamma_1(f) = +\infty$. Proposition 9.2 shows that γ_1 defines a norm on the set $\mathcal{F}_1$ of functions such that $\gamma_1(f)$ is finite.

Proposition 9.2 *Assume $K \subset \mathbb{R}^{d+1}$ and $\mathcal{X} \subset \mathbb{R}^d$ are compact sets. The set $\mathcal{F}_1$ of functions such that $\gamma_1(f)$ defined in equation (9.5) is finite is a vector space, a subset of the set of Lipschitz-continuous functions on $\mathcal{X}$. Moreover, γ_1 is a norm on $\mathcal{F}_1$.*

Proof If $\gamma_1(f_1)$ and $\gamma_1(f_2)$ are finite, with f_1 and f_2 represented by measures ν_1 and ν_2, and $\lambda_1, \lambda_2 \in \mathbb{R}$, then $\lambda_1 f_1 + \lambda_2 f_2$ is represented by $\lambda_1 \nu_1 + \lambda_2 \nu_2$, with total variation $\int_K |d\nu(w, b)| \leqslant |\lambda_1| \int_K |d\nu_1(w, b)| + |\lambda_2| \int_K |d\nu_2(w, b)|$. Thus $\gamma(\lambda_1 f_1 + \lambda_2 f_2) \leqslant |\lambda_1|\gamma(f_1) + |\lambda_2|\gamma(f_2)$. This implies that the set $\mathcal{F}_1$ is a vector space and that γ_1 is convex. Moreover, γ_1 is absolutely homogeneous (i.e., $\gamma_1(\lambda f) = |\lambda|\gamma_1(f)$ for any $\lambda \in \mathbb{R}$)

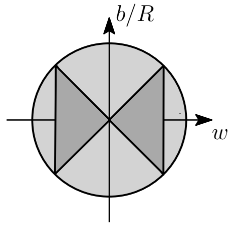
[4] When ν has density $d\nu/d\tau$ with respect to a base measure τ with full support in K, then the total variation is defined as the integral $\int_K |d\nu/d\tau(w, b)|d\tau(w, b)$ and is independent of the choice of τ. See https://en.wikipedia.org/wiki/Total_variation for more details.

and for any function f, $\sup_{x\in\mathcal{X}} |f(x)| \leqslant \gamma_1(f) \sup_{x\in\mathcal{X},(w,b)\in K} |w^\top x + b|$, which implies that if $\gamma_1(f) = 0$, then $f = 0$. Thus, γ_1 is a norm on $\mathcal{F}_1$. Finally, for any $x, y \in \mathcal{X}$, $|f(x)-f(y)| \leqslant \gamma_1(f) \sup_{(w,b)\in K} \|w\|_2 \cdot \|x-y\|_2$; therefore, all functions in $\mathcal{F}_1$ are Lipschitz-continuous. ∎

We then obtain a Banach space $\mathcal{F}_1$ of functions (the proof of completeness is left as a technical exercise), with a norm γ_1 that is often referred to as the "variation norm" (Kurková and Sanguineti, 2001). This characterizes the set of functions that can be asymptotically reached by neural networks with a bounded ℓ_1-norm of output weights, regardless of the number of neurons. The index 1 in γ_1 will become natural when we compare with the positive-definite kernels in section 9.5. Note that although we defined it for the ReLU activation, the same argument applies to all continuous activation functions. Finally, in order to obtain upper bounds on $\gamma_1(f)$, it suffices to represent f as an integral of neurons as in equation (9.5), and compute the corresponding total variation; for example, for a single neuron $f(x) = (w^\top x + b)_+$ for $(w, b) \in K$, $\gamma_1(f) \leqslant 1$, a property that will be used several times in section 9.3.3.

Note that due to the positive homogeneity of the ReLU activation function, the norm γ_1 does not change if we replace the compact set K with $\bigcup_{c\in[0,1]} cK$ (i.e., the union of all segments $[0, v]$ for $v \in K$), with a proof left as an exercise. Therefore, choosing the unit ℓ_2-sphere or the unit ℓ_2-ball for K gives the same results. (We will make a slightly different choice below.)

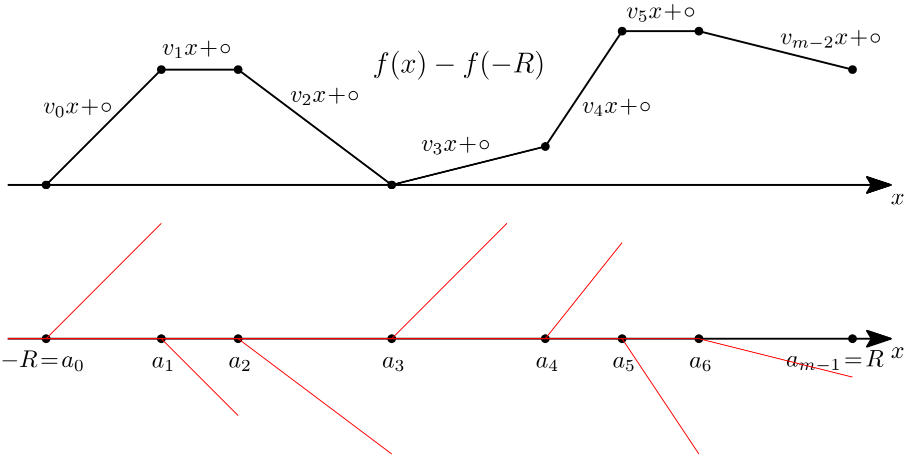
Studying the approximation properties of $\mathcal{F}_1$. Now that we have characterized the function space $\mathcal{F}_1$ through equation (9.5), we need to describe the set of functions with finite norm and relate this norm to classical smoothness properties (as done for kernel methods in chapter 7). To do so, as illustrated below, we consider a smaller set K than the unit ℓ_2-ball; that is, the set K 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 of $\sqrt{2}$, we consider the normalization $K = \{(w, b/R) \in \mathbb{R}^{d+1},\ \|w\|_2 = 1,\ |b| \leqslant R\}$ and the norm γ_1 defined in equation (9.5) with this set K. Note that for $d = 1$, we have $K = \{(w, b/R) \in \mathbb{R}^2,\ w \in \{-1, 1\},\ \text{and } |b| \leqslant R\}$, as illustrated below for $d = 1$ (with the new set $\bigcup_{c\in[0,1]} cK$ in dark gray, and the old one in light gray). We could stick to the ℓ_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 continuous 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) and Barron and Klusowski (2018).

Continuous 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 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 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. 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 $[a_0, a_{m-1}] = [-R, R]$ 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 follows (using that on the interval $[-R, R]$, $(x-a_0)_+ = (x+R)_+ = x+R$):

$$f(x) = f(-R) + v_0(x+R) + \sum_{j=1}^{m-2}(v_j - v_{j-1})(x-a_j)_+. \tag{9.6}$$

To obtain a representation that is invariant under a sign change, we also consider the same representation starting from the right (which can, for example, be obtained by applying equation (9.6) to $x \mapsto f(-x)$):

$$f(x) = f(R) - v_{m-2}(R-x) + \sum_{j=1}^{m-2}(v_j - v_{j-1})(a_j - x)_+. \tag{9.7}$$

Note that this also shows that such representations are not unique. By averaging equations (9.6) and (9.7), and using that $\frac{1}{2R}(x+R)_+ + \frac{1}{2R}(-x+R)_+$ is equal to 1 on $[-R, R]$, we get

$$\begin{aligned} f(x) \ &= \ \frac{1}{2}\big[f(R)+f(-R)\big]\Big[\frac{1}{2R}(x+R)_+ + \frac{1}{2R}(-x+R)_+\Big] \\ &+\frac{1}{2}v_0(x+R)_+ - \frac{1}{2}v_{m-2}(-x+R)_+ + \frac{1}{2}\sum_{j=1}^{m-2}(v_j - v_{j-1})\big[(x-a_j)_+ + (a_j-x)_+\big], \end{aligned}$$

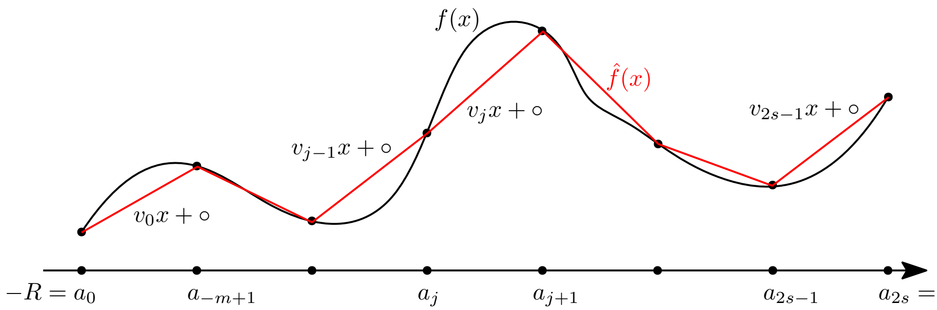
and thus, by construction of norm γ_1, we have

$$\gamma_1(f) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)] + v_0\Big| + \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)] - v_{m-2}\Big| + \sum_{j=1}^{m-2}|v_j - v_{j-1}|.$$

The norm is thus upper-bounded by the values of f and its derivatives at the boundaries of the interval and the sums of changes in slope.

Twice continuously differentiable functions. Now 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 Taylor's 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 $\hat{f}$ from values $a_j, f(a_j)$ (and slopes v_j on $[a_j, a_{j+1}]$), with $j \in \{0, \dots, 2s\}$, for s that will tend to infinity (see the following illustration, where we have $m - 1 = 2s$):



For the piecewise affine approximant $\hat{f}$, the slope v_0 on $[a_0, a_1]$ is equal to $v_0 = \frac{s}{R}[f(-R + R/s) - f(-R)] \sim f'(-R)$, and the slope v_{2s-1} on $[a_{2s-1}, a_2 s]$ is equal to $v_{2s-1} = \frac{s}{R}[f(R) - f(R - R/s)] \sim f'(R)$ when s tends to infinity, while the differences in slopes $|v_j - v_{j-1}|$ equal

$$\begin{aligned} & \Big|\tfrac{s}{R}\big(f(-R + \tfrac{j+1}{s}R) - f(-R + \tfrac{j}{s}R)\big) - \tfrac{s}{R}\big(f(-R + \tfrac{j}{s}R) - f(-R + \tfrac{j-1}{s}R)\big)\Big| \\ = \; & \tfrac{s}{R}\big|f(-R + \tfrac{j+1}{s}R) - 2f(-R + \tfrac{j}{s}R) + f(-R + \tfrac{j-1}{s}R)\big|, \end{aligned}$$

which is equivalent to $\frac{R}{s}\big|f''(-R + \frac{j}{s}R)\big|$ when $s \to +\infty$ (using a second-order Taylor expansion, where zeroth and first order terms vanish). A fully rigorous proof that takes into account the fact that the range of indices j depends on s is left as an exercise.

Thus, the approximant $\hat{f}$ has a γ_1-norm $\gamma_1(\hat{f})$ upper-bounded asymptotically by

$$\frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+f'(-R)\Big|+\frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-f'(R)\Big|+\frac{R}{s}\sum_{j=1}^{2s-1}\big|f''(-R+\frac{j}{s}R)\big|.$$

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, as the reasoning given next will make it more formal)

$$\gamma_1(f) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+f'(-R)\Big|+\frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-f'(R)\Big|+\int_{-R}^{R}|f''(x)|dx. \tag{9.8}$$

This notably shows that although the number of neurons is allowed to grow, the ℓ_1-norm of the weights remains bounded by the quantity in equation (9.8).

Direct proof through Taylor's formula. Equation (9.8) can be extended to continuously differentiable functions, which are only twice differentiable almost everywhere with integrable second-order derivatives. In this section, we assume that function f is twice continuously differentiable but we could extend to only integrable second derivatives by a density argument (see, e.g., Rudin, 1987). For such a function, using Taylor's formula with integral remainder, we have, for $x \in [-R, R]$, using the fact that $(x - b)_+ = 0$ as soon as $b \geqslant x$,

$$\begin{aligned} f(x) &= f(-R) + f'(-R)(x+R) + \int_{-R}^{x} f''(b)(x-b)db \\ &= f(-R) + f'(-R)(x+R) + \int_{-R}^{R} f''(b)(x-b)_+ db. \end{aligned} \tag{9.9}$$

We also have the symmetric version (obtained by applying equation (9.9) to $x \mapsto f(-x)$, replacing x by $-x$, and by making a change of variable $b \to -b$ in the integral) as follows:

$$f(x) = f(R) - f'(R)(R-x) - \int_{-R}^{R} f''(b)(-x+b)_+ db.$$

By averaging the two equalities an using that $\frac{1}{2R}(x+R) + \frac{1}{2R}(R-x) = 1$, we get

$$\begin{aligned} f(x) &= \frac{1}{2}\Big[\frac{f(-R)+f(R)}{2R} + f'(-R)\Big](x+R) + \frac{1}{2}\Big[\frac{f(-R)+f(R)}{2R} - f'(R)\Big](R-x) \\ &\quad + \frac{1}{2}\int_{-R}^{R} f''(b)(x-b)_+ db - \frac{1}{2}\int_{-R}^{R} f''(b)(-x+b)_+ db. \end{aligned}$$

This leads to the exact same upper bound on $\gamma_1(f)$ as obtained from piecewise affine interpolation:

$$\gamma_1(f) \leqslant \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]+f'(-R)\Big| + \frac{1}{2}\Big|\frac{1}{2R}[f(-R)+f(R)]-f'(R)\Big| + \int_{-R}^{R} |f''(x)|dx. \tag{9.10}$$

One can check that the upper bound is indeed a norm (proof left as an exercise).

We will also use a simpler upper bound, obtained from the triangle inequality:

$$\gamma_1(f) \leqslant \frac{1}{2R}\big|f(-R)+f(R)\big| + \frac{1}{2}|f'(R)| + \frac{1}{2}|f'(-R)| + \int_{-R}^{R} |f''(x)|dx. \tag{9.11}$$

Exercise 9.3 (♦♦) *Assume $-R = x_1 < \cdots < x_n = R$, $y_1, \dots, y_n \in \mathbb{R}$, show that the piecewise-affine interpolant on $[-R, R]$ is a minimum norm interpolant.*

9.3.4 Variation Norm in an Arbitrary Dimension

In order to extend to larger dimensions than $d = 1$, we will use Fourier transforms. This requires to consider functions on $\mathcal{X}$, the ball with center zero and radius R, as restrictions of functions defined on $\mathbb{R}^d$ with compact support (so that they belong to $L_2(\mathbb{R}^d)$, the space of square-integrable functions for the Lebesgue measure, and $L_1(\mathbb{R}^d)$ the space of integrable functions); this can be done in a number of ways (see Rudin, 1987 and the end of section 7.5.2).

Since $f \in L_1(\mathbb{R}^d)$, the Fourier transform $\hat{f}(\omega) = \int_{\mathbb{R}^d} f(x) e^{-i\omega^\top x} dx$ is defined everywhere and continuous, and, assuming that Fourier transform is integrable, we can write f as the inverse Fourier transform of $\hat{f}$; that is, for all $x \in \mathbb{R}^d$ (and thus for $x \in \mathcal{X}$),

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^\top x} d\omega. \tag{9.12}$$

To compute an upper bound on $\gamma_1(f)$, it suffices to upper-bound for each $\omega \in \mathbb{R}^d$ the norm $\gamma_1(x \mapsto e^{i\omega^\top x})$ (using complex-valued functions, for which the developments of the section 9.3.3 still apply, or using sines and cosines), which is possible because we can represent the function $g^{(\rho)} : u \mapsto e^{iu\rho}$, for $u \in [-R, R]$ using section 9.3.3 and equation (9.11); that is, we obtain two measures on $[-R, R]$, $\nu_+^{(\rho)}$ and $\nu_-^{(\rho)}$, such that for all $u \in [-R, R]$,

$$e^{iu\rho} = \int_{-R}^{R} (u - b)_+ d\nu_+^{(\rho)}(b) + \int_{-R}^{R} (-u - b)_+ d\nu_-^{(\rho)}(b),$$

with

$$\begin{aligned}
&\int_{-R}^{R} |d\nu_+^{(\rho)}(b)| + \int_{-R}^{R} |d\nu_-^{(\rho)}(b)| \\
\leqslant\ & \frac{1}{2R}\big|g^{(\rho)}(-R) + g^{(\rho)}(R)\big| + \frac{1}{2}|(g^{(\rho)})'(R)| + \frac{1}{2}|(g^{(\rho)})'(-R)| + \int_{-R}^{R} |(g^{(\rho)})''(x)| dx \\
\leqslant\ & \frac{1}{R} + \rho + 2R\rho^2 \leqslant \frac{1}{R} + 2R\rho^2 + \big(\frac{1}{2R} + \frac{1}{2}R\rho^2\big) \leqslant \frac{2}{R}(1 + 2R^2\rho^2),
\end{aligned} \tag{9.13}$$

using $\rho \leqslant \frac{1}{2R} + \frac{1}{2}R\rho^2$. We can, therefore, decompose the function defined on the ball with center 0 and radius R:

$$\begin{aligned}
e^{i\omega^\top x} &= e^{i(x^\top \omega/\|\omega\|_2)\|\omega\|_2} \\
&= \int_{-R}^{R} (x^\top(\omega/\|\omega\|_2) - b)_+ d\eta_+^{(\|\omega\|_2)}(b) + \int_{-R}^{R} (x^\top(-\omega/\|\omega\|_2) - b)_+ d\eta_-^{(\|\omega\|_2)}(b),
\end{aligned}$$

with weights being in the correct constraint set (unit norm for the slopes $\omega/\|\omega\|_2$ and constant terms $|b| \leqslant R$), leading to, using equation (9.13),

$$\gamma_1(x \mapsto e^{i\omega^\top x}) \leqslant \int_{-R}^{R} |d\nu_+^{(\|\omega\|_2)}(b)| + \int_{-R}^{R} |d\nu_+^{(\|\omega\|_2)}(b)| \leqslant \frac{2}{R}(1 + 2R^2\|\omega\|_2^2).$$

Thus, we obtain, from equation (9.12) and the triangular inequality for norm γ_1,

$$\gamma_1(f) \leqslant \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\hat{f}(\omega)| \gamma_1(x \mapsto e^{i\omega^\top x}) d\omega \leqslant \frac{1}{(2\pi)^d} \frac{2}{R} \int_{\mathbb{R}^d} |\hat{f}(\omega)|(1 + 2R^2\|\omega\|_2^2) d\omega. \tag{9.14}$$

Given function $g : \mathbb{R}^d \to \mathbb{R}$, $\int_{\mathbb{R}^d} |\hat{g}(\omega)| d\omega$ is a measure of smoothness of g, so $\gamma_1(f)$ being finite imposes that f and all second-order derivatives of f have this form of smoothness.

The right side of equation (9.14) is often referred to as the "Barron norm," which is named after Barron (1993, 1994). See Klusowski and Barron (2018) for more details.

To relate norm γ_1 to other function spaces such as Sobolev spaces, we will consider further upper bounds (and relate them to another norm γ_2, described in section 9.5).

Exercise 9.4 (Step activation function (♦)) *Consider the step activation function defined as $\sigma(u) = 1_{u>0}$. Show that the corresponding variation norm can be upper-bounded by a constant times $\int_{\mathbb{R}^d}|\hat{f}(\omega)|(1 + R\|\omega\|_2)d\omega$.*

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 the Cauchy-Schwarz inequality, the norm γ_1 as

$$\begin{aligned}
\gamma_1(f) &\leqslant \frac{1}{(2\pi)^d}\frac{2}{R}\int_{\mathbb{R}^d}|\hat{f}(\omega)|(1+2R^2\|\omega\|_2^2)d\omega \quad \text{from equation (9.14)},\\
&= \frac{1}{(2\pi)^d}\frac{2}{R}\int_{\mathbb{R}^d}|\hat{f}(\omega)|(1+2R^2\|\omega\|_2^2)^{d/4+5/4}\frac{d\omega}{(1+2R^2\|\omega\|_2^2)^{d/4+1/4}}\\
&\leqslant \frac{1}{(2\pi)^d}\frac{2}{R}\sqrt{\int_{\mathbb{R}^d}|\hat{f}(\omega)|^2(1+2R^2\|\omega\|_2^2)^{d/2+5/2}d\omega}\sqrt{\int_{\mathbb{R}^d}\frac{d\omega}{(1+2R^2\|\omega\|_2^2)^{d/2+1/2}}}, \quad (9.15)
\end{aligned}$$

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, as well as 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 the $L_2(\mathbb{R}^d)$-norm ε requires a γ_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 next.

Linear latent variables. We consider a target function f_* that depends only on a r-dimensional projection of the data, with $r < d$; that is, f_* is of the form $f_*(x) = g(V^\top x)$, where $V \in \mathbb{R}^{d\times r}$ has 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 $\mathbb{R}^r$), one can write

$$g(z) = \int_{\mathbb{R}^{r+1}}(w^\top z + b)_+ d\mu(w, b),$$

with μ supported on $\{(w,b) \in \mathbb{R}^{r+1},\ \|w\|_2 = 1, |b| \leqslant R\}$, 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 V has orthonormal columns, $\|Vw\|_2 = 1$ as soon as $\|w\|_2 = 1$; therefore, the measure μ on $(w,b) \in \mathbb{R}^{r+1}$ defines a measure for $(Vw,b) \in \mathbb{R}^{d+1}$ on $\{(w',b) \in \mathbb{R}^{d+1},\ \|w'\|_2 = 1, |b| \leqslant R\}$, which is supported in the subspace spanned by the columns of V and has a total variation that is less than that of μ.[5] Thus, we have $\gamma_1(f_*) \leqslant \int_{\mathbb{R}^{r+1}} |d\mu(w,b)| = \gamma_1(g)$. 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.

For example, (1) if g has more than $r/2+5/2$ square-integrable derivatives, then $\gamma_1(g)$ and thus $\gamma_1(f_*)$ is finite, or (2) if g is Lipschitz-continuous, then both g and f can be approached in $L_2(\mathbb{R}^d)$ with error ε with a function with a γ_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 cannot use the dependence on a linear latent variable to reduce the approximation error. In other words, as shown in section 9.5, using the ℓ_2-norm instead of the ℓ_1-norm on the output weights leads to worse performance when such linear latent variables are present.

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 probability measure p on $\mathcal{X} \subset \mathbb{R}^d$, and a function $g : \mathcal{X} \to \mathbb{R}$ 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

$$f(x) = \sum_{j=1}^m \eta_j (w_j^\top x + b_j)_+$$

is close to g for the norm $L_2(p)$.

Since input weights are fixed in K, the bound on $\gamma_1(g)$ should translate into a bound on the ℓ_1-norm of η: $\|\eta\|_1 \leqslant \gamma_1(g)$. The set of functions f such that $\gamma_1(f) \leqslant \gamma_1(g)$ is the convex hull of functions $s\gamma_1(g)(w^\top x + b)_+$, for $s \in \{-1, 1\}$, as well as $\|w\|_2 = 1$, $|b| \leqslant R$. Thus, we are faced with the problem of approximating elements of a convex hull as an explicit linear combination of extreme points with as few extreme points as possible.

[5] We use here the property that the total variation of a measure is equal to the total variation of this same measure restricted to its support.

In a finite dimension, Carathéodory's theorem[6] says that the number of such extreme points can be taken as 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) $\|f-g\|_{L_2(p)}^2$ 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 (aka conditional gradient algorithm). This algorithm is applicable more generally; for more details, see Jaggi (2013) and 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 is 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):

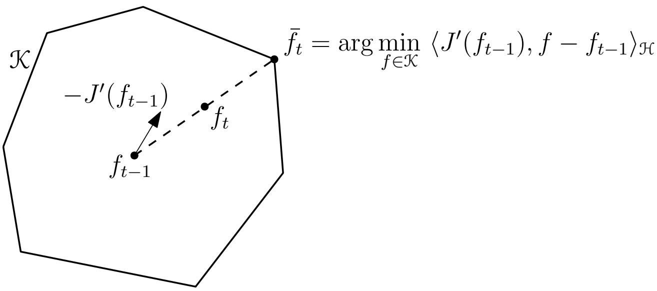
$$J(g) + \langle J'(g), f-g \rangle_{\mathcal{H}} \ \leqslant \ J(f) \ \leqslant \ J(g) + \langle J'(g), f-g \rangle_{\mathcal{H}} + \frac{L}{2}\|f-g\|_{\mathcal{H}}^2.$$

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 was required in section 5.2.5.

We consider the following recursive algorithm, starting from any vector $f_0 \in \mathcal{K}$:

$$\bar{f}_t \ \in \ \arg\min_{f \in \mathcal{K}} \ \langle J'(f_{t-1}), f - f_{t-1} \rangle_{\mathcal{H}}, \tag{9.16}$$

$$f_t \ = \ \frac{t-1}{t+1} f_{t-1} + \frac{2}{t+1}\bar{f}_t = f_{t-1} + \frac{2}{t+1}(\bar{f}_t - f_{t-1}). \tag{9.17}$$



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 an extreme point of $\mathcal{K}$ so that f_t is the convex combination of t 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

$$J(f_t) - \inf_{f \in \mathcal{K}} J(f) \leqslant \frac{2L}{t+1} \mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2.$$

[6] See https://en.wikipedia.org/wiki/Caratheodory's_theorem_(convex_hull).

Proof of convergence rate (⧫). This is obtained by using smoothness as follows:

$$\begin{aligned}
J(f_t) &\leqslant J(f_{t-1}) + \langle J'(f_{t-1}), f_t - f_{t-1}\rangle_{\mathcal{H}} + \frac{L}{2}\|f_t - f_{t-1}\|_{\mathcal{H}}^2 \\
&= J(f_{t-1}) + \frac{2}{t+1}\langle J'(f_{t-1}), \bar{f}_t - f_{t-1}\rangle_{\mathcal{H}} + \frac{4}{(t+1)^2}\frac{L}{2}\|\bar{f}_t - f_{t-1}\|_{\mathcal{H}}^2 \\
&\leqslant J(f_{t-1}) + \frac{2}{t+1}\inf_{f\in\mathcal{K}}\langle J'(f_{t-1}), f - f_{t-1}\rangle_{\mathcal{H}} + \frac{4}{(t+1)^2}\frac{L}{2}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2,
\end{aligned}$$

by definition of $\bar{f}_t$ in equation (9.16). By convexity of J, we have for all element $f \in \mathcal{K}$, $J(f) \geqslant J(f_{t-1}) + \langle J'(f_{t-1}), f - f_{t-1}\rangle_{\mathcal{H}}$, 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

$$J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \big[J(f_{t-1}) - \inf_{f\in\mathcal{K}} J(f)\big]\Big(1 - \frac{2}{t+1}\Big) + \frac{4}{(t+1)^2}\frac{L}{2}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2,$$

leading to

$$\begin{aligned}
t(t+1)\big[J(f_t) - \inf_{f\in\mathcal{K}} J(f)\big] &\leqslant (t-1)t\big[J(f_{t-1}) - \inf_{f\in\mathcal{K}} J(f)\big] + 2L\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2 \\
&\leqslant 2Lt\,\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2 \text{ by using a telescoping sum,}
\end{aligned}$$

and thus $J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \dfrac{2L}{t+1}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2$, as claimed earlier.

Exercise 9.5 *Show that if we replace equation (9.17) with* $f_t = \frac{t-1}{t} f_{t-1} + \frac{1}{t}\bar{f}_t$, f_t *is the uniform convex combination of* $\bar{f}_1, \dots, \bar{f}_t$, *and we have the convergence rate* $J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \frac{L}{t}(1 + \log t)\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2$.

Exercise 9.6 (Frank-Wolfe with line search) *The update in equation (9.17) is often replaced by* $f_t = (1-\rho_t)f_{t-1} + \rho_t\bar{f}_t$ *with* $\rho_t = \arg\min_{\rho\in[0,1]} \rho\langle J'(f_{t-1}), \bar{f}_t - f_{t-1}\rangle_{\mathcal{H}} + \frac{L}{2}\rho^2\|\bar{f}_t - f_{t-1}\|_{\mathcal{H}}^2$. *Show that we have* $J(f_t) - \inf_{f\in\mathcal{K}} J(f) \leqslant \frac{4L}{t+1}\mathrm{diam}_{\mathcal{H}}(\mathcal{K})^2$.

Application to approximate representations with a finite number of neurons. We can apply this to $\mathcal{H} = L_2(p)$ and $J(f) = \|f - g\|_{L_2(p)}^2$, leading to $L = 2$, with $\mathcal{K} = \{f \in L_2(p),\ \gamma_1(f) \leqslant \gamma_1(g)\}$, which is the convex hull of single neurons $s(w^\top \cdot + b)_+$ scaled by $\gamma_1(g)$ and with an extra sign $s \in \{-1, 1\}$.

We thus obtain after t steps a function f_t that can be represented with t neurons for which

$$\|f_t - g\|_{L_2(p)}^2 \leqslant \frac{16\gamma_1(g)^2}{t+1}\sup_{(w,b)\in K}\|(w^\top \cdot + b)_+\|_{L_2(p)}^2. \tag{9.18}$$

Thus, it is sufficient to have t of order $O(\gamma_1(g)^2/\varepsilon^2)$ to achieve $\|f_t - g\|_{L_2(p)} \leqslant \varepsilon$. 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 in equation (9.18) becomes $\|f_t - g\|_{L_2(p)}^2 \leqslant \frac{16\gamma_1(g)^2}{t+1}(2R)^2$; note that the dependence of the number of neurons in ε as ε^{-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 γ_1 bounded by a fixed constant D. After t iterations, the general convergence result given here leads to an approximate minimizer with an explicit provable convergence guarantee in $O(1/t)$.

However, as discussed previously, the linear minimization oracle requires optimizing with respect to single neurons of the form $s(w^\top \cdot + b)_+$ scaled by D and with an extra sign $s \in \{-1, 1\}$. Therefore, to implement the linear minimization oracle, given the derivative α_i of the loss function associated with the ith observation, for $i = 1, \dots, n$, we need to minimize with respect to s, w, and b the quantity $\sum_{i=1}^n s\alpha_i (w^\top x_i + b)_+$, 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 in Bach, 2017).

This incremental approach to estimating a neural network is related to the boosting procedures that we present in section 10.3.

Exercise 9.7 *Extend the bound in equation (9.18) to all activation functions.*

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, which 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 minimize the empirical risk for a G-Lipschitz-continuous loss subject to $\gamma_1(f) \leqslant D$. Proposition 9.1 leads to 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 equation (7.13) because the extra factor of 4, i.e., 16 instead of 4, in the estimation error):

$$\varepsilon_n = G \sqrt{2 \inf_{f \in \mathcal{F}_1} \Big\{ \|f - f_*\|_{L_2(p)}^2 + \frac{256 R^2}{n} \gamma_1(f)^2 \Big\}}. \tag{9.19}$$

As shown in section 7.5.1, given this bound, we can recover the bound D as $\frac{\sqrt{n}}{16RG\sqrt{2}} \varepsilon_n$, 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 ε_n with a number of neurons m greater than a constant times $D^2 R^2 G^2 / \varepsilon_n^2$ (see equation (9.18)), 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 with 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 γ_1 with the Sobolev norm of order $s = \frac{d}{2} + \frac{5}{2}$ in equation (9.15), 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; it cannot be much improved anyway, 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 γ_1 projected to the subspace (as 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 to $1/n^{1/(r+5)}$ (with constants independent of d). This is where neural networks have a strong advantage over kernel methods and sparse methods: they are adaptive to linear latent variables and can thus perform variable selection with nonlinear predictions (as detailed next).
- "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 less than a constant times to k.

Exercise 9.8 *Consider target functions of the form* $f_*(x) = \sum_{j=1}^k f_j(w_j^\top x)$ *for one-dimensional Lipschitz-continuous functions* $f_1, \dots, f_k$*. 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.

Nonlinear variable selection (♦). In this chapter, we focused primarily on ℓ_2-norm constraints or penalties 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 ℓ_1-norm, leading to the high-dimensional behavior detailed in section 8.3.3, but this time selecting variables with a *nonlinear* prediction on top of them. In terms of algorithms, we would need to replace (stochastic) gradient descent on $w_1, \dots, w_m$ by proximal extension (as detailed in section 5.2.5). 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 the constraints $\|w_j\|_2^2 + b_j^2/R^2 = 1$ by $\|w_j\|_1^2 + b_j^2/R^2 = 1$, and thus replace equation (9.3) with

$$\begin{aligned}
\mathrm{R}_n(\mathcal{G}) &\leqslant 2GD\bigg(\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_\infty^2 + R^2\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big]\bigg)^{1/2} \qquad (9.20)\\
&\leqslant 2GD\bigg(\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i x_i\Big\|_\infty^2\Big]\bigg)^{1/2} + 2GDR\bigg(\mathbb{E}\Big[\Big(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\Big)^2\Big]\bigg)^{1/2}\\
&\leqslant 2GDR\frac{\sqrt{2\log(2d)}}{\sqrt{n}} + \frac{2GDR}{\sqrt{n}} \leqslant 4GRD\sqrt{\frac{\log(4d)}{n}},
\end{aligned}$$

using expectations of maxima from section 1.2.4.

Thus, in estimation rates, we need to consider, instead of equation (9.19),

$$\varepsilon_n = 2G\sqrt{\inf_{f\in\mathcal{F}_1}\Big\{\|f-f_*\|_{L_2(p)}^2 + \frac{256R^2\log(4d)}{n}\gamma_1(f)^2\Big\}}$$

(Note the extra factor $\log(4d)$ and the definition of R as an ℓ_∞-bound.) Regarding approximation error, we simply use the bound $\|w\|_1 \leqslant \sqrt{k}\|w\|_2$ if w has only k nonzero elements. Thus, if the target function f_* is a Lipschitz-continuous function of only k unknown variables, we can use the approximation result for ℓ_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 $\big(\frac{\log(4d)}{n}\big)^{1/(k+3)}$, and 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 only to 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 τ with full support on $K \subset \mathbb{R}^{d+1}$, we can define another norm as

$$\gamma_2^2(f) = \inf_{\nu\in\mathcal{M}(K)}\int_K\Big|\frac{d\nu(w,b)}{d\tau(w,b)}\Big|^2 d\tau(w,b) \text{ such that } \forall x\in\mathcal{X},\ f(x) = \int_K (w^\top x+b)_+ d\nu(w,b). \qquad (9.21)$$

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 $\mathcal{F}_1$ (in addition, γ_2 depends on the choice of the base measure τ, while γ_1 does not).

Moreover, as shown in proposition 9.3, the space $\mathcal{F}_2$ is an RKHS on the set $\mathcal{X} = \{x \in \mathbb{R}^d, \|x\|_2 \leqslant R\}$, as defined in chapter 7.

Proposition 9.3 *The space $\mathcal{F}_2$ is the RKHS associated with the positive-definite kernel function*

$$k(x, x') = \int_K (w^\top x + b)_+ (w^\top x' + b)_+ d\tau(w, b). \tag{9.22}$$

Proof For a formal proof for all compact sets K, see Bach (2017, appendix A). We only provide a proof for finite K and τ the uniform probability measure on K, we then have $\gamma_2^2(f) = \inf_{\nu \in \mathbb{R}^K} \frac{1}{|K|} \sum_{(w,b)\in K} \nu_{(w,b)}^2$, such that $f(x) = \frac{1}{|K|} \sum_{(w,b)\in K} \nu_{(w,b)} (w^\top x + b)_+$, which corresponds to penalizing the ℓ_2-norm of $\theta = \frac{1}{\sqrt{|K|}} \nu \in \mathbb{R}^K$ for $f(x) = \theta^\top \varphi(x)$ and $\varphi(x)_{(w,b)} = \frac{1}{|K|^{1/2}} (w^\top x + b)_+$. We thus exactly get the desired kernel $k(x, x') = \frac{1}{|K|} \sum_{(w,b)\in K} (w^\top x + b)_+ (w^\top x' + b)_+$. ■

Interpretation in terms of random features. As already mentioned in section 7.4, the kernel defined in equation (9.22) can be approximated by sampling from τ, m points (w_j, b_j), $j = 1, \dots, m$, and approximating $k(x, x')$ by

$$\hat{k}(x, x') = \frac{1}{m} \sum_{j=1}^{m} (w_j^\top x + b_j)_+ (w_j^\top x' + b_j)_+.$$

This corresponds to using $f(x) = \sum_{j=1}^{m} \eta_j (w_j^\top x + b_j)_+$, 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 γ_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 distributions of weights; see section 9.5.2, Cho and Saul (2009), and 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, 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 η_j are optimized. Full optimization of all weights can thus be seen as learning the kernel function.

Because Dirac measures are not square-integrable with respect to the Lebesgue measure, the prediction function $x \mapsto (w^\top x + b)_+$ (i.e., a single neuron that belongs to $\mathcal{F}_1$) is typically not in the RKHS $\mathcal{F}_2$, which is typically composed of smooth functions.

Table 9.1. Summary of properties of th norms γ_1 and γ_2.

$\mathcal{F}_2$	$\mathcal{F}_1$
Hilbert Space	Banach Space
$\gamma_2(f)^2 = \inf \int_{\mathbb{R}^{d+1}} \lvert\eta(w,b)\rvert^2 d\tau(w,b)$ s. t. $f(x) = \int_{\mathbb{R}^{d+1}} \eta(w,b)(w^\top x+b)_+ d\tau(w,b)$	$\gamma_1(f) = \inf \int_{\mathbb{R}^{d+1}} \lvert\eta(w,b)\rvert d\tau(w,b)$ s. t. $f(x) = \int_{\mathbb{R}^{d+1}} \eta(w,b)(w^\top x+b)_+ d\tau(w,b)$
Smooth functions Single neurons $\notin \mathcal{F}_2$	Potentially nonsmooth functions Single neurons $\in \mathcal{F}_1$

Link between the two norms. To relate the two norms more precisely, we rewrite γ_1 using the fixed probability measure τ (assuming it has full support in K) as

$$\gamma_1(f) = \inf_{\eta: K \to \mathbb{R}} \int_K |\eta(w,b)| d\tau(w,b) \text{ such that } \forall x \in \mathcal{X}, f(x) = \int_K (w^\top x+b)_+ \eta(w,b) d\tau(w,b).$$

The only difference with the squared RKHS norm is that we consider the L_1-norm instead of the squared L_2-norm of η (with respect to the probability measure τ). The minimum achievable norm is exactly $\gamma_1(f)$.

Note that typically, the infimum over all η is not achieved, as the optimal measure in equation (9.5) may not have a density with respect to τ. 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 $\int_{\mathbb{R}^{d+1}} |d\mu(\eta,b)| = \int_{\mathbb{R}^{d+1}} |\eta(w,b)| d\tau(w,b)$.

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 γ_1-norm less than 1, as the mass of a Dirac measure is equal to 1.

Another link between the two norms can be established by seeing the norm γ_1 as the optimization of the norm γ_2 with respect to the base measure τ, which can be seen as a form of kernel learning (see Gönen and Alpaydın, 2011, and exercise 9.9).

Exercise 9.9 (Link with kernel learning (♦)) *With the setup presented in this section, show that the infimum of* $\int_K \left|\frac{d\nu(w,b)}{d\tau(w,b)}\right|^2 d\tau(w,b)$ *over probability distributions* τ *on* K *is equal to* $\left(\int_K |d\nu(w,b)|\right)^2$. *Using exercise 8.16, show how the penalty* γ_1 *can be interpreted as kernel learning.*

9.5.2 Kernel Function (♦♦)

For the ReLU activation function, we can compute in closed form the kernel function, which is useful computationally only 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).

Dimension $d = 1$. In one dimension, with w uniform on the unit sphere (i.e., $w \in \{-1, 1\}$), and with b uniform on $[-R, R]$, we have the following kernel:

$$k(x, x') = \frac{1}{4R} \int_{-R}^{R} \Big((x - b)_+ (x' - b)_+ + (-x - b)_+ (-x' - b)_+ \Big) db.$$

We can compute it in closed form as (assuming $x \leqslant x'$)

$$\begin{aligned}
k(x, x') &= \frac{1}{4R} \int_{-R}^{x} (x - b)(x' - b) db + \frac{1}{4R} \int_{-R}^{-x'} (-x - b)(-x' - b) db \\
&= \frac{1}{4R} \Big[xx'(x + R) - (x + x')\big(\frac{x^2}{2} - \frac{R^2}{2}\big) + \frac{x^3}{3} - \frac{R^3}{3} \\
&\qquad + xx'(-x' + R) + (x + x')\big(\frac{(x')^2}{2} - \frac{R^2}{2}\big) - \frac{(x')^3}{3} - \frac{R^3}{3} \Big] \\
&= \frac{R^2}{6} + \frac{xx'}{2} + \frac{1}{24R}(x' - x')^3.
\end{aligned}$$

This leads to

$$k(x, x') = \frac{R^2}{6} + \frac{xx'}{2} + \frac{1}{24R}|x - x'|^3. \tag{9.23}$$

Generalization to all dimensions. In higher dimension, we can use the one-dimensional expression in equation (9.23), with τ the uniform distribution on the sphere:

$$\begin{aligned}
k(x, x') &= \int_{\|w\|_2 = 1} \frac{1}{2R} \Big(\int_{-R}^{R} (w^\top x + b)_+ (w^\top x' + b)_+ db \Big) d\tau(w) \\
&= \mathbb{E}_{\tau(w)} \big[k(w^\top x, w^\top x') \big] = \mathbb{E}_{\tau(w)} \Big[\frac{R^2}{6} + \frac{w^\top x (w^\top x')}{2} + \frac{1}{24R} |w^\top x - w^\top x'|^3 \Big] \\
&= \frac{R^2}{6} + \frac{x^\top x'}{2} \cdot \mathbb{E}_{\tau(w)} \big[|w_1|^2 \big] + \frac{1}{24R} \|x - x'\|_2^3 \cdot \mathbb{E}_{\tau(w)} \big[|w_1|^3 \big],
\end{aligned}$$

by invariance by rotation. The variable $|w_1|^2 \in [0, 1]$ is distributed as a Beta[7] random variable with parameters $(1/2, (d - 1)/2)$. Thus $\mathbb{E}_{\tau(w)}\big[|w_1|^2\big] = 1/d$ and $\mathbb{E}_{\tau(w)}\big[|w_1|^3\big] = \frac{\Gamma(2)\Gamma(\frac{d}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{d}{2} + \frac{3}{2})}$, leading to (see Bach, 2023, for more details and extensions):

$$k(x, x') = \frac{R^2}{6} + \frac{1}{2d} x^\top x' + \frac{1}{24R} \frac{\Gamma(2)\Gamma(\frac{d}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{d}{2} + \frac{3}{2})} \|x - x'\|_2^3. \tag{9.24}$$

Note that the expression differs from what was obtained in section 7.4.3 because we consider here a constant term. See figure 9.2 for examples of comparing the RKHS

[7] See https://en.wikipedia.org/wiki/Beta_distribution.

(corresponding to $m = +\infty$ neurons) and the approximation with finite m. We see that for large m (right plot), the linear combination of single neurons provides a good approximation of the associated kernel, but not for small m (left plot).

Exercise 9.10 (Step activation function) *Consider, instead of equation (9.22), the kernel* $k(x,x') = \int_K 1_{w^\top x+b\geqslant 0} 1_{w^\top x'+b\geqslant 0} d\tau(w,b)$. *Show that it can be expressed in closed form as* $k(x,x') = \frac{1}{2} - \frac{1}{4R}\frac{\Gamma(1)\Gamma(\frac{d}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{d}{2}+\frac{1}{2})}\|x-y\|_2$.

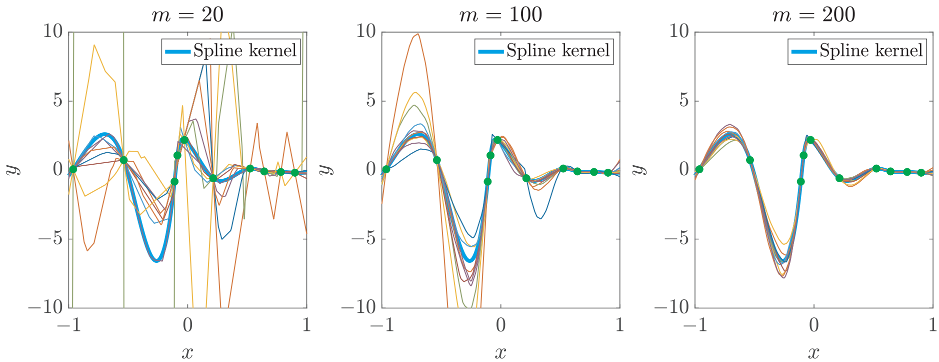


Figure 9.2. Examples of functions in the RKHS $\mathcal{F}_2$ and its approximation based on random features, with $m = 20, 100$, and 200; that is, functions that are linear combinations of $(w_j^\top x + b_j)_+$, $j = 1, \dots, m$, where (w_j, b_j) are independently sampled. All the functions are the 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 achieved by the piecewise affine interpolator (see exercise 9.3), and can be achieved with $m = n$ (well selected and thus nonrandom) neurons, where n is the number of observed points.

9.5.3 Upper Bound on RKHS Norm (♦♦)

We can now find upper bounds on norm γ_2. We can either use the kernel function from equation (9.24) or the random feature interpretation from equation (9.21). We first use the random feature interpretation in one dimension.

Upper bound on RKHS norm γ_2 in one dimension. Using the same reasoning as in section 9.3.3, 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 $\gamma_2(f)^2 \leqslant \int_{-R}^{R} \eta_+(b)^2 \frac{db}{4R} + \int_{-R}^{R} \eta_-(b)^2 \frac{db}{4R}$.

By using Taylor's formula with an integral remainder as in section 9.3.3, we get, for any twice differentiable function f on $[-R, R]$,

$$\begin{aligned} f(x) &= \frac{1}{2}f(-R) + \frac{1}{2}f(R) + \frac{1}{2}f'(-R)(x+R) - \frac{1}{2}f'(R)(-x+R) \\ &\qquad + \frac{1}{2}\int_{-R}^{R} f''(b)(x-b)_+ db - \frac{1}{2}\int_{-R}^{R} f''(b)(-x+b)_+ db \\ &= \frac{1}{2}\big[f(R)+f(-R)\big] + \frac{R}{2}\big[f'(-R)-f'(R)\big] + \frac{x}{2}\big[f'(-R)+f'(R)\big] \\ &\qquad + \frac{1}{2}\int_{-R}^{R} f''(b)(x-b)_+ db - \frac{1}{2}\int_{-R}^{R} f''(b)(-x+b)_+ db. \end{aligned}$$

We can now use explicit representations of constants and linear functions, *without* Dirac measures as we need finite L_2-norms, as follows:

$$\begin{aligned} \frac{x}{2} &= \int_{-R}^{R} \frac{(x-b)_+ - (b-x)_+}{4R} db = \int_{-R}^{R} \frac{x-b}{4R} db \\ -\frac{R^2}{6} &= \int_{-R}^{R} b(x-b)_+ \frac{db}{4R} + \int_{-R}^{R} b(-x-b)_+ \frac{db}{4R}. \end{aligned}$$

After a short calculation left as an exercise, this leads to a norm γ_2 upper-bounded by a constant times

$$2R\int_{-R}^{R} f''(x)^2 dx + \big[f'(R)+f'(-R)\big]^2 + \big[R(f'(R)-f'(-R)) - f(-R) - f(R)\big]^2.$$

The main difference with γ_1 is that the second derivative is penalized by an L_2-norm and not by an L_1-norm, and this L_2-norm can be infinite when the L_1-norm is finite (the classic example is the hidden neuron functions $(x-b)_+$).

⚠ The RKHS is combining infinitely many hidden neuron functions $(x-b)_+$, none of which 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 a square-integrable second derivative is dense in L_2).

Upper bound on RKHS norm γ_2 in all dimensions. We can first find a bound directly from the one on γ_1 in equation (9.14), which is exactly equation (9.15), ending up with the restriction on the ball with center 0 and radius R of the Sobolev space corresponding to the square-integrable $s = \frac{d}{2} + \frac{5}{2}$ derivatives on $\mathbb{R}^d$. It turns out that this provides a bound on γ_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 that γ_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 as done in equation (9.24) (Wahba, 1990; Bach, 2023).

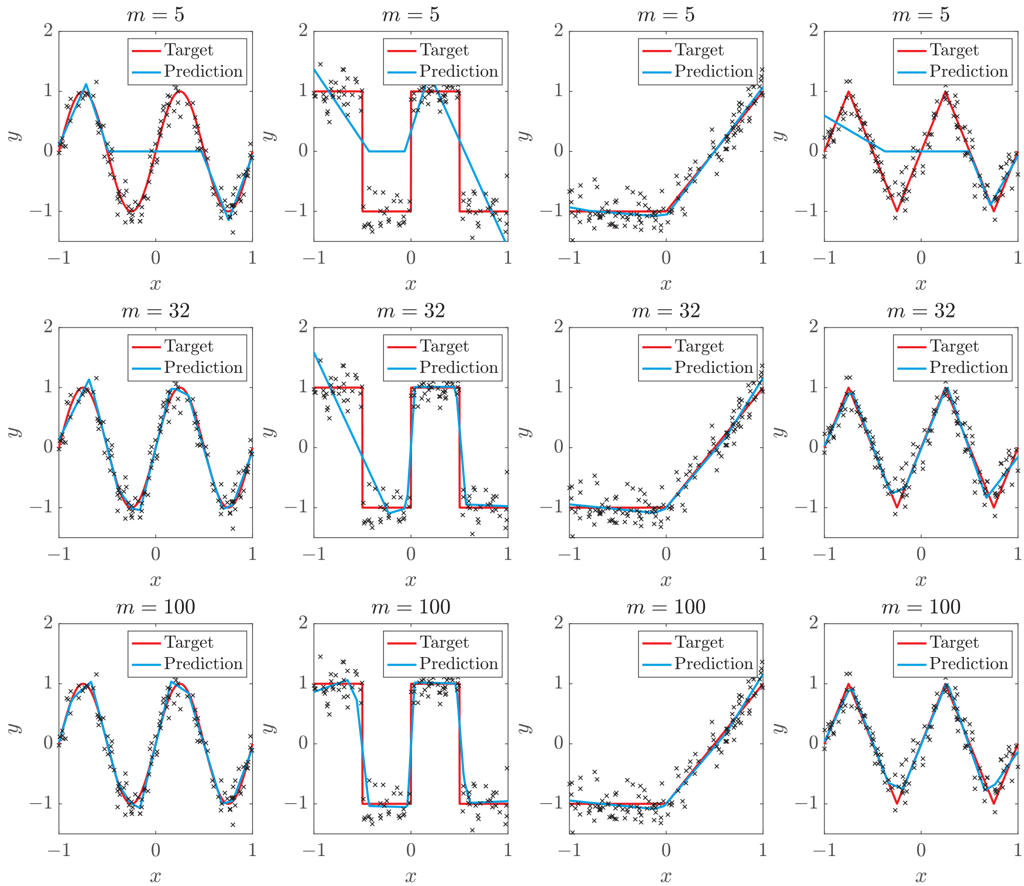


Figure 9.3. Fitting one-dimensional functions with various numbers of neurons m and no additional regularization (top: $m = 5$; middle: $m = 32$; bottom: $m = 100$), with four prediction problems (one per column).

9.6 Experiments

We consider the same experimental setup 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 nonsmooth targets. We consider several values for the number m of hidden neurons and a neural network with ReLU activation functions and an additional global constant term. Training is done by SGD 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 underfitting behavior of neural networks. See also Stewart et al. (2023) for a formulation of regression through classification that alleviates some of these issues, as well as `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 ℓ. For these models, obtaining simple and powerful theoretical results is still an active area of research in terms of approximation, estimation, and optimization errors. See, for instance, Lu et al. (2021), Ma et al. (2020), and Yang and Hu (2021). Among these results, the so-called "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 section 12.4 and, e.g., Jacot et al., 2018; Chizat et al., 2019).

- **Residual networks:** An alternative to stacking layers one after the other as before is to introduce a different architecture of the following 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 nonlinearity, originating from He et al. (2016), can be seen as a discretization of an ordinary differential equation (see Chen et al., 2018).

- **Convolutional neural networks:** To tackle large data and improve performances, it is important to use prior knowledge about the typical data structure to process. For instance, for signals, images, and 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).

- **Transformers:** One approach to capture long-range dependencies in sequential data $X = (x_1, \dots, x_L) \in \mathbb{R}^{L\times d}$, is to learn query $Q = W^{(Q)}X$, key $K = W^{(K)}X$, and value $V = W^{(V)}X$ matrices obtained by linear operators on X of compatible sizes, which are combined together to form an attention mapping (Bahdanau et al., 2014):

$$\text{attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d}}\Big)V.$$

Such a mapping is capable of capturing a variety of semantic relationships over sequences of data (e.g., grammatical relationships between query and key tokens within a corpus of text). The transformer (Vaswani et al., 2017) is an architecture that consists of stacked blocks made up of attention mappings, fully-connected layers and residual connections. The transformer architecture and its variants have a multitude of applications in fields such as natural language processing, audio, and computer vision.

9.8 Conclusion

In this chapter, we have focused primarily on neural networks with one hidden layer 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, so long as the norm of the weights is controlled.

We pursue the study of overparameterized models in chapter 12, where we show how optimization algorithms both globally converge and lead 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 or statistical benefits.
- Averaging/bagging: Averaging predictors on several reshuffled, resampled, or uniformly projected datasets will typically lower the estimator's variance with a potentially limited increase in bias.
- Boosting: Iteratively refining the prediction function by retraining 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: as for local averaging methods in chapter 6, which combine labels from nearby 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 having 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) that are stable by linear combinations, such linear combinations do not lead to new functions that could not be accessed initially. However, for nonlinear models, this leads to new functions with typically better approximation properties. For example, a one-hidden-layer neural network, as presented in section 9.2, is the combination of simple functions that lead to arbitrarily complex prediction functions if sufficiently many of them are added.

The construction of a new dataset given an old one $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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 (i.e., as many times as the weight) and then using an existing algorithm for regularized empirical risk minimization on the enlarged dataset. In particular, for stochastic gradient descent (SGD) 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 (e.g., uniformly distributed or constant). A similar effect can be obtained by modifying the original dataset using random projections. This is studied in sections 10.1 and 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 computational and statistical 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 nonlinear models: From a set of functions $\varphi(\cdot, w)$, with $w \in \mathcal{W}$, linear combinations increase the set of models to $\int_{\mathcal{W}} \varphi(x, w) d\nu(w)$, for ν a signed measure on $\mathcal{W}$. These will be adapted to boosting techniques (we already saw some of these 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, in terms of prediction performance, these are typically not adapted to ensemble learning techniques unless some variable/feature selection is added (as we do in section 10.2). However, there may be some computational benefits, such as the possibility of parallel processing.

10.1 Averaging/Bagging

In this section, for simplicity, we consider the regression with square loss where we have an explicit bias/variance decomposition, noting that most results extend to other situations using convex surrogates (see exercise 10.1) or using majority votes (see exercise 10.3).

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 independent and identically distributed (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 $j \in \{1, \ldots, m\}$ and λ is an associated hyperparameter specific to the learning procedure. Since we consider least-squares regression, the new predictor, $\hat{f}_\lambda^{\text{bag}}$, is simply the average of all $\hat{f}_\lambda^{(j)}$, $j = 1, \ldots, m$.

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 that x is fixed and only taking expectations with respect to the data), then they are the same for all $j \in \{1, \ldots, m\}$ and the bias of $\hat{f}_\lambda^{\text{bag}}$ 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 $m = 1$. In the context of independent datasets, it is relevant to concatenate all m datasets into one large dataset with $N = nm$ observations and learn a single predictor with these: the generalization performance of the average of m predictors will often be the same as the one of the single predictor on the large dataset, but with potential computational benefits. We now give a few examples for regression (we consider binary classification in exercises 10.1 and 10.3).

The 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 proposition 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 σ^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

$$\frac{\sigma^2}{km} + 8B^2 \text{diam}(\mathcal{X})^2 \Big(\frac{2k}{n}\Big)^{2/d}.$$

When optimizing this bound 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 1, and thus, m cannot grow larger than $n^{2/d}$.

Exercise 10.1 (♦) *Consider k-nearest neighbor multicategory classification with a majority vote rule. Using the relationship between the quadratic loss and the 0–1 loss from section 4.1.4 to derive an upper bound on the expected risk, what is the corresponding 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 (and the same regularization parameter). Again, with the proper choice of regularization parameter (lower λ 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 (see exercise 10.2), since, as shown in section 7.4, the training time for ridge regression is superlinear in the number of observations with running-time complexities between $O(n^2)$ and $O(n^3)$ if no low-rank approximations are used.

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?*

Exercise 10.3 (♦) *In the setup of this section with m independent datasets, consider an estimator $\hat{f}^{(j)} : \mathcal{X} \to \{-1, 1\}$ learned on the jth dataset for a binary classification problem, for $j \in \{1, \dots, m\}$, with $\hat{f}^{\text{bag}}(x) = \text{sign}\big(\sum_{j=1}^m \hat{f}^{(j)}(x)\big)$ the majority vote classifier. Denoting $f_*(x) \in \{-1, 1\}$ the optimal prediction at $x \in \mathcal{X}$, as defined in section 2.2.3, and $\varepsilon(x) = \mathbb{E}[\hat{f}^{(j)}(x) f_*(x)]$ (which is the same for all j), show that we have $\mathbb{E}[1_{\hat{f}^{\text{bag}}(x) \neq f_*(x)}] \leqslant \exp\big(-\frac{m}{2}(\varepsilon(x))_+^2\big)$. If $\forall x \in \mathcal{X}$, $\varepsilon(x) \geqslant \eta > 0$, show that the expected excess risk is less than $\exp\big(-\frac{m}{2}\eta^2\big)$.*

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 section 10.1.2, with a reduced variance still, but this time with a potentially higher bias.

10.1.2 Bagging

We consider datasets $\mathcal{D}^{(b)}$, obtained with random weights $v_i^{(b)} \in \mathbb{R}_+$, $i = 1, \dots, n$. For the bootstrap,[1] we consider n samples from the original n data points with replacement, which correspond to integer weights $v_i^{(b)} \in \mathbb{N}$, $i = 1, \dots, n$, 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,

[1] See https://en.wikipedia.org/wiki/Bootstrapping_(statistics) and Efron and Tibshirani (1994) for an introduction to bootstrapping methods in statistics.

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; see also section 7.6.1) 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 (see related examples in section 10.2). Overall, as will be shown for 1-nearest-neighbor, bagging will reduce 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 10.4 *Show that when sampling n elements with replacement from n items, the expected fraction of distinct items is $1-(1-1/n)^n$ and 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 in this subsection follows from Biau et al. (2010). The key observation is that if we denote as $(x_{(i)}(x), y_{(i)}(x))$ the pair of observations that is the ith-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 nonnegative weights V_i sum to 1 and *do not depend on x*. The weight V_i is the probability that the ith-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 that $s \geqslant 2$.

To select the ith-nearest neighbor as the 1-nearest-neighbor in a subsample, we need that the ith-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 B^2 multiplied by the expectation of the distance to the first nearest neighbor from a random sample of size s, and thus, for the ℓ_∞-norm on a subset $\mathcal{X}$ of $\mathbb{R}^d$, from lemma 6.1, less than $4B^2\mathrm{diam}(\mathcal{X})^2\frac{1}{s^{2/d}}$ 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 the 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.

10.2 Random Projections and Averaging

In section 10.1, we reweighted observations to be able to rerun the original algorithm. This can also be done through random projections of all observations. Such random projections can be performed in several ways: (1) for data in $\mathbb{R}^d$ by selecting s of the d variables; (2) still for data in $\mathbb{R}^d$, by projecting the data in a more general s-dimensional subspace; and (3) 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 (this will depend if the design matrix is left- or right-multiplied by a matrix of reduced size).

In this section, we consider random projections for ordinary least-squares (OLS), 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:

- *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

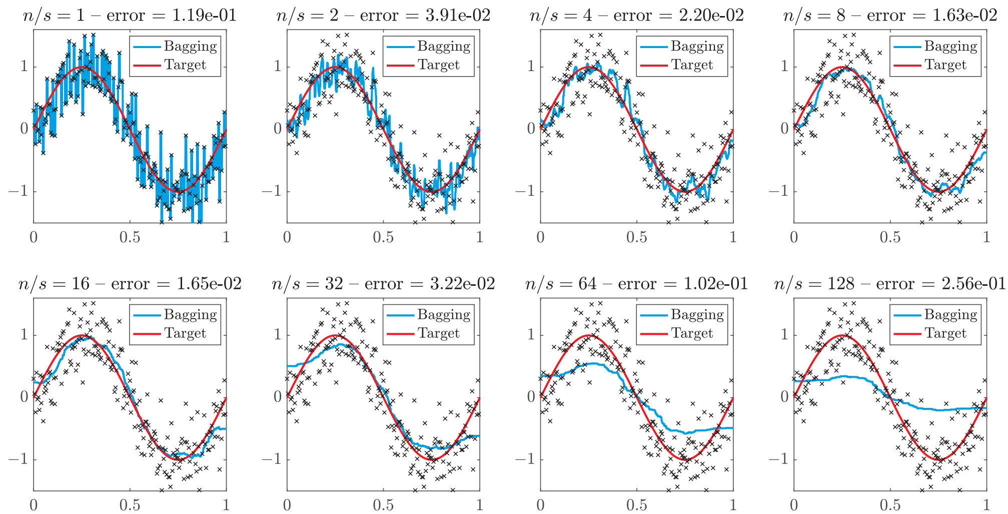


Figure 10.1. Subsampling estimates with $m = 20$ subsampled datasets, for varying subsampling ratios n/s, with an estimation of the testing error. When $n/s = 1$, we recover the 1-nearest neighbor classifier (which overfits), and when n/s grows, we get better fits until underfitting kicks in. Optimal testing error is obtained for $n/s = 8$.

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 $\mathbb{R}^{n\times d}$).

- *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 twofold: 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 matrices, and draw connections with kernel methods.

In the following sections, we study these precisely for the OLS framework (it could also be done for ridge regression). We first briefly mention a commonly used and 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, while selecting a random subset of features at every splitting decision. This algorithm has nice properties (invariance to rescaling of the variables and 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 OLS, we consider a design matrix $\Phi \in \mathbb{R}^{n\times d}$ with rank d (i.e., $\Phi^\top\Phi \in \mathbb{R}^{d\times d}$ 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 that follows.

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 fixed design 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 σ^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 expectations with respect to both the learning problem (in the fixed design setting, the noise vector ε) 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 ε is fixed.

We first introduce a representation tool that will allow simple expressions of all prediction vectors $S^{(j)}\Phi$. Since the Gaussian matrices $S^{(j)}$ are invariant under left and right multiplication by an orthogonal matrix, we can assume that the singular value decomposition (SVD) 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., $U^\top U = I$; remember than $n \geqslant d$), is such that $U = \binom{I}{0}$, and we can write $S^{(j)} = \begin{pmatrix} S_1^{(j)} & S_2^{(j)}\end{pmatrix}$, 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 the normal equation that defines $\hat\theta^{(j)} \in \mathbb{R}^d$, for each $j \in \{1,\dots,m\}$ (i.e., $(\Phi^\top(S^{(j)})^\top S^{(j)}\Phi)\hat\theta^{(j)} = \Phi^\top(S^{(j)})^\top S^{(j)}y)$, leading to the following closed-form estimators $\hat\theta^{(j)} = (\Phi^\top(S^{(j)})^\top S^{(j)}\Phi)^{-1}\Phi^\top(S^{(j)})^\top S^{(j)}y$.[2] Using the assumptions given previously regarding the SVD of Φ, we have $S^{(j)}\Phi = S_1^{(j)}DV^\top$. We can then expand the prediction 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 $\Phi\hat\theta_{\mathrm{OLS}} = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top y = \begin{pmatrix} I & 0 \\ 0 & 0\end{pmatrix}y$. Moreover,

[2] If $s \geqslant d$, then $S^{(j)}\Phi$ has rank d almost surely, and thus $\hat\theta^{(j)}$ is uniquely defined.

we have the model $y = \Phi\theta_* + \varepsilon$ and, if we split ε as $\varepsilon = \binom{\varepsilon_1}{\varepsilon_2}$, we have $y = \binom{I}{0} DV^\top\theta_* + \varepsilon$, and thus $y_2 = \varepsilon_2$. We therefore get

$$\mathbb{E}_{S^{(j)}}\Big[\big\|\Phi\hat{\theta}^{(j)} - \mathbb{E}_{S^{(j)}}\Phi\hat{\theta}^{(j)}\big\|_2^2\Big] = \mathbb{E}_{S^{(j)}}\big[\|((S_1^{(j)})^\top S_1^{(j)})^{-1}(S_1^{(j)})^\top S_2^{(j)}\varepsilon_2\|_2^2\big].$$

Taking the expectation with respect to ε (through $\mathbb{E}[\varepsilon_2\varepsilon_2^\top] = \sigma^2 I$), using the trace trick, and using expectations for the Wishart and inverse Wishart distributions,[3] this leads to

$$\begin{aligned}\mathbb{E}_{\varepsilon,S^{(j)}}\Big[\big\|\Phi\hat{\theta}^{(j)} - \mathbb{E}_{S^{(j)}}\Phi\hat{\theta}^{(j)}\big\|_2^2\Big] &= \sigma^2\mathbb{E}_{S^{(j)}}\big[\operatorname{tr}\big((S_2^{(j)})^\top S_1^{(j)}((S_1^{(j)})^\top S_1^{(j)})^{-2}(S_1^{(j)})^\top S_2^{(j)}\big)\big]\\ &= \sigma^2\mathbb{E}_{S^{(j)}}\big[\operatorname{tr}\big(S_2^{(j)}(S_2^{(j)})^\top S_1^{(j)}((S_1^{(j)})^\top S_1^{(j)})^{-2}(S_1^{(j)})^\top\big)\big]\\ &= (n-d)\sigma^2\mathbb{E}_{S_1^{(j)}}\big[\operatorname{tr}\big(((S_1^{(j)})^\top S_1^{(j)})^{-1}\big)\big] = \frac{d}{s-d-1}(n-d)\sigma^2.\end{aligned}$$

We can now compute the overall expected generalization error:

$$\begin{aligned}\frac{1}{n}\mathbb{E}_{\varepsilon,S^{(j)}}\Big[\Big\|\frac{1}{m}\sum_{j=1}^m\Phi\hat{\theta}^{(j)} - \Phi\theta_*\Big\|_2^2\Big] &= \frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\mathbb{E}_{S^{(1)}}\big[\Phi\hat{\theta}^{(1)}\big] - \Phi\theta_*\big\|_2^2\Big]\\ &\quad + \frac{1}{nm}\mathbb{E}_{\varepsilon,S^{(1)}}\Big[\big\|\Phi\hat{\theta}^{(1)} - \mathbb{E}_{S^{(1)}}\Phi\hat{\theta}^{(1)}\big\|^2\Big]\\ &= \frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\Phi\hat{\theta}_{\rm OLS} - \Phi\theta_*\big\|_2^2\Big] + \sigma^2\frac{d}{nm}\frac{n-d}{s-d-1}\\ &= \sigma^2\frac{d}{n} + \sigma^2\frac{d}{nm}\frac{n-d}{s-d-1}.\end{aligned}$$

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.

As in section 10.1, 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, for example, 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 the references therein. For the random projections that follow, our analysis will apply to more general sketches.

[3] 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.

10.2.2 Random Projections

We also consider the fixed design setup, 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 setup) and the rank of Φ is n. In this high-dimensional setup, we need some form of regularization, which will come here from random projections.

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 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 the 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 takes the form $\hat y^{(j)} = \Pi^{(j)}y$. Matrix $\Pi^{(j)}$ is almost surely an orthogonal projection matrix into an s-dimensional vector space, and its expectation is denoted as $\Delta \in \mathbb{R}^{n\times n}$, which satisfies $\mathrm{tr}(\Delta) = s$. We have, moreover, $0 \preccurlyeq \Delta \preccurlyeq I$; that is, all eigenvalues of Δ are between 0 and 1.

We can then compute expectations and variances as follows:

$$\begin{aligned}
\mathbb{E}_{S^{(j)}}\big[\hat y^{(j)}\big] &= \mathbb{E}_{S^{(j)}}\big[\Pi^{(j)}y\big] = \Delta y = \Delta\big[\Phi\theta_* + \varepsilon\big] = \Delta\varepsilon + \Delta\Phi\theta_* \\
\mathbb{E}_{S^{(j)}}\big[\hat y^{(j)}\big] - \Phi\theta_* &= \Delta\varepsilon + [\Delta - I]\Phi\theta_* \qquad (10.1)\\
\mathbb{E}_{S^{(j)}}\Big[\big\|\hat y^{(j)} - \mathbb{E}_{S^{(j)}}\big[\hat y^{(j)}\big]\big\|_2^2\Big] &= \mathbb{E}_{S^{(j)}}\Big[\big\|(\Pi^{(j)} - \Delta)y\big\|_2^2\Big] = y^\top\mathbb{E}_{S^{(j)}}\Big[\big(\Pi^{(j)} - \Delta\big)^2\Big]y \\
&= y^\top\mathbb{E}_{S^{(j)}}\big[\Pi^{(j)} - \Delta\Pi^{(j)} - \Pi^{(j)}\Delta + \Delta^2\big]y \ \text{ since } \Pi^{(j)}\Pi^{(j)} = \Pi^{(j)}, \\
&= y^\top\big(\Delta - \Delta^2\big)y, \ \text{since } \mathbb{E}[\Pi^{(j)} = \Delta. \qquad (10.2)
\end{aligned}$$

Thus, the overall (fixed design) expected generalization error is equal to, using that $S^{(1)},\dots,S^{(m)}$ are i.i.d. matrices,

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}_{\varepsilon,S}\Big[\big\|\tfrac{1}{m}\textstyle\sum_{j=1}^m \hat y^{(j)} - \Phi\theta_*\big\|_2^2\Big] \\
=\ &\frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\mathbb{E}_{S^{(1)}}\big[\hat y^{(1)}\big] - \Phi\theta_*\big\|_2^2 + \frac{1}{m}\mathbb{E}_{S^{(1)}}\Big[\big\|\hat y^{(1)} - \mathbb{E}_{S^{(1)}}\big[\hat y^{(1)}\big]\big\|_2^2\Big]\Big] \\
&\qquad\qquad\text{by taking expectations with respect to all } S^{(j)}, \\
=\ &\frac{1}{n}\mathbb{E}_\varepsilon\Big[\big\|\Delta\varepsilon + [\Delta - I]\Phi\theta_*\big\|_2^2 + \frac{1}{m}y^\top\big(\Delta - \Delta^2\big)y\Big] \ \text{using equations (10.1) and (10.2).}
\end{aligned}$$

Using the model $y = \Phi\theta_* + \varepsilon$ and the fact that $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I$, we get

$$\begin{aligned}
& \frac{1}{n}\mathbb{E}_{\varepsilon,S}\Big[\big\|\tfrac{1}{m}\textstyle\sum_{j=1}^m \hat{y}^{(j)} - \Phi\theta_*\big\|_2^2\Big] \\
= \ & \frac{\sigma^2}{n}\operatorname{tr}(\Delta^2) + \frac{1}{n}\theta_*^\top\Phi^\top[I-\Delta]^2\Phi\theta_* + \frac{1}{nm}\big[\sigma^2(\operatorname{tr}(\Delta) - \operatorname{tr}(\Delta^2)) + \theta_*^\top\Phi^\top\big(\Delta-\Delta^2\big)\Phi\theta_*\big] \\
& \qquad \text{using the model } y = \Phi\theta_* + \varepsilon \text{ and the fact that } \mathbb{E}[\varepsilon] = 0 \text{ and } \mathbb{E}[\varepsilon\varepsilon^\top] = \sigma^2 I, \\
= \ & \frac{\sigma^2}{n}\big(1-\frac{1}{m}\big)\operatorname{tr}(\Delta^2) + \frac{\sigma^2 s}{nm} + \frac{1}{n}\theta_*^\top\Phi^\top[\Delta-I]^2\Phi\theta_* + \frac{1}{nm}\theta_*^\top\Phi^\top\big(\Delta-\Delta^2\big)\Phi\theta_* \\
= \ & \frac{\sigma^2}{n}\big(1-\frac{1}{m}\big)\operatorname{tr}(\Delta^2) + \frac{\sigma^2 s}{nm} + \frac{1}{n}\theta_*^\top\Phi^\top\big[I-\Delta+\big(\frac{1}{m}-1\big)(\Delta-\Delta^2)\big]\Phi\theta_* \\
\leqslant \ & \frac{\sigma^2 s}{n} + \frac{1}{n}\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_*, \ \text{ since } \Delta^2 \preccurlyeq \Delta, \qquad (10.3)
\end{aligned}$$

which is the value for $m = 1$ (single replication). Note that the expectation (before taking the bound) decreases in m, with a limit $\frac{\sigma^2\operatorname{tr}(\Delta^2)}{n} + \frac{1}{n}\theta_*^\top\Phi^\top[I-\Delta]^2\Phi\theta_*$ when $m \to +\infty$ (with improved bias and variance terms). We now follow Kabán (2014) and Thanei et al. (2017) to bound the matrix $I - \Delta$.

Since Δ is the expectation of a projection matrix, we already know that $0 \preccurlyeq \Delta \preccurlyeq I$. We omit the superscript $^{(j)}$ for clarity, and we 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 this to get

$$\mathbb{E}_S\Big[\|z-\Phi SS^\top v\|_2^2\Big] = \|z\|_2^2 - 2z^\top\Phi\mathbb{E}_S\big[SS^\top\big]v + v^\top\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]v,$$

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

$$\frac{\sigma^2 s}{n} + \frac{1}{n}\theta_*^\top\Phi^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_*,$$

composed of expectations that can be readily computed. As shown next for special cases, we obtain a bias-variance trade-off similar to equation (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 error when $s = n$, as opposed to the random design setting studied in section 12.2 in the context of "double descent" (where generalization to unseen inputs is required).

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,[4]

$$\mathbb{E}_S[SS^\top] = sI \text{ and } \mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big] = s(s+1)\Phi^\top\Phi + s\,\mathrm{tr}(\Phi^\top\Phi)I.$$

We then get

$$\begin{aligned}
\theta_*^\top\Phi^\top\big[I-\Delta\big]\Phi\theta_* &\leqslant \theta_*^\top\Phi^\top\Big(I - \Phi\mathbb{E}_S\big[SS^\top\big]\big(\mathbb{E}_S\big[SS^\top\Phi^\top\Phi SS^\top\big]\big)^{-1}\mathbb{E}_S\big[SS^\top\big]\Phi^\top\Big)\Phi\theta_* \\
&= \theta_*^\top\Phi^\top\Big(I - s^2\Phi\big(s(s+1)\Phi^\top\Phi + s\,\mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\Phi^\top\Big)\Phi\theta_* \\
&= \theta_*^\top\Phi^\top\Phi\theta_* - s\theta_*(\Phi^\top\Phi)^2\big((s+1)\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&= \theta_*^\top\Phi^\top\Phi\big(\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)\big((s+1)\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&\leqslant 2\,\mathrm{tr}(\Phi^\top\Phi)\cdot\theta_*^\top\Phi^\top\Phi\big((s+1)\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}\theta_* \\
&\quad\ \text{using that } \Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I \text{ has eigenvalues less than } 2\,\mathrm{tr}(\Phi^\top\Phi), \\
&\leqslant 2\,\mathrm{tr}(\Phi^\top\Phi)\frac{\|\theta_*\|_2^2}{s+1},
\end{aligned}$$

since $\Phi^\top\Phi\big((s+1)\Phi^\top\Phi + \mathrm{tr}(\Phi^\top\Phi)I\big)^{-1}$ has eigenvalues less than $1/(s+1)$. The overall excess risk is then less than

$$\frac{\sigma^2 s}{n} + \frac{2}{n}\,\mathrm{tr}(\Phi^\top\Phi)\frac{\|\theta_*\|_2^2}{s+1}, \tag{10.4}$$

which is exactly of the form obtained for ridge regression in equation (3.6) with the identification $s \sim \frac{\mathrm{tr}(\Phi^\top\Phi)}{\lambda}$. We can consider other sketching matrices with additional properties, such as sparsity (see exercise 10.5).

Exercise 10.5 *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 equation (10.4).*

[4]If $W = S_1S_1^\top$ 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 $\mathbb{E}[WD^2W] = s(s+1)D^2 + s\,\mathrm{tr}(D^2)I$.

Kernel methods. (♦) The random projection idea can be extended to kernel methods discussed in chapter 7. We consider the kernel matrix $K = \Phi\Phi^\top \in \mathbb{R}^{n\times n}$, and the assumption $y = \Phi\theta_* + \varepsilon$ with $\|\theta_*\|_2$ bounded is turned into $y = y_* + \varepsilon$ with $y_*^\top K^{-1} y_*$ bounded. This corresponds to $y_* = K\alpha$, with a reproducing kernel Hilbert space (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 $\hat{y} = \hat{\Phi}(\hat{\Phi}^\top\hat{\Phi})^{-1}\hat{\Phi}^\top y$. Matrix Π is then $\Pi = \hat{\Phi}(\hat{\Phi}^\top\hat{\Phi})^{-1}\hat{\Phi}^\top$, and for the analysis, we need to compute its expectation, Δ (this corresponds to replacing ΦS in earlier developments starting at the beginning of section 10.2.2 with $\hat{\Phi}$). We have, following the same reasoning as before, 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 this 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 $v = \big(\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]\big)^{-1}\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]z$,

$$
z^\top (I-\Delta) z \leqslant z^\top \Big(I - \mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\big(\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]\big)^{-1}\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\Big)z. \tag{10.5}
$$

We then need to apply equation (10.5) to $z = y_*$ to get

$$
\theta_*^\top \Phi^\top \big[I - \Delta\big]\Phi\theta_* \leqslant y_*^\top \Big(I - \mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\big(\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big]\big)^{-1}\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big]\Big)y_*.
$$

We can, for example, consider each column of $\hat{\Phi}$ to be sampled from a Gaussian distribution with mean zero and covariance matrix K, for which we have

$$
\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\big] = sK \quad\text{and}\quad \mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big] = s(s+1)K^2 + s\,\mathrm{tr}(K)\cdot K.
$$

Using equation (10.3), with the same derivations that led to equation (10.4), this leads to a bound on the expected excess risk equal to $\frac{\sigma^2 s}{n} + \frac{2}{n}\,\mathrm{tr}(K)\frac{y_*^\top K^{-1}y_*}{s+1}$, which is exactly the bound in equation (10.4) 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 a running-time complexity of $O(n^3)$.

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 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. Then we have $\mathbb{E}\big[\hat{\Phi}\hat{\Phi}^\top\big] = sK$ by construction, while a short calculation (left as an exercise) shows that the second-order moment can be bounded as

$$\mathbb{E}_{\hat{\Phi}}\big[\hat{\Phi}\hat{\Phi}^\top\hat{\Phi}\hat{\Phi}^\top\big] \preccurlyeq s(s-1)K^2 + sR^2K.$$

This leads to the bound $\frac{\sigma^2 s}{n} + \frac{1}{n}R^2\frac{y_*^\top K^{-1}y_*}{s-1}$, which is almost the same as before, but with an efficient practical algorithm (since we now have to solve a least-squares regression problem in dimension s, which is more efficient than using the kernel trick if $s < n$).

Experiments. In figure 10.2, we consider a polynomial regression problem in dimension $d_{\mathcal{X}} = 20$, with polynomials of a maximum degree of 2, and thus a feature space of dimension $d = 1 + d_{\mathcal{X}} + d_{\mathcal{X}}(d_{\mathcal{X}} + 1)/2 = 231$. We also compare ridge regression with Gaussian random projections. We see better performance as m grows, consistent with our bounds (underfitting for small s, overfitting with large s). Moreover, when the number m of times the dataset is randomly projected goes from 10 to 100, we obtain almost the same plot, with better performance than $m = 1$, highlighting the fact that $m = 1$ is not optimal but taking m too large is not useful.

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 independent of the underlying dimension. Lemma 10.1 shows that all pairwise distances are preserved (a small modification would lead to the preservation of 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 a 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*

$$\forall i, j \in \{1, \dots, n\},\ (1-\varepsilon)\|\varphi_i - \varphi_j\|_2^2 \leqslant \big\|s^{-1/2}S^\top\varphi_i - s^{-1/2}S^\top\varphi_j\|_2^2 \leqslant (1+\varepsilon)\|\varphi_i - \varphi_j\|_2^2. \tag{10.6}$$

Proof (♦) Let $\psi \in \mathbb{R}^d$ with the ℓ_2-norm equal to 1. 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 jth 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.[5] 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 $|u| \leqslant \frac{1}{2}$, applied to ε and $-\varepsilon$, 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

[5] See https://en.wikipedia.org/wiki/Chi-squared_distribution.

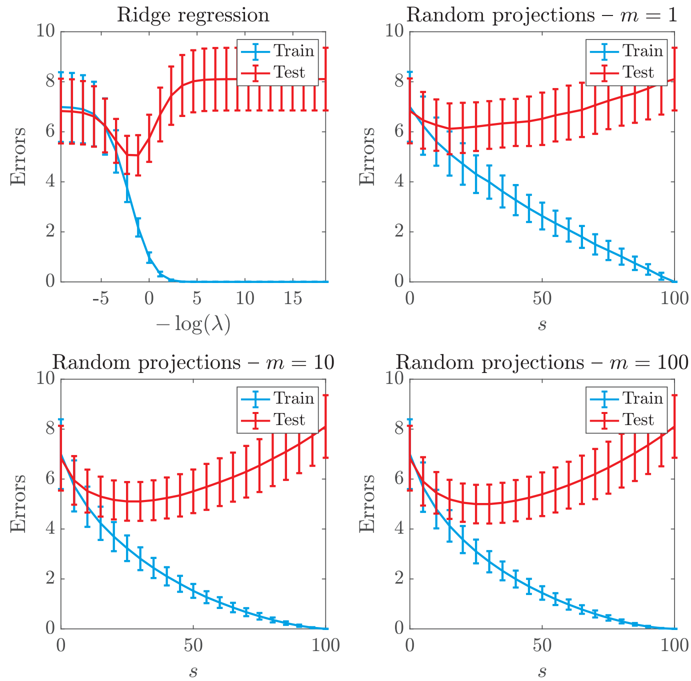


Figure 10.2. Polynomial regression in dimension 20, with polynomials of a maximum degree of 2, with $n = 100$. Top left: training and testing errors for ridge regression in the fixed design setting (the input data are fixed, and only the noise variables are resampled for computing the test error). All 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 the curves are averaged over 100 replications of the noise variables and the random projections.

this bound for ψ being the $n(n-1)/2$ vectors $\varphi_i - \varphi_j$, for $i \neq j$, leading to, using a union bound, a probability that equation (10.6) is not satisfied with a 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 that aims to characterize directly the predictive performance of such random projections (using the Johnson-Lindenstrauss lemma to obtain similar bounds is not straightforward as we consider unregularized regression, where perturbations of matrix inverses are harder to control).

10.3 Boosting

In sections 10.1 and 10.2, 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 independent 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. While the natural parallelizability is lost, we will see that we get additional statistical benefits.

In the early boosting procedures adapted to binary classification, the original learning algorithms (going from datasets to prediction functions with binary values) were used directly on a reweighted version, such as 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 this 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 Setup

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 $\varphi(\cdot, w) : \mathcal{X} \to \mathbb{R}$, for $w \in \mathcal{W}$, with $\mathcal{W}$ typically being a compact subset of a finite-dimensional vector space.

The main assumption is that given weights $\alpha \in \mathbb{R}^n$, one can reasonably easily find the function $\varphi(\cdot, w)$ that minimizes with respect to $w \in \mathcal{W}$:

$$\sum_{i=1}^{n} \alpha_i \varphi(x_i, w); \tag{10.7}$$

that is, the dot product between α and the n outputs of $\varphi(\cdot, w)$ 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 the following:

- Linear stumps for $\mathcal{X} = \mathbb{R}^d$: $\varphi(x, w) = \pm(w_0^\top x + w_1)_+$, where $w = (w_0, w_1) \in \mathbb{R}^d \times \mathbb{R}$, with sometimes the restriction that w has a single nonzero component (where the weak learning tractability assumption is indeed verified; see exercise 10.6). This will lead to a predictor, which is a one-hidden-layer neural network as presented in chapter 9, but learned sequentially rather than by GD 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.

Exercise 10.6 *For linear stumps with only one nonzero coordinate for the slope, show how to minimize equation (10.7) efficiently.*

- 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 with respect to 0 (i.e., for any $w \in \mathcal{W}$, there is $w' \in \mathcal{W}$ such that $\varphi(\cdot, w) = -\varphi(\cdot, w')$), which is the case for the two examples above.

Boosting procedures will make sequential calls to the weak learner oracle that outputs $w_1, \ldots, w_t \in \mathcal{W}$ with t the number of iterations, and *linearly combine* the function $\varphi(\cdot, w_1), \ldots, \varphi(\cdot, w_t)$. Therefore, the set of predictors that are explored are not only the functions $\varphi(\cdot, w)$, but all linear combinations; that is, functions of the form

$$f(x) = \int_{\mathcal{W}} \varphi(x, w) d\nu(w), \tag{10.8}$$

for ν 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 ν; that is:

$$\int_{\mathcal{W}} |d\nu(w)|.$$

Note that since we have assumed that the features are centrally symmetric, assuming that ν is a positive measure does not change anything.[6] Moreover, for a finite measure $\nu = \sum_{i=1}^t b_i \delta_{w_i}$, we have $f = \sum_{i=1}^t b_i \varphi_i(\cdot, w)$ and the penalty is $\|b\|_1$.

For functions $f : \mathcal{X} \to \mathbb{R}$ that can be represented as integrals in equation (10.8), 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 as $\mathcal{F}_1$, with a norm γ_1. Like in section 9.3.2, this is to distinguish it from the squared norm $\int_{\mathcal{W}} \left|\frac{d\nu(w)}{d\tau(w)}\right|^2 d\tau(w)$ for a fixed positive measure τ, which corresponds to a reproducing kernel Hilbert space (RKHS; see chapter 7).

The choice of this norm is motivated by the possibility of obtaining a generalization bound for gradient boosting. For linear stumps, the approximation properties have been characterized in section 9.3 for neural networks. For decision trees, this is more difficult as

[6] If $f = \int_{\mathcal{W}} \varphi(\cdot, w) d\nu(w)$ for $\nu = \nu_+ - \nu_-$ a signed measure with ν_+ and ν_- positive measures with disjoint supports, because of the central symmetry, there is a positive measure $\tilde{\nu}_-$ such that $\int_{\mathcal{W}} \varphi(\cdot, w) d\nu_-(w) = -\int_{\mathcal{W}} \varphi(\cdot, w) d\tilde{\nu}_-(w)$, leading to the representation $f(x) = \int_{\mathcal{W}} \varphi(\cdot, w) d(\nu_+ + \tilde{\nu}_-)(w)$ with $\int_{\mathcal{W}} d(\nu_+ + \tilde{\nu}_-)(w) = \int_{\mathcal{W}} |d\nu(w)|$.

the weak learner is defined as a greedy algorithm rather than through a space of functions used with empirical risk minimization (see an analysis by Scornet et al., 2015).

Note that by definition, for any $w \in \mathcal{W}$, $\gamma_1(\varphi(\cdot, w)) \leqslant 1$, 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 γ defined on $\mathbb{R}^n$ with a definition similar to γ_1 defined on functions from $\mathcal{X}$ to $\mathbb{R}$, with more properties that 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 γ is always convex and positively homogeneous. Since we further assumed central symmetry of the features, the set $\{\psi(w), w \in \mathcal{W}\} \subset \mathbb{R}^n$ is centrally symmetric, leading to $\gamma(-u) = \gamma(u)$, 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 ν), $\|u\|_2 = \big\| \int_{\mathcal{W}} \psi(w) d\nu(w) \big\|_2 \leqslant \int_{\mathcal{W}} \|\psi(w)\|_2 |d\nu(w)| \leqslant R\sqrt{n}\gamma(u)$.

The gauge function may not be a norm since it may not be finite everywhere; that is, there may be $u \in \mathbb{R}^n$ 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 of Cauchy-Schwarz inequality, as $u^\top v \leqslant \gamma(u)\gamma^*(v)$ (see chapter 15 in Rockafellar, 1997, for more details).

Assumptions. Following our traditional empirical risk minimization framework presented in chapter 4, we consider a loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, for both 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 for 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 ℓ_i is smooth, with smoothness constant G_2 (e.g., 1/4 for the logistic loss, 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, $G_0 = \log 2$). However, these statistical results could also be extended to the square loss.

Finite $\mathcal{W}$. While boosting methods can be applied for any compact set $\mathcal{W}$ (so 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

$$\min_{u \in \mathbb{R}^n} \; F(u) \;\text{ such that }\; \exists \alpha \in \mathbb{R}^d, \; u = \sum_{j=1}^{d} \alpha_j \psi(w_j),$$

and thus

$$\min_{\alpha \in \mathbb{R}^d} F\bigg(\sum_{j=1}^{d} \alpha_j \psi(w_j) \bigg). \tag{10.9}$$

We can thus see this as an optimization problem either in $u \in \mathbb{R}^n$ *or* in $\alpha \in \mathbb{R}^d$, and, given our assumptions regarding central symmetry, the gauge function $\gamma(u)$ is upper-bounded by the ℓ_1-norm $\|\alpha\|_1$ of the corresponding α. Seeing equation (10.9) as a problem in u may be advantageous because of strong-convexity properties that could be lost for the problem in α (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 α is not strongly convex (but it may still exhibit linear convergence for other reasons; see section 12.1.1).

Note that for finite sets, we could simply use gradient descent to find an approximate solution of equation (10.9), with a running-time complexity proportional to d at each iteration. However, this is not feasible when the set $\mathcal{W}$ is infinite, which is the standard setup of boosting algorithms, hence the need for incremental learning procedures that we present next.

10.3.2 Incremental Learning

The simplest version of boosting-like algorithms aims to construct linear combinations of functions of the form $x \mapsto \varphi(x, w_t)$ by selecting incrementally $w_t \in \mathcal{W}$. Starting from the function $g_0 = 0$, we thus consider the simplest update

$$g_t = g_{t-1} + b_t \varphi(\cdot, w_t), \tag{10.10}$$

where the linear combination coefficients $b_1, \dots, b_{t-1}$ for $\varphi(\cdot, w_1), \dots, \varphi(\cdot, w_{t-1})$ 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

$$\min_{b_t \in \mathbb{R}_+, \; w_t \in \mathcal{W}} \widehat{\mathcal{R}}\big(g_{t-1} + b_t \varphi(\cdot, w_t)\big). \tag{10.11}$$

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 equation (10.10) then becomes

$$u_t = u_{t-1} + b_t \psi(w_t)$$

and the optimization problem in equation (10.11) becomes

$$\min_{b_t \in \mathbb{R}_+, \; w_t \in \mathcal{W}} F\big(u_{t-1} + b_t \psi(w_t)\big). \tag{10.12}$$

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 consider the more widely applicable gradient boosting algorithm in section 10.3.5, which only needs to minimize an upper bound on equation (10.12), and analyze its expected risk in section 10.3.6, 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 equation (10.12) 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 assume 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 $\|\psi(w)\|_2^2 = n$, we have

$$\begin{aligned} F(u_t) &= F(u_{t-1}) + F'(u_{t-1})^\top (u_t - u_{t-1}) + \frac{1}{2n}\|u_t - u_{t-1}\|_2^2 \\ &= F(u_{t-1}) + F'(u_{t-1})^\top b_t \psi(w_t) + \frac{b_t^2}{2}. \end{aligned}$$

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

$$F(u_{t-1}) - \frac{1}{2}|F'(u_{t-1})^\top \psi(w_t)|^2. \tag{10.13}$$

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 $t \geqslant 1$,

$$\left\{ \begin{array}{lcl} w_t &=& \displaystyle\arg\max_{w \in \mathcal{W}} \ \big|(u_{t-1} - y)^\top \psi(w)\big| \\ u_t &=& u_{t-1} - \frac{1}{n}\big|(u_{t-1} - y)^\top \psi(w_t)\big|\psi(w_t) = u_{t-1} - \frac{1}{n}\gamma^*(u_{t-1} - y)\psi(w_t), \end{array} \right.$$

by definition of the polar gauge function γ^*.

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 that 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 toward 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 10.8), section 10.3.5 for a related result for all smooth loss functions (and with a detailed proof), Sil'nichenko (2004) for 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 γ 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 $\kappa > 0$ that 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 next, this ensures constant multiplicative progress across matching pursuit iterations. Indeed, we then have, from equation (10.13),

$$\frac{1}{2n}\|y - u_t\|_2^2 = \frac{1}{2n}\|y - u_{t-1}\|_2^2 - \frac{1}{2n^2}\gamma^*(u_{t-1} - y)^2 \leqslant (1-\kappa^2)\cdot \frac{1}{2n}\|y - u_{t-1}\|_2^2,$$

leading to exponential convergence.

Exercise 10.7 (♦) *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 $\varphi(x, w) \in \{-1, 1\}$ (i.e., 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 equation (10.12) 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

$$\begin{aligned}
&\min_{b_t\in\mathbb{R}, w_t\in\mathcal{W}} \sum_{i=1}^n \Big\{\frac{e^{-b_t}}{n}1_{y_i\psi(w_t)_i=1} + \frac{e^{b_t}}{n}1_{y_i\psi(w_t)_i=-1}\Big\}e^{-y_i(u_{t-1})_i} \\
= \ &\min_{b_t\in\mathbb{R}, w_t\in\mathcal{W}} \sum_{i=1}^n \Big\{\frac{e^{-b_t}}{n}\frac{1+y_i\psi(w_t)_i}{2} + \frac{e^{b_t}}{n}\frac{1-y_i\psi(w_t)_i}{2}\Big\}e^{-y_i(u_{t-1})_i} \\
= \ &\min_{b_t\in\mathbb{R}, w_t\in\mathcal{W}} nF(u_{t-1})\cdot\sum_{i=1}^n \Big\{\frac{e^{-b_t}}{n}\frac{1+y_i\psi(w_t)_i}{2} + \frac{e^{b_t}}{n}\frac{1-y_i\psi(w_t)_i}{2}\Big\}\pi_i,
\end{aligned}$$

where π is a vector in the simplex, defined as $\pi_i = \frac{e^{-y_i(u_{t-1})_i}}{\sum_{j=1}^n e^{-y_j(u_{t-1})_j}} = \frac{e^{-y_i(u_{t-1})_i}}{nF(u_{t-1})}$.

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_- (equal to $1 \mp \sum_{i=1}^n y_i\psi(w_t)_i\pi_i$), which is attained as $b_t = \frac{1}{2}\log\frac{a_-}{a_+}$, with an 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

$$F(u_{t-1})\Big[1 - \Big(\sum_{i=1}^n y_i\psi(w_t)_i\pi_i\Big)^2\Big]^{1/2}.$$

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 π. We thus get the following iteration:

$$\begin{cases} \pi_i = \dfrac{e^{-y_i(u_{t-1})_i}}{\sum_{j=1}^n e^{-y_j(u_{t-1})_j}} \text{ for } i \in \{1,\dots,n\} \\ w_t \in \arg\max\limits_{w\in\mathcal{W}} \sum\limits_{i=1}^n y_i\psi(w_t)_i\pi_i \\ u_t = u_{t-1} + \dfrac{1}{2}\log\dfrac{1 + \sum_{i=1}^n y_i\psi(w_t)_i\pi_i}{1 - \sum_{i=1}^n y_i\psi(w_t)_i\pi_i}\psi(w_t). \end{cases}$$

After this iteration, we have $F(u_t) = F(u_{t-1})\big[1 - \big(\sum_{i=1}^n y_i\psi(w_t)_i\pi_i\big)^2\big]^{1/2}$. Therefore, the empirical risk (with the exponential loss) strictly decreases if the weak learner gets an empirical weighted 0–1 loss that is 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 linear convergence. Note that if we make the same assumption as for matching pursuit at the end of section 10.3.3 (i.e., γ is a norm), then $\sum_{i=1}^n y_i\psi(w_t)_i\pi_i = \gamma^*(\pi\circ y) \geqslant \kappa\sqrt{n}\|\pi\circ y\|_2 = \kappa\sqrt{n}\|\pi\|_2 \geqslant \kappa\|\pi\|_1 \geqslant \kappa$, and we have a similar exponential convergence rate.

10.3.5 Greedy Algorithm Based on Gradient Boosting

In sections 10.3.3 and 10.3.4, the incremental update was performed in closed form, which was possible because the of the special structures of the square and exponential losses. It turns out that exact minimization is not needed for good predictive performance.

In this section, we describe a boosting algorithm which, at each iteration, performs a first-order Taylor expansion at the current point (which requires computing derivatives of the loss functions) and finds the weak learner $x \mapsto \varphi(x, w)$ that reduces the most this approximation of the risk. We thus consider the following greedy algorithm, starting from the zero function $g_0 = 0$, and iterating over $t \geqslant 1$ this procedure that makes locally optimal progress:

- 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 $\sum_{i=1}^n \alpha_i \varphi(x_i, w)$ 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 $g_t = g_{t-1} + b_t \varphi(\cdot, w_t)$ 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 $\varphi(\cdot, w_u)$, for $u \in \{1, \dots, t\}$, with only t atoms, thus leading to sparse combinations (in other words, the estimated measure ν is a sum of Dirac measures). For the square loss, this will be the exact matching pursuit algorithm presented in section 10.3.3. In general, these algorithms are often referred to as "gradient boosting" procedures (Friedman, 2001).

We 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 follows (using the definition of smoothness in equation (5.10)):

$$\begin{aligned}
F(u_t) &\leqslant F(u_{t-1}) + F'(u_{t-1})^\top (u_t - u_{t-1}) + \frac{L}{2}\|u_t - u_{t-1}\|_2^2 \\
&\leqslant F(u_{t-1}) + b_t F'(u_{t-1})^\top \psi(w_t) + \frac{L}{2} b_t^2 \|\psi(w_t)\|_2^2 \\
&\qquad\qquad\qquad\qquad \text{using the expression } u_t = u_{t-1} + b_t\psi(w_t), \\
&\leqslant F(u_{t-1}) + b_t F'(u_{t-1})^\top \psi(w_t) + \frac{L}{2} b_t^2 C^2, \qquad (10.14)
\end{aligned}$$

with L 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, with the optimal $b_t = \frac{-1}{LC^2} F'(u_{t-1})^\top \psi(w_t)$,

$$\left\{ \begin{array}{l} w_t \in \arg\max\limits_{w \in \mathcal{W}} \ F'(u_{t-1})^\top \psi(w) \\ u_t = u_{t-1} - \frac{1}{LC^2} F'(u_{t-1})^\top \psi(w_t) \cdot \psi(w_t), \end{array} \right. \qquad (10.15)$$

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 the gradient boosting algorithm) *Consider an L-smooth convex 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 γ is centrally symmetric. Consider the iteration in equation (10.15). 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

$$\gamma(u_t) \leqslant \gamma(u_0) + \frac{\sqrt{t}}{LC^2}\Big(2LC^2[F(u_0) - F(u_t)]\Big)^{1/2}. \tag{10.16}$$

Proof (♦) We have by construction of the iteration and from equation (10.14):

$$\begin{aligned} F(u_t) - F(v) &\leqslant F(u_{t-1}) - F(v) - \frac{1}{2LC^2}\big[F'(u_{t-1})^\top \psi(w_t)\big]^2 \\ &= F(u_{t-1}) - F(v) - \frac{1}{2LC^2}\gamma^*(F'(u_{t-1}))^2, \end{aligned} \tag{10.17}$$

by definition of the polar gauge γ^*. Moreover, using the convexity of F and properties of gauge functions, we have

$$F(u_t) - F(v) \leqslant F'(u_t)^\top(u_t - v) \leqslant \gamma^*(F'(u_t))\gamma(u_t - v). \tag{10.18}$$

Finally, using the triangular inequality for γ, we obtain, from equation (10.15),

$$\gamma(u_t - v) \leqslant \gamma(u_{t-1} - v) + \frac{1}{LC^2}\gamma^*(F'(u_{t-1})),$$

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 equation (10.17), we get

$$\Delta_t \leq \big(\Delta_{t-1} - \tfrac{1}{2LC^2}\gamma^*(F'(u_{t-1}))^2\big)_+, \tag{10.19}$$

and from equation (10.18), we get $\Delta_t \leqslant \Gamma_t\gamma^*(F'(u_t))$. Thus, using the monotonicity of the sequence (Γ_t),

$$\begin{aligned} \Delta_t\Gamma_t^{-2} &\leqslant \Delta_t\Gamma_{t-1}^{-2} \leqslant \big(\Delta_{t-1}\Gamma_{t-1}^{-2} - \tfrac{1}{2LC^2}\Gamma_{t-1}^{-2}\gamma^*(F'(u_{t-1}))^2\big)_+ \text{ from equation (10.19)}, \\ &\leqslant \big(\Delta_{t-1}\Gamma_{t-1}^{-2} - \tfrac{1}{2LC^2}\Gamma_{t-1}^{-2}(\Delta_{t-1}\Gamma_{t-1}^{-1})^2\big)_+ = \big(\Delta_{t-1}\Gamma_{t-1}^{-2} - \tfrac{1}{2LC^2}(\Delta_{t-1}\Gamma_{t-1}^{-2})^2\big)_+. \end{aligned}$$

This leads to[7]

$$\Delta_t\Gamma_t^{-2} \leqslant \frac{1}{\frac{t}{2LC^2} + \Gamma_0^2\Delta_0^{-1}} \leqslant \frac{2LC^2}{t}. \tag{10.20}$$

Moreover, by definition of Γ_t and using equation (10.18) that leads to the identity $\Delta_{t-1} \leqslant \Gamma_{t-1}\gamma^*(F'(u_{t-1}))$, we have

$$\Gamma_t = \Gamma_{t-1} + \frac{1}{LC^2}\gamma^*(F'(u_{t-1})) \leqslant \Gamma_{t-1}\Big(1 + \frac{1}{LC^2}\frac{\gamma^*(F'(u_{t-1})^2}{\Delta_{t-1}}\Big).$$

[7] We can use the following lemma, whose proof is left as an exercise: if (a_t) is a nonincreasing, nonnegative sequence such that $a_t \leqslant (a_{t-1} - a_{t-1}^2/c)_+$ for all $t \geqslant 1$, then $a_t \leqslant \frac{1}{t/c+1/a_0}$ for all $t \geqslant 0$.

Thus, by taking the product of the square of equation (10.19) and the previous inequality, we get

$$\Gamma_t \Delta_t^2 \leqslant \Gamma_{t-1}\Delta_{t-1}^2 \Big(1 + \frac{1}{LC^2}\frac{\gamma^*(F'(u_{t-1}))^2}{\Delta_{t-1}}\Big)\Big(1 - \frac{1}{2}\frac{1}{LC^2}\frac{\gamma^*(F'(u_{t-1}))^2}{\Delta_{t-1}}\Big)_+^2. \tag{10.21}$$

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 $\Gamma_t\Delta_t^2 \leqslant \Gamma_0\Delta_0^2$. This leads to, using equation (10.20), $\Delta_t^5 = (\Gamma_t\Delta_t^2)^2 \cdot \Delta_t\Gamma_t^{-2} \leqslant (\Gamma_0\Delta_0^2)^2\frac{2LC^2}{t}$, and thus to the first result. We can also bound the norm $\gamma(u_t)$ as follows:

$$\gamma(u_t) \quad \leqslant \quad \gamma(u_0) + \frac{1}{LC^2}\sum_{i=1}^{t}\gamma^*(F'(u_{i-1})). \tag{10.22}$$

Using equation (10.17) and a telescoping sum, we then get

$$\sum_{i=1}^{t}\gamma^*(F'(u_{i-1}))^2 \leqslant 2LC^2\big[F(u_0) - F(u_t)\big],$$

which, with equation (10.22) leads to equation (10.16). ■

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:

$$F(u_t) - F(u_*) \leqslant \Big(\frac{2LC^2\gamma(u_0-u_*)^2(F(u_0)-F(u_*))^4}{t}\Big)^{1/5} \leqslant \frac{LC^2\gamma(u_0-u_*)^2}{t^{1/5}},$$

which can be compared to the bound for regular GD applied directly to F (proposition 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. However, as mentioned earlier, it cannot be run when the set $\mathcal{W}$ is infinite (moreover, even when it can be run, e.g., with finite sets, iterates of GD are not expressed as linear combinations of a maximum of t iterates).

As done in section 10.3.6, assuming that $u_0 = 0$ and F is nonnegative everywhere, proposition 10.1 leads to

$$\big(F(u_t) - F(v)\big)_+ \leqslant \Big(\frac{2LC^2\gamma(v)^2F(0)^4}{t}\Big)^{1/5} \quad \text{and} \quad \gamma(u_t) \leqslant \frac{\sqrt{2t}}{\sqrt{LC^2}}F(0)^{1/2}. \tag{10.23}$$

This expression shows that the gauge function γ controls the convergence of the gradient-boosting algorithm in the same way that the Euclidean norm controls the convergence of GD (hence the introduction of γ_1 and γ). For finite sets $\mathcal{W}$, where the gauge function is essentially an ℓ_1-norm in a reparameterization, the link with an ℓ_1-norm penalization can be made explicit (see, e.g., Rosset et al., 2004, for details).

Exercise 10.8 (♦) *Show that when function F is quadratic, then we have the following guarantee: $F(u_t)-F(u_*) \leqslant \frac{LC^2}{2t^{1/3}}\gamma(u_0-u_*)^2$. Hint: replace equation (10.18) with $F(u_t)-F(u_*) = \frac{1}{2}F'(u_t)^\top(u_t-u_*)$.*

10.3.6 Convergence of Expected Risk

To bound the expected risk, we need to relate empirical risk $\widehat{\mathcal{R}}$ and expected risk $\mathcal{R}$ for a function f with bounded penalty $\gamma_1(f)$. To study the generalization performance of constraining or penalizing by the variation norm defined earlier in this chapter, 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

$$\mathbb{E}\Big[\sup_{\gamma_1(g)\leqslant D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)\Big] = D \cdot \mathbb{E}\Big[\sup_{w\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i \varphi(x_i, w)\Big], \tag{10.24}$$

where the expectation is taken with respect to both 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 upper bound 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 an ℓ_1-norm constraint on neural network weights, showing that although set $\mathcal{W}$ is infinite, we can bound the uniform deviations. See another example in exercise 10.9. In the following, we will assume that, for a universal constant $\rho_\varphi > 0$,

$$\mathbb{E}\Big[\sup_{\gamma_1(g)\leqslant D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i)\Big] \leqslant \frac{DR}{\sqrt{n}}\rho_\varphi. \tag{10.25}$$

Exercise 10.9 *Given a metric space $\mathcal{X}$ with distance d and finite diameter, consider $\varphi(w) = \sigma(d(x, w))$ for $w \in \mathcal{W} = \mathcal{X}$. Compute an upper bound on the Rademacher complexity in equation (10.24).*

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 that the feature maps φ form a centrally symmetric set and they are uniformly bounded by R and satisfy equation (10.25). Assume that the loss function ℓ is nonnegative, 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 tth iterate of the gradient boosting procedure, then, for any function $f : \mathcal{X} \to \mathbb{R}$,*

$$\mathbb{E}\big[\mathcal{R}(g_t)\big] \leqslant \mathcal{R}(f) + \Big[\sqrt{2t}\frac{G_0^{1/2}}{G_2^{1/2}} + R\gamma_1(f)\Big] \cdot 2G_1 \cdot \frac{\rho_\varphi}{\sqrt{n}} + \frac{(R\gamma_1(f))^{2/5}}{t^{1/5}}(2G_2G_0^4)^{1/5}. \tag{10.26}$$

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 $F(0) \leqslant G_0$: equation (10.23) leads to $\gamma_1(g_t) \leqslant \frac{\sqrt{2t}}{\sqrt{G_2 R^2}} G_0^{1/2}$, 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, in particular, the contraction principle applied to Lipschitz-continuous loss functions,

$$\mathbb{E}\big[\mathcal{R}(g_t) - \mathcal{R}(f)\big] \quad \leqslant \quad \Big[\frac{\sqrt{2t}}{\sqrt{G_2 R^2}} G_0^{1/2} + \gamma_1(f)\Big] \cdot 2G_1 \cdot \frac{\rho_\varphi R}{\sqrt{n}} + \Big(\frac{2G_2R^2\gamma_1(f)^2G_0^4}{t}\Big)^{1/5},$$

which leads to the desired result. ■

Up to constants that do not depend on t or n, the bound in equation (10.26) takes 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 to be of order $t \sim n^{5/7}(R\gamma_1(f))^{4/7}$, then this leads to $\mathcal{R}(f) + \frac{R\gamma_1(f)}{\sqrt{n}} \cdot \rho_\varphi + 2\big(\frac{R\gamma_1(f)}{\sqrt{n}}\big)^{2/7}$.

Assuming for simplicity that ρ_φ is a constant (as for neural networks), the dominant term is $\mathcal{R}(f) + (R\gamma_1(f)/\sqrt{n})^{2/7}$. If the Bayes predictor f_* is such that $\gamma_1(f^*)$ is finite, we immediately get an excess risk that goes to zero as $(R\gamma_1(f_*)/\sqrt{n})^{2/7}$. If the model that we consider is misspecified, then, as 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 γ_1. The bound discussed here 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 $\rho_\varphi RD/\sqrt{n}$, 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 $\rho_\varphi RD/\sqrt{n}$ for any f_* such that $\gamma_1(f_*) \leqslant D$ (this assumes that $\gamma_1(f_*)$ is finite). With the optimal choice of D, this leads to $\mathcal{R}(f_*)$ plus a constant times $\rho_\varphi R\gamma_1(f_*)/\sqrt{n}$, which is significantly better than for boosting. This, however, requires setting the constant D, which involves running the algorithm several times to tune it by cross-validation.

Comparison with early-stopping for gradient descent. Compared to the end of section 5.2.4, where we analyzed GD on the empirical risk with early-stopping and rates in $O(1/n^{1/4})$, our analysis of boosting also leads to consistent estimation, but with slightly worse rates. However, it can be applied to infinite sets $\mathcal{W}$ when an efficient algorithm for obtaining weak learners is available (and the analysis can probably be tightened).

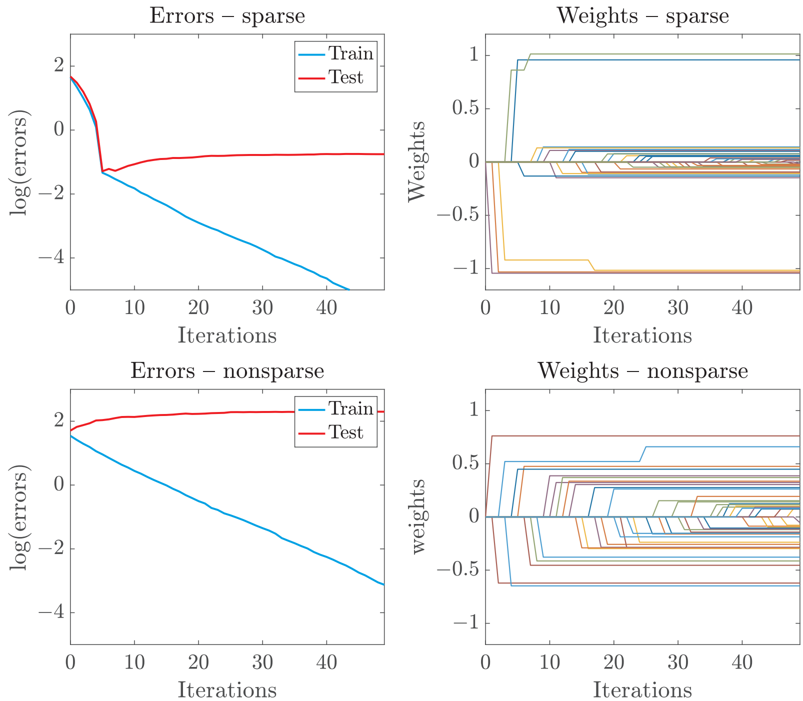


Figure 10.3. Matching pursuit on a problem with a sparse solution (top) and a nonsparse solution (bottom). Left: plots of training and testing errors; right: plots of weights.

10.3.7 Experiments

In this section, we compare the gradient boosting algorithm on a simple linear regression task with feature selection, noting that gradient boosting provides optimization algorithms with similar properties and iteration complexities as the ones derived for explicit ℓ_1-regularization in section 8.3.1. This corresponds to $F(u) = \frac{1}{2n}\|y-u\|_2^2$, which is $(1/n)$-smooth and strongly convex, and $\gamma(u) = \inf_{\alpha \in \mathbb{R}^d} \|\alpha\|_1$ with constraint that $u = \Phi\alpha$.

We consider $n = 100$ observations in dimension $d = 1,000$, sampled from a standard Gaussian random vector. A predictor β_* with $k = 5$ nonzero 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).

As in section 8.4, we also consider a rotation of the data, so this is no longer a sparse problem (bottom plot). We observe linear convergence of the training errors, as proved in section 10.3.3, but with overfitting at convergence, which is strong for the nonsparse case (bottom row) and weak for the sparse case (top row).

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 ℓ_1-norm; in section 12.1, we analyze the implicit bias of GD, when run to convergence, with a link to ℓ_2-penalties.

Chapter 11

From Online Learning to Bandits

Chapter Summary

- Beyond empirical and expected risk minimization with independent and identically distributed data, more complex settings can be considered.
- Online convex optimization with gradients: Stochastic gradient descent (SGD) 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.
- Zeroth-order optimization: Randomization can be used to obtain a stochastic gradient from function values with an additional dependence on dimension.
- Multiarmed 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 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

$F_s(\theta) = \ell(y_s, f_\theta(x_s))$, $s = 1, \ldots, t$, are independent and identically distributed (i.i.d.), 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 (i.e., $F(\theta_t)$). This is important when the loss functions can be interpreted as actual financial losses incurred while learning parameter θ (such as in advertising or finance applications).

  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 an expectation (since θ_s is random because it depends on 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 $\frac{1}{t}\sum_{s=1}^{t} F_s(\theta_{s-1}) - \inf_{\theta \in \mathcal{C}} \frac{1}{t}\sum_{s=1}^{t} F_s(\theta)$, which is the comparison to the optimal constant prediction. This allows it to be robust to adversarial functions and adapted to potentially nonstationary 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 (zeroth-order):** Independent 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 zeroth-order optimization, which can be treated through gradient-based algorithms (section 11.2) or the framework of multiarmed 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), and Orabona (2019). This chapter aims to give the main ideas involved here, explore how they differ from classical learning theory (using the unified notations that we provide in this book), and encourage readers to study these references. Along the way, we will describe the mirror descent framework, which has many applications beyond online learning.

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 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 subgradient 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,

$$\mathbb{E}\big[g_s|\mathcal{F}_{s-1}\big] = F_s'(\theta_{s-1}). \tag{11.2}$$

Given the added randomness, we consider the expected regret as a criterion.

Oblivious versus adaptive adversaries. Online learning is often cast in a game-theoretic framework, where the parameter θ that we try to estimate is the "player," while the "adversary" provides potentially hard functions to minimize (data in the machine learning context). Two types of adversaries are typically considered:

- *Oblivious adversary*: For each $t > 0$, function F_t is generated beforehand, without adaptation to the choice of θ_s for $s < t$, with the noise at time t that can depend on past information, but so that the observed gradient is an unbiased estimate of F_t at the (random) parameter θ_{t-1}.
- *Adaptive adversary*: For each $t > 0$, function F_t may depend on $\theta_1, \dots, \theta_{t-1}$, and more generally, on all the information up to time $t - 1$.

In this section, for simplicity, we focus primarily on oblivious adversaries but briefly show how to extend our results to adaptive adversaries.

Regularity assumptions. For simplicity, beyond convexity, 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] In this section, we only present the nonsmooth case. The smooth case will be proposed as exercises but leads to similar results compared to the regular stochastic case.

Applications. The online convex optimization framework applies beyond the independent and identically distributed (i.i.d.) framework that has been the main focus of this book. In the machine learning context, $F_t(\theta)$ is of the form $F_t(\theta) = \ell(y_t, f_\theta(x_t))$, for a

[1] The square loss is not Lipschitz-continuous on an unbounded domain, but it is once it has been constrained to a bounded domain.

pair or random observations $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, a loss function ℓ, and a prediction function $f_\theta : \mathcal{X} \to \mathbb{R}$. The distribution of the pair (x_t, y_t) may thus depend on past observations, without statistical independence. This is thus adapted to a nonstationary environment, where the data distribution varies over time, either stochastically or even adversarially (based on earlier predictions). As opposed to the rest of this book, where the performance of our estimates θ_t was measured according to the data distribution that generated the i.i.d. data, we consider here the regret framework, which compares the incurred losses to the ones obtained from a constant predictor (it is also possible to consider regrets computed by comparing to potentially slowly varying estimates, but this is out of scope of this chapter).

11.1.1 Convex Case

We consider the projected SGD recursion:

$$\theta_s = \Pi_{\mathcal{C}}(\theta_{s-1} - \gamma_s g_s), \tag{11.3}$$

for a certain positive step size γ_s (which we assume to be deterministic for simplicity), where $\Pi_{\mathcal{C}}$ is the orthogonal projection onto set $\mathcal{C}$. Proposition 11.1 provides a bound on the expected regret.

Proposition 11.1 (Online convex optimization–convex functions) *Consider a sequence of deterministic convex functions $(F_t)_{t\geqslant 1}$, and gradients $(g_t)_{t\geqslant 0}$ satisfying equation (11.2) and $\|g_t\|_2^2 \leqslant B^2$ for each $t \geqslant 1$. For the SGD recursion in equation (11.3) with step size sequence $\gamma_t = \frac{\mathrm{diam}(\mathcal{C})}{B\sqrt{t}}$, we have:*

$$\frac{1}{t}\sum_{s=1}^{t} \mathbb{E}\big[F_s(\theta_{s-1})\big] - \inf_{\theta\in\mathcal{C}} \frac{1}{t}\sum_{s=1}^{t} F_s(\theta) \leqslant \frac{3B\mathrm{diam}(\mathcal{C})}{2\sqrt{t}}. \tag{11.4}$$

Proof The proof follows the same steps as the one of proposition 5.7 with key differences: (1) the functions $(F_t)_{t\geqslant 1}$ are not all equal, and (2) we compare the objective functions to their values for *any* $\theta \in \mathcal{C}$ (as opposed to a fixed $\theta = \eta_*$ being the global optimum of the unique function F).

We thus have, for any $\theta \in \mathcal{C}$,

$$\begin{aligned}
\|\theta_s - \theta\|_2^2 &\leqslant \|\theta_{s-1} - \theta\|_2^2 - 2\gamma_s g_s^\top(\theta_{s-1} - \theta) + \gamma_s^2 B^2 \text{ by contractivity of projections,}\\
\mathbb{E}\big[\|\theta_s - \theta\|_2^2 \big| \mathcal{F}_{s-1}\big] &\leqslant \|\theta_{s-1} - \theta\|_2^2 - 2\gamma_s F_s'(\theta_{s-1})^\top(\theta_{s-1} - \theta) + \gamma_s^2 B^2,\\
&\qquad\qquad \text{using the unbiasedness of the gradient,}\\
&\leqslant \|\theta_{s-1} - \theta\|_2^2 - 2\gamma_s\big[F_s(\theta_{s-1}) - F_s(\theta)\big] + \gamma_s^2 B^2, \text{ using convexity.}
\end{aligned}$$

Taking full expectations and isolating $F_s(\theta_{s-1}) - F_s(\theta)$, we get

$$\mathbb{E}\big[F_s(\theta_{s-1}) - F_s(\theta)\big] \leqslant \frac{1}{2\gamma_s}\Big(\mathbb{E}\big[\|\theta_{s-1} - \theta\|_2^2\big] - \mathbb{E}\big[\|\theta_s - \theta\|_2^2\big]\Big) + \frac{\gamma_s}{2}B^2.$$

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 technique is exactly the same as the one of proposition 5.7, with only the appearances of functions F_s that depend on s.

In chapter 5 (i.e., the proof of proposition 5.7), we considered nonuniform 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 (γ_s), as, using the notation $\delta_s=\mathbb{E}\big[\|\theta_s-\theta\|_2^2\big]$ and for decreasing step sizes,

$$\begin{aligned}
\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big]-\frac{1}{t}\sum_{s=1}^{t}F_s(\theta) &\leqslant \frac{1}{t}\sum_{s=1}^{t}\frac{1}{2\gamma_s}\big(\delta_{s-1}-\delta_s\big)+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2 \\
&\qquad\text{from the last equation,} \\
&= \frac{1}{t}\sum_{s=1}^{t-1}\delta_s\Big(\frac{1}{2\gamma_{s+1}}-\frac{1}{2\gamma_s}\Big)+\frac{\delta_0}{2t\gamma_1}-\frac{\delta_t}{2t\gamma_t}+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2 \\
&\qquad\text{using Abel's summation formula,} \\
&\leqslant \frac{1}{t}\sum_{s=1}^{t-1}\mathrm{diam}(\mathcal{C})^2\Big(\frac{1}{2\gamma_{s+1}}-\frac{1}{2\gamma_s}\Big)+\frac{\mathrm{diam}(\mathcal{C})^2}{2t\gamma_1}+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2 \\
&\qquad\text{using that } \delta_s\leqslant\mathrm{diam}(\mathcal{C})^2 \text{ for all } s, \\
&= \frac{\mathrm{diam}(\mathcal{C})^2}{2t\gamma_t}+\frac{1}{t}\sum_{s=1}^{t}\frac{\gamma_s}{2}B^2.
\end{aligned}$$

By choosing $\gamma_s=\frac{\mathrm{diam}(\mathcal{C})}{B\sqrt{s}}$, we get using the same inequalities as for the proof of proposition 5.7,

$$\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big]-\frac{1}{t}\sum_{s=1}^{t}F_s(\theta)\leqslant\frac{3B\mathrm{diam}(\mathcal{C})}{2\sqrt{t}}, \tag{11.5}$$

leading to equation (11.4) after optimizing over $\theta\in\mathcal{C}$. ■

We show in section 11.1.4 that the rate in equation (11.4) is, up to constants, the best possible over all Lipschitz-continuous functions over a compact set. Moreover, the bound on the expected regret in equation (11.4) is essentially the same as for stochastic optimization (discussed in section 5.4). This is no surprise, as the proof ended up being almost the same. In fact, one can get proofs for the regular stochastic case (functions F_t all equal) from online learning bounds, with an "online-to-batch" conversion that we now present.

Online-to-batch conversion. All results on the expected regret for online convex optimization provide upper bounds on $\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big] - \inf_{\theta\in\mathcal{C}}\frac{1}{t}\sum_{s=1}^{t}F_s(\theta)$. If all functions F_t, $t \geqslant 1$, are equal to a convex function $F : \mathbb{R}^d \to \mathbb{R}$. We can use Jensen's inequality to get

$$\mathbb{E}\Big[F\Big(\frac{1}{t}\sum_{s=1}^{t}\theta_{s-1}\Big)\Big] - \inf_{\theta\in\mathcal{C}} F(\theta) \leqslant \frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F(\theta_{s-1})\big] - \inf_{\theta\in\mathcal{C}} F(\theta).$$

Therefore, online learning bounds lead to a bound for the averaged iterate of SGD with the classical sampling assumptions from section 5.4 (either empirical risk minimization when sampling data with replacement from a finite pool, or expected risk minimization for a single pass over the data).

Adaptive adversaries. In the proof of proposition 11.1, we considered only oblivious adversaries by assuming that the functions F_t were deterministic. For adaptive adversaries where F_t may depend on information up to time $t-1$, we need to replace in equation (11.5) $F_s(\theta)$ by $\mathbb{E}[F_s(\theta)]$, and the final result in equation (11.4) becomes

$$\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big] - \inf_{\theta\in\mathcal{C}}\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta)\big] \leqslant \frac{3B\mathrm{diam}(\mathcal{C})}{2\sqrt{t}},$$

which is a bound on a quantity called the "pseudo-regret" and that is only a lower bound on the expected regret $\mathbb{E}\big[\frac{1}{t}\sum_{s=1}^{t}F_s(\theta_{s-1}) - \inf_{\theta\in\mathcal{C}}\frac{1}{t}\sum_{s=1}^{t}F_s(\theta)\big]$. Note that (1) the extension to adaptive adversaries will hold in sections 11.1.2 and 11.1.3, and (2) when gradients are non-noisy, then there is no need for expectations (and thus proposition 11.1 applies to adaptive adversaries).

Exercise 11.1 (♦) *In the unconstrained online optimization with smooth convex functions (i.e., 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 $B^2\log(t)/(\mu t)$, as proposition 11.2 shows.

Proposition 11.2 (Online convex optimization–strongly convex functions) *Consider a sequence of deterministic μ-strongly-convex functions $(F_t)_{t\geqslant 1}$ on a compact convex set $\mathcal{C}$, and gradients $(g_t)_{t\geqslant 0}$ satisfying equation (11.2) and $\|g_t\|_2^2 \leqslant B^2$ for each $t \geqslant 1$. For the SGD recursion in equation (11.3) with step size sequence $\gamma_t = \frac{1}{\mu t}$, we have:*

$$\frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\big[F_s(\theta_{s-1})\big] - \inf_{\theta\in\mathcal{C}}\frac{1}{t}\sum_{s=1}^{t}F_s(\theta) \leqslant \frac{1}{2\mu t}(1+\log t). \tag{11.6}$$

Proof We can modify the proof of proposition 11.1 with the step size $\gamma_s = 1/(\mu s)$, in the same way as the proof of proposition 5.8 modified the one of proposition 5.7, to get

(with modifications in red)

$$\begin{aligned} \|\theta_s - \theta\|_2^2 &\leqslant \|\theta_{s-1} - \theta\|_2^2 - 2\gamma_s g_s^\top(\theta_{s-1} - \theta) + \gamma_s^2 B^2 \\ \mathbb{E}\big[\|\theta_s - \theta\|_2^2 \big| \mathcal{F}_{s-1}\big) &\leqslant \|\theta_{s-1} - \theta\|_2^2 - 2\gamma_s F_s'(\theta_{s-1})^\top(\theta_{s-1} - \theta) + \gamma_s^2 B^2 \\ &\leqslant \|\theta_{s-1} - \theta\|_2^2 - 2\gamma_s\big[F_s(\theta_{s-1}) - F_s(\theta) + \frac{\mu}{2}\|\theta_{s-1} - \theta\|_2^2\big] + \gamma_s^2 B^2. \end{aligned}$$

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, thus leading to equation (11.6). ■

After online-to-batch conversion, the bound in equation (11.6) exactly leads to proposition 5.8 for the uniformly averaged iterate. In section 5.4, exercise 5.32 showed a bound without the logarithmic term when using the step size $\gamma_t = \frac{2}{(t+1)\mu}$ (which is essentially twice larger than the one used to obtain the logarithmic term), but for a different averaging scheme with weights proportional to s (and thus not adapted to online learning that focuses on uniform averaging). 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 SGD recursion analysis from section 11.1.1 to the *online mirror descent* framework, which will apply as well to the regular stochastic case where all functions are equal (it is then referred to as stochastic mirror descent).

Mirror map. We assume that we are given a "mirror map" $\Phi : \mathcal{C}_\Phi \to \mathbb{R}$, which is differentiable and μ-strongly convex (on the set $\mathcal{C} \subset \mathcal{C}_\Phi$) with respect to a norm $\|\cdot\|$; that is, for all η, θ in the relative interior[2] of $\mathcal{C}$:

$$\Phi(\eta) \geqslant \Phi(\theta) + \Phi'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|^2.$$

We also assume that the gradient Φ' is a bijection from $\mathcal{C}_\Phi$ to $\mathbb{R}^d$. Classical examples are the following:

[2]See https://en.wikipedia.org/wiki/Relative_interior for a precise definition of relative interior.

- 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 $\mu = 1$ when $\mathcal{C}$ is the simplex $\{\theta \in (\mathbb{R}_+^*)^d,\ \sum_{i=1}^d \theta_i = 1\}$ (a result which is equivalent to Pinsker's inequality[3]).
- Squared ℓ_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 the proof of strong convexity by Ball et al., 2002).

See also exercise 13.4 in chapter 13 for an example of a mirror map for matrices.

Online mirror descent. We consider the same setup as the beginning of section 11.1.1 (i.e., we have convex Lipschitz-continuous functions F_s, for $s \geqslant 1$), and we access an unbiased subgradient 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

$$\theta_t = \arg\min_{\theta \in \mathcal{C}}\ g_t^\top(\theta - \theta_{t-1}) + \frac{1}{\gamma} D_\Phi(\theta, \theta_{t-1}), \tag{11.7}$$

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 Φ, and γ is a step size. If $\mathcal{C} = \mathcal{C}_\Phi$, then the update is simply defined by $\Phi'(\theta_t) = \Phi'(\theta_{t-1}) - \gamma g_t$.

Proposition 11.3 (Online mirror descent) *Given the mirror descent recursion in equation (11.7), 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 This proof follows the same structure as for online SGD in section 11.1.1. From the optimality conditions of the update in equation (11.7), 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

$$\begin{aligned}
& D_\Phi(\theta, \theta_t) - D_\Phi(\theta, \theta_{t-1}) \\
= \ & \Phi(\theta_{t-1}) + \Phi'(\theta_{t-1})^\top(\theta - \theta_{t-1}) - \Phi(\theta_t) - \Phi'(\theta_t)^\top(\theta - \theta_t) \\
= \ & \Phi(\theta_{t-1}) - \Phi(\theta_t) + \Phi'(\theta_{t-1})^\top(\theta_t - \theta_{t-1}) + (\Phi'(\theta_{t-1}) - \Phi'(\theta_t))^\top(\theta - \theta_t) \\
\leqslant \ & \Phi(\theta_{t-1}) - \Phi(\theta_t) + \Phi'(\theta_{t-1})^\top(\theta_t - \theta_{t-1}) + \gamma g_t^\top(\theta - \theta_t) \text{ using optimality conditions,} \\
= \ & -D_\Phi(\theta_t, \theta_{t-1}) - \gamma g_t^\top(\theta_{t-1} - \theta) - \gamma g_t^\top(\theta_t - \theta_{t-1}) \text{ by rearranging terms,} \\
\leqslant \ & -\frac{\mu}{2}\|\theta_t - \theta_{t-1}\|^2 - \gamma g_t^\top(\theta_{t-1} - \theta) + \gamma\|g_t\|_* \cdot \|\theta_t - \theta_{t-1}\| \leqslant \frac{\|g_t\|_*^2\gamma^2}{2\mu} - \gamma g_t^\top(\theta_{t-1} - \theta),
\end{aligned}$$

[3] See https://en.wikipedia.org/wiki/Pinsker's_inequality.

using the strong convexity of Φ and the bound on gradients (as well as the identity $a^\top b \leqslant \|a\| \cdot \|b\|_*$). By taking conditional expectations, we get

$$\mathbb{E}\big[D_\Phi(\theta,\theta_t) - D_\Phi(\theta,\theta_{t-1})\big|\mathcal{F}_{t-1}\big] \leqslant \frac{B^2\gamma^2}{2\mu} - \gamma F_t'(\theta_{t-1})^\top(\theta_{t-1}-\theta). \tag{11.8}$$

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 as in regular SGD (which corresponds precisely to the feature map $\Phi = \frac{1}{2}\|\cdot\|_2^2$).
- With online-to-batch conversion, we also get the same bound when all F_t's are equal for the averaged iterate, leading to stochastic mirror descent.
- 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 componentwise product), with then a normalization step to sum to 1, which is a *multiplicative* update, and the regret bound equals $B\sqrt{2\log(d)}/\sqrt{t}$. This regret bound would be of order $\sqrt{d}$ (instead of $\sqrt{\log d}$) if the Euclidean feature map was used.
- Online mirror descent is similar to the "follow-the-regularized-leader" algorithm, which is an online version of "dual averaging" optimization algorithm; see Xiao (2010), chapter 7 in Orabona (2019), and references therein.

Exercise 11.2 (Stochastic mirror descent for ℓ_1-regularization) *In the context of proposition 11.3, 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 θ_* is the minimizer of F, with an appropriate choice of γ, $\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta_*) \leqslant \frac{2B\|\theta_*\|_1\sqrt{\log d}}{\sqrt{t}}$.*

11.1.4 Lower Bounds (♦♦)

To prove a lower bound in the noiseless case, 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) functions $F_s^{(\varepsilon)} : \mathbb{R}^d \to \mathbb{R}$ defined as $F_s^{(\varepsilon)}(\theta) = \varepsilon_s^\top\theta$, for $\varepsilon_s \in \{-1,1\}^d$ for all $s \in \{1,\dots,t\}$; we denote as ε the concatenation of all ε_s for $s \in \{1,\dots,t\}$. The gradient vectors g_s are then simply equal to ε_s. We here have deterministic gradients, with constants $B = \sqrt{d}$ and $\mathrm{diam}(\mathcal{C}) = 2\sqrt{d}$.

To obtain a lower bound of performance, it suffices to show that for any sequence (θ_s),

$$\sup_{\varepsilon\in\mathcal{E}} \left\{ \frac{1}{t}\sum_{s=1}^t F_s^{(\varepsilon)}(\theta_{s-1}) - \inf_{\theta\in\mathcal{C}} \frac{1}{t}\sum_{s=1}^t F_s^{(\varepsilon)}(\theta) \right\}$$

is lower-bounded for $\mathcal{E}$ a well-chosen set. As already used in proving lower bounds in section 3.7 and as done 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 θ_{s-1} does not use the information about ε_s. Moreover, using that the ℓ_1-norm is dual to the ℓ_∞-norm,

$$\mathbb{E}\bigg[\inf_{\theta\in\mathcal{C}} \ \frac{1}{t}\sum_{s=1}^t \varepsilon_s^\top\theta\bigg] \ = \ \mathbb{E}\bigg[-\Big\|\frac{1}{t}\sum_{s=1}^t\varepsilon_s\Big\|_1\bigg] = -d\,\mathbb{E}\bigg[\Big|\frac{1}{t}\sum_{s=1}^t(\varepsilon_s)_1\Big|\bigg].$$

Therefore, from equation (11.10) in lemma 11.1 with $p=1$, $\eta = ((\varepsilon_s)_1)_{s\in\{1,\dots,t\}}$, and $x=(1,\dots,1)\in\mathbb{R}^t$, the regret is greater than $\mathbb{E}\big[d\big|\frac{1}{t}\sum_{s=1}^t(\varepsilon_s)_1\big|\big] \geqslant d/(144\sqrt{t})$, which is equal to a constant times $B\mathrm{diam}(\mathcal{C})/\sqrt{t}$, a lower bound that matches the upper bound from SGD from equation (11.4), 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 $x\in\mathbb{R}^t$. Let $p\in[1,\infty)$. Then*

$$\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \leqslant 3\sqrt{p}\cdot\|x\|_2, \tag{11.9}$$

and

$$\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \geqslant \begin{cases} \frac{1}{144}\|x\|_2 & \text{if } p\in[1,2], \\ \|x\|_2 & \text{if } p\geqslant 2. \end{cases} \tag{11.10}$$

Proof (♦) Without loss of generality, we assume $\|x\|_2=1$. We have, for $s=x^\top\eta$, and $p>0$, using the change of variable $u=\lambda^p$,

$$\mathbb{E}[|s|^p] = \int_0^{+\infty} u\,\mathbb{P}(|s|^p\geqslant u)du = p\int_0^{+\infty}\lambda^{p-1}\mathbb{P}(|s|\geqslant\lambda)d\lambda.$$

We then compute directly, using the independence of $\eta_1,\dots,\eta_t$,

$$\mathbb{E}[e^{ts}] = \prod_{i=1}^d\Big(\frac{1}{2}e^{tx_i}+\frac{1}{2}e^{-tx_i}\Big) = \prod_{i=1}^d\cosh(tx_i) \leqslant \exp(t^2\|x\|_2^2/2) = \exp(t^2/2),$$

using $\cosh\alpha\leqslant\exp(\alpha^2/2)$ for any $\alpha\in\mathbb{R}$. Thus, for $\lambda\geqslant 0$,

$$\begin{aligned}\mathbb{P}(|s|\geqslant\lambda) \ &= \ 2\mathbb{P}(s\geqslant\lambda) = 2\inf_{t\geqslant 0}\mathbb{P}(e^{ts}\geqslant e^{t\lambda}) \leqslant 2\inf_{t\geqslant0}e^{-\lambda t}\mathbb{E}[e^{ts}] \text{ using Markov's inequality,}\\ &\leqslant \ 2\inf_{t\geqslant0}e^{-\lambda t}\exp(t^2/2) = 2\exp(-\lambda^2/2), \ \text{ with } t=\lambda.\end{aligned}$$

Thus, through the change of variable $u = \lambda^2/2$,

$$\mathbb{E}[|s|^p] \leqslant 2p\int_0^{+\infty} \lambda^{p-1}\exp(-\lambda^2/2)d\lambda = 2^{p/2}p\int_0^{+\infty} u^{p/2-1}e^{-u}du = p2^{p/2}\Gamma(p/2),$$

where Γ is the Gamma function.[4] Through the Stirling formula $\Gamma(p/2)^{1/p} \sim \sqrt{p/(2e)}$; thus we have $\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \leqslant B_p$ with $B_p \sim \sqrt{p/e}$; one can then check the bound $B_p \leqslant 3\sqrt{p}$ for $p \geqslant 1$, leading to equation (11.9).

Moreover, we have, by independence and zero means of the components for η, $\mathbb{E}[|x^\top\eta|^2] = \sum_{i=1}^t x_i^2\mathbb{E}[\eta_i^2] = \|x\|_2^2 = 1$, and, by the Cauchy-Schwarz inequality, for $p \in [1,2]$:

$$\begin{aligned} 1 &= \mathbb{E}[|x^\top\eta|^2] = \mathbb{E}[|x^\top\eta|^{p/2}|x^\top\eta|^{2-p/2}] \leqslant \big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/2}\big(\mathbb{E}[|x^\top\eta|^{4-p}]\big)^{1/2} \\ &\leqslant \big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/2}B_{4-p}^{2-p/2}, \end{aligned}$$

leading to, for $p \in [1,2]$,

$$\big(\mathbb{E}[|x^\top\eta|^p]\big)^{1/p} \geqslant B_{4-p}^{1-4/p} \geqslant (3\sqrt{4-p})^{1-4/p} \geqslant (3\sqrt{3})^{1-4/p} \geqslant (3\sqrt{3})^{-3} \geqslant 1/144.$$

Moreover, for $p \geqslant 2$, we have by Jensen's inequality (which applies since $u \mapsto u^p$ is then convex on $\mathbb{R}_+$) $\|x\|_2 \leqslant \big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p}$ proving equation (11.10) for $p \geqslant 2$. ■

The optimal constant in for the inequality $\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \leqslant B_p\|x\|_2$ 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 for the inequality $\big(\mathbb{E}\big[|x^\top\eta|^p\big]\big)^{1/p} \geqslant A_p\|x\|_2$ is $A_p = 1$ if $p \geqslant 2$ and $A_p = 2^{1/2-1/p}$ if $p < 1.847$.[5]

Exercise 11.3 (♦) *What would upper and lower bounds be if the regret criterion were 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* (α_s) *of positive numbers?*

11.2 Zeroth-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 *zeroth-order optimization* (since the function value is the zeroth-order derivative of F, while the gradient is the vector of first-order derivatives). As presented in section 11.2.3, extensions to online learning naturally follow.

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:

$$\hat{F}'(\theta) = \sum_{i=1}^d \frac{1}{\delta}\big[F(\theta+\delta e_i) - F(\theta)\big]e_i \in \mathbb{R}^d, \tag{11.11}$$

[4] See https://en.wikipedia.org/wiki/Gamma_function.
[5] See more details in 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 δ tends to zero. Indeed, using the smoothness inequality from equation (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 θ and one at each $\theta+\delta e_i$, for $i \in \{1,\dots,d\}$), we can compute the exact gradient and use gradient descent (GD). 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 (i.e., essentially with bounded second-order derivatives) but only accessible with a stochastic first-order oracle (unbiased, with variance σ^2), for which, in equation (11.11), the noise in the function values will explode when δ goes to zero.

That is, we 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 ζ_t and ζ_t' are zero-mean random variables with variance σ^2, corresponding to the additive noise on the two function evaluations, and z_t is sampled from a distribution with mean $\mathbb{E}[z_t] = 0$ and covariance matrix $\mathbb{E}[z_t z_t^\top] = I$. By writing $\varepsilon_t = \zeta_t - \zeta_t'$, we get

$$\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{\delta}\big(F(\theta_{t-1}+\delta z_t) - F(\theta_{t-1}) + \varepsilon_t\big)z_t\Big], \tag{11.12}$$

where ε_t corresponds to the noise with the two function evaluations at θ_{t-1} and $\theta_{t-1}+\delta z_t$, thus of variance $2\sigma^2$.

There are two natural candidates for the distribution of z: (1) z, a signed canonical basis vector selected uniformly at random (i.e., $\pm\sqrt{d}e_i$, 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 as in equation (11.11); or (2) z, a standard Gaussian vector (with mean zero and identity covariance matrix). We consider the second option here, 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 ξ_t. We can then compute the function values $f(\theta,\xi_t)$ at *two* different points with the *same* ξ_t, and we can get an improved bound (see the end of section 11.2.1).

The key to analyzing the iteration in equation (11.12) is to study the gradient estimate $g = \frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta)\big)z$ for a certain vector θ and for a standard Gaussian vectorz z.

For δ being small, a simple Taylor expansion around θ 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 δ being 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 δ goes to zero, thus leading to some trade-offs that 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 section 11.2.2 for the nonsmooth version). An important tool will be to define the function $F_\delta : \mathbb{R}^d \to \mathbb{R}$ as

$$F_\delta(\theta) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[F(\theta+\delta z)], \tag{11.13}$$

which is the expectation of F taken at a point distributed as a Gaussian with mean θ and covariance matrix $\delta^2 I$. This function is useful because it turns out that the expectation of the gradient estimate in equation (11.12) is exactly the gradient of F_δ as shown in lemma 11.2.

Lemma 11.2 *Assume F is a smooth function. For $\theta \in \mathbb{R}^d$ and $\delta > 0$, we have:*

$$\mathbb{E}_{z\sim\mathcal{N}(0,I)}\Big[\frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta)\big)z\Big] = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\Big[\frac{1}{\delta}F(\theta+\delta z)z\Big] = F'_\delta(\theta).$$

Proof The first equality is a consequence of z having zero mean. For the second equality, we 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,

$$\begin{aligned}
F'_\delta(\theta) &= \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} F'(\theta+\delta\eta)\exp\Big(-\frac{1}{2}\|\eta\|_2^2\Big)d\eta \\
&= \frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} \frac{\partial F(\theta+\delta\eta)}{\partial\eta}\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)d\eta \\
&= -\frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} F(\theta+\delta\eta)\frac{\partial\exp\big(-\frac{1}{2}\|\eta\|_2^2\big)}{\partial\eta}d\eta \text{ by integration by parts,} \\
&= -\frac{1}{(2\pi)^{d/2}}\frac{1}{\delta}\int_{\mathbb{R}^d} F(\theta+\delta\eta)\exp\Big(-\frac{1}{2}\|\eta\|_2^2\Big)(-\eta)d\eta = \mathbb{E}\Big[\frac{1}{\delta}F(\theta+\delta z)z\Big],
\end{aligned}$$

leading to the desired result. ■

Approximation properties. We can analyze the difference between F and F_δ when F is L-smooth as follows (using that z has zero mean):

$$\forall \theta \in \mathbb{R}^d,\ F_\delta(\theta) - F(\theta) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[F(\theta+\delta z) - F(\theta) - \delta F'(\theta)^\top z\big].$$

Since F is convex, it is above its tangent at θ; thus, we get $F_\delta(\theta) \geqslant F(\theta)$ and, using the smoothness bound from equation (5.10), we get

$$\forall \theta \in \mathbb{R}^d,\ 0 \leqslant F_\delta(\theta) - F(\theta) \leqslant \frac{L\delta^2}{2}\mathbb{E}_{z\sim\mathcal{N}(0,I)}[\|z\|_2^2] = \frac{L}{2}\delta^2 d. \tag{11.14}$$

Moreover, we can compute the expectation of the squared norm of the gradient estimate:

$$\begin{aligned}
& \mathbb{E}\Big[\Big\|\frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta)\big)z\Big\|_2^2\Big] \\
\leqslant\ & 2\mathbb{E}\Big[\Big\|\frac{1}{\delta}\big(F(\theta+\delta z) - F(\theta) - \delta F'(\theta)^\top z\big)z\Big\|_2^2\Big] + 2\mathbb{E}\big[\|zz^\top F'(\theta)\|_2^2\big] \\
\leqslant\ & 2\mathbb{E}\Big[\frac{L^2\delta^2}{4}\|z\|_2^6\Big] + 2F'(\theta)^\top\mathbb{E}\big[\|z\|_2^2 zz^\top\big]F'(\theta),\ \text{using smoothness,} \\
=\ & \frac{L^2\delta^2}{2}d(d+2)(d+4) + 2\|F'(\theta)\|_2^2\cdot 3d \leqslant \frac{15}{2}L^2\delta^2 d^3 + 6d\|F'(\theta)\|_2^2, \qquad (11.15)
\end{aligned}$$

where we have used that $\|z\|_2^2$ is a chi-squared random variable, and 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 exercise 11.4).

Exercise 11.4 *Show that for a standard Gaussian vector $z \in \mathbb{R}^d$ (with zero mean and covariance matrix identity), we have $\mathbb{E}[\|z\|_2^6] = d(d+2)(d+4)$ and $\mathbb{E}\big[\|z\|_2^2 zz^\top\big] = 3dI$.*

Exercise 11.5 (Improved variance bound, Akhavan et al., 2023 (♦)) *Using the Poincaré inequality for standard Gaussian vectors z (i.e., for any differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, $\mathbb{E}\big[(f(z) - \mathbb{E}[f(z)])^2\big] \leqslant \mathbb{E}\big[\|f'(z)\|_2^2\big]$), show that the term $\frac{15}{2}L^2\delta^2 d^3$ in equation (11.15) can be improved to a constant times $L^2\delta^2 d^2$.*

We can now analyze GD and prove proposition 11.4 that gives a convergence rate that will be analyzed below its proof.

Proposition 11.4 (zeroth-order optimization–smooth case) *Let F be an L-smooth convex function with minimizer θ_*, and $(\theta_t)_{t\geqslant 0}$ defined by the recursion in equation (11.12), with $\gamma, \delta > 0$. Then for the averaged iterate $\bar\theta_t = \frac{1}{t}\sum_{s=1}^t \theta_{s-1}$, we have:*

$$\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta_*) \leqslant \frac{1}{\gamma t}\|\theta_0 - \theta_*\|_2^2 + 2L\delta^2 d^2 + 4d\frac{\gamma}{\delta^2}\sigma^2. \tag{11.16}$$

Proof Following previous SGD proofs, we 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 and using lemma 11.2,

$$\begin{aligned}
&\mathbb{E}\big[\|\theta_s-\theta_*\|_2^2|\mathcal{F}_{s-1}\big]\\
\leqslant\ & \|\theta_{s-1}-\theta_*\|_2^2-2\gamma F_\delta'(\theta_{s-1})^\top(\theta_{s-1}-\theta_*)\\
&\qquad +2\gamma^2\mathbb{E}\Big[\Big\|\frac{1}{\delta}\big(F(\theta_{s-1}+\delta z_s)-F(\theta_{s-1})\big)z_s\Big\|_2^2\Big|\mathcal{F}_{s-1}\Big]+2\frac{\gamma^2}{\delta^2}\mathbb{E}[\varepsilon_s^2\|z_s\|_2^2]\\
\leqslant\ & \|\theta_{s-1}-\theta_*\|_2^2-2\gamma F_\delta'(\theta_{s-1})^\top(\theta_{s-1}-\theta_*)\\
&\qquad +2\gamma^2\cdot\Big[\frac{L^2\delta^2}{2}15d^3+6d\|F'(\theta_{s-1})\|_2^2\Big]+2\frac{\gamma^2}{\delta^2}\cdot 2d\sigma^2, \text{ using equation (11.15).}
\end{aligned}$$

Moreover, using co-coercivity (proposition 5.4) and $F'(\theta_*)=0$, we get

$$\begin{aligned}
\mathbb{E}\big[\|\theta_s-\theta_*\|_2^2|\mathcal{F}_{s-1}\big] &\leqslant \|\theta_{s-1}-\theta_*\|_2^2-2\gamma\big[F_\delta(\theta_{s-1})-F_\delta(\theta_*)\big]\\
&\qquad +15\gamma^2L^2\delta^2d^3+24L\gamma^2d\big[F(\theta_{s-1})-F(\theta_*)\big]+4d\frac{\gamma^2}{\delta^2}\sigma^2\\
&\leqslant \|\theta_{s-1}-\theta_*\|_2^2-2\gamma\big[F(\theta_{s-1})-F(\theta_*)\big]+2\gamma\cdot\frac{L}{2}\delta^2 d\\
&\qquad +15\gamma^2L^2\delta^2d^3+24L\gamma^2d\big[F(\theta_{s-1})-F(\theta_*)\big]+4d\frac{\gamma^2}{\delta^2}\sigma^2,
\end{aligned}$$

using equation (11.14) and $F_\delta(\theta_*)\geqslant F(\theta_*)$. Thus, if $\gamma\leqslant\frac{1}{24dL}$, we have $24L\gamma^2d\leqslant\gamma$ and we get

$$\begin{aligned}
\mathbb{E}\big[\|\theta_s-\theta_*\|_2^2|\mathcal{F}_{s-1}\big] &\leqslant \|\theta_{s-1}-\theta_*\|_2^2-\gamma\big[F(\theta_{s-1})-F(\theta_*)\big]+\gamma L\delta^2 d\\
&\qquad +\frac{15}{24}\gamma L\delta^2d^2+4d\frac{\gamma^2}{\delta^2}\sigma^2\\
&\leqslant \|\theta_{s-1}-\theta_*\|_2^2-\gamma\big[F(\theta_{s-1})-F(\theta_*)\big]+2\gamma L\delta^2d^2+4d\frac{\gamma^2}{\delta^2}\sigma^2,
\end{aligned}$$

leading to, taking full expectations,

$$\mathbb{E}\big[F(\theta_{s-1})\big]-F(\theta_*) \leqslant \frac{1}{\gamma}\Big(\mathbb{E}\big[\|\theta_{s-1}-\theta_*\|_2^2\big]-\mathbb{E}\big[\|\theta_s-\theta_*\|_2^2\big]\Big)+2L\delta^2d^2+4d\frac{\gamma}{\delta^2}\sigma^2.$$

Summing from $s=1$ to $s=t$, we get

$$\frac{1}{t}\sum_{s=1}^t\mathbb{E}\big[F(\theta_{s-1})\big]-F(\theta_*)\leqslant\frac{1}{\gamma t}\|\theta_0-\theta_*\|_2^2+2L\delta^2d^2+4d\frac{\gamma}{\delta^2}\sigma^2,$$

leading to equation (11.16) using Jensen's inequality. ■

We can now analyze various situations depending on the presence or absence of noise (see the empirical illustration in figure 11.1):

- If $\sigma=0$, then we can take δ as close to zero as possible and get the rate, with $\gamma=\frac{1}{24dL}$:

$$\mathbb{E}\big[F(\bar\theta_t)\big]-F(\theta_*)\leqslant\frac{24Ld}{t}\|\theta_0-\theta_*\|_2^2. \tag{11.17}$$

  As suggested at the beginning of section 11.2, we only lose a factor of d compared to regular GD in section 5.2.4.

- If $\sigma > 0$, we can optimize over δ to get (assuming that σ 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 in $O(1/t)$ only up to the noise level with a limiting bound σd. We can also use a step size γ that depends on the horizon t, by taking $\gamma = \frac{1}{24Ld}t^{-2/3}$, leading to the following bound that goes to 0 when t tends to infinity:

$$\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta_*) \quad \leqslant \quad \frac{d}{t^{1/3}}\big[24L\|\theta_0 - \theta_*\|_2^2 + \sigma\big].$$

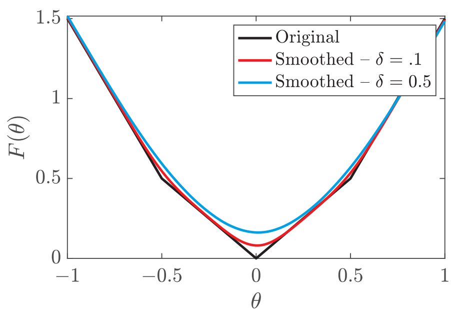
  We not only lose a factor of d in the bound, but the dependence in t is worsened from $1/t$ to $1/t^{1/3}$. Note that the dependence in σ could be improved if the noise level were known.

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 proportional to δ^{-2} due to two noisy evaluations, removing in equation (11.16) the last term, and thus achieving an improved behavior. For related lower bounds, see Duchi et al. (2015).

Exercise 11.6 *When two function evaluations are available, compute optimal values of δ and γ 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_δ as the convolution of 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 here:



Lemma 11.2 already showed that the function $F_\delta : \mathbb{R}^d \to \mathbb{R}$, defined in equation (11.13) has gradient equal to $F_\delta'(\theta) = \frac{1}{\delta}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[F(\theta+\delta z)z\big] = \frac{1}{\delta}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\big[(F(\theta+\delta z) - F(\theta))z\big]$.

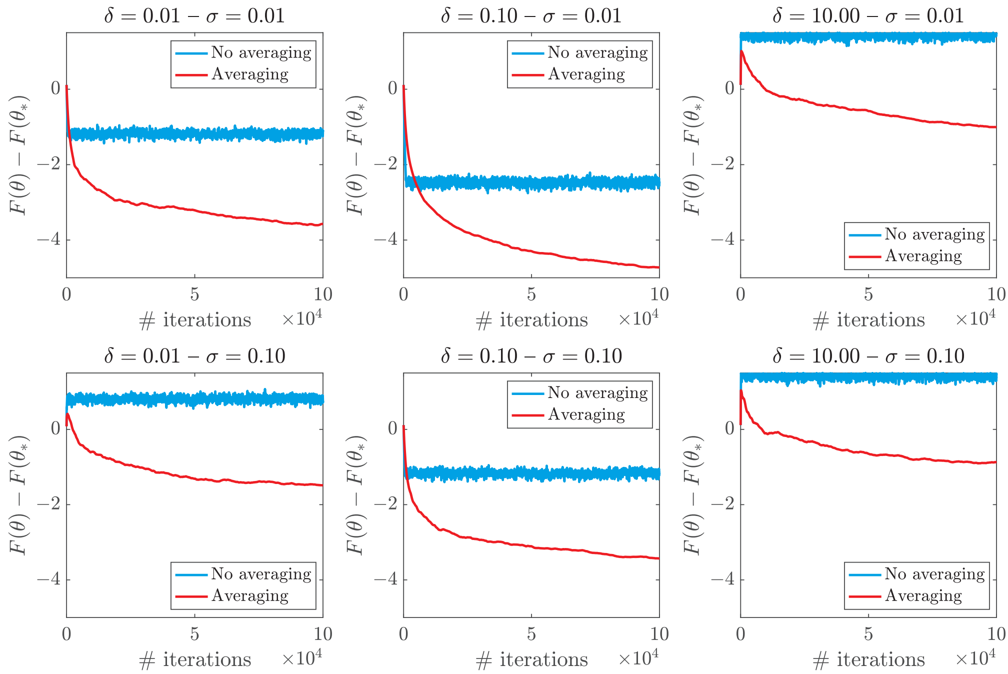


Figure 11.1. Zeroth-order optimization with Gaussian smoothing on a quadratic function F in dimension $d = 10$, with step size $\gamma = 1/(4Ld)$: two levels of noise added to the function values, $\sigma = 0.01$ (top), and $\sigma = 0.1$ (bottom), with three smoothing constants, $\delta = 0.01$ (left), $\delta = 0.1$ (middle), and $\delta = 10$ (right). Performance improves with smaller noise variance σ^2, while δ should be chosen to be not too large (then too much bias) and not too small (then too much variance).

Lemma 11.3 shows that it transforms a Lipschitz-continuous function into a smooth function.

Lemma 11.3 (Randomized smoothing) *Assume that F is B-Lipschitz-continuous. Then function $F_\delta : \mathbb{R}^d \to \mathbb{R}$, defined in equation (11.13) is B-Lipschitz-continuous and $\big(\frac{\sqrt{d}}{\delta}B\big)$-smooth. Moreover, $\forall \theta \in \mathbb{R}^d$, $|F_\delta(\theta) - F(\theta)| \leqslant B\delta\sqrt{d}$.*

Proof Function F_δ is $\big(\frac{\sqrt{d}}{\delta}B\big)$-smooth since for $\theta, \theta' \in \mathbb{R}^d$,

$$\|F_\delta'(\theta) - F_\delta'(\theta')\|_2 \leqslant \frac{1}{\delta}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\Big[|F(\theta+\delta z) - F(\theta'+\delta z)|\|z\|\Big] \leqslant \frac{B}{\delta}\|\theta-\theta'\|_2\mathbb{E}_{z\sim\mathcal{N}(0,I)}[\|z\|_2],$$

which is less than $B\sqrt{d}/\delta$. Moreover, $|F_\delta(\theta) - F(\theta)| \leqslant \mathbb{E}_{z\sim\mathcal{N}(0,I)}[|F(\theta+\delta z) - F(\theta)|] \leqslant B\mathbb{E}_{z\sim\mathcal{N}(0,I)}[\delta\|z\|_2]$, leading to the desired bound. ■

In other words, the expectation of the gradient estimate happens to be exactly the gradient of a smoothed version F_δ of F. This will be used in the proof that follows. Moreover, the expression of F_δ' 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

$$\mathbb{E}\big[\|\hat{F}_\delta'(\theta)\|_2^2\big] \leqslant \mathbb{E}\big[B^2\|z\|_2^4\big] \leqslant 4B^2d^2.$$

We can now analyze GD and prove proposition 11.5 that gives a convergence rate that will be analyzed below its proof.

Proposition 11.5 (zeroth-order optimization–nonsmooth case) *Consider a function F that is a B-Lipschitz-continuous convex function with minimizer θ_*, and (θ_t) defined by the recursion in equation (11.12), with $\gamma, \delta > 0$. Then for the averaged iterate $\bar{\theta}_t = \frac{1}{t}\sum_{s=1}^t \theta_{s-1}$, we have:*

$$\mathbb{E}\big[F(\bar{\theta}_t)\big] - F(\theta_*) \leqslant \frac{1}{2\gamma t}\|\theta_0 - \theta_*\|_2^2 + 4\gamma B^2d^2 + \frac{\gamma}{\delta^2}\sigma^2 d + 2B\delta\sqrt{d}. \tag{11.18}$$

Proof We have, for θ_* 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

$$\frac{1}{t}\sum_{s=1}^{t} F(\theta_{s-1}) - F(\theta_*) \quad \leqslant \quad \frac{1}{2\gamma t}\|\theta_0-\theta_*\|_2^2 + 4\gamma B^2 d^2 + \frac{\gamma}{\delta^2}\sigma^2 d + 2B\delta\sqrt{d}.$$

Jensen's inequality then leads to equation (11.18). ■

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 δ to be as small as possible so rounding errors do not perturb the finite differences; we then obtain the rate $\frac{1}{2\gamma t}\|\theta_0-\theta_*\|_2^2 + 4\gamma B^2 d^2$, losing a factor of d (after γ is optimized) compared to the standard subgradient method studied in section 5.3.
- When $\sigma > 0$, then we can optimize over δ, with $\delta^3 = \gamma\sigma^2 B^{-1}\sqrt{d}$. We then get

$$\mathbb{E}\big[F(\bar\theta_t)\big] - F(\theta) \quad \leqslant \quad \frac{1}{2\gamma t}\|\theta_0-\theta\|_2^2 + 4\gamma B^2 d^2 + 3d^{2/3}\gamma^{1/3}\sigma^{2/3}B^{2/3}.$$

  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

$$\frac{d^{1/2}}{2t^{1/4}}\Big(B^2\|\theta_0-\theta\|_2^2 + 6\sigma^{2/3}\Big) + 4\frac{d^{3/2}}{t^{3/4}}.$$

  Note that the dependence in t is not optimal; see Agarwal et al. (2013) for an improved rate proportional to $t^{-1/2}$ (but a worse dependence in d).

11.2.3 Extensions

In this section on zeroth-order algorithms, we have focused on optimization algorithms with potentially stochastic noise, with a criterion that 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, and then to using the regret criterion in equation (11.1). Online zeroth-order optimization is significantly more complicated, and in section 11.3, we will focus only on multiarmed 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 Multiarmed Bandits

This section aims to provide the simplest results for multiarmed stochastic bandits. There is an extensive and rich body of literature; for instance, see Bubeck and Cesa-Bianchi (2012), Lattimore and Szepesvári (2020), and Slivkins (2019) for more detailed accounts.

Multiarmed 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 applies to clinical trials, prediction of clicks on web pages, and routing in networks.

We consider $k \geqslant 2$ potential arms, each associated with a mean $\mu^{(i)} \in \mathbb{R}$, $i \in \{1, \ldots, k\}$. Every time we select arm i, we receive a reward sampled independent of all other rewards and the previous arm choices from a sub-Gaussian distribution with mean $\mu^{(i)}$, and sub-Gaussian parameter σ. At time s, we select arm i_s based on the information $\mathcal{F}_{s-1}$ up to time $s-1$ (i.e., the rewards received at time $s-1$ and before) and receive reward r_s. In this chapter, we focus on plain bandits, noting that many variations also 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

$$R_t = t \cdot \max_{i \in \{1,\ldots,k\}} \mu^{(i)} - \sum_{s=1}^{t} \mathbb{E}[r_s].$$

As opposed to online learning in section 11.1, here we are not dividing the regret by t.

Denoting $\Delta^{(j)} = \max_{i\in\{1,\ldots,k\}} \mu^{(i)} - \mu^{(j)} \geqslant 0$ as the difference between the mean of the best arm and the mean of arm j, and $n_t^{(j)}$ as the number of times that arm j was selected in the first t iterations, we can express the regret as

$$R_t = \sum_{j=1}^{k} \Delta^{(j)} \mathbb{E}\big[n_t^{(j)}\big]. \tag{11.19}$$

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 mean 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 that we select rewards from a sequence of independent and identically distributed (i.i.d.) samples $x_a^{(j)}$ with mean $\mu^{(j)}$ from each arm. This implies that as we select some arms multiple times, we get a more accurate estimate of $\mu^{(j)}$, as the expected squared distance between $\hat{\mu}_t^{(j)}$ and $\mu^{(j)}$ is proportional to $1/n_t^{(j)}$. To simplify the exposition, we ignore the equality cases among the various estimated values of $\hat{\mu}_t^{(j)}$, which is safe so 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 equation (11.19) 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 pure exploration strategy involved ignoring the online estimates $\hat{\mu}_t^{(j)}$. The pure exploitation strategy does the opposite, 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 nonzero probability that the best arm will never be selected again.

Exercise 11.7 *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 m estimates $\hat{\mu}^{(1)}, \dots, \hat{\mu}^{(k)}$, which are all independent random variables with means $\mu^{(1)}, \dots, \mu^{(k)}$ and sub-Gaussian parameters σ^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 equation (11.19), 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 mk steps, for which this is the exact contribution of the regret; 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 $\Delta^{(i_*)} = 0$):

$$\begin{aligned} R_t &\leqslant m\sum_{j=1}^k \Delta^{(j)} + (t-mk)\sum_{j=1}^k \Delta^{(j)} \mathbb{P}(\hat{\mu}_{mk}^{(j)} \geqslant \hat{\mu}_{mk}^{(i_*)}) \\ &\leqslant m\sum_{j\neq i_*} \Delta^{(j)} + t\sum_{j\neq i_*} \Delta^{(j)} \exp\Big(-\frac{(\Delta^{(j)})^2 m}{4\sigma^2}\Big), \qquad (11.20) \end{aligned}$$

by using sub-Gaussian tail bounds (see section 1.2.1) on the difference of the m arm values between j and i_* (i.e., $\hat{\mu}_{mk}^{(i_*)} - \hat{\mu}_{mk}^{(j)}$ 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 $i \neq i_*$:

$$R_t \leqslant m\Delta + t\Delta \exp\Big(- \frac{\Delta^2 m}{4\sigma^2}\Big). \tag{11.21}$$

We can minimize it approximately with respect to m by taking the gradient with respect to m (assuming for a moment that it is not restricted to being an integer), leading to $\Delta = t\frac{\Delta^3}{4\sigma^2} \exp\big(- \frac{\Delta^2 m}{4\sigma^2}\big)$. We thus consider the following candidate $m = \max\big\{1, \big\lceil \frac{4\sigma^2}{\Delta^2} \log \frac{\Delta^2 t}{4\sigma^2} \big\rceil\big\}$, for which we can further bound equation (11.21) as follows:

$$\begin{aligned} R_t &\leqslant m\Delta + t\Delta \exp\Big(- \frac{\Delta^2}{4\sigma^2}\frac{4\sigma^2}{\Delta^2} \log \frac{\Delta^2 t}{4\sigma^2}\Big) = m\Delta + t\Delta \frac{4\sigma^2}{\Delta^2 t} \\ &\leqslant (1 + \frac{4\sigma^2}{\Delta^2} \log \frac{\Delta^2 t}{4\sigma^2})\Delta + \frac{4\sigma^2}{\Delta} = \Delta + \frac{4\sigma^2}{\Delta}\Big(1 + \log \frac{\Delta^2 t}{4\sigma^2}\Big). \end{aligned} \tag{11.22}$$

We can now obtain two different results, depending on the desired asymptotic dependence on the gap. The best dependence in t, is obtained by starting from equation (11.22), and using $\log \alpha \leqslant \alpha - 1$ for $\alpha = \frac{\Delta^2}{4\sigma^2}$, to get

$$R_t \leqslant \Delta + \frac{4\sigma^2}{\Delta}\Big(1 + \log \frac{\Delta^2}{4\sigma^2} + \log(t)\Big) \leqslant \Delta + \frac{4\sigma^2}{\Delta}\Big(\frac{\Delta^2}{4\sigma^2} + \log(t)\Big) = 2\Delta + \frac{4\sigma^2}{\Delta}\log(t), \tag{11.23}$$

with a dominant term in $\frac{4\sigma^2}{\Delta}\log(t)$. Another way of bounding with a worse dependence in t, but a better one in Δ, leads to, still starting from equation (11.22) and using $\log \alpha \leqslant \alpha - 1$, this time for $\alpha = \Delta\sqrt{t}/(2\sigma)$:

$$R_t \leqslant \Delta + \frac{4\sigma^2}{\Delta}\Big(1 + 2\log\frac{\Delta\sqrt{t}}{2\sigma}\Big) \leqslant \Delta + \frac{4\sigma^2}{\Delta}\Big(1 + 2\frac{\Delta\sqrt{t}}{2\sigma} - 2\Big) \leqslant \Delta + 4\sigma\sqrt{t}.$$

Combining the two upper bounds, we get the following regret bound for the explore-then-commit algorithm:

$$R_t \leqslant 2\Delta + 4 \min\Big\{\sigma\sqrt{t}, \frac{\sigma^2}{\Delta}\log(t)\Big\}. \tag{11.24}$$

We thus obtain in equation (11.24) a bound with two behaviors that are reminiscent of convex optimization in section 5.4, where we obtain rates in $O(1/(t\mu))$ for μ-strongly-convex functions and in $O(1/\sqrt{t})$ for convex functions: a gap-dependent asymptotic bound $\frac{\sigma^2}{\Delta}\log(t)$ with logarithmic dependence in $\log(t)$ but a dependence in Δ that explodes when the gap Δ is small. Then, the bound in $\sigma\sqrt{t}$ takes over, with a worse dependence in t but no dependence in Δ.

As shown in section 11.3.3, this simple algorithm will achieve the lower bound (up to constant factors) for all possible algorithms. However, this requires knowing Δ and t in advance to select m appropriately. As shown in exercise 11.8, without knowing Δ, a regret in $t^{2/3}$ can be achieved.

Exercise 11.8 *Show that with $m = t^{2/3}$ the bound in equation (11.21) is smaller than $(\Delta + \sigma)t^{2/3}$.*

More than two arms ($k \geqslant 2$). We use a different argument from the one leading to equation (11.20). 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. Using sub-Gaussian tail bounds and the union bound, we have $\mathbb{P}(\mathcal{A}^c) \leqslant k \exp(-\frac{r^2}{2\sigma^2})$.

If 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); moreover, if $\frac{2r}{\sqrt{m}} < \Delta^{\min} = \min_{i \neq i_*} \Delta^{(i)}$ and $\mathcal{A}$ is true, the optimal arm has to be selected (thus with zero regret). If $\mathcal{A}$ is not true, we pay a cost less than $\Delta^{\max} = \max_{i \neq i_*} \Delta^{(i)}$.

We can thus distinguish two cases (as with two arms), with or without a term exploding in $\Delta^{\min}$ (note that for $k = 2$, $\Delta^{\min} = \Delta^{\max} = \Delta$). If $\Delta^{\min} > \frac{2r}{\sqrt{m}}$, we can select $m = 4r^2/(\Delta^{\min})^2$ (we assume for simplicity that $m \geqslant 1$, dealing with the other case is left as an exercise), and get the bound

$$R_t = mk\Delta^{\max} + tk\Delta^{\max} \exp\Big(-\frac{r^2}{2\sigma^2}\Big) = k\Delta^{\max}\Big(\frac{4r^2}{(\Delta^{\min})^2} + t\exp\Big(-\frac{r^2}{2\sigma^2}\Big)\Big),$$

where the first term corresponds to the explore phase and the last term to the commit phase. We then get, taking $r^2 = 2\sigma^2 \log(t)$,

$$R_t \leqslant k\Delta^{\max}\Big(\frac{8\sigma^2 \log(t)}{(\Delta^{\min})^2} + 1\Big). \tag{11.25}$$

We recover, up to constants, the same gap-dependent bound as for $k = 2$ arms in equation (11.23), with a logarithmic dependence in t, but a potentially exploding dependence in $\Delta^{\max}/(\Delta^{\min})^2$.

Alternatively, in all cases (i.e., without the constraint $\Delta^{\min} > \frac{2r}{\sqrt{m}}$), the regret is less than

$$R_t \leqslant mk\Delta^{\max} + 2\frac{r}{\sqrt{m}}t + t\Delta^{\max}k \exp\Big(-\frac{r^2}{2\sigma^2}\Big),$$

where the first term corresponds to the explore phase and the last two terms to the commit phase. With $m^{3/2} \approx rt/(k\Delta^{\max})$, we can minimize the first two terms and get

$$R_t \leqslant 3(rt)^{2/3}(k\Delta^{\max})^{1/3} + \Delta^{\max}kt\exp\Big(-\frac{r^2}{2\sigma^2}\Big).$$

With $r = \sigma\sqrt{2\log(kt)}$, we then get $R_t \leqslant \Delta^{\max} + 3t^{2/3}k^{1/3}(\Delta^{\max})^{1/3}\sigma^{2/3}\big(2\log(kt)\big)^{1/3}$, which grows as $t^{2/3}$ and does not achieve the lower bound (see a better algorithm in section 11.3.3).

ε-greedy. We can mix exploration and exploitation with the so-called "ε-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, Cesa-Bianchi, and Fischer, 2002).

11.3.3 Optimism in the Face of Uncertainty (♦)

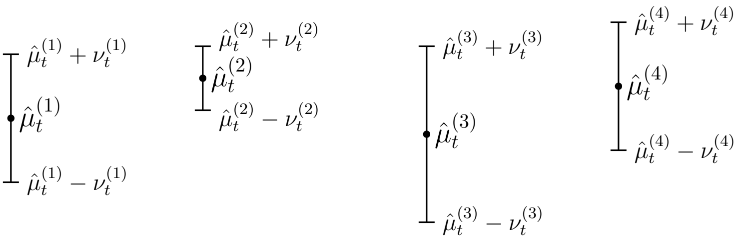
We consider the classical "upper confidence bound (UCB)" algorithm (Auer, Cesa-Bianchi, and Fischer, 2002), whose principle is simple. As arms are being selected, confidence intervals for the values of each arm are maintained as $[\hat{\mu}_t^{(i)} - \nu_t^{(i)}, \hat{\mu}_t^{(i)} + \nu_t^{(i)}]$. The 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, 2014).

The precise algorithm is as follows (assuming that σ 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 $\nu_k^{(i)} = \sqrt{2\rho\sigma^2 \log(k+1)/n_k^{(i)}} = \sqrt{2\rho\sigma^2 \log(k+1)}$, with $\rho > 0$ to be determined later.
- For all other $t > k$, select the arm i_t that 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 $i \in \{1, \ldots, k\}$, with the interval width $\nu_t^{(i)} = \sqrt{2\rho\sigma^2 \log(t+1)/n_t^{(i)}}$.

The confidence interval length for arm i is naturally proportional to $\sigma/\sqrt{n_t^{(i)}}$ with an extra factor that will ensure sublinear regret.

Thus, as illustrated for $k = 4$, we have k confidence intervals, and we select the arm with the largest upper confidence bound (here, $i = 4$):



The analysis consists in upper-bounding $\mathbb{E}\big[n_t^{(i)}\big]$ for $i \neq i_*$ and using equation (11.19) (i.e., $R_t = \sum_{i \neq i_*} \Delta^{(i)} \mathbb{E}\big[n_t^{(i)}\big]$), 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

$$\begin{aligned}
\mathbb{E}\big[n_t^{(i)}\big] &= \sum_{u=1}^{t} \mathbb{P}(i_u = i) \\
&= \sum_{u=1}^{t} \mathbb{P}(i_u = i, \hat{\mu}_{u-1}^{(i)} + \nu_{u-1}^{(i)} > \mu^{(i_*)}) + \sum_{u=1}^{t} \mathbb{P}(i_u = i, \hat{\mu}_{u-1}^{(i)} + \nu_{u-1}^{(i)} \leqslant \mu^{(i_*)}) \\
&\leqslant \sum_{u=1}^{t} \mathbb{P}\big(i_u = i, \hat{\mu}_{u-1}^{(i)} + \nu_{u-1}^{(i)} > \mu^{(i_*)}\big) + \sum_{u=1}^{t} \mathbb{P}\big(\hat{\mu}_{u-1}^{(i_*)} + \nu_{u-1}^{(i_*)} \leqslant \mu^{(i_*)}\big),
\end{aligned} \tag{11.26}$$

since if we select arm i at time u (i.e., $i_u = i$), then, by design of the upper confidence bounds, $\hat{\mu}^{(i_*)}_{u-1} + \nu^{(i_*)}_{u-1} \leqslant \hat{\mu}^{(i)}_{u-1} + \nu^{(i)}_{u-1}$.

To bound $\mathbb{P}\big(\hat{\mu}^{(i_*)}_{u-1} + \nu^{(i_*)}_{u-1} \leqslant \mu^{(i_*)}\big)$, it is tempting to apply a concentration inequality for the average of $n^{(i_*)}_{u-1}$ independent random variables distributed from the optimal arm distribution. However, $n^{(i_*)}_{u-1}$ is a random variable which is *not* independent from the rewards (because arm choice depends on past rewards). As done several times in this book to remove some unwanted randomness, we consider a uniform result based on a union bound. We thus use our sequence of i.i.d. samples $x^{(i_*)}_a$, $a \geqslant 1$, 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 $u \in \{1, \dots, t\}$,

$$\mathbb{P}\big(\hat{\mu}^{(i_*)}_{u-1} + \nu^{(i_*)}_{u-1} \leqslant \mu^{(i_*)}\big) \leqslant \sum_{s=1}^{u-1} \exp(-\rho \log(u)) \leqslant \frac{1}{u^{\rho-1}}.$$

Assuming $\rho > 2$, we can then use a comparison with an integral to get

$$\sum_{u=1}^{t-1} \mathbb{P}\big(\hat{\mu}^{(i_*)}_{u-1} + \nu^{(i_*)}_{u-1} \leqslant \mu^{(i_*)}\big) \leqslant \int_1^t \frac{1}{u^{\rho-1}} du = \frac{1}{\rho-2}(1 - t^{2-\rho}) \leqslant \frac{1}{\rho-2}.$$

Thus, the rightmost term in equation (11.26) is less than $\frac{1}{\rho-2}$. We now bound the left term in equation (11.26) as follows, for $t \geqslant k$:

$$\begin{aligned}
& \sum_{u=1}^{t} \mathbb{P}\big(i_u = i, \hat{\mu}^{(i)}_{u-1} + \nu^{(i)}_{u-1} > \mu^{(i_*)}\big) \\
= \ & \sum_{u=1}^{t} \mathbb{P}\Big(i_u = i, \hat{\mu}^{(i)}_{u-1} + \sqrt{2\rho\sigma^2 \log(u)/n^{(i)}_{u-1}} > \mu^{(i_*)}\Big) \text{ by definition of } \nu^{(i)}_{u-1}, \\
\leqslant \ & \sum_{u=1}^{t} \mathbb{P}\Big(i_u = i, \hat{\mu}^{(i)}_{u-1} + \sqrt{2\rho\sigma^2 \log(t)/n^{(i)}_{u-1}} > \mu^{(i_*)}\Big) \text{ since } u \leqslant t, \\
= \ & \sum_{u=1}^{t} \sum_{s=1}^{u-1} \mathbb{P}\Big(i_u = i, n^{(i)}_{u-1} = s, \frac{1}{s}\sum_{a=1}^{s} x^{(i)}_a + \sqrt{2\rho\sigma^2 \log(t)/s} > \mu^{(i_*)}\Big).
\end{aligned}$$

We can now swap the two summations to get the bound:

$$\begin{aligned}
& \sum_{s=1}^{t-1} \sum_{u=s+1}^{t} \mathbb{P}\Big(i_u = i, n^{(i)}_{u-1} = s, \frac{1}{s}\sum_{a=1}^{s} x^{(i)}_a + \sqrt{2\rho\sigma^2 \log(t)/s} > \mu^{(i_*)}\Big) \\
\leqslant \ & \sum_{s=1}^{t-1} \mathbb{P}\Big(\frac{1}{s}\sum_{a=1}^{s} x^{(i)}_a + \sqrt{2\rho\sigma^2 \log(t)/s} > \mu^{(i_*)}\Big),
\end{aligned} \tag{11.27}$$

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 nontrivial (i.e., less than 1) as soon as

$\Delta^{(i)} \geqslant \sqrt{2\rho\sigma^2 \log(t)/s}$, leading to the following bound for the term in equation (11.27):

$$\sum_{s=1}^{+\infty} \exp\big[-\big(\Delta^{(i)} - \sqrt{2\rho\sigma^2 \log(t)/s}\big)_+^2 s/(2\sigma^2)\big].$$

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 the following, 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 1, and get a term equal to $\frac{8\rho\sigma^2 \log(t)}{(\Delta^{(i)})^2}$. Thus, overall, we get that

$$\mathbb{E}\big[n_t^{(i)}\big] \leqslant \frac{1}{\rho-2} + \frac{8\rho\sigma^2 \log(t)}{(\Delta^{(i)})^2} + \frac{8\sigma^2}{(\Delta^{(i)})^2}.$$

For $\rho = 3$, this leads to a regret bound for the UCB algorithm:

$$R_t \leqslant \sum_{i \neq i_*} \Delta^{(i)} \Big(1 + \frac{\sigma^2}{(\Delta^{(i)})^2}(24 \log t + 8)\Big). \tag{11.28}$$

For $k = 2$, it is the same (up to constants) as the explore-then-commit algorithm in equation (11.23), but without the need to know the gap between means in advance (this applies as well to $k > 2$). For $k > 2$, it has the same logarithmic dependence in t as equation (11.25), but an asymptotic constant proportional to $\sum_{i \neq i_*} \frac{\sigma^2}{\Delta^{(i)}}$ rather than $k \frac{\sigma^2}{\min_{i \neq i_*} \Delta^{(i)}}$, which is a substantial gain when only few arms have means close to the maximal one. Moreover, it happens to achieve the lower bound (up to constants; see the following discussion on lower bounds).

As for explore-then-commit, we can 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, for $\rho = 3$,

$$\begin{aligned} R_t &= \sum_{i,\ \Delta^{(i)} < \Delta} \Delta^{(i)} \mathbb{E}[n_t^{(i)}] + \sum_{i,\ \Delta^{(i)} \geqslant \Delta} \Delta^{(i)} \mathbb{E}[n_t^{(i)}] \text{ for a certain } \Delta, \\ &\leqslant t\Delta + \sum_{i,\ \Delta^{(i)} \geqslant \Delta} \Delta^{(i)} \Big(1 + \frac{\sigma^2}{(\Delta^{(i)})^2}(24 \log t + 8)\Big) \text{ as in equation (11.28)}, \\ &\leqslant t\Delta + \sum_i \Delta^{(i)} + k\frac{\sigma^2}{\Delta}(24 \log t + 8) \leqslant \sum_i \Delta^{(i)} + 4\sigma\sqrt{2kt(3 \log t + 1)}, \end{aligned} \tag{11.29}$$

by optimizing over Δ, leading to a bound which is an improvement over explore-then-commit and also optimal up to logarithmic terms (as discussed next), Note that if $\rho > 2$, then we only pay an increase in the bound proportional to ρ, while if $\rho \leqslant 1$, the upper bound on regret can start to be superlinear (and thus vacuous).

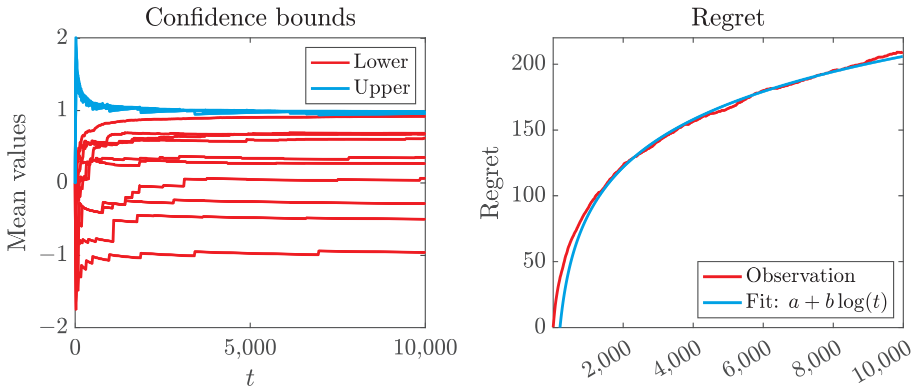


Figure 11.2. Upper-confidence bounds for $k = 10$ Bernoulli arms with random means: plot of upper and lower bounds as a function of time t (left), and regret (right).

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, an improved order $\log(t) \sum_{i \neq i_*} \frac{\sigma^2}{\Delta^{(i)}}$ (see, e.g., Bubeck and Cesa-Bianchi, 2012). From the regret bounds in equation (11.28) and equation (11.29), we can see that UCB is optimal up to logarithmic terms.

Illustration. In figure 11.2, we plot the performance of the UCB algorithm with $k = 10$ arms. We can observe in the left plot that upper confidence bounds tend to converge to each other, while the right plot highlights the logarithmic dependence of the regret.

Thomson sampling. Another common strategy for stochastic multiarmed bandits is based on Bayesian inference. Assuming a common prior distribution for each mean $\mu^{(j)}$, and, at time t, given the sequence $S_t^{(j)} \in \mathbb{R}^{n_t^{(j)}}$ of $n_t^{(j)}$ received rewards for arm $j \in \{1, \dots, k\}$, we consider the posterior distribution $p(\mu^{(j)} | S_t^{(j)})$. Thomson sampling corresponds to sampling at time t each ν_j independently from this posterior distribution and selecting the arm j with the largest ν_j. This algorithm comes with guarantees similar to UCB, but can be applied more generally as it implicitly relies on Bayesian confidence intervals; however, it is only implementable if posterior distributions can be easily accessed. See Russo et al. (2018) for details.

11.3.4 Adversarial Bandits (♦)

We finish this section on multiarmed bandits by studying a nonstochastic setup referred to as the "adversarial setup." We now have arbitrary reward vectors $\mu_t \in [0, 1]^k$, $t \geqslant 1$, which may vary with time and are assumed to be deterministic, and at each time step, we choose an arm i_t and receive reward $\mu_t^{(i_t)}$. As for online convex optimization in section 11.1, 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 setup, there is no randomness in the environment and we receive rewards that are elements of $[0,1]$. 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 a function of the past information), then there is a reward sequence (μ_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 fewer 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. We thus minimize a sequence of linear functions on the simplex in k dimensions, $F_t : \pi \to \mu_t^\top \pi$, with observation of the gradient μ_t (which does not depend on π). The Hedge algorithm (Freund and Schapire, 1997) consists in starting with π_0 uniform and updating π_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, applied with no randomness to the linear functions F_t, with the entropy mirror map. We thus immediately get from proposition 11.3 a normalized regret which is less than $\sqrt{2\log(k)}/\sqrt{t}$ for the choice $\gamma = \sqrt{2\log(k)}/\sqrt{t}$. We therefore get an 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 component $\mu_t^{(i_t)}$, where i_t is sampled from π_{t-1}. The estimator suggested by Auer, Cesa-Bianchi, Freund, and Schapire (2002) is an importance sampling estimator and leads to the "Exp3" algorithm. It is defined as $\hat\mu_t^{(i)} = \mu_t^{(i_t)} 1_{i=i_t}/\pi_{t-1}^{(i)}$; it thus has expectation μ_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 nonexplosive bound. However, an improvement on proposition 11.3 may be obtained for the simplex.

Proposition 11.6 *The mirror descent recursion in equation (11.7), for $\mathcal{C}$ the simplex and Φ 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 nonnegative components and $\mathbb{E}\big[\sum_{i=1}^d \theta_{s-1}^{(i)} (g_s^{(i)})^2 | \mathcal{F}_{s-1}\big] \leqslant B^2$ 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 proposition 11.3, we have $D_\Phi(\theta_t, \theta_{t-1}) + \gamma g_t^\top (\theta_t - \theta_{t-1}) = -\gamma \sum_{i=1}^d \theta_{t-1}^{(i)} g_t^{(i)} - \log\big(\sum_{i=1}^d \theta_{t-1}^{(i)} \exp(-\gamma g_t^{(i)})\big)$, which, using exercise 1.19, is greater than the quantity $-\frac{\gamma^2}{2} \sum_{i=1}^d \theta_{t-1}^{(i)} (g_t^{(i)})^2$. This leads to the desired result. ■

We can now provide a regret bound for the Exp3 algorithm by using proposition 11.6 and noticing that we need to bound $\mathbb{E}[\sum_{i=1}^k \pi_{s-1}^{(i)} (\hat{\mu}_s^{(i)})^2 | \mathcal{F}_{s-1}] = \sum_{i=1}^k (\mu_t^{(i)})^2 \leqslant k$. This leads to, after optimizing with respect to the step size, a nonnormalized regret bound proportional to $\sqrt{kt \log k}$.

From adversarial to stochastic. In the stochastic setup, the UCB algorithm provided an unnormalized regret of order $\sqrt{kt \log t}$, which is the same (up to the logarithmic term) 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 gap-dependent logarithmic regret bound. Note that the optimal convergence rate is proportional to $\sqrt{kt}$ (Audibert and Bubeck, 2009).

11.4 Conclusion

In this chapter, we have provided extensions to the classical i.i.d. setting that is the book's main focus. In the convex case, algorithms and analyses 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 by Shalev-Shwartz (2011), Bubeck and Cesa-Bianchi (2012), Hazan (2022), Slivkins (2019), and Lattimore and Szepesvári (2020).

Chapter 12

Overparameterized Models

Chapter Summary

- A model is said to be overparameterized 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 gradient descent (GD) typically do not.
- Implicit regularization of GD: For linear models, when there are several minimizers (typically for overparameterized models), GD techniques tend to converge to the one with a minimum Euclidean norm.
- Double descent: For unregularized models learned with gradient descent techniques, as the number of parameters grows, the performance can exhibit a second descent after the testing error blows up, when the number of parameters goes beyond the number of observations.
- Global convergence of GD for two-layer neural networks: In the infinite width (and thus strongly overparameterized) limit, GD exhibits some globally convergent behavior despite the lack of convexity of such problems, which can be analyzed for simple architectures.

In this chapter, we will cover three recent topics within learning theory, all related to prediction models (such as neural networks or positive definite kernel methods) in the *overparameterized* 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 made numerically and statistically efficient by adding penalties to the empirical risk. In this section, we consider primarily nonpenalized problems and prove that some regularization will come from the choice of optimization algorithm, here gradient descent (GD), and possibly all its hyperparameters (e.g., initialization, step size).

The number of parameters is not what generally characterizes the generalization capabilities of regularized learning methods. See sections 3.6 and 9.2.3.

12.1 Implicit Bias of Gradient Descent

Given an optimization problem that corresponds to minimizing a function $\theta \mapsto F(\theta)$ over a parameter $\theta \in \mathbb{R}^d$, if there is a unique global minimizer θ_*, then the goal of optimization algorithms is to find this minimizer; that is, we want the tth iterate θ_t to be reliably in the vicinity of θ_*. When there are multiple minimizers (thus for a function that cannot be strongly convex), we could only show in chapter 5 that $F(\theta_t) - \inf_{\theta \in \mathbb{R}^d} F(\theta)$ is converging to zero if F is convex (and only if a minimizer exists; see section 5.2.4).

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. The main question is: *which one?* The selection principle behind the convergence toward one specific minimizer is referred to as the implicit regularization properties of optimization algorithms, here, GD and its variants.

This is especially interesting in machine learning because when $F(\theta)$ is the empirical risk 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., ℓ_2-norms as in chapters 3 and 7, or ℓ_1-norms as in chapters 8 and 9). In this section, we show that optimization algorithms may have similar regularizing effects. In a nutshell, GD usually leads to minimum ℓ_2-norm solutions, in a similar way that boosting algorithms were related to ℓ_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 of this chapter.

12.1.1 Least-Squares Regression

Now 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 (i.e., a whole affine subspace of) solutions such that $y = X\theta$ since the column space of X is the entire space $\mathbb{R}^n$ and θ has dimension $d > n$. We apply GD with step size $\gamma < \frac{1}{L} = \lambda_{\max}(\frac{1}{n}X^\top X)^{-1}$, which is equal to $\lambda_{\max}(\frac{1}{n}XX^\top)^{-1}$, starting from $\theta_0 = 0$ and leading to $\theta_t = \theta_{t-1} - \frac{\gamma}{n}X^\top(X\theta_{t-1} - y)$. Therefore, we have

$$X\theta_t - y = X\theta_{t-1} - y - \frac{\gamma}{n}XX^\top(X\theta_{t-1} - y) = \Big(I - \frac{\gamma}{n}XX^\top\Big)(X\theta_{t-1} - y),$$

[1] We use X as a notation for the design matrix in order to highlight that in this section, we will consider predictions that are also linear in x.

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 $\|X\theta_t - y\|_2^2 \leqslant \big(1 - \frac{\gamma}{n}\lambda_{\min}(XX^\top)\big)^{2t}\|y\|_2^2$, and hence linear convergence of $X\theta_t$ toward y, with a convergence rate depending on the condition number of the kernel matrix $XX^\top$.

Moreover, when started at $\theta_0 = 0$, GD techniques (whether stochastic or not) will always have iterates θ_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 α_t converges to $K^{-1}y$, and thus $\theta_t = X^\top\alpha_t$ converges to $X^\top K^{-1}y$. One may have recognized in $X^\top K^{-1} = X^\top(XX^\top)^{-1}$ the pseudo-inverse[2] of X, and hence $X^\top K^{-1}y$ is the minimum ℓ_2-norm solution of $\{X\theta = y\}$, as shown next with standard Lagrangian duality (Boyd and Vandenberghe, 2004):

$$\begin{aligned}
\inf_{\theta\in\mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \text{ such that } y = X\theta &= \inf_{\theta\in\mathbb{R}^d}\sup_{\alpha\in\mathbb{R}^n} \frac{1}{2}\|\theta\|_2^2 + \alpha^\top(y - X\theta)\\
&= \sup_{\alpha\in\mathbb{R}^n} \alpha^\top y - \frac{1}{2}\|X^\top\alpha\|_2^2 \text{ with } \theta = X^\top\alpha \text{ at optimum,}\\
&= \sup_{\alpha\in\mathbb{R}^n} \alpha^\top y - \frac{1}{2}\alpha^\top K\alpha. \qquad (12.2)
\end{aligned}$$

The problem in equation (12.2) is exactly solved for $\alpha = K^{-1}y$, with $\theta = X^\top\alpha$ at optimum. Note that in chapter 7, we used this formula for function interpolation to compare different reproducing kernel Hilbert spaces (RKHSs) (see proposition 7.2).

Łojasiewicz's inequality (♦). It turns out that the linear convergence obtained from equation (12.1) can be obtained directly for any L-smooth function, for which we have the so-called Łojasiewicz's inequality:

$$\forall\theta\in\mathbb{R}^d,\ F(\theta) - F(\theta_*) \leqslant \frac{1}{2\mu}\|F'(\theta)\|_2^2 \tag{12.3}$$

for some $\mu > 0$ and any minimizer θ_*.

In chapter 5, we have seen that this is a consequence of μ-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 θ_*,

$$\begin{aligned}
\|F'(\theta)\|_2^2 &= \Big\|\frac{1}{n}X^\top X(\theta-\theta_*)\Big\|_2^2 = \frac{1}{n^2}(\theta-\theta_*)^\top X^\top XX^\top X(\theta-\theta_*)\\
&\geqslant \frac{\lambda_{\min}^+(XX^\top)}{n^2}(\theta-\theta_*)^\top X^\top X(\theta-\theta_*),
\end{aligned}$$

[2]See https://en.wikipedia.org/wiki/Moore-Penrose_inverse.

where $\lambda_{\min}^+(XX^\top) = \lambda_{\min}^+(X^\top X)$ is the smallest *nonzero* 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, equation (12.3) is satisfied with $\mu = \frac{1}{n}\lambda_{\min}^+(K)$. Note that this also includes the strongly convex case since $\lambda_{\min}^+(X^\top X) \geqslant \lambda_{\min}(X^\top X)$.

When equation (12.3) is satisfied, we have for the tth iterate of GD with step size $\gamma = 1/L$, following the analysis of chapter 5 (proposition 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) and Karimi et al. (2016) for more details.

12.1.2 Separable Classification

We now consider binary classification with the smooth convex surrogate introduced in section 4.1.1 leading to logistic regression; that is, for $y_i \in \{-1,1\}$, $i=1,\dots,n$,

$$F(\theta) = \frac{1}{n}\sum_{i=1}^n \log(1+\exp(-y_i x_i^\top\theta)), \tag{12.4}$$

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 $\text{sign}(X\theta) = y$, which happens when $y\circ(X\theta)$ (where $\circ$ is the componentwise product) has strictly positive components. For $d > n$, if $XX^\top$ is invertible, such an interpolator always exists (e.g., the one for regression on y).

Maximum margin classifier. As in the case of regression, there are infinitely many perfect linear classifiers. Among them, it is tempting to define the one that maximizes the "margin," as described in section 4.1.2. Since $XX^\top$ is invertible, there is at least one $\eta \in \mathbb{R}^d$ of a unit norm such that $\forall i \in \{1,\dots,n\}$, $y_i x_i^\top\eta > 0$ (e.g., the unit norm vector associated to $X^\top(XX^\top)^{-1}y$). We denote by η_* the one in which the so-called "margin"

$$\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$$

is maximal (and thus strictly positive). We denote as $\frac{1}{\rho} > 0$ the value of this maximization problem. Then, using Lagrange duality, we write

$$\begin{aligned}
\frac{1}{\rho} = \sup_{\|\eta\|_2 \leqslant 1} \min_{i \in \{1,\dots,n\}} y_i x_i^\top \eta &= \sup_{\|\eta\|_2 \leqslant 1, t \in \mathbb{R}} t \ \text{ such that } \ \forall i \in \{1,\dots,n\}, \ y_i x_i^\top \eta \geqslant t \\
&= \inf_{\alpha \in \mathbb{R}_+^n} \sup_{\|\eta\|_2 \leqslant 1, t \in \mathbb{R}} t + \sum_{i=1}^n \alpha_i (y_i x_i^\top \eta - t) \\
&= \inf_{\alpha \in \mathbb{R}_+^n} \Big\| \sum_{i=1}^n \alpha_i y_i x_i \Big\|_2 \text{ such that } \sum_{i=1}^n \alpha_i = 1, \qquad (12.5)
\end{aligned}$$

with $\eta \propto \sum_{i=1}^n \alpha_i y_i x_i$ at optimum. Moreover, by complementary slackness, at optimality, a nonnegative α_i is nonzero only for i attaining the minimum in $\min_{i \in \{1,\dots,n\}} y_i x_i^\top \eta$.

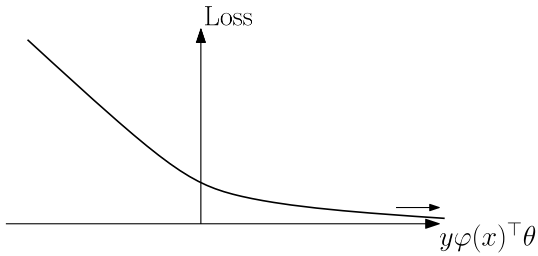
Reformulation as a support vector machine (SVM). Because we only consider the sign of the linear function, there are two equivalent ways to write the max-margin problem. Indeed, instead of maximizing the quantity $\min_{i \in \{1,\dots,n\}} y_i x_i^\top \eta$ and constraining $\|\eta\|_2$, we can decide to constrain the first and minimize over the second. In other words, we can see η_* as the unit-norm direction of the solution θ_* of the following optimization problem (with nonnegative Lagrange multipliers $\alpha_1, \dots, \alpha_n$):

$$\begin{aligned}
\inf_{\theta \in \mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \ \text{ such that } \ y \circ (X\theta) \geqslant 1_n &= \inf_{\theta \in \mathbb{R}^d} \sup_{\alpha \in \mathbb{R}_+^n} \frac{1}{2}\|\theta\|_2^2 + \alpha^\top (1_n - y \circ (X\theta)) \\
&= \sup_{\alpha \in \mathbb{R}_+^n} \alpha^\top 1_n - \frac{1}{2}\|X^\top (y \circ \alpha)\|_2^2, \qquad (12.6)
\end{aligned}$$

with $\theta = X^\top (y \circ \alpha)$ at optimum (the optimal dual vectors $\alpha \in \mathbb{R}^n$ from equations (12.5) and (12.6) are proportional to each other). Note that here, $y \circ (X\theta) \geqslant 1_n$ is the compact formulation of the inequality constraints $\forall i \in \{1,\dots,n\}$, $y_i x_i^\top \theta \geqslant 1$. Given η, θ is equal to $\eta / \min_{i \in \{1,\dots,n\}} y_i x_i^\top \eta$, so the optimal value of the previous optimization problem is $\frac{1}{2}\rho^2$.

The optimal vector θ_* is the solution of the separable SVM from section 4.1.2 with vanishing regularization parameter; that is, the minimizer of $\frac{1}{2}\|\theta\|_2^2 + C \sum_{i=1}^n (1 - y_i x_i^\top \theta)_+$ for C that is 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 equation (12.4) has an infimum equal to zero, which is not attained. However, for any sequence θ_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, as GD reaches optimality by convexity of F, it cannot converge to a point and has to diverge. It turns out that it diverges along a direction; that is, $\|\theta_t\|_2 \to +\infty$, with $\frac{\theta_t}{\|\theta_t\|_2} \to \eta$ for some $\eta \in \mathbb{R}^d$ of unit ℓ_2-norm. This direction η has to lead to perfect classification (i.e., $y_i x_i^\top \eta > 0$ for all $i \in \{1, \dots, n\}$). Among all of them, it is exactly the one with maximum margin as defined in the previous paragraphs (i.e., which maximizes $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta > 0$). See Gunasekar et al. (2018) for a detailed proof. For brevity, we give a simple argument for 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 asymptotically the case in an overparameterized regime). Moreover, we replace the GD recursion $\theta_t = \theta_{t-1} - \gamma G'(\theta_{t-1})$ by the gradient flow

$$\xi'(\tau) = -G'(\xi(\tau)). \tag{12.7}$$

This ordinary differential equation (ODE) approximates GD for vanishing step sizes, as $\xi(\gamma t) \approx \theta_t$ for γ tending to zero. The use of gradient flows instead of GD 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 $\alpha_i = \frac{\exp(-y_i x_i^\top \theta)}{\sum_{j=1}^n \exp(-y_j x_j^\top \theta)}$ and $i \in \{1, \dots, n\}$, leading to α in the simplex (with nonnegative components and summing to 1). Thus, from equation (12.5) defining the maximum margin hyperplane, we get

$$\|G'(\theta)\|_2 \geqslant \frac{1}{\rho}. \tag{12.8}$$

Moreover, comparing maxima and soft maxima,[3] we get:

$$-\log(n) - \min_{i\in\{1,\dots,n\}} y_i x_i^\top \theta \ \leqslant\ G(\theta) \leqslant -\min_{i\in\{1,\dots,n\}} y_i x_i^\top \theta. \tag{12.9}$$

[3] We use $0 \geqslant \log\big(\frac{1}{n}\sum_{i=1}^n e^{z_i}\big) - \max_{i\in\{1,\dots,n\}} z_i = \log\big(\frac{1}{n}\sum_{i=1}^n e^{z_i - \max_{j\in\{1,\dots,n\}} z_j}\big) \geqslant \log(1/n)$.

We thus have a flow $\tau \mapsto \xi(\tau)$ that cannot converge as by equations (12.7) and (12.8), $\|\xi'(\tau)\|_2 \geqslant 1/\rho$. Moreover, by equation (12.9), 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.

Using $\frac{d}{d\tau}G(\xi(\tau)) = G'(\xi(\tau))^\top \xi'(\tau) = -\|G'(\xi(\tau))\|_2^2$, we get:

$$\begin{aligned}\min_{i\in\{1,\dots,n\}} y_i x_i^\top \xi(\tau) &\geqslant -G(\xi(\tau)) - \log(n) \text{ from equation (12.9)},\\ &= -G(\xi(0)) + \int_0^\tau \|G'(\xi(u))\|_2^2 du - \log(n).\end{aligned}$$

Then, using equation (12.8) twice, we get:

$$\min_{i\in\{1,\dots,n\}} y_i x_i^\top \xi(\tau) \geqslant -G(\xi(0)) + \frac{1}{\rho}\int_0^\tau \|G'(\xi(u))\|_2 du - \log(n) \tag{12.10}$$

$$\geqslant -G(\xi(0)) + \frac{1}{\rho^2}\tau - \log(n). \tag{12.11}$$

Note that equation (12.10) is not needed to derive equation (12.11), but will be used later. Thus, from equation (12.11), for $\tau \geqslant \rho^2\big[\log(n) + G(\xi(0))\big]$, we have a nonnegative lower bound on the margin. 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

$$\|\xi(\tau)\|_2 \leqslant \|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du. \tag{12.12}$$

We thus get, from equations (12.10) and (12.12),

$$\begin{aligned}\min_{i\in\{1,\dots,n\}} y_i x_i^\top\big(\tfrac{\xi(\tau)}{\|\xi(\tau)\|_2}\big) &\geqslant \frac{-G(\xi(0)) + \frac{1}{\rho}\int_0^\tau \|G'(\xi(u))\|_2 du - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du}\\ &= \frac{-G(\xi(0)) + \frac{1}{\rho}\big(\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du\big) - \frac{1}{\rho}\|\xi(0)\|_2 - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du}\\ &= \frac{1}{\rho} + \frac{-G(\xi(0)) - \frac{1}{\rho}\|\xi(0)\|_2 - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du}\\ &\geqslant \frac{1}{\rho} - \frac{G(\xi(0)) + \frac{1}{\rho}\|\xi(0)\|_2 + \log(n)}{\|\xi(0)\|_2 + \tau/\rho}, \text{ since } \int_0^\tau \|G'(\xi(u))\|_2 du \geqslant \frac{\tau}{\rho}.\end{aligned}$$

The lower bound that appears here tends to $\frac{1}{\rho}$ when τ tends to infinity, which is the maximal value. We thus get convergence to the maximum margin hyperplane.

Alternative (informal) 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 $F'(\theta) = -\frac{1}{n}\sum_{i=1}^n \frac{\exp(-y_i x_i^\top \theta)}{1+\exp(-y_i x_i^\top \theta)} y_i x_i$.

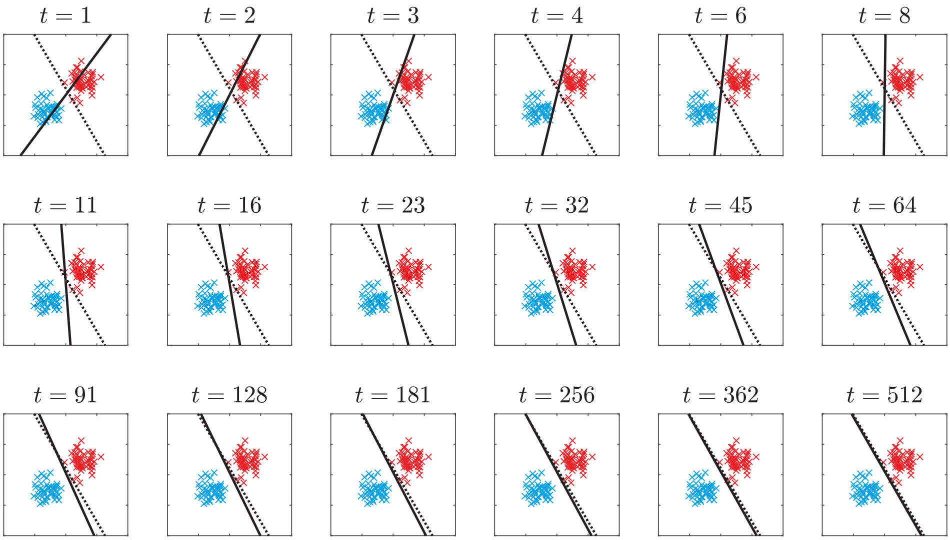


Figure 12.1. Logistic regression on separable data estimated with GD on the unregularized empirical risk, at various numbers of iterations t. This is implemented by minimizing the logistic loss function with data $\binom{x_i}{1} \in \mathbb{R}^3$. The dotted line represents the maximum margin hyperplane, while the solid line represents the current classification hyperplane.

We assume that we know a priori that $\|\theta_t\| \to +\infty$ and $\theta_t/\|\theta_t\|_2 \to \eta$. 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, of which $-y_i x_i^\top \eta$ is the largest. Moreover, all these values must be negative (indeed, we can only attain zero loss for well-classified training data). We denote this set as I. 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 $-\eta$, η has to be proportional to a vector $\sum_{i \in I} \alpha_i y_i x_i$, where $\alpha \geqslant 0$, and $\alpha_i = 0$ so long as i is not among the minimizers of $y_i x_i^\top \eta$. This is exactly the optimality condition for η_* in equation (12.5). Thus, $\eta = \eta_*$.

Summary. Overall, we obtain a classifier corresponding to a minimum ℓ_2-norm separating hyperplane. See examples in two dimensions shown in figure 12.1, where, after a few iterations, the linear classifier makes no error on the training data and then slowly converges to the maximum margin one. Note that GD on the logistic regression problem may not be the most efficient way to obtain a maximum margin hyperplane. See the slow convergence rates in $1/\log(t)$ derived by Soudry et al. (2018), Ji and Telgarsky (2018), and a simpler subgradient algorithm presented next.

Subgradient method for the hinge loss and perceptron (♦). For linearly separable data, dedicated algorithms to find the max-margin classifier can be obtained from optimization algorithms described in chapter 5. They are explicitly or implicitly based on the hinge loss. We consider the "margin" $\rho > 0$, defined earlier as in the SVM reformulation

$$\rho^2 = \inf_{\theta \in \mathbb{R}^d} \|\theta\|_2^2 \text{ such that } y \circ (X\theta) \geqslant 1_n. \tag{12.13}$$

To obtain a linear separator θ, 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

$$\theta_t = \theta_{t-1} + \gamma 1_{y_{i_t} x_{i_t}^\top \theta_{t-1} < 1} y_{i_t} x_{i_t}, \tag{12.14}$$

where $i_t \in \arg\min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta_{t-1}$, and γ is the step size. With θ_* being the minimizer in equation (12.13), we have $F(\theta_*) = 0 = \min_{\theta \in \mathbb{R}^d} F(\theta)$, and after t steps, following the analysis of proposition 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 shown here is less than ε so long 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 $t = \frac{\rho^2}{\gamma \varepsilon} = \frac{\rho^2 R^2}{\varepsilon^2}$, leading to an objective function less than $\varepsilon = \frac{\rho R}{\sqrt{t}}$. 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 equation (12.14) is a variation on the perceptron algorithm (Rosenblatt, 1958; Novikoff, 1962), as presented in exercise 12.1.

Exercise 12.1 *Extend this analysis 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 setup as exercise 12.1, consider the iteration, started at* θ_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 the 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 GD can be observed and analyzed in various models other than 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 a number of learning dynamics, which we study in continuous time for simplicity.

From gradient flow to mirror flow. The gradient flow dynamics on θ is the ODE:

$$\frac{d}{dt}\theta(t) = -F'(\theta(t)) = -\frac{1}{n}X^\top(X\theta(t) - y).$$

With the same proof as for GD in section 12.1.1, $\theta(t)$ converges exponentially fast to the minimum ℓ_2-norm interpolator. This can be extended to the continuous-time limit of mirror descent presented in section 11.1.3. If we consider a μ-strongly convex twice-differentiable 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 $\theta(\gamma k) = \tilde\theta_k$ (and interpolating between these values), leading to the ODE:

$$\frac{d}{dt}\big[\Phi'(\theta(t))\big] = -F'(\theta(t)), \tag{12.15}$$

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

$$\frac{d}{dt}\big[X\theta(t) - y\big] = X\frac{d}{dt}\theta(t) = -\frac{1}{n}X\Phi''(\theta(t))^{-1}X^\top\big[X\theta(t) - y\big],$$

which in turn leads to

$$\begin{aligned}\frac{d}{dt}\big[\|X\theta(t) - y\|_2^2\big] &= -\frac{2}{n}\big[X\theta(t) - y\big]^\top X\Phi''(\theta(t))^{-1}X^\top\big[X\theta(t) - y\big] \\ &\leqslant -\frac{2\lambda_{\min}(XX^\top)}{n\mu}\|X\theta(t) - y\|_2^2, \text{ using } \Phi''(\theta(t)) \succcurlyeq \mu I.\end{aligned}$$

Thus, like for the gradient flow dynamics, $X\theta(t)$ converges exponentially fast to y. Since from equation (12.15), $\Phi'(\theta)$ takes the form $\Phi'(\theta_0) + X^\top\alpha$ for some $\alpha \in \mathbb{R}^n$, the corresponding limit α_∞ (with the corresponding θ_∞ such that $\Phi'(\theta_\infty) = \Phi'(\theta_0) + X^\top\alpha_\infty$) of $\alpha(t)$ when t tends to infinity (the existence of such limits is left as an exercise) is such that $X\theta_\infty = y$ 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 $D_\Phi(\theta,\theta_0) = \Phi(\theta) - \Phi(\theta_0) - \Phi'(\theta_0)^\top(\theta - \theta_0)$.[4] Note that for $\Phi(\theta) = \frac{1}{2}\|\theta\|_2^2$, we recover back the result for GD. Hence, choosing a mirror map corresponds to selecting a solution to the interpolation problem by projecting the initialization through the Bregman divergence (see exercise 12.3).

Exercise 12.3 *Show that for all $\theta_* \in \{\theta \in \mathbb{R}^d,\ X\theta = y\}$, we have $D_\Phi(\theta_*,\theta_\infty) + D_\Phi(\theta_\infty,\theta_0) = D_\Phi(\theta_*,\theta_0)$. Conclude on the resulting implicit bias for θ_∞.*

[4]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 $X\theta = y$.

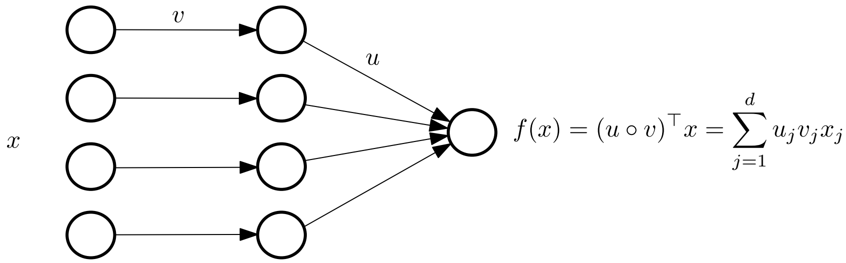
Exercise 12.4 (Implicit bias of mirror descent) *Consider the sequence* $(\tilde{\theta}_k)_{k \geqslant 0}$ *following the discrete version of equation (12.15); that is, mirror descent for all integer* k, $\Phi'(\tilde{\theta}_{k+1}) - \Phi'(\tilde{\theta}_k) = -\gamma F'(\tilde{\theta}_k)$, *for* $\gamma > 0$. *Show that, for* γ *sufficiently small, the same conclusion holds:*

- $\tilde{\theta}_k$ *converges toward an interpolator* $\tilde{\theta}_\infty$,
- $\tilde{\theta}_\infty$ *is the interpolator with minimum Bregman divergence* $D_\Phi(\theta, \theta_0)$.

Diagonal linear networks. Following Woodworth et al. (2020), we consider "diagonal linear networks," which are simple one-hidden-layer neural networks defining a prediction function of the following form (with a diagonal input weight matrix, no activation function, and no constant terms):

$$f(x) = (u \circ v)^\top x = \sum_{j=1}^{d} u_j v_j x_j,$$

for $u, v \in \mathbb{R}^d$, and with $u \circ v$ denoting the pointwise product. It corresponds to a special case of the network we defined in section 9.2:



This is thus an alternative nonlinear way of defining a linear model through $\theta = u \circ v \in \mathbb{R}^d$. 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 as 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 θ_j of θ:

$$\frac{d}{dt}\theta_j(t) = -F'(\theta(t))_j \sqrt{4\theta_j(t)^2 + \alpha^4}.$$

It can be exactly cast as a mirror flow, defined in equation (12.15), with mirror map $\Phi(\theta) = \sum_{j=1}^d q(\theta_j)$ for $q : \mathbb{R} \to \mathbb{R}$ a convex function such that $q''(\eta) = (4\eta^2 + \alpha^4)^{-1/2}$. By integrating twice, one obtains, by imposing $q'(0) = 0$,

$$q'(\eta) = \frac{1}{2} \operatorname{arg\,sinh}(2\eta/\alpha^2) = -\log \alpha + \frac{1}{2} \log \big[2\eta + \sqrt{4\eta^2 + \alpha^4}\big],$$

and then, imposing $q(0) = 0$, we get an even function of η defined as

$$\begin{aligned} q(\eta) &= \frac{\eta}{2} \operatorname{arg\,sinh}(2\eta/\alpha^2) + \frac{\alpha^2}{4}\big[1 - \sqrt{4\eta^2/\alpha^4 + 1}\big] \\ &= \frac{\eta}{2} \log \big[2\eta + \sqrt{4\eta^2 + \alpha^4}\big] - \frac{1}{4}\sqrt{4\eta^2 + \alpha^4} + \frac{\alpha^2}{4} - \frac{\eta}{2} \log \alpha^2. \end{aligned}$$

The point of computing the mirror potential q is to understand the implicit bias of the gradient flow. Indeed, being a mirror initialized at the origin, we know from the previous analysis that the limiting point is the Bregman projection of the origin, and hence controlled by the function q. It depends on the parameter α, which corresponds to the initialization. When α tends to $+\infty$, we get the asymptotic expansion $q(\eta) \sim \frac{\eta^2}{2\alpha^2}$ (proof left as an exercise), and thus the mirror map is asymptotically $\Phi(\theta) \sim \frac{1}{2\alpha^2}\|\theta\|_2^2$: the implicit bias is governed by the Euclidean norm, and we recover the traditional geometry of GD directly on θ.

However, when α tends to zero, we get the equivalent $q(\eta) \sim |\eta| \log \frac{1}{\alpha}$ (proof left as an exercise), and thus the mirror map is asymptotically $\Phi(\theta) \sim \log \frac{1}{\alpha} \cdot \|\theta\|_1$: the implicit bias corresponds to the ℓ_1-norm, showing how nonconvex optimization can lead to an a priori quite unexpected implicit bias. See more details by Woodworth et al. (2020), as well as an analysis of the extra regularizing effect of stochastic gradient descent by Pesme et al. (2021).[5] Note that the analysis previously given for diagonal networks explicitly shows quantitative global convergence of the gradient flow for a nonconvex objective; in section 12.3, we consider qualitative results that apply more generally.

Beyond linear networks. Characterizing the implicit bias of GD can be done in more complex situations. For example, Chizat and Bach (2020) show that with a neural network with rectified linear unit (ReLu) activations and infinitely many neurons estimated by GD 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 that are exactly the ones obtained in section 9.3.[6]

12.1.4 Remarks on Implicit Bias

Is the implicit bias always beneficial? In this subsection, we have seen that the parameterization of the predictor encodes a large part of the *implicit bias* of gradient-type methods. Linear predictors (and similarly kernel methods) benefit from ℓ_2-type of implicit

[5] See https://francisbach.com/implicit-bias-sgd/.

[6] See https://www.di.ens.fr/~fbach/ltfp/wide_implicit_bias.html for more details.

regularization, while, for example, multiplicative parameterizations, such as diagonal linear networks (with small initialization), lead to ℓ_1-type of implicit regularization. In each case, the combination of the initialization *and* the "architecture" of the predictor family constrain the capacity of the model implicitly, but an important question remains: Is this effect always beneficial for good generalization?

Relevance of prior knowledge. In this subsection, we have avoided any statistical questions on purpose (and we refer to chapters 7, 8 and 9 for a treatment of these) as the answers are a priori specific to the precise class of problem one considers. To be concrete, if the ground truth to a statistical model is expected to be sparse, then there is no reason that the low ℓ_2-norm implicit bias of linear methods should result in any substantial benefit: on the contrary explicit regularization by the ℓ_1-norm, as presented in section 8.3, or implicit regularization, as presented here, should still be used to constrain the estimator.

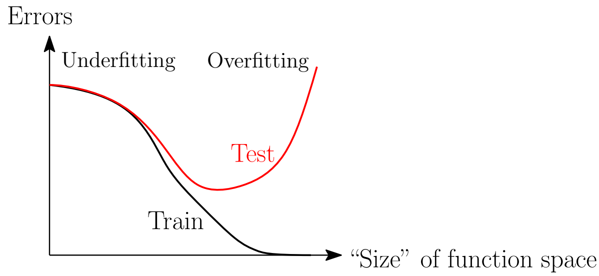
What does the implicit bias depend on? To a large extent, we can define the implicit bias as anything that happens to constrain the estimator *which has not been encoded a priori in the model*. Hence, any algorithmic hyperparameter or even any computation-oriented manipulation enters into this category: this is why the name *algorithmic regularization* has also been adopted to specify this phenomenon. To be concrete, we have seen that the parameterization of the predictor family and the initialization of a gradient method are of great importance. But, we can go beyond: for example the type of gradient method and any related parameter; the step size in general, the stochasticity of SGD, the momentum or renormalization in Adam, any manipulation like batch normalization or dropout, are all known to influence the prediction; see Smith et al. (2021), Lyu et al. (2022), and Andriushchenko et al. (2023) for more precise references, as well as Vardi (2023) for a recent review.

12.2 Double Descent

In this section, we consider an interesting phenomenon described in several works (Opper et al., 1990; Belkin et al., 2019; Mei and Montanari, 2022; Geiger et al., 2020; Hastie et al., 2022), which shows a particular behavior for overparameterized models learned by gradient descent (GD).

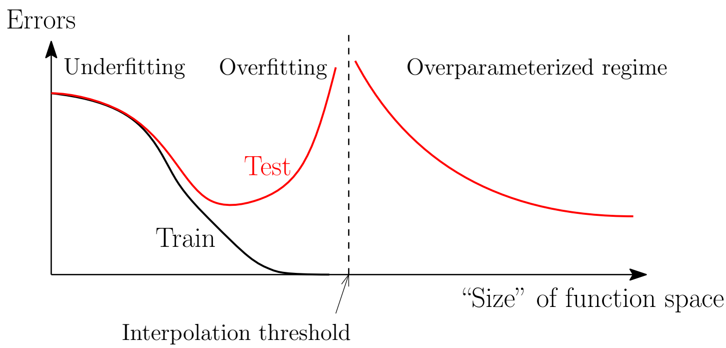
12.2.1 The Double Descent Phenomenon

As seen in chapters 2 and 4, typical learning curves look like this one:



Typically, the "capacity" or "size" of the space of functions 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 used in chapter 4, using Rademacher averages for the space of functions, is controlled by a uniform ℓ_2-norm of the parameters (say a positive number D) and 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 GD.

When the model is overparameterized (or in other words, the capacity gets large), a new phenomenon can occur: after the testing error explodes as the capacity grows, it goes down again, as illustrated below:



This section aims to understand this so-called "double descent" phenomenon, starting with empirical evidence.

12.2.2 Empirical Evidence

There may be no double descent phenomenon if other empirical risk minimizers are used instead of the one obtained by (stochastic) GD.

Toy example with random features. Here, we consider a random feature model as in chapters 7 and 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 outputs $y = \left(\frac{1}{4} + (v_*^\top x)^2\right)^{-1} + \mathcal{N}(0, \sigma^2)$, with $\sigma = 2$, for some random v_*.

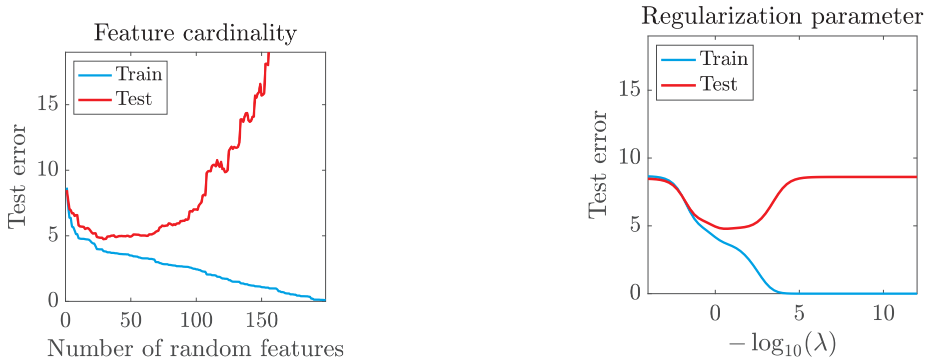


Figure 12.2. Classical learning curves: (left) training and testing errors as functions of the number of random features, always less than the number of observations; (right) training and testing errors for ridge regression with the same features (i.e., using ℓ_2-regularization).

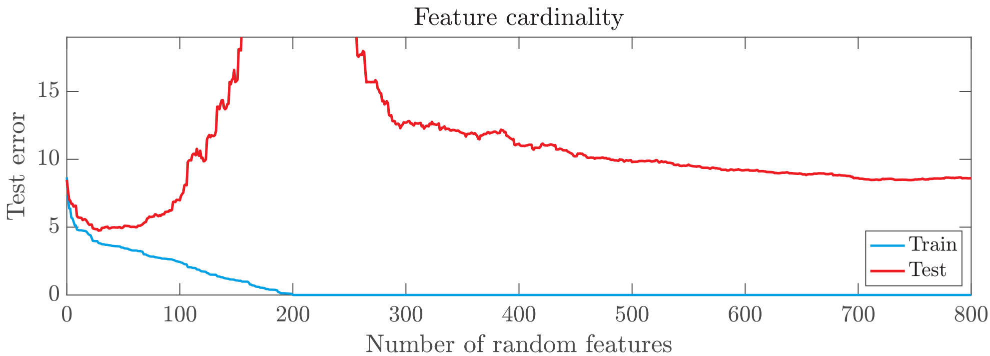


Figure 12.3. Double descent curve: training and testing errors as functions of the number of random features. For $m \leqslant n = 200$, this is exactly the same as the curves from figure 12.2 (left).

We sample m random features $v_1, \ldots, v_m \in \mathbb{R}^d$ uniformly at random on the sphere, and we learn parameters $\theta \in \mathbb{R}^m$ by minimizing the regularized empirical risk

$$\frac{1}{n}\sum_{i=1}^{n}\Big(y_i - \sum_{j=1}^{m}\theta_j (v_j^\top x_i)_+\Big)^2 + \lambda\|\theta\|_2^2. \tag{12.16}$$

We report in figure 12.2 training and testing errors after learning with GD until convergence: (left) varying m with $\lambda = 0$, (right) varying λ 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).

In the left plot of figure 12.2, the number of random features m is less than n as the testing error diverges (and there is no difference in testing performance compared to ridge regression in the right plot). However, when this number m is allowed to grow past n, we see the double descent phenomenon shown in figure 12.3. Similar experiments are shown by Belkin et al. (2019), Geiger et al. (2020), and Mei and Montanari (2022), in particular for neural networks.

No phenomenon when using regularization. When an extra regularizer is used (i.e., $\lambda \neq 0$ in equation (12.16)), then the double descent phenomenon is reduced. In particular, if the regularization parameter λ 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 Inputs

In this book, we will study two setups for the analysis of double descent. In this subsection, we consider Gaussian models with varying input dimensions, where we can show an explosion of the expected risk when dimension d is equal to the number n of observations, but the model is too simple to see a U-shaped curve for underparameterized models; moreover, the prediction problem changes as dimension grows, which is not standard. In section 12.2.4, we consider a model with fewer symmetries that lead to a proper full double descent behavior.

We now consider a d-dimensional Gaussian random vector with mean 0 and covariance matrix identity, with n observations $x_1, \dots, x_n \in \mathbb{R}^d$, and responses $y_i = x_i^\top \theta_* + \varepsilon_i$, with ε_i normal with mean zero and variance $\sigma^2 I$, for $i = 1, \dots, n$. We will compute an exact 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. 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. As shown in section 3.8, the excess risk is $\mathcal{R}(\hat\theta) = (\hat\theta - \theta_*)\Sigma(\hat\theta - \theta_*) = \|\hat\theta - \theta_*\|_2^2$ since $\Sigma = I$.

Underparameterized regime. In the underparameterized regime, the minimum norm empirical risk minimizer is simply the ordinary least-squares estimator, which is unbiased; that is, $\mathbb{E}\big[\hat\theta\big] = \theta_*$. We then have an expected excess risk equal to (see the random design analysis in proposition 3.10):

$$\mathbb{E}\big[\mathcal{R}(\hat\theta)\big] = \frac{\sigma^2}{n}\mathbb{E}\big[\operatorname{tr}(\Sigma\hat\Sigma^{-1})\big] = \sigma^2\mathbb{E}\big[\operatorname{tr}\big((X^\top X)^{-1}\big)\big].$$

The matrix $X^\top X \in \mathbb{R}^{d\times d}$ has a Wishart distribution with n degrees of freedom.[7] It is almost surely invertible if $n \geqslant d$, and is such that $\mathbb{E}\big[\operatorname{tr}\big((X^\top X)^{-1}\big)\big] = \frac{d}{n-d-1}$ if $n \geqslant d+2$. The expectation is infinite for $n = d$ and $n = d+1$ (see, e.g., Haff, 1979, for computations of moments of the Wishart distribution).

Therefore, we have for $n \geqslant d+2$ an expected excess risk equal to

$$\mathbb{E}\big[\mathcal{R}(\hat\theta)\big] = \sigma^2\frac{d}{n-d-1}. \tag{12.17}$$

[7] See https://en.wikipedia.org/wiki/Wishart_distribution for details.

Overparameterized regime. In the overparameterized regime, when $n \leqslant d$, the kernel matrix is almost surely invertible, and the minimum ℓ_2-norm interpolator $\hat\theta$ is equal to (using the formulas in section 12.1.1) $\hat\theta = X^\top(XX^\top)^{-1}y = X^\top(XX^\top)^{-1}X\theta_* + X^\top(XX^\top)^{-1}\varepsilon$. The expected excess risk decomposes into a bias and a variance term.

The *variance* term is equal to, since $\Sigma = I$,

$$\begin{aligned}\mathbb{E}\big[\varepsilon^\top(XX^\top)^{-1}X\Sigma X^\top(XX^\top)^{-1}\varepsilon\big] &= \sigma^2\mathbb{E}\big[\operatorname{tr}\big((XX^\top)^{-1}XX^\top(XX^\top)^{-1}\big)\big]\\ &= \sigma^2\mathbb{E}\big[\operatorname{tr}\big((XX^\top)^{-1}\big)\big],\end{aligned}$$

which is the same expectation of the trace of an inverse Wishart matrix, but with the order of n and d reversed; that is, $\sigma^2\frac{n}{d-n-1}$ for $d \geqslant n+2$.

The *bias term* is equal to

$$\mathbb{E}\big[\mathcal{R}(X^\top(XX^\top)^{-1}X\theta_*)\big] = \mathbb{E}\Big[\big\|\Sigma^{1/2}\big(X^\top(XX^\top)^{-1}X\theta_* - \theta_*\big)\big\|_2^2\Big].$$

Since $\Sigma = I$, we get a bias term equal to

$$\mathbb{E}\Big[\theta_*^\top\big(I - X^\top(XX^\top)^{-1}X\big)\theta_*\Big].$$

The matrix $X^\top(XX^\top)^{-1}X \in \mathbb{R}^{d\times d}$ is the projection matrix on a random subspace of size $n \leqslant d$. By rotational invariance of the Gaussian distribution, this random subspace is uniformly distributed among all subspaces, and therefore we can replace θ_* by $\|\theta_*\|_2 \cdot e_j$, that is,

$$\mathbb{E}\big[\theta_*^\top X^\top(XX^\top)^{-1}X\theta_*\big] = \|\theta_*\|_2^2 \cdot \mathbb{E}\big[e_j^\top X^\top(XX^\top)^{-1}Xe_j\big]$$

for any of the d canonical basis vectors e_j, $j = 1,\dots,d$, and thus

$$\begin{aligned}\mathbb{E}\big[\theta_*^\top X^\top(XX^\top)^{-1}X\theta_*\big] &= \frac{\|\theta_*\|_2^2}{d}\sum_{j=1}^d \mathbb{E}\big[e_j^\top X^\top(XX^\top)^{-1}Xe_j\big]\\ &= \frac{\|\theta_*\|_2^2}{d}\mathbb{E}\big[\operatorname{tr}\big(X^\top(XX^\top)^{-1}X\big)\big] = \frac{\|\theta_*\|_2^2}{d}\cdot n.\end{aligned}$$

The bias term is thus equal to $\frac{d-n}{d}\|\theta_*\|_2^2$.

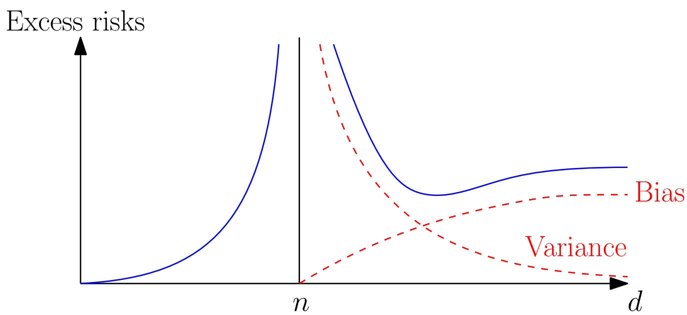
Therefore, the overall expected risk in the overparameterized regime is

$$\frac{\sigma^2 n}{d-n-1} + \|\theta_*\|_2^2\frac{d-n}{d}. \tag{12.18}$$

Summary. We can now combine equation (12.17) and equation (12.18) to get:

$$\begin{aligned}&\text{if } d \leqslant n-2, &\mathbb{E}\big[\mathcal{R}(\hat\theta)\big] &= \sigma^2\frac{d}{n-d-1}\\ &\text{if } d \geqslant n+2, &\mathbb{E}\big[\mathcal{R}(\hat\theta)\big] &= \frac{\sigma^2 n}{d-n-1} + \|\theta_*\|_2^2\frac{d-n}{d}.\end{aligned}$$

This leads to the following picture, where (1) there is indeed an explosion around $d = n$, but (2) there is no U-shaped curve for $d < n$ while there is one for $d > n$ (see exercise 12.5):



This extends to more general sampling models, see Hastie et al. (2022), and to random non-linear features (Mei and Montanari, 2022).

Exercise 12.5 *Show that for $d > n$, the generalization risk has a U-shaped curve and find its minimizer.*

12.2.4 Linear Regression with Gaussian Projections (♦♦)

To provide a more precise theoretical justification for the double descent phenomenon, we consider a linear regression model in the random design setting, with Gaussian inputs and Gaussian noise. We assume $d > n$ as in section 10.2.2 (so that a perfect fit is possible).

That is, we consider a Gaussian random variable with mean 0 and general covariance matrix Σ, with n observations $x_1, \dots, x_n$, and responses $y_i = x_i^\top \theta_* + \varepsilon_i$, with ε_i Gaussian with mean zero and covariance matrix $\sigma^2 I$, for $i = 1, \dots, n$.

To have a unique prediction problem with a varying number of features, we consider additional random projections; that is, as in section 10.2.2, 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 (1) we will perform an analysis in the random design setting, and (2) we will also need to tackle the overparameterized regime $m > n$.

We will compute the expectation of the expected risk of the minimum norm empirical risk minimizer (as detailed in section 12.1.1), which is the one GD converges to. See Bach (2024) for further more precise asymptotic results using random matrix theory.

We denote by $X \in \mathbb{R}^{n \times d}$ the design matrix, by $\hat{\Sigma} = \frac{1}{n} X^\top X$ the noncentered covariance matrix, and by $K = XX^\top \in \mathbb{R}^{n \times n}$ the invertible kernel matrix. We will need to compute expectations with respect to the data (X, ε) *and* the random projection matrix S (we thus make explicit which variables are considered in expectations). The estimator $\hat{\theta}$ is equal to $S\hat{\eta}$, with $\hat{\eta} \in \mathbb{R}^m$ as a minimizer of $\|y - XS\eta\|_2^2$.

The excess risk is denoted as $\mathcal{R}(\hat{\theta}) = (\hat{\theta} - \theta_*)\Sigma(\hat{\theta} - \theta_*)$, and we now consider the two regimes $m < n$ (underparameterized) and $m > n$ (overparameterized). In both cases, as already seen in chapter 3 and in section 12.2.3, 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 underparameterized regime where $n > m$, the minimum norm empirical risk minimizer is simply the ordinary least-squares (OLS) 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 $\Pi_S^\top \Sigma \Pi_S = \Sigma \Pi_S$.

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 it can be expanded using θ_* as follows:

$$\begin{aligned}
\hat\theta \ &= \ S\hat\eta \ = \ S(S^\top X^\top X S)^{-1} S^\top X^\top y \\
&= \ S(S^\top X^\top X S)^{-1} S^\top X^\top X \theta_* + S(S^\top X^\top X S)^{-1} S^\top X^\top \varepsilon \ \text{ using } y = X\theta_* + \varepsilon, \\
&= \ N\theta_* + S(S^\top X^\top X S)^{-1} S^\top X^\top \varepsilon,
\end{aligned}$$

with $N = S(S^\top X^\top X S)^{-1} S^\top X^\top X$. Conditioned on S and X, the expected excess risk is equal to

$$\begin{aligned}
\mathbb{E}_\varepsilon\big[\mathcal{R}(\hat\theta)\big] \ &= \ \sigma^2 \operatorname{tr}\big(XS(S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top\big) + \big\|\Sigma^{1/2}\big(N\theta_* - \theta_*\big)\big\|_2^2 \\
&= \ \sigma^2 \operatorname{tr}\big(S^\top \Sigma S (S^\top X^\top X S)^{-1}\big) + \operatorname{tr}\big(\big(N\theta_* - \theta_*\big)^\top \Sigma \big(N\theta_* - \theta_*\big)\big).
\end{aligned}$$

For the variance term, equal to $\sigma^2 \operatorname{tr}\big(S^\top \Sigma S (S^\top X^\top X S)^{-1}\big)$, and for S fixed, since X has a Gaussian distribution, the matrix $S^\top X^\top X S$ 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

$$\mathbb{E}_X\big[(S^\top X^\top X S)^{-1}\big] = \frac{1}{n-m-1}(S^\top \Sigma S)^{-1},$$

which in turn implies $\mathbb{E}_{S,X,\varepsilon}\big[\mathcal{R}^{(\mathrm{var})}(\hat\theta)\big] = \mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{var})}(\hat\theta)\big] = \dfrac{\sigma^2 m}{n-m-1}$, independent of the choice of the sketching matrix S.

For the bias term, the computation is more involved. We expand

$$\begin{aligned}
\mathbb{E}_\varepsilon\big[\mathcal{R}^{(\mathrm{bias})}(\hat\theta)\big] \ &= \ \operatorname{tr}\big(\big(N\theta_* - \theta_*\big)^\top \Sigma \big(N\theta_* - \theta_*\big)\big) \\
&= \ \theta_*^\top \Sigma \theta_* + 2\theta_*^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \theta_* \\
&\qquad + \theta_*^\top X^\top X S (S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \theta_*.
\end{aligned}$$

To compute the expectation, we will first condition on (XS, S) 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):

$$\begin{aligned}
\mathbb{E}\big[X \big| XS, S\big] \ &= \ XS(S^\top \Sigma S)^{-1} S^\top \Sigma = X\Pi_S \\
\mathbb{E}\big[\operatorname{tr}(AX^\top B X) \big| XS, S\big] \ &= \ \operatorname{tr}\big(A\Pi_S^\top X^\top B X \Pi_S\big) + \operatorname{tr}(B)\operatorname{tr}(A\Sigma(I - \Pi_S)).
\end{aligned}$$

This leads to, with S assumed to be fixed, using the two identities just above:

$$\begin{aligned}
\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{bias})}(\hat\theta)\big] \ &= \ \theta_*^\top \Sigma \theta_* + \mathbb{E}_X\big[2\theta_*^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \Pi_S \theta_*\big] \\
&+ \mathbb{E}_X\big[\theta_*^\top \Pi_S^\top X^\top X S (S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \Pi_S \theta_*\big] \\
&+ \mathbb{E}_X\big[\operatorname{tr}\big(XS(S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top\big) \cdot \operatorname{tr}\big(\theta_*\theta_*^\top \Sigma(I - \Pi_S)\big)\big].
\end{aligned}$$

Now, using properties of Π_S, we get

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma\theta_* + \mathbb{E}_X\big[2\theta_*^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\theta_*\big] \\ &+\mathbb{E}_X\big[\theta_*^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\theta_*\big] \\ &+\mathbb{E}_X\big[\operatorname{tr}\big(S^\top\Sigma S(S^\top X^\top XS)^{-1}\big)\cdot\operatorname{tr}\big(\theta_*\theta_*^\top\Sigma(I-S(S^\top\Sigma S)^{-1}S^\top\Sigma)\big)\big].\end{aligned}$$

We can now group some terms and use $\Pi_S^\top\Sigma\Pi_S = \Sigma\Pi_S$ to get

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\Big[1+\mathbb{E}_X\big[\operatorname{tr}\big(S^\top\Sigma S(S^\top X^\top XS)^{-1}\big)\big]\Big] \\ &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\Big(1+\operatorname{tr}\Big(\frac{1}{n-m-1}(S^\top\Sigma S)^{-1}S^\top\Sigma S\Big)\Big),\end{aligned}$$

using expectations of Wishart random variables. Overall, we get

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\frac{n-1}{n-m-1} \\ &= \theta_*^\top\big(\Sigma-\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\big)\theta_*\cdot\frac{n-1}{n-m-1}.\end{aligned}$$

We can further bound the bias term; we have, for S Gaussian, following the same reasoning as in section 10.2.2,

$$\begin{aligned}&\mathbb{E}_S\big[\theta_*^\top(I-\Pi_S)^\top\Sigma(I-\Pi_S)\theta_*\big] \\ =\ &\mathbb{E}_S\big[\min_{\eta\in\mathbb{R}^m}(\theta_*-S\eta)^\top\Sigma(\theta_*-S\eta)\big] \text{ by definition of } \Pi_S, \\ \leqslant\ &\mathbb{E}_S\big[\min_{\xi\in\mathbb{R}^d}(\theta_*-SS^\top\xi)^\top\Sigma(\theta_*-SS^\top\xi)\big], \text{ with } \eta \text{ constrained in the column space of } S^\top, \\ \leqslant\ &\min_{\xi\in\mathbb{R}^d}\mathbb{E}_S\big[(\theta_*-SS^\top\xi)^\top\Sigma(\theta_*-SS^\top\xi)\big] \text{ by swapping the minimum and expectation,} \\ =\ &\min_{\xi\in\mathbb{R}^d}\Big(\theta_*^\top\Sigma\theta_*-2\xi^\top\mathbb{E}\big[SS^\top\big]\Sigma\theta_*+\xi^\top\mathbb{E}\big[SS^\top\Sigma SS^\top\big]\xi\Big) \text{ by developing,} \\ =\ &\theta_*^\top\Big(\Sigma-\Sigma\mathbb{E}\big[SS^\top\big]\big(\mathbb{E}\big[SS^\top\Sigma SS^\top\big]\big)^{-1}\mathbb{E}\big[SS^\top\big]\Sigma\Big)\theta_* \text{ by minimizing in closed form.}\end{aligned}$$

Using Wishart expectations, we then get:

$$\begin{aligned}\mathbb{E}_S\big[\theta_*^\top(I-\Pi_S)^\top\Sigma(I-\Pi_S)\theta_*\big] &\leqslant \theta_*^\top\Big(\Sigma-m\Sigma\big((m+1)\Sigma+\operatorname{tr}(\Sigma)I\big)^{-1}\Sigma\Big)\theta_* \\ &= \theta_*^\top(\Sigma+\operatorname{tr}(\Sigma)I)\big((m+1)\Sigma+\operatorname{tr}(\Sigma)I\big)^{-1}\Sigma\theta_* \\ &\leqslant \frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\theta_*^\top\Big(\Sigma+\frac{\operatorname{tr}(\Sigma)}{m+1}I\Big)^{-1}\Sigma\theta_* \\ &\leqslant \frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\|\theta_*\|_2^2, \text{ using } \Sigma\preccurlyeq\operatorname{tr}[\Sigma]\cdot I.\end{aligned}$$

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 has a similar excess risk bound as for ridge regression from section 3.6 and 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 that explodes when m approaches n. See Bach (2024) for sharper results where expectations with respect to S are taken using random matrix theory.

Overparameterized regime. In the overparameterized regime, when $m \geqslant n$, then the kernel matrix $XSS^\top X^\top$ is almost surely invertible, and the minimum ℓ_2-norm interpolator $\hat\theta$ is equal to (using the expression of this interpolator based on the kernel matrix 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 ε of the excess risk $\mathcal{R}(\hat\theta)$ as follows:

$$\begin{aligned}\mathbb{E}_\varepsilon\big[\mathcal{R}(\hat\theta)\big] &= \sigma^2 \operatorname{tr}\big((XSS^\top X^\top)^{-1} XSS^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1}\big) \\ &\quad + \big\|\Sigma^{1/2}\big(SS^\top X^\top (XSS^\top X^\top)^{-1}X - I\big)\theta_*\|_2^2.\end{aligned}$$

We can now use the same reasoning as in the underparameterized regime, but now taking expectations with respect to S (with X fixed). We have, for any symmetric matrices A and B of compatible sizes (proof left as an exercise),

$$\begin{aligned}\mathbb{E}\big[\operatorname{tr}(ASBS^\top)\big|XS,X\big] &= \operatorname{tr}\big(S^\top X^\top (XX^\top)^{-1} XAX^\top (XX^\top)^{-1}XSB\big) \\ &\quad + \operatorname{tr}(B)\operatorname{tr}\big[A(I - X^\top (XX^\top)^{-1}X)\big] \\ \mathbb{E}\big[S\big|XS,X\big] &= X^\top (XX^\top)^{-1}XS.\end{aligned}$$

Therefore, for the variance term proportional to σ^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 two identities just 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, marginalizing with respect to S and using the expectation of the inverse Wishart distribution, the variance term is

$$\begin{aligned}\mathbb{E}_{\varepsilon,S}[\mathcal{R}^{(\mathrm{var})}(\hat\theta)] &= \sigma^2 \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big) \\ &\quad + \sigma^2 \frac{\operatorname{tr}\big((XX^\top)^{-1}\big)}{m-n-1}\cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big].\end{aligned}$$

For the bias term, we have

$$\begin{aligned}
& \big\|\Sigma^{1/2}\big(SS^\top X^\top (XSS^\top X^\top)^{-1}X - I\big)\theta_*\big\|_2^2 \\
= \;& \theta_*^\top \Sigma\theta_* + \theta_*^\top X^\top (XSS^\top X^\top)^{-1} XSS^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\
& \qquad - 2\theta_*^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\
= \;& \theta_*^\top \Sigma\theta_* + \operatorname{tr}(ASBS^\top) - 2\theta_*^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*,
\end{aligned}$$

with $A = \Sigma$ and $B = S^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XS$. Taking conditional expectations given (XS, X), simplifying the product $XSS^\top X^\top (XSS^\top X^\top)^{-1} = I$, and using $XSBS^\top X^\top = X\theta_*\theta_*^\top X^\top$, we obtain the following expression of the expected bias given (XS, X):

$$\begin{aligned}
& \theta_*^\top \Sigma\theta_* \\
& + \operatorname{tr}\big(X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XSS^\top\big) \\
& + \operatorname{tr}(S^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XS) \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big] \\
& - 2\theta_*^\top \Sigma X^\top (XX^\top)^{-1} XSS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\
= \;& \theta_*^\top \Sigma\theta_* \\
& + \theta_*^\top X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} X\theta_* \\
& + \operatorname{tr}((XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top) \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big] \\
& - 2\theta_*^\top \Sigma X^\top (XX^\top)^{-1} X\theta_* \text{ by simplifying,} \\
= \;& \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma(I - X^\top (XX^\top)^{-1} X)\theta_* \\
& + \operatorname{tr}((XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top) \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big],
\end{aligned}$$

by grouping terms. Marginalizing the variable S, this leads to, by rearranging terms,

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}^{(\text{bias})}(\hat\theta)] &= \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma(I - X^\top (XX^\top)^{-1} X)\theta_* \\
& \quad + \frac{1}{m-n-1}\theta_*^\top X^\top (XX^\top)^{-1} X\theta_* \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big].
\end{aligned}$$

Pulling together bias and variance, when m tends to infinity, we get the following performance:

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}_\infty(\hat\theta)] &= \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma(I - X^\top (XX^\top)^{-1} X)\theta_* \\
& \quad + \sigma^2 \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big).
\end{aligned}$$

Overall, we get

$$\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}(\hat\theta)] &= \mathbb{E}_{\varepsilon,S}[\mathcal{R}_\infty(\hat\theta)] + \frac{1}{m-n-1}\operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big] \\
& \quad \cdot \big(\theta_*^\top X^\top (XX^\top)^{-1} X\theta_* + \sigma^2 \operatorname{tr}((XX^\top)^{-1})\big).
\end{aligned}$$

Thus, as a function of m, we get a descent curve on the right side of $m = n$. See Bach (2024) for a detailed expression obtained after taking the expectation with respect to X

using random matrix theory. While the limiting bias term typically has a better value than for the underparameterized 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 the illustration in figure 12.4.

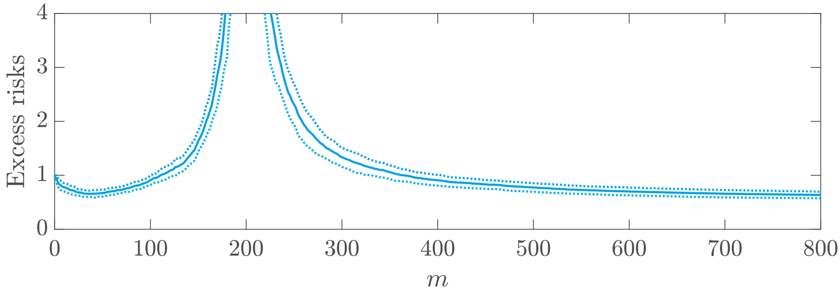


Figure 12.4. Example of a double descent curve, for linear regression with random projections with $n = 200$ observations, in dimension $d = 400$ and a nonisotropic covariance matrix. The data are normalized so that predicting zero leads to an excess risk of 1 and the noise so that the optimal expected risk is 1/4. The empirical estimate is obtained by sampling 20 datasets and 20 random projections from the same distribution and averaging the corresponding excess risks

12.3 Global Convergence of Gradient Descent

In section 9.2.1, we alluded to the property of GD for overparameterized neural networks, which converges to a global minimum of the objective function despite being nonconvex. We present more formal arguments in this section, with a general result without proof, as well as a detailed proof for linear neural networks.

12.3.1 Mean Field Limits

In this section, we present results from Chizat and Bach (2018), closely following the exposition from Bach and Chizat (2022).[8] More precisely, we consider neural networks with a hidden layer of size m with m tending to infinity, and we first rescale the prediction function by $1/m$ (which can be obtained by rescaling all output weights by $1/m$), expressing it explicitly as an empirical average as

$$h(x) = \frac{1}{m} \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j),$$

where $\eta_j \in \mathbb{R}$ is the output weight associated to the jth neuron, and $(w_j, b_j) \in \mathbb{R}^d \times \mathbb{R}$ the corresponding vector of input weights. The key observation is that the prediction

[8] See also https://www.di.ens.fr/~fbach/ltfp/wide_convergence.html.

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 parameters* (which is not true if extra layers of hidden neurons are added).

To highlight this parameter separability, we define $v_j = \big[\eta_j, w_j^\top, b_j\big]^\top \in \mathbb{R}^{d+2}$ as the set of weights associated with 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

$$h = \frac{1}{m} \sum_{j=1}^{m} \Psi(v_j). \tag{12.19}$$

The expected risk takes the form

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

which is convex in h for convex loss functions (which is the case throughout this book, even for neural networks, such as the logistic or square loss), but typically nonconvex 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, such as for sparse deconvolution (Chizat, 2022).

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 equation (12.19) 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 the empirical measure associated with $(v_1, \dots, v_m)$ (i.e., an average of Dirac measures at each $v_1, \dots, v_m \in \mathcal{V}$). Following a physics analogy, we will refer to each v_j as a *particle*. When the number m of particles grows, by the law of large number (see exercise 12.6), the empirical measure $\frac{1}{m} \sum_{j=1}^{m} \delta_{v_j}$ 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.

Exercise 12.6 *Consider n i.i.d. random variables $x_1, \dots, x_n$ in $\mathbb{R}^d$ with distribution μ and $\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$, the associated random empirical measure. Show that the strong law of large number for $(x_i)_{i \geqslant 1}$ implies the weak convergence of μ_n toward μ.*[9]

The optimization problem that we are facing is equivalent to

$$\inf_{\mu \in \mathcal{P}(\mathcal{V})} \mathcal{R}\Big(\int_{\mathcal{V}} \Psi(v) d\mu(v)\Big), \tag{12.20}$$

[9]We say that μ_n weakly converges toward μ if for all f continuous and bounded, $\int f d\mu_n \to \int f d\mu$.

with the constraint that μ is an average of m Dirac measures. We now follow a long line of work in statistics and signal processing (see Barron, 1993; Kurková and Sanguineti, 2001), consider the optimization problem *relaxing 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 $\mathcal{P}(\mathcal{V})$, as we already did in section 9.3.2.

Note that we now have a convex optimization problem with a convex objective in μ 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 GD on $(v_1, dots, v_m)$, corresponding to standard practice in neural networks (e.g., backpropagation). 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 GD recursion on $V = (v_1, \dots, v_m) \in \mathcal{V}^m$, defined as

$$V_k = V_{k-1} - \gamma m G'(V_{k-1}), \tag{12.21}$$

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 backpropagation algorithm. We include factor m in the step size to obtain a well-defined limit when m tends to infinity (as discussed next).

For convenience in the analysis, we look at the limit when the step size γ goes to zero. If we consider 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 ODE (Suli and Mayers, 2003):

$$\dot{W} = -mG'(W). \tag{12.22}$$

This gradient flow will be our main focus in this discussion. As already highlighted, and with extra regularity assumptions, it is the limit of the gradient recursion in equation (12.21) for vanishing step sizes γ. Moreover, under appropriate conditions, stochastic gradient descent (SGD) 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 on risks that are expectations over whole probability distributions over data (x, y), for which single-pass SGD corresponds, in this limit, to the gradient flow on the *testing* error.

Wasserstein gradient flow. Previously, we have described a general framework where we want to minimize a function F defined on probability measures:

$$F(\mu) = \mathcal{R}\Big(\int_{\mathcal{V}} \Psi(v) d\mu(v)\Big), \tag{12.23}$$

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 (when the step size goes to zero) 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 follows (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 μ and ν. In a nutshell, the gradient flow is formally defined as the limit when γ tends to zero of the extension of the following discrete-time dynamics:

$$\mu(t+\gamma) = \arg\min_{\nu \in \mathcal{P}(\mathcal{V})} F(\nu) + \frac{1}{2\gamma} W_2(\mu(t),\nu)^2.$$

When applying such a definition in a Euclidean space with the Euclidean metric, we recover the usual gradient flow $\dot\mu = -F'(\mu)$, but with the Wasserstein metric, this defines a specific flow on the set of measures. When the initial measure is a weighted sum of Dirac measures, this is precisely asymptotically (i.e., when $\gamma \to 0$) equivalent to backpropagation (in other words, the Wasserstein gradient flow limit also applies to finitely many neurons, and we will need the extension to measures with densities only for the global convergence result in proposition 12.2). When initialized with an arbitrary probability measure, $(\mu_t)_{t \geqslant 0}$ is the solution of a partial differential equation (PDE) in the weak sense (i.e., in the sense of distributions). Moreover, when the sum of Dirac measures converges in distribution to some measure, the flow converges to the solution of the PDE. More precisely, assuming that $\Psi : \mathbb{R}^{d+2} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space (in our neural network example, $\mathcal{H}$ is the space of square-integrable functions on $\mathbb{R}^d$), and $\mathcal{R}'(h) \in \mathcal{H}$ is the gradient of $\mathcal{R}$, we consider the *mean potential*

$$J(v|\mu) = \Big\langle \Psi(v), \mathcal{R}'\Big(\int_{\mathcal{V}} \Psi(w) d\mu(w)\Big)\Big\rangle. \tag{12.24}$$

With these notations, note that the gradient flow equation (12.22) written on each neuron, also called here "particle," becomes

$$\dot v_j = -J'(v_j|\mu),$$

where μ is the time-dependent aggregation of all particles, and the derivative J' stands for the gradient of the function $v \mapsto J(v|\mu)$.

At the level of the measure μ that describes the aggregated movement of all particles, this equation becomes a PDE, also called "continuity equation" in physics, which writes (see, e.g., Evans, 2022):

$$\partial_t \mu_t(v) = \mathrm{div}(\mu_t(v) J'(v|\mu_t)), \tag{12.25}$$

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).

Proposition 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 R is Lipschitz-continuous 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 $\mu_{t,m} = \frac{1}{m}\sum_{j=1}^{m} \delta_{v_j(t)}$, where $(v_1(t), \ldots, v_m(t))$ solves the ODE (12.22). If $\mu_{0,m}$ weakly converges to some $\mu_0 \in \mathcal{P}(\mathcal{V})$, then $\mu_{t,m}$ weakly converges to μ_t, where $(\mu_t)_{t\geq 0}$ is the unique weakly continuous solution to equation (12.25) initialized with μ_0.*

Next, we will study the solution of this PDE (i.e., the Wasserstein gradient flow), interpreting it as the limit of the gradient flow in equation (12.22) when the number of particles m tends to infinity.

Global convergence. We consider the Wasserstein gradient flow defined here, which leads to the PDE in equation (12.25). We aim to understand under what circumstances we can expect that when $t \to \infty$, μ_t converges to a global minimum of F defined in equation (12.23). 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.

Proposition 12.2 (Informal) *If the support of the initial distribution includes all directions in $\mathbb{R}^{d+2}$, and if function Ψ 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.*

Proposition 12.2 applies when initializing the gradient flow with a distribution with full support, for example, a distribution with strictly positive density everywhere. Chizat and Bach (2018) present another version of this result that allows *partial* homogeneity (e.g., with respect to a subset of variables) of degree 1, at the cost of a more technical assumption on the initialization. For neural networks, we have $\Psi(\eta, w, b)(x) = \eta\sigma(w^\top x + b)$, 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 about 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., in 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 nonlinear infinite width limits for deeper networks is an important research area (Nguyen and Pham, 2023; Araújo et al., 2019; Fang et al., 2021; Hanin and Nica, 2019; Sirignano and Spiliopoulos, 2022; 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 now 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 Θ taking the form $\Theta = UV^\top \in \mathbb{R}^{k\times d}$. We aim to minimize $G(UV^\top)$, where $G : \mathbb{R}^{k\times d} \to \mathbb{R}$ is a smooth convex risk function.

It can be rewritten as function G applied to a linear projection of matrix $\binom{U}{V}\binom{U}{V}^\top = \begin{pmatrix} UU^\top & UV^\top \\ VU^\top & VV^\top \end{pmatrix}$, which takes 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 thus now to minimize a convex function G over positive-semidefinite (PSD) matrices, using plain GD techniques on a nonlinear parameterization of such matrices. This is done to illustrate optimization for neural networks, noting that faster algorithms based on projected GD presented in chapter 5 could also be used. We already studied a special case in section 12.1.3, where the ODE 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), with gradients that are 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 symmetric matrices 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 to simplify later formulas. Since F is twice differentiable, this

ODE is defined for all $t \geqslant 0$. To compute the gradient of F, we perform an asymptotic expansion as follows:

$$\begin{aligned}
F(W+\Delta) &= G\big(WW^\top + \Delta W^\top + W\Delta^\top + o(\|\Delta\|_2)\big)\\
&= F(W) + \operatorname{tr}\big[G'\big(WW^\top\big)\big(\Delta W^\top + W\Delta^\top\big)\big] + o(\|\Delta\|_2)\\
&= F(W) + 2\operatorname{tr}\big[\Delta^\top G'\big(WW^\top\big)W\big] + o(\|\Delta\|_2), \text{ using symmetry of } G',
\end{aligned}$$

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 column $w_j \in \mathbb{R}^d$ of W, called a "particle":

$$\dot{w}_j = -G'\big(WW^\top\big)w_j,$$

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 (which implies $m \geqslant d$), 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 overparameterized situation, and the columns of W are initialized randomly (e.g., from a standard Gaussian random vector), then $WW^\top$ indeed has full rank.

Exercise 12.7 (♦) *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 conditions: (1) $\operatorname{tr}[MG'(M)] = 0$ and (2) $G'(M) \succcurlyeq 0$, as we now show. Note that once (2) is satisfied, (1) is equivalent to $MG'(M) = 0$.[10]

- *Necessary conditions (no need for convexity).* If M is optimal, then for all Δ such that $M + \Delta \succcurlyeq 0$, $G(M+\Delta) - G(M) \geqslant 0$. When Δ is small, this leads to $\operatorname{tr}[\Delta G'(M)] \geqslant 0$.

  Taking Δ small along $-M$ or M, we get $\operatorname{tr}[MG'(M)] = 0$ as a necessary condition.

  Taking $\Delta = uu^\top$ for all $u \in \mathbb{R}^d$, we get $G'(M) \succcurlyeq 0$ as a necessary condition.

[10] For two PSD matrices A and B of the same sizes, $AB = 0 \Leftrightarrow \operatorname{tr}(AB) = 0$.

- *Sufficient conditions (convexity is needed).* If the conditions are met, then for any matrix $N \succcurlyeq 0$, we get from the subgradient inequality for the convex function G:

$$G(N) \geqslant G(M) + \operatorname{tr}\big[G'(M)(N-M)\big].$$

  Using condition (1), we get $\operatorname{tr}\big[G'(M)M\big] = 0$, while condition (2) ensures that $\operatorname{tr}\big[G'(M)N\big] \geqslant 0$. Thus, $G(N) \geqslant G(M)$, and therefore 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 (note that this implies $m \geqslant d$) and converges to some M_∞,[11] we now show that it satisfies the two optimality conditions described 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_∞ (see the following examples).

Condition (1) is a direct consequence of $-G'(M_\infty)M_\infty - M_\infty G'(M_\infty) = 0$ (and then taking the trace), which is satisfied at convergence (this is the stationary condition, stating that all particles stop). The difficult part is to show condition (2), which can be interpreted as ensuring that no other potential particles could enter and increase 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_∞ (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 the Cauchy-Schwarz inequality and the impossibility of having $A_\infty u = -\eta u$ (which is the equality condition for the Cauchy-Schwarz inequality). We denote by $\beta > \eta$ the minimal value of such $\|A_\infty u\|_2$ (for all u that satisfy $\|u\|_2 = 1$ and $u^\top A_\infty u = -\eta$).

The idea is to show that sufficiently close to convergence, once a particle has a direction in the set

$$K = \big\{u \in \mathbb{R}^d,\ \|u\|_2 = 1,\ u^\top A_\infty u < -\eta\big\},$$

its direction never gets out of K, and it leads to a contradiction (set K is not empty because $\lambda_{\min}(A_\infty) < -\eta$).

Next, we introduce the time dependence explicitly.

Choice of particle close to convergence (♦♦). We have $M(t) \to M_\infty$. Thus there is t_0 such that $\|A(t) - A_\infty\|_{\mathrm{op}} \leqslant \varepsilon$, for all $t \geqslant t_0$, with ε 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 a full rank equal to d, then there is $\alpha_0 \in \mathbb{R}^m$ such that $y_0 = W(t_0)\alpha_0$.

[11] It does under basic assumptions on G, such as piecewise analyticity, see Bolte et al. (2006).

We then consider 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 $\|A(t_1) - A_\infty\|_{\mathrm{op}} \leqslant \varepsilon$ and $z(t_1)^\top A_\infty z(t_1) = -\eta z(t_1)^\top z(t_1)$, then leads to

$$\begin{aligned}
\frac{d}{dt}\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)}\Big|_{t=t_1} &\leqslant -2\frac{z_1^\top A_\infty^2 z_1}{z_1^\top z_1} + 2\frac{\|A_\infty z_1\|_2 \varepsilon}{\|z_1\|_2} + 2\eta^2 + 2\eta\varepsilon \\
&\leqslant -2\beta^2 + 2\eta^2 + 2\|A_\infty\|_{\mathrm{op}}\varepsilon + 2\eta\varepsilon,
\end{aligned}$$

which is strictly negative for ε small enough, which is a contradiction because that 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, $z(t) \in \mathbb{R}_+ K$.

We can now obtain our final contradiction. We now have that particule $z(t)$ is in $\mathbb{R}_+ K$ for all $t \geqslant t_0$. We then have, for all $t \geqslant t_0$,

$$\frac{d}{dt} z(t)^\top z(t) = -2 z(t)^\top A(t) z(t) \geqslant 2\big(- z(t)^\top A_\infty z(t) - \|z(t)\|_2^2 \varepsilon \big) \geqslant 2(\eta - \varepsilon)\|z\|_2^2,$$

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 $z(t) = W(t)\alpha_0$.

Alternative global convergence proof. (♦) Under mild regularity conditions on the objective, we know from Lee et al. (2016) that gradient flow can only converge to local minimizers (and never saddle points). Exercise 12.8 shows that in our case (where F is a multiplicative reparameterization of a convex function G), then local minimizers of F are, in fact, global. This implies that the flow can only converge to global minimizers of G.

Exercise 12.8 (♦) *With the notations of this section, show that any local minimizer of F is global.*

Link with Burer-Monteiro methods. The reparameterization of a convex problem $M \mapsto G(M)$ over the set of PSD matrices into a non-convex problem $W \mapsto G(WW^\top)$ has not been solely motivated by the will to explain the behavior of (linear) neural networks.

It has also been considered in the *low-rank matrix recovery* literature as an efficient tool to solve low-rank semidefinite-programming problems. This reparameterization that *deconvexifies* the problem for the sake of numerical computations enters into the class of "Burer-Monteiro" methods (Burer and Monteiro, 2003). Yet, as in the neural network case, its success is not yet totally explained (see Waldspurger, 2021, Section 6).

12.3.4 Special Cases

Oja Flow. As an illustration of the convergence results discussed in section 12.3.3, 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 be integrated in closed form. We have $G'(M) = M - C$, and thus the following gradient flow:

$$\dot{W} = -G'(WW^\top)W = CW - WW^\top W \quad \text{and} \quad \dot{M} = CM + MC - 2M^2.$$

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,[12] whose rank can be strictly less than m. Hence, the flow eventually converges to the global minimum of R over PSD matrices only if the number of particles is larger than the number of positive eigenvalues of C.

Vanishing initialization. For $m = d$, 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, with the same eigenvectors and eigenvalues equal to $\lambda = \frac{\alpha e^{2ct}}{1+\alpha c^{-1}(e^{2ct}-1)} = \frac{c}{1+e^{-2ct}(c/\alpha-1)} \approx \frac{c}{1+e^{-2ct}c/\alpha}$ for small α, where c is the corresponding eigenvalue of C.

Thus, when α is infinitesimally small (and therefore an initialization close to a stationary point), the eigenvalues λ stay near zero until they increase almost instantaneously to the final positive values c, and this increase happens at $t_c = \frac{1}{2c}\log\frac{1}{\alpha}$. We thus observe incremental learning for each eigenvector, with each eigenvector corresponding to a positive eigenvalue c, which is a very different optimization dynamic from the one obtained from projected GD, which corresponds to $\lambda = c(1 - e^{-t})$ and where all eigenvectors come in

[12] For a symmetric matrix C, with eigenvalue decomposition $C = \sum_{i=1}^m \lambda_i u_i u_i^\top$, the positive part is $C = \sum_{i=1}^m (\lambda_i)_+ u_i u_i^\top$.

together. This incremental learning at different time scales is common in nonconvex optimization; see Saxe et al. (2019) and Gidel et al. (2019) for linear networks, and Berthier (2023) and Pesme and Flammarion (2023) for diagonal linear networks, where precise statements can be made.

Matrix sensing. Another illustration of this problem is a simple instance of matrix sensing (Candes and Recht, 2012), where one has some observation matrices $(X_1, \dots, X_n)$ in $(\mathbb{R}^{d\times d})^n$ that we are assumed to be PSD, and observations $(y_1, \dots, y_n) \in \mathbb{R}^n$. We minimize the empirical loss

$$F(W) = \frac{1}{n}\sum_{i=1}^{n}(\langle WW^\top, X_i\rangle - y_i)^2 = G(WW^\top),$$

where $\langle A, B\rangle = \mathrm{tr}(AB^\top)$ is the usual dot product for matrices. Note that this fits into our framework because G is a convex function; hence, if the initialization has full rank, global convergence is guaranteed by our previous study. The question of which interpolator is selected by gradient methods has first been addressed by Gunasekar et al. (2017), but remains, in all its generality open. We address a simple instance of this problem in exercise 12.9.

Exercise 12.9 (♦♦) *Assume that for all $i, j \in \{1, \dots, n\}$, $X_iX_j = X_jX_i$, and consider the gradient flow $\dot{W} = -F'(W)$, initialized at $W_0 = \sqrt{\alpha}I$.*

- *Show that the dynamics of $M = WW^\top$ can be rewritten as a mirror flow.*
- *Give a variational characterization of the asymptotic implicit bias, i.e., the resulting $M_\infty = \lim_{t\to\infty} W_tW_t^\top$. In particular, show that when $\alpha \to 0$, M_∞ is the minimum nuclear norm interpolator.*

12.4 Lazy Regime and Neural Tangent Kernels (♦)

For overparameterized one-hidden-layer neural networks, with prediction functions of the following form (note the rescaling by $1/m$):

$$h(x; v_1, \dots, v_n) = \frac{1}{m}\sum_{j=1}^{m}\eta_j\sigma(w_j^\top x + b_j) = \frac{1}{m}\sum_{j=1}^{m}\Psi(v_j), \tag{12.26}$$

with Ψ and $v_1, \dots, v_m$ 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 both layers, leading to the mean-field limit thoroughly discussed in section 12.3.1 and associated with the non-Hilbertian norm γ_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, associated with the Hilbertian norm γ_2 defined in section 9.5. The input weights (w_j, b_j) are all

sampled without an extra scaling that depends on m. For the output weights, η_j can be $O(1)$ or $O(\sqrt{m})$ if initialized with zero mean (so 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 training regime, which we refer to as the "lazy" regime, following Chizat et al. (2019). It corresponds to initializing each η_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 α (here $\sqrt{m}$) multiplied by a classical model with a mean-field limit.

In the lazy regime, we end up minimizing $G(V) = \mathcal{R}(\alpha h(V))$ with respect to $V = (v_1, \dots, v_m)$, with a scaling factor $\alpha > 0$ 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 in equation (12.26) (where we consider only the dependence in V). Note that in this example $\alpha h(V) = \frac{1}{\sqrt{m}} \sum_{j=1}^m \Psi(v_j)$, and the overall rescaling constant is now $1/\sqrt{m}$.

We consider the gradient flow to minimize $G(V)$, with a step size $1/\alpha^2$ (scaling adapted to have a nontrivial dynamic); that is,

$$\frac{d}{dt} V(t) = -\frac{1}{\alpha^2} G'(V) = -\frac{1}{\alpha} Dh(V)^\top \mathcal{R}'(\alpha h(V(t))), \tag{12.27}$$

where $Dh(V)$ is the differential of h at V (i.e., a linear function from $\mathbb{R}^{(d+2)\times m}$ to $\mathcal{H}$). For the predictor $\alpha h(V)$, we get

$$\frac{d}{dt}[\alpha h(V(t))] = -Dh(V(t)) Dh(V(t))^\top \mathcal{R}'(\alpha h(V(t))). \tag{12.28}$$

We now describe informally the dynamics in the limit $\alpha \to +\infty$. At initialization $t = 0$, $\alpha h(V(t))$ is 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 $O(1)$. From equation (12.27), we obtain that the parameters V change at rate $O(1/\alpha)$, while from equation (12.28), the predictor changes at a rate that is independent of α. Thus, in the limit of large α, the parameters move only 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 of V, leading to attractive global convergence results for neural network training (see, e.g., Du et al., 2018). Moreover, since we have an explicit linear model, the lazy regime is well represented by a positive definite kernel, which we now define (with the same properties and guarantees 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 in pairs, with the same input weights and opposite output weights), then the affine model has only 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:

$$k(x, x') = \mathbb{E}\big[\sigma(w^\top x + b)\sigma(w^\top x' + b)\big] + \mathbb{E}\big[\sigma'(w^\top x + b)\sigma'(w^\top x' + b)(x^\top x' + 1)\big], \quad (12.29)$$

where the expectations are taken with respect to parameters (w, b) with distributions given by the chosen initialization (e.g., Gaussians). The first part in the right side of equation (12.29) is the traditional random feature kernel discussed in section 9.5, but it also has an additional part, which creates a richer model but cannot correct entirely the intrinsic 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 overparameterized 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 GD 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 overparameterization, 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 nonasymptotic results (in terms of convergence times and 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 support generic loss functions, starting with multicategory classification.
- As in 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.
- Nonsmooth surrogates can be defined in the general structured prediction framework, then extending support vector machines (SVMs).

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 (SVM 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 (i.e., 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 the examples in section 13.2). Most of the developments from section 4.1 will extend with appropriate modifications.

We start in section 13.1 with the natural extension to multicategory 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 nonsmooth surrogates in section 13.5. We then present generalization bounds in section 13.6 and experiments in section 13.7.

13.1 Multicategory 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 multidimensional vector-space valued functions. To preserve symmetry among classes, we will consider k-dimensional outputs (rather $(k-1)$-dimensional). That is, for $\mathcal{Y} = \{1, \dots, k\}$, we will estimate a function $g : \mathcal{X} \to \mathbb{R}^k$ and predict the label through $f(x) \in \arg\max_{j \in \{1,\dots,k\}} g_j(x) \subset \mathcal{Y}$.[1]

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 under the addition of a constant vector to $g(x) \in \mathbb{R}^k$.

In the binary case, the convex surrogates that we considered were all of the form $\Phi(yg(x))$ for a convex function Φ. In the multicategory case, there is significantly more diversity. In section 13.1.1, 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 (SGD) 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 applied to general output spaces.

Throughout this section on multicategory 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 1 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 as 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:

$$\mathbb{P}(y = j|x) = \frac{\exp(g_j(x))}{\sum_{i=1}^k \exp(g_i(x))} = \text{softmax}(g(x))_j$$

[1] Like for binary classification in section 4.1, equality cases do not really matter, and precise statements based on randomized predictions are left as exercises.

by definition of the softmax function from $\mathbb{R}^k$ to the simplex in $\mathbb{R}^k$, defined as $\mathrm{softmax}(u)_j = \exp(u_j)/\sum_{i=1}^k \exp(u_i)$.[2]

The negative log-likelihood (often referred to as the "cross-entropy loss") for this model is then equal to

$$\begin{aligned} S(y, g(x)) &= -\log \frac{\exp(g_y(x))}{\sum_{i=1}^k \exp(g_i(x))} = -g_y(x) + \log\bigg(\sum_{i=1}^k \exp(g_i(x))\bigg) \\ &= -\bar{y}^\top g(x) + \log\bigg(\sum_{i=1}^k \exp(g_i(x))\bigg). \end{aligned}$$

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, $\arg\max_{j\in\{1,\dots,k\}} g_*(x)_j$ is the Bayes predictor, which will lead to Fisher consistent estimation, as in the binary case in section 4.1.3. 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); see exercise 13.6.

⚠ Practitioners sometimes refer to the cross-entropy loss without the precision that the softmax function is taken beforehand (they, in fact, mean the softmax loss).

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 $g_*(x) = \mathbb{E}[\bar{y}|x] \in \mathbb{R}^k$. 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 versus all" framework presented later in this section), 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. Here, 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 ξ can obtained in closed form, leading to the surrogate:

$$S(y, g(x)) = \sup_{j\in\{1,\dots,k\}} \big\{ 1_{y\neq j} + g_j(x) - g_y(x) \big\}. \tag{13.1}$$

Finding the minimizer of the expected surrogate risk is not as easy. We have for a given $x \in \mathcal{X}$,

$$\mathbb{E}[S(y, g(x))|x] = \sum_{i=1}^k \mathbb{P}(y = i|x) \sup_{j\in\{1,\dots,k\}} \big\{ 1_{i\neq j} + g_j(x) - g_i(x) \big\}.$$

[2]Note that in order to avoid numerical instabilities, it is preferable to subtract the maximal value $U = \max\{u_1, \dots, u_k\}$ before computing the softmax function, as $\mathrm{softmax}(u) = \mathrm{softmax}(u - U1_k)$, where $1_k \in \mathbb{R}^k$ is the vector of all 1s.

Assuming without loss of generality that posterior probabilities given x are nonincreasing (i.e., $\mathbb{P}(y = 1|x) \geqslant \cdots \geqslant \mathbb{P}(y = k|x)$), the global minimizer of this quantity 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

$$\mathbb{P}(y = 1|x)(1 + g_2(x) - g_1(x))_+ + (1 - \mathbb{P}(y = 1|x))(1 + g_1(x) - g_2(x))_+.$$

If $\mathbb{P}(y = 1|x) > 1/2$, then the optimal $g_1(x) - g_2(x)$ can be shown to equal 1, and the prediction is optimal. Otherwise, it is not; hence, we lose Fisher consistency in general, since, for $k > 2$, it is not always the case that the conditional probability of the most likely class exceeds $1/2$ (see a precise statement in exercise 13.1). A consistent version of the nonsmooth hinge loss will be discussed in section 13.5.

Exercise 13.1 (♦) *For multicategory classification with the 0–1 loss, show that the loss defined in equation (13.1) is Fisher consistent if* $\forall x \in \mathcal{X}$, $\max_{j\in\{1,\ldots,k\}} \mathbb{P}(y = j|x) > \frac{1}{2}$.

One versus all (♦). This class of techniques essentially solve k binary optimization problems by solving *independently* for each $g_j(x)$ predicting $y_j \in \{0, 1\}$. Using convex surrogates for these binary classification 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 Φ being 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, 2004a, theorem 11).

Beyond. As reviewed by Zhang (2004a), there are many examples of convex surrogates to estimate the k functions $g_1, \ldots, g_k : \mathcal{X} \to \mathbb{R}$ based on several principles. Reductions to binary classification problems go beyond one versus all approaches, for example, by considering several subsets A of $\{1, \ldots, k\}$ and solving the binary classification problems of deciding $y \in A$ versus $y \notin A$. This approach based on error-correcting codes (Dietterich and Bakiri, 1994) will also be considered within the general surrogate framework.

Exercise 13.2 *Consider the following surrogate* $S(y, g(x)) = \sum_{i \neq y} \Phi(-g_i(x))$, *with the additional constraint that* $\sum_{i=1}^k g_i(x) = 0$, *and a strictly convex decreasing function* Φ. *Show that if* g_* *is the minimizer of* $\mathbb{E}[S(y, g(x))]$, *then for all* $i, j \in \{1, \dots, k\}$, *we have* $\mathbb{P}(y = i|x) > \mathbb{P}(y = j|x) \Rightarrow g_*(x)_i > g_*(x)_j$.

Exercise 13.3 *Consider the surrogate* $S(y, g(x)) = \sum_{i=1}^k \Phi(g_y(x) - g_i(x))$, *with a non-increasing function* Φ *such that that* $\Phi(z) < \Phi(-z)$ *for* $z > 0$. *Show that if* g_* *is the minimizer of* $\mathbb{E}[S(y, g(x))]$, *then for all* $i, j \in \{1, \dots, k\}$, $\mathbb{P}(y = i|x) > \mathbb{P}(y = j|x) \Rightarrow g_*(x)_i \geqslant g_*(x)_j$.

13.1.2 Generalization Bound I: Stochastic Gradient Descent

In sections 13.1.2 and 13.1.3, we will consider generalization bounds for losses that are Lipschitz-continuous (all losses in section 13.1.1 are Lipschitz-continuous, potentially once restricted to a bounded set), as done in chapter 4 with estimation errors controlled by Rademacher complexities, and in chapter 5 using single-pass SGD. We start with SGD because the analysis can be done without the need for new tools.

Linear models. Since we will use convergence results for convex optimization algorithms, we need to consider linear models (which are linear in their parameters). We thus assume that we have a feature vector $\varphi : \mathcal{X} \to \mathbb{R}^d$ almost surely bounded by R in the ℓ_2-norm, and 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 θ, restricted to a ball of radius D for a certain norm Ω.

Several candidates are natural for the norm Ω: the simplest 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 matrix θ seen as a vector, and for which all results related to SGD will apply. Another classical norm is the nuclear norm (aka trace norm) defined as the sum of singular values of θ, which will push for low-rank θ with similar properties as the ℓ_1-norm in section 8.3. Other norms, such as the ℓ_1-norm or "grouped" norms (discussed in section 8.5), 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 13.4 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 this book, we assume that we have n independent and identically distributed (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 given previously, 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 section 13.1.3, 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). Thus, we focus on the excess surrogate risk in this section.

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 is 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 that 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 (with respect to the second variable) $S'(y,u) \in \mathbb{R}^k$ are bounded by G in the ℓ_2-norm (for example, for softmax regression, we have $G = 1$).

Single-pass SGD. We consider the following SGD iteration for $t \in \{1,\dots,n\}$, with an arbitrary subgradient of surrogate S, started from $\theta_0 = 0$:

$$\theta_t = \Pi_D\big(\theta_{t-1} - \gamma_t \varphi(x_t) S'(y_t, \theta_{t-1}^\top \varphi(x_t))^\top\big),$$

where Π_D is the orthogonal projection on the set of matrices with Frobenius norm less than D. The analysis of section 5.4 exactly applies, and with the choice of constant step size $\gamma_t = \frac{D}{RG\sqrt{n}}$, we obtain the following generalization bound for the averaged iterate $\bar\theta_n$:

$$\mathbb{E}\Big[\mathcal{R}_S(g^{(\bar\theta_n)}) - \inf_{\|\theta\|_{\mathrm{F}}\leqslant D} \mathcal{R}_S(g^{(\theta)})\Big] \leqslant \frac{DRG}{\sqrt{n}}, \tag{13.2}$$

which is exactly the same as for real-valued predictions. Note that in terms of dependence in the number k of categories, since θ has k columns, we can expect D to grow as $\sqrt{k}$, while typically R does not (for G, it depends on the convex surrogate, with $G = 1$ for the softmax loss).

Exercise 13.4 (Mirror descent for trace-norm penalty (♦)) *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 θ. Show that for $p \in (1,2)$, it 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 that we followed in this book is to assume that we can compute a minimizer of the empirical risk beyond linear models (in particular for neural networks). We thus assume a generic space $\mathcal{G}$ 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 sections 4.2 and 4.5, the expected estimation error $\mathcal{R}_S(\hat g) - \inf_{g\in\mathcal{G}} \mathcal{R}_S(g)$ is then less than four times the Rademacher complexity:

$$\mathrm{R}_n(S, \mathcal{G}) = \mathbb{E}\bigg[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i S(y_i, g(x_i))\bigg],$$

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 (proposition 4.3) that allows us to get rid of the surrogate cost S so long as it is Lipschitz-continuous. Such a contraction principle also exists for vector-valued prediction functions (Maurer, 2016) and is presented in proposition 13.1. Its application leads to

$$\mathrm{R}_n(S, \mathcal{G}) \leqslant \sqrt{2} G \cdot \mathbb{E}\bigg[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n (\varepsilon_i')^\top g(x_i)\bigg], \tag{13.3}$$

where 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 next (and compare them to the SGD bound from earlier) and neural networks in exercise 13.5.

Linear models. In the setup of section 13.1.2 (linear models with Frobenius bound), we can further upper-bound equation (13.3) as

$$\begin{aligned}
\mathrm{R}_n(S, \mathcal{G}) &\leqslant \sqrt{2} G \cdot \mathbb{E}\bigg[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n (\varepsilon_i')^\top g(x_i)\bigg] = \sqrt{2} G \cdot \mathbb{E}\bigg[\sup_{\|\theta\|_{\mathrm{F}} \leqslant D} \frac{1}{n} \sum_{i=1}^n (\varepsilon_i')^\top \theta^\top \varphi(x_i)\bigg] \\
&= \frac{D}{n} \sqrt{2} G \cdot \mathbb{E}\bigg[\Big\| \sum_{i=1}^n \varphi(x_i)(\varepsilon_i')^\top \Big\|_{\mathrm{F}}\bigg] \leqslant \frac{DG}{n} \sqrt{2} \cdot \bigg(\mathbb{E}\bigg[\Big\| \sum_{i=1}^n \varphi(x_i)(\varepsilon_i')^\top \Big\|_{\mathrm{F}}^2\bigg]\bigg)^{1/2} \\
&\leqslant \frac{DG}{n} \sqrt{2} \cdot \bigg(\mathbb{E}\bigg[\sum_{i=1}^n \|\varphi(x_i)\|_2^2 \, \|\varepsilon_i'\|_2^2\bigg]\bigg)^{1/2} \leqslant \frac{DGR\sqrt{2k}}{\sqrt{n}}.
\end{aligned}$$

We thus obtain the same bound as in equation (13.2), 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 nonconvex loss functions (so long as they are Lipschitz-continuous) and predictors that are nonlinear in their parameters (such as neural networks).

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 proposition 4.3 for $k = 1$, and we start with 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-function (with respect to the ℓ_2-norm), we have, for $\varepsilon \in \{-1, 1\}$, a Rademacher random variable, and for $\varepsilon' \in \{-1, 1\}^n$, 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 explicitly the expectation with respect to ε, 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 (θ,θ') and (θ',θ) 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 proposition 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 use Khintchine's inequality from lemma 11.1 (with optimal constants); 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 following bound:

$$\begin{aligned}&\sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{\sqrt{2}}{2}\mathbb{E}\Big[\Big|\sum_{j=1}^k\varepsilon'_j(a_j(\theta)-a_j(\theta'))\Big|\Big]\\ \leqslant\ &\mathbb{E}\Big[\sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{\sqrt{2}}{2}\Big|\sum_{j=1}^k\varepsilon'_j(a_j(\theta)-a_j(\theta'))\Big|\Big]\\ &\qquad\text{using properties of expectations and suprema,}\\ =\ &\mathbb{E}\Big[\sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{\sqrt{2}}{2}\sum_{j=1}^k\varepsilon'_j(a_j(\theta)-a_j(\theta'))\Big]\ \text{by symmetry,}\\ \leqslant\ &\mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sqrt{2}\sum_{j=1}^k\varepsilon'_j a_j(\theta)\Big\}\Big],\ \text{which is the desired result.}\end{aligned}$$

■

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 ℓ_2-norm), for $i=1,\dots,n$, we have, for $\varepsilon\in\{-1,1\}^n$, a vector of independent Rademacher random variables, and for $\varepsilon'\in\{-1,1\}^{n\times k}$, 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

$$\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n,\varepsilon'_{n+1}}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sqrt{2}\sum_{j=1}^{k}\varepsilon'_{n+1,j}a_{n+1,j}(\theta)+\sum_{i=1}^{n}\varepsilon_i c_i(a_i(\theta))\Big\}\Big],$$

and apply the induction hypothesis with ε'_{n+1} fixed to obtain the desired result. ■

Exercise 13.5 (Multicategory classification with neural networks (♦)) *Consider neural networks 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 $\eta_j \in \mathbb{R}^k$. 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 Setup and Examples

Now that the multicategory classification has been presented, we consider the same general setup discussed 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 $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

As in section 2.2, given a test distribution p on $\mathcal{X}\times\mathcal{Y}$, we can define the Bayes predictor

$$f_*(x) \in \arg\min_{z\in\mathcal{Y}} \int_{\mathcal{Y}} \ell(y,z)dp(y|x) \tag{13.4}$$

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):

- **Multicategory 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 (e.g., 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 ρ nonconvex. When ρ 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 nonrobust to strong outlier perturbations.

Having a nonconvex ρ, such as $\rho(\delta) = 1 - \exp(-\delta^2)$, leads to robust regression (see more developments in Huber and Ronchetti, 2009).

- **Ordinal regression**: This is a particular case of multicategory classification, where the loss matrix has a specific structure in which 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. See Pedregosa et al. (2017) and references therein.

- **Multiple labels**: $\mathcal{Y} = \{-1, 1\}^k$, with cardinality 2^k, with the traditional Hamming loss $\ell(y, z) = \frac{1}{2}\|y - z\|_1 = \frac{1}{4}\|y - z\|_2^2$, which counts the number of mistakes and will be a running example in this chapter. Other performance metrics, such as precision/recall[3] or F-scores,[4] are typically used (and may not be symmetric) and can be treated as well with the frameworks presented in this chapter. These are detailed in 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. Note here that choosing the 0–1 loss is not advocated, as it corresponds to a multicategory classification problem with 2^k classes and the 0–1 loss, for which we outlined in section 13.1 the statistical dependence of generalization bounds on the (here exponential) number of classes.

- **Permutations**: $\mathcal{Y}$ is the set of permutations among m elements; that is, y is a bijection from $\{1, \dots, m\}$ to $\{1, \dots, m\}$. We have then $|\mathcal{Y}| = m!$. A common loss function is the "pairwise disagreement," which counts the number of pairs where y and z provide different rankings; it is thus equal to $\ell(y, z) = \sum_{i,j=1}^m 1_{y(i)>y(j)} 1_{z(i)<z(j)}$, but other losses such as the discounted cumulative gain[5] can be used, or losses of the form $\sum_{i=1}^m \ell_i(a_{y(i)} - a_{z(i)})$ for a fixed vector $a \in \mathbb{R}^m$ (e.g., $a_j = j$) and functions $\ell_i : \mathbb{R} \to \mathbb{R}$ (e.g., the square function). 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 an 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 $\mathcal{Y}$ is thus large (or infinite), and the Hamming loss is commonly used.

- **Trees and graphs**: $\mathcal{Y}$ is the set of potentially labeled 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.

[3] See https://en.wikipedia.org/wiki/Precision_and_recall.
[4] See https://en.wikipedia.org/wiki/F-score.
[5] See https://en.wikipedia.org/wiki/Discounted_cumulative_gain.

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 precisely later in this chapter) *and* input dimension d, in addition to having 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 the following:

- Computational tractability by introducing convex surrogates (to use convex optimization) and efficient decoding steps (often dedicated algorithms). These convex surrogates lead to explicit guarantees: quantitative for linear models (as in sections 13.1.2 and 13.6) and qualitative for overparameterized models (as shown in section 12.3). Note here that in this chapter, we aim to design convex surrogates, which can then be applied to potentially nonlinear models (in their parameters, such as neural networks).
- Fisher consistency (excess risk equal to zero in the population case) and calibration functions (suboptimality for the convex surrogate leads to suboptimality for the true risk with an explicit dependence).

Following the rest of the book, we will always go through vector-space valued prediction functions. Thus, there will always be two components:

- Learning some vector-valued score functions from data, implicitly or explicitly, in a Hilbert space $\mathcal{H}$ or $\mathbb{R}^k$, where k is the potentially implicit "affine dimension" of $\mathcal{Y}$.
- Decoding procedure to go from scores 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 multicategory 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.

Beyond risks defined as expectations. In this book, in all situations, binary or multicategory classification, regression, or more generally structured prediction, we consider criteria based on (potentially empirical) expectations of $\ell(y, f(x))$, where $f(x)$ is the prediction, y is the label, and ℓ is a loss function. Some commonly used performance criteria do not fit this framework, such as the area under the ROC curve (referred to as the AUC [6]) and can rather be expressed as a "U-statistic,"[7] for which specific developments can be carried out (see, e.g., Gao and Zhou, 2015, and references therein).

[6] See https://en.wikipedia.org/wiki/Receiver_operating_characteristic for details.
[7] See https://en.wikipedia.org/wiki/U-statistic.

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

$$\forall (y, z) \in \mathcal{Y} \times \mathcal{Y}, \ \ell(y, z) = c + \langle \chi(z), \psi(y) \rangle. \tag{13.5}$$

This assumption is called "structure encoding loss function (SELF)" (Ciliberto et al., 2016, 2020). This can be an implicit or explicit embedding (see the examples that follow). Note that the representation is not unique, as given a pair (χ, ψ), any pair $(V^{-1}\chi, V^*\psi)$ is valid for any invertible operator.

⚠ There are *two* embeddings of outputs in $\mathcal{Y}$, while typically there is only one for the inputs in $\mathcal{X}$.

Bayes predictor. With the assumption in equation (13.5), we can now express the optimal predictor in equation (13.4) as

$$f_*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \Big\langle \chi(z), \int_{\mathcal{Y}} \psi(y) dp(y|x) \Big\rangle. \tag{13.6}$$

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 $f_*(x)$. This will lead to nonsmooth 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 the theoretical analysis in section 13.6, we will need a bound on $R_\ell = \sup_{z \in \mathcal{Y}} \|\chi(z)\|$, which we also provide here (all proofs left as exercises). Note that implicit SELF decompositions exist under general conditions (Ciliberto et al., 2020).

- **Binary classification**, with $\mathcal{Y} \in \{-1, 1\}$ and the 0–1 loss: $\mathcal{H} = \mathbb{R}$, $\chi(z) = -z/2$ and $\psi(y) = y$, since $\ell(y, z) = 1_{y \neq z} = \frac{1}{2} - \frac{yz}{2}$, with $R_\ell = 1/2$.
- **Multicategory classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i, j) = L_{ij}$. The decomposition corresponds to the usual "one-hot" encoding of discrete distributions, where $\psi(i) \in \mathbb{R}^k$ is the ith element of the canonical basis (with values in $\{0, 1\}$). We then have $\ell(i, j) = L_{ij} = \psi(i)^\top L \psi(j)$; that is, $\chi(j) = L\psi(j) = L(:, j) \in \mathbb{R}^k$ (the jth-column of matrix L). 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 $L_{ij} = 1_{i \neq j}$, we can write $\ell(i, j) = 1 - \psi(i)^\top \psi(j)$, and we can consider the simpler embedding $\chi(j) = -\psi(j)$, with $R_\ell = 1$ (note here that since the components of $\psi(i)$ sum to 1

for all i, we can add the same constant to the vectors $\chi(j)$, in a way that can reduce R_ℓ from $\sqrt{k-1}$ to 1).

We can also choose to have a feature map ψ 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 of the function $u \mapsto e^{-u^2}$, $\ell(y,z) = 1 - \frac{1}{2\sqrt{\pi}}\int_{-\infty}^{\infty} \exp(-\omega^2/4)\cos\omega(y-z)d\omega$, which in turn 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 $R_\ell = \sqrt{k}/2$.

- **Permutations on** $\{1,\dots,m\}$: 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 $\ell(y,z) = \sum_{i=1}^m (y(i)-z(i))^2$, we have $\psi(y) = y$ and $\chi(z) = -2z$, with $R_\ell \leqslant \sqrt{2}(m+1)$.

- **Sequences**: We consider binary sequences for simplicity (i.e., $\mathcal{Y} = \{-1,1\}^m$), but it extends more generally to all factor graphs (Wainwright and Jordan, 2008) and types of labels. Using the Hamming loss (as 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 features $y_1,\dots,y_m \in \{-1,1\}$, we can add the features $y_1y_2, y_2y_3, \dots, y_{m-1}y_m \in \{-1,1\}$, which allow for considering losses that encourage perfectly predicted sequences of size 2, for example, by considering the loss $\ell(y,z) = \sum_{j=1}^{m-1} 1_{(y_j,y_{j+1})\neq(z_j,z_{j+1})} = \sum_{j=1}^{m-1}\big\{1 - \frac{1}{4}(1-y_jz_j)(1-y_{j+1}z_{j+1})\big\}$.

As for binary classification or regression, the loss choice is independent of the function space considered (local averaging, kernels, neural networks).

Reduction to binary problems. We have encountered several examples in this discussion where the feature map Ψ has binary values in $\{-1,1\}^m$ or $\{0,1\}^m$. We will see next 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 multicategory classification with the 0–1 loss.

13.3 Surrogate Methods

In this section, our main concern will be to obtain Fisher consistent, convex surrogates: convex so that we can run efficient algorithms from chapter 5, and 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 function f defined as $f(x) = \text{sign}(g(x)) \in \{-1, 1\}$. 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 present the general surrogate framework next.

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."[8]
- $\text{dec} : \mathcal{H} \to \mathcal{Y}$ is the "decoding function," which can be randomized (in particular when taking maxima of functions that may have equal values).

We then need a surrogate loss $S : \mathcal{Y} \times \mathcal{H} \to \mathbb{R}$, which will be used to form empirical and expected surrogate risks:

$$\widehat{\mathcal{R}}_S(g) = \frac{1}{n} \sum_{i=1}^{n} S(y_i, g(x_i)) \quad \text{and} \quad \mathcal{R}_S(g) = \mathbb{E}\big[S(y, g(x))\big].$$

For example, for binary classification where $\mathcal{Y} = \{-1, 1\}$, we had $S(y, g(x)) = \Phi(yg(x))$ for Φ a convex function.

13.3.2 Fisher Consistency and Calibration Functions

Following the same definition as in section 4.1, we denote $\mathcal{R}_S^*$ as the minimim S-risk, which 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

$$\mathcal{R}_S^* = \mathbb{E}\Big[\inf_{h \in \mathcal{H}} \mathbb{E}\big[S(y, h)|x\big]\Big].$$

As for binary classification in section 4.1.3 (where we also used the term classification-calibrated), the loss is said to be "Fisher consistent" if the optimal predictor in the population case (i.e., when minimizing the surrogated expected risk) is the Bayes predictor defined in equation (13.6).

As in binary classification in section 4.1.4, 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*

[8] In statistics, the score function often refers to the gradient of the log density with respect to parameters (and sometimes with respect to inputs). There is no link between these two definitions.

function (i.e., an increasing function $H : \mathbb{R}_+ \to \mathbb{R}_+$) such that $\mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}^* \leq H\big[\mathcal{R}_S(g) - \mathcal{R}_S^*\big]$. Note that, as 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).

13.3.3 Main Surrogate Frameworks

As described in section 4.1, for binary classification, we saw two main 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 include 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.
- Nonsmooth surrogates, where the predictor minimizing the expected surrogate risk already provided a thresholded version; that is, $\mathrm{sign}(\mathbb{E}[y|x])$. 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) quadratic (or, more generally, smooth surrogates), (2) max-margin (nonsmooth functions that estimate 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 commonly used (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 defined in section 13.2.2. We 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 equation (13.5), 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 $\|\cdot\|$ of the space $\mathcal{H}$. In other words, we aim to estimate $\mathbb{E}\big[\psi(y)|x\big]$ directly for every $x \in \mathcal{X}$. The decoding function is then naturally[9]

$$\operatorname{dec}(s) \in \arg\min_{z\in\mathcal{Y}} \ \langle \chi(z), s\rangle, \tag{13.7}$$

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 (i.e., $\ell(y,z) = k(y,y) + k(z,z) - 2k(y,z)$), 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 here, we can prove a precise calibration result by using the properties of the square loss, as obtained in equation (4.8) (section 4.1.4). We first notice that

$$\mathcal{R}_S(g) - \mathcal{R}_S^* = \mathbb{E}\Big[\big\|g(x) - \mathbb{E}[\psi(y)|x]\big\|^2\Big]. \tag{13.8}$$

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 $f_* = \operatorname{dec} \circ g_*$.

We can then express the excess risk using the decomposition of the loss as

$$\begin{aligned}
& \mathcal{R}(\operatorname{dec}\circ g) - \mathcal{R}^* \\
= \ & \mathcal{R}(\operatorname{dec}\circ g) - \mathcal{R}(\operatorname{dec}\circ g_*) \\
= \ & \mathbb{E}\Big[\mathbb{E}\big[\ell(y, \operatorname{dec}\circ g(x)) - \ell(y, \operatorname{dec}\circ g_*(x))\big|x\big]\Big] \\
= \ & \mathbb{E}\Big[\mathbb{E}\big[\big\langle \psi(y), \chi(\operatorname{dec}\circ g(x)) - \chi(\operatorname{dec}\circ g_*(x))\big\rangle\big|x\big]\Big] \text{ by the SELF decomposition,} \\
= \ & \mathbb{E}\Big[\big\langle \mathbb{E}\big[\psi(y)|x\big], \chi(\operatorname{dec}\circ g(x)) - \chi(\operatorname{dec}\circ g_*(x))\big\rangle\Big] \text{ by moving expectations,} \\
= \ & \mathbb{E}\Big[\big\langle \mathbb{E}\big[\psi(y)|x\big] - g(x), \chi(\operatorname{dec}\circ g(x)) - \chi(\operatorname{dec}\circ g_*(x))\big\rangle\Big] \\
& \qquad + \mathbb{E}\Big[\big\langle g(x), \chi(\operatorname{dec}\circ g(x)) - \chi(\operatorname{dec}\circ g_*(x))\big\rangle\Big]
\end{aligned}$$

by adding and subtracting $g(x)$. The definition of the decoding function in equation (13.7) implies the negativity of the second term. Thus, we get, using the Cauchy-Schwarz

[9] As in binary classification in section 4.1, when the minimizer is not unique, we predict uniformly at random among the minimizers. Moreover, in bounds, expectations are taken with respect to this additional (independent) randomization.

inequality:

$$\begin{aligned}
\mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}^* \leqslant\ & \mathbb{E}\Big[\big\langle \mathbb{E}\big[\psi(y)|x\big] - g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g_*(x))\big\rangle\Big] \\
\leqslant\ & 2 \sup_{z \in \mathcal{Y}} \|\chi(z)\| \cdot \mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|\Big] \\
\leqslant\ & 2 \sup_{z \in \mathcal{Y}} \|\chi(z)\| \cdot \sqrt{\cdot \mathbb{E}\Big[\big\|\langle \mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|^2\Big]} \text{ using Jensen's inequality,} \\
=\ & 2R_\ell \cdot \sqrt{\mathcal{R}_S(g) - \mathcal{R}_S^*} \text{ because of equation (13.8),}
\end{aligned} \tag{13.9}$$

which is precisely a calibration function result. A key feature of this result is that the constant R_ℓ typically does not explode, even for sets $\mathcal{Y}$ with large cardinality (see the examples in section 13.2.2). To get a learning bound for the structured prediction problem, 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 regression 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 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

$$\mathrm{dec} \circ g(x) \in \arg\min_{z \in \mathcal{Y}} \Big\langle \chi(z), \sum_{i=1}^n w_i(x)\psi(y_i) \Big\rangle = \arg\min_{z \in \mathcal{Y}} \sum_{i=1}^n w_i(x)\ell(y_i, z). \tag{13.10}$$

This corresponds to approximating the conditional distribution of y given x by the discrete distribution $\sum_{i=1}^n w_i(x)\delta_{y_i}$ (note that for local averaging methods, this approximation is a probability distribution, but for kernel methods, the weights w_i may not be nonnegative).

Only the loss function is needed for equation (13.10); thus, there is no need to know the explicit loss decomposition to run the testing algorithm. This makes the decoding step even easier in the following examples:

- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with the loss $\ell(y, z) = 1 - \exp\big[-(y - z)^2\big]$. equation (13.10) 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.

- **Multicategory classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k\times k}$, with $\ell(i,j) = L_{ij}$. Equation (13.10) 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 $\ell(y,z) = \frac{k}{2} - \frac{1}{2} y^\top z$. Equation (13.10) 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 no longer has a closed form 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 $\sum_{j=1}^m u_j z_j + \sum_{j=1}^{m-1} v_j z_j z_{j+1}$ for some vectors u and v, 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 following form, also considered by Blondel et al. (2020):

$$S(y, g(x)) = c(y) - 2\langle \psi(y), g(x)\rangle + 2a(g(x)),$$

where $c : \mathcal{Y} \to \mathbb{R}$ is an arbitrary function, $a : \mathcal{H} \to \mathbb{R}$ is convex and β-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 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

$$\mathrm{dec}(h) \in \arg\min_{z\in\mathcal{Y}} \ \chi(z)^\top a'(h), \tag{13.11}$$

with randomized predictions when several $z \in \mathcal{Y}$ minimize $\chi(z)^\top a'(h)$. For the square loss, we recover exactly the quadratic surrogate.

Examples. Three examples are particularly interesting:

- **Softmax regression:** For multicategory 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 $a(y) = \log\big(\sum_{j=1}^k \exp(y_j)\big)$.
- **Reduction to binary logistic regression:** When $\psi(y) \in \{-1,1\}^m$, we can consider $a(h) = \sum_{i=1}^m \log(\exp(h_i/2) + \exp(-h_i/2))$, leading to independent logistic regressions. This is a specific instance of error-correcting codes (Dietterich and Bakiri, 1994).

- **Graphical models (⧫):** These last two examples 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). To build 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 regressions, while for the sequence models, for $\psi(y)$ composed of all y_j and $y_j y_{j+1}$, 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, when 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 the function a_σ through $a_\sigma(z) = \mathbb{E}\big[a_0(z+\sigma u)\big]$ 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,[10] but the framework is more generally applicable (see Papandreou and Yuille, 2011; Berthet et al., 2020).

Calibration function. The computations in section 13.4.2 for quadratic surrogates can be extended to smooth surrogates, leading to an extension of the binary case discussed in section 4.1.4.

We have, by definition of the Fenchel conjugate $a^*(u) = \sup_{h\in\mathcal{H}} \langle u, h\rangle - a(h)$,

$$
\begin{aligned}
\mathcal{R}_S(g) &= \mathbb{E}\Big[\mathbb{E}[c(y)|x] - 2\langle \mathbb{E}[\psi(y)|y], g(x)\rangle + 2a(g(x))\Big] \\
\mathcal{R}_S^* &= \mathbb{E}\Big[\mathbb{E}[c(y)|x] + \inf_{h\in\mathcal{H}} \big\{ - 2\langle \mathbb{E}[\psi(y)|x], h\rangle + 2a(h)\big\}\Big] \\
&= \mathbb{E}\big[\mathbb{E}[c(y)|x] - 2a^*(\mathbb{E}[\psi(y)|x])\big], \text{ by definition of } a^*,
\end{aligned}
$$

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^*, with the same reasoning as in section 4.1.4.

[10] See https://francisbach.com/the-gumbel-trick/ for details.

Moreover, as in section 13.4.2, we can express the excess risk as

$$\begin{aligned}
\mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}^* &= \mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}(\mathrm{dec}\circ g_*) \\
&= \mathbb{E}\Big[\mathbb{E}\big[\ell(y,\mathrm{dec}\circ g(x)) - \ell(y,\mathrm{dec}\circ g_*(x))\big|x\big]\Big] \\
&= \mathbb{E}\Big[\mathbb{E}\big[\big\langle\psi(y),\chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g_*(x))\big\rangle\big|x\big]\Big] \\
&= \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big],\chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g_*(x))\big\rangle\Big] \\
&= \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - a'(g(x)),\chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g_*(x))\big\rangle\Big] \\
&\qquad\qquad + \mathbb{E}\Big[\big\langle a'(g(x)),\chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g_*(x))\big\rangle\Big].
\end{aligned}$$

By definition of the decoding step in equation (13.11), we get

$$\begin{aligned}
\mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}^* &\leqslant \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - a'(g(x)),\chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g_*(x))\big\rangle\Big] \\
&\leqslant 2\sup_{z\in\mathcal{Y}}\|\chi(z)\|\,\mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - a'(g(x))\big\|\Big] \\
&\leqslant 2\sup_{z\in\mathcal{Y}}\|\chi(z)\|\cdot\sqrt{\mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - a'(g(x))\big\|^2\Big]} \leqslant 2\sqrt{\beta}R_\ell\sqrt{\mathcal{R}_S(g) - \mathcal{R}_S^*}.
\end{aligned}$$

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 (see exercise 13.6) and conditional random fields.

Exercise 13.6 (Softmax calibration function) *Derive a calibration function for the softmax regression loss function described at the beginning of section 13.1.1.*

Exercise 13.7 (Reduction to binary problems) *Derive a calibration function when $\psi(y) \in \{-1,1\}^m$ for all $y \in \mathcal{Y}$, and we use the logistic surrogate independently on each of the m components.*

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 for a ranking example).

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, with reference to the geometric interpretation from section 4.1.2. In this section, we assume that for any $z \in \mathcal{Y}$, $y \mapsto \ell(y,z)$ is minimized at z; that is, the loss $\ell(\cdot,z)$ provides a measure of dissimilarity with z.

13.5.1 Structured Support Vector Machines

Following Taskar et al. (2005) and Tsochantaridis et al. (2005), we consider a traditional extension of the support vector machine (SVM) with a simple interpretation.

To introduce the convex surrogate in its full generality, we consider a score function h that 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 surrogate function $S(y, h(x, \cdot))$ 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 to make $h(x, y)$ 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. For multicategory classification with the 0–1 loss, we recover exactly the hinge loss from section 13.1.1, while for binary classification, we recover the SVM from section 4.1.2. Taking the smallest possigle ξ leads to the following convex surrogate: $\max_{z \in \mathcal{Y}} \; h(x, z) - h(x, y) + \ell(z, y) - \ell(y, y)$.

If we take the particular form $h(x, z) = \langle \psi(z), g(x) \rangle$ for ψ defined in section 13.2.2 and $g : \mathcal{X} \to \mathcal{H}$, 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

$$\xi \geqslant \langle \psi(z) - \psi(y), \chi(y) + g(x) \rangle.$$

Thus, the surrogate function is

$$S(y, g(x)) = \max_{z \in \mathcal{Y}} \langle \psi(z) - \psi(y), \chi(y) + g(x) \rangle. \tag{13.12}$$

This convex loss is computable as soon as we can maximize linear functions of $\psi(z)$; 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 multicategory classification with the 0–1 loss and a "majority class," where the approach is consistent (Liu, 2007) (see also exercise 13.1).

13.5.2 Max-Min Formulations (♦♦)

Following Fathony et al. (2016) and Nowak-Vila et al. (2020), we can provide a nonsmooth surrogate, which is both Fisher-consistent and comes with a calibration function that does not have a square root. In the binary case, the SVM led to a target surrogate function, which was exactly the Bayes predictor, with values in $\{-1, 1\}$. We will see that it is possible to reproduce this behavior in the general case. We still assume that for any

$z \in \mathcal{Y}$, $y \mapsto \ell(y,z)$ is minimized at z (and only at z); that is, the loss provides a measure of dissimilarity with y.

Given the expression of the Bayes predictor in equation (13.6)–that is,

$$f_*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \big\langle \chi(z), \mathbb{E}[\psi(y)|x] \big\rangle,$$

we consider the function $g_*(x) = -\chi(f_*(x)) \in \mathcal{H}$, which is defined as

$$g_*(x) \in -\arg\min_{h \in \chi(\mathcal{Y})} \big\langle h, \mathbb{E}[\psi(y)|x] \big\rangle = \arg\max_{h \in -\chi(\mathcal{Y})} \big\langle h, \mathbb{E}[\psi(y)|x] \big\rangle. \tag{13.13}$$

(For simplicity, we only analyze the case where the minimizer is unique, leaving the general case as an exercise.) The value $g_*(x)$ happens to be a subgradient at $\mathbb{E}[\psi(y)|x]$ of the convex function

$$b : \mu \mapsto -\min_{h \in \chi(\mathcal{Y})} \langle h, \mu \rangle = -\min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle. \tag{13.14}$$

Thus, if we can design a surrogate function S in such as way that $\mathbb{E}[S(y, g(x))|x]$ has minimizer $g_*(x)$ as defined in equation (13.13), we can consider the following decoding function with our desired consistent behavior:

$$\mathrm{dec} \circ g(x) \in \arg\max_{y' \in \mathcal{Y}} \ \psi(y')^\top g(x),$$

since we have

$$\begin{aligned} \mathrm{dec} \circ g_*(x) \quad &\in \quad \arg\max_{y' \in \mathcal{Y}} \ \psi(y')^\top g_*(x) = \arg\min_{y' \in \mathcal{Y}} \ \psi(y')^\top \chi(f_*(x)) \\ &= \quad \arg\min_{y' \in \mathcal{Y}} \ \ell(y', f_*(x)) = f_*(x), \end{aligned}$$

because we assumed that the loss $\ell(y,z)$ is minimized with respect to y at z. Note the difference with the decoding function of the quadratic surrogate in equation (13.7) (minimization instead of maximization, and ψ instead of χ).

To enforce that a subgradient $g_*(x)$ of b at $\mathbb{E}[\psi(y)|x]$ is a minimizer of $\mathbb{E}[S(y, g(x))|x]$, it is sufficient to consider the following (but note that this is not the only choice):

$$S(y, g(x)) = b^*(g(x)) - \langle g(x), \psi(y) \rangle, \tag{13.15}$$

where b^* is the Fenchel conjugate of b restricted to $\mathcal{M}(\psi) \subset \mathcal{H}$ defined as the closure of the convex hull of all $\psi(z), z \in \mathcal{Y}$; that is,

$$b^*(h) = \max_{\mu \in \mathcal{M}(\psi)} \langle \mu, h \rangle - b(\mu) = \max_{\mu \in \mathcal{M}(\psi)} \langle \mu, h \rangle + \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle.$$

Indeed, with the definition of $S(y, \cdot)$ in equation (13.15), 0 is a subgradient of the function $h \mapsto \mathbb{E}[S(y,h)|x] = b^*(h) - \langle h, \mathbb{E}[\psi(y)|x] \rangle$ if and only if $\mathbb{E}[\psi(y)|x]$ is a subgradient of b^*

at h, which is equivalent to h being a subgradient of b at $\mathbb{E}[\psi(y)|x]$ (and by construction, $g_*(x)$ is such a subgradient).

We then have

$$\begin{aligned} S(y, g(x)) &= \max_{\mu \in \mathcal{M}(\psi)} \Big\{ \langle g(x), \mu \rangle + \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle \Big\} - \langle g(x), \psi(y) \rangle \qquad (13.16) \\ &= \max_{\mu \in \mathcal{M}(\psi)} \min_{y' \in \mathcal{Y}} \Big\{ \langle g(x) + \chi(y'), \mu - \psi(y) \rangle + \ell(y, y') \Big\}. \end{aligned}$$

Note the similarity with the max-margin SVM loss in equation (13.12), 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 it leads to a Fisher consistent estimator.

Fisher consistency. We now confirm 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; that is,

$$\operatorname{dec} \circ g_*(x) = \arg\max_{y' \in \mathcal{Y}} \ \psi(y')^\top g_*(x) = f_*(x).$$

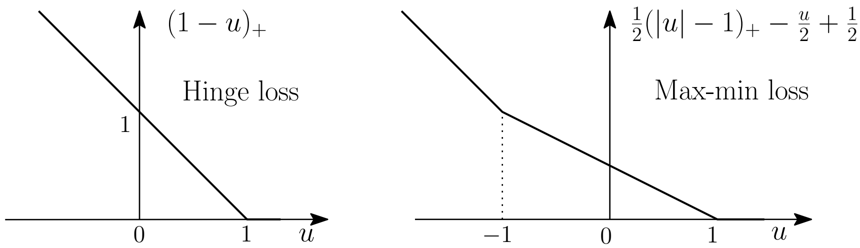
As in section 13.4.4, for a fixed $x \in \mathcal{X}$, any minimizer g_* has a value $g_*(x)$ that minimizes

$$\mathbb{E}\big[S(y, g(x))|x\big] = b^*(g(x)) - \big\langle g(x), \mathbb{E}[\psi(y)|x] \big\rangle,$$

with respect to $g(x)$. By the definition of b^*, $-g_*(x)$ is a minimizer of $h \mapsto \big\langle h, \mathbb{E}[\psi(y)|x] \big\rangle$ over $h \in \mathcal{M}(\chi)$. Thus, given the expression of the Bayes predictor in equation (13.6), 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 $y \mapsto \ell(y, z)$ is minimized at z. We can also get a linear calibration function in generic situations; see Nowak-Vila et al. (2020) for details and exercise 13.8 for binary classification.

Optimization algorithms. In this book, we have focused on optimization methods based on subgradients. For the loss defined in equation (13.16), this requires an optimizer $\mu \in \mathcal{M}(\psi)$, which in turn requires solving a min-max problem in general. Next, we consider the multicategory 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, $\psi(y) = y$ and $\chi(z) = -z/2$. We can first compute the structured SVM cost function from equation (13.12) as $(1 - 2yg(x))_+$ and recover (up to the factor two) the regular SVM from section 4.1.2. We can also compute $b(\mu) = \frac{1}{2}|\mu|$ from equation (13.14) with domain $[-1, 1]$, leading to $b^*(f) = (|f| - 1/2)_+$, and, from equation (13.15), the convex surrogate $S(y, g(x)) = (|g(x)| - 1/2)_+ - yg(x)$, a formulation that is close to the binary SVM (but nonidentical), as shown in the following plot, where $u = 2yg(x)$, and we plot the hinge loss $(1 - u)_+$ and the new max-min loss $\frac{1}{2}(|u| - 1)_+ - \frac{u}{2} + \frac{1}{2}$ (where we added the constant $\frac{1}{2}$ so that it remains nonnegative).



Exercise 13.8 *For the binary classification problem with 0–1 loss, show that the max-min formulation leads to a linear calibration function between excess and excess surrogate risks.*

Exercise 13.9 (Multicategory classification, Fathony et al., 2016 (♦)) *For $\mathcal{Y} = \{1, \dots, k\}$ and the 0–1 loss, with $\psi(y) = \bar{y}$ and $\chi(z) = -\bar{z}$ (one-hot encodings), show that the min-max convex surrogate is $S(y, g(x)) = \max_{A \subset \{1,\dots,k\}, A \neq \varnothing} \frac{\sum_{j \in A} g_j(x) - 1}{|A|} - g_y(x)$, where the maximizers in A can be obtained in closed form by sorting the vector $g(x)$ (together with a subgradient).*

13.6 Generalization Bounds (♦)

In this section, we provide generalization bounds for the structured prediction problem with quadratic surrogates $S(y, g(x)) = \|\psi(y) - g(x)\|^2$ as defined in section 13.4.1. For simplicity, we will assume that (1) the SELF decomposition in section 13.2.2 is finite-dimensional (i.e., $\psi : \mathcal{Y} \to \mathbb{R}^k$); (2) we consider linear models of the form $g^{(\theta)}(x) = \theta^\top \varphi(x) \in \mathbb{R}^k$ with feature vector $\varphi : \mathcal{X} \to \mathbb{R}^d$ and $\theta \in \mathbb{R}^{d \times k}$; and (3) the feature vector is flexible enough so the minimizer of the expected surrogate risk is indeed a linear function of φ; that is, $\mathbb{E}[\psi(y)|x] = \theta_*^\top \varphi(x)$ for some $\theta_* \in \mathbb{R}^{d \times k}$. 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 SGD. For multicategory classification in section 13.1, we considered both, and we only consider SGD in this section because it leads to better bounds. Moreover, we could use kernel methods when φ is known only 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 surrogates, we could directly extend the ridge regression analysis from section 7.6, which does not use Rademacher complexities. Instead, we focus on bounds obtained by single-pass SGD. We only cover the quadratic surrogate from section 13.4.1 (see exercise 13.10 for Lipschitz-continuous smooth losses); thus, we will use results from section 5.4.3, in particular the bound in equation (5.28).

We assume i.i.d. observations $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, as well as the single-pass SGD recursion, initialized at 0 and defined for $t \in \{1, \dots, n\}$, as

$$\theta_t = \theta_{t-1} - \gamma_t \varphi(x_t) S'(y_t, \theta_{t-1}^\top \varphi(x_t))^\top = \theta_{t-1} - 2\gamma_t \varphi(x_t)(\theta_{t-1}^\top \varphi(x_t) - \psi(y_t))^\top.$$

The bound in equation (5.28) exactly applies (with the improvement of exercise 5.35), and with the choice $\gamma_t = \gamma$, we obtain the following generalization bound for the surrogate risk of the averaged iterate $\bar{\theta}_n$ (note a difference of a factor of 2 compared to equation (5.28) due to different normalization):

$$\mathbb{E}\big[\mathcal{R}_S(g^{(\bar{\theta}_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{1}{\gamma n}\|\theta_*\|_{\mathrm{F}}^2 + \gamma \sup_{x \in \mathcal{X}} \|\varphi(x)\|_2^2 \cdot \mathcal{R}_S(g^{(\theta_*)}).$$

With the optimal choice of γ, we get

$$\mathbb{E}\big[\mathcal{R}_S(g^{(\bar{\theta}_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{2}{\sqrt{n}}\|\theta_*\|_{\mathrm{F}} \cdot \sup_{x \in \mathcal{X}} \|\varphi(x)\|_2 \cdot \mathcal{R}_S(g^{(\theta_*)})^{1/2}.$$

Without any information on $\mathcal{R}_S(g^{(\theta_*)})$, we can bound it using $\theta_* = 0$ (then with a bound $\sup_{y \in \mathcal{Y}} \|\Psi(y)\|_2^2$), leading to

$$\mathbb{E}\big[\mathcal{R}_S(g^{(\bar{\theta}_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{2}{\sqrt{n}}\|\theta_*\|_{\mathrm{F}} \cdot \sup_{x \in \mathcal{X}} \|\varphi(x)\|_2 \cdot \sup_{y \in \mathcal{Y}} \|\Psi(y)\|_2.$$

We can then use the calibration result in equation (13.9) to obtain consistency for the structured prediction problem with the following bound:

$$\mathbb{E}\big[\mathcal{R}(\mathrm{dec}(g^{(\bar{\theta}_n)}))\big] - \mathcal{R}^* \leqslant \frac{2^{3/2}}{n^{1/4}}\|\theta_*\|_{\mathrm{F}}^{1/2} \cdot \sup_{z \in \mathcal{Y}} \|\chi(z)\|_2 \cdot \sup_{x \in \mathcal{X}} \|\varphi(x)\|_2^{1/2} \cdot \sup_{y \in \mathcal{Y}} \|\Psi(y)\|_2^{1/2}. \quad (13.17)$$

Thus, if all embeddings have bounded norms, we obtain a convergence rate proportional to $n^{-1/4}$ for the excess risk (after decoding). Note that the lack of explicit dependence in the dimension k of the output embeddings ψ and χ; however, such dependence is implicit in the norms of ψ and χ, as well as in the norm of θ_*.

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 (as we did in this section) or the nuclear norm (which requires using stochastic mirror descent, as presented in exercise 13.4) as a regularization or constraint. However, there are setups where the k columns of θ have a specific structure, and thus, some particular squared norm $\mathrm{tr}[\theta^\top \theta M^{-1}]$ can be natural. This can then be obtained by preconditioning the gradient in the SGD recursion as done in section 5.4.2.

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).

Exercise 13.10 (♦) *Instead of the quadratic surrogate from section 13.4.1, consider a smooth surrogate from section 13.4.4 with the additional assumption that function a has bounded gradients. Extend the bounds presented in this section.*

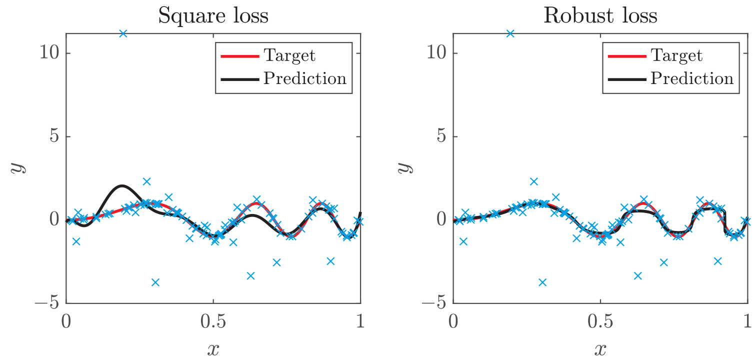


Figure 13.1. Robust regression in one dimension, with heavy-tail noise (fifth power of Gaussian noise): regular square loss (left) versus robust loss (right).

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

Here, we consider a toy robust regression problem to illustrate the use of quadratic surrogates presented in section 13.4.1. We look at 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 generate data with heavy-tail additive noise and plot in figure 13.1 the best performance for kernel ridge regression with the Gaussian kernel, with the optimal regularization parameter (selected for best 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 compute $\arg\min_{z\in\mathbb{R}} \sum_{i=1}^n w_i(x)\ell(y_i,z) = \arg\min_{z\in\mathbb{R}} \sum_{i=1}^n w_i(x)(1-\exp(-(y_i-z)^2))$ at test time, which can be done by grid searching. See the results shown 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 loss functions:

- **Square loss:** $\ell(y,x) = \sum_{i=1}^m (y(i)-z(i))^2$, with output embedding simply equal to $\psi^{(\mathrm{sq})}(y) = y \in \{1,\dots,m\}^m$. For this loss, we only consider the (natural) square surrogate. We thus need to fit a function $h : \mathcal{X} \to \mathbb{R}^m$ using least-squares regression directly on y. The decoding step for test point x is then simply to sort the m components of $h(x)$.

- **Pairwise disagreement:** $\ell(y,z) = \sum_{i,j=1}^m \left(1_{y(i)<y(j)} - 1_{z(i)<z(j)}\right)^2$, with the feature $\psi^{(\mathrm{pw})}(y) \in \{-1,1\}^{m(m-1)/2}$, defined as $\psi_{ij}^{(\mathrm{pw})}(y) = 2 \cdot 1_{y(i)<y(j)} - 1$ for $i<j$ (up to constants, this is the same formulation as in section 13.2.1). We thus need to learn function $g: \mathcal{X} \to \mathbb{R}^{m(m-1)/2}$; we consider the square loss, where we minimize expectations of $\|\psi^{(\mathrm{pw})}(y) - g(x)\|_2^2$, as well as the logistic loss, where we minimize the expectation of $\sum_{i<j} \log\left(1 + \exp(-\psi(y)_{ij}^{(\mathrm{pw})}(x) g_{ij}(x))\right)$ (then, the estimate of $\mathbb{E}[\psi_{ij}^{(\mathrm{pw})}(y)|x]$ is equal to $\tanh(g_{ij}(x)/2)$).

  In terms of decoding, given function g, at test point x, we need to maximize $\sum_{i<j} g_{ij}(x) 1_{z(i)<z(j)}$ with respect to permutation z when using the square loss, while we need to maximize $\sum_{i<j} \tanh(g_{ij}(x)/2) 1_{z(i)<z(j)}$ when using the logistic loss. This is an instance of the "minimum feedback arc set problem", which is an NP-hard problem (Cormen et al., 2022) with known approximation algorithms (Ailon et al., 2008).[11] When the weights $g_{ij}(x)$ take the form $u(h_j(x) - h_i(x))$ for a nondecreasing 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, it turns out that using this specific model leads exactly to performing least-squares regression on $y \in \mathbb{R}^m$ (proof left as an exercise).

Plackett-Luce model. We generate data from the Plackett-Luce model (Marden, 1996): from m functions $h_1(x), \dots, h_m(x) \in \mathbb{R}$, we obtain a random permutation by sorting the m real values $h_1(x) + \eta_1, \dots, h_m(x) + \eta_m$ in ascending order, where each η_i, $i = 1, \dots, m$, is a Gumbel random variable.[12] 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 smallest, and $y(i) = m$ if $h_i(x) + \eta_i$ is the largest.

If $\pi(x) = \mathrm{softmax}(h(x))$, 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

$$\pi(x)_{z(m)} \frac{\pi(x)_{z(m-1)}}{1 - \pi(x)_{z(m)}} \frac{\pi(x)_{z(m-2)}}{1 - \pi(x)_{z(m)} - \pi(x)_{z(m-1)}} \cdots \frac{\pi(x)_{z(2)}}{\pi(x)_{z(1)} + \pi(x)_{z(2)}}.$$

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 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, stated as $\mathbb{E}[y|x]$, can be expressed as a product of softmax functions of subsets of $h_1, \dots, h_m$, and thus it is not linear in these functions.

We consider $\mathcal{X} = [0,1]$ and functions $h_1, \dots, h_m$, which are linear combinations of cosine functions $\cos(2\pi k x)$ and sine functions $\sin(2\pi k x)$ for $k \in \{0,1\}$ for the generating functions. See figure 13.2 for an illustration.

[11] In our experiments, since m is small, we use exhaustive search.

[12] See https://en.wikipedia.org/wiki/Gumbel_distribution.

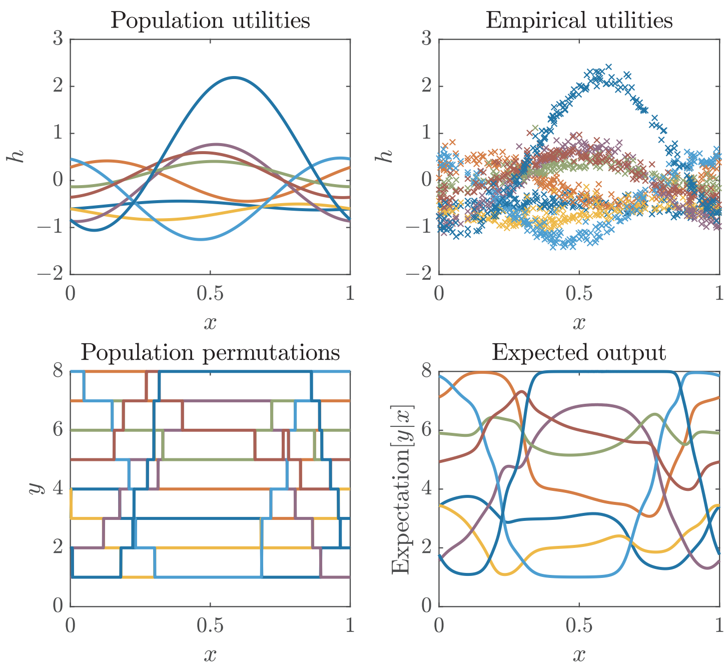


Figure 13.2. Top: utilities $h_1, \ldots, h_m$ (left) and empirical utilities $h_1 + \eta_1, \ldots, h_m + \eta_m$ (right). Bottom: population permutations $y^*(x) \in \{1, \ldots, m\}$ (left) and conditional expectation $\mathbb{E}[\psi^{(\mathrm{sq})}(y)|x] = \mathbb{E}[y|x] \in [1, m]$ (right).

We consider situations where the prediction model used 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 $h_1, \ldots, h_m$ (i.e., $k \in \{0, 1\}$), while for the middle model, $k \in \{0, \ldots, 3\}$, and for the large model, $k \in \{0, \ldots, 15\}$. We use a fixed regularization parameter proportional to $1/n$.

In the top-left plot of figure 13.3, the quadratic surrogate with $\psi^{(\mathrm{sq})}(y) = y$ is considered and the small model is misspecified. Thus, when n grows, the testing error does not go down to zero (regardless of the way it is measured, as the three top plots exhibit a similar behavior), and it is similar for the bottom-left plot, where the square surrogate is used with the pairwise disagreement loss function. For the logistic surrogate, however, where even the small model is well specified, we obtain a learning curve that goes closer to zero, as shown in the middle and right bottom plots in figure 13.3; in the rightmost plot, when considering predictors that follow Plackett-Luce model (i.e., $g_{ij} = h_i - h_j$), since fewer functions are learned, we obtain as expected a slightly better behavior.

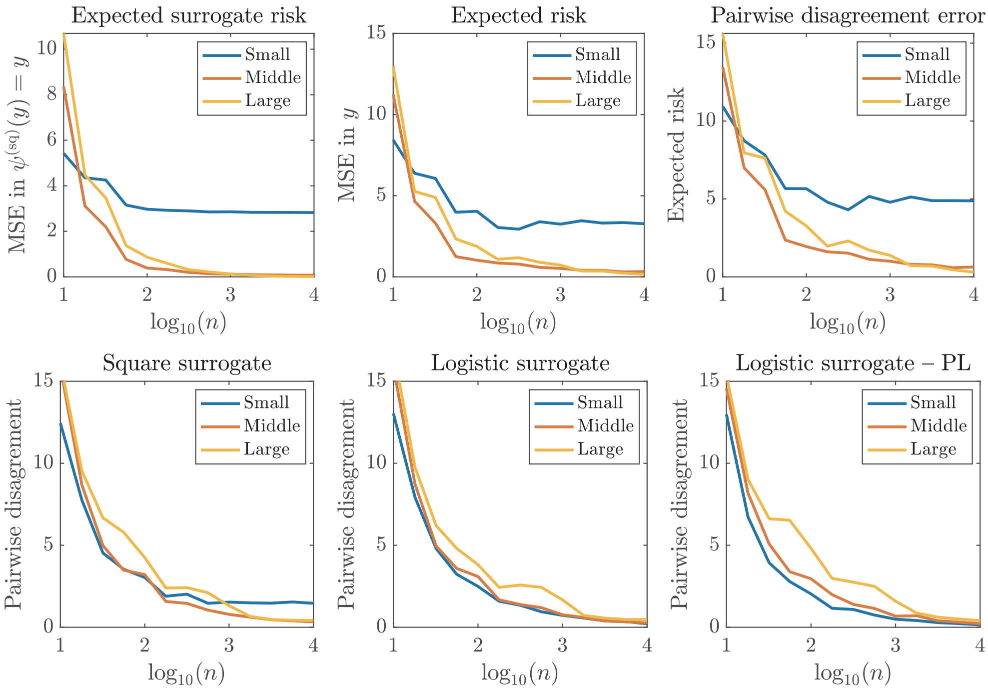


Figure 13.3. Top: using the square loss $\ell(y, z) = \|y - z\|_2^2$ with the square surrogate; left: testing error (expected risk) for the quadratic surrogate (before sorting); middle: testing error (expected risk) for the square loss (after sorting); right: testing pairwise disagreement (after sorting). Bottom: expected risks using the pairwise disagreement loss $\ell(y, z) = \|\psi^{(\mathrm{pw})}(y) - \psi^{(\mathrm{pw})}(z)\|_2^2$ with the square surrogate (left) and logistic surrogates (middle: with $m(m-1)/2$ functions, right: using the Plackett-Luce model, i.e., pairwise differences of m functions).

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, such as quadratic surrogates, margin-based techniques, and frameworks with probabilistic interpretations 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 error (MMSE) estimation is preferable (for the square loss).
- Generative models, such as linear discriminant analysis (LDA), which 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, 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-Bayesian 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

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

multiply by $-n$, and take the exponential to get

$$\begin{aligned}\exp(-n\widehat{\mathcal{R}}(\theta)) &= \exp\Big(-\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) - \lambda\Omega(\theta)\Big) \\ &= \prod_{i=1}^{n} \exp\big[-\ell(y_i, f_\theta(x_i))\big] \cdot \exp\big[-\lambda\Omega(\theta)\big]. \qquad (14.1)\end{aligned}$$

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

$$\exp(-n\widehat{\mathcal{R}}(\theta)) \quad \propto \quad \prod_{i=1}^{n} p(y_i|x_i, \theta) \cdot p(\theta),$$

which is precisely the conditional likelihood for the model where θ is a parameter and where, given θ, all pairs (x_i, y_i) are independent and identically distributed (i.i.d.).

⚠ Be careful with the 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, such as, among others, generative models, Bayesian inference for hyperparameter learning (as done in later sections), and dealing with missing data through the expectation-maximization algorithm.

⚠ We only scratch the surface here, from a learning theory point of view. See Murphy (2012) and Bishop (2006) for many more details.

In this section, we primarily focus on the formulation in equation (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 $p(y_i|x_i) = \text{sigmoid}(y_i f_\theta(x_i))$, where sigmoid : $\alpha \mapsto (1 + e^{-\alpha})^{-1}$ is the sigmoid function

To apply logistic regression, there is no need to assume that the model is well specified; that is, there exists a θ_* so that the data are actually generated from the conditional model above. For the nonparametric analysis (with flexible models such as kernel methods or neural networks), 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).

There is no need to have Gaussian noise! Having zero mean and bounded variance is enough for the analysis.

Exercise 14.1 *Show that the negative log density of the Gaussian distribution with mean μ and variance σ^2 (i.e., $-\log p(y|\mu, \sigma) = \frac{1}{2\sigma^2}(x - \mu)^2 + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log\sigma^2$) is not convex in (μ, σ^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 ℓ_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 ℓ_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 as well. One that is common in the Bayesian setting is the multivariate Student distribution (often used marginally for independent components,

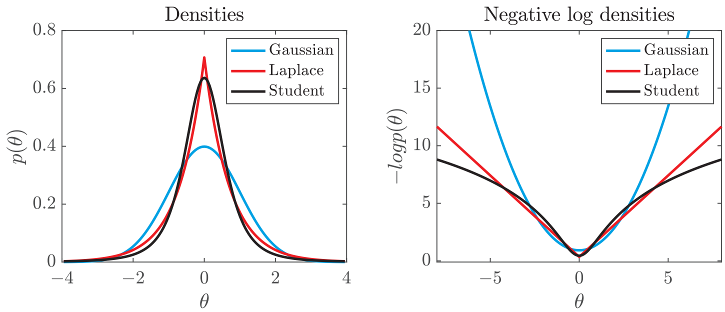


Figure 14.1. Classical priors in one dimension, all normalized to zero mean and unit variance: (left) densities, (right) negative log densities.

such as in section 14.1.3):

$$p(\theta) \propto \left(\beta + \tfrac{1}{2}\|\theta\|_2^2\right)^{-\alpha-d/2},$$

for some $\alpha, \beta > 0$, leading to the regularizer $(\alpha+d/2)\log(\beta+\frac{1}{2}\|\theta\|_2^2)$, which is not convex in θ. This will be used within sparse priors in section 14.1.3.

Exercise 14.3 (♦) *Consider a random vector θ that is Gaussian with mean zero and covariance matrix ηI, with $1/\eta$ being distributed as a Gamma random variable with parameters α and β; that is, η with density $p(\eta) = \frac{\beta^\alpha}{\Gamma(\alpha)}(1/\eta)^{\alpha+1}\exp(-\beta/\eta)$. Show that the marginal density of θ 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 $\alpha > 1$.*

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 (i.e., there exists θ_* such that $p(y|x)$ is indeed proportional to $\exp(-\ell(y, f_{\theta_*})))$; and (2) the optimal θ_* is sampled from the prior distribution proportional to $\exp(-\lambda\Omega(\theta))$. As we explain in section 14.1.4, this is not the case *at all*.

14.1.3 Sparse Priors

As shown in section 14.1.4, the Laplace prior is not adapted to sparse data (which may seem counterintuitive since the MAP estimate leads to ℓ_1-penalization). We consider the following ones instead. For each one-dimensional component, we consider the following (see illustrations in figures 14.1 and 14.2):

- 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: $p(\theta) = \frac{1}{(2\pi)^{1/2}}\frac{\beta^\alpha\Gamma(\alpha+1/2)}{\Gamma(\alpha)}(\beta+\frac{1}{2}\theta^2)^{-\alpha-1/2}$, with variance $\frac{\beta}{\alpha-1}$ if $\alpha > 1$.

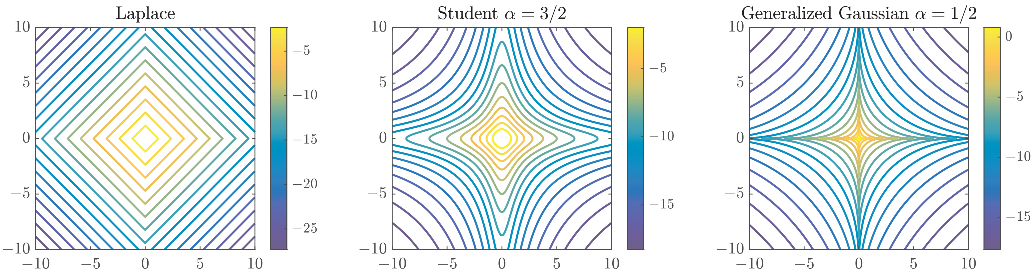


Figure 14.2. Sparse priors in two dimensions. Left: Laplace distribution, middle: Student distribution, right: generalized Gaussian distribution.

- Mixture of two Gaussians: $p(\theta) = \alpha\mathcal{N}(\theta|0,\sigma_0^2) + (1-\alpha)\mathcal{N}(\theta|0,\tau^2)$, with variance $\alpha\sigma_0^2 + (1-\alpha)\tau^2$.

It turns out that all 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 means 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 Dirac measures at σ_0^2 and τ^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 (i.e., 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 "η-trick" from section 8.3.1, and thus can provide a Bayesian interpretation.

Exercise 14.4 *A density $p(\theta)$ on $\mathbb{R}$ is said to be "super-Gaussian" if $\log p(\theta)$ is convex in θ^2 and nonincreasing. 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 ε is Gaussian with zero mean and covariance matrix $\sigma^2 I$, assuming that σ^2 is known. We have prior knowledge about θ in the form of a prior density $q(\theta)$.[2] Given

[1] The converse is not true; see Palmer et al. (2005).

[2] We favor the notation $q(\theta)$ over $p(\theta)$ to avoid confusion with $p(y)$ later in this subsection.

the observation of y, our goal is to obtain an estimator of θ 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

$$J(\hat{\theta}) = \int_{\mathbb{R}^d} \|\theta - \hat{\theta}(y)\|_2^2 q(\theta) d\theta.$$

As shown in section 2.2.3, the optimal estimator (i.e., function) $\hat{\theta} : \mathbb{R}^d \to \mathbb{R}^d$ is equal to the *a posteriori mean*; that is,

$$\hat{\theta}_{\mathrm{MMSE}}(y) = \mathbb{E}\big[\theta | y\big],$$

assuming that θ is sampled according to $q(\theta)$ and y follows the model in equation (14.2). Here, MMSE stands for "minimum mean square error." We now want to compare it with the maximum a posteriori (MAP) parameter:

$$\hat{\theta}_{\mathrm{MAP}}(y) \in \arg\max_{\theta \in \mathbb{R}^d} \; p(\theta|y) = \arg\max_{\theta \in \mathbb{R}^d} \; q(\theta) p(y|\theta).$$

Gaussian prior. When q is a Gaussian distribution with mean zero and covariance matrix $\tau^2 I$, then (θ, y) is a Gaussian vector; and from conditioning results presented in section 1.1.3, we have

$$\hat{\theta}_{\mathrm{MMSE}}(y) = \mathbb{E}\big[\theta | y\big] = \frac{\tau^2}{\tau^2 + \sigma^2} y,$$

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 in this chapter, 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,

$$\begin{aligned} p(y) &= \int_{\mathbb{R}^d} p(y,\theta) d\theta = \int_{\mathbb{R}^d} q(\theta) p(y|\theta) d\theta \\ &= \int_{\mathbb{R}^d} q(\theta) \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\Big(- \frac{1}{2\sigma^2} \|\theta - y\|_2^2 \Big) d\theta, \end{aligned}$$

using the expression of the Gaussian density. We can now express the a posteriori mean as follows, introducing the gradient of the Gaussian density:

$$\begin{aligned} \hat{\theta}_{\mathrm{MMSE}}(y) &= \mathbb{E}\big[\theta|y\big] = \int_{\mathbb{R}^d} p(\theta|y)\theta d\theta = \int_{\mathbb{R}^d} \frac{p(\theta,y)}{p(y)} \theta d\theta \\ &= y + \sigma^2 \int_{\mathbb{R}^d} \frac{p(y|\theta) q(\theta)}{p(y)} \frac{1}{\sigma^2} (\theta - y) d\theta \\ &= y + \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q(\theta) \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\Big(- \frac{1}{2\sigma^2} \|\theta - y\|_2^2 \Big) \frac{1}{\sigma^2} (\theta - y) d\theta \\ &= y - \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q(\theta) \frac{\partial}{\partial\theta} \Big[\exp\Big(- \frac{1}{2\sigma^2} \|\theta - y\|_2^2 \Big) \Big] d\theta. \end{aligned}$$

Thus, using integration by parts,[3] we get

$$\begin{aligned}
\hat{\theta}_{\mathrm{MMSE}}(y) &= y + \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q'(\theta) \exp\Big(-\frac{1}{2\sigma^2}\|\theta - y\|_2^2\Big) d\theta \\
&= y + \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{\sigma^2}{p(y)} \int_{\mathbb{R}^d} q'(y-\eta) \exp\Big(-\frac{1}{2\sigma^2}\|\eta\|_2^2\Big) d\eta, \text{ with } \theta = y - \eta, \\
&= y + \frac{\sigma^2}{p(y)} p'(y) = y + \sigma^2 \frac{d}{dy}(\log p(y)). \qquad (14.3)
\end{aligned}$$

We thus get an explicit expression of the MMSE estimate. Note that for a Gaussian prior, y is marginally distributed as a Gaussian; 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 follows:* $\mathrm{var}(\theta|y) = \sigma^2 I + \sigma^4 \dfrac{d^2}{dy dy^\top}(\log p(y))$.

Link with empirical Bayes. We have shown that for any arbitrary prior distribution $q(\theta)$, with the conditional model $y|\theta$ being Gaussian with mean 0 and covariance matrix $\sigma^2 I$, the conditional expectation $\mathbb{E}[\theta|y] = y + \sigma^2 \frac{d}{dy}(\log p(y))$ can be expressed through the density of y. This density depends only implicitly on the prior distribution of θ. This generalizes to other noise models (see exercise 14.6) and can be used within "empirical Bayes" procedures described in section 14.3.2.

Exercise 14.6 *Assume that θ has an arbitrary prior distribution supported on $\mathbb{R}_+$, and, given θ, y has a Poisson distribution with mean θ; that is, $\mathbb{P}(y = k|\theta) = \frac{1}{k!} e^{-\theta}\theta^k$, for k a nonnegative integer. Show that $\mathbb{E}[\theta|y] = (y+1)\frac{p(y+1)}{p(y)}$. Extend this result to the geometric distribution $\mathbb{P}(y = k|\theta) = (1-\theta)\theta^k$, for θ supported on $[0, 1]$.*

Expression of the MAP estimate. If $q(\theta) = \exp(-h(\theta))$, then the MAP estimate is

$$\hat{\theta}_{\mathrm{MAP}}(y) \in \arg\min_{\theta\in\mathbb{R}^d} \ \frac{1}{2\sigma^2}\|\theta - y\|_2^2 + h(\theta),$$

with optimality condition, for differentiable h, $\theta - y - \sigma^2 \frac{d}{d\theta}(\log q(\theta)) = 0$; thus we have

$$\hat{\theta}_{\mathrm{MAP}}(y) = y + \sigma^2 \frac{d}{dy}(\log q)\big[\hat{\theta}_{\mathrm{MAP}}(y)\big]. \qquad (14.4)$$

Exercise 14.7 (Gribonval, 2011 (♦♦)) *We denote $f(y) = -\log p(y)$. Show that the MMSE estimator $\hat{\theta}_{\mathrm{MMSE}}(y) = y - \sigma^2 f'(y)$ defined in equation (14.3) is the MAP estimator for the negative log-prior g that satisfies $g(\hat{\theta}_{\mathrm{MMSE}}(y)) = f(y) - \frac{\sigma^2}{2}\|f'(y)\|_2^2$ for all $y \in \mathbb{R}^d$.*

[3] Integration by parts applies more generally to any Gaussian random vector z with mean μ and covariance matrix Σ, leading to Stein's lemma, stating that for any real-valued differentiable function g on $\mathbb{R}^d$, $\mathbb{E}[g(z)(z-\mu)] = \Sigma \cdot \mathbb{E}[g'(z)]$; see https://en.wikipedia.org/wiki/Stein's_lemma.

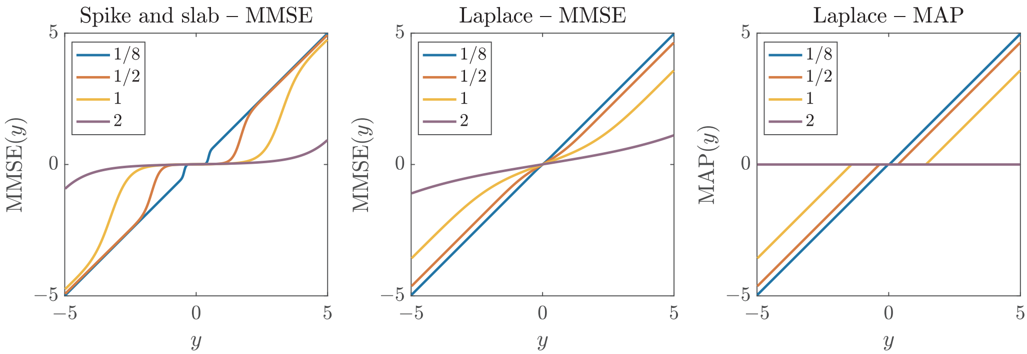


Figure 14.3. Comparison of MMSE and MAP for the spike-and-slab and Laplace priors for $\sigma = 1$, and varying standard deviations for the prior distribution (1/8, 1/2, 1, and 2): MMSE for the spike-and-slab prior (left), MMSE for the Laplace prior (middle), MAP for the Laplace prior (right).

Differences between MMSE and MAP. Given the expressions in equations (14.3) and (14.4), we can now study how the two estimators differ for the various sparse priors that we have described here, where we consider the one-dimensional case for simplicity (which extends to independent marginal priors in the multidimensional case); see the plots in figure 14.3:

- Spike-and-slab: This is the model essentially used in the analysis of the Lasso problem in chapter 8 (i.e., weight vectors with exact zeros), for which MAP with the Laplace prior (i.e., the Lasso problem) is shown to have favorable properties. We consider the prior, which is the mixture of a Dirac measure at zero (with weight α) and a Gaussian distribution with mean zero and variance τ^2. The variance is then equal to $(1-\alpha)\tau^2$, and $p(y)$ is the mixture of two Gaussian distributions, centered on zero, with variances σ^2 and $\sigma^2 + \tau^2$.

  Exercise 14.8 *Show that the marginal density $p(y)$ for the spike-and-slab prior is equal to $p(y) = \alpha \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(\frac{-y^2}{2\sigma^2}\right) + (1-\alpha)\frac{1}{(2\pi(\sigma^2+\tau^2))^{1/2}} \exp\left(\frac{-y^2}{2(\sigma^2+\tau^2)}\right)$. Provide an expression of $\hat{\theta}_{\mathrm{MMSE}}(y)$ and of $\hat{\theta}_{\mathrm{MAP}}(y)$.*

- 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 equals $2/\lambda^2$. We can compute the MMSE by explicitly computing $p(y)$ by integrating separately over positive and negative numbers (see exercise 14.9). We see in figure 14.3 that the MMSE is very far from the soft-thresholding operator from section 8.3.1 (right plot). 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; see a more quantitative analysis in Gribonval et al. (2012).

  Exercise 14.9 *Show that the marginal density $p(y)$ for the Laplace prior can be* *expressed using the Gauss error function* $\operatorname{erf}(\alpha) = \frac{2}{\sqrt{\pi}}\int_0^\alpha \exp(-t^2)dt$*, as* $p(y) = \frac{1}{\lambda}\exp\big(\frac{\lambda^2\sigma^2}{2} - \lambda y\big)\big[1 - \operatorname{erf}\big(\frac{\lambda\sigma - \frac{y}{\sigma}}{\sqrt{2}}\big)\big] + \frac{1}{\lambda}\exp\big(\frac{\lambda^2\sigma^2}{2} + \lambda y\big)\big[1 - \operatorname{erf}\big(\frac{\lambda\sigma + \frac{y}{\sigma}}{\sqrt{2}}\big)\big]$*. Provide an expression of* $\hat\theta_{\mathrm{MMSE}}(y)$ *and of* $\hat\theta_{\mathrm{MAP}}(y)$*.*

Exercise 14.10 *When q is a Gaussian distribution with mean zero and covariance matrix C, provide an expression of the MMSE and MAP estimates.*

Exercise 14.11 (♦) *Provide a closed-form expression for the marginal density $p(y)$ for the Student prior.*

14.2 Discriminative versus Generative Models

We now consider a traditional supervised learning setup, with $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The goal of supervised learning can be cast as follows: for any $x \in \mathcal{X}$, obtaining a good conditional predictive model of y given x; that is, getting a good model for $p(y|x)$.

We can first directly model $p(y|x)$ with a parameterized conditional model (as 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's 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, leading to $p(y|x) \propto p(x|y)p(y)$. 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 $y \in \{1, \dots, k\}$:

$$\begin{aligned} y &\sim \text{multinomial}(\pi) \\ x|y = i &\sim \text{Gaussian}(\mu_i, \Sigma). \end{aligned}$$

We can then compute the distribution of y given x as follows (removing all parts that are independent of i):

$$\begin{aligned} \mathbb{P}(y = i|x) &\propto \mathbb{P}(y = i, x) = \pi_i \exp\big[-\tfrac{1}{2}(x - \mu_i)^\top \Sigma^{-1}(x - \mu_i)\big] \\ &\propto \pi_i \exp\big[-\tfrac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i\big]\exp(\mu_i^\top \Sigma^{-1} x). \end{aligned}$$

This implies that, defining the softmax function $\mathrm{softmax} : \mathbb{R}^k \to \mathbb{R}^k$ through $\mathrm{softmax}(v)_j = \frac{e^{v_j}}{e^{v_1} + \cdots + e^{v_k}}$,

$$\mathbb{P}(y = i|x) = \mathrm{softmax}\big[(\mu_i^\top \Sigma^{-1} x + \log \pi_i - \tfrac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i)_i\big] = \mathrm{softmax}\big[(w_i^\top x + b_i)_i\big]_i;$$

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 $b_i = \log \pi_i - \frac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i$. This model is referred to as linear discriminant analysis (LDA).[4]

The availability of a generative model will lead to alternative parameter estimation algorithms (as discussed next). Note that (1) for $k = 2$, we recover logistic regression, and (2) 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 discussed here 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 π is the vector of empirical proportions of each class. Similarly, the estimator of each mean μ_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 (see exercise 14.12).

Exercise 14.12 *For the LDA model, show that the maximum likelihood estimates of the parameters are* $\hat{\pi}_i = \frac{1}{n}\sum_{j=1}^n 1_{y_j=i}$, $\hat{\mu}_i = \frac{1}{n\hat{\pi}_i}\sum_{j=1}^n 1_{y_j=i} x_j$, *for* $i \in \{1, \dots, k\}$, *and* $\widehat{\Sigma} = \frac{1}{n}\sum_{j=1}^n \sum_{i=1}^k 1_{y_j=i}(x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)^\top$.

Exercise 14.13 (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 (i.e., $x \in \{1, \dots, m\}^d$ and $y \in \{1, \dots, k\}$) and the following generative model:

$$\begin{aligned} y &\sim \text{multinomial}(\pi) \\ x | y = i &\sim \prod_{j=1}^d \text{multinomial}(x_j | \theta_{ji}), \end{aligned}$$

where $\pi \in \mathbb{R}^k$ (in the simplex), and each θ_{ji} is in the simplex in $\mathbb{R}^m$. In other words, given y, the d components $x_1, \dots, x_d$ are independent. This is called the "naive Bayes" model, often used for text documents.[5]

Using the usual one-hot encoding of discrete distributions, 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 $\prod_{a=1}^m \theta_{jia}^{x_{ja}}$.

[4]LDA is not to be confused with latent Dirichlet allocation (Blei et al., 2003), which is a generative model for collections of text documents.

[5]See https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

We can then compute

$$\begin{aligned}\mathbb{P}(y=i|x) &\propto \mathbb{P}(y=i,x) = \prod_{i=1}^{k} \pi_i^{y_i} \prod_{j=1}^{d}\prod_{a=1}^{m} \theta_{jia}^{x_{ja}y_i} \\ \log \mathbb{P}(y=i|x) &\propto \sum_{i=1}^{k} y_i \Big(\log \pi_i + \sum_{j=1}^{d}\sum_{a=1}^{m} (\log \theta_{jia}) x_{ja} \Big).\end{aligned}$$

As for LDA in section 14.2.1, 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, as for LDA, we can obtain maximum likelihood estimates for each parameter of multinomial variables using empirical proportions (proof left as an exercise).

14.2.3 Maximum Likelihood Estimations

As shown in sections 14.2.1 and 14.2.2, for LDA 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 LDA, as shown in exercise 14.12, 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 from section 14.2.1, instead of having to solve a convex problem. The key question is: Which one is better?

Discriminative versus generative learning. When making an even simpler assumption of Σ diagonal, we can study the potential benefits of the discriminative and the generative setup, following Ng and Jordan (2001): the generative approach has a stronger bias, but potentially a lower variance.

For both LDA 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 union bound, we should expect to have n larger than a constant multiplied by $\log d$ 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).

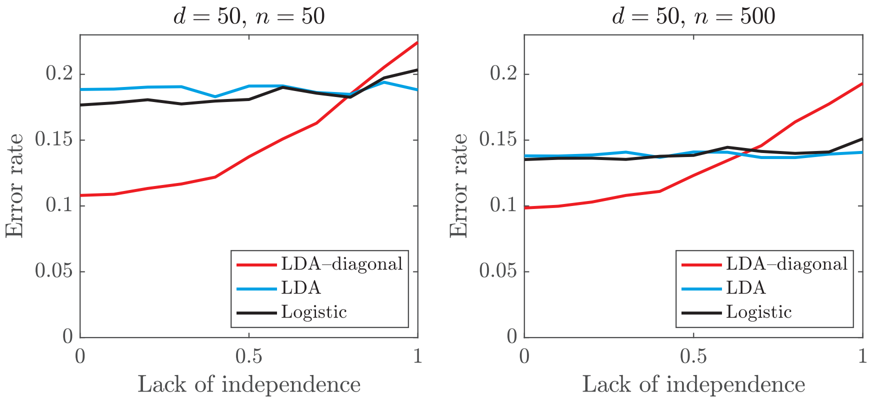


Figure 14.4. Comparison of LDA with a full covariance matrix, LDA with a diagonal covariance matrix, and logistic regression, on a well-specified binary classification problem (Gaussian class-conditional densities with same covariance matrix), with independent components and nonindependent components (with a smooth transition, which is linear in the matrix logarithm). As expected, for independent components (left parts of the plots), LDA with the independence assumptions leads to better performance; for larger n (right plot), LDA with independent components underfits when components are not independent (right part of the plot).

14.3 Bayesian Inference

For simplicity, in this section, we consider random observations $z \in \mathcal{Z}$, which 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 as $p(z|\theta)$ 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 θ is sampled once from the prior distribution and we obtain i.i.d. observations $z_1, \dots, z_n \in \mathcal{Z}$ sampled from $p(z|\theta)$.

Since observations are i.i.d., the overall joint distribution of the data and θ 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 θ given the data $(z_1, \dots, z_n)$, which is proportional to $p(z_1, \dots, z_n, \theta)$, and with the 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 MAP estimate, which is rarely used within Bayesian inference (some reasons are discussed 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 key computational task (described next).

Posterior mean. A good summary of the posterior distribution is the posterior mean $\int_\Theta \theta p(\theta|z_1, \dots, z_n) d\theta$, which 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 on z:

$$\int_\Theta p(z|\theta) p(\theta|z_1, \dots, z_n) d\theta.$$

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-Bayesian" 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 used to characterize posterior distributions. See many more details in Gelman et al. (1995) and Robert (2007).

Conjugate priors. In rare instances, the posterior distribution takes 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.[6]

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

$$\int_\Theta q(\eta) \prod_{i=1}^n p(z_i|\eta) d\eta = \int_\Theta \exp\Big[n \times \Big(\frac{1}{n} \log q(\eta) + \frac{1}{n} \sum_{i=1}^n \log p(z_i|\eta)\Big)\Big] d\eta,$$

and thus as $\int_\Theta \exp(nh(\eta)) d\eta$ 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 Θ.[7] This maximizer is exactly the MAP estimate $\hat{\theta}_{\text{MAP}}$, and the approximation is exactly equivalent to modeling the posterior density as a Gaussian with mean $\hat{\theta}_{\text{MAP}}$ and covariance matrix $\frac{1}{n} h''(\hat{\theta}_{\text{MAP}})^{-1}$.

[6] See https://en.wikipedia.org/wiki/Conjugate_prior for more details.
[7] See https://francisbach.com/laplace-method/ for details.

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), with interesting links with randomized gradient descent through Langevin diffusions (Dalalyan, 2017; Durmus and Moulines, 2017).

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 Wainwright and Jordan (2008), 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"), which themselves have their own prior distributions (often called "hyperprior distribution"). For example, using these notations, the prior distribution is $q(\theta|\lambda)$ with a hyperprior $r(\lambda)$, with often a data distribution that depends on both θ and λ.

While we could still treat λ as a random variable on which Bayesian inference is performed, it is common to perform maximum-likelihood estimation on λ, or more generally, maximum a posteriori estimation. This is sometimes called "type II maximum likelihood" or parametric[8] "empirical Bayes." This leads to a form of hyperparameter selection for λ. 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 hyperparameter selection, with many applications. We present briefly two of them next.

Selection among finitely many models. A classical application of marginal likelihood maximization is to consider m different models; that is, m distributions $p_j(z|\theta_j)$, with potentially parameters $\theta_j \in \Theta_j$ living in different spaces, with prior distributions $q_j(\theta_j)$. With a uniform distribution on the models, model selection is performed by maximizing with respect to $j \in \{1,\dots,m\}$:

$$\int_{\Theta_j} \prod_{i=1}^n p_j(z_i|\theta_j)q_j(\theta_j)d\theta_j.$$

Consider the Gaussian approximation obtained from the Laplace approximation. One can show that we obtain penalized maximum log-likelihood with a penalty equal to $\frac{d_j}{2}\log n$,

[8] Nonparametric variants can be considered, in particular using the expression of posterior means from section 14.1.4. See `https://en.wikipedia.org/wiki/Empirical_Bayes_method` for more details.

where d_j is the dimension of Θ_j, leading to the Bayesian information criterion (BIC) (see chapter 7 in Robert, 2007). See also the discussion in section 8.2.2.

Sparsity with automatic relevance determination. As mentioned in section 14.1.3, we can consider a prior distribution $q(\theta|\eta)$, which is Gaussian with mean zero and covariance matrix ηI. Maximizing the penalized marginal likelihood ends up being similar to the "η-trick" from section 8.3.1. Indeed, when we consider regression with Gaussian noise (i.e., when y given θ is normal with mean $\Phi\theta$ and covariance matrix $\sigma^2 I$), then y given η 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 nonconvex cost function to estimate η. See more details in Tipping (2001).

Gaussian processes. The linear regression 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 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 $K \in \mathbb{R}^{n\times n}$, with $K_{ij} = k(x_i, x_j)$, where k is a positive-definite kernel function (with no need to have an explicit representation as the dot product between feature vectors). This allows us to combine Bayesian inference with nonparametric kernel learning. See more details in Rasmussen and Williams (2006) and explicit connections with theoretical developments from chapter 7 in Kanagawa et al. (2018).

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 (2024) and the many references therein.

14.4.1 Setup

Here, we consider the classical supervised learning framework that we have been following throughout this 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 ρ over θ. Given this probability distribution, we can consider the following:

- A randomized predictor f_θ, where θ is sampled from ρ. Then, the generalization performance will be considered with this extra randomness (on top of the randomness of the training data).
- 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 needs to be considered. Note that in this

situation, the final prediction function is not in the set of all f_θ, $\theta \in \Theta$ and is often called an "aggregated predictor."

The generalization bounds that will be presented will be valid for *all* potential probability distributions ρ, including ones that depend on the data, which implies that we can then optimize the bounds over the distribution, leading to a candidate that is very close to the Bayesian posterior distribution (but with an added temperature; see section 14.4.2). As in Bayesian inference, we consider a fixed probability distribution q on Θ, 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 θ), 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 functional with expectation $\mathcal{R}$).

14.4.2 Uniformly Bounded Loss Functions

We assume that almost surely, for all $\theta \in \Theta$, we have $\ell(y, f_\theta(x)) \in [0, \ell_\infty]$ (e.g., with the 0–1 loss for binary classification or with bounded predictors for regression). Following the exposition of Alquier (2024) and 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 θ, 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 Θ and $D(\rho\|q)$ the Kullback-Leibler (KL) divergence between ρ and q, defined as follows (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 the Chernoff bound, [9] we obtain that with a probability greater than $1-\delta$,

$$\sup_{\rho \in \mathcal{P}(\theta)} \int_\Theta s(\mathcal{R}(\theta) - \widehat{\mathcal{R}}(\theta))d\rho(\theta) - D(\rho\|q) \leqslant \frac{s^2\ell_\infty^2}{8n} + \log\frac{1}{\delta},$$

[9] See exercise 1.10 and https://en.wikipedia.org/wiki/Chernoff_bound.

or, in other words, with probability at least $1-\delta$, 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 scaling of the bound between empirical and population quantities is of form $\frac{C}{s} + \frac{s}{n} C'$ for constants C, C', thus leading to a natural choice of $s \propto \sqrt{n}$, to obtain the traditional scaling in $O(1/\sqrt{n})$.

The bound can be empirically computed for any ρ 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, while for $s \neq n$, a different scaling is used (often referred to as the "temperature," because of the statistical physics analogy). Denoting $\hat{\rho}_s$ as this distribution, we get with a probability greater than $1-\delta$ that

$$\int_\Theta \mathcal{R}(\theta) d\hat{\rho}_s(\theta) \leqslant \inf_{\rho \in \mathcal{P}(\Theta)} \left\{ \int_\Theta \widehat{\mathcal{R}}(\theta) d\rho(\theta) + \frac{1}{s} D(\rho \| q) \right\} + \frac{1}{s} \log \frac{1}{\delta} + \frac{s\ell_\infty^2}{8n}.$$

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 equation (14.5), we can get a bound in expectation as for all $\rho \in \mathcal{P}(\theta)$ (again, ρ 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 distribution, by applying Jensen's inequality, we get

$$\mathbb{E}\Big[\int_\Theta \mathcal{R}(\theta) d\hat{\rho}_s(\theta) \Big] \leqslant \inf_{\rho \in \mathcal{P}(\Theta)} \left\{ \int_\Theta \mathcal{R}(\theta) d\rho(\theta) + \frac{1}{s} D(\rho \| q) \right\} + \frac{s\ell_\infty^2}{8n}. \tag{14.6}$$

Finite set of models. We consider m prediction functions $\hat{f}_1, \dots, \hat{f}_n$. By considering all Dirac measures in equation (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 $\ell_\infty \sqrt{\frac{\log m}{n}}$, as we obtained for empirical risk minimization in section 4.4.3.

Lipschitz-continuous losses, linear predictions, and Gaussian priors. See the tutorial from Alquier (2024) 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) and 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 ℓ_0-penalty from section 8.2.2, but it 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 posteriori (MAP) estimation may not correctly employ the knowledge of the prior distribution; (2) the use of generative models to obtain alternatives to discriminative estimators; and (3) the link with Bayesian inference.

Chapter 15

Lower Bounds

Chapter Summary

- Statistical lower bounds on the expected risk: 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 this book. The lower bounds can be obtained through information theory or Bayesian analysis.
- Optimization lower bounds on the optimization error: 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 (SGD): After t iterations, the rates proportional to $O(1/\sqrt{t})$ for convex functions and $O(1/(\mu t))$ for μ-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 nonasymptotic 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 look at lower bounds on generalization and optimization errors; 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). Obtaining such lower bounds sometimes explicitly constructs hard problems (as for optimization), and sometimes not (when they are based on information theory, such as for prediction performance).

Lower bounds will be obtained in a “minimax” setting, in which we look at the worst-case performance over the entire problem class of the best algorithm in the algorithm 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 divided into three sections: section 15.1 considers statistical lower bounds, section 15.2 considers optimization lower bounds, and 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 functions of some d-dimensional features (i.e., $f_*(x) = \langle \theta_*, \varphi(x) \rangle$) for some $\theta_* \in \mathbb{R}^d$, potentially in an ℓ_2-ball, and/or with fewer than k nonzero elements.
- Functions with all partial derivatives up to order s bounded in the L_2-norm (e.g., Sobolev spaces).

Since we are looking for lower bounds, we are free to make extra assumptions (which can only make the problem simpler) and reduce the lower bounds. For example, we will focus on Gaussian noise with constant variance σ^2 that is independent of x.

We will either consider fixed design assumptions, as studied for ordinary least-squares in chapter 3, or random designs with the simplest input distributions (which can only make the problem simpler), as studied in most of this book.

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); for example, see section 15.3 for Lipschitz-continuous loss functions.

15.1.1 Minimax Lower Bounds

We consider a set of probability distributions p_θ indexed by a parameter $\theta \in \Theta$ (that can characterize input distributions and the smoothness of the target function). We consider some data $\mathcal{D}$, generated from this distribution, most often independent and identically distributed (i.i.d.), and we denote $\mathbb{E}_\theta$ expectations with respect to data coming from the distribution indexed by θ, and $\mathbb{P}_\theta$ the associated probability measure. Note that throughout this section, the distribution of the data $\mathcal{D}$ depends on θ.

We consider an estimator $\mathcal{A}(\mathcal{D})$ of $\theta \in \Theta$, with some squared distance δ^2 between two elements of Θ, so $\delta(\theta, \theta')^2$ measures the performance of θ' when the true estimator is θ.

The testing error of $\mathcal{A}$ when the data $\mathcal{D}$ come from θ_* is defined as

$$\mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big].$$

The goal is to find an algorithm so $\sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big]$ is as small as possible, and the lower bound on testing error is thus

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big]. \tag{15.1}$$

This is often referred to as "minimax" lower bounds.

Since by Markov's inequality, $\mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big] \geqslant A\,\mathbb{P}_{\theta_*}\big(\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A\big)$, up to multiplicative constants, it is sufficient to lower-bound

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{P}_{\theta_*}\big(\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A\big) \tag{15.2}$$

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:

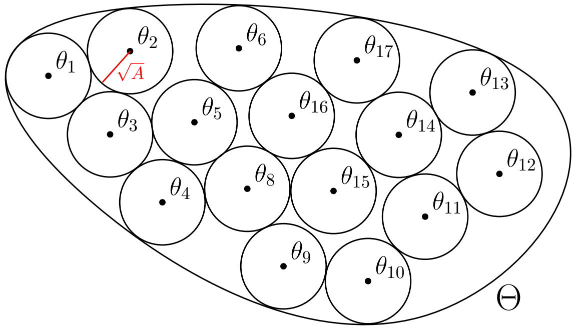
- **Reduction to a hypothesis test:** By selecting a finite subset $\{\theta_1, \dots, \theta_M\}$ of distribution parameters from Θ that is maximally spread, a good estimator leads to a good hypothesis test that can identify which θ_j (among the M possibilities) was used to generate the data. We can then use information theory to lower-bound the probability of error of such a test. This versatile technique can handle most situations, ranging from fixed to random design.
- **Bayesian analysis:** We can lower-bound the supremum for all Θ by any expectation over a distribution supported on Θ. Once we have an expectation, we can use the same decision-theoretic argument as the ones used to compute the Bayes risk in section 3.7; for instance, 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 it is 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 Θ with balls of some radius $\sqrt{A}$; that is, find $\theta_1, \dots, \theta_M \in \Theta$ such that

$$\forall i \neq j,\ \delta(\theta_i, \theta_j)^2 \geqslant 4A, \tag{15.3}$$

and transform the estimation problem into a hypothesis test; that is, an algorithm going from data $\mathcal{D}$ to one of M potential outcomes (see the following illustration in two dimensions with the Euclidean geometry).



Then, because we take the supremum over a smaller set,

$$\sup_{\theta_* \in \Theta} \mathbb{P}_{\theta_*}\big(\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A\big) \geqslant \max_{j \in \{1,\ldots,M\}} \mathbb{P}_{\theta_j}\big(\delta(\theta_j, \mathcal{A}(\mathcal{D}))^2 > A\big). \tag{15.4}$$

Any algorithm $\mathcal{A}(\mathcal{D}) \in \Theta$ gives a "test"; that is, a function $g \circ \mathcal{A} : \mathcal{D} \to \{1, \ldots, M\}$ defined as

$$g(\mathcal{A}(\mathcal{D})) = \operatorname*{arg\,min}_{j \in \{1,\ldots,M\}} \delta(\theta_j, \mathcal{A}(\mathcal{D})) \in \{1, \ldots, M\},$$

where ties are broken arbitrarily (e.g., by selecting the minimal index). Because of the packing condition in equation (15.3), the testing error of $\mathcal{A}$ can be lower-bounded by the error rate of $g \circ \mathcal{A}$ (with the 0–1 loss).

Indeed, if, for some $j \in \{1, \ldots, M\}$, $g(\mathcal{A}(\mathcal{D})) \neq j$, there is $k \neq j$ such that $\delta(\theta_k, \mathcal{A}(\mathcal{D})) < \delta(\theta_j, \mathcal{A}(\mathcal{D}))$. Moreover, using the triangle inequality for δ, we get

$$\delta(\theta_j, \theta_k)^2 \leqslant 2\big[\delta(\theta_j, \mathcal{A}(\mathcal{D}))^2 + \delta(\theta_k, \mathcal{A}(\mathcal{D}))^2\big].$$

Then,

$$\begin{aligned} \delta(\theta_j, \mathcal{A}(\mathcal{D}))^2 &\geqslant \frac{1}{2}\delta(\theta_j, \theta_k)^2 - \delta(\theta_k, \mathcal{A}(\mathcal{D}))^2 \\ &> \frac{1}{2}\delta(\theta_j, \theta_k)^2 - \delta(\theta_j, \mathcal{A}(\mathcal{D}))^2 \text{ by the choice of } k, \end{aligned}$$

which implies $\delta(\theta_j, \mathcal{A}(\mathcal{D}))^2 > \frac{1}{4}\delta(\theta_j, \theta_k)^2 \geqslant A$. Thus, we have the following inequality for the probabilities of these two events:

$$\mathbb{P}_{\theta_j}\big(\delta(\theta_j, \mathcal{A}(\mathcal{D}))^2 > A\big) \geqslant \mathbb{P}_{\theta_j}\big(g(\mathcal{A}(\mathcal{D})) \neq j\big),$$

which leads to, using equations (15.2) and (15.4),

$$\begin{aligned} \inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big] &\geqslant A \cdot \inf_h \max_{j \in \{1,\ldots,M\}} \mathbb{P}_{\theta_j}\big(h(\mathcal{D}) \neq j\big) \\ &\geqslant A \cdot \inf_h \frac{1}{M} \sum_{j=1}^{M} \mathbb{P}_{\theta_j}\big(h(\mathcal{D}) \neq j\big), \end{aligned} \tag{15.5}$$

where h is any (measurable) function from the data $\mathcal{D}$ to $\{1, \ldots, M\}$. We have thus lower-bounded the minimax statistical error by the minimax error of a hypothesis test h, which is a function that takes the data $\mathcal{D}$ to a value in $\{1, \ldots, M\}$. 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 nonnegative. Moreover, using Jensen's inequality for the logarithm (which is a concave function), we have

$$H(y) = \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \log \frac{1}{\mathbb{P}(y = y')} \leqslant \log \bigg(\sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \frac{1}{\mathbb{P}(y = y')} \bigg) = \log |\mathcal{Y}|.$$

The entropy $H(y)$ represents the uncertainty associated with the random variable y, going from $H(y) = 0$ if y is deterministic (i.e., $\mathbb{P}(y = y') = 1$ for some $y' \in \mathcal{Y}$) to $\log |\mathcal{Y}|$ when y has a uniform distribution.

⚠ Some authors use the binary logarithm, while the natural logarithm is used in this book.

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 as

$$H(x, y) = -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x', y = y') \log \mathbb{P}(x = x', y = y').$$

Using conditional distributions, it can be decomposed as

$$\begin{aligned}
H(x, y) &= -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \big[\mathbb{P}(y = y'|x = x')\mathbb{P}(x = x')\big] \\
&= -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \mathbb{P}(y = y'|x = x') \\
&\qquad\qquad -\sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \mathbb{P}(x = x') \\
&= -\sum_{x' \in \mathcal{X}} \mathbb{P}(x = x') \bigg\{ \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y'|x = x') \log \mathbb{P}(y = y'|x = x') \bigg\} \\
&\qquad\qquad -\sum_{x' \in \mathcal{X}} \mathbb{P}(y = y') \log \mathbb{P}(x = x') \\
&= \sum_{x' \in \mathcal{X}} \mathbb{P}(x = x') H(y|x = x') + H(x),
\end{aligned}$$

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 $\hat{y}$ (i.e., $H(y|\hat{y})$ is large), 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 given $\hat{y}$ through conditional and marginal entropies in 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 $\hat{y}$). Moreover, we have $H(e|\hat{y})\leqslant H(e)\leqslant\log 2$ (because $e\in\{0,1\}$), and

$$\begin{aligned}H(y|e,\hat{y}) &= \mathbb{P}(e=1)H(y|\hat{y},e=1)+\mathbb{P}(e=0)H(y|\hat{y},e=0)\\ &= \mathbb{P}(e=1)H(y|\hat{y},e=1)+0\leqslant\mathbb{P}(\hat{y}\neq y)\log|\mathcal{Y}|\end{aligned}$$

(because $e=0\Leftrightarrow y=\hat{y}$). Thus,

$$\mathbb{P}(\hat{y}\neq y)\geqslant\frac{1}{\log|\mathcal{Y}|}H(y|e,\hat{y})=\frac{1}{\log|\mathcal{Y}|}(H(y|\hat{y})-H(e|\hat{y}))\geqslant\frac{1}{\log|\mathcal{Y}|}(H(y|\hat{y})-\log 2).$$

■

Data-processing inequality. A fundamental result in information theory allows us to lower-bound conditional entropies in the presence of conditional independencies. That is, if we have three random variables x, y, and z, such that x and z are conditionally independent given y, then $H(x|z)\geqslant H(x|y)$: stated in words, the uncertainty of x given z has to be larger than the uncertainty of x given y, which is natural because the statistical dependence between x and z is occurring through y. In other words, the sequence $x\to y\to z$ 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, given 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 using Jensen's inequality for the concave function $a : t \mapsto -t \log t$, we get

$$\begin{aligned}
H(x|z) &= \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z') H(x|z = z') = \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z') \sum_{x' \in \mathcal{X}} a\big(\mathbb{P}(x{=}x'|z{=}z')\big) \\
&\geqslant \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z') \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y{=}y'|z{=}z') a\big(\mathbb{P}(x{=}x'|y{=}y')\big) \\
&= \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z') \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y'|z = z') H(x|y = y') \\
&= \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') H(x|y = y') = H(x|y).
\end{aligned}$$

This leads immediately to the full version of Fano's inequality given in proposition 15.2.

Proposition 15.2 (Fano's inequality–full version) *If the random variable y and $\hat{y}$ have values in the same finite set $\mathcal{Y}$, and if we have a Markov chain $y \to z \to \hat{y}$ for an arbitrary random variable z, then*

$$\mathbb{P}(\hat{y} \neq y) \geqslant \frac{H(y|\hat{y}) - \log 2}{\log |\mathcal{Y}|} \geqslant \frac{H(y|z) - \log 2}{\log |\mathcal{Y}|}.$$

Now, we need to look at two last concepts from information theory–namely, Kullback-Leibler (KL) divergence and mutual information, both for discrete and continuous-valued random variables.

KL divergence. Given two distributions on $\mathcal{Z}$, p and q (which are nonnegative functions on $\mathcal{Z}$ that sum to 1), 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 nonnegative by convexity of the function $t \mapsto t \log t$, and equal to zero if and only if $p = q$. It is a classical dissimilarity measure for probability distributions that is jointly convex in (p, q).[1] Note that it can also be seen as a Bregman divergence (see section 11.1.3).

Mutual information. Given two random variables x and y, then we can define their mutual information as

$$I(x, y) = H(x) - H(x|y) = H(x) + H(y) - H(x, y) = H(y) - H(y|x). \tag{15.6}$$

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

$$\begin{aligned}
I(x, y) &= H(x) + H(y) - H(x, y) \\
&= \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x', y = y') \log \frac{\mathbb{P}(x = x', y = y')}{\mathbb{P}(x = x')\mathbb{P}(y = y')}, \qquad (15.7)
\end{aligned}$$

[1] See more properties in `https://en.wikipedia.org/wiki/Kullback-Leibler_divergence`.

which can be seen as the KL divergence between the joint distribution of (x, y) and the corresponding product of marginals (it is thus nonnegative).

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 a base measure. Then, many properties (which were obtained through convex arguments) extend when z is continuous-valued, especially the data-processing inequality and Fano's inequality (see more details in Cover and Thomas, 1999).

For example, the KL divergence between two distributions can be defined as

$$D_{\mathrm{KL}}(p||q) = \mathbb{E}_p\Big[\log \frac{dp}{dq}(x)\Big],$$

where $\frac{dp}{dq}$ is the density of p with respect to q. A short calculation (left as an exercise) shows that for two Gaussian distributions of mean vectors μ_1, μ_2 and equal covariance matrices (with value Σ), 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 p_{θ_j} associated with the parameter $\theta_j \in \Theta$. We consider $\hat{y} = h(\mathcal{D})$, where h is a function with values in $\{1, \dots, M\}$. 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 equation (15.5), which provides a lower bound on error, is exactly the probability that $\hat{y} \neq y$. This is exactly what Fano's inequality from proposition 15.2 gives us, leading to corollary 15.1.

Corollary 15.1 (Fano's inequality for multiple hypothesis testing) *Given M probability distributions p_{θ_j}, $j = 1, \dots, m$, on $\mathcal{D}$, then*

$$\inf_h \frac{1}{M} \sum_{j=1}^{M} \mathbb{P}_{\theta_j}\big(h(\mathcal{D}) \neq j\big) \geqslant 1 - \frac{1}{M^2 \log M} \sum_{j,j'=1}^{M} D_{\mathrm{KL}}(p_{\theta_j}||p_{\theta_{j'}}) - \frac{\log 2}{\log M}. \tag{15.8}$$

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 from the distribution p_{θ_j}. We have, using the definition of the mutual information in equation (15.6) and the property in equation (15.7),

$$\begin{aligned} H(y|\mathcal{D}) &= H(y) - I(y, \mathcal{D}) = \log M - \frac{1}{M} \sum_{j=1}^{M} D_{\mathrm{KL}}\Big(p_{\theta_j} \Big\| \frac{1}{M} \sum_{j'=1}^{M} p_{\theta_{j'}}\Big) \\ &\geqslant \log M - \frac{1}{M^2} \sum_{j,j'=1}^{M} D_{\mathrm{KL}}(p_{\theta_j} \| p_{\theta_{j'}}) \end{aligned}$$

by the convexity of the KL divergence. We can then apply proposition 15.2 and get equation (15.8). ■

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 (with all x_i's deterministic), we get exactly the following:

$$D_{\mathrm{KL}}(p_{\theta_j}||p_{\theta_{j'}}) = \frac{1}{2\sigma^2}\sum_{i=1}^n \big[f_{\theta_j}(x_i) - f_{\theta_{j'}}(x_i)\big]^2 = \frac{n}{2\sigma^2}\delta(\theta_j, \theta_{j'})^2, \tag{15.9}$$

where $\delta(\theta, \theta')^2 = \frac{1}{n}\sum_{i=1}^n \big[f_\theta(x_i) - f_{\theta'}(x_i)\big]^2$ 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 have a common distribution p for x, then

$$\begin{aligned} D_{\mathrm{KL}}\big(p_{\theta_j}(\mathcal{D})||p_{\theta_{j'}}(\mathcal{D})\big) &= \sum_{i=1}^n D_{\mathrm{KL}}\big(p(x_i)p_{\theta_j}(y_i|x_i)||p(x_i)p_{\theta_{j'}}(y_i|x_i)\big) \\ &= \frac{n}{2\sigma^2}\int_{\mathcal{X}} \big[f_{\theta_j}(x) - f_{\theta_{j'}}(x)\big]^2 dp(x) = \frac{n}{2\sigma^2}\|f_{\theta_j} - f_{\theta_{j'}}\|^2_{L_2(p)}, \end{aligned}$$

which we define as $\frac{n}{2\sigma^2}\delta(\theta_j, \theta_{j'})^2$.

Overall, to obtain a lower bound with Gaussian noise, we need to find $\theta_1, \dots, \theta_M$ in Θ such that (for fixed designs)

- $\displaystyle\frac{1}{M^2}\sum_{j,j'=1}^M \frac{n}{2\sigma^2}\delta(\theta_j, \theta_{j'})^2 \leqslant \frac{1}{4}\log(M)$ and $\displaystyle\frac{\log 2}{\log M} \leqslant \frac{1}{4}$ (i.e., $M \geqslant 16$), so equations (15.8) and (15.9) lead to a lower bound on testing error, when equation (15.5) is applied, that is equal to $A\big(1 - \frac{1}{\log M}\cdot\frac{1}{4}\log(M) - \frac{1}{4}\big) = \frac{A}{2}$.
- $\min_{j\neq k}\delta(\theta_j, \theta_k)^2 \geqslant 4A$, so we can apply equation (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 Θ, which are all $2\sqrt{A}$ apart. There are two main tools to find such packings: (1) a direct volume argument and (2) Varshamov-Gilbert's lemma. Next, we present them before going over some examples.

Volume argument. Lemma 15.1 provides the simplest argument.

Lemma 15.1 (Packing ℓ_2-balls) *Let M be the maximal number of elements of the Euclidean ball of radius 1 in dimension d, which are at least 2ε-apart in the ℓ_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. All balls of center θ_j and radius ε 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 ball; that is, $M\varepsilon^d \leqslant (1+\varepsilon)^d$.

Since M is maximal, for any θ such that $\|\theta\|_2 \leqslant 1$, there is $j \in \{1, \dots, M\}$ 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 M balls of radius θ_j and radius 2ε. Thus, by using volumes, we get $1 \leqslant M(2\varepsilon)^d$. ■

Packing with the Varshamov-Gilbert's lemma. The maximal number of points in the hypercube $\{0,1\}^d$ that are at least $d/4$-apart in the Hamming loss (i.e., the ℓ_1-distance) is greater than $\exp(d/8)$, with a nice probabilistic argument obtained from lemma 15.2 with $\alpha = 1/2$.

Lemma 15.2 (Varshamov-Gilbert's lemma) *For any $\alpha \in (0,1)$, there is 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 ℓ_1-balls of radius $(1-\alpha)\frac{d}{2}$ includes all of $\{0,1\}^d$. Therefore, by comparing cardinalities,

$$2^d \leqslant \sum_{x \in \mathcal{B}} \Big|\Big\{y \in \{0,1\}^d, \|y - x\|_1 \leqslant (1-\alpha)\frac{d}{2}\Big\}\Big|.$$

By symmetry, the value of each $\big|\big\{y \in \{0,1\}^d, \|y - x\|_1 \leqslant (1-\alpha)\frac{d}{2}\big\}\big|$ is independent of x, and thus equal to the value for $x = 0$. Therefore, we get

$$2^d \leqslant |\mathcal{B}| \cdot \Big|\Big\{y \in \{0,1\}^d, \|y\|_1 \leqslant (1-\alpha)\frac{d}{2}\Big\}\Big| = |\mathcal{B}| \cdot \Big|\Big\{y \in \{0,1\}^d, \sum_{i=1}^d y_i \leqslant (1-\alpha)\frac{d}{2}\Big\}\Big|.$$

We can now estimate this cardinality by considering a random variable z, which is binomial with parameters d and $1/2$ (i.e., the sum of d independent uniform Bernoulli random variables). Then,

$$2^{-d}\Big|\Big\{y \in \{0,1\}^d, \sum_{i=1}^d y_i \leqslant (1-\alpha)\frac{d}{2}\Big\}\Big| = \mathbb{P}\Big(z \leqslant (1-\alpha)\frac{d}{2}\Big) = \mathbb{P}\Big(z \geqslant (1+\alpha)\frac{d}{2}\Big).$$

The probability $\mathbb{P}(z \geqslant (1+\alpha)\frac{d}{2})$ equals $\mathbb{P}(\frac{z}{d} - \frac{1}{d}\mathbb{E}[z] \geqslant \frac{\alpha}{2})$. Thus, using Hoeffding's inequality (proposition 1.2), it is less than $\exp(-2d(\alpha/2)^2) = \exp(-d\alpha^2/2)$. This leads to the desired result. ■

15.1.5 Examples

Fixed-design linear regression. We consider linear regression with $\Phi \in \mathbb{R}^{n \times d}$ being a design matrix with isotropic noncentered empirical covariance matrix $\frac{1}{n}\Phi^\top\Phi = I$ (which imposes $n \geqslant d$). We consider the ball $\Theta = \{\theta \in \mathbb{R}^d, \|\theta\|_2 \leqslant D\}$, with D to be set later. Moreover, we are in the situation where $\delta(\theta,\theta')^2 = \frac{1}{n}\sum_{i=1}^n \big[f_\theta(x_i) - f_{\theta'}(x_i)\big]^2 = \frac{1}{n}\|\Phi\theta - \Phi\theta'\|_2^2 = \|\theta-\theta'\|_2^2$.

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 we also define $\theta_i = \beta(2x_i - 1_d) \in \{-\beta,\beta\}$ for each $i \in \{1,\dots,M\}$. Thus, $\|\theta_i\|_2^2 = \beta^2 d$, and, for $i \neq j$,

$$\|\theta_i - \theta_j\|_2^2 \leqslant 4\beta^2 d \leqslant 32\beta^2\log(M) \text{ and } \|\theta_i-\theta_j\|_2^2 \geqslant (2\beta)^2\frac{d}{4} = \beta^2 d.$$

We thus need $\beta^2 d \leqslant D^2$ (so each $\theta_i \in \Theta$) and $32\beta^2\log(M)\frac{n}{2\sigma^2} \leqslant \frac{\log M}{4}$ (i.e., $64\beta^2\frac{n}{\sigma^2} \leqslant 1$), so $\frac{1}{M^2}\sum_{j,j'=1}^M \frac{n}{2\sigma^2}\delta(\theta_j,\theta_{j'})^2 \leqslant \frac{1}{4}\log(M)$. Thus, we can use $A = \beta^2 d/4$ (packing constraint), and the optimal rate is greater than $A/2 = \beta^2 d/8$. By choosing the largest possible β, the lower bound on (excess) testing error is then greater than

$$\frac{1}{8}\min\Big\{D^2, \frac{\sigma^2 d}{64n}\Big\}. \tag{15.10}$$

Therefore, when $D^2 \geqslant \frac{\sigma^2 d}{64n}$, we get a lower bound of $\frac{\sigma^2 d}{512n}$, which is, up to a multiplicative constant, 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 using 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 for fixed-design linear regression.*

Random design linear regression. We consider the same model as before, but with x_i, $i = 1,\dots,n$, sampled i.i.d. from a fixed distribution such that $\mathbb{E}[\varphi(x)\varphi(x)^\top] = I$, so $\delta(\theta,\theta')^2 = \|f_\theta - f_{\theta'}\|_{L_2(p)}^2 = \|\theta-\theta'\|_2^2$, which is the same distance as for fixed design. Thus, the result for fixed design regression also applies to the random design setting (i.e., a constant times $\sigma^2 d/n$).

Nonparametric 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 is a compact positive semidefinite self-adjoint operator T on $L_2(p)$ such that $\langle\theta, T^{-1}\theta\rangle_{L_2(p)} \leqslant D^2$ (which implies that θ is in the range of $T^{1/2}$). We denote by $(\lambda_m)_{m\geqslant 1}$ the nonincreasing sequence of positive eigenvalues of T, with the associated orthonormal eigenvectors ψ_m in $L_2(p)$.

We consider a certain integer K and $M \geqslant \exp(K/8)$ elements $x_1,\dots,x_M$ of $\{0,1\}^K$ obtained from lemma 15.2. We define $\theta_i = \beta\sum_{m=1}^K (2(x_i)_m - 1)\psi_m$. We then have

$\langle \theta, T^{-1}\theta\rangle_{L_2(p)} = \beta^2 \sum_{m=1}^K \lambda_m^{-1} \leqslant K\beta^2\lambda_K^{-1}$, and, for $i \neq j$,

$$\|\theta_i - \theta_j\|_{L_2(p)}^2 \leqslant 4\beta^2 K \leqslant 32\beta^2 \log(M) \text{ and } \|\theta_i - \theta_j\|_{L_2(p)}^2 \geqslant (2\beta)^2 (K/4) = \beta^2 K.$$

We thus need $\beta^2 K \leqslant D^2 \lambda_K$ (to be in the desired ellipsoid) 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, choosing $A = \beta^2 K/4$, 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 on compact subsets of $\mathbb{R}^d$ with piecewise smooth boundaries. The corresponding norm is the sum of squared norms of all derivatives of order s or less (Adams and Fournier, 2003). As described in section 7.6.6, it can then be shown that it corresponds to an operator T for which $\lambda_K \geqslant C \cdot K^{-\alpha}$, with $\alpha = 2s/d$, for a constant C. The lower bound then 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 a 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 bound matches the upper bound obtained via kernel ridge regression in chapter 7. It turns out that the lower bound on the minimax rate for the estimation of Lipschitz-continuous functions is the same as for $s = 1$ (see section 2.6 in Tsybakov, 2008).

15.1.6 Minimax Lower Bounds through Bayesian Analysis

We can use a Bayesian analysis as outlined for least-squares regression in section 3.7. We consider a particular probability distribution $q(\theta_*)$ on parameters θ_*, whose support is included in Θ. Then we have, since the supremum is greater than the expectation,

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big] \geqslant \inf_{\mathcal{A}} \mathbb{E}_{q(\theta_*)} \mathbb{E}_{\theta_*}\big[\delta(\theta_*, \mathcal{A}(\mathcal{D}))^2\big].$$

This reasoning is particularly simple when the optimal algorithm $\mathcal{A}$ is easy to estimate (with no need for a packing argument). In particular, this is the case when d is a Euclidean norm, as then $\mathcal{A}_*(\mathcal{D}) = \mathbb{E}\big[\theta_* | \mathcal{D}\big]$. If the prior $q(\theta_*)$ and the likelihood $p_{\theta_*}(\mathcal{D})$ 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 Θ because Θ was unbounded. When dealing with bounded balls, we need to use different distributions, as used originally by Donoho and Johnstone (1994).

Least-squares regression on a Euclidean ball. As in section 15.1.5, we consider linear regression with a fixed design (with a bound $\|\theta_*\|_2 \leqslant D$) and $\frac{1}{n}\Phi^\top\Phi = I$ (which impose $n \geqslant d$). By rotational invariance of the Gaussian distribution of the noise variable ε, we can assume that the first d rows of Φ equal $\sqrt{n}I$ and the rest of the rows equal zero. Thus, we can assume that the model is $y = \theta_* + \frac{1}{\sqrt{n}}\varepsilon$, where $\varepsilon \in \mathbb{R}^d$ with Gaussian distribution with mean zero and covariance $\sigma^2 I$, and $y \in \mathbb{R}^d$ (the $n-d$ extra observations do not bring any information on θ_*).

We then consider a prior distribution on θ_* as $\theta_* = \beta x$, where $x \in \{-1,1\}^d$ are independent Rademacher random variables (thus θ_* is unifom on its support). We need $\beta^2 d \leqslant D^2$ to be in the correct set $\Theta = \{\theta \in \mathbb{R}^d,\ \|\theta\|_2 \leqslant D\}$. We then need to compute $\mathbb{E}[\theta_*|y]$. The posterior probability of θ_* is supported on $\{-\beta,\beta\}^n$. Moreover, given the independence by component, we can treat each one separately. Then, by keeping only terms that depend on the posterior value, we get, using that $y_i|(\theta_*)_i$ is Gaussian with mean $(\theta_*)_i$ and variance σ^2/n (and θ_* uniform on its support),

$$\mathbb{P}((\theta_*)_i = \pm\beta|y_i) \propto \exp\Big(-\frac{n}{2\sigma^2}(y_i - \pm\beta)^2\Big) \propto \exp\Big(\pm\frac{n}{\sigma^2}y_i\beta\Big).$$

Thus,

$$\mathbb{E}\big[(\theta_*)_i|y_i\big] = \beta\frac{\exp(\frac{n}{\sigma^2}y_i\beta) - \exp(\frac{-n}{\sigma^2}y_i\beta)}{\exp(\frac{n}{\sigma^2}y_i\beta) + \exp(\frac{-n}{\sigma^2}y_i\beta)} = \beta\frac{1-\exp(\frac{-2n}{\sigma^2}y_i\beta)}{1+\exp(\frac{-2n}{\sigma^2}y_i\beta)} = \beta\Big[2\,\mathrm{sigmoid}\Big(\frac{2n}{\sigma^2}y_i\beta\Big) - 1\Big],$$

where $\mathrm{sigmoid}(\alpha) = 1/(1+\exp(-\alpha))$.

The posterior variance for the ith component is equal to, using that $\mathbb{P}((\theta_*)_i = \beta) = \mathbb{P}((\theta_*)_i = -\beta) = 1/2$,

$$\begin{aligned}\mathbb{E}\big[\big((\theta_*)_i - \mathbb{E}\big[(\theta_*)_i|y_i\big]\big)^2\big] &= \frac{1}{2}\mathbb{E}_{\varepsilon_i}\Big(\beta - \beta\Big[2\,\mathrm{sigmoid}\Big(2\frac{n}{\sigma^2}\beta(\beta + \varepsilon_i/\sqrt{n})\Big) - 1\Big]\Big)^2 \\ &\quad + \frac{1}{2}\mathbb{E}_{\varepsilon_i}\Big(-\beta - \beta\Big[2\,\mathrm{sigmoid}\Big(2\frac{n}{\sigma^2}\beta(-\beta + \varepsilon_i/\sqrt{n})\Big) - 1\Big]\Big)^2.\end{aligned}$$

Using that $-\varepsilon_i$ and ε_i have the same distribution, and $\mathrm{sigmoid}(-\alpha) = 1 - \mathrm{sigmoid}(\alpha)$, we get

$$\begin{aligned}\mathbb{E}\big[\big((\theta_*)_i - \mathbb{E}\big[(\theta_*)_i|y_i\big]\big)^2\big] &= 4\beta^2\mathbb{E}_{\varepsilon_i\sim\mathcal{N}(0,\sigma^2)}\Big[\Big(\mathrm{sigmoid}\Big(-2\frac{n}{\sigma^2}\beta^2 + 2\frac{\sqrt{n}}{\sigma^2}\beta\varepsilon_i)\Big)^2\Big] \\ &= 4\beta^2\mathbb{E}_{\tilde{\varepsilon}_i\sim\mathcal{N}(0,1)}\Big[\Big(\mathrm{sigmoid}(-2\frac{n}{\sigma^2}\beta^2 + 2\frac{\beta\sqrt{n}}{\sigma}\tilde{\varepsilon}_i)\Big)^2\Big].\end{aligned}$$

Now we consider the even function $\psi : \alpha \mapsto \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,1)}\big[\big(\mathrm{sigmoid}(-2\alpha^2 + 2\alpha\varepsilon)\big)^2\big]$. We have $\psi(0) = \mathrm{sigmoid}(0)^2 = 1/4$ and $\psi(\alpha) \to 0$ when $\alpha \to +\infty$. Moreover, for $\alpha > 0$, since the sigmoid function is greater than $1/2$ for positive numbers, $\psi(\alpha) \geqslant \frac{1}{4}\mathbb{P}_{\varepsilon\sim\mathcal{N}(0,1)}(\varepsilon > \alpha)$. We can then use a simple lower bound on Gaussian tails $\mathbb{P}_{\varepsilon\sim\mathcal{N}(0,1)}(\varepsilon > \alpha) \geqslant \frac{1}{4}\exp(-\alpha^2)$ (see exercise 1.12) to obtain $\psi(\alpha) \geqslant \frac{1}{16}\exp(-\alpha^2)$.

Thus, the total posterior variance $\mathbb{E}\big[\big\|\theta_* - \mathbb{E}\big[\theta_*|y\big]\big\|_2^2\big]$ is greater than

$$d \cdot 4\beta^2 \psi\Big(\frac{\beta\sqrt{n}}{\sigma}\Big) \geqslant \frac{\beta^2 d}{4} \exp(-n\beta^2/\sigma^2).$$

By choosing $\beta^2 = \min\big\{\frac{\sigma^2}{n}, \frac{D^2}{d}\big\} \leqslant \frac{\sigma^2}{n}$, the lower bound becomes

$$\frac{\beta^2 d}{4} \exp(-n\beta^2/\sigma^2) \geqslant \frac{\beta^2 d}{4} \exp(-1) \geqslant \frac{1}{12} \min\Big\{\frac{\sigma^2 d}{n}, D^2\Big\},$$

which leads to the same bound as in section 15.1.5, but with a more direct argument.

Sparse case (♦). To deal with the sparse case, we could consider a prior on θ_* that only selects k nonzero elements out of d, and perform an analysis based on the posterior probability of θ_*. 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 for each of the k blocks by selecting one of the d/k variables uniformly at random and setting its value to β. In contrast, all others are set to zero.

To compute the posterior probability of θ_*, 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 ith one, which is proportional to (keeping only the terms that depend on i),

$$\exp\Big(-\frac{n}{2\sigma^2}(y_i-\beta)^2\Big)\prod_{j\neq i}\exp\Big(-\frac{n}{2\sigma^2}(y_j)^2\Big) \propto \exp(n\beta y_i/\sigma^2).$$

The conditional expectation of θ_* then equals

$$\mathbb{E}[(\theta_*)_i|y] = \beta\frac{\exp(n\beta y_i/\sigma^2)}{\sum_{j=1}^{d/k}\exp(n\beta y_j/\sigma^2)}.$$

To compute the posterior variance, we need to sample from the prior θ_*. By symmetry, we may consider that $(\theta_*)_1 = \beta$. If $y_1 \leqslant \max_{j\neq 1} y_j$, then

$$\mathbb{E}[(\theta_*)_1|y] = \beta\frac{\exp(n\beta y_1/\sigma^2)}{\sum_{j=1}^{d/k}\exp(n\beta y_j/\sigma^2)} \leqslant \beta\frac{\exp(n\beta y_1/\sigma^2)}{\exp(n\beta y_1/\sigma^2)+\exp(n\beta\max_{j\neq 1} y_j/\sigma^2)} \leqslant \beta/2,$$

and then the risk is at least $(\beta - \mathbb{E}[(\theta_*)_1|y])^2 \geqslant \beta^2/4$.

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 β is greater than $1/2$. Moreover, by independence of all y_j, $j \neq 1$,

$$\mathbb{P}\big(\{\beta \leqslant \max_{j\neq 1} y_j\}\big) \geqslant 1 - \big(1 - \mathbb{P}_{t\sim\mathcal{N}(0,1)}(t \geqslant \beta\sqrt{n}/\sigma)\big)^{\frac{d}{k}-1}.$$

Thus, combining the contributions of all k blocks, the lower bound is greater than

$$k \cdot \frac{\beta^2}{4} \cdot \frac{1}{2}\Big[1 - \big(1 - \mathbb{P}_{t\sim\mathcal{N}(0,1)}(t \geqslant \beta\sqrt{n}/\sigma)\big)^{\frac{d}{k}-1}\Big] \geqslant k\frac{\beta^2}{16}\Big[1 - \Big(1 - \frac{1}{2}\exp(-\beta^2 n/\sigma^2)\Big)^{\frac{d}{k}-1}\Big],$$

using the Gaussian tail bound $\mathbb{P}_{t\sim\mathcal{N}(0,1)}(t \geqslant z) \geqslant \frac{1}{4}\exp(-z^2)$. We can then consider $\beta^2 = \frac{\sigma^2}{n}\log(d/k)$, leading to the lower bound

$$\frac{\sigma^2 k}{16n}\log(d/k)\big[(1 - (1 - \tfrac{1}{2}(k/d))^{\frac{d}{k}-1}\big].$$

Assuming that $d \geqslant 2k$, and using that for any $u \in [0, 1/2]$, $1-(1-u/2)^{-1+1/u} \geqslant 1/4$, we obtain the lower bound $\frac{\sigma^2 k}{64n}\log(d/k)$, which is the same, up to a multiplicative constant, as the upper bound for ℓ_0-penalty-based methods in chapter 8.

15.2 Optimization Lower Bounds

In this section, we consider ways of obtaining lower bounds for optimization algorithms corresponding to upper bounds derived in chapter 5 for gradient-based algorithms. While the statistical lower bounds from section 15.1 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 to do 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 section 2.1.2 in Nesterov (2018) and section 3.5 in Bubeck (2015), 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 subgradients of F at the previously observed points.

The key is finding functions with the proper regularity properties, for which we know that a few iterations provably lead to suboptimal performance. These functions will only reveal one new variable at each iteration and, after k iterations, can achieve only the minimum on the first k variables.

Nonsmooth functions. We consider the following function, which will be designed for 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 η, μ positive parameters that will be set later in this discussion.

The subdifferential of $F(\theta)$ is equal to

$$\mu\theta + \eta \cdot \mathrm{hull}\big(\{e_i,\ \theta_i = \max_{i' \in \{1,\dots,k+1\}} \theta_{i'}\}\big),$$

which is bounded in the ℓ_2-norm on the ball of radius D, by $\mu D + \eta$ (here, e_i denotes the ith basis vector). We consider the oracle where the output gradient is $\mu\theta + \eta e_i$, where i is the smallest index within the maximizers of $\theta_{i'}$.

Starting from $\theta_0 = 0$, θ_1 is supported on the first variable, and by recursion, after $k \leqslant d$ steps of subgradient descent, θ_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 κ that minimizes $\eta\kappa + \frac{(k+1)\mu}{2}\kappa^2$, so $\kappa = -\frac{\eta}{\mu(k+1)}$, and thus $\theta_* = -\frac{\eta}{\mu(k+1)} \sum_{i=1}^{k+1} e_i$, with value $F(\theta_*) = -\frac{\eta^2}{2\mu(k+1)}$. Therefore,

$$F(\theta_k) - F(\theta_*) \geqslant 0 - F(\theta_*) = \frac{\eta^2}{2\mu(k+1)},$$

with $\|\theta_*\|_2^2 = \frac{\eta^2}{\mu^2(k+1)}$.

To obtain a B-Lipschitz-continuous function on a ball with center 0 and radius D, we can take $\eta = B/2$ and $\mu = B/(2D)$, and we get a lower bound of $\frac{B^2}{8\mu(k+1)}$, which is valid so long as $k < d$ and matches the upper bound in proposition 5.8.

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 so long as $k < d$ and matches the upper bound in proposition 5.6.

⚠ The lower bounds are valid for $k < d$ only because there are 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 require that the gradient for iterates supported on the first i components be supported on the first $i+1$ components, so the kth iterate starting from 0 only has its first k coordinates that can be nonzero. We consider the example from section 2.1.2 in Nesterov (2018) and highlight the main arguments without proof:

$$F_k(\theta) = \frac{L}{4}\bigg\{\frac{1}{2}\bigg[\theta_1^2 + \theta_k^2 + \sum_{i=1}^{k-1}(\theta_i - \theta_{i+1})^2\bigg] - \theta_1\bigg\}.$$

The function F_k is convex and smooth, with a smoothness constant that is 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 θ is supported on the first i components for $i < k$, then $F_k'(\theta)$ is supported on the first $i+1$ components. Thus, the ith 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

$$\frac{F_{2k+1}(\theta_k) - F_{2k+1}^*}{\|\theta_0 - \theta_*\|_2^2} \geqslant \frac{L}{8}\frac{\frac{-k}{k+1} - \frac{-2k-1}{2k+2}}{\frac{2k+2}{3}} \geqslant \frac{3L}{32}\frac{1}{(k+1)^2}.$$

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 section 2.1.4 in Nesterov (2018), we consider a function defined on the space ℓ_2 of square-summable sequences as

$$F(\theta) = \frac{L-\mu}{4}\left\{\frac{1}{2}\left[\theta_1^2 + \sum_{i=1}^{\infty}(\theta_i - \theta_{i+1})^2\right] - \theta_1\right\} + \frac{\mu}{2}\|\theta\|_2^2.$$

This function is L-smooth and μ-strongly convex. Its global minimizer is θ_* such that

$$(\theta_*)_k = \left(\frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}\right)^k = q^k,$$

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 $F(\theta_k) - F_* \geqslant \frac{\mu}{2}\|\theta_k - \theta_*\|_2^2 \geqslant q^{2k}\|\theta_0 - \theta_*\|_2^2$, which shows a lower bound which is exponentially convergent with the same rate as accelerated gradient descent presented in section 5.2.5.

15.2.2 Nonconvex Optimization (♦)

While upper and lower bounds can behave well with respect to dimension in the convex case, this is not the case when the convexity assumption is removed. 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 that are not exponential in dimension.

⚠ This does not mean that all problem instances will require exponential time, but that in the worst case, there will always be a bad function for any algorithm.

We consider minimizing a function F on a bounded subset Θ of $\mathbb{R}^d$ based only on function evaluations, a problem often referred to as “zeroth-order optimization” or “derivative-free optimization" (see the 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, the main reason being that without convexity, local information does not lead to global properties. While we focus on access to function values in this section, no further access to higher-order derivatives (if they exist) can improve the exponential dependence on dimension (see exercise 15.2).

Clearly, algorithms for nonconvex zeroth-order optimization 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, and other elements). See, for instance, Snoek et al. (2012) for applications in this area.

We will assume some regularity for the functions that we want to minimize, typically bounded derivatives. We will thus assume that $f \in \mathcal{F}$ for a space $\mathcal{F}$ of functions from Θ to $\mathbb{R}$. We will take a worst-case approach, in which 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.

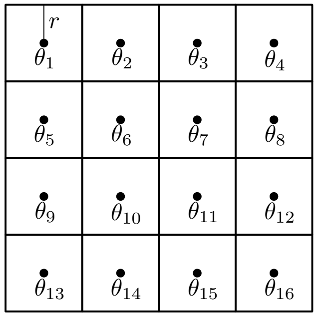
Algorithm $\mathcal{A}$ will be characterized by (1) the choice of points $\theta_1, \ldots, \theta_n \in \Theta$ to query the function, and (2) the algorithm to output a candidate $\hat{\theta} \in \Theta$ such that $F(\hat{\theta}) - \inf_{\theta \in \Theta} F(\theta)$ is small. The estimate $\hat{\theta}$ can only depend on $(\theta_i, F(\theta_i))$ for $i \in \{1, \ldots, n\}$. In this section, the choice of points $\theta_1, \ldots, \theta_n$ is made once (without seeing any function values).[2]

Given a selection of points and 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 that we denote as $\varepsilon_n^{\mathrm{opt}}$ and that we aim to characterize.

Direct lower/upper bounds for Lipschitz-continuous functions. The argument is particularly simple for a bounded metric space Θ with distance δ, and $\mathcal{F}$ being the class of L-Lipschitz-continuous functions; that is, 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.

As in section 4.4.4, we will need to cover set Θ with balls of a given radius. The minimal radius r of a cover of Θ by n balls of radius r is denoted as $r_n(\Theta, \delta)$. This corresponds to n ball centers $\theta_1, \ldots, \theta_n$. See the following example for the unit cube $\Theta = [0,1]^2$ and the metric obtained from the ℓ_∞-norm, with $n = 16$, and $r_n([0,1]^2, \ell_\infty) = 1/8$:

[2] It turns out that going *adaptive*, where the point θ_{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) (Novak, 2006).



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 dth 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).

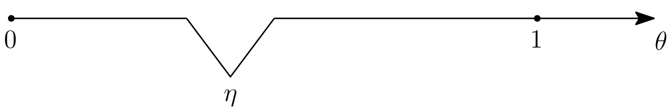
Naive algorithm. Given the ball centers $\theta_1, \dots, \theta_n$, outputting the minimum of function values $F(\theta_i)$ for $i = 1, \dots, n$ leads to an error that is less than $L r_n(\Theta, \delta)$, as the optimal $\theta_* \in \Theta$ is at most at distance $r_n(\Theta, \delta)$ from one of the cluster centers (let's say θ_k), and thus $F(\theta_k) - F(\theta_*) \leqslant L\delta(\theta_k, \theta_*) \leqslant L r_n(\Theta, \delta)$. This provides an upper bound on $\varepsilon_n^{\text{opt}}$. The algorithm that 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 zero). We now simply need to construct a function $F \in \mathcal{F}$ such that F is zero at these points but smaller than zero as much as possible at a different point.

Given the $n+1$ points defined above, there is at least a point $\eta \in \Theta$ that is at a distance of at most $r_{n+1}(\Theta, \delta)$ from all of them (otherwise, we obtain a cover of Θ with $n+1$ points). We can then construct the function

$$F(\theta) = -L\big(r_{n+1}(\Theta, \delta) - \delta(\theta, \eta)\big)_+ = -L \max\big\{r_{n+1}(\Theta, \delta) - \delta(\theta, \eta), 0\big\},$$

which is L-Lipschitz-continuous, equal to zero on all points of the algorithm and the output point $\hat{\theta}$, and with the minimum value $-L r_{n+1}(\Theta, \delta)$ attained at η. Thus, we must have $\varepsilon_n^{\text{opt}} \geqslant 0 - (-L r_{n+1}(\Theta, \delta)) = L r_{n+1}(\Theta, \delta)$. This difficult function is plotted here in one dimension:



Thus, the optimization error of any algorithm from n function values has to be worse than $L r_{n+1}(\Theta, \delta)$. Thus, so far, we have shown that

$$L r_{n+1}(\Theta, \delta) \leqslant \varepsilon_n^{\text{opt}} \leqslant L r_n(\Theta, \delta).$$

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 $n+1$ 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; it comes from the equivalence in terms of worst-case guarantees between optimization and uniform approximation (Novak, 2006).

Exercise 15.2 *Consider the space of differentiable L-Lipschitz-continuous functions on $\Theta \subset \mathbb{R}^d$ and a first-order oracle that outputs both function values and gradients at the query points. Modify the argument proposed in this section to show that the lower bound of optimization error is still of order $n^{-1/d}$.*

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 Θ, and selecting the points with the smallest function value $F(\theta_i)$. The optimization error 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 15.3). 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 use the boundedness of higher-order derivatives (as for target functions 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, leading to a dependence in $n^{-s/d}$ when sth-order derivatives are assumed bounded.[3] See also exercise 15.4 for functions with Lipschitz-continuous gradients.

Exercise 15.3 (♦) *Consider sampling independently and uniformly n points $\theta_1, \dots, \theta_n$ in $\Theta \subset \mathbb{R}^d$.*

(a) For a given L-Lipschitz-continuous function F, show that the worst-case optimization error of outputting the lower function value is less than $L \max_{\theta \in \Theta} \min_{i \in \{1,\dots,n\}} \delta(\theta, \theta_i)$.

(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 optimization error proportional to $L\big(\frac{\log n}{n}\big)^{1/d}$ with probability greater than $1 - \frac{\log n}{n}$.

Exercise 15.4 (♦) *Consider the space of differentiable functions with gradients that are L-Lipschitz-continuous on $\Theta \subset \mathbb{R}^d$ and a zeroth-order query oracle. Modify the argument proposed in this section to show that the lower bound of optimization error is of order $n^{-2/d}$.*

[3] See https://francisbach.com/optimization-is-as-hard-as-approximation/ for more details, as well as Novak (2006).

15.3 Lower Bounds for Stochastic Gradient Descent (♦)

In this section, our goal is to show that the convergence rates for SGD shown in section 5.4 are "optimal," in a sense that will be made precise. We consider a class $\mathcal{F}$ of functions, here the convex B-Lipschitz-continuous functions on the ball with center zero and radius D (for the Euclidean norm). We consider a class $\mathcal{A}$ of algorithms that can sequentially access independent random, unbiased estimates of the gradients of a function F in $\mathcal{F}$, with a squared norm bounded by B^2. We denote $A_t(F) \in \mathbb{R}^d$ as the output of algorithm A after t iterations on function F. Our goal is to find upper and lower bounds of

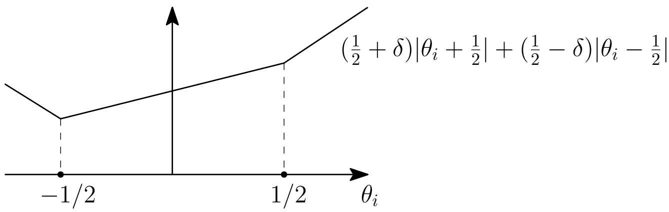
$$\varepsilon_t(\mathcal{A}, \mathcal{F}) = \inf_{A \in \mathcal{A}} \sup_{F \in \mathcal{F}} \mathbb{E}\Big[F(A_t(F)) - \inf_{\|\theta\|_2 \leqslant D} F(\theta)\Big].$$

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 at most 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

$$F_\alpha(\theta) = \frac{B}{2d} \sum_{i=1}^{d} \Big\{ \Big(\frac{1}{2} + \alpha_i \delta\Big) \cdot \Big|\theta_i + \frac{1}{2}\Big| + \Big(\frac{1}{2} - \alpha_i \delta\Big) \cdot \Big|\theta_i - \frac{1}{2}\Big| \Big\}, \tag{15.11}$$

with $\alpha \in \{-1, 1\}^d$ as a well-chosen vector, $\delta \in (0, 1/4]$, and $B > 0$. One element of the sum is plotted here:



Function F_α is convex and Lipschitz-continuous with gradients bounded in the L_2-norm by $B/(2\sqrt{d})$. Moreover, the global minimizer of F_α is $\theta = -\frac{\alpha}{2}$, with an optimal value equal to $F_\alpha^* = \frac{B}{4}(1 - 2\delta)$. That is, minimizing F_α on $[-1/2, 1/2]^d$ exactly corresponds to finding an element of the hypercube α. Moreover, it turns out that minimizing it approximately also leads to identifying α among a set of α's which are sufficiently different, as shown in the following lemma.

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 $\theta \in \mathbb{R}^d$,

$$\begin{aligned} F_\beta(\theta) + F_\alpha(\theta) - F_\beta^* - F_\alpha^* &\geqslant \frac{B}{2d} \sum_{i,\ \alpha_i \neq \beta_i} \Big\{ \Big|\theta_i + \frac{1}{2}\Big| + \Big|\theta_i - \frac{1}{2}\Big| + 2\delta - 1 \Big\} \\ &\geqslant \frac{B}{2d} \sum_{i,\ \alpha_i \neq \beta_i} \{2\delta\} = \frac{B\delta}{2d} \|\alpha - \beta\|_1. \end{aligned}$$ ■

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 ε exactly identifies which of the functions $F_{\alpha^{(i)}}$ is 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 (each α will lead to a distribution of stochastic gradients; we denote as $\mathbb{E}_\alpha$ and $\mathbb{P}_\alpha$ the associated expectation and measure),

$$\sup_{\alpha \in \mathcal{A}} \ \mathbb{E}_\alpha\big[F_\alpha(\hat\theta) - F_\alpha^*\big] \geqslant \varepsilon \cdot \sup_{\alpha \in \mathcal{A}} \ \mathbb{P}_\alpha\big(F_\alpha(\hat\theta) - F_\alpha^* > \varepsilon\big) \geqslant \varepsilon \cdot \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \mathbb{P}_\alpha\big(F_\alpha(\hat\theta) - F_\alpha^* > \varepsilon\big).$$

From an estimate $\hat\theta$, we can build a test $g(\hat\theta) \in \mathcal{A}$ by selecting the $\alpha \in \mathcal{A}$ (which is unique if $\varepsilon < \frac{B\delta}{8}$) such that $F_\alpha(\hat\theta) - F_\alpha^* \leqslant \varepsilon$ if it exists, and uniformly at random in $\mathcal{A}$ otherwise. Therefore, the minimax generalization error is greater than ε multiplied by 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 $b \in \{0,1\}$ with parameter $\frac{1}{2} + \alpha_i \delta$.

(3) Consider $\hat F(\theta) = b\big|\theta_i + \frac{1}{2}\big| + (1-b)\big|\theta_i - \frac{1}{2}\big|$, with zero gradient components except

$$\hat F_\alpha'(\theta)_i = \frac{B}{2}\big[b \,\mathrm{sign}(\theta_i + 1/2) + (1-b)\,\mathrm{sign}(\theta_i - 1/2)\big].$$

The stochastic gradients have an ℓ_2-norm bounded by B and are unbiased. Moreover, observation of the gradient for $\theta \in [-1/2, 1/2]^d$ reveals the outcome of the Bernoulli random variable b.

Therefore, after t steps, we can apply Fano's inequality (corollary 15.1) to the following setup: the random variable $\alpha \in \mathcal{A}$ is uniform, and given α, 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 α_i.

We then need to upper-bound the mutual information between α 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

$$I(\alpha, (i,b)) \ = \ I(\alpha, i) + I(\alpha, b|i) = 0 + \mathbb{E}_i \mathbb{E}_\alpha\big[D_{\mathrm{KL}}(p(b|i,\alpha)||p(b|i))\big],$$

where $p(b|i,\alpha)$ and $p(b|i)$ denote the probability distributions of b given i,α and given i. 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 $b|i,\alpha$ is a Bernoulli random variable with parameter $\frac{1}{2}+\delta$ or $\frac{1}{2}-\delta$, these KL divergences are bounded by the KL divergence 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

$$\varepsilon\Big(1-\frac{16t\delta^2+\log 2}{\log M}\Big) \geqslant \varepsilon\Big(1-\frac{16t\delta^2+\log 2}{d/8}\Big).$$

We assume $d \geqslant 32\log 2$, and $t \geqslant d/16$, and take $\delta = \frac{1}{16}\sqrt{d/t} \in [0,1/4]$, with $\varepsilon = \frac{B\delta}{16}$ (so that ε-optimality leads to identification of $\alpha \in \mathcal{A}$ from lemma 15.3). Then $\varepsilon_t(\mathcal{A},\mathcal{F})$ has the following lower bound:

$$\varepsilon_t(\mathcal{A},\mathcal{F}) \geqslant \varepsilon\Big(1-\frac{16t\delta^2+\log 2}{d/8}\Big) \geqslant \varepsilon\Big(1-\frac{1}{2}-\frac{1}{4}\Big) \geqslant \frac{B\delta}{64} = \frac{1}{1024}\frac{B\sqrt{d}}{\sqrt{t}} = \frac{1}{1024}\frac{BD}{\sqrt{t}},$$

where D is the diameter of the set of θ. The lower bound is thus, up to a multiplicative constant, the same as the upper bound achieved by SGD in section 5.4. This result can be extended to strongly convex problems (See theorem 2 in Agarwal et al., 2012).

Exercise 15.5 (♦♦) *Modify the argument from this section to show a lower bound for the generalization error of stochastic gradient methods for quadratic functions on a bounded convex set.*

15.4 Conclusion

This chapter was entirely dedicated to lower bounds of generalization error 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 in which 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.

Conclusion

The aim of this book was to provide a fundamental understanding of machine learning with the simplest possible arguments, but still an analysis that is fine enough to characterize when particular algorithms may or may not provide good predictions. Doing so, a few general important concepts were presented, which are summarized and discussed below.

Need for regularization. Generalization to unseen data cannot occur without some form of control of the "size" of the function space that a learning algorithm is exploring. This can be done explicitly, by constraining the number of parameters or penalizing a norm on these parameters, but also implicitly by computational regularization through (stochastic) gradient descent and its natural resistance to overfitting (see section 5.4) and potentially its implicit bias (see section 12.1).

Need for prior knowledge. Universal learning techniques that can learn on any supervised learning problem exist, such as local averaging techniques, neural networks, or kernel methods. However, without any assumption, there are prediction problems on which they will be arbitrarily slow (in terms of the required number of observations), as shown by our "no free lunch" theorems (see section 2.5) and lower bounds (see chapter 15). In order to obtain learning algorithms with good practical performance, prior knowledge is needed, such as: construction of relevant features by domain experts, explicit learning of representations with linear latent structure (such as neural networks), dependence on a small number of variables, or smoothness of the prediction function. Like all prior knowledge, however, it will only be useful if adapted to the learning task.

Need for adaptivity. Prior knowledge is typically imprecise; that is, we expect the predictions to depend on a small number of variables, but which ones? The prediction function should be a smooth function, but how smooth? Adaptive techniques will learn efficiently in these circumstances, typically by estimating by cross-validation a hyperparameter that controls the capacity of the learning problem (e.g., regularization parameter, or number of iterations in stochastic gradient descent or boosting techniques). In this book, we considered adaptivity to the smoothness of prediction functions (kernel methods and neural networks were adaptive, as opposed to local averaging techniques) and to linear latent variables, as needed for nonlinear variable selection (only neural networks were then adaptive).

Interplay between estimation, approximation, and estimation errors. The theoretical analysis of learning methods requires the study of typically three types of errors: the estimation error characterizes the effect of having a finite number of observations, the approximation error characterizes the effect of a reduced set of prediction functions; for methods based on empirical risk minimization (all in this book except local averaging techniques), the optimization error characterizes how well gradient-descent algorithms achieve a global minimum of the cost function they aim at minimizing.

Overfitting versus underfitting. One of the difficult and most interesting aspects of machine learning for high-dimensional problems is the constant dilemma between potential overfitting and potential underfitting. The interplay between the three types of errors can then be delicate, in particular for nonconvex objective functions, where the optimization error can often remain difficult to control.

Probabilistic analysis. The entire book considers a probabilistic analysis of supervised learning, which allows for making precise nonasymptotic statements. However, it relies on the common but rarely satisfied assumption of identically and independently distributed (i.i.d.) training data coming from the same distribution as the testing data. Dealing with extensions to the simplest framework is a key practical and theoretical challenge.

Role of convexity. Convexity plays a major role in the analysis of machine learning, in particular, because optimization errors can be controlled for convex objective functions, and precise guarantees and convergence rates can then be obtained. Since most of the loss functions that are used in practice are convex (at least after classical surrogates are used, even for complex output spaces; see chapter 13), the use of linear models (in their parameters) makes the problem convex. Still, even for nonlinear models such as neural networks, the use of convexity is crucial to obtain qualitative guarantees for overparameterized models (such as in section 12.3). Within the broader differentiable programming paradigm (see, e.g., Blondel and Roulet, 2024, and references therein), understanding when convexity is needed and when it is not remains an active area of research.

Going beyond supervised learning. This textbook focused primarily on the traditional supervised learning paradigm. Many applications require extensions to this basic framework, such as presented in section 2.7, which also lead to many interesting theoretical developments, such as unsupervised learning, semisupervised learning, active learning, reinforcement learning, and generative modeling. The goal of this book was to lay out the foundations for studying such further topics.

References

Abernethy, J., P. L. Bartlett, A. Rakhlin, and A. Tewari (2008). Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the Conference on Learning Theory.* (cited on page 321)

Adams, R. A. and J. J. F. Fournier (2003). *Sobolev Spaces.* Elsevier. (cited on page 438)

Agarwal, A., P. L. Bartlett, P. Ravikumar, and M. J. Wainwright (2012). Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory 58*(5), 3235–3249. (cited on pages 137, 141, 447, and 449)

Agarwal, A., D. P. Foster, D. Hsu, S. M. Kakade, and A. Rakhlin (2013). Stochastic convex optimization with bandit feedback. *SIAM Journal on Optimization 23*(1), 213–240. (cited on page 331)

Ailon, N., M. Charikar, and A. Newman (2008). Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM 55*(5), 1–27. (cited on page 405)

Akhavan, A., E. Chzhen, M. Pontil, and A. B. Tsybakov (2023). Gradient-free optimization of highly smooth functions: Improved analysis and a new algorithm. *arXiv 2306.02159.* (cited on page 326)

Alpaydin, E. (2020). *Introduction to Machine Learning.* MIT Press. (cited on page xii)

Alpaydin, E. (2022). *Maschinelles Lernen.* de Gruyter. (cited on page xii)

Alquier, P. (2024). User-friendly introduction to PAC-Bayes bounds. *Foundations and Trends in Machine Learning 17*(2), 174–303. (cited on pages 423, 424, and 425)

Alquier, P. and K. Lounici (2011). PAC-Bayesian bounds for sparse regression estimation with exponential weights. *Electronic Journal of Statistics 5*, 127–145. (cited on page 426)

Ambrosio, L., N. Gigli, and G. Savaré (2008). *Gradient Flows: In Metric Spaces and in the Space of Probability Measures.* Springer Science+Business Media. (cited on page 368)

Ambrosio, L., N. Gigli, and G. Savaré (2013). Density of Lipschitz functions and equivalence of weak gradients in metric measure spaces. *Revista Matemática Iberoamericana 29*(3), 969–996. (cited on page 175)

Andriushchenko, M., A. V. Varre, L. Pillaud-Vivien, and N. Flammarion (2023). SGD with large step sizes learns sparse features. In *Proceedings of the International Conference on Machine Learning.* (cited on page 355)

Araújo, D., R. I. Oliveira, and D. Yukimura (2019). A mean-field limit for certain deep neural networks. *arXiv 1906.00193.* (cited on page 370)

Arlot, S. and A. Celisse (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys 4*, 40–79. (cited on page 24)

Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics 16*(1), 1–3. (cited on page 112)

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society 68*, 337–404. (cited on pages 183 and 184)

Audibert, J.-Y. and S. Bubeck (2009). Minimax policies for adversarial and stochastic bandits. In *Proceedings of the Conference on Learning Theory*. (cited on page 341)

Audibert, J.-Y. and A. B. Tsybakov (2007). Fast learning rates for plug-in classifiers. *Annals of Statistics 35*(2), 608–633. (cited on pages 102 and 163)

Auer, P., N. Cesa-Bianchi, and P. Fischer (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning 47*(2), 235–256. (cited on pages 335 and 336)

Auer, P., N. Cesa-Bianchi, Y. Freund, and R. E. Schapire (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing 32*(1), 48–77. (cited on page 340)

Azencott, C.-A. (2019). *Introduction au Machine Learning*. Dunod. (cited on page xii)

Ba, J. L., J. R. Kiros, and G. E. Hinton (2016). Layer normalization. *arXiv 1607.06450*. (cited on page 251)

Bach, F. (2008). Consistency of trace norm minimization. *Journal of Machine Learning Research 9*(June), 1019–1048. (cited on page 244)

Bach, F. (2013). Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the Conference on Learning Theory*. (cited on page 220)

Bach, F. (2014). Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *Journal of Machine Learning Research 15*(1), 595–627. (cited on page 146)

Bach, F. (2015). Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization 25*(1), 115–129. (cited on page 267)

Bach, F. (2017). Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research 18*(1), 629–681. (cited on pages 266, 268, 269, 270, and 272)

Bach, F. (2023). On the relationship between multivariate splines and infinitely-wide neural networks. *arXiv 2302.03459*. (cited on pages 274 and 276)

Bach, F. (2024). High-dimensional analysis of double descent for linear regression with random projections. *SIAM Journal on Mathematics of Data Science 6*(1), 26–50. (cited on pages 360, 363, and 364)

Bach, F. and L. Chizat (2022). Gradient descent on infinitely wide neural networks: Global convergence and generalization. In *Proceedings of the International Congress of Mathematicians*. (cited on pages 251, 365, and 369)

Bach, F. and Z. Harchaoui (2007). Diffrac: A discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems*. (cited on page 103)

Bach, F., D. Heckerman, and E. Horvitz (2006). Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research 7*, 1713–1741. (cited on page 26)

Bach, F., R. Jenatton, J. Mairal, and G. Obozinski (2012a). Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning 4*(1), 1–106. (cited on pages 233 and 244)

Bach, F., R. Jenatton, J. Mairal, and G. Obozinski (2012b). Structured sparsity through convex optimization. *Statistical Science 27*(4), 450–468. (cited on page 244)

Bach, F. and E. Moulines (2013). Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems*. (cited on pages 140 and 146)

Bahdanau, D., K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. *arXiv 1409.0473*. (cited on page 279)

Ball, K., E. A. Carlen, and E. H. Lieb (2002). Sharp uniform convexity and smoothness inequalities for trace norms. In *Inequalities: Selecta of Elliott H. Lieb*, pp. 171–190. (cited on page 320)

Bansal, N. and A. Gupta (2019). Potential-function proofs for gradient methods. *Theory of Computing 15*(1), 1–32. (cited on page 125)

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory 39*(3), 930–945. (cited on pages 265 and 367)

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning 14*, 115–133. (cited on page 265)

Barron, A. R., A. Cohen, W. Dahmen, and R. A. DeVore (2008). Approximation and learning by greedy algorithms. *Annals of Statistics 36*(1), 64–94. (cited on page 308)

Barron, A. R. and J. M. Klusowski (2018). Approximation and estimation for high-dimensional deep learning networks. *arXiv 1809.03090*. (cited on page 260)

Bartlett, P. L., O. Bousquet, and S. Mendelson (2005). Local Rademacher complexities. *Annals of Statistics 33*(4), 1497–1537. (cited on page 98)

Bartlett, P. L., M. I. Jordan, and J. D. McAuliffe (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association 101*(473), 138–156. (cited on pages 78, 79, and 82)

Bartlett, P. L., P. M. Long, G. Lugosi, and A. Tsigler (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences 117*(48), 30063–30070. (cited on page 365)

Bartlett, P. L. and S. Mendelson (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research 3*(Nov.), 463–482. (cited on page 91)

Bartlett, P. L. and M. Traskin (2007). Adaboost is consistent. *Journal of Machine Learning Research 8*(78), 2347–2368. (cited on page 305)

Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research 18*(153), 1–43. (cited on page 324)

Beck, A. and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*(1), 183–202. (cited on page 128)

Belkin, M., D. Hsu, S. Ma, and S. Mandal (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences 116*(32), 15849–15854. (cited on pages 355 and 358)

Berlinet, A. and C. Thomas-Agnan (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer. (cited on page 184)

Berthet, Q., M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach (2020). Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*. (cited on page 397)

Berthier, R. (2023). Incremental learning in diagonal linear networks. *Journal of Machine Learning Research 24*(171), 1–26. (cited on page 375)

Bhatia, R. (2009). *Positive Definite Matrices*. Princeton University Press. (cited on page 117)

Bhatia, R. (2013). *Matrix Analysis*. Springer Science+Business Media. (cited on page 7)

Biau, G., F. Cérou, and A. Guyader (2010). On the rate of convergence of the bagged nearest neighbor estimate. *Journal of Machine Learning Research 11*(22), 687–712. (cited on page 287)

Biau, G. and L. Devroye (2015). *Lectures on the Nearest Neighbor Method*. Springer. (cited on pages 168, 170, and 176)

Biau, G. and E. Scornet (2016). A random forest guided tour. *Test 25*(2), 197–227. (cited on page 289)

Bietti, A. and F. Bach (2021). Deep equals shallow for ReLU networks in kernel regimes. In *International Conference on Learning Representations*. (cited on page 377)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. (cited on pages 40 and 410)

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association 112*(518), 859–877. (cited on page 422)

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research 3*(Jan.), 993–1022. (cited on page 418)

Blondel, M., A. F. T. Martins, and V. Niculae (2020). Learning with Fenchel-Young losses. *Journal of Machine Learning Research 21*(35), 1–69. (cited on pages 393 and 396)

Blondel, M. and V. Roulet (2024). The elements of differentiable programming. *arXiv 2403.14606*. (cited on page 452)

Blumensath, T. and M. E. Davies (2009). Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis 27*(3), 265–274. (cited on page 231)

Bolte, J., A. Daniilidis, and A. Lewis (2006). A nonsmooth Morse–Sard theorem for subanalytic functions. *Journal of Mathematical Analysis and Applications 321*(2), 729–740. (cited on page 372)

Bolte, J., A. Daniilidis, O. Ley, and L. Mazet (2010). Characterizations of Lojasiewicz inequalities and applications. *Transactions of the American Mathematical Society 362*(6), 3319–3363. (cited on pages 344 and 346)

Bolte, J. and E. Pauwels (2022). Curiosities and counterexamples in smooth convex optimization. *Mathematical Programming 195*(1), 553–603. (cited on page 112)

Boucheron, S., O. Bousquet, and G. Lugosi (2005). Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics 9*, 323–375. (cited on page 91)

Boucheron, S., G. Lugosi, and P. Massart (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press. (cited on page 9)

Bousquet, O. and A. Elisseeff (2002). Stability and generalization. *Journal of Machine Learning Research 2*, 499–526. (cited on page 102)

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press. (cited on pages 76, 81, 96, 117, 118, 123, 128, 199, and 345)

Brass, H. and K. Petras (2011). *Quadrature Theory: The Theory of Numerical Integration on a Compact Interval*. Number 178 in Mathematical Surveys and Monographs. American Mathematical Society. (cited on page 18)

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory 39*(3), 999–1013. (cited on page 260)

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32. (cited on page 289)

Breiman, L. and D. Freedman (1983). How many variables should be entered in a regression equation? *Journal of the American Statistical Association 78*(381), 131–136. (cited on page 65)

Bronstein, M. M., J. Bruna, T. Cohen, and P. Veličković (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv 2104.13478*. (cited on page 278)

Brouard, C., M. Szafranski, and F. d'Alché Buc (2016). Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research 17*(176), 1–48. (cited on page 394)

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning 8*(3–4), 231–357. (cited on pages 109, 133, 152, 441, and 442)

Bubeck, S. and N. Cesa-Bianchi (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning 5*(1), 1–122. (cited on pages 314, 331, 339, and 341)

Burer, S. and R. D. C. Monteiro (2003). A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming 95*(2), 329–357. (cited on page 374)

Cabannes, V., L. Pillaud-Vivien, F. Bach, and A. Rudi (2021). Overcoming the curse of dimensionality with Laplacian regularization in semi-supervised learning. In *Advances in Neural Information Processing Systems*. (cited on page 103)

Candes, E. and B. Recht (2012). Exact matrix completion via convex optimization. *Communications of the ACM 55*(6), 111–119. (cited on page 375)

Catoni, O. (2003). A PAC-Bayesian approach to adaptive classification. Technical Report 840, Laboratoire de Probabilités et Modèles Aléatoires, Université Paris 6. (cited on page 424)

Catoni, O. (2007). *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*. Institute of Mathematical Statistics. (cited on page 102)

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games*. Cambridge University Press. (cited on page 41)

Chan, S. H. (2024). Tutorial on diffusion models for imaging and vision. *arXiv 2403.18103*. (cited on page 41)

Chandrasekaran, V., B. Recht, P. A. Parrilo, and A. S. Willsky (2012). The convex geometry of linear inverse problems. *Foundations of Computational Mathematics 12*, 805–849. (cited on page 299)

Chapelle, O., B. Scholkopf, and A. Zien (Eds.) (2010). *Semi-supervised Learning*. MIT Press. (cited on page 40)

Chaudhuri, K. and S. Dasgupta (2014). Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*. (cited on page 163)

Chen, G. H. and D. Shah (2018). Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends in Machine Learning 10*(5-6), 337–588. (cited on page 170)

Chen, R. T. Q., Y. Rubanova, J. Bettencourt, and D. K. Duvenaud (2018). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*. (cited on page 278)

Chen, T. and C. Guestrin (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*. (cited on page 299)

Chizat, L. (2022). Sparse optimization on measures with overparameterized gradient descent. *Mathematical Programming 194*(1), 487–532. (cited on page 366)

Chizat, L. and F. Bach (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*. (cited on pages 251, 365, 368, and 369)

Chizat, L. and F. Bach (2020). Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Proceedings of the Conference on Learning Theory*. (cited on pages 354 and 377)

Chizat, L., E. Oyallon, and F. Bach (2019). On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*. (cited on pages 278 and 376)

Cho, Y. and L. K. Saul (2009). Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*. (cited on page 272)

Christmann, A. and I. Steinwart (2008). *Support Vector Machines*. Springer. (cited on pages xiii, 25, 107, and 179)

Ciliberto, C., L. Rosasco, and A. Rudi (2016). A consistent regularization approach for structured prediction. In *Advances in Neural Information Processing Systems*. (cited on page 390)

Ciliberto, C., L. Rosasco, and A. Rudi (2020). A general framework for consistent structured prediction with implicit loss embeddings. *Journal of Machine Learning Research 21*(98), 1–67. (cited on pages 387, 390, 393, 395, and 396)

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein (2022). *Introduction to Algorithms*. MIT Press. (cited on page 405)

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*, 273–297. (cited on page 75)

Cover, T. M. and J. A. Thomas (1999). *Elements of Information Theory*. John Wiley &

Sons. (cited on pages 431 and 434)

Cucker, F. and S. Smale (2002). On the mathematical foundations of learning. *Bulletin of the American Mathematical Society 39*(1), 1–49. (cited on page 217)

Cuturi, M., K. Fukumizu, and J.-P. Vert (2005). Semigroup kernels on measures. *Journal of Machine Learning Research 6*, 1169–1198. (cited on page 195)

Dalalyan, A. S. (2017). Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society Series B: Statistical Methodology 79*(3), 651–676. (cited on page 422)

d'Aspremont, A. (2008). Smooth optimization with approximate gradient. *SIAM Journal on Optimization 19*(3), 1171–1183. (cited on page 135)

d'Aspremont, A., D. Scieur, and A. Taylor (2021). Acceleration methods. *Foundations and Trends in Optimization 5*(1-2), 1–245. (cited on page 127)

Davis, P. J. and P. Rabinowitz (1984). *Methods of Numerical Integration*. Academic Press. (cited on page 18)

Defazio, A., F. Bach, and S. Lacoste-Julien (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*. (cited on page 147)

Défossez, A. and F. Bach (2015). Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. (cited on page 146)

Défossez, A., L. Bottou, F. Bach, and N. Usunier (2022). A simple convergence proof of Adam and Adagrad. *Transactions on Machine Learning Research*. (cited on page 143)

DeVore, R. A. and V. N. Temlyakov (1996). Some remarks on greedy algorithms. *Advances in Computational Mathematics 5*, 173–187. (cited on page 302)

Devroye, L., L. Györfi, and G. Lugosi (1996). *A Probabilistic Theory of Pattern Recognition*. Springer Science+Business Media. (cited on pages xiii, 38, and 39)

Dietterich, T. G. and G. Bakiri (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research 2*, 263–286. (cited on pages 382 and 396)

Dieuleveut, A. and F. Bach (2016). Nonparametric stochastic approximation with large step sizes. *Annals of Statistics 44*(4), 1363–1399. (cited on page 146)

Dieuleveut, A., N. Flammarion, and F. Bach (2017). Harder, better, faster, stronger convergence rates for least-squares regression. *Journal of Machine Learning Research 18*(1), 3520–3570. (cited on page 146)

Dobriban, E. and S. Liu (2019). Asymptotics for sketching in least squares regression. In *Advances in Neural Information Processing Systems*. (cited on page 291)

Donoho, D. L. and I. M. Johnstone (1994). Minimax risk over ℓ_p-balls for ℓ_q-error. *Probability Theory and Related Fields 99*(2), 277–303. (cited on pages 438 and 440)

Du, S. S., X. Zhai, B. Poczos, and A. Singh (2018). Gradient descent provably optimizes overparameterized neural networks. In *International Conference on Learning Representations*. (cited on page 376)

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*(61), 2121–2159. (cited on pages 143 and 251)

Duchi, J. C., M. I. Jordan, M. J. Wainwright, and A. Wibisono (2015). Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory 61*(5), 2788–2806. (cited on page 328)

Durmus, A. and E. Moulines (2017). Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *Annals of Applied Probability 27*(3), 1551–1587. (cited on page 422)

E, W. and S. Wojtowytsch (2020). On the Banach spaces associated with multilayer ReLU networks: Function representation, approximation theory and gradient descent dynamics. *arXiv 2007.15623*. (cited on page 370)

Efron, B. and R. J. Tibshirani (1994). *An Introduction to the Bootstrap*. Chapman and Hall. (cited on page 286)

Eldar, Y. C. and G. Kutyniok (Eds.) (2012). *Compressed Sensing: Theory and Applications*. Cambridge University Press. (cited on page 241)

Evans, L. C. (2022). *Partial Differential Equations*. American Mathematical Society. (cited on page 368)

Fan, J., T. Gasser, I. Gijbels, M. Brockmann, and J. Engel (1997). Local polynomial regression: Optimal kernels and asymptotic minimax efficiency. *Annals of the Institute of Statistical Mathematics 49*, 79–99. (cited on page 177)

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society, Series B 70*(5), 849–911. (cited on page 419)

Fang, C., J. Lee, P. Yang, and T. Zhang (2021). Modeling from features: A mean-field framework for overparameterized deep neural networks. In *Proceedings of the Conference on Learning Theory*. (cited on page 370)

Fathony, R., A. Liu, K. Asif, and B. Ziebart (2016). Adversarial multiclass classification: A risk minimization perspective. In *Advances in Neural Information Processing Systems*. (cited on pages 399 and 402)

Fercoq, O. and P. Richtárik (2015). Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization 25*(4), 1997–2023. (cited on page 233)

Freund, Y., R. Schapire, and N. Abe (1999). A short introduction to boosting. *Japanese Society for Artificial Intelligence 14*(771–780), 1612. (cited on page 298)

Freund, Y. and R. E. Schapire (1996). Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning*. (cited on pages 302 and 303)

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences 55*(1), 119–139. (cited on page 340)

Friedman, J., T. Hastie, and R. Tibshirani (2009). *The Elements of Statistical Learning*. Springer. (cited on page 160)

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics 29*(5), 1189–1232. (cited on page 305)

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research 17*(1), 2096–2030. (cited on page 103)

Gao, W. and Z.-H. Zhou (2015). On the consistency of AUC pairwise optimization. In *Proceedings of the International Conference on Artificial Intelligence*. (cited on page 389)

Garivier, A. and O. Cappé (2011). The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the Conference on Learning Theory*. (cited on page 336)

Gauss, C. F. (1809). *Theoria Motus Corporum Coelestium*. Perthes & Besser. (cited on page 45)

Geiger, M., A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. d'Ascoli, G. Biroli, C. Hongler, and M. Wyart (2020). Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment 2020*(2), 023401. (cited on pages 355 and 358)

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (1995). *Bayesian Data Analysis*. Chapman and Hall/CRC. (cited on page 421)

Gidel, G., F. Bach, and S. Lacoste-Julien (2019). Implicit regularization of discrete gradient dynamics in linear neural networks. In *Advances in Neural Information Processing Systems*. (cited on page 375)

Giraud, C. (2014). *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC. (cited on pages 227, 234, 237, 240, 241, and 243)

Giraud, C., S. Huet, and N. Verzelen (2012). High-dimensional regression with unknown variance. *Statistical Science 27*(4), 500–518. (cited on page 230)

Goldstein, A. A. (1962). Cauchy's method of minimization. *Numerische Mathematik 4*(1), 146–150. (cited on page 112)

Golub, G. H. and C. F. V. Loan (1996). *Matrix Computations*. Johns Hopkins University Press. (cited on pages 7, 49, 67, 115, 129, and 197)

Gönen, M. and E. Alpaydın (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research 12*, 2211–2268. (cited on pages 244 and 273)

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. (cited on pages 40, 278, and 369)

Gower, R. M., M. Schmidt, F. Bach, and P. Richtárik (2020). Variance-reduced methods for machine learning. *Proceedings of the IEEE 108*(11), 1968–1983. (cited on page 150)

Gribonval, R. (2011). Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Transactions on Signal Processing 59*(5), 2405–2410. (cited on pages 413 and 415)

Gribonval, R., V. Cevher, and M. E. Davies (2012). Compressible distributions for high-dimensional statistics. *IEEE Transactions on Information Theory 58*(8), 5016–5034. (cited on page 416)

Gunasekar, S., J. Lee, D. Soudry, and N. Srebro (2018). Characterizing implicit bias in terms of optimization geometry. In *Proceedings of the International Conference on Machine Learning*. (cited on page 348)

Gunasekar, S., B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro (2017). Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*. (cited on page 375)

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press. (cited on page 195)

Györfi, L., M. Kohler, A. Krzyzak, and H. Walk (2006). *A Distribution-Free Theory of Nonparametric Regression*. Springer Science+Business Media. (cited on page 166)

Haff, L. R. (1979). An identity for the Wishart distribution with applications. *Journal of Multivariate Analysis 9*(4), 531–544. (cited on pages 358 and 361)

Hamm, T. and I. Steinwart (2021). Adaptive learning rates for support vector machines working on data with low intrinsic dimension. *Annals of Statistics 49*(6), 3153–3180. (cited on page 40)

Hanin, B. and M. Nica (2019). Finite depth and width corrections to the neural tangent kernel. In *International Conference on Learning Representations*. (cited on page 370)

Harchaoui, Z., F. Bach, and E. Moulines (2008). Testing for homogeneity with kernel Fisher discriminant analysis. *arXiv 0804.1026*. (cited on page 218)

Hastie, T., A. Montanari, S. Rosset, and R. J. Tibshirani (2022). Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics 50*(2), 949–986. (cited on pages 355 and 360)

Hazan, E. (2022). *Introduction to Online Convex Optimization*. MIT Press. (cited on pages 41, 314, 331, and 341)

Hazan, E. and S. Kale (2014). Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly convex optimization. *Journal of Machine Learning Research 15*(1), 2489–2512. (cited on page 319)

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. (cited on page 278)

Holtz, M. (2010). *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*. Springer Science+Business Media. (cited on page 18)

Hsu, D., S. M. Kakade, and T. Zhang (2012). Random design analysis of ridge regression. In *Proceedings of the Conference on Learning Theory*. (cited on page 65)

Huber, P. J. and E. M. Ronchetti (2009). *Robust Statistics*. John Wiley & Sons. (cited on page 388)

Hyvärinen, A., J. Karhunen, and E. Oja (2001). *Independent Component Analysis*. John Willey and Sons. (cited on page 40)

Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*. (cited on page 251)

Jacot, A., F. Gabriel, and C. Hongler (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*. (cited on pages 278 and 377)

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the International Conference on Machine Learning*. (cited on page 267)

Ji, Z. and M. Telgarsky (2018). Risk and parameter convergence of logistic regression. *arXiv 1803.07300*. (cited on page 350)

Johnson, R. and T. Zhang (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*. (cited on page 147)

Johnson, W. B. and J. Lindenstrauss (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Conference on Modern Analysis and Probability*. (cited on page 296)

Joulin, A., É. Grave, P. Bojanowski, and T. Mikolov (2017). Bag of tricks for efficient text classification. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. (cited on page 195)

Juditsky, A. and A. Nemirovski (2011a). First order methods for nonsmooth convex large-scale optimization, I: General purpose methods. In *Optimization for Machine Learning*, pp. 121–148. MIT Press. (cited on page 133)

Juditsky, A. and A. Nemirovski (2011b). First order methods for nonsmooth convex large-scale optimization, II: Utilizing problems structure. In *Optimization for Machine Learning*, pp. 149–183. MIT Press. (cited on page 133)

Kabán, A. (2014). New bounds on compressive linear least squares regression. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. (cited on page 293)

Kanagawa, M., P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv 1807.02582*. (cited on pages 185 and 423)

Karimi, H., J. Nutini, and M. Schmidt (2016). 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*. (cited on page 346)

Kimeldorf, G. and G. Wahba (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications 33*, 82–95. (cited on page 181)

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv 1412.6980*. (cited on pages 143 and 251)

Klusowski, J. M. and A. R. Barron (2018). Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls. *IEEE Transactions on Information Theory 64*(12), 7649–7656. (cited on page 265)

Klusowski, J. M. and J. W. Siegel (2023). Sharp convergence rates for matching pursuit. *arXiv 2307.07679*. (cited on page 303)

Koltchinskii, V. (2011). *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: École d'Été de Probabilités de Saint-Flour (2008)*. Springer Science+Business Media. (cited on page xiii)

Koltchinskii, V. and O. Beznosova (2005). Exponential convergence rates in classification. In *International Conference on Computational Learning Theory*. (cited on page 102)

Kpotufe, S. (2011). k-NN regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems*. (cited on page 40)

Kurková, V. and M. Sanguineti (2001). Bounds on rates of variable-basis and neural-network approximation. *IEEE Transactions on Information Theory 47*(6), 2659–2665. (cited on pages 259, 299, and 367)

Kushner, H. J. and G. G. Yin (2003). *Stochastic Approximation and Recursive Algorithms* *and Applications*. Springer-Verlag. (cited on page 367)

Lattimore, T. and C. Szepesvári (2020). *Bandit Algorithms*. Cambridge University Press. (cited on pages 314, 331, and 341)

Le Roux, N. and Y. Bengio (2007). Continuous neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. (cited on page 198)

Lecué, G. and S. Mendelson (2016). Performance of empirical risk minimization in linear aggregation. *Bernoulli 22*(3), 1520–1534. (cited on page 65)

Ledoux, M. and M. Talagrand (1991). *Probability in Banach Spaces: Isoperimetry and Processes*. Springer Science+Business Media. (cited on pages 94 and 95)

Lee, J. D., M. Simchowitz, M. I. Jordan, and B. Recht (2016). Gradient descent only converges to minimizers. In *Proceedings of the Conference on Learning Theory*. (cited on page 373)

Lee, Y., Y. Lin, and G. Wahba (2004). 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. (cited on page 393)

Legendre, A.-M. (1805). *Nouvelles Méthodes pour la Détermination des Orbites des Comètes*. Firmin Didot. (cited on page 45)

Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks 6*(6), 861–867. (cited on page 256)

Liberzon, D. (2011). *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press. (cited on page 41)

Lindholm, A., N. Wahlström, F. Lindsten, and T. B. Schön (2022). *Machine Learning: A First Course for Engineers and Scientists*. Cambridge University Press. (cited on page xii)

Liu, Y. (2007). Fisher consistency of multicategory support vector machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. (cited on page 399)

Long, P. and R. Servedio (2013). Consistency versus realizable H-consistency for multiclass classification. In *Proceedings of the International Conference on Machine Learning*. (cited on pages 83 and 393)

Lu, J., Z. Shen, H. Yang, and S. Zhang (2021). Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis 53*(5), 5465–5506. (cited on page 278)

Lugosi, G. and N. Vayatis (2004). On the Bayes-risk consistency of regularized boosting methods. *Annals of Statistics 32*(1), 30–55. (cited on page 308)

Lyu, K. and J. Li (2019). Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*. (cited on page 377)

Lyu, K., Z. Li, and S. Arora (2022). Understanding the generalization benefit of normalization layers: Sharpness reduction. In *Advances in Neural Information Processing Systems*. (cited on page 355)

Ma, C., S. Wojtowytsch, and L. Wu (2020). Towards a mathematical understanding of neural network-based machine learning: What we know and what we don't. *arXiv 2009.10713*. (cited on page 278)

Mahoney, M. W. and P. Drineas (2009). CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences 106*(3), 697–702. (cited on page 197)

Mairal, J., F. Bach, and J. Ponce (2014). Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision 8*(2–3), 85–283. (cited on pages 40 and 245)

Mairal, J. and B. Yu (2012). Complexity analysis of the Lasso regularization path. In *Proceedings of the International Conference on International Conference on Machine Learning*. (cited on page 234)

Mallat, S. G. and Z. Zhang (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing 41*(12), 3397–3415. (cited on pages 302 and 303)

Marden, J. I. (1996). *Analyzing and Modeling Rank Data*. CRC Press. (cited on page 405)

Martinsson, P.-G. and J. A. Tropp (2020). Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica 29*, 403–572. (cited on pages 129 and 197)

Maurer, A. (2016). A vector-contraction inequality for Rademacher complexities. In *International Conference on Algorithmic Learning Theory*. (cited on page 385)

Mei, S., T. Misiakiewicz, and A. Montanari (2019). Mean-field theory of two-layer neural networks: Dimension-free bounds and kernel limit. In *Proceedings of the Conference on Learning Theory*. (cited on page 370)

Mei, S. and A. Montanari (2022). The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics 75*(4), 667–766. (cited on pages 355, 358, and 360)

Mei, S., A. Montanari, and P.-M. Nguyen (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences 115*(33), E7665–E7671. (cited on page 368)

Meir, R. and T. Zhang (2003). Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research 4*(Oct.), 839–860. (cited on pages 86 and 94)

Minsker, S. (2017). On some extensions of Bernstein's inequality for self-adjoint operators. *Statistics & Probability Letters 127*, 111–119. (cited on page 20)

Mohri, M. and A. Rostamizadeh (2010). Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research 11*(26), 789–814. (cited on page 103)

Mohri, M., A. Rostamizadeh, and A. Talwalkar (2018). *Foundations of Machine Learning*. MIT Press. (cited on page xiii)

Mourtada, J. (2022). Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. *Annals of Statistics 50*(4), 2157–2178. (cited on pages 61, 64, and 65)

Mourtada, J. and L. Rosasco (2022). An elementary analysis of ridge regression with random design. *Comptes Rendus. Mathématique 360*, 1055–1063. (cited on page 212)

Munos, R. (2014). 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. (cited on page 336)

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press. (cited on pages 40, 396, 397, and 410)

Nagaraj, D., P. Jain, and P. Netrapalli (2019). SGD without replacement: Sharper rates for general smooth convex functions. In *Proceedings of the International Conference on Machine Learning*. (cited on page 135)

Neal, R. M. (1995). *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto. (cited on page 272)

Nekvinda, A. and L. Zajíček (1988). A simple proof of the Rademacher theorem. *Časopis pro pěstování matematiky 113*(4), 337–341. (cited on page 130)

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Doklady Akademii Nauk SSSR 269*(3), 543. (cited on page 127)

Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer. (cited on page 126)

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical Programming 140*(1), 125–161. (cited on pages 127 and 128)

Nesterov, Y. (2018). *Lectures on Convex Optimization*. Springer. (cited on pages 109, 119, 120, 126, 152, 441, 442, and 443)

Nesterov, Y. and V. Spokoiny (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics 17*(2), 527–566. (cited on page 112)

Neyshabur, B., R. Tomioka, and N. Srebro (2015). Norm-based capacity control in neural networks. In *Proceedings of the Conference on Learning Theory*. (cited on page 255)

Ng, A. Y. and M. I. Jordan (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*. (cited on page 419)

Nguyen, P.-M. and H. T. Pham (2023). A rigorous framework for the mean field limit of multilayer neural networks. *Mathematical Statistics and Learning 6*(3), 201–357. (cited on page 370)

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM. (cited on page 18)

Nitanda, A. and T. Suzuki (2017). Stochastic particle gradient descent for infinite ensembles. *arXiv 1712.05438*. (cited on page 368)

Nocedal, J. and S. J. Wright (1999). *Numerical Optimization*. Springer. (cited on page 123)

Novak, E. (2006). *Deterministic and Stochastic Error Bounds in Numerical Analysis*. Springer. (cited on pages 444 and 446)

Novikoff, A. B. J. (1962). On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*. (cited on page 351)

Nowak, A., F. Bach, and A. Rudi (2019). Sharp analysis of learning with discrete losses. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. (cited on pages 387 and 388)

Nowak-Vila, A., F. Bach, and A. Rudi (2019). A general theory for structured prediction with smooth convex surrogates. *arXiv 1902.01958*. (cited on page 396)

Nowak-Vila, A., F. Bach, and A. Rudi (2020). Consistent structured prediction with max-min margin Markov networks. In *Proceedings of the International Conference on*

Machine Learning. (cited on pages 399 and 401)

Oliveira, R. I. (2016). The lower tail of random quadratic forms with applications to ordinary least squares. *Probability Theory and Related Fields 166*, 1175–1194. (cited on page 65)

Opper, M., W. Kinzel, J. Kleinz, and R. Nehl (1990). On the ability of the optimal perceptron to generalise. *Journal of Physics A: Mathematical and General 23*(11), L581. (cited on page 355)

Orabona, F. (2019). A modern introduction to online learning. *arXiv 1912.13213*. (cited on pages 314 and 321)

Osborne, M. R., B. Presnell, and B. A. Turlach (2000). On the Lasso and its dual. *Journal of Computational and Graphical Statistics 9*(2), 319–337. (cited on page 234)

Osokin, A., F. Bach, and S. Lacoste-Julien (2017). On structured prediction theory with calibrated convex surrogate losses. In *Advances in Neural Information Processing Systems*. (cited on page 393)

Ostrovskii, D. and F. Bach (2021). Finite-sample analysis of M-estimators using self-concordance. *Electronic Journal of Statistics 15*(1), 326–391. (cited on pages 107 and 239)

Palmer, J., K. Kreutz-Delgado, B. Rao, and D. Wipf (2005). Variational EM algorithms for non-Gaussian latent variable models. In *Advances in Neural Information Processing Systems*. (cited on page 413)

Papandreou, G. and A. L. Yuille (2011). Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *International Conference on Computer Vision*. (cited on page 397)

Pati, Y. C., R. Rezaiifar, and P. S. Krishnaprasad (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*. (cited on page 228)

Pedregosa, F., F. Bach, and A. Gramfort (2017). On the consistency of ordinal regression methods. *Journal of Machine Learning Research 18*, 1–35. (cited on page 388)

Pesme, S. and N. Flammarion (2023). Saddle-to-saddle dynamics in diagonal linear networks. In *Advances in Neural Information Processing Systems*. (cited on page 375)

Pesme, S., L. Pillaud-Vivien, and N. Flammarion (2021). Implicit bias of SGD for diagonal linear networks: A provable benefit of stochasticity. In *Advances in Neural Information Processing Systems*. (cited on page 354)

Pillaud-Vivien, L., A. Rudi, and F. Bach (2018). Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. In *Advances in Neural Information Processing Systems*. (cited on page 218)

Pimentel, M. A. F., D. A. Clifton, L. Clifton, and L. Tarassenko (2014). A review of novelty detection. *Signal Processing 99*, 215–249. (cited on page 40)

Platt, J. (1998). Using analytic QP and sparseness to speed training of support vector machines. In *Advances in Neural Information Processing Systems*. (cited on page 76)

Potters, M. and J.-P. Bouchaud (2020). *A First Course in Random Matrix Theory: For Physicists, Engineers and Data Scientists*. Cambridge University Press. (cited on pages 37 and 107)

Rahimi, A. and B. Recht (2008). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems.* (cited on pages 198 and 272)

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning.* MIT Press. (cited on pages 185, 192, 193, and 423)

Reed, M. and B. Simon (1978). *Methods of Modern Mathematical Physics, Volume 2.* Academic Press. (cited on page 192)

Rigollet, P. and A. Tsybakov (2011). Exponential screening and optimal rates of sparse estimation. *Annals of Statistics 39*(2), 731–771. (cited on page 426)

Rigollet, P. and A. B. Tsybakov (2007). Linear and convex aggregation of density estimators. *Mathematical Methods of Statistics 16*(3), 260–280. (cited on page 223)

Robert, C. P. (2007). *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*, Volume 2. Springer. (cited on pages 420, 421, and 423)

Robert, C. P. and G. Casella (2005). *Monte Carlo Statistical Methods*, Volume 2. Springer. (cited on page 422)

Rockafellar, R. T. (1997). *Convex Analysis.* Princeton University Press. (cited on pages 131 and 300)

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review 65*(6), 386–408. (cited on page 351)

Rosset, S., J. Zhu, and T. Hastie (2004). Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research 5*, 941–973. (cited on page 307)

Rotskoff, G. M. and E. Vanden-Eijnden (2018). Parameters as interacting particles: Long time convergence and asymptotic error scaling of neural networks. In *Advances in Neural Information Processing Systems.* (cited on page 368)

Rudi, A., R. Camoriano, and L. Rosasco (2015). Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems.* (cited on pages 197, 212, and 215)

Rudi, A. and L. Rosasco (2017). Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems.* (cited on pages 198, 212, and 215)

Rudin, W. (1987). *Real and Complex Analysis.* McGraw-Hill. (cited on pages 257, 262, and 263)

Russo, D. J., B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen (2018). A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning 11*(1), 1–96. (cited on page 339)

Santambrogio, F. (2015). *Optimal Transport for Applied Mathematicians.* Springer. (cited on page 368)

Saxe, A. M., J. L. McClelland, and S. Ganguli (2019). A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences 116*(23), 11537–11546. (cited on page 375)

Schaback, R. and H. Wendland (2006). Kernel techniques: From machine learning to meshless methods. *Acta Numerica 15*, 543–639. (cited on page 180)

Schapire, R. E. and Y. Freund (2012). *Boosting: Foundations and Algorithms.* MIT Press. (cited on page 298)

Schmidt, M., N. Le Roux, and F. Bach (2011). Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems*. (cited on page 135)

Schmidt, M., N. Le Roux, and F. Bach (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming 162*(1–2), 83–112. (cited on page 147)

Schölkopf, B., R. Herbrich, and A. J. Smola (2001). A generalized representer theorem. In *International Conference on Computational Learning Theory*. (cited on page 181)

Schölkopf, B. and A. J. Smola (2001). *Learning with Kernels*. MIT Press. (cited on pages 179 and 201)

Schölkopf, B., K. Tsuda, and J.-P. Vert (Eds.) (2004). *Kernel Methods in Computational Biology*. MIT Press. (cited on page 195)

Scieur, D., V. Roulet, F. Bach, and A. d'Aspremont (2017). Integration methods and optimization algorithms. In *Advances in Neural Information Processing Systems*. (cited on pages 123 and 348)

Scornet, E., G. Biau, and J.-P. Vert (2015). Consistency of random forests. *Annals of Statistics 43*(4), 1716–1741. (cited on page 300)

Seldin, Y., C. Szepesvári, P. Auer, and Y. Abbasi-Yadkori (2013). Evaluation and analysis of the performance of the EXP3 algorithm in stochastic environments. In *European Workshop on Reinforcement Learning*. (cited on page 341)

Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences. (cited on page 41)

Shakhnarovich, G., T. Darrell, and P. Indyk (Eds.) (2005). *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press. (cited on page 161)

Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Foundations and Trends in Machine Learning 4*(2), 107–194. (cited on pages 314 and 341)

Shalev-Shwartz, S. and S. Ben-David (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. (cited on page xiii)

Shawe-Taylor, J. and N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press. (cited on pages 179, 194, 195, and 201)

Sil'nichenko, A. V. (2004). Rate of convergence of greedy algorithms. *Mathematical Notes 76*, 582–586. (cited on page 303)

Silva Filho, T., H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach (2023). Classifier calibration: A survey on how to assess and improve predicted class probabilities. *Machine Learning 112*(9), 3211–3260. (cited on page 79)

Silverman, B. W. (1982). 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. (cited on page 163)

Sirignano, J. and K. Spiliopoulos (2020). Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics 80*(2), 725–752. (cited on page 368)

Sirignano, J. and K. Spiliopoulos (2022). Mean field analysis of deep neural networks. *Mathematics of Operations Research 47*(1), 120–152. (cited on page 370)

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends in*

Machine Learning 12(1-2), 1–286. (cited on pages 314, 331, and 341)

Smith, S. L., B. Dherin, D. G. T. Barrett, and S. De (2021). On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*. (cited on page 355)

Snoek, J., H. Larochelle, and R. P. Adams (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*. (cited on page 444)

Soudry, D., E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro (2018). The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research 19*(1), 2822–2878. (cited on page 350)

Sridharan, K., S. Shalev-Shwartz, and N. Srebro (2009). Fast rates for regularized objectives. In *Advances in Neural Information Processing Systems*. (cited on page 98)

Stein, M. L. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science+Business Media. (cited on page 193)

Steinwart, I. (2003). Sparseness of support vector machines. *Journal of Machine Learning Research 4*(Nov.), 1071–1105. (cited on page 76)

Steinwart, I. and C. Scovel (2012). Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constructive Approximation 35*, 363–417. (cited on pages 190 and 217)

Stewart, G. W. and J.-G. Sun (1990). *Matrix Perturbation Theory*. Academic Press. (cited on pages 7 and 20)

Stewart, L., F. Bach, Q. Berthet, and J.-P. Vert (2023). Regression as classification: Influence of task formulation on neural network features. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. (cited on page 278)

Stone, C. J. (1977). Consistent nonparametric regression. *Annals of Statistics 5*(4), 595–620. (cited on page 174)

Sugiyama, M., M. Krauledat, and K.-R. Müller (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research 8*(5). (cited on page 103)

Suli, E. and D. F. Mayers (2003). *An Introduction to Numerical Analysis*. Cambridge University Press. (cited on page 367)

Sutton, C. and A. McCallum (2012). An introduction to conditional random fields. *Foundations and Trends in Machine Learning 4*(4), 267–373. (cited on pages 393 and 397)

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction*. MIT Press. (cited on page 41)

Taskar, B., V. Chatalbashev, D. Koller, and C. Guestrin (2005). Learning structured prediction models: A large margin approach. In *Proceedings of the International Conference on Machine learning*. (cited on page 399)

Thanei, G.-A., C. Heinze, and N. Meinshausen (2017). Random projections for large-scale regression. In *Big and Complex Data Analysis*, pp. 51–68. Springer. (cited on page 293)

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288. (cited on page 231)

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine.

Journal of Machine Learning Research 1, 211–244. (cited on page 423)

Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics 12*(4), 389–434. (cited on pages 19, 20, and 65)

Tsochantaridis, I., T. Joachims, T. Hofmann, Y. Altun, and Y. Singer (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research 6*(50), 1453–1484. (cited on page 399)

Tsybakov, A. B. (2008). *Introduction to Nonparametric Estimation*. Springer Science+Business Media. (cited on pages 177 and 438)

van der Vaart, A. W. (2000). *Asymptotic Statistics*. Cambridge University Press. (cited on pages 105 and 106)

van Engelen, J. E. and H. H. Hoos (2020). A survey on semi-supervised learning. *Machine Learning 109*(2), 373–440. (cited on page 40)

Vapnik, V. N. and A. Y. Chervonenkis (1964). On a perceptron class. *Automation and Remote Control 25*, 112–120. (cited on page 74)

Vapnik, V. N. and A. Y. Chervonenkis (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pp. 11–30. Springer. (cited on pages xiii and 72)

Varadhan, S. R. S. (2001). *Probability Theory*. American Mathematical Society. (cited on page 192)

Vardi, G. (2023). On the implicit bias in deep-learning algorithms. *Communications of the ACM 66*(6), 86–93. (cited on page 355)

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*. (cited on page 279)

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press. (cited on page 9)

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM. (cited on page 276)

Wainwright, M. J. (2019). *High-Dimensional Statistics: A Nonasymptotic Viewpoint*. Cambridge University Press. (cited on pages 90, 91, 240, and 241)

Wainwright, M. J. and M. I. Jordan (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning 1*(1–2), 1–305. (cited on pages 391 and 422)

Waldspurger, I. (2021). Lecture notes on non-convex algorithms for low-rank matrix recovery. *arXiv 2105.10318*. (cited on page 374)

Wang, S., A. Gittens, and M. W. Mahoney (2018). Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. *Journal of Machine Learning Research 18*, 1–50. (cited on page 291)

Wasserman, L. (2006). *All of Nonparametric Statistics*. Springer Science+Business Media. (cited on page 177)

Weston, J., O. Chapelle, V. Vapnik, A. Elisseeff, and B. Schölkopf (2002). Kernel dependency estimation. In *Advances in Neural Information Processing Systems*. (cited on page 394)

Williams, C. and M. Seeger (2000). Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems.* (cited on page 197)

Woodworth, B., S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro (2020). Kernel and rich regimes in overparametrized models. In *Proceedings of the Conference on Learning Theory.* (cited on pages 353 and 354)

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research 11*, 2543–2596. (cited on page 321)

Xu, L., J. Neufeld, B. Larson, and D. Schuurmans (2004). Maximum margin clustering. In *Advances in Neural Information Processing Systems.* (cited on page 103)

Yan, W.-Y., U. Helmke, and J. B. Moore (1994). Global analysis of Oja's flow for neural networks. *IEEE Transactions on Neural Networks 5*(5), 674–683. (cited on page 374)

Yang, G. and E. J. Hu (2020). Feature learning in infinite-width neural networks. *arXiv 2011.14522.* (cited on page 370)

Yang, G. and E. J. Hu (2021). Tensor programs IV: Feature learning in infinite-width neural networks. In *Proceedings of the International Conference on Machine Learning.* (cited on page 278)

Yang, Y. (1999). Minimax nonparametric classification. i. rates of convergence. *IEEE Transactions on Information Theory 45*(7), 2271–2284. (cited on page 428)

Zhang, J., M. Marszałek, S. Lazebnik, and C. Schmid (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision 73*(2), 213–238. (cited on page 195)

Zhang, L., M. Mahdavi, and R. Jin (2013). Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems.* (cited on page 147)

Zhang, T. (2004a). Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research 5*(Oct.), 1225–1251. (cited on page 382)

Zhang, T. (2004b). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics 32*(1), 56–85. (cited on page 79)

Zhang, T. (2006). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory 52*(4), 1307–1321. (cited on page 102)

Zhang, T. (2009). On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research 10*(19), 555–568. (cited on page 228)

Zhang, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory 57*(7), 4689–4708. (cited on page 230)