

Learning Theory from First Principles

Draft

Francis Bach
francis.bach@inria.fr

April 5, 2021

Preface

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

The goal of the class (and thus of this textbook) is to present old and recent results in learning theory, for the most widely-used learning architectures. This class is geared towards theory-oriented students as well as students who want to acquire a basic mathematical understanding of algorithms used throughout machine learning and associated fields that are large users of learning methods such as computer vision or natural language processing.

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 that show-case in simple but relevant instances the important concepts in learning theory. Some general results will also be presented without proofs. Of course the concept of first principles is subjective, and I will assume a good knowledge of linear algebra, probability theory and differential calculus.

Moreover, I will focus on the part of learning theory that does not exist outside of algorithms that can be run in practice, and thus all algorithmic frameworks described in this book are routinely used. For most learning methods, some simple **illustrative experiments** are presented, with the plan to have accompanying code (Matlab, Julia, and Python) so that students can see for themselves that the algorithms are simple and effective in synthetic experiments.

⚠ This is *not* an introductory textbook on machine learning. There are already several good ones in several languages [3, 7].

The choice of topics is arbitrary (and thus personal). Many important algorithmic frameworks are forgotten (e.g., reinforcement learning, unsupervised learning, etc.). Suggestions of extra themes are welcome! A few additional chapters are currently being written such as:

- Ensemble learning
- Bandit optimization
- Probabilistic methods
- Structured prediction

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

⚠ Many topics are not covered, and many more are not covered in much depth. There are many good textbooks on learning theory that go deeper [75, 96, 38].

⚠ This is still work in progress. In particular, there are still a lot of typos, probably some mistakes, 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 simpler mathematical arguments are possible in many places in the book. If you are aware of elegant and simple ideas that I have overlooked, please let me know.

Mathematical notations

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

- Random variables: given a set \mathcal{X} , we will use the lower-case notation for a random variable with values in \mathcal{X} , as well for its observations. Probability distributions will be denoted $d\mu$ or dp 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).
- 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 $A \in \mathbb{R}^n$, $A \succcurlyeq 0$ means that A is positive semi-definite (that is, all of its eigenvalues are non-negative), and for two symmetric matrices A and B , $A \succcurlyeq B$ means that $A - B \succcurlyeq 0$.
- For a differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, its gradient at x is denoted $f'(x) \in \mathbb{R}^d$, and if it is twice differentiable, its Hessian is denoted $f''(x) \in \mathbb{R}^{d \times d}$.

Acknowledgements

These class notes have been adapted from the notes of many colleagues I have the pleasure to work with, in particular L  na  c Chizat, Pierre Gaillard, Alessandro Rudi and Simon Lacoste-Julien. Special thanks to L  na  c Chizat for his help for the chapter on neural networks and for proof-reading many of the chapters, to Jaouad Mourtada for his help on lower bounds and random design analysis for least-squares regression, to Alex Nowak-Vila for his help on calibration functions, to Vivien Cabannes for the help on consistency proofs for local averaging techniques, to Alessandro Rudi for his help on kernel methods. The notes from Philippe Rigollet have also been a very precious help for model selection.

Typos have been found by Ritobrata Ghosh (⚠ Add your name to the list!).

Contents

Preface	i
I Preliminaries	1
1 Mathematical preliminaries	3
1.1 Linear algebra and diferentiable calculus	3
1.1.1 Minimization of quadratic forms	3
1.1.2 Inverting a 2×2 matrix	4
1.1.3 Inverting matrices defined by blocks (+ matrix inversion lemma)	4
1.1.4 Differential calculus	6
1.2 Concentration inequalities	6
1.2.1 Hoeffding's inequality	7
1.2.2 MacDiarmid's inequality	9
1.2.3 Bernstein's inequality (\blacklozenge)	10
1.2.4 Expectation of the maximum	11
1.2.5 Concentration inequalities for matrices ($\blacklozenge\blacklozenge$)	12
2 Introduction to supervised learning	13
2.1 From training data to predictions	13
2.2 Decision theory	16
2.2.1 Loss functions	16
2.2.2 Risks	17
2.2.3 Bayes risk and Bayes predictor	18

2.3	Learning from data	19
2.3.1	Local averaging	20
2.3.2	Empirical risk minimization	21
2.4	Statistical learning theory	24
2.4.1	Measures of performance	24
2.4.2	Notions of consistency over classes of problems	24
2.5	No free lunch theorems (♦)	25
2.6	Quest for adaptivity	27
3	Linear least-squares regression	29
3.1	Introduction	29
3.2	Least-squares framework	30
3.3	Ordinary least-squares (OLS) estimator	31
3.3.1	Closed-form solution	31
3.3.2	Geometric interpretation	32
3.3.3	Numerical resolution	32
3.4	Statistical analysis of OLS	33
3.5	Fixed design setting	34
3.5.1	Statistical properties of the OLS estimator	35
3.5.2	Experiments	37
3.6	Ridge least-squares regression	38
3.6.1	Experiments	40
3.6.2	Choice of λ	40
3.7	Lower-bound (♦)	41
3.8	Random design analysis	43
3.8.1	Gaussian designs	45
3.8.2	General designs (♦♦)	45
II	Generalization bounds for learning algorithms	47
4	Empirical risk minimization	49

4.1	Convexification of the risk	49
4.1.1	Convex surrogates	50
4.1.2	Geometric interpretation of the support vector machine (♦)	51
4.1.3	Conditional Φ -risk and classification calibration (♦)	53
4.1.4	Relationship between risk and Φ -risk (♦♦)	55
4.2	Risk minimization decomposition	57
4.3	Approximation error	58
4.4	Estimation error	59
4.4.1	Application of MacDiarmid's inequality	59
4.4.2	Easy case I: quadratic functions	60
4.4.3	Easy case II: Finite number of models	60
4.4.4	Beyond finite number models through covering numbers (♦)	61
4.5	Rademacher complexity	63
4.5.1	Symmetrization	63
4.5.2	Lipschitz-continuous losses	64
4.5.3	Ball-constrained linear predictions	66
4.5.4	Putting things together (linear predictions)	66
4.5.5	From constrained to regularized estimation (♦)	67
4.6	Relationship with asymptotic statistics (♦)	68
5	Optimization for machine learning	71
5.1	Optimization in machine learning	71
5.2	Gradient descent	72
5.2.1	Simplest analysis: ordinary least-squares	73
5.2.2	Convex functions and their properties	77
5.2.3	Analysis of GD for strongly convex and smooth functions	79
5.2.4	Analysis of GD for convex and smooth functions (♦)	83
5.2.5	Beyond gradient descent (♦)	84
5.2.6	Non-convex objective functions (♦)	86
5.3	Gradient methods on non-smooth problems	86
5.4	Convergence rate of stochastic gradient descent (SGD)	88

5.4.1	Strongly convex problems (◆)	91
5.4.2	Variance reduction (◆)	93
5.5	Conclusion	96
6	Local averaging methods	99
6.1	Introduction	99
6.2	Local averaging methods	100
6.2.1	Linear estimators	101
6.2.2	Partition estimators	102
6.2.3	Nearest-neighbors	103
6.2.4	Nadaraya-Watson estimator a.k.a. kernel regression (◆)	104
6.3	Generic “simplest” consistency analysis	106
6.3.1	Fixed partition	107
6.3.2	k -nearest neighbor	110
6.3.3	Kernel regression (Nadaraya-Watson) (◆)	112
6.4	Universal consistency (◆)	115
6.5	Adaptivity (◆◆)	117
7	Kernel methods	119
7.1	Introduction	119
7.2	Representer theorem	120
7.3	Kernels	122
7.3.1	Linear and polynomial kernels	124
7.3.2	Translation-invariant kernels on $[0, 1]$	124
7.3.3	Translation-invariant kernels on \mathbb{R}^d	126
7.3.4	Beyond (◆)	129
7.4	Algorithms	130
7.5	Generalization guarantees - Lipschitz-continuous losses	134
7.5.1	Risk decomposition	134
7.5.2	Approximation error for translation-invariant kernels on \mathbb{R}^d	136
7.6	Theoretical analysis of ridge regression (◆◆)	138

7.6.1	Kernel ridge regression	138
7.6.2	Relationship between covariance operators	140
7.6.3	Analysis for well-specified problems	141
7.6.4	Analysis beyond well-specified problems	143
7.6.5	Balancing bias and variance ($\blacklozenge\blacklozenge$)	144
7.7	Experiments	145
8	Sparse methods	147
8.1	Introduction	147
8.1.1	Dedicated proof technique for constrained least-squares	148
8.1.2	Probabilistic and combinatorial lemmas	150
8.2	Variable selection by ℓ_0 penalty	151
8.2.1	Assuming k is known	151
8.2.2	Estimating k (\blacklozenge)	153
8.3	High-dimensional estimation through ℓ_1 -regularization	155
8.3.1	Intuition and algorithms	155
8.3.2	Slow rates	159
8.3.3	Fast rates (\blacklozenge)	161
8.3.4	Zoo of conditions ($\blacklozenge\blacklozenge$)	162
8.4	Experiments	164
8.5	Extensions	165
9	Neural networks	167
9.1	Introduction	167
9.2	Single hidden layer neural network	168
9.2.1	Optimization	170
9.2.2	Estimation error	170
9.3	Approximation properties of single-hidden layer neural networks	172
9.3.1	Link with kernel methods	172
9.3.2	From L_2 -norms to L_1 -norms	174
9.3.3	Variation norm in one dimension	175

9.3.4	Variation norm in arbitrary dimension	179
9.3.5	From the variation norm to a finite number of neurons	181
9.4	Experiments	183
9.5	Global convergence of gradient descent for infinite widths (♦♦)	184
9.6	Extensions	184
III	Special topics	187
10	Implicit bias of gradient descent	189
10.1	Implicit bias of gradient descent	189
10.1.1	Least-squares	190
10.1.2	Separable classification	192
10.2	Double descent	194
10.2.1	The double descent phenomenon	194
10.2.2	Empirical evidence	196
10.2.3	Simplest analysis	197
10.3	Global convergence of gradient descent for two-layer neural networks	199
11	Lower bounds on performance	201
11.1	Statistical lower bounds	202
11.1.1	Minimax lower bounds	202
11.1.2	Reduction to an hypothesis test	203
11.1.3	Information theory	204
11.1.4	Lower-bound on hypothesis testing based on information theory	207
11.1.5	Examples	209
11.1.6	Minimax lower bounds through Bayesian analysis	210
11.2	Optimization lower bounds	212
11.2.1	Convex optimization	212
11.2.2	Non-convex optimization	214
11.3	Lower bounds for stochastic gradient descent (♦)	215

Part I

Preliminaries

Chapter 1

Mathematical preliminaries

Chapter summary

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

In this chapter, I will review the non-elementary mathematical tools that will be used throughout the book: first linear algebra tricks, then concentration inequalities. The chapter can be safely skipped since relevant results will be referenced when needed.

1.1 Linear algebra and differentiable calculus

In this section, we review basic linear algebra and differential calculus results that will be used throughout the book. Using these may usually greatly simplifies computations. As much as possible, matrix notations will be used.

1.1.1 Minimization of quadratic forms

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

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

with minimizer $x_* = A^{-1}b$. If A is in fact not invertible and b is not in the column space of A , then the infimum is $-\infty$.

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

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

This form is exactly Fenchel-Young inequality for quadratic forms (see Chapter 5), and is often used in one dimension in the form $ab \leq \frac{a^2}{2\eta} + \frac{\eta b^2}{2}$, for a well chosen η .

1.1.2 Inverting a 2×2 matrix

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

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

This can be simply checked by multiplying the two matrices or by using Cramer's rule¹. This can be generalized to matrices defined by blocks, as we present next.

1.1.3 Inverting matrices defined by blocks (+ matrix inversion lemma)

The example above may be generalized to matrices of the form $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$, 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 done in block form. Given the two matrices $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ and $N = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$, we may linearly combine lines (with the same coefficients for the two matrices). Once M has been transformed to the identity matrices, N has been transformed to the inverse of M .

We first make the assumption that A is invertible, we use the notation $(M/A) = D - CA^{-1}B$ for the Schur complement of the block A , and also assume that (M/A) is invertible. We get by Gaussian elimination, referring to L_i , $i = 1, 2$ as the two blocks of lines:

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

This shows that

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

¹https://en.wikipedia.org/wiki/Cramer%27s_rule

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

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

By identifying the upper-left and lower-right blocks in Eq. (1.1) and Eq. (1.2), we obtain the identities:

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

which are often referred to as the *matrix inversion lemma*. These are particularly interesting when the blocks A and D have very different sizes, as the inverse of a large matrix may be obtained from the inverse of a small matrix.

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

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

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

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

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

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

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

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

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

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

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

1.1.4 Differential calculus

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

- Quadratic forms: $f(x) = \frac{1}{2}x^\top Ax - b^\top x$, $f'(x) = Ax - b$, $f''(x) = I$
- Least-squares: $f(x) = \frac{1}{2}\|y - Xw\|_2^2$, $f'(x) = X^\top(Xw - y)$, $f''(x) = X^\top X$.

1.2 Concentration inequalities

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

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

$$\text{var}\left(\frac{1}{n} \sum_{i=1}^n Z_i\right) = \frac{1}{n^2} \sum_{i=1}^n \text{var}(Z_i) = \frac{\sigma^2}{n}.$$



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

The equality above can be interpreted as

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

which provides the simplest proof of the law of large numbers when variances exist, and also highlights the convergence of the random variable $\frac{1}{n} \sum_{i=1}^n Z_i$ to a constant.

In order to characterize the deviations in a finer way, there are two classical tools: the *central limit theorem* which states that $\frac{1}{n} \sum_{i=1}^n Z_i$ is approximately normal with mean $\mathbb{E}Z$ and variance σ^2/n . This is an asymptotic statement (formally $\sqrt{n}\left(\frac{1}{n} \sum_{i=1}^n Z_i - \mathbb{E}Z\right)$ converges in distribution to a normal law with mean zero and variance σ^2). Although it gives the right scaling in n , in this textbook, we will look mostly at non-asymptotic results that quantify the deviation for any n .



Below, we will always give versions of inequalities for *averages* of random variables (some authors equivalently consider sums).



Homogeneity: for all non-asymptotic bounds with non-normalized data, it is crucial to make sure the bounds are “dimensionally homogeneous”.

See https://en.wikipedia.org/wiki/Dimensional_analysis.

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

$$\mathbb{P}\left(\bigcup_{f \in \mathcal{F}} A_f\right) \leq \sum_{f \in \mathcal{F}} \mathbb{P}(A_f).$$

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

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

We will only cover the most useful inequalities for machine learning. For more advanced inequalities, see, e.g., [26, 104].

1.2.1 Hoeffding’s inequality

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

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

$$\mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Z_i] \geq t\right) \leq \exp(-2nt^2). \quad (1.3)$$

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

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

Proof: we can simply compute the first two derivatives of $\varphi : s \mapsto \log(\mathbb{E}[\exp(s(Z - \mathbb{E}[Z]))])$, which is a “log-sum-exp” function. We have (readers familiar with probability distributions from exponential families will recognize the usual derivatives of log-partition functions):

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

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 proportional to $e^{s(z - \mathbb{E}[Z])} d\mu(z)$ where $d\mu(z)$ is the distribution of Z . We can thus bound the variance of \tilde{Z} as

$$\text{var}(\tilde{Z}) = \inf_{\mu \in [0, 1]} \mathbb{E}(\tilde{Z} - \mu)^2 \leq \mathbb{E}(\tilde{Z} - 1/2)^2 = \frac{1}{4} \mathbb{E}(2\tilde{Z} - 1)^2 \leq \frac{1}{4} \text{ since } \tilde{Z} \in [-1, 1] \text{ almost surely.}$$

Thus, by Taylor’s formula, $\varphi(s) \leq \frac{s^2}{8}$.

- (b) We recall Markov inequality for any non-negative random variable X and $a > 0$, which states $\mathbb{P}(X \geq a) \leq \frac{1}{a} \mathbb{E}X$. For any $t \geq 0$, and denoting $\bar{Z} = \frac{1}{n} \sum_{i=1}^n Z_i$:

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

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

Note the difference with the central limit theorem, which states that when n goes to infinity, then the probability in Eq. (1.3) is asymptotically equivalent to

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

where $\sigma^2 = \lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=1}^n \text{var}(Z_i)$. The central limit theorem is more precise (as it involves the variance of Z_i 's), but is asymptotic. Bernstein inequalities (see Section 1.2.3) will be in between as they use the variance and an almost sure bound.

Extensions. By just applying the inequality to Z_i 's and $1 - Z_i$'s and using the union bound, we get the following corollary.

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 $t \geq 0$,*

$$P\left(\left|\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Z_i]\right| \geq t\right) \leq 2 \exp(-2nt^2).$$

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

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

- Such an inequality is often used “in the other direction”, that is by starting from the probability and deriving t from it. That is, for any $\delta \in (0, 1)$, with probability greater than $1 - \delta$, we have:

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

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

- 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 https://en.wikipedia.org/wiki/Azuma%27s_inequality.
- Hoeffding's inequality is often applied to so-called “sub-Gaussian” random variables, that is, variables X for which

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

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

- **Exercise 1.3** If Z_1, \dots, Z_n are independent random variables which are sub-Gaussian with constant τ^2 , then, for any $t \geq 0$, $P\left(\left|\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Z_i]\right| \geq t\right) \leq 2 \exp\left(-\frac{nt^2}{2\tau^2}\right)$.

1.2.2 MacDiarmid's inequality

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

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

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

Then

$$\mathbb{P}\left(|f(Z_1, \dots, Z_n) - \mathbb{E}f(Z_1, \dots, Z_n)| \geq t\right) \leq 2 \exp(-2t^2/(nc^2)).$$

Proof (♦) It generalizes Hoeffding's inequality, which corresponds to $f(z) = \frac{1}{n} \sum_{i=1}^n z_i$ of the one-sided inequality, and we will only consider $\mathbb{P}(f(Z_1, \dots, Z_n) - \mathbb{E}f(Z_1, \dots, Z_n) \geq t) \leq \exp(-2t^2/(nc^2))$, which is sufficient to get the two-sided one.

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

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

We have $\mathbb{E}(V_i | Z_1, \dots, Z_{i-1}) = 0$, $|V_i| \leq c$ almost surely, and $f(Z_1, \dots, Z_n) - \mathbb{E}f(Z_1, \dots, Z_n) = \sum_{i=1}^n V_i$. Using the exact same argument as in part (a) of the proof of Hoeffding's inequality, we get for any $s > 0$, $\mathbb{E}(e^{sV_i} | Z_1, \dots, Z_{i-1}) \leq e^{s^2 c^2 / 8}$, and we can obtain a proof with the same steps as part (b) of Hoeffding's inequality by being careful with conditioning. See [26] for details. ■

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

Exercise 1.4 Use Mac Diarmid's inequality to prove a Hoeffding-type bound for vectors, that is, if Z_1, \dots, Z_n are independent vector such that $\|Z_i\|_2 \leq c$ almost surely, then with probability greater than $1 - \delta$, we have

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

1.2.3 Bernstein's inequality (◆)

As mentioned earlier, Hoeffding's inequality only uses an almost sure bound, and not on the variance. Bernstein's inequality allows to use the variance to get a finer result.

Proposition 1.3 (Bernstein's inequality) Let Z_1, \dots, Z_n be n independent random variables such that $|Z_i| \leq c$ almost surely and $\mathbb{E}(Z_i) = 0$. Then

$$\mathbb{P}\left(\left|\frac{1}{n} \sum_{i=1}^n Z_i\right| \geq t\right) \leq 2 \exp\left(-\frac{nt^2}{2\sigma^2 + 2ct/3}\right), \quad (1.4)$$

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

$$\left|\frac{1}{n} \sum_{i=1}^n Z_i\right| \leq \sqrt{\frac{2\sigma^2 \log(1/\delta)}{n}} + \frac{2c \log(1/\delta)}{3n}.$$

Proof The proof is also divided in two parts.

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

Proof: using the entire series development of the exponential, we get:

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

Using the bound $1 + \alpha \leq e^\alpha$, this leads to the desired result.

- (b) With $\sigma_i^2 = \text{var}(Z_i)$, we have:

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

using the lemma above. Thus, by choosing $s = \frac{1}{c} \log(1 + tc/\sigma^2)$, we get a bound equal to $\exp\left(-\frac{n\sigma^2}{c^2} h(ct/\sigma^2)\right)$, with $h(\alpha) = (1 + \alpha) \log(1 + \alpha) - \alpha \geq \frac{\alpha^2}{2+2\alpha/3}$, which leads to the first inequality. The second inequality can be obtained by standard algebra. See [26] for details. ■

Note here that we get the same dependence than for the central limit theorem for small deviations t (and a strict improvement on Hoeffding because the variance is essentially bounded by the squared diameter of the support), while for t large, 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 mean. Then Eq. (1.4) is replaced by

$$\mathbb{P}\left(\left|\frac{1}{n} \sum_{i=1}^n Z_i - \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Z_i]\right| \geq t\right) \leq 2 \exp\left(-\frac{nt^2}{2\sigma^2 + 2ct/3}\right)$$

Exercise 1.5 (♦) *Prove the inequality above.*

1.2.4 Expectation of the maximum

Concentration inequalities bound the deviation from the expectation. Often, computing the expectation is the tricky part, in particular for maxima of random variables. In a nutshell, taking the maximum of n bounded random variable 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.

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

⚠ The variables do not need to be independent.

Proposition 1.4 (expectation of the maximum) *If Z_1, \dots, Z_n are (potentially dependent) random variables which are sub-Gaussian with constant τ^2 , then*

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

Proof We have:

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

using the definition of sub-Gaussianity in Section 1.2.1. ■

1.2.5 Concentration inequalities for matrices (◆◆)

It turns out the concentration inequalities that were shown in this chapter applies as well to matrices with the positive semi-definite order. The following bounds are adapted from [101] and presented without proofs.

Proposition 1.5 (matrix Hoeffding bound) [101, Theorem 1.3] *Given n independent symmetric matrices $M_i \in \mathbb{R}^{d \times d}$, such that for all $i \in \{1, \dots, n\}$, $\mathbb{E}[M_i] = 0$, $M_i^2 \preceq C_i^2$ almost surely. Then for all $t \geq 0$,*

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

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

Proposition 1.6 (matrix Bernstein bound) [101, Theorem 1.4] *Given n independent symmetric matrices $M_i \in \mathbb{R}^{d \times d}$, such that for all $i \in \{1, \dots, n\}$, $\mathbb{E}[M_i] = 0$, $\lambda_{\max}(M_i) \leq c$ almost surely. Then for all $t \geq 0$,*

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

for $\sigma^2 = \lambda_{\max}\left(\frac{1}{n} \sum_{i=1}^n M_i^2\right)$.

We can make the following observations:

- Note the similarity with the corresponding bound for scalar random variables when $d = 1$.
- These bounds applies as well to rectangular matrices $M_i \in \mathbb{R}^{d_1 \times d_2}$ by considering the so-called Jordan-Wielandt matrices, which are the symmetric matrices $\widetilde{M}_i = \begin{pmatrix} 0 & M_i \\ M_i^\top & 0 \end{pmatrix} \in \mathbb{R}^{(d_1+d_2) \times (d_1+d_2)}$, whose eigenvalues are plus and minus the singular values of M_i [99, Theorem 4.2]. For the Hoeffding bound, if $\|M_i\|_{\text{op}} \leq c$ almost surely for all $i \in \{1, \dots, n\}$, then $\mathbb{P}(\|\frac{1}{n} \sum_{i=1}^n M_i\|_{\text{op}} \geq t) \leq (d_1 + d_2) \cdot \exp\left(-\frac{nt^2}{8c^2}\right)$, while the Bernstein bound leads to $\mathbb{P}(\|\frac{1}{n} \sum_{i=1}^n M_i\|_{\text{op}} \geq t) \leq (d_1 + d_2) \cdot \exp\left(-\frac{nt^2/2}{\sigma^2 + ct/3}\right)$ with $\sigma^2 = \max\left\{\lambda_{\max}\left(\frac{1}{n} \sum_{i=1}^n M_i^\top M_i\right), \lambda_{\max}\left(\frac{1}{n} \sum_{i=1}^n M_i M_i^\top\right)\right\}$.
- MacDiarmid's inequality can also be extended [101, Corollary 7.5].

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.

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 previous 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, in particular when it comes to providing theoretical guarantees. The focus on algorithms and computational scalability is arguably stronger within machine learning (but also present in statistics), while the focus on models and their interpretability beyond their predictive performance is more prominent within statistics (but also present in machine learning).

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

- **Inputs $x \in \mathcal{X}$:** they can be images, sounds, videos, text, proteins, sequence of DNA bases, web pages, social network activity, sensors from industry, financial time series, etc. The set \mathcal{X} may thus have a variety of structures that can be leveraged. All learning methods that we present in this textbook will use at one point a vector-space representation of inputs, either by building an explicit mapping from \mathcal{X} to a vector space (such as \mathbb{R}^d), or implicitly by using a notion of pairwise dissimilarity or similarity between pairs of inputs. The choice of these representations

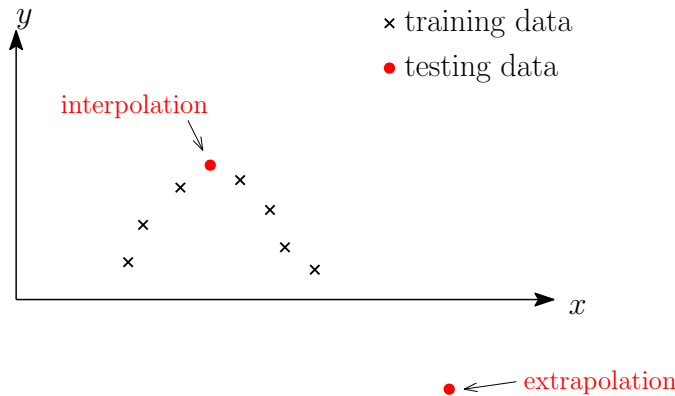
are highly domain dependent, though we note that (a) common topologies are encountered in many diverse areas (such as sequences, two-dimensional or three-dimensional objects) and thus common tools are used, and (b) learning these representations in active area of research (see discussion in Chapter 7 and Chapter 9).

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

- **Outputs $y \in \mathcal{Y}$:** the most classical examples are binary labels $\mathcal{Y} = \{0, 1\}$, multcategory classification problems with $\mathcal{Y} = \{1, \dots, k\}$, and classical regression with real responses/outputs $\mathcal{Y} = \mathbb{R}$. These will be the main examples we treat in most of the book. Note however that most of the concepts extend to the more general *structured prediction* set-up, where more general structured outputs (e.g., graph prediction, visual scene analysis, source separation) can be considered (see [83, 84] and the many references therein).

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

- The label y is not a deterministic function of x : given $x \in \mathcal{X}$, the outputs are noisy, as $y = f(x) + \varepsilon$, e.g., with (potentially non-additive) noise ε due to diverging views between labellers, or dependence on random external unobserved quantities (that is $y = f(x, z)$, z random).
- The prediction function f may be quite complex, highly non-linear when \mathcal{X} is a vector space, and even hard to define when \mathcal{X} is not a vector space.
- Only a few x 's are observed: we thus need interpolation and potentially extrapolation (see below for an illustration for $\mathcal{X} = \mathcal{Y} = \mathbb{R}$), and therefore overfitting (predicting well on the training data but not as well on the testing data) is always a possibility.



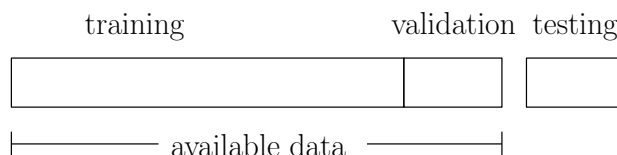
- The input space \mathcal{X} may be very large, that is, with high dimension when this is a vector space. This leads to both computational issues (scalability) and statistical issues (generalization to unseen data). One usually refers to the *curse of dimensionality*.
- There may be a weak link between training and testing distributions.
- 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, and the criterion is to minimize the expectation of some “performance” measure with respect to the distribution of the test data. The main assumption is that the random variables (x_i, y_i) are independent and identically distributed (i.i.d.) with the same distribution as the testing distribution. In this course, we will ignore the potential mismatch between train and test distributions (although this is an important research topic as in most applications training data are not i.i.d. from the same distribution as the test data).

A machine learning algorithm \mathcal{A} is 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),
- the testing set, to evaluate the performance of the final model (formally, the test set can only be used once!)



Cross-validation is often preferred to use a maximal amount of training data, and reduce the variability of the validation procedure: the available data are divided in k folds (typically $k = 5$ or 10), and all models are estimated k times, each time choosing a different fold as validation data (pink data below), and averaging the k obtained error measures.



“Debugging” a machine learning implementation is often an art: on top of classical bugs, the learning method may not predict well enough on testing data. This is where theory can be useful, to understand when a method is supposed to work or not.

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, if we have a perfect knowledge of the underlying probability distribution of the data, what should be done? We will thus introduce the concept of *loss function*, *risk*, and “*Bayes*” *predictor*.

We consider a fixed (testing) distribution $dp_{x,y}(x,y)$ on $\mathcal{X} \times \mathcal{Y}$, with marginal distribution $dp_x(x)$ 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 dp , to denote $dp_{X,Y}$ and dp_X , where the context can always make the definition unambiguous. For example, when $f : \mathcal{X} \rightarrow \mathbb{R}$ and $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, we have $\mathbb{E}f(x) = \int_{\mathcal{X}} f(x)dp(x)$ and $\mathbb{E}g(x,y) = \int_{\mathcal{X} \times \mathcal{Y}} g(x,y)dp(x,y)$.

⚠ We ignore on purpose measurability issues.

2.2.1 Loss functions

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

⚠ Some authors swap y and z in the definition above. ⚠ Some related research communities (e.g., economics) use the concept of “utility”, which is then maximized.

The loss function is only concerned with the output space \mathcal{Y} . The main examples are:

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



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

- **Multi-category classification:** $\mathcal{Y} = \{1, \dots, k\}$, and $\ell(y,z) = 1_{y \neq z}$ (“0-1” loss).
- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y,z) = (y - z)^2$ (square loss). The absolute loss $\ell(y,z) = |y - z|$ is often used for “robust” estimation (since the penalty for large errors is smaller).
- **Structured prediction:** while this textbook focuses primarily on the examples above, there are many practical problems where \mathcal{Y} is more complicated, with associated algorithms and theoretical results. For examples, when $\mathcal{Y} = \{0, 1\}^k$ (leading to multi-label classification), the Hamming loss $\ell(y,z) = \sum_{j=1}^k 1_{y_j \neq z_j}$ is commonly used; also ranking problems involve losses on permutations. See, e.g., many examples in [82] and references therein.

Throughout the textbook, we will assume that the loss function is given to us. Note that in practice, the loss function is imposed by the final user, as this is the way models will be evaluated. Clearly, a single real number may not be enough to characterize the entire prediction behavior (think of binary classification, with the two types of errors, false positives and false negatives, where the concept of “ROC curve” is commonly used to characterize the two types of errors), but for simplicity, we will stick to a single loss function ℓ .

2.2.2 Risks

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

Definition 2.1 (expected risk) *Given a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, and a distribution $dp(x, y)$, the expected risk of a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as:*

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

The risk depends on the distribution $dp = dp_{x,y}$ on (x, y) . We sometimes use the notation $\mathcal{R}_{dp}(f)$ to make it explicit. The expected risk is the main performance criterion we will use in this textbook.



Be careful with the randomness or lack thereof of f : when performing learning from data, f will depend on the random training data and not on the testing data, and thus $\mathcal{R}(f)$ is typically random because of the dependence on the training data. However, as a function on functions, the 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 function $f : \mathcal{X} \rightarrow \mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, and data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, the empirical risk of a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as:*

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

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

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

- **Binary classification:** $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$), and $\ell(y, z) = 1_{y \neq z}$ (“0-1” loss). We can express the risk as $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$. This is simply the probability of making a mistake on the testing data, while the empirical risk is the proportion of mistake on the training data.
- **Multi-category classification:** $\mathcal{Y} = \{1, \dots, k\}$, and $\ell(y, z) = 1_{y \neq z}$ (“0-1” loss). We can also express the risk as $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$. This is also the probability of making a mistake.
- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). The risk is then $\mathcal{R}(f) = \mathbb{E}(y - f(x))^2$.



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

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 data)?

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

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

which we can rewrite

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

⚠ In order to make the distinction between the random variable x and a value it may take, we use the notation x' .

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

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

which leads to

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

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 $r(z|x') = \mathbb{E}(\ell(y, z)|x = x')$. We can therefore **consider all x' as being treated independently**. This leads to the following propositions.

Proposition 2.1 (Bayes predictor and Bayes risk) *The expected risk is minimized at a Bayes predictor $f^* : \mathcal{X} \rightarrow \mathcal{Y}$ satisfying for all $x' \in \mathcal{X}$, $f^*(x') \in \arg \min_{z \in \mathcal{Y}} \mathbb{E}(\ell(y, z)|x = x') = \arg \min_{z \in \mathcal{Y}} r(z|x')$. The Bayes risk \mathcal{R}^* is the risk of all Bayes predictors and is equal to*

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

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

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

Therefore, machine learning is “trivial”: *given* the distribution $y|x$ for any x , the optimal predictor is known. The difficulty will be that this distribution is unknown.

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

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

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

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

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

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

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

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

$$f^*(x') \in \arg \min_{z \in \mathbb{R}} \mathbb{E}[(y - z)^2 | x = x'] = \arg \min_{z \in \mathbb{R}} \left\{ \mathbb{E}[(y - \mathbb{E}(y | x = x'))^2 | x = x'] + (z - \mathbb{E}(y | x = x'))^2 \right\}.$$

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

Exercise 2.1 What is the Bayes predictor for regression with the absolute loss $\ell(y, z) = |y - z|$?

Exercise 2.2 (♦) We consider a random prediction rule where we predict from the probability distribution of y given $x = x'$. When is this achieving the Bayes risk?

2.3 Learning from data

The decision theory framework outlined in the previous section gives a test performance criterion and optimal predictors, but it depends on the full knowledge of the test distribution $dp(x, y)$. 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.

There are two main classes of prediction algorithms that will be studied in this textbook:

- (1) Local averaging (Chapter 6).
- (2) Empirical risk minimization (Chapters 3, 4, 7, 8, 9).

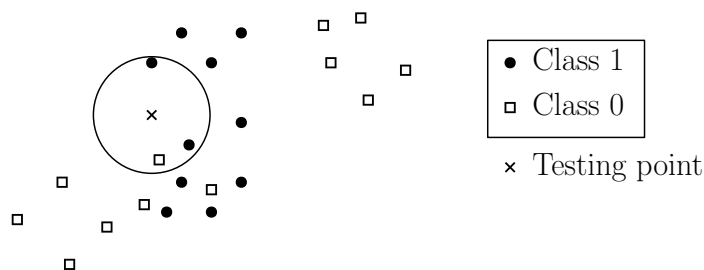
Note that there are prediction algorithms that do not fit exactly into one of these two categories, such as boosting or ensemble classifiers.

2.3.1 Local averaging

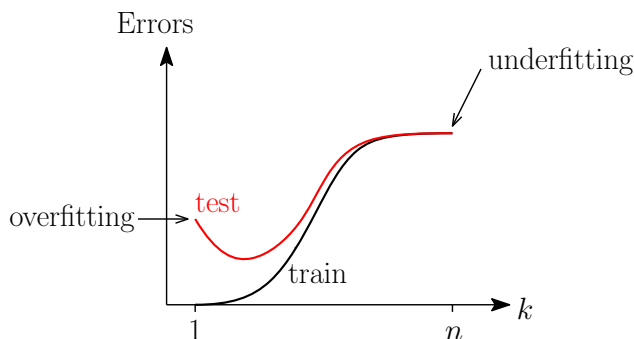
The goal here is to try to approximate / emulate the Bayes predictor, e.g., $f^*(x') = \mathbb{E}(y|x = x')$ for least-squares regression, from empirical data. This is done often by explicit/implicit estimation of the conditional distribution by *local averaging* (k -nearest neighbors, which is used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

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

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



- Pros: (a) no optimization or training, (b) often easy to implement, (c) can get very good performance in low dimensions (in particular for non-linear dependences between x and y).
- Cons: (a) slow at query time: must pass through all training data at each testing point (there are algorithmic tools to reduce complexity, see Chapter 6), (b) bad for high-dimensional data (curse of dimensionality, more on this in Chapter 6), (c) the choice of local distance function is crucial, (d) the choice of “width” parameters (or k) has to be performed.
- Plot of training error and testing errors as a function 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 2.3** How would the curve move when n increases?

2.3.2 Empirical risk minimization

Consider a parameterized family of prediction functions $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ for $\theta \in \Theta$ and minimize the empirical risk with respect to $\theta \in \Theta$:

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

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

The most classical example is linear least-squares regression (studied at length in Chapter 3), where we minimize

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

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

- Pros: (a) can be relatively easy to optimize (e.g., least-squares with simple derivation and numerical algebra, see Chapter 3), many algorithms available (mostly based on gradient descent, see Chapter 5), (b) can be applied in any dimension (if a reasonable feature vector is available).
- Cons: (a) can be relatively hard to optimize (e.g., neural networks), (b) need a good feature vector for linear methods, (c) dependence on parameters can be complex (e.g., neural networks), (d) need some capacity control to avoid overfitting, (e) how to parameterize functions with values in $\{0, 1\}$ (see Chapter 4 for the use of convex surrogates)?

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

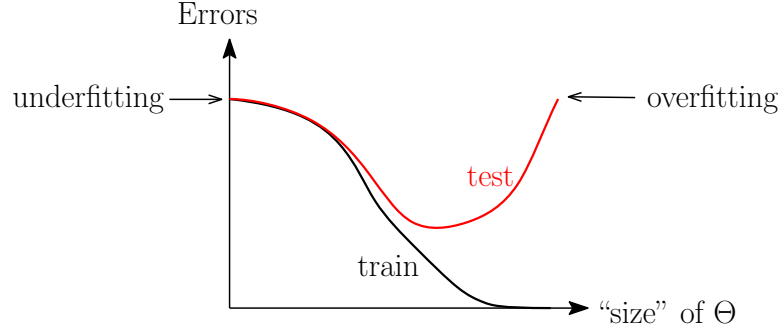
- Risk decomposition in estimation error + approximation error : given any $\hat{\theta} \in \Theta$, we can write the excess risk of $f_{\hat{\theta}}$ as:

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

The approximation error 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 deterministic function, that characterizes the modelling assumptions made by the models. When Θ grows, the approximation error goes down, to zero if arbitrary functions can be approximated arbitrary well by the functions f_θ . It is also independent of n .

The estimation error is typically random, because the function $f_{\hat{\theta}}$ is random. It is typically decreasing in n , and usually goes up 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, for θ' a minimizer on Θ of the expected risk $\mathcal{R}(f_{\theta'})$:

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

The uniform deviation grows with the “size” of Θ , and usually decays with n . See more details in Chapter 4.

Capacity control. In order to avoid overfitting, we need to make sure that the set of allowed functions is not too large, by typically reducing number of parameters, or by restricting the norm of predictors (thus by reducing the “size” of Θ): this typically leads to constrained optimization, and allows for risk decompositions as done above.

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

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

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

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

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



There is a difference between parameters (e.g., θ) learned on the training data and hyperparameters (e.g., λ) learned on the validation data.

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

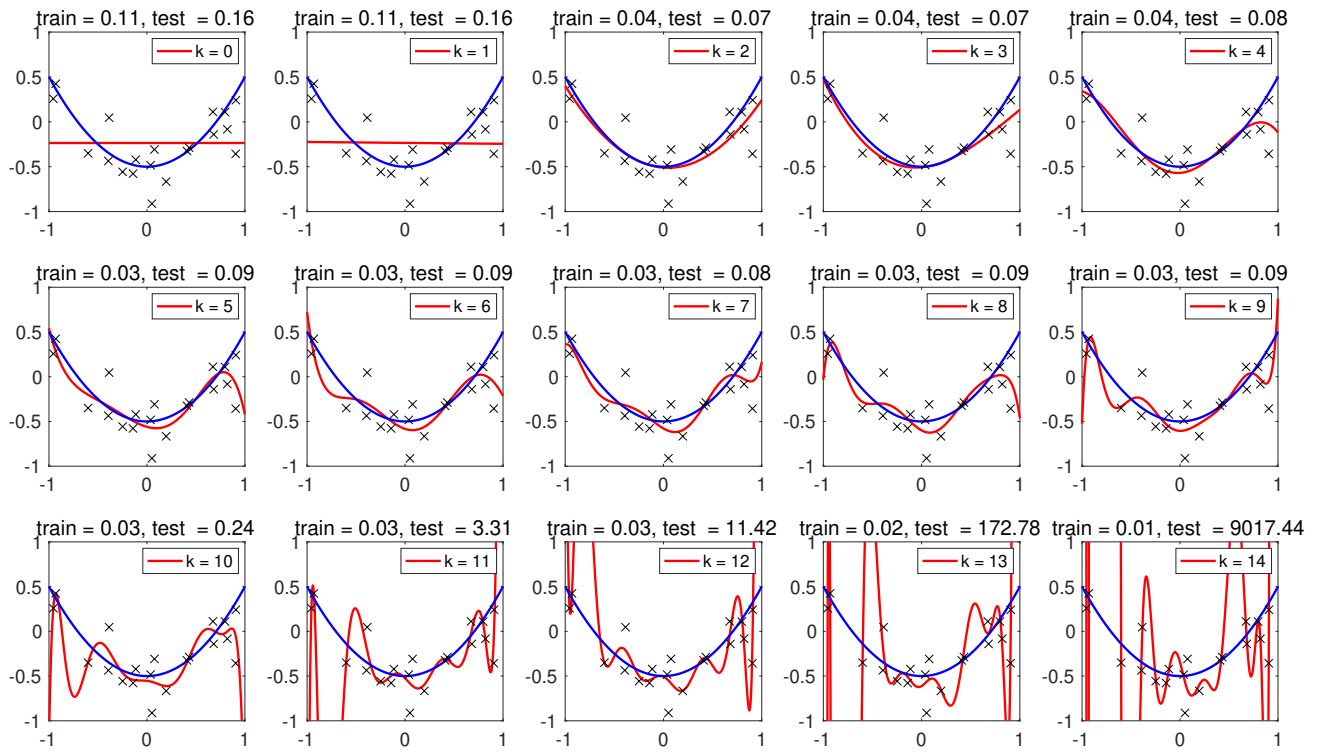


Figure 2.1: Polynomial regression with increasing orders. Plots of estimated functions, with training and testing errors.

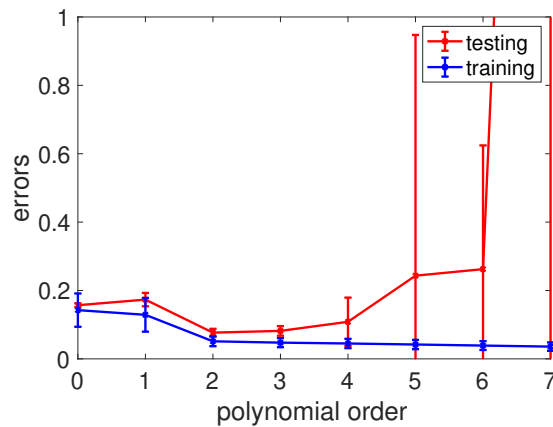


Figure 2.2: Polynomial regression with increasing orders. Plots of training and testing errors with error bars (obtained from 32 replications).

2.4 Statistical learning theory

The goal of learning theory is to provide some guarantees of performance on unseen data. A common assumption is that the data $\mathcal{D}_n(dp) = \{(x_1, y_1), \dots, (x_n, y_n)\}$ is obtained as independent and identically distributed (i.i.d.) observations, from some unknown distribution dp from a family \mathcal{P} .

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

$$\mathcal{R}_{dp}(\mathcal{A}(\mathcal{D}_n(dp))) - \mathcal{R}_{dp}^*$$

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

2.4.1 Measures of performance

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

- *Expected error*: we measure performance as

$$\mathbb{E} \left[\mathcal{R}_{dp}(\mathcal{A}(\mathcal{D}_n(dp))) \right],$$

where the expectation is with respect to the training data.

An algorithm \mathcal{A} is said *consistent in expectation* for the distribution dp , if $\mathbb{E} \left[\mathcal{R}_{dp}(\mathcal{A}(\mathcal{D}_n(dp))) \right] - \mathcal{R}_{dp}^*$ goes to zero when n tends to infinity. In this course, we will use primarily this notion of consistency.


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

$$\mathbb{P} \left(\left| \mathcal{R}_{dp}(\mathcal{A}(\mathcal{D}_n(dp))) - \mathcal{R}_{dp}^* \right| \leq \varepsilon \right) \geq 1 - \delta.$$

The crux is to find ε which is as small as possible (typically as a function of δ). 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 said *universally consistent* (in expectation) if for all distributions $dp = dp_{x,y}$ on (x, y) the algorithm \mathcal{A} is consistent in expectation for the distribution dp .

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

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

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

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

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

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

- Upper-bounding the optimal performance: One given algorithm with a convergence proof provides an upper-bound.
- Lower-bounding the optimal performance: in some set-ups, it is possible to show that the infimum over all algorithms is greater than a certain quantity. Machine learners are happy when upper-bounds and lower-bounds match (up to constant factors).

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

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



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

2.5 No free lunch theorems (◆)

Although it may be tempting to define the optimal learning algorithm that works optimally for all distributions, this is impossible. In other words, learning is not possible without assumptions. See [41, Chapter 7] for details.

The following theorem shows that for any algorithm, for a fixed n , there is a data distribution that makes the algorithm useless.

Theorem 2.1 (no free lunch - fixed n) *Consider the binary classification with 0 – 1 loss, with \mathcal{X} infinite. Let \mathcal{P} denote the set of all probability distributions on $\mathcal{X} \times \{0, 1\}$. For any $n > 0$ and learning algorithm \mathcal{A} ,*

$$\sup_{dp \in \mathcal{P}} \mathbb{E} \left[\mathcal{R}_{dp}(\mathcal{A}(\mathcal{D}_n(dp))) \right] - \mathcal{R}_{dp}^* \geq 1/2.$$

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

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

Denoting $\hat{f}_{\mathcal{D}_n} = \mathcal{A}(\mathcal{D}_n(dp))$ the classifier, and $S(r) = \mathbb{E}[\mathcal{R}_{dp}(\hat{f}_{\mathcal{D}_n})]$ 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 distribution dq on r , in particular the uniform distribution (each r_j an independent unbiased Bernoulli variable). Then

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

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

Then, we get:

$$\begin{aligned} \mathbb{E}_{r \sim dq(r)} S(r) &= \mathbb{E}[\mathbb{P}(\hat{f}_{\mathcal{D}_n}(x) \neq r_x | x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n})] \text{ by the law of total expectation,} \\ &\geq \mathbb{E}[\mathbb{P}(\hat{f}_{\mathcal{D}_n}(x) \neq r_x \ \& \ x \notin \{x_1, \dots, x_n\} | x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n})] \\ &\hspace{15em} \text{by monotonicity of probabilities,} \\ &= \mathbb{E}\left[\frac{1}{2} \mathbb{P}(x \notin \{x_1, \dots, x_n\} | x_1, \dots, x_n, r_{x_1}, \dots, r_{x_n})\right], \end{aligned}$$

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

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

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

A caveat is that the hard distribution may depend on n (and, from the proof, it takes k values, with k tending to infinity). The following theorem is given without proof and is much “stronger” [41, Theorem 7.2], as it more convincingly shows that learning can be arbitrarily slow without assumption (note that the earlier one is not a corollary of the later one).

Theorem 2.2 (no free lunch - sequence of errors) *Consider the binary classification with 0 – 1 loss, with \mathcal{X} infinite. Let \mathcal{P} denote the set of all probability distributions on $\mathcal{X} \times \{0, 1\}$. For any decreasing sequence a_n tending to zero and such that $a_1 \leq 1/16$, for any learning algorithm \mathcal{A} , there exists $dp \in \mathcal{P}$, such that for all $n \geq 1$:*

$$\mathbb{E}[\mathcal{R}_{dp}(\mathcal{A}(\mathcal{D}_n(dp)))] - \mathcal{R}_{dp}^* \geq a_n.$$

2.6 Quest for adaptivity

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

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

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

Note that another form of adaptivity which is often considered is to 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.

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 settings: when the inputs are assumed deterministic and $d > n$, the excess risk is equal to $\sigma^2 d/n$.
- 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 performance: under well-specification, the rate $\sigma^2 d/n$ is unimprovable.

3.1 Introduction

In this chapter, we introduce and analyze linear least-squares regression, a tool that can be traced back to Legendre (1805) and Gauss (1809)—see https://en.wikipedia.org/wiki/Least_squares#The_method for an interesting discussion and the claim that Gauss knew about it already in 1795.

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

- It already captures many of the concepts in learning theory, such as the bias-variance trade-off, as well as the dependence of generalization performance on the underlying dimension of the problem, or on dimension-less quantities.
- Because of its simplicity, many results can be easily derived without the need for complicated mathematics (simple linear algebra for the simplest results).
- Using non-linear features, it can be extended to arbitrary non-linear predictions (see kernel methods in Chapter 7).

In subsequent chapters, we will extend many of these results beyond least-squares.

3.2 Least-squares framework

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

In this chapter, we consider empirical risk minimization. We choose a parameterized family of prediction functions $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ for $\theta \in \Theta$ and minimize the empirical risk

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

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

Least-squares regression can be carried out with parameterizations of the function f_θ which may be non-linear in the parameter θ . 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, and which we take to be \mathbb{R}^d for simplicity.



Being linear in x or linear θ is different!

While we assume linearity in the parameter θ , nothing forces $f_\theta(x)$ to be linear in the input x . In fact, even the concept of linearity may be meaningless if \mathcal{X} is not a vector space. Through Riesz representation theorem, for any $x \in \mathcal{X}$, there exists a vector in \mathbb{R}^d , which we denote $\varphi(x)$, such that

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

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

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

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

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

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

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 notations. We strongly advise against it as it leads to long and error-prone formulas.

3.3 Ordinary least-squares (OLS) estimator

We make the assumption 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 “over-determined”, and we must have $d \leq n$. Equivalently, we assume that $\Phi^\top \Phi \in \mathbb{R}^{d \times d}$ is invertible.

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

3.3.1 Closed-form solution

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

Proposition 3.1 *When 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 (non-centered¹) empirical covariance matrix $\hat{\Sigma} := \frac{1}{n} \Phi^\top \Phi \in \mathbb{R}^{d \times d}$; we have $\hat{\theta} = \frac{1}{n} \hat{\Sigma}^{-1} \Phi^\top y$.

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

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

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

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

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

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

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

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

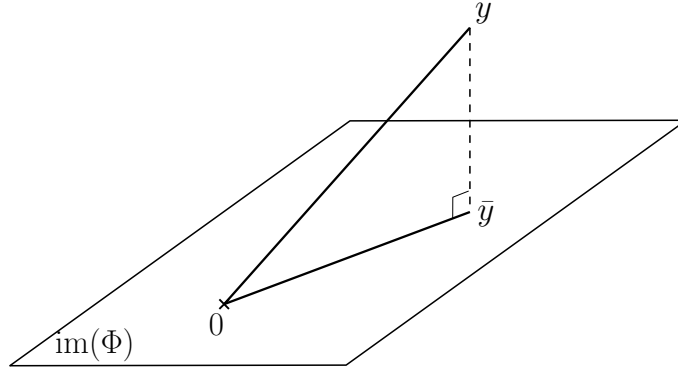
3.3.2 Geometric interpretation

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

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

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

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



3.3.3 Numerical resolution

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

QR factorization. The QR decomposition factorizes the matrix Φ as $\Phi = QR$ where $Q \in \mathbb{R}^{n \times d}$ has orthonormal columns and $R \in \mathbb{R}^{d \times d}$ is upper triangular (see [47]). Computing a QR decomposition is faster and more stable than inverting a matrix. One has

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

It only remains to solve a triangular linear system which is easy. The overall running time complexity remains $O(d^3)$. The conjugate gradient algorithm can also be used (see [47] for details).

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

$$\theta_{k+1} = \theta_k - \gamma \hat{\mathcal{R}}'(\theta_k) \quad \text{for } k \geq 0,$$

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

3.4 Statistical analysis of OLS

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

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

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

This assumption allows a complete analysis with basic linear algebra. It is justified in some settings, e.g., when the input is 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.

In the fixed design setting, no attempts are made to generalize to unseen input points x , and we want to estimate well a label vector y resampled from the same distribution than the observed y . The risk in Eq. (3.3) is often called the *in-sample prediction error*.

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

Relationship to maximum likelihood estimation. If, in the fixed design setting, we make the stronger assumption that the noise is Gaussian with mean zero and variance 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 fixed). Indeed, the density / likelihood of y is, using independence and the density of the normal distribution:

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

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 What is $\tilde{\sigma}^2$ the maximum likelihood of σ^2 ?

3.5 Fixed design setting

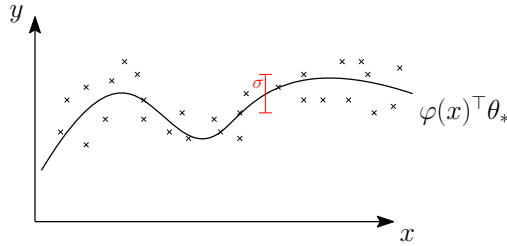
We now assume that Φ is deterministic, and as before, we assume that $\hat{\Sigma} = \frac{1}{n}\Phi^\top\Phi$ is invertible. Any kind of guarantee requires assumptions about how the data are generated. We assume that:

- there exists a vector $\theta_* \in \mathbb{R}^d$ such that the relationship between input and output is for $i \in \{1, \dots, n\}$

$$y_i = \varphi(x_i)^\top \theta_* + \varepsilon_i. \quad (3.4)$$

- For all $i \in \{1, \dots, n\}$, ε_i are independent of 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 which are due to unobserved factors or to noise. The “homoscedasticity” assumption above, where the noise variances are uniform, is made for simplicity (and allows for the later bound $\sigma^2 d/n$ bound to be an equality). Note that to prove upper-bounds in performance, we could also only assume that $\mathbb{E}[\varepsilon_i^2] \leq \sigma^2$ 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 $\varphi(x_i)^\top \theta_*$.



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

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

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

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

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 now $\hat{\theta}$ is a random variable (such as an estimator of θ_*), then

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

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

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

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 that 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:

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

(NB: this is also a simple application of $\mathbb{E}\|z - a\|_M^2 = \|\mathbb{E}z - a\|_M^2 + \mathbb{E}\|z - \mathbb{E}z\|_M^2$ 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 we thus need to compute 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 $\text{Var}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta} - \theta_*)(\hat{\theta} - \theta_*)^\top\right] = \frac{\sigma^2}{n}\widehat{\Sigma}^{-1}$; $\widehat{\Sigma}^{-1}$ is often called the precision matrix.*

Proof

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

$$\text{var}(\hat{\theta}) = \mathbb{E} \left[(\Phi^\top \Phi)^{-1} \Phi^\top \varepsilon \varepsilon^\top \Phi (\Phi^\top \Phi)^{-1} \right] = \sigma^2 (\Phi^\top \Phi)^{-1} (\Phi^\top \Phi) (\Phi^\top \Phi)^{-1} = \sigma^2 (\Phi^\top \Phi)^{-1} = \frac{\sigma^2}{n} \hat{\Sigma}^{-1}.$$

■

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

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

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

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

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


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

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

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

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

We can make the following observations:

-  In the fixed design setting, the expectation over ε appears twice: (1) in the definition of the risk of some θ in Eq. (3.3), and when taking expectation over the data in Eq. (3.5).
- **Exercise 3.2** Show that the expected empirical risk $\mathbb{E}[\hat{\mathcal{R}}(\hat{\theta})]$ is equal to $\mathbb{E}[\hat{\mathcal{R}}(\hat{\theta})] = \frac{n-d}{n} \sigma^2$. In particular, when $n > d$, deduce that an unbiased estimator of the noise variance σ^2 is given by $\frac{\|Y - \Phi \hat{\theta}\|_2^2}{n-d}$.
- In the exercise above, we have an expression of the expected training error, which is equal to $\frac{n-d}{n} \sigma^2 = \sigma^2 - \frac{d}{n} \sigma^2$, while the expected testing error is $\sigma^2 + \frac{d}{n} \sigma^2$. We thus see that in context of least-squares, the training error underestimates (in expectation) the testing error by a factor of $2\sigma^2 d/n$, which characterizes the amount of overfitting. This difference can be used to perform model selection (see https://en.wikipedia.org/wiki/Mallows%27s_Cp).

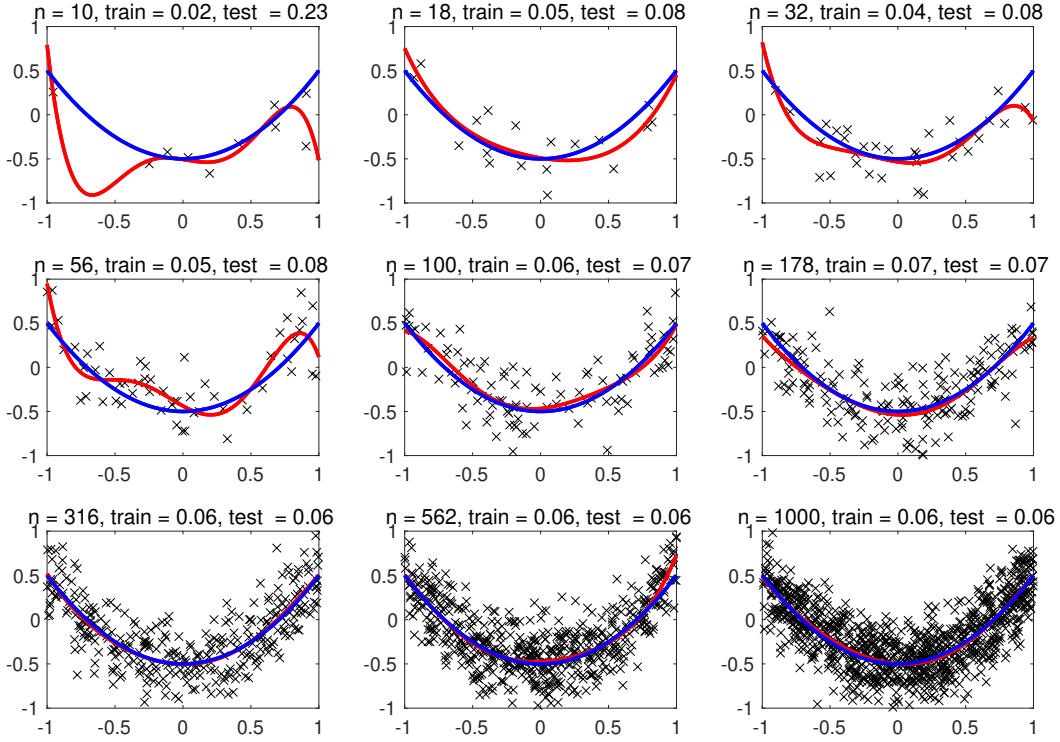


Figure 3.1: Polynomial regression with varying number of observations.

- In the fixed design setting, OLS thus leads to unbiased estimation, with an excess risk of $\sigma^2 d/n$.
- On the positive side, the math is very simple, and as we will show below, 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 closed to n (let alone problems where $d > n$ or d much larger than n). Regularization (ridge in this chapter or ℓ_1 in Chapter 8) will come to the rescue.
- This is only for the fixed design setting. We consider the random design setting below, which is a bit more involved mathematically, mostly because of the presence of $\hat{\Sigma}^{-1}$ which does not cancel anymore (leading to the term $\hat{\Sigma}^{-1}\Sigma$).

3.5.2 Experiments

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

We can now plot in Figure 3.2 the expected excess risk as a function of n , estimated by 32 replications of the experiment, together with the bound. In the right plot, we consider the random design setting (generalization error), while in the left plot we consider the fixed design setting (in-sample error). Notice

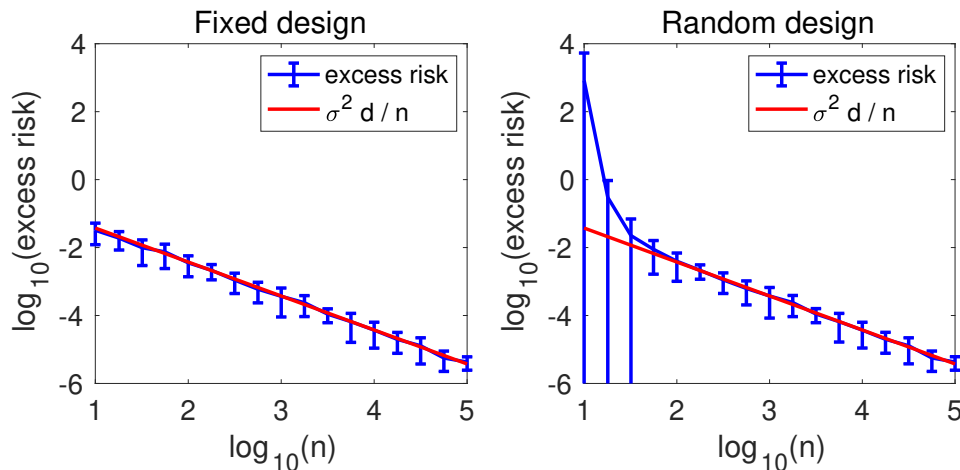


Figure 3.2: Convergence rate for polynomial regression with error bars (obtained from 32 replications by adding/subtracting standard deviations), plotted in logarithmic scale, with fixed design (left plot) and random design (right plot). The large error bars for small n in the right plot are due to the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

the closeness of the bound for all n for the fixed design (as predicted by our bounds), while this is only true for n large enough in the random design setting.

3.6 Ridge least-squares regression

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

Several solutions exist to fix these issues. The most common is to regularize the least-squares objective, either by adding an ℓ_1 -penalty $\|\theta\|_1$ to the empirical risk (leading to “Lasso” regression, see Chapter 8) or $\|\theta\|_2^2$ (leading to *ridge* regression, as done in this chapter and also Chapter 7).

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

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

The ridge regression estimator can be obtained in closed form

Proposition 3.6 We recall that $\hat{\Sigma} = \frac{1}{n} \Phi^\top \Phi \in \mathbb{R}^{d \times d}$. We have

$$\hat{\theta}_\lambda = \frac{1}{n} (\hat{\Sigma} + \lambda I)^{-1} \Phi^\top y.$$

Proof As for proof of Proposition 3.1, we can compute the gradient of the objective function, which is equal to $\frac{2}{n}(\Phi^\top \Phi - \Phi^\top y) + 2\lambda\theta$. Setting it to zero leads to the estimator. ■

As for the OLS estimator, we can analyze the statistical properties of this estimator under the linear model and fixed design assumptions. See Chapter 7 for an analysis for 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 $\hat{\theta}_\lambda = \frac{1}{n}\Sigma_\lambda^{-1}\Phi^\top y$ has the following excess risk*

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

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

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

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

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

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

We can make the following observations:

- The result above is also a bias / variance decomposition with the bias term equal to $B = \lambda^2 \theta_*^\top (\hat{\Sigma} + \lambda I)^{-2} \hat{\Sigma} \theta_*$, and the variance term equal to $V = \frac{\sigma^2}{n} \text{tr}[\hat{\Sigma}^2 (\hat{\Sigma} + \lambda I)^{-2}]$.
- The bias term is increasing in λ and equal to zero for $\lambda = 0$ if $\hat{\Sigma}$ is invertible, while when λ goes to infinity, the bias goes to $\theta_*^\top \hat{\Sigma} \theta_*$. 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 equal to $\sigma^2 d/n$ for $\lambda = 0$ if $\hat{\Sigma}$ 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 $\text{tr}[\hat{\Sigma}^2 (\hat{\Sigma} + \lambda I)^{-2}]$ is often called the “degrees of freedom”, and is often considered as an implicit number of parameters. It can be expressed as $\sum_{j=1}^d \frac{\lambda_j^2}{(\lambda_j + \lambda)^2}$, where $(\lambda_j)_{j \in \{1, \dots, d\}}$

are the eigenvalues of $\hat{\Sigma}$. This quantity will be very important in the analysis of kernel methods in Chapter 7.

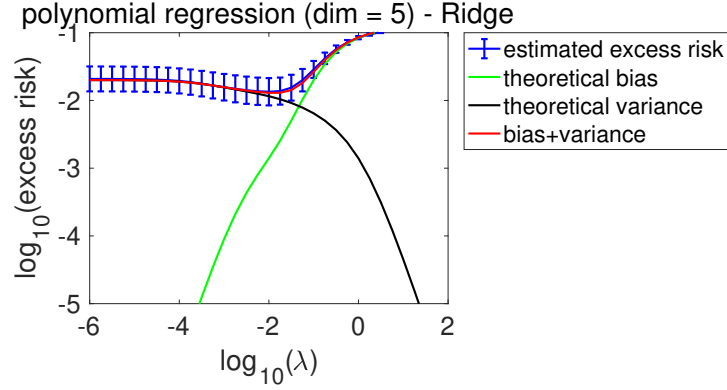


Figure 3.3: Bias/variance trade-offs for ridge regression.

- Observe how this converges to the OLS estimator (when it is defined) as $\lambda \rightarrow 0$.
- In most cases, $\lambda = 0$ is not the optimal choice, that is biased estimation (with controlled bias) is preferable to unbiased estimation.

3.6.1 Experiments

With the same polynomial regression set-up as above, with $k = 10$, we can plot the various quantities above as a function of λ . We can see the monotonicity of bias and variance with respect to λ as well as the presence of an optimal choice of λ . See Figure 3.3.

3.6.2 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 $\lambda^* = \frac{\sigma \sqrt{\text{tr}(\hat{\Sigma})}}{\|\theta_*\|_2 \sqrt{n}}$, we have*

$$\mathbb{E} \left[\mathcal{R}(\hat{\theta}_{\lambda^*}) \right] - \mathcal{R}^* \leq \frac{\sigma \sqrt{\text{tr}(\hat{\Sigma})} \|\theta_*\|_2}{\sqrt{n}}.$$

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

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

Similarly, we have $V = \frac{\sigma^2}{n} \text{tr} [\hat{\Sigma}^2 (\hat{\Sigma} + \lambda I)^{-2}] = \frac{\sigma^2}{\lambda n} \text{tr} [\hat{\Sigma} \lambda \hat{\Sigma} (\hat{\Sigma} + \lambda I)^{-2}] \leq \frac{\sigma^2 \text{tr} \hat{\Sigma}}{2\lambda n}$. Plugging in λ^* (which was chosen to minimize the upper bound on $B + V$) gives the result. ■

We can make the following observations:

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

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

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




The number of parameters is not the only way to measure the generalization capabilities of a learning method.

- 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 large constant, and of “slow” rate $O(n^{-1/2})$ with a smaller constant will appear several times in this course.



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

- The value of λ^* involves quantities which 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).
- Note here that the choice of $\lambda^* = \frac{\sigma \sqrt{\text{tr}(\widehat{\Sigma})}}{\|\theta_*\|_2 \sqrt{n}}$ is optimizing the *upper-bound* $\frac{\lambda}{2} \|\theta_*\|_2^2 + \frac{\sigma^2 \text{tr} \widehat{\Sigma}}{2\lambda n}$, and is thus typically not optimal for the true expected risk.
-  Check homogeneity!

Choosing λ in practice. The regularization λ is an example of a *hyper-parameter*. This term refers broadly 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 how to best choose the hyper-parameters, their precise numerical value depends on quantities which are often difficult to know or even guess. In practice, we typically resort to validation and cross-validation.

Exercise 3.3 Compute the expected risk of the estimator obtained by regularizing by $\theta^\top \Lambda \theta$, where $\Lambda \in \mathbb{R}^{d \times d}$ is a positive definite matrix.

3.7 Lower-bound (◆)

In order to show a lower bound in the fixed design setting, we will consider only Gaussian noise, that is, ε has a joint Gaussian distribution with mean zero and covariance matrix $\sigma^2 I$ (adding an extra assumption can only make the lower bound smaller). We follow the elegant and simple proof technique outlined by [76].

The only uncertainty in the model is the location of θ_* . In order to make the dependence on θ_* explicit, we denote by $\mathcal{R}_{\theta_*}(\theta)$ the excess risk (in the previous chapter, we were using the notation \mathcal{R}_{dp}

to make the dependence on the distribution explicit), which is equal to

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

Our goal is to lower bound

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

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 an input and outputs a vector of parameters in \mathbb{R}^d .

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

$$\sup_{\theta_* \in \mathbb{R}^d} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon)) \geq \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \mathcal{R}_{\theta_*}(\mathcal{A}(\Phi\theta_* + \varepsilon)). \quad (3.6)$$

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

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

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

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 a similar operation as for computing the Bayes predictor in Chapter 2. This will be done by conditioning on y , by writing

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

Thus, for each y , the optimal $\mathcal{A}(y)$ has to minimize $\int_{\mathbb{R}^d} \|\mathcal{A}(y) - \theta_*\|_{\Sigma}^2 dp(\theta_* | y)$, 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 $dp(\theta_* | y)$.

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), but we can also use the property that for Gaussian variables, the posterior mean given y is equal to the posterior mode given y , that is, it can be obtained by maximizing the log-likelihood $\log p(\theta_*, y)$ with respect to θ_* . Up to constants and using independence of ε and θ_* , this log-likelihood is

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

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

$$\begin{aligned}
& \inf_{\mathcal{A}} \sup_{\theta_* \in \mathbb{R}^d} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \mathcal{R}_{\theta_*}(\mathcal{A}(\Phi \theta_* + \varepsilon)) - \mathcal{R}^* \\
& \geq \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)} \mathcal{R}_{\theta_*}(\mathcal{A}(\Phi \theta_* + \varepsilon)) - \mathcal{R}^* \text{ using Eq. (3.6),} \\
& = \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \mathcal{R}_{\theta_*}(\mathcal{A}^*(\Phi \theta_* + \varepsilon)) - \mathcal{R}^* \text{ using the reasoning above,} \\
& = \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \|\mathcal{A}^*(\Phi \theta_* + \varepsilon) - \theta_*\|_{\hat{\Sigma}}^2 \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)} \|(\Phi^\top \Phi + n\lambda I)^{-1} \Phi^\top (\Phi \theta_* + \varepsilon) - \theta_*\|_{\hat{\Sigma}}^2 \text{ using the closed-form expression,} \\
& = \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \|(\Phi^\top \Phi + n\lambda I)^{-1} \Phi^\top \varepsilon - n\lambda(\Phi^\top \Phi + n\lambda I)^{-1} \theta_*\|_{\hat{\Sigma}}^2 \\
& = \mathbb{E}_{\theta_* \sim \mathcal{N}(0, \frac{\sigma^2}{\lambda n} I)} \left\| -n\lambda(\Phi^\top \Phi + n\lambda I)^{-1} \theta_* \right\|_{\hat{\Sigma}}^2 + \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, \sigma^2 I)} \|(\Phi^\top \Phi + n\lambda I)^{-1} \Phi^\top \varepsilon\|_{\hat{\Sigma}}^2 \text{ by independence,} \\
& = \frac{\sigma^2}{n\lambda} (n\lambda)^2 \text{tr} [(\hat{\Sigma} + \lambda I)^{-2} \hat{\Sigma}] + \frac{\sigma^2}{n} \text{tr} [(\hat{\Sigma} + \lambda I)^{-2} \hat{\Sigma}^2] \\
& = \frac{\sigma^2}{n} \text{tr} [(\hat{\Sigma} + \lambda I)^{-1} \hat{\Sigma}].
\end{aligned}$$

This risk tends to $\frac{\sigma^2 d}{n}$ when λ tends to zero. This such shows that

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

This gives us a lower-bound on performance, which exactly matches the upper-bound obtained by OLS. In the general non least-squares case, such results are significantly harder to show. See Chapter 4 and Chapter 7.

3.8 Random design analysis

In this section, we consider the regular random design setting, that is, both x and y are considered random, and each pair (x_i, y_i) is assumed independent and identically distributed from a distribution $dp(x, y)$. Our goal is to show that the bound on the 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 $dp(x, y)$, transposed from the fixed design setting to the random design setting:

- there exists a vector $\theta_* \in \mathbb{R}^d$ such that the relationship between input and output is

$$y = \varphi(x)^\top \theta_* + \varepsilon.$$

- the noise $\varepsilon \in \mathbb{R}$ is independent from x , and $\mathbb{E}[\varepsilon] = 0$ and with variance $\mathbb{E}[\varepsilon^2] = \sigma^2$.

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

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

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

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

Proof We have:

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

which leads to the desired result. ■

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

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

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

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

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

Thus, to compute the expected risk of the OLS estimator, we need to compute $\mathbb{E}[\text{tr}(\Sigma \hat{\Sigma}^{-1})]$. One difficulty here is the potential non-invertibility of $\hat{\Sigma}$. Under simple assumptions (e.g., $\varphi(x)$ has a density on \mathbb{R}^d), as soon as $n > d$, $\hat{\Sigma}$ is almost surely invertible, however its smallest eigenvalue can be very small. Extra assumptions are then needed to control it (see, e.g., [76, Section 3]).

Exercise 3.4 *Show that for the random design setting with the same assumptions as Prop. 3.10, the expected risk of the ridge regression estimator is*

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

3.8.1 Gaussian designs

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

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

It turns out that for Gaussians, the matrix $(Z^\top Z)^{-1}$ has a specific distribution, called the inverse Wishart distribution², with an expectation that can be computed exactly as $\mathbb{E}[(Z^\top Z)^{-1}] = \frac{1}{n-d+1}I$. Thus, we have: $\mathbb{E}[\text{tr}(Z^\top Z)^{-1}] = \frac{d}{n-d+1}$ if $n > d+1$, thus leading to the expected excess risk of

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

See [30] for further details. Note here that for Gaussian designs, the expected risk is exactly equal to the expression above, and that later in this course, we will only consider upper-bounds.

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

3.8.2 General designs (◆◆)

In this last more technical section, we highlight 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 $\frac{\sigma^2}{n} \text{tr}(\Sigma \widehat{\Sigma}^{-1})$, this immediately shows that with high probability, the excess risk is less than $\frac{\sigma^2 d}{n} \frac{1}{1-t}$.

In order to obtain such results, more refined concentration inequalities are needed, such as described in [101], [55], [85], and [66]. The sharpest known results for least-squares regression can be found in [76].

Matrix concentration inequality. We will use the matrix Bernstein bound, adapted from [101, Theorem 1.4], already discussed in Section 1.2.5 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 $i \in \{1, \dots, n\}$, $\mathbb{E}[M_i] = 0$, $\lambda_{\max}(M_i) \leq b$ almost surely. Then for all $t \geq 0$,*

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

for $\tau^2 = \lambda_{\max}\left(\frac{1}{n} \sum_{i=1}^n M_i^2\right)$.

²See https://en.wikipedia.org/wiki/Inverse-Wishart_distribution.

Application to re-scaled covariance matrices. We can now prove the following proposition that will give the desired high-probability bound for the excess risk, with one extra assumption.

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

$$\lambda_{\max}\left(\mathbb{E}\left[\varphi(x)^\top \Sigma^{-1} \varphi(x) \varphi(x) \varphi(x)^\top\right]\right) \preceq \rho d \Sigma. \quad (3.7)$$

For $\delta \in (0, 1)$, if $n \geq 8\rho d \log \frac{d}{\delta}$, then with probability greater than $1 - \delta$

$$\Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} \succcurlyeq (1 - t)I. \quad (3.8)$$

Before giving the proof, note that from the discussion earlier, the bound in Eq. (3.8) leads to an excess risk less than $\frac{\sigma^2 d}{n} \frac{1}{1-t}$. Moreover, without surprise, the bound is non vacuous only for $n \geq d$.

Regarding the extra assumption in Eq. (3.7), it can be interpreted as follows. We consider the random vector $z = \Sigma^{-1/2} \varphi(x) \in \mathbb{R}^d$, which is such that $\mathbb{E}[zz^\top] = I$ and $\mathbb{E}[\|z\|_2^2] = d$. The assumption in Eq. (3.7) is then equivalent to

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

A sufficient condition is that almost surely $\|z\|_2^2 \leq \rho d$, that is, $\varphi(x)^\top \Sigma^{-1} \varphi(x) \leq \rho d$. Moreover, for a Gaussian distribution with zero mean for z , one can check as an exercise that $\rho = (1 + 2/d)$.

Proof We consider the random symmetric matrix $M_i = I - z_i z_i^\top$, which is such that $\mathbb{E} M_i = 0$, $\lambda_{\max}(M_i) \leq 1$ almost surely, and $\mathbb{E}[M_i^2] = \mathbb{E}[\|z_i\|^2 z_i z_i^\top] - I$ with largest eigenvalue less than ρd . We thus have for any $t \geq 0$,

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

Thus, if t is such that $\frac{nt^2}{2\rho d + 2t/3} \geq \log \frac{d}{\delta}$, then, with probability greater than $1 - \delta$, we have $I - \Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} \preceq tI$, that is, the desired result $\Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} \succcurlyeq (1 - t)I$. We have used the order between symmetric matrices, defined as $A \succcurlyeq B \Leftrightarrow B \preceq A \Leftrightarrow A - B$ positive semi-definite.

This is possible when $t \geq \sqrt{\frac{2\rho d}{n} \log \frac{d}{\delta}} + \frac{2}{3n} \log \frac{d}{\delta}$. The bound is non-vacuous only when $t < 1$, that is, it is sufficient to impose $\frac{2}{3n} \log \frac{d}{\delta} < 1/2$ and $\sqrt{\frac{2\rho d}{n} \log \frac{d}{\delta}} < 1/2$, which is equivalent to $n \geq \frac{4}{3} \log \frac{d}{\delta}$, and $n \geq 8\rho d \log \frac{d}{\delta}$. Given that we always $\rho \geq 1$, only the second constraint is necessary. ■

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 and the estimation error.

-Rademacher complexity: To study estimation errors and compute expected uniform deviations, Rademacher complexities are a very flexible and powerful tool.

-Relationship with asymptotic statistics: classical asymptotic results provide a finer picture of the behavior of empirical risk minimization at they provide asymptotic limits of performance as a well-defined constant times $1/n$, but they do not characterize small-sample effects.

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

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

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

4.1 Convexification of the risk


In this section, for simplicity, we focus on binary classification where $\mathcal{Y} = \{-1, 1\}$ with the 0-1 loss, but many of the concepts extend to the more general structured prediction set-up (see [83, 84] and the many references therein).

As our goal is to estimate a binary-valued function, the first idea that comes into mind is to minimize the empirical risk over a hypothesis space of binary-valued functions (or equivalently, space of subsets of \mathcal{X}). However, this approach leads to a combinatorial problem which can be computationally intractable and moreover, it is not clear how to control the capacity (i.e., how to regularize) for these type of hypothesis spaces. Learning a real-valued function instead through the framework of convex surrogates simplifies and overcomes this problem as it convexifies the problem and classical penalty-based regularization techniques can be used for theoretical analysis (this chapter) and for algorithms (Chapter 5).

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


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

Note here, that the value at 0 could also be chosen to be -1 . Within our context, this corresponds, for maximally ambiguous observations, to choose one of the two labels which are equally likely (and thus equally bad in expectation, with a 50% chance of being incorrect).

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

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

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

 Note the slightly overloaded denomination above where the 0-1 loss function is defined on \mathbb{R} , with 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 $g : \mathcal{X} \rightarrow \mathbb{R}$ 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 non-convex) 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 (all will be convex). See classical examples 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 key question is: does it make sense to simply convexify the problem? In other words, does it lead to good predictions for the 0-1 loss?

Classical examples. We first review the main examples used in practice:

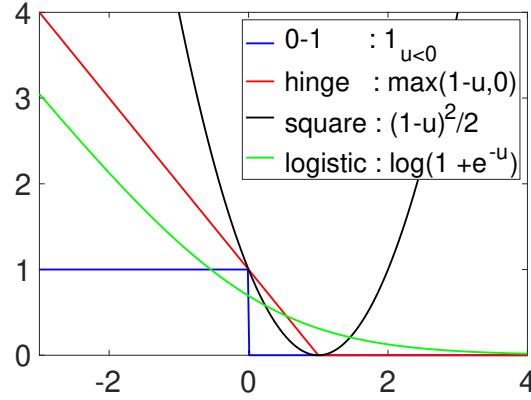


Figure 4.1: Classical convex surrogates for binary classification with the 0-1 loss.

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

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

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

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

The risk is then the negative conditional log-likelihood $\mathbb{E}[-\log p(y|x)]$. It is also often called the cross-entropy loss (see https://en.wikipedia.org/wiki/Logistic_regression for details).

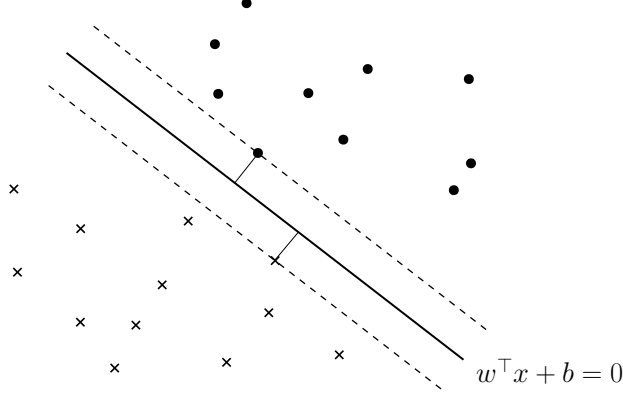
- **Hinge loss:** $\Phi(u) = \max(1 - u, 0)$. With linear predictors, this leads to the support vector machine, and $yf(x)$ is often called the “margin” in this context. This loss has a geometric interpretation (see section below). See also https://en.wikipedia.org/wiki/Support_vector_machine for details.
- **Squared hinge loss:** $\Phi(u) = \max(1 - u, 0)^2$. This is a smooth counterpart to the regular hinge loss.

4.1.2 Geometric interpretation of the support vector machine (♦)

In this section, we provide a geometrical (and historical perspective) on the hinge loss, to highlight the reason 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. We first assume that the data are separable by an affine hyperplane, that is, there exist $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that for all $i \in \{1, \dots, n\}$, $y_i(w^\top x_i + b) > 0$. Among the infinitely

many separating hyperplane, we aim at selecting the one for which the closest point from the dataset is farthest.



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

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

and we thus aim at maximizing this quantity. Because of the invariance by rescaling (that is, we can multiply w and b by the same scalar constant without modifying the affine separator), this problem is equivalent to the following problem

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

General data. When data may not be separated by an hyperplane, then we can introduce so-called “slack variables” $\xi_i \geq 0, i = 1, \dots, n$, allowing the constraint $y_i(w^\top x_i + b) \geq 1$ to be not satisfied, by introducing instead the constraint $y_i(w^\top x_i + b) \geq 1 - \xi_i$. The overall amount of slack is then minimized, leading to the following problem (with $C > 0$)

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

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

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

which is exactly an 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 Eq. (4.2) is a linearly constrained convex optimization problem, and use Lagrangian duality [28]. We consider non-negative Lagrange

multipliers α_i and β_i , $i \in \{1, \dots, n\}$, and the following Lagrangian:

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

Minimizing with respect to $\xi \in \mathbb{R}^n$ leads to the constraints $\forall i \in \{1, \dots, n\}$, $\alpha_i + \beta_i = C$, 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:

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

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 $x_i^\top x_j$, $i, j = 1, \dots, n$, and 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 $\alpha_i (y_i(w^\top x_i + b) - 1 + \xi_i) = 0$ 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 are equal to zero, and the optimal predictor is a linear combination of only a few of the data points x_i ’s which are then called “support vectors”.

4.1.3 Conditional Φ -risk and classification calibration (♦)



Most of the convex surrogates are upper-bounds on the 0-1 loss and all can be made so with rescaling. Using this as the sole justification of the good performance of a convex surrogate is a misleading justification, with the exception of problems with almost surely zero loss for the Bayes (i.e., optimal) predictor (which is only possible when the Bayes risk is zero).

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

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

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

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

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

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

with

$$C_{\eta}(\alpha) = \eta\Phi(\alpha) + (1 - \eta)\Phi(-\alpha).$$

The least we can expect from a convex surrogate is that in the population case, where all x 's decouple, the optimal $g(x)$ obtained by minimizing the conditional Φ -risk exactly leads to the same prediction as the Bayes predictor (at least when this prediction is unique). In other words, since the prediction is $\text{sign}(g(x))$, we want that for any $\eta \in [0, 1]$:

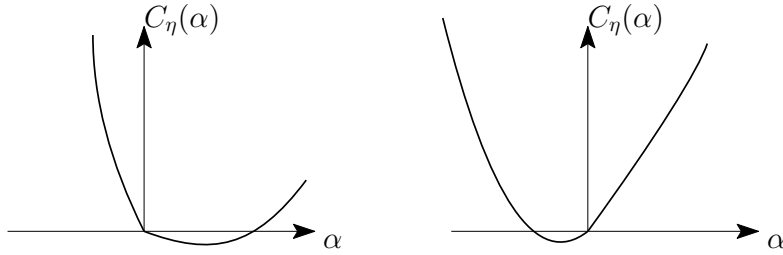
$$(\text{positive optimal prediction}) \quad \eta > 1/2 \Leftrightarrow \arg \min_{\alpha \in \mathbb{R}} C_\eta(\alpha) \subset \mathbb{R}_+^* \quad (4.4)$$

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

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

Proposition 4.1 [17] *let $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ convex. Φ calibrated $\Leftrightarrow \Phi$ is differentiable at 0 and $\Phi'(0) < 0$.*

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



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

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

- (a) Assume Φ is calibrated. By letting η tend to $1/2+$ in Eq. (4.6), this leads to $(C_{1/2})_+(0)' = \frac{1}{2}[\Phi'_+(0) - \Phi'_-(0)] \leq 0$. Since Φ is convex, we always have $\Phi'_+(0) - \Phi'_-(0) \geq 0$. Thus the left and right derivatives are equal, which implies that Φ is differentiable at 0. Then $C'_\eta(0) = (2\eta - 1)\Phi'(0)$, and from Eq. (4.4) and Eq. (4.6), we need to have $\Phi'(0) < 0$.
- (b) Assume Φ is differentiable at 0 and $\Phi'(0) < 0$, then $C'_\eta(0) = (2\eta - 1)\Phi'(0)$; Eq. (4.4) and Eq. (4.5) are then direct consequences of Eq. (4.6) and Eq. (4.7).

■

Note that the proposition above excludes the convex surrogate $u \mapsto (-u)_+ = \max\{-u, 0\}$, which is not differentiable at zero.

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

4.1.4 Relationship between risk and Φ -risk (◆◆)

Now that we know that for any $x \in \mathcal{X}$, minimizing $C_{\eta(x)}(g(x))$ with respect to $g(x)$ leads to the optimal prediction through $\text{sign}(g(x))$, we would like to make sure that an explicit control of the excess Φ -risk (which we aim to do with empirical risk minimization using tools from later sections) leads to an explicit control of the original excess risk. In other words, we are looking for a monotonic function $H : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that $\mathcal{R}(g) - \mathcal{R}^* \leq H[\mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*]$, where \mathcal{R}_Φ^* is the minimum possible Φ -risk. The function H is often called the *calibration function*.



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

We first start with a simple lemma expressing the excess risk, as well as an upper bound (adapted from Theorem 2.2 from [41]), that we will need for comparison inequalities below:

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

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

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

Proof We express the excess risk as:

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

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

For the second result, we simply use the fact that if $g(x)g^*(x) < 0$, then, by splitting the cases into two (the first one being $\eta(x) > 1/2$ and $g(x) < 0$, the second one being $\eta(x) < 1/2$ and $g(x) > 0$), we get $|2\eta(x) - 1| \leq |2\eta(x) - 1 - g(x)|$, and thus the second result.

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

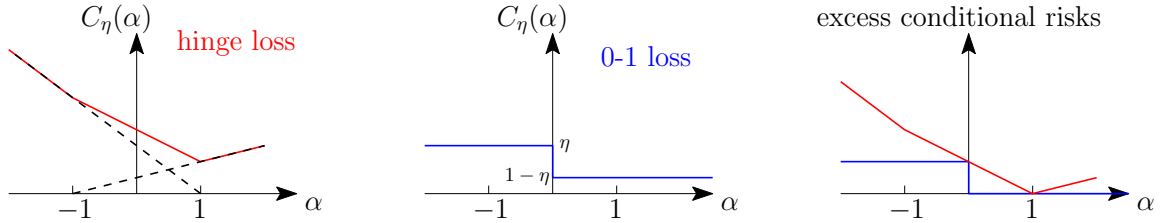
We see that the excess risk is the expectation of a quantity $|2\eta(x) - 1| \cdot 1_{g(x)g^*(x) < 0}$, which is equal to 0 if the classification is the same as the Bayes predictor and equal to $|2\eta(x) - 1|$ otherwise. The excess conditional Φ -risk is the quantity

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

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

In [17], a general framework is proposed. We will only consider the hinge loss and smooth losses for simplicity.

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



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

Note that when the Bayes risk is zero, that is, $\eta(x) \in \{0, 1\}$ almost surely, then using the fact that the hinge loss is an upper-bound on the 0-1 loss is enough to show that the excess risk is less than the excess Φ -risk (indeed, the two optimal risks \mathcal{R}^* and \mathcal{R}_Φ^* are equal to zero).

- We consider smooth losses of the form (up to additive and multiplicative constants) $\Phi(v) = a(v) - v$, where $a(v) = \frac{1}{2}v^2$ for the quadratic loss, $a(v) = 2 \log(e^{v/2} + e^{-v/2})$ for the logistic loss. We assume that a is even, $a(0) = 0$, a is β -smooth (that is, as defined in Chapter 5, $a''(v) \leq \beta$ for all $v \in \mathbb{R}$). This implies¹ that for all $v \in \mathbb{R}$, $a(v) - \alpha v - \inf_{w \in \mathbb{R}} \{a(w) - \alpha w\} \geq \frac{1}{2\beta} |\alpha - a'(v)|^2$, leading to:

$$\begin{aligned}
 \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^* &= \mathbb{E}[a(g(x)) - (2\eta(x) - 1)g(x) - \inf_{w \in \mathbb{R}} \{a(w) - (2\eta(x) - 1)w\}] \\
 &\geq \frac{1}{2\beta} \mathbb{E}[|2\eta(x) - 1 - a'(g(x))|^2] \text{ by the property above,} \\
 &\geq \frac{1}{2\beta} (\mathbb{E}[|2\eta(x) - 1 - a'(g(x))|])^2 \text{ by Jensen's inequality,} \\
 &= \frac{1}{2\beta} (\mathcal{R}(g) - \mathcal{R}^*)^2 \text{ using Lemma 4.1.}
 \end{aligned}$$

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

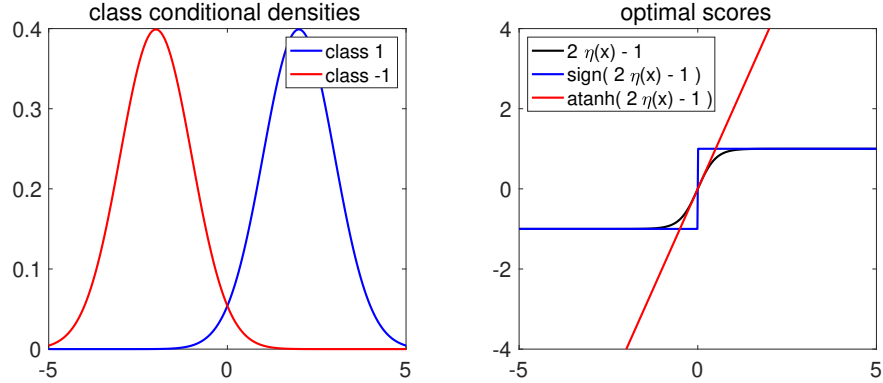
Exercise 4.1 (♦) Show that the function a^* satisfies $a^*(\mathcal{R}(g) - \mathcal{R}^*) \leq \mathcal{R}_\Phi(g) - \mathcal{R}_\Phi^*$ for any function $g : \mathcal{X} \rightarrow \mathbb{R}$.

¹Using the Fenchel conjugate $a^* : \mathbb{R} \rightarrow \mathbb{R}$ which is $1/(2\beta)$ -strongly convex (see Chapter 5), we have: $a(v) - \alpha v - \inf_{w \in \mathbb{R}} \{a(w) - \alpha w\} = a(v) - \alpha v + a^*(\alpha) = a^*(\alpha) - a^*(a'(v)) - (\alpha - a'(v))(a^*)'(a(v)) \geq \frac{1}{2\beta} |\alpha - a'(v)|^2$, where a^* is the Fenchel conjugate of a [28].

We can make the following observations:

- For the (non-smooth) hinge loss, the calibration function is identity, so if the excess Φ -risk goes to zero at a certain rate, the excess risk goes to zero at the same rate, whereas 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, that is, after thresholding the function g at zero, the observed rates may be worse.
- Note that the noiseless case where $\eta(x) \in \{0, 1\}$ (zero Bayes risk) leads to stronger calibration function, as well as a series of intermediate “low-noise” conditions (see [17] for details).

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



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, $\text{sign}(2\eta(x) - 1)$ has to be in the class of functions we use (for the hinge loss), or $2\eta(x) - 1$ for the square loss, or $\text{atanh}(2\eta(x) - 1)$ for the logistic loss.

Exercise 4.2 For the logistic loss, show that for data generated as $x|y = 1$ and $x|y = -1$ Gaussians with the same covariance matrix, then the function $g(x)$ minimizing the expected logistic loss is affine in x (this model is often referred to as linear discriminant analysis).

4.2 Risk minimization decomposition

We consider a family \mathcal{F} of prediction functions $f : \mathcal{X} \rightarrow \mathcal{Y}$. Empirical risk minimization aims at finding

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

We can decompose the risk as follows into two terms:

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

A classical example is the situation where the family of functions is parameterized by a subset of \mathbb{R}^d , 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 certain 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 $\|\theta\|_2$.

We now turn separately to the approximation and estimation errors.

4.3 Approximation error

Bounding the approximation error corresponds to bounding $\inf_{f \in \mathcal{F}} \mathcal{R}(f) - \mathcal{R}^*$ and requires assumptions on the Bayes predictor (sometimes also called the “target function”) f^* (and hence on the testing distribution) to achieve non-trivial learning rates.

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

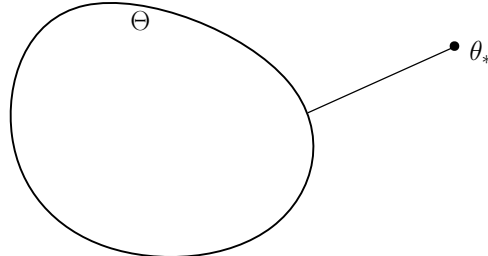
$$\inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \mathcal{R}^* = \left(\inf_{\theta \in \Theta} \mathcal{R}(f_\theta) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) \right) + \left(\inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) - \mathcal{R}^* \right).$$

- 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 .
- The function $\theta \mapsto \mathcal{R}(f_\theta) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta)$ is a positive function on \mathbb{R}^d , which can be typically upperbounded by a certain 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 “distance” between θ_* and Θ .

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

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

and thus this second 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 distance $d(\theta, \theta') = \mathbb{E}[|f_\theta(x) - f_{\theta'}(x)|]$.



A classical example will be $f_\theta(x) = \theta^\top \varphi(x)$, and $\Theta = \{\theta \in \mathbb{R}^d, \|\theta\|_2 \leq D\}$, leading to the upper bound $G\mathbb{E}[\|\varphi(x)\|_2](\|\theta_*\|_2 - D)_+$, which is equal to zero if $\|\theta_*\|_2 \leq D$ (well-specified model).

- **Exercise 4.3** Perform the same computation for the ℓ_1 -norm on Θ .

4.4 Estimation error

We will consider general techniques, and all apply them to linear models with bounded ℓ_2 -norm by D , and G -Lipschitz-losses for illustration.

The estimation error is often decomposed as, using $g \in \arg \min_{g \in \mathcal{F}} \mathcal{R}(g)$ the minimizer of the expected risk for our class of models, and $\hat{f} \in \arg \min_{f \in \mathcal{F}} \hat{\mathcal{R}}(f)$ the minimizer of the empirical risk:

$$\begin{aligned} \mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f) = \mathcal{R}(\hat{f}) - \mathcal{R}(g) &= \left\{ \mathcal{R}(\hat{f}) - \hat{\mathcal{R}}(\hat{f}) \right\} + \left\{ \hat{\mathcal{R}}(\hat{f}) - \hat{\mathcal{R}}(g) \right\} + \left\{ \hat{\mathcal{R}}(g) - \mathcal{R}(g) \right\} \\ &\leq \sup_{f \in \mathcal{F}} \left\{ \mathcal{R}(f) - \hat{\mathcal{R}}(f) \right\} + \left\{ \hat{\mathcal{R}}(\hat{f}) - \hat{\mathcal{R}}(g) \right\} + \sup_{f \in \mathcal{F}} \left\{ \hat{\mathcal{R}}(f) - \mathcal{R}(f) \right\} \\ &\leq \sup_{f \in \mathcal{F}} \left\{ \mathcal{R}(f) - \hat{\mathcal{R}}(f) \right\} + 0 + \sup_{f \in \mathcal{F}} \left\{ \hat{\mathcal{R}}(f) - \mathcal{R}(f) \right\} \text{ by definition of } \hat{f}. \end{aligned}$$

This is often further upper-bounded by $2 \sup_{f \in \mathcal{F}} \left| \hat{\mathcal{R}}(f) - \mathcal{R}(f) \right|$. We can make the following observations:

- When \hat{f} is not the global minimizer of $\hat{\mathcal{R}}$ but simply satisfies $\hat{\mathcal{R}}(\hat{f}) \leq \inf_{f \in \mathcal{F}} \hat{\mathcal{R}}(f) + \varepsilon$, then the *optimization error* ε has to be added to the bound above (see more details in Chapter 5).
- The uniform deviation grows with the “size” of \mathcal{F} , and usually decays with n . See examples below.
- A key issue 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 values of f , there is always a small chance that one of these deviations get large. We thus need an explicit control of this phenomenon, which we now tackle, by first showing that we can focus on the expectation alone.

4.4.1 Application of MacDiarmid’s inequality

Let $H(z_1, \dots, z_n) = \sup_{f \in \mathcal{F}} \left\{ \mathcal{R}(f) - \hat{\mathcal{R}}(f) \right\}$, where the random variables $z_i = (x_i, y_i)$ are independent and identically distributed, and $\hat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i))$. We let ℓ_∞ denote the maximal absolute value of the loss functions for all (x, y) in the support of the data generating distribution and $f \in \mathcal{F}$.

When changing a single $z_i \in \mathcal{X} \times \mathcal{Y}$ into $z'_i \in \mathcal{X} \times \mathcal{Y}$, the deviation in H is almost surely at most $\frac{2}{n} \ell_\infty$.

Thus, applying Mac Diarmid inequality (see Section 1.2.2 in Chapter 1), with probability greater than $1 - \delta$, we have:

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

We thus only need to bound the expectation of $\sup_{f \in \mathcal{F}} \left\{ \mathcal{R}(f) - \hat{\mathcal{R}}(f) \right\}$ and of $\sup_{f \in \mathcal{F}} \left\{ \hat{\mathcal{R}}(f) - \mathcal{R}(f) \right\}$ (which will typically have the same bound), and add on top of it $\frac{\ell_\infty \sqrt{2}}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}}$.

We now provide a series of bounds to bound these expectations, from simple to more refined, culminating in Rademacher complexities in Section 4.5.

4.4.2 Easy case I: quadratic functions

We will show what happens with a quadratic loss function and an ℓ_2 -ball constraint. We remember that in this case $\ell(y, \theta^\top \varphi(x)) = (y - \theta^\top \varphi(x))^2$. From that we get

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

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

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

where $\|M\|_{\text{op}}$ is the operator norm of the matrix M defined as $\|M\|_{\text{op}} = \sup_{\|u\|_2=1} \|Mu\|_2$.

Thus, in order to get a uniform bound, we simply need to upper-bound the three *non-uniform* expectations of deviations, and thus of order $O(1/\sqrt{n})$, and we get an overall uniform deviation bound. This particular 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 not possible, so we need to introduce new tools.

Exercise 4.4 (♦) *Provide an explicit bound on $\sup_{\|\theta\|_2 \leq D} |\mathcal{R}(f) - \widehat{\mathcal{R}}(f)|$ above, and compare it to the use of Rademacher complexities in Section 4.5. Concentration of averages of matrices from Section 1.2.5 can be used.*

⚠ Note that from now on, in the sections below, 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 $-\ell_\infty$ and ℓ_∞ , using the upper-bound $2 \sup_{f \in \mathcal{F}} |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)|$ on the estimation error, and the union bound:

$$\mathbb{P} \left(\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f) \geq t \right) \leq \mathbb{P} \left(2 \sup_{f \in \mathcal{F}} |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)| \geq t \right) \leq \sum_{f \in \mathcal{F}} \mathbb{P} \left(2 |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)| \geq t \right).$$

We have, for $f \in \mathcal{F}$ fixed, $\widehat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(y_i))$, and we can apply Hoeffding's inequality from Section 1.2.1 to bound each $\mathbb{P}\left(2|\widehat{\mathcal{R}}(f) - \mathcal{R}(f)| \geq t\right)$, leading to

$$\mathbb{P}\left(\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f) \geq t\right) \leq \sum_{f \in \mathcal{F}} 2 \exp(-nt^2/2\ell_\infty^2) = 2|\mathcal{F}| \exp(-nt^2/2\ell_\infty^2).$$


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

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f) \leq \frac{2\ell_\infty}{\sqrt{n}} \sqrt{\log \frac{2|\mathcal{F}|}{\delta}} = \frac{2\ell_\infty}{\sqrt{n}} \sqrt{\log(|\mathcal{F}|) + \log \frac{2}{\delta}} \leq 2\ell_\infty \sqrt{\frac{\log(|\mathcal{F}|)}{n}} + \frac{2\ell_\infty}{\sqrt{n}} \sqrt{\log \frac{2}{\delta}}.$$

Exercise 4.5 (♦) *In terms of expectation, we get (using the proof of the max of random variables from Section 1.2.4 in Chapter 2, which applies because bounded random variables are sub-Gaussian):*

$$\mathbb{E}[\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f)] \leq 2\mathbb{E}\left[\sup_{f \in \mathcal{F}} |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)|\right] \leq \ell_\infty \sqrt{\frac{2\log(|\mathcal{F}|)}{n}}.$$

Thus, according to the bound, when the logarithm of the number of models is small compared to n , learning is possible. This is a first generic control of the uniform deviations.

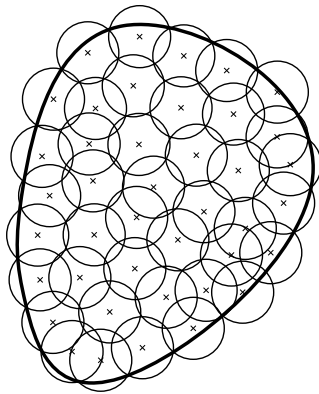
 Note that this is only an upper-bound and learning is possible with infinitely many models (which is the most classical scenario). See below.

4.4.4 Beyond finite number models through covering numbers (♦)

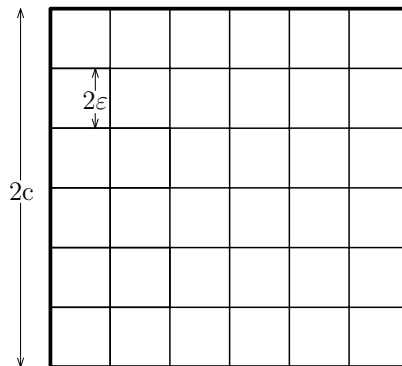
The simple idea behind covering numbers is to deal with function spaces with infinitely many elements by approximating them through a finite number of elements. This is often referred to as an “ ε -net argument”.

We first need to assume that the risks \mathcal{R} and $\widehat{\mathcal{R}}$ are regular, for example, that they are G -Lipschitz-continuous with respect to some distance d on \mathcal{F} .

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



The covering number $m(\varepsilon)$ is a non-increasing function of ε . Typically, $m(\varepsilon)$ grows with ε as a power ε^{-d} when $\varepsilon \rightarrow 0$, 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 . See below.



Given that all norms are equivalent in dimension d , we get the same dependence in d for all bounded subsets of a finite-dimensional vector space.

For some sets (e.g., all Lipschitz-continuous functions in d dimensions) $\log m(\varepsilon)$ grows faster, for example as ε^{-d} . See, e.g., [105].

ε -net argument. Given a cover of \mathcal{F} , for all $f \in \mathcal{F}$, and with $(f_i)_{i \in \{1, \dots, m(\varepsilon)\}}$ the associated cover elements,

$$\begin{aligned} |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)| &\leq |\widehat{\mathcal{R}}(f) - \widehat{\mathcal{R}}(f_i)| + |\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)| + |\mathcal{R}(f_i) - \mathcal{R}(f)| \\ &\leq 2G\varepsilon + \sup_{i \in \{1, \dots, m(\varepsilon)\}} |\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)|. \end{aligned}$$

This implies that, using bounds on the expectation of the maximum (Section 1.2.4), which apply because bounded random variables are sub-Gaussian (with the sub-Gaussianity parameter proportional to the almost sure bound):

$$\mathbb{E} \left[\sup_{f \in \mathcal{F}} |\widehat{\mathcal{R}}(f) - \mathcal{R}(f)| \right] \leq 2G\varepsilon + \mathbb{E} \left[\sup_{i \in \{1, \dots, m(\varepsilon)\}} |\widehat{\mathcal{R}}(f_i) - \mathcal{R}(f_i)| \right] \leq 2G\varepsilon + \ell_\infty \sqrt{\frac{2 \log(2m(\varepsilon))}{n}}.$$

Therefore, if $m(\varepsilon) \sim \varepsilon^{-d}$, ignoring constants, we need to balance $\varepsilon + \sqrt{d \log(1/\varepsilon)/n}$, which leads to, with a choice of ε proportional to $1/\sqrt{n}$, to a rate proportional $\sqrt{(d/n) \log(n)}$, a rate essentially proportional to $\sqrt{d/n}$. Unfortunately, this often leads to a non-optimal dependence on dimension.

One very powerful tool that avoids these undesired dependences on dimension is Rademacher complexities [25] or Gaussian complexities [18]. In this chapter, we will focus on Rademacher complexity.

4.5 Rademacher complexity

We consider n independent and identically distributed random variables $z_1, \dots, z_n \in \mathcal{Z}$, and a class \mathcal{H} of functions from \mathcal{Z} to \mathbb{R} . In our context, the space of functions is related to the learning problem as: $\mathcal{H} = \{(x, y) \mapsto \ell(y, f(x)), f \in \mathcal{F}\}$.

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 happens to be equal to

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

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

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

$$R_n(\mathcal{H}) = \mathbb{E}_{\varepsilon, \mathcal{D}} \left(\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i h(z_i) \right), \quad (4.8)$$

where $\varepsilon \in \mathbb{R}^n$ is a vector of independent Rademacher random variable (that is taking values -1 or 1 with equal probabilities), which is also independent of \mathcal{D} . It is a deterministic quantity that only depends on n and \mathcal{H} .

In words, the Rademacher complexity is equal to the expectation of the maximal dot-product between values of a function h at the observations x_i and random labels. It is a measure of the “capacity” of the set of functions \mathcal{H} . We will see later that it can be computed in many interesting cases and leads to interesting and powerful bounds.

4.5.1 Symmetrization

First, we relate it to the uniform deviation through a general “symmetrization” property, which shows that the Rademacher complexity directly controls the expected uniform deviation.

Proposition 4.2 (symmetrization) *Given the Rademacher complexity of \mathcal{H} defined in Eq. (4.8), we have:*

$$\mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n h(z_i) - \mathbb{E}[h(z)] \right) \right] \leq 2R_n(\mathcal{H}) \quad \text{and} \quad \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\mathbb{E}[h(z)] - \frac{1}{n} \sum_{i=1}^n h(z_i) \right) \right] \leq 2R_n(\mathcal{H}).$$

Proof (♦) Let $\mathcal{D}' = \{z'_1, \dots, z'_n\}$ an independent copy of the data $\mathcal{D} = \{z_1, \dots, z_n\}$. Let $(\varepsilon_i)_{i \in \{1, \dots, n\}}$ be i.i.d. Rademacher random variables, which are also independent of \mathcal{D} and \mathcal{D}' . Using that for all i in $\{1, \dots, n\}$, $\mathbb{E}[h(z'_i)|\mathcal{D}] = \mathbb{E}[h(z)]$, we have:

$$\begin{aligned} \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\mathbb{E}[h(z)] - \frac{1}{n} \sum_{i=1}^n h(z_i) \right) \right] &= \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n \mathbb{E}[h(z'_i)|\mathcal{D}] - \frac{1}{n} \sum_{i=1}^n h(z_i) \right) \right] \\ &= \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n \mathbb{E}[h(z'_i) - h(z_i)|\mathcal{D}] \right) \right] \end{aligned}$$

by definition of the independent copy \mathcal{D}' . Then

$$\begin{aligned} \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\mathbb{E}[h(z)] - \frac{1}{n} \sum_{i=1}^n h(z_i) \right) \right] &\leq \mathbb{E} \left[\mathbb{E} \left(\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n [h(z'_i) - h(z_i)] \right) \middle| \mathcal{D} \right) \right] \\ &\quad \text{using that the supremum of the expectation is less than expectation of the supremum,} \\ &= \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n [h(z'_i) - h(z_i)] \right) \right] \quad \text{by the towering law of expectation,} \\ &= \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i (h(z'_i) - h(z_i)) \right) \right] \quad \text{by symmetry of the law of } \varepsilon_i, \\ &\leq \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i (h(z_i)) \right) \right] + \mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i (-h(z_i)) \right) \right] \\ &= 2\mathbb{E} \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i h(z_i) \right) \right] = 2R_n(\mathcal{H}). \end{aligned}$$

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

Exercise 4.6 If \mathcal{H} is finite, and so that, for all $h \in H$ and almost all z , $|h(z)| \leq \ell_\infty$, compute an upperbound on $R_n(\mathcal{H})$ and relate it to Section 4.4.3.

4.5.2 Lipschitz-continuous losses

A particularly appealing property in our context is the following property, sometimes called the “contraction principle”, using a simple proof from [73, Lemma 5].

Proposition 4.3 (Contraction principle - Lipschitz-continuous functions) *Given any functions $b, a_i : \Theta \rightarrow \mathbb{R}$ (no assumption) and $\varphi_i : \mathbb{R} \rightarrow \mathbb{R}$ any 1-Lipschitz-functions, for $i = 1, \dots, n$, we have, for $\varepsilon \in \mathbb{R}^n$ a vector of independent Rademacher random variables:*

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

Proof (♦) We consider a proof by induction on n . The case $n = 0$ is trivial, and we show how to go from $n \geq 0$ to $n + 1$. We thus consider $\mathbb{E}_{\varepsilon_1, \dots, \varepsilon_{n+1}} \left[\sup_{\theta \in \Theta} b(\theta) + \sum_{i=1}^{n+1} \varepsilon_i \varphi_i(a_i(\theta)) \right]$ and compute the expectation with respect to ε_{n+1} explicitly, by considering the two potential values with probability 1/2:

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

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

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

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

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

which leads to the desired result. ■

We can apply the contraction principle above to supervised learning situations where $u_i \mapsto \ell(y_i, u_i)$ is G -Lipschitz-continuous for all i almost surely (which is possible for regression or when using a convex surrogate for binary classification as presented in Section 4.1), leading to:

$$\mathbb{E}_\varepsilon \left(\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \ell(y_i, f(x_i)) \mid \mathcal{D} \right) \leq G \cdot \mathbb{E}_\varepsilon \left(\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i f(x_i) \mid \mathcal{D} \right) \text{ by the contraction principle,}$$

which leads to

$$\mathcal{R}_n(\mathcal{H}) \leq G \cdot \mathcal{R}_n(\mathcal{F}). \quad (4.9)$$

Thus the Rademacher complexity of the class of prediction functions controls the uniform deviations of the empirical risk. We now consider simple examples.

4.5.3 Ball-constrained linear predictions

We now assume that $\mathcal{F} = \{f_\theta(x) = \theta^\top \varphi(x), \Omega(\theta) \leq D\}$ where Ω is a norm on \mathbb{R}^d . We denote by $\Phi \in \mathbb{R}^{n \times d}$ the design matrix. We have

$$\begin{aligned} \mathcal{R}_n(\mathcal{F}) &= \mathbb{E} \left[\sup_{\Omega(\theta) \leq D} \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i \theta^\top \varphi(x_i) \right) \right] = \mathbb{E} \left[\sup_{\Omega(\theta) \leq D} \frac{1}{n} \varepsilon^\top \Phi \theta \right] \\ &= \frac{D}{n} \mathbb{E} \left[\Omega^*(\Phi^\top \varepsilon) \right], \end{aligned}$$

where $\Omega^*(u) = \sup_{\Omega(\theta) \leq 1} u^\top \theta$ 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 [28].

Thus, computing Rademacher complexities is equivalent to computing expectation of norms. When $\Omega = \|\cdot\|_2$, we get:

$$\begin{aligned} \mathcal{R}_n(\mathcal{F}) &= \frac{D}{n} \mathbb{E} \left[\|\Phi^\top \varepsilon\|_2 \right] \\ &\leq \frac{D}{n} \sqrt{\mathbb{E} [\|\Phi^\top \varepsilon\|_2^2]} \text{ by Jensen's inequality,} \\ &= \frac{D}{n} \sqrt{\mathbb{E} [\text{tr}[\Phi^\top \varepsilon \varepsilon^\top \Phi]]} \\ &= \frac{D}{n} \sqrt{\mathbb{E} [\text{tr}[\Phi^\top \Phi]]} \text{ using that } \mathbb{E}[\varepsilon \varepsilon^\top] = I, \\ &= \frac{D}{n} \sqrt{\sum_{i=1}^n \mathbb{E}(\Phi^\top \Phi)_i} = \frac{D}{n} \sqrt{\sum_{i=1}^n \mathbb{E} \|\varphi(x_i)\|_2^2} = \frac{D}{\sqrt{n}} \sqrt{\mathbb{E} \|\varphi(x)\|_2^2}. \end{aligned} \tag{4.10}$$

We thus obtain a dimension-independent Rademacher complexity that we can use in the summary below.

Exercise 4.7 Upper-bound the Rademacher complexity for $\Omega = \|\cdot\|_1$.

4.5.4 Putting things together (linear predictions)

With all the elements above, we can now propose the following general result (where no convexity of the loss function is assumed).

Proposition 4.4 (Estimation error) Assume a G -Lipschitz-continuous loss function, linear prediction functions with $\mathcal{F} = \{f_\theta(x) = \theta^\top \varphi(x), \|\theta\|_2 \leq D\}$, where $\mathbb{E} \|\varphi(x)\|_2^2 \leq R^2$. Let $\hat{f} = f_{\hat{\theta}} \in \mathcal{F}$ be the minimizer of the empirical risk, then:

$$\mathbb{E}[\mathcal{R}(f_{\hat{\theta}})] \leq \inf_{\|\theta\|_2 \leq D} \mathcal{R}(f_\theta) + \frac{2GRD}{\sqrt{n}}.$$

Proof Using Prop. 4.2, Eq. (4.9) and Eq. (4.10), we get the desired result. \blacksquare

If we assume that there exists a minimizer θ_* of $\mathcal{R}(f_\theta)$ over \mathbb{R}^d , the approximation error is upper-bounded by

$$\begin{aligned} \inf_{\|\theta\|_2 \leq D} \mathcal{R}(f_\theta) - \mathcal{R}(f_{\theta_*}) &\leq G \inf_{\|\theta\|_2 \leq D} \mathbb{E}[|f_\theta(x) - f_{\theta_*}(x)|] \\ &= G \inf_{\|\theta\|_2 \leq D} \mathbb{E}[|\varphi(x)^\top (\theta - \theta_*)|] \\ &\leq G \inf_{\|\theta\|_2 \leq D} \|\theta - \theta_*\|_2 \mathbb{E}[\|\varphi(x)\|_2^2] \leq GR \inf_{\|\theta\|_2 \leq D} \|\theta - \theta_*\|_2. \end{aligned}$$

This leads to

$$\mathbb{E}[\mathcal{R}(f_{\hat{\theta}})] \leq GR \inf_{\|\theta\|_2 \leq D} \|\theta - \theta_*\|_2 + \frac{2GRD}{\sqrt{n}} = GR(\|\theta_*\|_2 - D)_+ + \frac{2GRD}{\sqrt{n}}.$$

We see that for $D = \|\theta_*\|_2$, we obtain the bound $\frac{2GR\|\theta_*\|_2}{\sqrt{n}}$, but this setting requires to know $\|\theta_*\|_2$ which is not possible in practice. If D is too large, the estimation error gets larger (overfitting), while 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.

Exercise 4.8 We consider a learning problem with 1-Lipschitz-continuous loss (with respect to the second variable), with a function class $f_\theta(x) = \theta^\top \varphi(x)$, with $\|\theta\|_1 \leq D$, and $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$ with $\|\varphi(x)\|_\infty$ almost surely less than R . Given the expected risk $\mathcal{R}(f_\theta)$ and the empirical risk $\hat{\mathcal{R}}(f_\theta)$. Compute an upper-bound of

$$\mathbb{E} \left[\sup_{\|\theta\|_1 \leq 1} |R(f_\theta) - \hat{R}(f_\theta)| \right].$$

4.5.5 From constrained to regularized estimation (\blacklozenge)

In practice, it is preferable to penalize by the norm $\Omega(\theta) = \|\theta\|_2$ instead of constraining (the main reasons being that the hyperparameter is easier to find and the optimization is easier). For simplicity, we consider only the ℓ_2 -norm in this section.

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

$$\hat{\mathcal{R}}(f_\theta) + \frac{\lambda}{2} \|\theta\|_2^2. \quad (4.11)$$

If the loss is always positive, then

$$\frac{\lambda}{2} \|\hat{\theta}\|_2^2 \leq \hat{\mathcal{R}}(f_{\hat{\theta}}) + \frac{\lambda}{2} \|\hat{\theta}\|_2^2 \leq \hat{\mathcal{R}}(f_0),$$

leading to a bound $\|\hat{\theta}\|_2 = O(1/\sqrt{\lambda})$. Thus, with $D = O(1/\sqrt{\lambda})$ in the bound above, this leads to a deviation of $O(1/\sqrt{\lambda n})$, which is not optimal.

We now cite without proof an interesting stronger result using the strong convexity of the squared ℓ_2 -norm.

Proposition 4.5 (Fast rates for regularized objectives [98]) *Assume a G -Lipschitz-continuous **convex** loss function, linear prediction functions with $\mathcal{F} = \{f_\theta(x) = \theta^\top \varphi(x), \|\theta\|_2 \leq D\}$, where $\mathbb{E}\|\varphi(x)\|_2^2 \leq R^2$. Let $\hat{\theta}_\lambda \in \mathbb{R}^d$ be the minimizer of the regularized empirical risk in Eq. (4.11), then:*

$$\mathbb{E}[\mathcal{R}(f_{\hat{\theta}_\lambda})] \leq \inf_{\theta \in \mathbb{R}^d} \left\{ \mathcal{R}(f_\theta) + \frac{\lambda}{2} \|\theta\|_2^2 \right\} + \frac{32G^2 R^2}{\lambda n}.$$

Note that we obtain a “fast rate” in $O(R^2/(\lambda n))$, which has a better dependence in n , but depends on 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

$$\mathbb{E}[\mathcal{R}(f_{\hat{\theta}_\lambda})] \leq \mathcal{R}(f_{\theta_*}) + O\left(\frac{GR}{\sqrt{n}} \|\theta_*\|_2\right).$$

This is the similar result as for Chapter 3, but now for all Lipschitz-continuous losses.

Extensions and improvements. When dealing with binary classification, or more generally discrete outputs, further analysis can be carried through, with potentially different convergence rates for the convex surrogate which is used 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., [64, 6]).

There are other ways of obtaining generic generalization bounds like presented in this section, such as PAC-Bayesian analysis [32, 110] or stability-based arguments [27].

4.6 Relationship with asymptotic statistics (◆)

In this last section, we will relate the non-asymptotic analysis presented in this chapter to results from asymptotic statistics (see the comprehensive book [103], which presents this large literature).

To make this concrete, we will assume that we have a set of models $\mathcal{F} = \{f_\theta : \mathcal{X} \rightarrow \mathbb{R}, \theta \in \mathbb{R}^d\}$ parameterized by a vector $\theta \in \mathbb{R}^d$, and we consider the empirical risk and expected risks (with a slight overloading of notations):

$$\mathcal{R}(\theta) = \mathcal{R}(f_\theta) = \mathbb{E}[\ell(y, f_\theta(x))] \quad \text{and} \quad \hat{\mathcal{R}}(\theta) = \hat{\mathcal{R}}(f_\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i)).$$

We assume that we have a loss function $\ell : \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}$ (such as for regression or any of the convex surrogates for classification), which is sufficiently differentiable with respect to the second variable, so that results from [103] (e.g., Theorems 5.21 or 5.41 on “M-estimation”, which cover empirical risk minimization). In this section, we will only report their final result and provide an intuitive justification.

We assume that $\theta_* \in \mathbb{R}^d$ is a minimizer of $\mathcal{R}(\theta)$, and that the Hessian $\mathcal{R}''(\theta_*)$ is positive-definite (it has to be positive semi-definite as θ_* is a minimizer, we assume invertibility on top of it).

We let $\hat{\theta}_n$ be a minimizer of $\hat{\mathcal{R}}$. Since $\mathcal{R}'(\theta_*) = 0$, and $\hat{\mathcal{R}}'(\theta_*) = \frac{1}{n} \sum_{i=1}^n \frac{\partial \ell(y_i, f_\theta(x))}{\partial \theta}$, by the law of large numbers, $\hat{\mathcal{R}}'(\theta_*)$ tends to $\mathcal{R}'(\theta_*) = 0$ (e.g., almost surely), and we should thus expect that $\hat{\theta}_n$ (which is defined through $\hat{\mathcal{R}}'(\hat{\theta}_n) = 0$) tends to θ_* (all these statements can be made rigorous, see [103]).

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 normal with mean zero and covariance matrix $\frac{1}{n} \mathbb{E} \left[\left(\frac{\partial \ell(y, f_{\theta}(x))}{\partial \theta} \right) \left(\frac{\partial \ell(y, f_{\theta}(x))}{\partial \theta} \right)^{\top} \Big|_{\theta=\theta_*} \right] = \frac{1}{n} G(\theta_*)$. Therefore, we (intuitively) obtain that $\hat{\theta}_n$ is asymptotically normal 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:

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

For example, for well specified linear regression (like analyzed in Chapter 3), it turns out that we have $G(\theta_*) = \sigma^2 H(\theta_*)$ (proof left as an exercise), and thus we recover the rate $\sigma^2 d/n$.

Benefits of the asymptotic analysis. As shown above, the asymptotic analysis gives a precise picture of the asymptotic behavior of empirical risk minimization. Much more than simply providing an upper-bound on $\mathbb{E}[\mathcal{R}(\hat{\theta}_n) - \mathcal{R}(\theta_*)]$, it gives also a limit normal 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.

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 not possible to tell without further analysis when the asymptotic behavior will kick in. Sometimes, this is for reasonably small n , sometimes for large n . Further asymptotic expansions can be carried out, but small sample effects are hard to characterize, in particular when the underlying dimension d gets large.

Bridging the gap. Studying the validity of the asymptotic expansion described above can be done in several ways. See, e.g., [87] (and references therein) for finite-dimensional models, and Chapter 7 for results similar to $\sigma^2 d/n$ when the dimension of the feature space gets infinite.

Chapter 5

Optimization for machine learning

Chapter summary

- Gradient descent: the workhorse first-order algorithm for optimization, which converges exponentially fast for well-conditioned convex problems.
- Stochastic gradient descent (SGD): the workhorse first-order algorithm for large scale machine learning, which converges as $1/t$ or $1/\sqrt{t}$, where t is the number of iterations.
- Generalization bounds through stochastic gradient descent: with only a single pass on the data, there is no risk of overfitting and we obtain generalization bounds for unseen data.
- Variance reduction: when minimization strongly-convex finite sums, this class of algorithms are exponentially convergent while having a small iteration complexity.

In this chapter, we present optimization algorithms based on gradient descent and analyze their performance, mostly on convex functions. We will consider generic algorithms that have applications beyond machine learning, and algorithms dedicated to machine learning (such as stochastic gradient methods). See [80, 31] for further details.

5.1 Optimization in machine learning

In supervised machine learning, we are given n i.i.d. samples (x_i, y_i) , $i = 1, \dots, n$ of a couple of random variables (x, y) on $\mathcal{X} \times \mathcal{Y}$ and the goal is to find a predictor $f : \mathcal{X} \rightarrow \mathbb{R}$ with a small risk

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

where $\ell : \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}$ is a loss function. This loss is typically convex in the second argument (see Chapter 4), which is thus considered as a weak assumption.

In the empirical risk minimization approach described in Chapter 4, we choose the predictor by minimizing the empirical risk over a parameterized set of predictors, potentially with regularization. For a parameterization $\{f_\theta\}_{\theta \in \mathbb{R}^d}$ and a regularizer $\Omega : \mathbb{R}^d \rightarrow \mathbb{R}$ (e.g., $\Omega(\theta) = \|\theta\|_2^2$ or $\Omega(\theta) = \|\theta\|_1$), this

requires to minimize

$$F(\theta) := \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i)) + \Omega(\theta).$$

In optimization, the function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is called the *objective function*.

In general, the minimizer has no closed form. Even when it has one (e.g., linear predictor and square loss in Chapter 3), it could be expensive to compute for large problems. We thus resort to iterative algorithms.

Accuracy of iterative algorithms. Solving optimization problems to high accuracy is computationally expensive, and the goal is not to minimize the training objective, but the error on unseen data.

Then, which accuracy is satisfying in machine learning? If the algorithm returns $\hat{\theta}$ and $\theta_* \in \arg \min_{\theta} \mathcal{R}(f_\theta)$, we have the risk decomposition (where the approximation error due to the use of a specific set of models f_θ , $\theta \in \Theta$ is ignored):

$$\mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) = \underbrace{\left\{ \mathcal{R}(f_{\hat{\theta}}) - \hat{\mathcal{R}}(f_{\hat{\theta}}) \right\}}_{\leq \text{estimation error}} + \underbrace{\left\{ \hat{\mathcal{R}}(f_{\hat{\theta}}) - \hat{\mathcal{R}}(f_{\theta_*}) \right\}}_{\leq \text{optimization error}} + \underbrace{\left\{ \hat{\mathcal{R}}(f_{\theta_*}) - \mathcal{R}(f_{\theta_*}) \right\}}_{\leq \text{estimation error}}.$$

It is thus sufficient to reach an optimization accuracy of the order of the estimation error (usually of the order $O(1/\sqrt{n})$ or $O(1/n)$, see Chapter 3 and Chapter 4).

In this chapter, we will first look at the minimization without focusing on machine learning problems (Section 5.2), with both smooth and non-smooth objective functions. We will then look at stochastic gradient descent in Section 5.4, which can be used to obtain bounds on both the training risk and the testing risk. We then briefly present variance reduction in Section 5.4.2.



θ_* may mean different things in optimization and machine learning: minimizer of the regularized empirical risk, or minimizer of the expected risk. For the sake of clarity, we will use the notation η_* for the minimizer of empirical (potential regularized risk), that is, when we look at optimization problems, and θ_* for the minimizer of the expected risk, that is, when we look at statistical problems.



Sometimes, we mention solving a problem with *high* precision. This corresponds to a *low* optimization error.

5.2 Gradient descent

Suppose we want to solve, for a function $F : \mathbb{R}^d \rightarrow \mathbb{R}$, the optimization problem

$$\min_{\theta \in \mathbb{R}^d} F(\theta).$$

We assume that we are given access to certain “oracles”: the k -th-order oracle corresponds to the access to: $\theta \mapsto (F(\theta), F'(\theta), \dots, F^{(k)}(\theta))$, that is all partial derivatives up to order k . All algorithms will call these oracles and thus their computational complexity will depend directly on the complexity of this oracle. For example, for least-squares with a design matrix in $\mathbb{R}^{n \times d}$, computing a single gradient of the empirical risk costs $O(nd)$.

In this section, for the algorithms and proofs, we do not assume that the function F is the regularized empirical risk, but this situation will be our motivating example throughout. We will study the following first-order algorithm.

Algorithm 5.1 (Gradient descent (GD)) Pick $\theta_0 \in \mathbb{R}^d$ and for $t \geq 1$, let

$$\theta_t = \theta_{t-1} - \gamma_t F'(\theta_{t-1}),$$

for a well (potentially adaptively) chosen step-size sequence $(\gamma_t)_{t \geq 1}$.

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, either decaying, either through a line search (see, e.g., https://en.wikipedia.org/wiki/Line_search). In practice, using some form of line search is strongly advantageous and is implemented in most applications. 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., [81]).

We first start with the simplest example, namely quadratic convex functions.

5.2.1 Simplest analysis: ordinary least-squares

We start with a case where the analysis is explicit: ordinary least squares (see Chapter 3 for the statistical analysis). Let $\Phi \in \mathbb{R}^{n \times d}$ be the design matrix and $y \in \mathbb{R}^n$ the vector of responses. Least-squares estimation amounts to finding a minimizer η_* of

$$F(\theta) = \frac{1}{2n} \|\Phi\theta - y\|_2^2.$$

 A factor of $\frac{1}{2}$ has been added compared to Chapter 3 to get nicer looking gradients.

The gradient of F is $F'(\theta) = \frac{1}{n} \Phi^\top (\Phi\theta - y) = \frac{1}{n} \Phi^\top \Phi\theta - \frac{1}{n} \Phi^\top y$. Thus, denoting $H = \frac{1}{n} \Phi^\top \Phi \in \mathbb{R}^{d \times d}$ the Hessian matrix (equal for θ), minimizers η_* are characterized by

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

Since $\frac{1}{n} \Phi^\top y \in \mathbb{R}^d$ is in the column space of H , there is always a minimizer, but unless H is invertible, the minimizer is not unique. But all minimizers η_* have the same function value $F(\eta_*)$, and we have, from a simple exact Taylor expansion (and using $F'(\eta_*) = 0$):

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

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 convexity of the objective, we have $0 \leq \mu \leq L$. We denote by $\kappa = \frac{L}{\mu} \geq 1$ the *condition number*.

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

Closed-form expression. Gradient descent iterates with fixed step-size $\gamma_t = \gamma$ can be computed in closed form:

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

leading to

$$\theta_t - \eta_* = \theta_{t-1} - \eta_* - \gamma H(\theta_{t-1} - \eta_*) = (I - \gamma H)(\theta_{t-1} - \eta_*),$$

that is, we have a linear recursion, and we can unroll the recursion, and now write

$$\theta_t - \eta_* = (I - \gamma H)^t (\theta_0 - \eta_*).$$

We can now look at various measures of performance:

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

The two optimization performance measures differ by the presence of the Hessian matrix $\textcolor{blue}{H}$ in the measure based on function values.

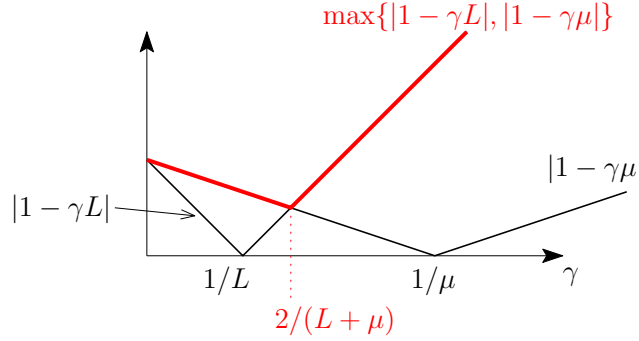
Convergence in distance to minimizer. If we hope to have $\|\theta_t - \eta_*\|_2^2$ going to zero, we need to have a single minimizer η_* , and thus H has to be invertible, that is $\mu > 0$. Given the form of $\|\theta_t - \eta_*\|_2^2$, we simply need to bound the eigenvalues of $(I - \gamma H)^{2t}$.

The eigenvalues of $(I - \gamma H)^{2t}$ are exactly $(1 - \gamma \lambda)^{2t}$ for λ an eigenvalue of H (which are all in the interval $[\mu, L]$). Thus all the eigenvalues of $(I - \gamma H)^{2t}$ have magnitude less than

$$\left(\max_{\lambda \in [\mu, L]} |1 - \gamma \lambda| \right)^{2t}.$$

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 geometrix “proof” below.



- 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.

With the weaker choice $\gamma = 1/L$, we get:

$$\|\theta_t - \eta_*\|_2^2 \leq \left(1 - \frac{1}{\kappa}\right)^{2t} \|\theta_0 - \eta_*\|_2^2,$$

which is often referred to as exponential, geometric, or also linear convergence.

⚠ The denomination “linear” is sometimes confusing and corresponds to a number of significant digits that grows linearly with the number of iterations.

We can further bound $\left(1 - \frac{1}{\kappa}\right)^{2t} \leq \exp(-1/\kappa)^{2t} = \exp(-2t/\kappa)$, 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 optimum of ε , we need at most $t = \frac{\kappa}{2} \log \frac{1}{\varepsilon}$ iterations.

For $\kappa = +\infty$, then the result remains true, but simply say that for all minimizers $\|\theta_t - \eta_*\|_2^2 \leq \|\theta_0 - \eta_*\|_2^2$, which is a good sign (the algorithm does not move away from minimizers) but not indicative of any form of convergence. We will need to use a different criterion.

Convergence in function values. Using the same step-size $\gamma = 1/L$ as above, and using the upper-bound on eigenvalues of $(I - \gamma H)^{2t}$, we get

$$F(\theta_t) - F(\eta_*) \leq \left(1 - \frac{1}{\kappa}\right)^{2t} [F(\theta_0) - F(\eta_*)] \leq \exp(-2t/\kappa) [F(\theta_0) - F(\eta_*)] \quad (5.1)$$

When $\kappa < \infty$ (that is, $\mu > 0$), then we also obtain linear convergence for this criterion, but when $\kappa = \infty$, this is non-informative.

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

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


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

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

$$F(\theta_t) - F(\eta_*) \leq \frac{1}{4t\gamma} \|\theta_0 - \eta_*\|_2^2. \quad (5.2)$$

We can make the following observations:

-  The convergence results in $\exp(-t/\kappa)$ in Eq. (5.1) for invertible Hessians or $1/t$ in general in Eq. (5.2) are only upper-bounds! It is good to understand the gap between the bounds and the actual performance, as this is possible for quadratic objective functions.

For the exponentially convergent case, the lowest eigenvalue 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, in order to see an $O(1/t)$ convergence rate in practice, we need to have sufficiently many small eigenvalues, and as t grows, we often go to a local linear convergence phase where the smallest non zero eigenvalue of H kicks in. See simulations and exercise below.

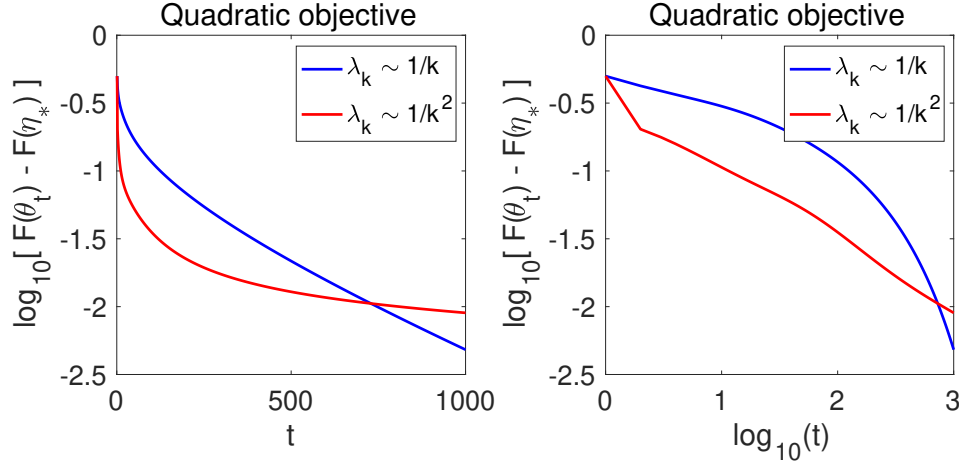
Exercise 5.1 Let μ_+ be the smallest non-zero eigenvalue of H . Show that gradient descent is linearly convergence with the contracting rate $(1 - \mu_+/L)$.

- From errors to number of iterations: the bound in Eq. (5.1) says that after t steps, the reduction in suboptimality in function values is multiplied by $\varepsilon = \exp(-2t/\kappa)$. This can be reinterpreted 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 the matrix Φ , then the conjugate gradient algorithm can be used with convergence rates in $\exp(-t/\sqrt{\kappa})$ and $1/t^2$ (see [47]). With only access to gradients of F (which is a bit weaker) Nesterov acceleration (see below) will also lead to the same convergence rates, which are then optimal (for a sense to be defined later).

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

Experiments. We consider two quadratic optimization problems in dimension $d = 1000$, with two different decays of eigenvalues (λ_k) for the Hessian matrix H , one as $1/k$ (in blue below) and one in $1/k^2$ (in red below), and for which we plot the performance for function values, both in semi-logarithm plots (left) and full-logarithm plots (right). For slow decays (blue), we see the linear convergence kicking in, while for fast decays (red), the rates in $1/t$ dominate.



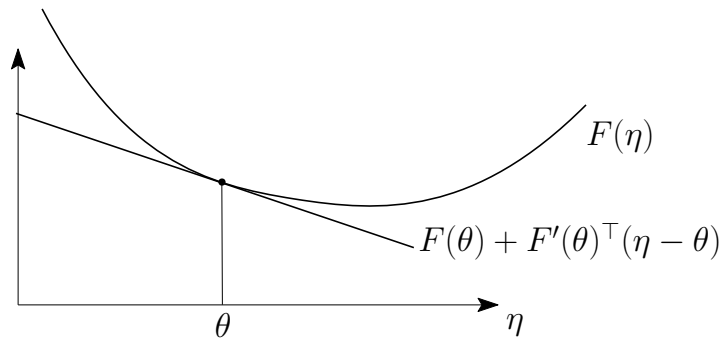
5.2.2 Convex functions and their properties

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 the analysis, not to define the algorithm.

Definition 5.1 (Convex function) *A differentiable function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is said convex if and only if*

$$F(\eta) \geq F(\theta) + F'(\theta)^\top (\eta - \theta), \quad \forall \eta, \theta \in \mathbb{R}^d. \quad (5.3)$$

This corresponds to the function F being above its tangent at θ , as illustrated below.



If f is twice-differentiable, this is equivalent to requiring $F''(x) \succcurlyeq 0$, $\forall x \in \mathbb{R}^d$ (here \succcurlyeq denotes the semidefinite partial ordering—also called Loewner order—characterized by $A \succcurlyeq B \Leftrightarrow A - B$ is positive semidefinite, see [28, 22]).

An important consequence that we will use a lot in this chapter is, for all $\theta \in \mathbb{R}^d$ (and using $\eta = \eta_*$)

$$F(\eta_*) \geq F(\theta) + F'(\theta)^\top (\eta_* - \theta) \Leftrightarrow F(\theta) - F(\eta_*) \leq F'(\theta)^\top (\theta - \eta_*), \quad (5.4)$$

that is the distance to optimum in function values is upperbounded by a function of the gradient.

A more general definition of convexity is that $\forall x, y \in \mathbb{R}^d$ and $\alpha \in [0, 1]$,

$$F(\alpha\eta + (1 - \alpha)\theta) \leq \alpha F(\eta) + (1 - \alpha)F(\theta),$$

which generalizes to the usual Jensen's inequality.

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

$$F\left(\int \theta d\mu(\theta)\right) \leq \int F(\theta) d\mu(\theta).$$

In words: “the image of the average is smaller than the average of the images”.

The class of convex functions satisfies the following stability properties (proofs left as an exercise), for more properties on convex functions, see [28]:

- If $(F_j)_{j \in \{1, \dots, m\}}$ are convex and $(\alpha_j)_{j \in \{1, \dots, m\}}$ are nonnegative, then $\sum_{j=1}^m \alpha_j F_j$ is convex.
- If $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex and $A : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$ is linear then $F \circ A : \mathbb{R}^{d'} \rightarrow \mathbb{R}$ is convex.

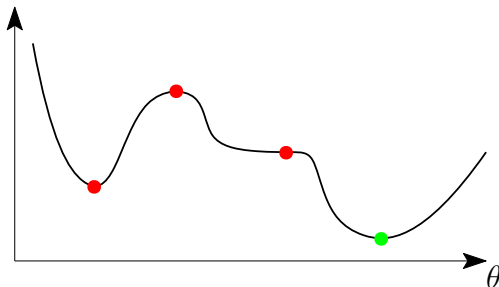
Classical machine learning example. Problems of the form in Eq. (5.1) are convex if the loss ℓ is convex in the second variable, $f_\theta(x)$ is linear in θ , and Ω is convex.

Global optimality from local information. It is also worth emphasizing on the following property (immediate from the definition).

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

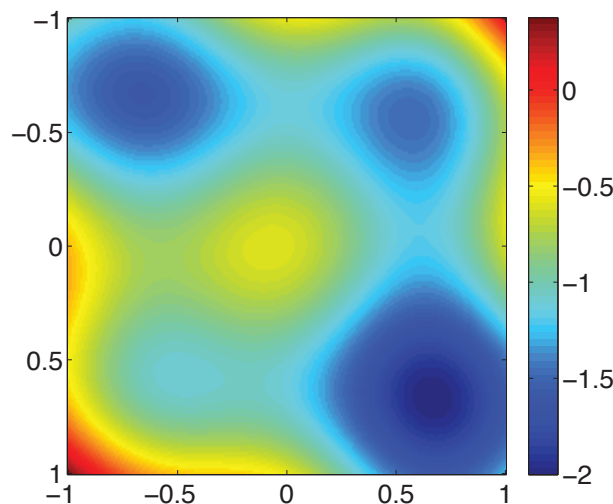
$$F'(\eta_*) = 0.$$

This implies that for convex functions, we only need to look for stationary points. This is *not* the case for potentially non-convex functions. For example, in one dimension below, all red points are stationary points which are not the global minimum (which is in green).



The situation is even more complex in higher dimensions. Note that without convexity assumptions, optimization of Lipschitz-continuous functions will need exponential time in dimension in the worst case (see Section 11.2.2).

Exercise 5.2 *Identify all stationary points in the function in \mathbb{R}^2 depicted below.*



5.2.3 Analysis of GD for strongly convex and smooth functions

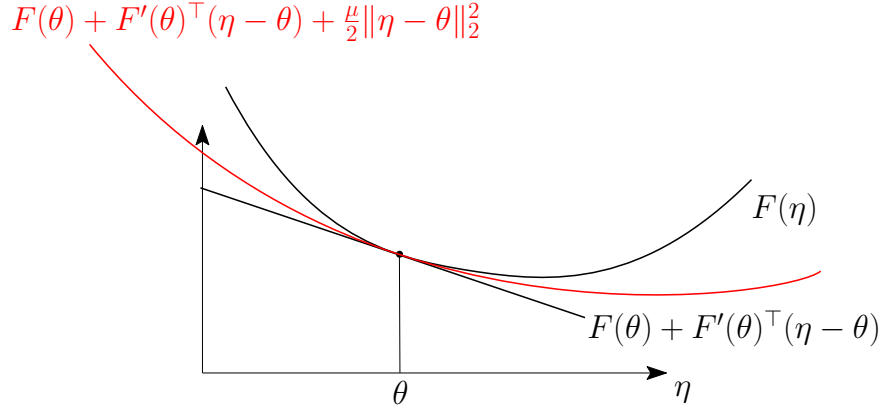
The analysis of optimization algorithms requires assumptions on the objective functions, like the ones introduced in this section. From these assumptions, additional properties are derived (typically inequalities), and then most convergence proofs look for a “Lyapunov function” (sometimes called a potential function) that goes down along the iterations. More precisely, if $V : \mathbb{R}^d \mapsto \mathbb{R}_+$ is such that $V(\theta_t) \leq (1 - \alpha)V(\theta_{t-1})$, then $V(\theta_t) \leq (1 - \alpha)^t V(\theta_0)$ and we obtain linear convergence. The art is then to find the appropriate Lyapunov function.

We first consider an assumption allowing exponential convergence rates.

Definition 5.2 (Strong convexity) *A differentiable function F is said μ -strongly convex, with $\mu > 0$, if and only if*

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

The function F is strongly-convex if and only if the function F is strictly above its tangent and the difference is at least quadratic in the distance to the point where the two coincide. This notably allows to defined quadratic lower bounds on F . See below.



For twice differentiable functions, this is equivalent to $F'' \succcurlyeq \mu I$ (see [80]).

Strong convexity through regularization. When an objective function F is convex, then $F + \frac{\mu}{2}\|\cdot\|_2^2$ is μ -strongly convex (proof left as an exercise). In practice, in machine learning problems, with linear models, so that the empirical risk is convex, strong convexity most often comes from the regularizer (and thus μ decays with n), leading to condition numbers that grow with n .

Lojasiewicz 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 optimality:

Lemma 5.1 (Lojasiewicz inequality) *If F is differentiable and μ -strongly convex with unique minimizer η_* , then we have:*

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

Proof The right-hand side in Definition 5.2 is strongly convex in θ and minimized with $\tilde{\theta} = \theta - \frac{1}{\mu}F'(\theta)$. Plugging this value into the bound and taking $\eta = \eta_*$ in the left-hand side we get

$$F(\eta_*) \geq F(\theta) - \frac{1}{\mu}\|F'(\theta)\|_2^2 + \frac{1}{2\mu}\|F'(\theta)\|_2^2 = F(\theta) - \frac{1}{2\mu}\|F'(\theta)\|_2^2.$$

The conclusion follows by rearranging. ■

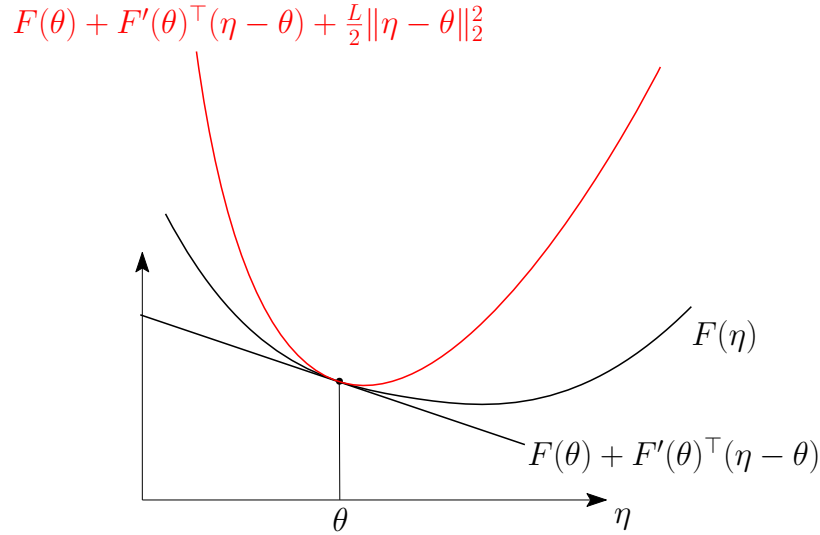
In order to obtain exponential convergence rates, strong-convexity is typically associated with smoothness, which we now define.

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

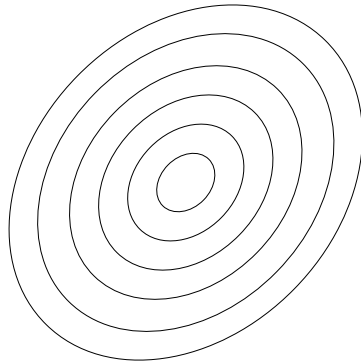
$$|F(\eta) - F(\theta) - F'(\theta)^\top(\eta - \theta)| \leq \frac{L}{2}\|\eta - \theta\|^2, \quad \forall \theta, \eta \in \mathbb{R}^d. \quad (5.5)$$

This is equivalent to F having a L -Lipschitz-continuous gradient, i.e., $\|F'(\theta) - F'(\eta)\|_2^2 \leq L^2 \|\theta - \eta\|_2^2$, $\forall \theta, \eta \in \mathbb{R}^d$. For twice differentiable functions, this is equivalent to $-LI \preceq F''(\theta) \preceq LI$ (see [80]).

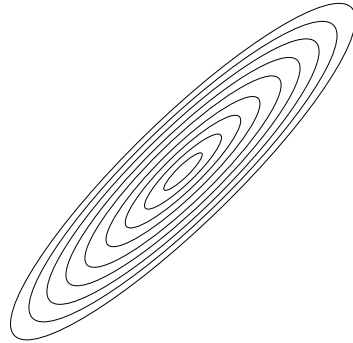
Note that when F is convex and L -smooth, we have a quadratic upper-bound which is tight at any given point (strong convexity implies the corresponding lower bound). See below.



When a function is both smooth and strongly convex, we denote by $\kappa = L/\mu \geq 1$ its condition number. See examples below of level sets of functions with varying condition numbers: the condition number impacts the shapes of the level sets.

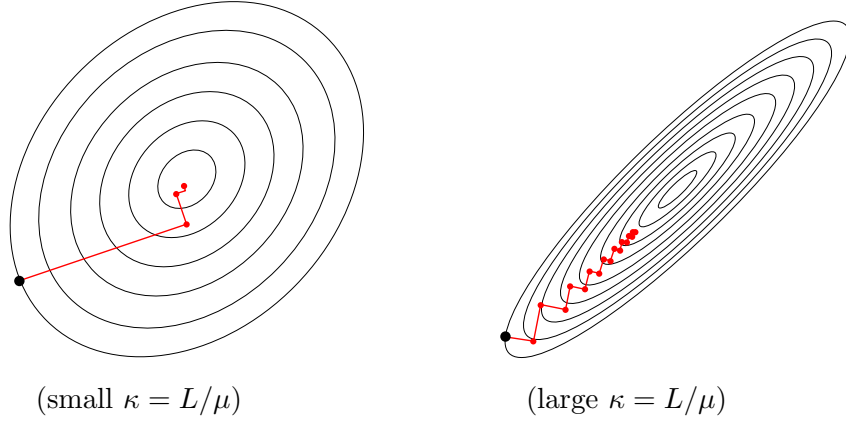


(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

The performance of gradient descent will depend on this condition number (see steepest descent below, that is, gradient descent with exact line search): with small condition number (left), we get fast convergence, while for a large condition number (right), we get oscillations.



For machine learning problems, for linear predictions and smooth losses (square or logistic), then 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 $1/\sqrt{n}$, leading to condition numbers between \sqrt{n} and n .

In this context, gradient descent on the empirical risk, is often called a “batch” technique, because all the data points are accessed at every iteration.

In the next theorem, we show that gradient descent converges exponentially for such smooth and strongly-convex problems.

Theorem 5.1 (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 $(\theta_t)_{t \geq 0}$ of GD on F satisfy*

$$F(\theta_t) - F(\eta_*) \leq \left(1 - \frac{\mu}{L}\right)^t (F(\theta_0) - F(\eta_*)) \leq \exp(-t\mu/L) (F(\theta_0) - F(\eta_*)).$$

Proof By the smoothness inequality in Eq. (5.5) applied to θ_{t-1} and $\theta_{t-1} - F'(\theta_{t-1})/L$, we have the following descent property, with $\gamma_t = 1/L$,

$$\begin{aligned} F(\theta_t) &= F(\theta_{t-1} - F'(\theta_{t-1})/L) \leq F(\theta_{t-1}) + F'(\theta_{t-1})^\top (-F'(\theta_{t-1})/L) + \frac{L}{2} \|-F'(\theta_{t-1})/L\|_2^2 \\ &= F(\theta_{t-1}) - \|F'(\theta_{t-1})\|_2^2 / L + \frac{1}{2L} \|F'(\theta_{t-1})\|_2^2. \end{aligned}$$

Rearranging, we get

$$F(\theta_t) - F(\eta_*) \leq (F(\theta_{t-1}) - F(\eta_*)) - \frac{1}{2L} \|F'(\theta_{t-1})\|_2^2.$$

Using Lemma 5.1, it follows

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

We conclude by a recursion. ■

We can make the following observations:

- As mentioned before, we necessarily have $\mu \leq L$; the ratio $\kappa := L/\mu$ is called the *condition number*.
- If we only assume that the function is smooth and convex (not strongly convex), then GD with constant step-size $\gamma = 1/L$ also converges when a minimizer exists, but at a slower rate in $O(1/t)$. See Section 5.2.4 below.
- Choosing the step-size only requires an upper bound L on the smoothness constant (in case it is over-estimated, the convergence rate only degrades slightly).

Exercise 5.3 Compute all constants for ℓ_2 -regularized logistic regression.

Adaptivity. Note that gradient descent is adaptive to strong convexity: the exact same algorithm applies to both strongly convex and convex cases, and the two bounds apply. This adaptivity is important in practice, as often, locally around the global optimum, the strong convexity constant converges to the minimal eigenvalue of the Hessian at η_* , which can be very significantly larger than μ (the global constant).

Fenchel conjugate (♦). Given some convex function $F : \mathbb{R}^d \rightarrow \mathbb{R}$, an important tool is the Fenchel conjugate F^* defined as $F^*(\alpha) = \sup_{\theta \in \mathbb{R}^d} \alpha^\top \theta - F(\theta)$. This is crucial when dealing with convex duality (which we will not cover in this chapter); see [28] for details.

5.2.4 Analysis of GD for convex and smooth functions (♦)

In order to obtain the $1/t$ convergence rate without strong-convexity, 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 5.3 (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:*

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

Moreover, we have: $F(\theta) \geq F(\eta) + F'(\eta)^\top (\theta - \eta) + \frac{1}{2L} \|F'(\theta) - F'(\eta)\|_2^2$.

Proof We will show the second inequality, which implies the first one by applying it twice with η and θ swapped, and summing them.

- Define $H(\theta) = F(\theta) - \theta^\top F'(\eta)$. The function $H : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex with global minimum at η , since $H'(\theta) = F'(\theta) - F'(\eta)$, which is equal to zero for $\theta = \eta$. The function H is also L -smooth.
- We can apply the definition of smoothness: $H(\eta) \leq H(\theta - \frac{1}{L} H'(\theta)) \leq H(\theta) + H'(\theta)^\top (-\frac{1}{L} H'(\theta)) + \frac{L}{2} \|-\frac{1}{L} H'(\theta)\|_2^2$, which is thus less than $H(\theta) - \frac{1}{2L} \|H'(\theta)\|_2^2$.
- This leads to $F(\eta) - \eta^\top F'(\eta) \leq F(\theta) - \theta^\top F'(\eta) - \frac{1}{2L} \|F'(\theta) - F'(\eta)\|_2^2$, which leads to the desired inequality by shuffling terms.

■

We can now state the following convergence result for gradient descent with potentially no strong-convexity. Up to constants, we obtain the same rate as for quadratic functions in Eq. (5.2).

Theorem 5.2 (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 $(\theta_t)_{t \geq 0}$ of GD on F satisfy*

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

Proof Following [15], 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. We can split the difference in Lyapunov functions in three terms (each with its own color):

$$V_t(\theta_t) - V_{t-1}(\theta_{t-1}) = t[F(\theta_t) - F(\theta_{t-1})] + \textcolor{red}{F(\theta_t) - F(\eta_*)} + \frac{L}{2} \|\theta_t - \eta_*\|_2^2 - \frac{L}{2} \|\theta_{t-1} - \eta_*\|_2^2.$$

In order to bound it, we use:

- We use $\textcolor{blue}{F(\theta_t) - F(\theta_{t-1})} \leq -\frac{1}{2L} \|F'(\theta_{t-1})\|_2^2$ like in the proof of Theorem 5.1.
- We use $\textcolor{red}{F(\theta_t) - F(\eta_*)} \leq F'(\theta_{t-1})^\top (\theta_{t-1} - \eta_*)$, as a consequence of convexity (function above the tangent at θ_{t-1}), as in Eq. (5.4).
- We get $\frac{L}{2} \|\theta_t - \eta_*\|_2^2 - \frac{L}{2} \|\theta_{t-1} - \eta_*\|_2^2 = -L\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + \frac{L\gamma^2}{2} \|F'(\theta_{t-1})\|_2^2$ by expanding the square.

This leads to, with the step-size $\gamma = 1/L$:

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

which leads to

$$t[F(\theta_t) - F(\eta_*)] \leq V_t(\theta_t) \leq V_0(\theta_0) = \frac{L}{2} \|\theta_0 - \eta_*\|_2^2,$$

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

The proof above is on purpose mysterious: the choice of Lyapunov function seems arbitrary at first, but all inequalities lead to nice cancellations. These proofs are sometimes hard to design. For an interesting line of work trying to automate these proofs, see <https://francisbach.com/computer-aided-analyses/>.

5.2.5 Beyond gradient descent (◆)

While gradient descent is the simplest algorithm with a simple analysis, there are multiple extensions that we will only briefly mention (see more details in [78, 79]):

Nesterov acceleration. For convex functions, a simple modification of gradient descent allows to obtain better convergence rates. The algorithm is as follows, and is based on updating the following iterates:

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L}F'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{t-1}{t+2}(\theta_t - \theta_{t-1}).\end{aligned}$$

This simple modification dates back to Nesterov in 1983, and leads to the following convergence rate $F(\theta_t) - F(\eta_*) \leq \frac{2L\|\theta_0 - \eta_*\|^2}{(t+1)^2}$.

For strongly convex functions, the algorithm has a similar form as for convex functions, but with all coefficients which are independent from t :

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L}g'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{1 - \sqrt{\mu/L}}{1 + \sqrt{\mu/L}}(\theta_t - \theta_{t-1}),\end{aligned}$$

and the convergence rate is $F(\theta_t) - F(\eta_*) \leq L\|\theta_0 - \eta_*\|^2(1 - \sqrt{\mu/L})^t$, 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.

Moreover, the last two rates are known to be optimal for the considered problems: for algorithms that access gradient and combine them linearly to select a new query point, it is not possible to have better dimension-independent rates. See [79] for more details.

Newton method. Given θ_{t-1} , the Newton method minimizes the second-order Taylor expansion around θ_{t-1} :

$$F(\theta_{t-1}) + F'(\theta_{t-1})^\top(\theta - \theta_{t-1}) + \frac{1}{2}(\theta - \theta_{t-1})^\top F''(\theta_{t-1})(\theta - \theta_{t-1}),$$

which leads to $\theta_t = \theta_{t-1} - F''(\theta_{t-1})^{-1}F'(\theta_{t-1})$, which is an expensive iteration, as the running-time complexity is $O(d^3)$ in general to solve the linear system. It leads to local quadratic convergence: If $\|\theta_{t-1} - \theta_*\|$ small enough, for some constant C , we have $(C\|\theta_t - \theta_*\|) = (C\|\theta_{t-1} - \theta_*\|)^2$. See [28] for more details, and for conditions for global convergence.

Note that for machine learning problems, quadratic convergence may be an overkill compared to the computational complexity of each iteration, since cost functions are averages of n terms and naturally have some uncertainty of order $O(1/\sqrt{n})$.

Proximal gradient descent (♦). Many optimization problems are said “composite”, that is, the objective function F is the sum of a smooth function G and a non-smooth function H (such as a norm). It turns out that a simple modification of gradient descent allows to benefit from the fast convergence rates of smooth optimization.

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

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

where, for a L -smooth function G , the objective function above is an upper-bound of $G(\theta)$ which is tight at θ_{t-1} .

While this reformulation does not bring much for gradient descent, we can extend this to the composite problem, and consider the iteration

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

where H is left as is. It turns out that the convergence rates for $G + H$ are the same as smooth optimization, with potential acceleration [79, 19].

The crux is to be able to compute the step above, 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 gradient descent. When H is the ℓ_1 -norm, that is $H = \lambda \|\cdot\|_1$, this can be shown to be soft-thresholding step, as for each coordinate $\theta_i = (|\eta_i| - \lambda/L)_+ \frac{\eta_i}{|\eta_i|}$ (proof left as an exercise).

5.2.6 Non-convex objective functions (◆)

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

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

leading to, summing the inequalities above for all iterations between 1 and t , we get:

$$\frac{1}{t} \sum_{s=1}^t \|F'(\theta_{s-1})\|_2^2 \leq \frac{F(\theta_0) - F(\eta_*)}{t}.$$

Thus there has to be one s in $\{0, \dots, t-1\}$ for which $\|F'(\theta_s)\|_2^2 \leq O(1/t)$. Note that this does not imply that any of the iterates is close to a stationary point.

5.3 Gradient methods on non-smooth problems

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

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

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

Exercise 5.4 Show that if F is differentiable, this is equivalent to the assumption $\|F'(\theta)\|_2 \leq B$, $\forall \theta \in \mathbb{R}^d$. Without additional assumptions, this setting is usually referred to as non-smooth optimization.

From gradients to subgradients. We can apply non-smooth optimization to objective functions which are not differentiable (such as the hinge loss). For convex Lipschitz-continuous objectives, the function is almost everywhere differentiable. In points where it is not, then one can define the set of slopes of lower-bounding tangents as the *subdifferential*, and any element of it as a *subgradient*. The gradient descent iteration is then meant as using any subgradient instead of $F'(\theta_{t-1})$. The method is then referred to as the subgradient method (it is not a descent method anymore, that is, the function values may go up once in a while).

Convergence rate of the subgradient method. We can prove convergence of the gradient descent algorithm, now with a decaying step-size, and a slower rate than for smooth functions.

Theorem 5.3 (Convergence of the subgradient method) *Assume that F is convex, B -Lipschitz-continuous, and admits a minimizer η_* that satisfies $\|\eta_* - \theta_0\|_2 \leq D$. By choosing $\gamma_t = \frac{D}{B\sqrt{t}}$ then the iterates $(\theta_t)_{t \geq 0}$ of GD on F satisfy*

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

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:

$$\|\theta_t - \eta_*\|_2^2 = \|\theta_{t-1} - \gamma_t F'(\theta_{t-1}) - \eta_*\|_2^2 = \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma_t F'(\theta_{t-1})^\top (\theta_{t-1} - \eta_*) + \gamma_t^2 \|F'(\theta_{t-1})\|_2^2.$$

Combining this with the convexity inequality $F(\theta_{t-1}) - F(\eta_*) \leq F'(\theta_{t-1})^\top (\theta_{t-1} - \eta_*)$ from Eq. (5.4), it follows (also using the boundedness of gradients):

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

and thus, by isolating the distance to optimum in function values:

$$\gamma_t (F(\theta_{t-1}) - F(\eta_*)) \leq \frac{1}{2} \left(\|\theta_{t-1} - \eta_*\|_2^2 - \|\theta_t - \eta_*\|_2^2 \right) + \frac{1}{2} \gamma_t^2 B^2. \quad (5.6)$$

It is sufficient to sum these inequalities to get, for any $\eta_* \in \mathbb{R}^d$,

$$\frac{1}{\sum_{s=1}^t \gamma_s} \sum_{s=1}^t \gamma_s (F(\theta_{s-1}) - F(\eta_*)) \leq \frac{\|\theta_0 - \eta_*\|_2^2}{2 \sum_{s=1}^t \gamma_s} + B^2 \frac{\sum_{s=1}^t \gamma_s^2}{2 \sum_{s=1}^t \gamma_s}.$$

The left-hand side is larger than $\min_{0 \leq s \leq t-1} (F(\theta_s) - F(\eta_*))$ (trivially) and 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 ∞ (to forget the initial condition, sometimes called the “bias”) and $\gamma_t \rightarrow 0$ (to decrease the “variance” term). Let us choose $\gamma_s = \tau / \sqrt{s}$ for some $\tau > 0$. By using the series-integral comparisons below, we get the bound

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

We choose $\tau = D/B$ (which is suggested by optimizing the previous bound when $\log(t) = 0$) which leads to the result. In the proof, we used the following series-integral comparisons for decreasing functions:

$$\sum_{s=1}^t \frac{1}{\sqrt{s}} \geq \int_0^t \frac{ds}{\sqrt{s+1}} = \left[2\sqrt{s+1} \right]_0^t = 2\sqrt{t+1} - 2 \geq \frac{1}{2}\sqrt{t}$$

$$\text{and } \sum_{s=1}^t \frac{1}{s} \leq 1 + \sum_{s=2}^t \frac{1}{s} \leq 1 + \int_0^t \frac{ds}{s} = 1 + \log(t). \quad \blacksquare$$

The proof scheme above is very flexible. It can be extended in the following directions:

- There is no need to know in advance an upper-bound D on the distance to optimum, we then get with the same step-size $\gamma_t = \frac{D}{B\sqrt{t}}$ a rate of the form $\frac{BD}{\sqrt{t}} \left(\frac{\|\theta_0 - \eta_*\|_2^2}{D^2} + (1 + \log(t)) \right)$.
- The algorithm applies to constrained minimization over a convex set, by inserting a projection step at each iteration (the proof, which is using the contractivity of orthogonal projections, is essentially the same).
- The algorithm applies to non-differentiable convex and Lipschitz objective functions (using subgradients, i.e. any vector satisfying Eq. (5.3) 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. This can be done using the “mirror descent” descent framework, and for instance can be applied to obtain multiplicative updates (see, e.g., [59, 60]).
- Often the uniformly averaged iterate is used, as $\frac{1}{t} \sum_{s=0}^{t-1} \theta_s$. Convergence rates (without the $\log t$ factor) can be obtained using Abel summation formula (see <https://francisbach.com/integration-by-parts-abel-transformation/>).
- Stochastic gradients can be used, as presented below (one interpretation is that the subgradient method is so slow that it is robust to noisy gradients).

Exercise 5.5 Compute all constants for ℓ_2 -regularized logistic regression.

5.4 Convergence rate of stochastic gradient descent (SGD)

For machine learning problems, where $F(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i)) + \Omega(\theta)$, at each iteration, the gradient descent algorithm requires to compute a “full” gradient $F'(\theta_{t-1})$ which could be costly as it requires accessing the entire data set. An alternative is to instead only compute *unbiased* stochastic estimations of the gradient $g_t(\theta_{t-1})$, i.e., such that $\mathbb{E}[g_t(\theta_{t-1}) | \theta_{t-1}] = F'(\theta_{t-1})$, which could be much faster to compute.

⚠ 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 only get convergence if the noise magnitudes go to zero.

This leads to the following algorithm.

Algorithm 5.2 (Stochastic gradient descent (SGD)) Choose a step-size sequence $(\gamma_t)_{t \geq 0}$, pick $\theta_0 \in \mathbb{R}^d$ and for $t \geq 0$, let

$$\theta_{t+1} = \theta_t - \gamma_t g_t(\theta_t).$$

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

- (1) **Empirical risk minimization:** If $F(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, f_\theta(x_i))$ then at iteration t we can choose uniformly at random $i(t) \in \{1, \dots, n\}$ and define g_t as the gradient of $\theta \mapsto \ell(y_{i(t)}, f_\theta(x_{i(t)}))$. There exists “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 more running time). 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.

- (2) **Expected risk minimization:** If $F(\theta) = \mathbb{E}[\ell(y, f_\theta(x))]$ then at iteration t we can take a fresh sample (x_t, y_t) and define g_t as the gradient of $\theta \mapsto \ell(y_t, f_\theta(x_t))$, for which, if we swap the orders of expectation and differentiation, we get the unbiasedness. Note here that to preserve the unbiasedness, only a single pass is allowed (otherwise, this would create dependencies that would break it).

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

Note that in practice, multiple passes over the data (that is, using each observation multiple times) lead to better performance. In order to avoid overfitting, either a regularization term is added to the empirical risk, or the SGD algorithm is stopped before its convergence, which is referred to as regularization by “early stopping”.

We can study the two situations above using the latter one, by considering the empirical risk as the expectation with respect to the empirical distribution of the data.



Stochastic gradient descent is not a descent method: the function values often go up.

Under the same usual assumptions on the objective functions, we now study SGD, with the following extra assumptions:

- (H1) unbiased gradient: $\mathbb{E}[g_t(\theta_{t-1}) | \theta_{t-1}] = F'(\theta_{t-1})$, $\forall t$,
- (H2) bounded gradient: $\|g_t(\theta_{t-1})\|_2^2 \leq B^2$, $\forall t$ almost surely

Assumption (H2) could be replaced by other regularity conditions (e.g., Lipschitz-continuous gradients). Assumption (H1) is crucial, and is often obtained by considering independent functions g_t , for which we shall have, $\mathbb{E}[g_t(\cdot)] = F'(\cdot)$.

Theorem 5.4 (Convergence of SGD) Assume that F is convex, B -Lipschitz and admits a minimizer θ_* that satisfies $\|\theta_* - \theta_0\|_2 \leq D$. Assume that the stochastic gradients satisfy (H1-2). Then, choosing $\gamma_t = (D/B)/\sqrt{t}$, the iterates $(\theta_t)_{t \geq 0}$ of SGD on F satisfy

$$\mathbb{E} \left[F(\bar{\theta}_t) - F(\theta_*) \right] \leq DB \frac{2 + \log(t)}{\sqrt{t}}.$$

where $\bar{\theta}_t = (\sum_{s=1}^t \gamma_s \theta_{s-1}) / (\sum_{s=1}^t \gamma_s)$.

Proof We follow essentially the same proof as in the deterministic case, adding some expectations at well chosen places. We have:

$$\begin{aligned} \mathbb{E} \left[\|\theta_t - \theta_*\|_2^2 \right] &= \mathbb{E} \left[\|\theta_{t-1} - \gamma_t g_t(\theta_{t-1}) - \theta_*\|_2^2 \right] \\ &= \mathbb{E} \left[\|\theta_{t-1} - \theta_*\|_2^2 \right] - 2\gamma_t \mathbb{E} \left[g_t(\theta_{t-1})^\top (\theta_{t-1} - \theta_*) \right] + \gamma_t^2 \mathbb{E} \left[\|g_t(\theta_{t-1})\|_2^2 \right]. \end{aligned}$$

We can then compute the expectation of the middle term as:

$$\begin{aligned} \mathbb{E} \left[g_t(\theta_{t-1})^\top (\theta_{t-1} - \theta_*) \right] &= \mathbb{E} \left[\mathbb{E} \left[g_t(\theta_{t-1})^\top (\theta_{t-1} - \theta_*) \middle| \theta_{t-1} \right] \right] \\ &= \mathbb{E} \left[\mathbb{E} \left[g_t(\theta_{t-1}) \middle| \theta_{t-1} \right]^\top (\theta_{t-1} - \theta_*) \right] = \mathbb{E} \left[F'(\theta_{t-1})^\top (\theta_{t-1} - \theta_*) \right]. \end{aligned}$$

This leads to

$$\mathbb{E} \left[\|\theta_t - \theta_*\|_2^2 \right] \leq \mathbb{E} \left[\|\theta_{t-1} - \theta_*\|_2^2 \right] - 2\gamma_t \mathbb{E} \left[F'(\theta_{t-1})^\top (\theta_{t-1} - \theta_*) \right] + \gamma_t^2 B^2.$$

Thus, combining with the convexity inequality $F(\theta_{t-1}) - F(\theta_*) \leq F'(\theta_{t-1})^\top (\theta_{t-1} - \theta_*)$ from Eq. (5.4), we get

$$\gamma_t \mathbb{E} [F(\theta_{t-1}) - F(\theta_*)] \leq \frac{1}{2} \left(\mathbb{E} [\|\theta_{t-1} - \theta_*\|_2^2] - \mathbb{E} [\|\theta_t - \theta_*\|_2^2] \right) + \frac{1}{2} \gamma_t^2 B^2. \quad (5.7)$$

Except for the expectations, this is the same bound as Eq. (5.6) so we can conclude as in the proof of Theorem 5.3, *mutatis mutandis*. We state our bound in terms of the average iterates because the cost of finding the best iterate could be high in comparison to that of evaluating a stochastic gradient. ■

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): this 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.
- The result that we obtain, when applied to single pass SGD, is a generalization bound, that is, after the n iterations, we have an excess risk proportion 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 non-stochastic gradient descent. It turns out that the estimation error due to having n observations is

exactly the same as the generalization bound obtained by SGD (see Section 4.5.4 in Chapter 4), but we need to add on top the optimization error proportional to $1/\sqrt{t}$ (with the same constants). The bounds match if $t = n$, that is, we run n iterations of gradient descent on the empirical risk. This leads to a running time complexity of $O(tnd) = O(n^2d)$ instead of $O(nd)$ using SGD, hence the strong gains in using SGD.

- The bound in $O(BD/\sqrt{t})$ is optimal for this class of problem. That is, as shown for example in [2], among all algorithms that can query stochastic gradients, having a better convergence rate (up to some constants) is impossible.
- As opposed to the deterministic case, the use of smoothness does not lead to significantly better results.

SGD or gradient descent on the empirical risk? As seen above, the number of iterations to reach a given precision will be larger for stochastic gradient descent, but with a complexity which is typically n times faster. Thus, for high precision, that is low values of $F(\theta) - F(\eta_*)$ (which is not needed for machine learning), the number of iterations of SGD may become prohibitively large, and deterministic full gradient descent could be preferred. However, for low precision and large n , SGD is the method of choice (see also recent improvements in Section 5.4.2).

5.4.1 Strongly convex problems (♦)

We consider the regularized problem $G(\theta) = F(\theta) + \frac{\mu}{2}\|\theta\|_2^2$, with the same assumption as above, and started at $\theta_0 = 0$. The SGD iteration is then:

$$\theta_t = \theta_{t-1} - \gamma_t [g_t(\theta_{t-1}) + \mu\theta_{t-1}]. \quad (5.8)$$

We then have an improved convergence rate in $O(1/t)$ with a different decay for the step-size.

Theorem 5.5 (Convergence of SGD for strongly-convex problems) *Assume that F is convex, B -Lipschitz and that $G + \frac{\mu}{2}\|\cdot\|_2^2$ admits a (necessary unique) minimizer θ_* . Assume that the stochastic gradient g satisfies (H1-2). Then, choosing $\gamma_t = 1/(\mu t)$, the iterates $(\theta_t)_{t \geq 0}$ of SGD from Eq. (5.8) satisfy*

$$\mathbb{E}[F(\bar{\theta}_t) - F(\theta_*)] \leq \frac{2B^2(1 + \log t)}{\mu t},$$

where $\bar{\theta}_t = \frac{1}{t} \sum_{s=1}^t \theta_{s-1}$.

Proof The beginning of the proof is essentially the same as for convex problems, leading to (with the new terms in blue):

$$\begin{aligned} \mathbb{E}[\|\theta_t - \theta_*\|_2^2] &= \mathbb{E}[\|\theta_{t-1} - \gamma_t(g_t(\theta_{t-1}) + \mu\theta_{t-1}) - \theta_*\|_2^2] \\ &= \mathbb{E}[\|\theta_{t-1} - \theta_*\|_2^2] - 2\gamma_t \mathbb{E}[(g_t(\theta_{t-1}) + \mu\theta_{t-1})^\top (\theta_{t-1} - \theta_*)] + \gamma_t^2 \mathbb{E}[\|g_t(\theta_{t-1}) + \mu\theta_{t-1}\|_2^2]. \end{aligned}$$

From the iterations, we see that $\theta_t = (1 - \gamma_t\mu)\theta_{t-1} + \gamma_t\mu[-\frac{1}{\mu}g_t(\theta_{t-1})]$ is a convex combination of gradients divided by $-\mu$, and thus $\|g_t(\theta_{t-1}) + \mu\theta_{t-1}\|_2^2$ is always less than $4B^2$. Thus

$$\mathbb{E}[\|\theta_t - \theta_*\|_2^2] \leq \mathbb{E}[\|\theta_{t-1} - \theta_*\|_2^2] - 2\gamma_t \mathbb{E}[G'(\theta_{t-1})^\top (\theta_{t-1} - \theta_*)] + 4\gamma_t^2 B^2.$$

Therefore, combining with the strong convexity inequality $G(\theta_{t-1}) - G(\eta_*) + \frac{\mu}{2} \|\theta_{t-1} - \theta_*\|_2^2 \leq G'(\theta_{t-1})^\top (\theta_{t-1} - \theta_*)$ it follows

$$\gamma_t \mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] \leq \frac{1}{2} \left((1 - \gamma_t \mu) \mathbb{E} \|\theta_{t-1} - \theta_*\|^2 - \mathbb{E} \|\theta_t - \theta_*\|^2 \right) + 2\gamma_t^2 B^2,$$

and thus, now using the specific step-size choice:

$$\begin{aligned} \mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] &\leq \frac{1}{2} \left((\gamma_t^{-1} - \mu) \mathbb{E} \|\theta_{t-1} - \theta_*\|^2 - \gamma_t^{-1} \mathbb{E} \|\theta_t - \theta_*\|^2 \right) + 2\gamma_t B^2, \\ &= \frac{1}{2} \left(\mu(t-1) \mathbb{E} \|\theta_{t-1} - \theta_*\|^2 - \mu t \mathbb{E} \|\theta_t - \theta_*\|^2 \right) + \frac{2B^2}{\mu t}. \end{aligned}$$

Thus, we get a telescoping sum: summing between all indices between 1 and t , and using the bound $\sum_{s=1}^t \frac{1}{s} \leq 1 + \log t$, we get the desired result. \blacksquare

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 [14].
- The bound in $O(B^2/\mu t)$ is optimal for this class of problems. That is, as shown for example in [2], among all algorithms that can query stochastic gradients, having a better convergence rate (up to some constants) is impossible.
- We note that for the same regularized problem, we could use a step size proportion 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.

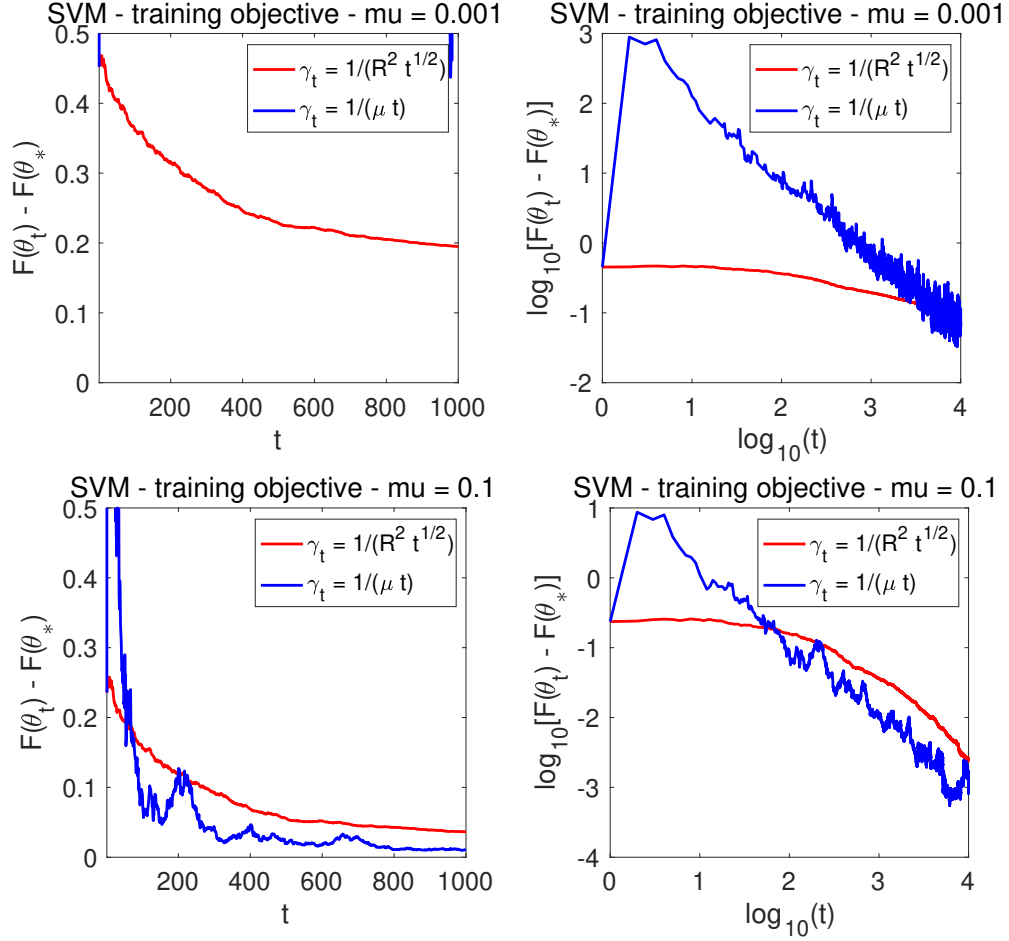
Note also the loss of adaptivity: the step-size now depends on the difficulty of the problem (this was not the case for deterministic gradient descent). See experiments below for illustrations.



Check homogeneity of the constants.

Exercise 5.6 *With the same assumptions as Theorem 5.5, show that with the step-size $\gamma_t = \frac{2}{\mu(t+1)}$, and with $\bar{\theta}_t = \frac{2}{t(t+1)} \sum_{s=1}^t \theta_{s-1}$, we have: $\mathbb{E}[F(\bar{\theta}_t) - F(\theta_*)] \leq \frac{2B^2}{\mu(t+1)}$.*

Experiments. We consider a simple binary classification problem with linear predictors and features with ℓ_2 -norm bounded by R . We consider the hinge loss with a square ℓ_2 -regularizer $\frac{\mu}{2} \|\cdot\|_2^2$. 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 $\gamma_t = 1/(\mu t)$. We see that for large enough μ , the strongly-convex step-size is better. This is not the case for small μ .



The experiments above highlight the danger of a step-size equal to $1/(\mu t)$. In practice, it is often preferable to use $\gamma_t = \frac{1}{B^2\sqrt{t} + \mu t}$.

5.4.2 Variance reduction (♦)

We consider a finite sum $F(\theta) = \frac{1}{n} \sum_{i=1}^n f_i(\theta)$, where each f_i is R^2 -smooth (for example logistic regression with features bounded by R in ℓ_2 -norm), and which is such that F is μ -strongly convex.

Using SGD, the convergence rate can be shown to be $O(\kappa/t)$, with iterations of complexity $O(d)$, while for GD, the convergence rates is $O(\exp(-t/\kappa))$, but each iteration has complexity $O(nd)$. We now present a result allowing to get exponential convergence with an iteration cost which is $O(d)$.

The idea is to use a form of *variance reduction*, which is made possible by keeping in memory past gradients. We denote by $z_i^{(t)} \in \mathbb{R}^d$ the version of gradient i stored at time t .

The SAGA algorithm [40], which combines the earlier algorithms SAG [94] and SVRG [57, 109], works as follows: at every iteration, an index $i(t)$ is selected uniformly at random in $\{1, \dots, n\}$, and we perform the iteration

$$\theta_t = \theta_{t-1} - \gamma \left[f'_{i(t)}(\theta_{t-1}) + \frac{1}{n} \sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)} \right],$$

with $z_{i(t)}^{(t)} = f'_{i(t)}(\theta_{t-1})$ and all others $z_i^{(t)}$ left unchanged (i.e., the same as $z_i^{(t-1)}$).

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 $f'_i(\eta_*)$, and then the update tends to have zero variance, and thus a constant step-size allows to obtain convergence. The key is then to show *simultaneously* that θ_t converges to η_* and that all $z_i^{(t)}$ converge to $f'_i(\eta_*)$, all at the same speed.

Theorem 5.6 (Convergence of SAGA) *If initializing with $z_i^{(0)} = f'_i(\theta_0)$, we have*

$$\mathbb{E}[\|\theta_t - \eta_*\|_2^2] \leq \left(1 - \min\left\{\frac{1}{3n}, \frac{3\mu}{4R^2}\right\}\right)^t \left(1 + \frac{n}{4}\right) \|\theta_0 - \eta_*\|_2^2.$$

Proof (◆◆) The proof consists in finding a Lyapunov function that decays along iterations.

Step 1. We first try our “usual” Lyapunov function, making the differences $\|z_i^{(t)} - f'_i(\theta_*)\|_2^2$ appear, with the update $\theta_t = \theta_{t-1} - \gamma \square_t$, with $\square_t = [f'_{i(t)}(\theta_{t-1}) + \frac{1}{n} \sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}]$,

$$\begin{aligned} \|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top \square_t + \gamma^2 \|\square_t\|_2^2 \text{ by expanding the square,} \\ \mathbb{E}_{i(t)} \|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + \gamma^2 \mathbb{E}_{i(t)} \|f'_{i(t)}(\theta_{t-1}) + \frac{1}{n} \sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\|_2^2 \\ &\quad \text{using the unbiasedness of the stochastic gradient,} \\ &\leq \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + 2\gamma^2 \mathbb{E}_{i(t)} \|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\|_2^2 \\ &\quad + 2\gamma^2 \mathbb{E}_{i(t)} \|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)} + \frac{1}{n} \sum_{i=1}^n z_i^{(t-1)}\|_2^2 \text{ using } \|a + b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2. \end{aligned}$$

In order to bound $\mathbb{E}_{i(t)} \|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\|_2^2$, we use co-coercivity of all functions f_i , to get:

$$\begin{aligned} \mathbb{E}_{i(t)} \|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\|_2^2 &= \frac{1}{n} \sum_{i=1}^n \|f'_i(\theta_{t-1}) - f'_i(\eta_*)\|_2^2 \leq \frac{1}{n} \sum_{i=1}^n R^2 [f'_i(\theta_{t-1}) - f'_i(\eta_*)]^\top (\theta_{t-1} - \eta_*) \\ &\leq R^2 F'(\theta_{t-1})^\top (\theta_{t-1} - \eta_*). \end{aligned} \tag{5.9}$$

In order to bound $\mathbb{E}_{i(t)} \|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)} + \frac{1}{n} \sum_{i=1}^n z_i^{(t-1)}\|_2^2$, we can simply use the identity $\mathbb{E}_{i(t)} \|Z - \mathbb{E}_{i(t)} Z\|_2^2 \leq \mathbb{E}_{i(t)} \|Z\|_2^2$. We thus get

$$\begin{aligned} \mathbb{E}_{i(t)} \|\theta_t - \eta_*\|_2^2 &\leq \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + 2\gamma^2 R^2 (\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \\ &\quad + 2\gamma^2 \frac{1}{n} \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2, \\ &\leq \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(1 - \gamma R^2)(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + 2\frac{\gamma^2}{n} \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2. \end{aligned}$$

Step 2. We see the term $\sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2$ appearing, so we try to study how it varies across iterations. We have:

$$\sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t)}\|_2^2 = \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2 + \|f'_{i(t)}(\eta_*) - f'_{i(t)}(\theta_{t-1})\|_2^2 - \|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)}\|_2^2$$

Taking expectations with respect to $i(t)$, we get

$$\begin{aligned} \mathbb{E}_{i(t)} \left[\sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t)}\|_2^2 \right] &= \left(1 - \frac{1}{n}\right) \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2 + \frac{1}{n} \sum_{i=1}^n \|f'_i(\eta_*) - f'_i(\theta_{t-1})\|_2^2 \\ &\leq \left(1 - \frac{1}{n}\right) \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2 + R^2(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}), \end{aligned}$$

where we use the bound in Eq. (5.9). Thus, for a positive number Δ to be chosen later,

$$\begin{aligned} \mathbb{E}_{i(t)} \left[\|\theta_t - \eta_*\|_2^2 + \Delta \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t)}\|_2^2 \right] &\leq \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(1 - \gamma R^2 - \frac{R^2 \Delta}{2\gamma})(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) \\ &\quad + \left[2\frac{\gamma^2}{n\Delta} + (1 - 1/n)\right] \Delta \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2. \end{aligned}$$

With $\Delta = 3\gamma^2$ and $\gamma = \frac{1}{4R^2}$, we get:

$$\mathbb{E}_{i(t)} \left[\|\theta_t - \eta_*\|_2^2 + \Delta \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t)}\|_2^2 \right] \leq \left(1 - \min\left\{\frac{1}{3n}, \frac{3\mu}{4R^2}\right\}\right) \left[\|\theta_{t-1} - \eta_*\|_2^2 + \Delta \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(t-1)}\|_2^2 \right].$$

Thus

$$\mathbb{E}[\|\theta_t - \eta_*\|_2^2] \leq \left(1 - \min\left\{\frac{1}{3n}, \frac{3\mu}{4R^2}\right\}\right)^t \left[\|\theta_0 - \eta_*\|_2^2 + \frac{3}{16R^4} \sum_{i=1}^n \|f'_i(\eta_*) - z_i^{(0)}\|_2^2 \right].$$

If initializing with $z_i^{(0)} = f'_i(\theta_0)$, we get the desired bound by using the Lipschitz-continuity of each f'_i . ■

We can make the following observations:

- The contraction rate after one iteration is $\left(1 - \min\left\{\frac{1}{3n}, \frac{3\mu}{4R^2}\right\}\right) \leq \exp\left(\min\left\{-\frac{1}{3n}, \frac{3\mu}{4R^2}\right\}\right)$. Thus, after an “effective pass” over the data, that is, n iterations, the contracting rate is $\exp\left(\min\left\{-\frac{1}{3}, \frac{3\mu n}{4R^2}\right\}\right)$. It is only an effective pass, because after we sample n indices with replacement, we will not see all functions.

In order to have a contracting effect of ε , that is, having $\|\theta_t - \eta_*\|_2^2 \leq \varepsilon \|\theta_0 - \eta_*\|_2^2$, we need to have $\exp\left(t \min\left\{-\frac{1}{3n}, \frac{3\mu}{4R^2}\right\}\right) n \leq \varepsilon$, which is equivalent to $t \geq \max\left\{3n, \frac{4R^2}{3\mu}\right\} \log \frac{n}{\varepsilon}$. It just suffices to have $t \geq \left(3n + \frac{4R^2}{3\mu}\right) \log \frac{n}{\varepsilon}$, and thus the running time complexity is equal to d times the minimal number, that is

$$d \left(3n + \frac{4R^2}{3\mu}\right) \log \frac{n}{\varepsilon}.$$

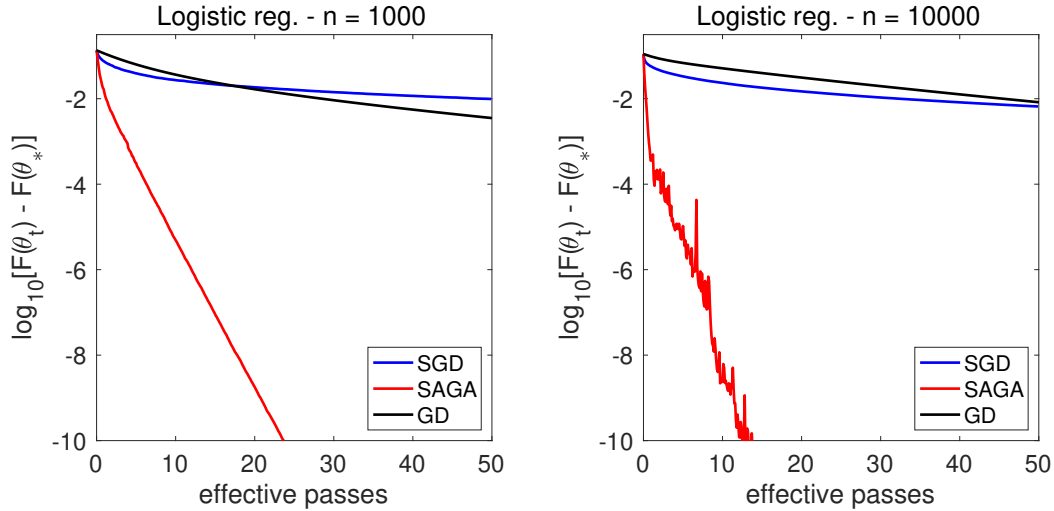
This to be contrasted with batch gradient descent with step-size $\gamma = 1/R^2$ (which is the simplest step-size that can be computed easily), whose complexity is

$$dn \frac{R^2}{\mu} \log \frac{n}{\varepsilon}.$$

We replace the product of n and condition number $\frac{R^2}{\mu}$ 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, acceleration. It is also worth mentioning that the need to store past gradients can be alleviated (see [49] for more details).
- Note that these fast algorithms allow to get very small optimization errors, and that the best testing risks will typically be obtained after a few (10 to 100) passes.

Experiments. We consider ℓ_2 -regularized logistic regression and we compare GD, SGD and SAGA, all with their corresponding step-sizes coming from the theoretical analysis, with two values of n (left: small, right: large). We see that for early iterations, SGD dominates GD, while for larger numbers of iterations, GD is faster. This last effect is not seen for large numbers of observations (right). In the two cases, SAGA gets to machine precision after 50 effective passes over the data.



5.5 Conclusion

We can now provide a summary of convergence rates below, with the main rates that we have seen in this chapter (and some that we have not seen). We separate between convex and strongly convex, and between smooth and non-smooth, as well as between deterministic and stochastic methods. Below, L is the smoothness constant, μ the strong convexity constant, B the Lipschitz constant and D the distance to optimum at initialization.

	convex	strongly convex
nonsmooth	deterministic: BD/\sqrt{t} stochastic: BD/\sqrt{t}	deterministic: $B^2/(t\mu)$ stochastic: $B^2/(t\mu)$
smooth	deterministic: LD^2/t^2 stochastic: LD^2/\sqrt{t} finite sum: n/t	deterministic: $\exp(-t\sqrt{\mu/L})$ stochastic: $L/(t\mu)$ finite sum: $\exp(-\min\{1/n, \mu/L\}t)$

Note that many important themes in optimization have been ignored, such as Frank-Wolfe methods (presented in Chapter 9), coordinate descent, duality. See [80, 31] for further details. See also Chapter 7 and Chapter 9 for optimization methods for kernel methods and neural networks.

Chapter 6

Local averaging methods

Chapter summary

- “Linear” estimators: These are estimators which are based on assigning weight functions to each observation so that each observation can vote for its label with the corresponding weight.
- Partitioning estimates: the input space is cut into non-overlapping cells and the predictor is piecewise-constant.
- Nadaraya-Watson estimators (a.k.a. kernel regression): each observation assigns a weight proportional to its distance in input space.
- k -nearest-neighbors: each observation assigns an equal weight to its k nearest neighbors.
- Consistency: All of these methods can provably learn complex non-linear functions with a convergence rate of the form $O(n^{-2/(d+2)})$, where d is the underlying dimension, leading to the curse of dimensionality.

6.1 Introduction

Like in most of this textbook, we are being given a training set: observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of inputs/outputs, features/variables are assumed independent and identically distributed (i.i.d.) random variables with common distribution $dp(x, y)$. We consider a fixed (testing) distribution dp on $\mathcal{X} \times \mathcal{Y}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$; $\ell(y, z)$ is the loss of predicting z while the true label is y .

Our goal is to minimize the risk, or generalization performance of a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$:

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

⚠ Like in the rest of the book, we assume that the testing distribution is the same as the training distribution.

⚠ Be careful with the randomness or lack thereof of f : The estimator \hat{f} we will use depends on the training data and not on the testing data, and thus $\mathcal{R}(\hat{f})$ is random because of the dependence on the training data.

As seen in Chapter 2, the two classical cases are:

- Binary classification: $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$), and $\ell(y, z) = 1_{y \neq z}$ (“0-1” loss). Then $\mathcal{R}(f) = \mathbb{P}(f(x) \neq y)$.
- Regression: $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). Then $\mathcal{R}(f) = \mathbb{E}(y - f(x))^2$.

As seen in Chapter 2, minimizing the expected risk leads to an optimal “target function”, called the Bayes predictor $f^* \in \arg \min \mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))]$.

Proposition 6.1 (Bayes predictor) *The risk is minimized at a Bayes predictor $f^* : \mathcal{X} \rightarrow \mathcal{Y}$ satisfying for all $x \in \mathcal{X}$, $f^*(x) \in \arg \min_{z \in \mathcal{Y}} \mathbb{E}(\ell(y, z)|x)$. The Bayes risk \mathcal{R}^* is the risk of all Bayes predictors and is equal to*

$$\mathcal{R}^* = \mathbb{E}_{x \sim dp(x)} \inf_{z \in \mathcal{Y}} \mathbb{E}(\ell(y, z)|x) = \mathbb{E}_{x \sim dp(x)} \inf_{z \in \mathcal{Y}} \mathbb{E}_{y \sim dp(y|x)}(\ell(y, z)|x).$$

Note that (a) the Bayes predictor is not unique, but that all lead to the same Bayes risk, and (b) that the Bayes risk is usually non zero (unless the dependence between x and y is deterministic). The goal of supervised machine learning is thus to estimate f^* , knowing only the training data $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ and the loss ℓ , with the goal of minimizing the risk or the excess risk $\mathcal{R}(f) - \mathcal{R}^*$. We have the following special cases:

- For binary classification: $\mathcal{Y} = \{0, 1\}$ and $\ell(y, z) = 1_{y \neq z}$, the Bayes predictor is equal to $f^*(x) \in \arg \max_{i \in \{0, 1\}} \mathbb{P}(y = i|x)$. This extends naturally to multi-category classification with the Bayes predictor $f^*(x) \in \arg \max_{i \in \{1, \dots, k\}} \mathbb{P}(y = i|x)$.
- 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 $\mathcal{R}(f) - \mathcal{R}^* = \int_{\mathcal{X}} (f(x) - f^*(x))^2 dp(x) = \|f - f^*\|_{L_2(dp(x))}^2$.

In Chapter 3 and Chapter 4, we explored methods based on empirical risk minimization (and we will as well in Chapter 7 and Chapter 9). In this chapter, we focus on local averaging methods.

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 $dp(y|x)$ of y given x , by some $d\hat{p}(y|x)$. We then replace $f^*(x) \in \arg \min_{z \in \mathcal{Y}} \int_{\mathcal{Y}} \ell(y, z) dp(y|x)$ by $\hat{f}(x) \in \arg \min_{z \in \mathcal{Y}} \int_{\mathcal{Y}} \ell(y, z) d\hat{p}(y|x)$. These are often called “plug-in” estimators.

In the usual cases, this leads to the following predictions:

- For classification with the 0-1 loss: $\hat{f}(x) \in \arg \max_{j \in \{1, \dots, k\}} \hat{\mathbb{P}}(y = j|x)$.
- For regression with the square loss: $\hat{f}(x) = \int_{\mathcal{Y}} y d\hat{p}(y|x)$.

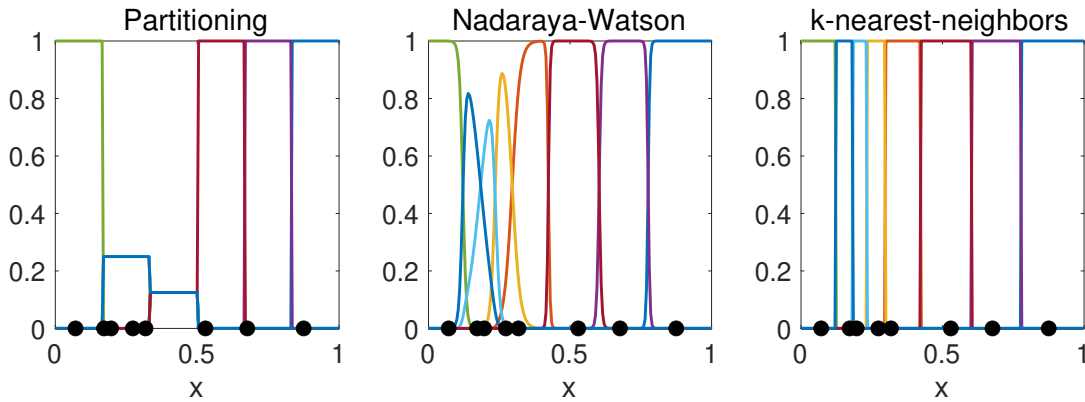


Figure 6.1: Weights of linear estimators in $d = 1$ dimension for the three types of local averaging estimators. The $n = 8$ weight functions $x \mapsto \hat{w}_i(x)$ are plotted with the observations in black.

6.2.1 Linear estimators

In this chapter we will consider “linear” estimators, where the conditional distribution is of the form

$$d\hat{p}(y|x) = \sum_{i=1}^n \hat{w}_i(x) \delta_{y_i}(y),$$

where δ_{y_i} is the Dirac probability distribution at y_i (putting a unit mass at y_i), and the weight functions $\hat{w}_i : \mathcal{X} \rightarrow \mathbb{R}$, $i = 1, \dots, n$, depend on the input data only (for simplicity) and satisfy (almost surely in x):

$$\forall x \in \mathcal{X}, \quad \forall i \in \{1, \dots, n\}, \quad \hat{w}_i(x) \geq 0, \quad \text{and} \quad \sum_{i=1}^n \hat{w}_i(x) = 1.$$

These conditions ensure that for all $x \in \mathcal{X}$, $d\hat{p}(y|x)$ is a probability distribution.

⚠ Some references allow for the weights not to sum to 1.

For our running examples, this leads to the following predictions:

- For classification: $\hat{f}(x) \in \arg \max_{j \in \{1, \dots, k\}} \sum_{i=1}^n \hat{w}_i(x) 1_{y_i=j}$, that is, each observation (x_i, y_i) votes for its label with weight $\hat{w}_i(x)$.
- For regression: $\mathcal{Y} = \mathbb{R}$: $\hat{f}(x) = \sum_{i=1}^n \hat{w}_i(x) y_i$. This is why the terminology “linear estimators” is sometimes used, since, as a function of the response vector in \mathbb{R}^n , the estimator is linear (note that this is the case as well for kernel ridge regression in Chapter 7).

Construction of weight functions. In most cases, for any i , the weight function $\hat{w}_i(x)$ is close to 1 for training points x_i which are close to x . We now show three classical ways of building them: (1) partition estimators, (2) Nearest-neighbors, and (3) Nadaraya-Watson estimator (a.k.a. kernel regression). See examples in Figure 6.1.

6.2.2 Partition estimators

If $\mathcal{X} = \bigcup_{j \in J} A_j$ is a partition (such that for all $j, j' \in J$, $A_j \cap A_{j'} = \emptyset$) of \mathcal{X} with a countable index set J (which we will assume finite for simplicity), then we can consider for any $x \in \mathcal{X}$ the corresponding element $A(x)$ of the partition (that is, $A(x)$ is the unique A_j , $j \in J$, such that $x \in A_j$), and define

$$\hat{w}_i(x) = \frac{1_{x_i \in A(x)}}{\sum_{i'=1}^n 1_{x_{i'} \in A(x)}}, \quad (6.1)$$

with the convention that if no training data point lies in $A(x)$, then $\hat{w}_i(x)$ is equal to $1/n$ for each $i \in \{1, \dots, n\}$. This implies that each w_i is piecewise constant with respect to the partition, that is, for any non-empty cell A_j (that is for which at least one observation falls in A_j), for any $x \in A_j$, the vectors $(w_i(x))_{i \in \{1, \dots, n\}}$ has weights equal to $1/n_{A_j}$ for $i \in A_j$ (where n_{A_j} is the number of training points in the set A_j), and 0 otherwise.

Equivalence with least-squares regression. When applied to regression where the estimator is $\hat{f}(x) = \sum_{i=1}^n \hat{w}_i(x) y_i$, then using a partition estimators can be seen as a least-squares estimator with feature vector $\varphi(x) = (1_{x \in A_j})_{j \in J} \in \mathbb{R}^J$. Indeed, from training data $(x_1, y_1), \dots, (x_n, y_n)$, as shown in Chapter 3, we need to find the weight vector $\hat{\theta}$ through the normal equations

$$\sum_{i=1}^n \varphi(x_i) \varphi(x_i)^\top \theta = \sum_{i=1}^n y_i \varphi(x_i).$$

It turns out that the matrix $n\hat{\Sigma} = \sum_{i=1}^n \varphi(x_i) \varphi(x_i)^\top$ is diagonal where for each $j \in J$, $n\hat{\Sigma}_{jj}$ is equal to n_{A_j} the number of data points lying in cell A_j . This implies that for a non-empty cell A_j , θ_j is the average of all y_i 's for x_i 's lying in A_j . Thus, for all $x \in A_j$, the prediction is exactly θ_j , like obtained from weights in Eq. (6.1). For empty cells, then θ_j is not determined. Among the many OLS estimators, we select the one for which the variance of the vector θ is smallest, that is $\sum_{j \in J} (\theta_j - \frac{1}{|J|} \sum_{j' \in J} \theta_{j'})^2$ is smallest. A short calculation shows that this exactly leads to $\theta_j = \frac{1}{n} \sum_{i=1}^n y_i$ for these empty cells, which corresponds to our chosen convention.

⚠ Other conventions exist (such as all zero weights when no data point lies in $A(x)$).

This equivalence with least-squares estimation with a diagonal (empirical or not) non-centered covariance matrix makes it attractive for theoretical purposes.

Choice of partitions. There are two standard applications of partition estimators:

- Fixed partitions: for example, when $\mathcal{X} = [0, 1]^d$, then we consider cubes of length h , with $|J| = h^{-d}$ (see example below in $d = 2$ dimension with $|J| = 25$). Note here that the computation time for each $x \in \mathcal{X}$ is not necessarily proportional to $|J|$, but to n (by simply considering the bins where the data lie). This estimator is sometimes called a “regressogram”. We need then to choose the bandwidth h (see analysis in Section 6.3.1). See Figure 6.2 for an illustration in one-dimension.

A_1	A_2	A_3	A_4	A_5
A_6	A_7	A_8	A_9	A_{10}
A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}

- Decision trees: for data in a hypercube, we can recursively partition it by selecting a variable to split leading to a maximum reduction in errors when defining the partitioning estimate (see more details in https://en.wikipedia.org/wiki/Decision_tree_learning). Note here that the partition depends on the labels (so the analysis below does not apply, unless the partitioning is learned on a different data than the one used for the estimation).

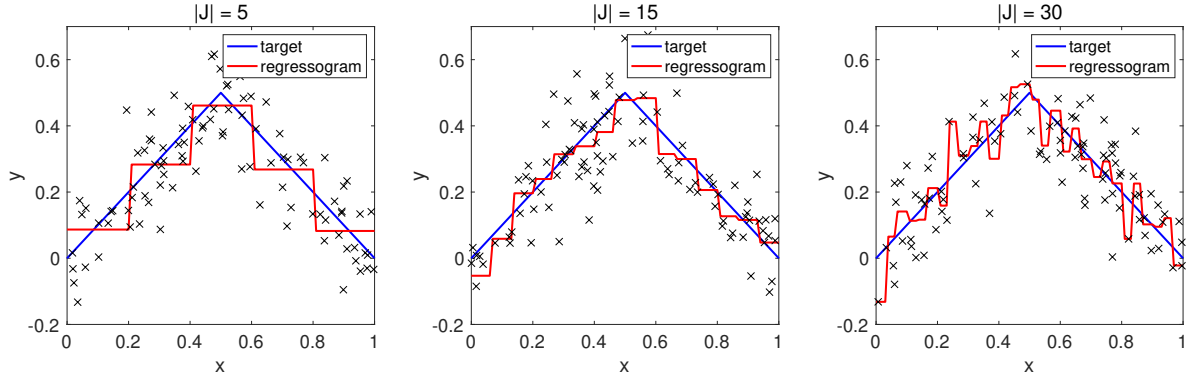


Figure 6.2: Regressograms in $d = 1$ dimension, with three values of $|J|$ (the number of sets in the partition). We can observe both underfitting ($|J|$ too small), or overfitting ($|J|$ too large). Note that the target function f^* is piecewise affine, and that on the affine parts, the estimator is far from linear, that is, the estimator cannot take advantage of extra-regularity (see Section 6.5 for more details).

6.2.3 Nearest-neighbors

Given an integer $k \geq 1$, and a distance d on \mathcal{X} , for any $x \in \mathcal{X}$, we can order the n observations so that

$$d(x_{i_1(x)}, x) \leq d(x_{i_2(x)}, x) \leq \dots \leq d(x_{i_n(x)}, x),$$

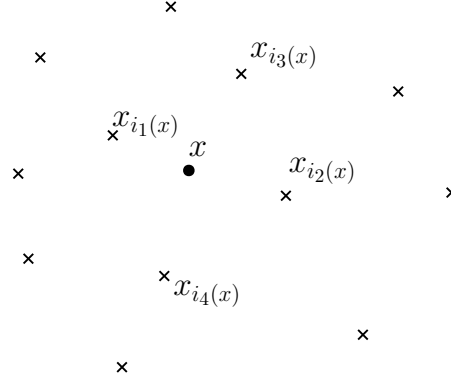
where $\{i_1(x), \dots, i_n(x)\} = \{1, \dots, n\}$, and ties are broken randomly¹ (that is, by sampling priorities randomly for each i once for all $x \in \mathcal{X}$). We then define

$$\hat{w}_i(x) = 1/k \text{ if } i \in \{i_1(x), \dots, i_k(x)\}, \text{ and } 0 \text{ otherwise.}$$

Given a new input $x \in \mathbb{R}^d$, the nearest neighbor predictor looks at the k nearest points x_i in the data set $\{(x_1, y_1), \dots, (x_n, y_n)\}$ and predicts a majority vote among them (for classification) or simply the

¹Other conventions share the weights among all ties.

averaged response (for regression). The number of nearest neighbors is the hyperparameter which needs to be estimated (typically by cross-validation), see Section 6.3.2 for an analysis. See a one-dimensional example in Figure 6.3.



Algorithms. Given a test point $x \in \mathcal{X}$, the naive algorithm looks at all training data points for computing the predicted response, thus the complexity is $O(nd)$ per test point in \mathbb{R}^d . When n is large, this is costly in time and memory. There exists indexing techniques for (potentially approximate) nearest-neighbor search, such as “k-d-trees”, with typically a logarithmic complexity in n (but with some additional compiling time) (see https://en.wikipedia.org/wiki/K-d_tree).

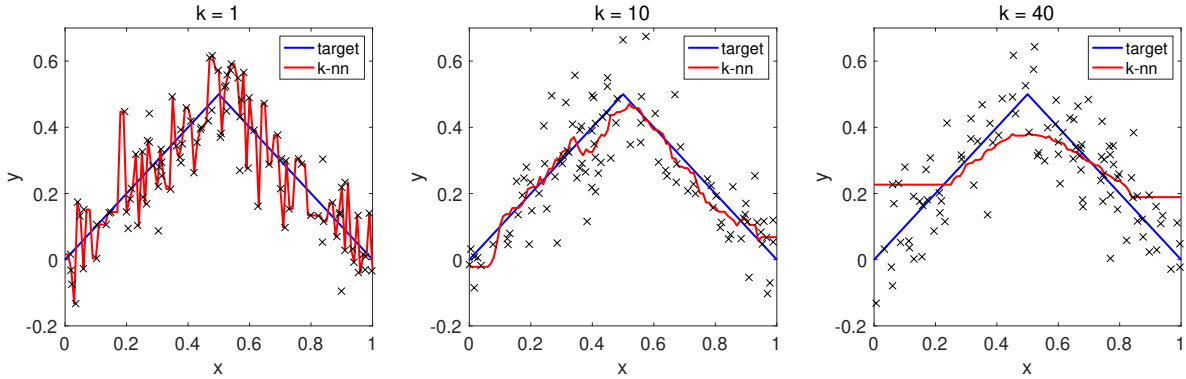


Figure 6.3: k -nearest neighbor regression in $d = 1$ dimension, with three values of k (the number of neighbors). We can observe both underfitting (k too large), or overfitting (k too small).

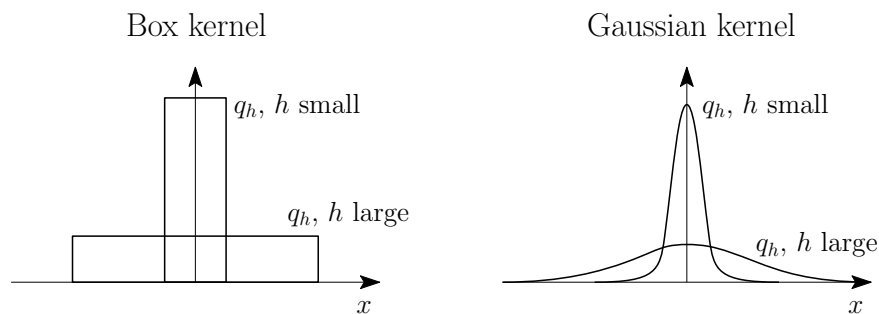
6.2.4 Nadaraya-Watson estimator a.k.a. kernel regression (♦)

Given a “kernel” function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$, which is pointwise non-negative, we define

$$\hat{w}_i(x) = \frac{k(x, x_i)}{\sum_{i'=1}^n k(x, x_{i'})},$$

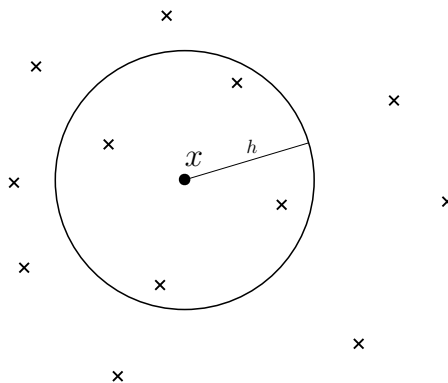
with the convention that if $k(x, x_i) = 0$ for all $i \in \{1, \dots, n\}$, then $\hat{w}_i(x)$ is equal to $1/n$ for each i . In most cases where $\mathcal{X} \subset \mathbb{R}^d$, we take $k(x, x') = h^{-d}q(\frac{1}{h}(x - x'))$ for a certain function $q : \mathbb{R}^d \rightarrow \mathbb{R}_+$

that has large values around 0, and $h > 0$ a “bandwidth” parameter to be selected (see analysis in Section 6.3.3). If we assume that q is integrable with integral equal to one, then $k(\cdot, x')$ is a probability density with mass around x' , which gets more concentrated as h goes to zero. See illustration below for the two typical windows.



Typical examples are:

- Box kernel: $q(x) = 1_{\|x\|_2 \leq 1}$. See below for an illustration in $d = 2$ dimensions.



- Gaussian kernel $q(x) = e^{-\|x\|^2/2}$, where we use the fact it is non-negative *pointwise* (as opposed to positive-definiteness in Chapter 7, see <https://francisbach.com/cursed-kernels/>). See a one-dimensional experiment in Figure 6.4.

In terms of algorithms, with a naive algorithm, for every test point, all the input data have to be considered, that is, a complexity proportional to n . The same techniques used for efficient k -nearest-neighbor search (e.g., k -d-trees) can be applied here as well.

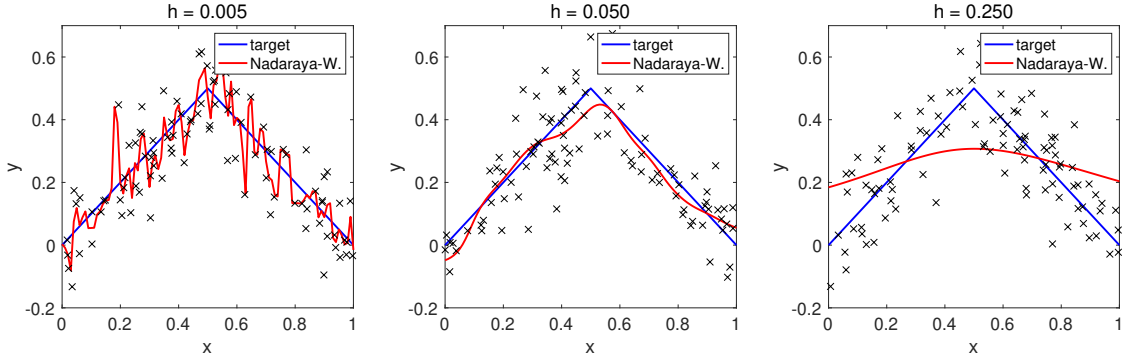


Figure 6.4: Nadaraya-Watson regression in $d = 1$ dimension, with three values of h (the bandwidth), for the Gaussian kernel. We can observe both underfitting (h too large), or overfitting (h too small).

6.3 Generic “simplest” consistency analysis

We consider for simplicity the regression case. For classification, calibration techniques like used in Chapter 4 can be used (with then a square root calibration function on top of the least-squares excess risk), but better rates can be obtained directly (see, e.g., [34, 23, 6, 33]).

We make the following generic assumptions:

- (H1) Bounded noise: There exists $\sigma \geq 0$ such that $|y - \mathbb{E}(y|x)|^2 \leq \sigma^2$ almost surely.
- (H2) Regular target function: The target function $f^*(x) = \mathbb{E}(y|x)$ is B -Lipschitz-continuous with respect to a distance d . For weaker assumptions, see Section 6.4.

We have, with the target function $f^*(x) = \mathbb{E}(y|x)$, at a test point $x \in \mathcal{X}$ (and using that the weights $w_i(x)$ sum to one):

$$\begin{aligned}
 \hat{f}(x) - f^*(x) &= \sum_{i=1}^n y_i \hat{w}_i(x) - \mathbb{E}(y|x) \\
 &= \sum_{i=1}^n \hat{w}_i(x) [y_i - \mathbb{E}(y_i|x_i)] + \sum_{i=1}^n \hat{w}_i(x) [\mathbb{E}(y_i|x_i) - \mathbb{E}(y|x)] \\
 &= \sum_{i=1}^n \hat{w}_i(x) [y_i - \mathbb{E}(y_i|x_i)] + \sum_{i=1}^n \hat{w}_i(x) [f^*(x_i) - f^*(x)].
 \end{aligned}$$

Given x_1, \dots, x_n (and because we have assumed the weight functions do not depend on the labels), the left term has zero expectation, while the right term is deterministic. We thus have, using the independence of all (x_i, y_i) , $i = 1, \dots, n$, and for x fixed:

$$\begin{aligned}
 \mathbb{E}[(\hat{f}(x) - f^*(x))^2 | x_1, \dots, x_n] &= (\mathbb{E}[\hat{f}(x) | x_1, \dots, x_n] - f^*(x))^2 + \text{var}[\hat{f}(x) | x_1, \dots, x_n] \\
 &= \left[\sum_{i=1}^n \hat{w}_i(x) [f^*(x_i) - f^*(x)] \right]^2 + \sum_{i=1}^n \hat{w}_i(x)^2 \mathbb{E}[(y_i - \mathbb{E}(y_i|x_i))^2 | x_i] \\
 &= \text{bias} + \text{variance},
 \end{aligned}$$

with a “bias” term which is zero if f^* is constant, and a “variance” term which is zero, when y is a deterministic function of x . We can further bound as:

$$\begin{aligned} \mathbb{E}[(\hat{f}(x) - f^*(x))^2 | x_1, \dots, x_n] &\leq \left[\sum_{i=1}^n \hat{w}_i(x) |f^*(x_i) - f^*(x)| \right]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H1),} \\ &\leq \left[\sum_{i=1}^n \hat{w}_i(x) B d(x_i, x) \right]^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using (H2),} \\ &\leq B^2 \sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 + \sigma^2 \sum_{i=1}^n \hat{w}_i(x)^2 \text{ using Jensen's inequality.} \end{aligned} \quad (6.2)$$

We then have for the expected excess risk:

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leq B^2 \int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \right] dp(x) + \sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x). \quad (6.3)$$

⚠ 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 confusions.

This upper bound can be divided into:

- A variance term $\sigma^2 \sum_{i=1}^n \int_{\mathcal{X}} \mathbb{E}[\hat{w}_i(x)^2] dp(x)$, that depends on the noise on top of the optimal predictions. Since the weights sum to one, we can write $\sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)^2] = \sum_{i=1}^n \mathbb{E}[(\hat{w}_i(x) - 1/n)^2] + 2/n - 1/n^2$, that is, up to vanishing constant, the variance term measures the deviation to uniform weights.
- A bias term $B^2 \int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \right] dp(x)$, which depends on the regularity of the target function.

This leads to two conditions: both variance and bias have to go to zero when n grows, and this corresponds to two simple quantities on the weights. For the variance, the worst case scenario is that $\hat{w}_i(x)^2 \approx \hat{w}_i(x)$, that is, weights are putting all the mass into a single label (usually different for different testing point), thus leading to overfitting. For the bias, the worst case scenario is that weights are uniform (leading to underfitting).

In the following, we will specialize it for \mathcal{X} a subset of \mathbb{R}^d , with a density $dp(x)$ with some minor regularity properties (all will have compact support, that is, \mathcal{X} compact), where we show that a proper setting of the hyperparameters leads to “good” predictions. This will be done for all three cases of local averaging methods.

We look at universal consistency in Section 6.4.

6.3.1 Fixed partition

For the partitioning estimate defined in Section 6.2.2, we can prove the following convergence rate.

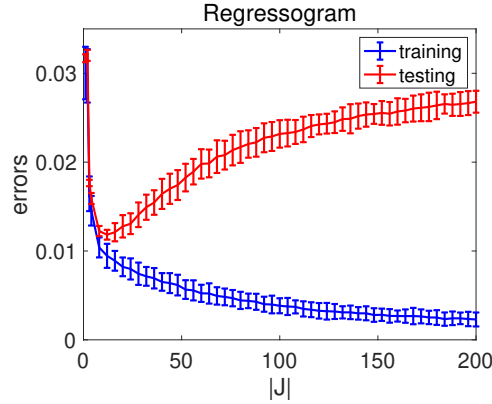
Proposition 6.2 (Convergence rate for partition estimates) *Assume bounded noise (H1) and a Lipschitz-continuous target function (H2), and a partition $\mathcal{X} = \bigcup_{j \in J} A_j$; then for the partitioning estimate \hat{f} , we have:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leq (8\sigma^2 + \frac{B^2}{2} \text{diam}(\mathcal{X})^2) \frac{|J|}{n} + B^2 \max_{j \in J} \text{diam}(A_j)^2. \quad (6.4)$$

Optimal trade-off between bias and variance. Before we look at the proof (which is based on Eq. (6.3)), we can look at the consequence of the bound in Eq. (6.4). We need to balance the terms (up to constants) $\max_{j \in J} \text{diam}(A_j)^2$ 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 $\frac{|J|}{n} \propto \frac{1}{nh^d}$ and $\max_{j \in J} \text{diam}(A_j)^2 \propto h^2$, which, with $h \propto n^{-1/(2+d)}$ to make them equal, leads to a rate proportional to $n^{-2/(2+d)}$. As shown in [52], this rate is optimal for estimation of Lipschitz-continuous functions.

While optimal, this is a very slow rate, and a typical example of the curse of dimensionality. For this rate to be small, n has to be exponentially large in dimension. This is unavoidable with so little regularity (only bounded first-order derivatives). In Chapter 7 (and also in Section 6.5), we show how to leverage smoothness to get significantly improved bounds. In Chapter 8, we will leverage dependence on a small number of variables.

Experiments. For the problem shown in Section 6.2, below, we plot training and testing errors averaged over 32 replications (with error bars showing the standard deviations), where we clearly see the trade-off in the choice of $|J|$.



Proof of Proposition 6.2 (♦) We consider an element A_j of the partition with at least one observation in it (a non-empty cell). Then for $x \in A_j$, and i among the indices of the points lying in A_j , $\hat{w}_i(x) = 1/n_{A_j}$ where $n_{A_j} \in \{1, \dots, n\}$ is the number of data points lying in A_j .

Variance. From Eq. (6.3), the variance term is bounded from above by 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 $n \times \frac{1}{n^2} = 1/n$ for all $x \in A_j$. Thus, we get

$$\begin{aligned} \int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x)^2 \right] dp(x) &= \int_{\mathcal{X}} \mathbb{E} \left[\sum_{j \in J} 1_{x \in A_j} \mathbb{E} \left[\frac{1}{n_{A_j}} 1_{n_{A_j} > 0} + \frac{1}{n} 1_{n_{A_j} = 0} \right] \right] dp(x) \\ &= \sum_{j \in J} \mathbb{P}(A_j) \mathbb{E} \left[\frac{1}{n_{A_j}} 1_{n_{A_j} > 0} + \frac{1}{n} 1_{n_{A_j} = 0} \right]. \end{aligned}$$

Intuitively, by the law of large numbers, n_{A_j}/n tends to $\mathbb{P}(A_j)$, so the variance term is expected to be of the order $\sigma^2 \sum_{j \in J} \mathbb{P}(A_j) \frac{1}{n \mathbb{P}(A_j)} = \sigma^2 \frac{|J|}{n}$, which is to be expected as this is essentially equivalent to least-squares regression with features $(1_{x \in A_j})_{j \in J}$.

More formally, we have $\mathbb{P}(n_{A_j} = 0) = (1 - \mathbb{P}(A_j))^n$, and, using Bernstein’s inequality for the random variables $1_{x_i \in A_j}$, which have mean and variance upper bounded by $\mathbb{P}(A_j)$, we have: $\mathbb{P} \left(\frac{n_{A_j}}{n} \leq \mathbb{P}(A_j) - \frac{1}{2} \mathbb{P}(A_j) \right) \leq \exp \left(- \frac{n \mathbb{P}(A_j)^2 / 4}{2 \mathbb{P}(A_j) + 2(\mathbb{P}(A_j)/2)/3} \right) \leq \exp(-n \mathbb{P}(A_j)/10) \leq \frac{5}{n \mathbb{P}(A_j)}$, leading to a bound

$$\begin{aligned} \sum_{j \in J} \mathbb{P}(A_j) \mathbb{E} \left[\frac{1}{n_{A_j}} 1_{n_{A_j} > 0} + \frac{1}{n} 1_{n_{A_j} = 0} \right] &\leq \sum_{j \in J} \mathbb{P}(A_j) \mathbb{E} \left[\mathbb{P} \left(\frac{n_{A_j}}{n} \leq \mathbb{P}(A_j)/2 \right) + \frac{2}{n \mathbb{P}(A_j)} + \frac{1}{n} \mathbb{P}(n_{A_j} = 0) \right] \\ &\leq \sum_{j \in J} \mathbb{P}(A_j) \mathbb{E} \left[\frac{5}{n \mathbb{P}(A_j)} + \frac{2}{n \mathbb{P}(A_j)} + \frac{1}{n \mathbb{P}(A_j)} \right] \leq \frac{8|J|}{n}. \end{aligned}$$

Bias. We have, for $x \in A_j$ and a non-empty cell,

$$\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \leq \text{diam}(A_j)^2,$$

with $\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 = \frac{1}{n} \sum_{i=1}^n d(x, x_i)^2 \leq \text{diam}(\mathcal{X})^2$ for empty-cells. Thus, separating the cases $n_{A_j} = 0$ and $n_{A_j} > 0$:

$$\begin{aligned} \int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \right] dp(x) &\leq \sum_{j \in J} \mathbb{P}(A_j) \mathbb{E} \left[\text{diam}(A_j)^2 1_{n_{A_j} > 0} + 1_{n_{A_j} = 0} \text{diam}(\mathcal{X})^2 \right] \\ &\leq \sum_{j \in J} \mathbb{P}(A_j) \left[\text{diam}(A_j)^2 + (1 - \mathbb{P}(A_j))^n \text{diam}(\mathcal{X})^2 \right] \\ &= \sum_{j \in J} \mathbb{P}(A_j) \text{diam}(A_j)^2 + \sum_{j \in J} \mathbb{P}(A_j) (1 - \mathbb{P}(A_j))^n \times \text{diam}(\mathcal{X})^2 \\ &\leq \sum_{j \in J} \mathbb{P}(A_j) \text{diam}(A_j)^2 + \sum_{j \in J} \mathbb{P}(A_j) \frac{1}{2n \mathbb{P}(A_j)} \times \text{diam}(\mathcal{X})^2 \\ &= \sum_{j \in J} \mathbb{P}(A_j) \text{diam}(A_j)^2 + \frac{1}{2} \frac{|J|}{n} \times \text{diam}(\mathcal{X})^2, \end{aligned}$$

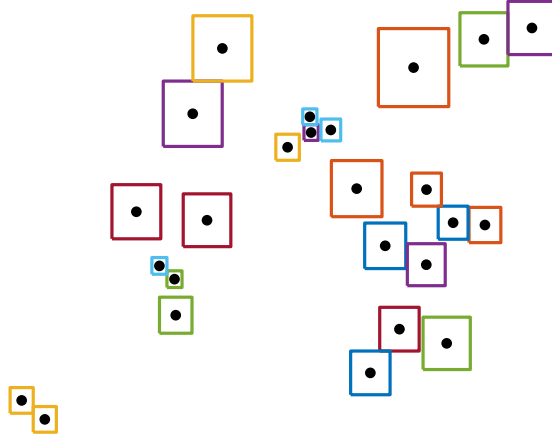
which leads to the desired term. ■

6.3.2 k -nearest neighbor

Here, we immediately have $\sum_{i=1}^n \hat{w}_i(x)^2 = \frac{1}{k}$, so the variance term will go down as soon as k tends to infinity. For the bias term, the needed term $\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2$ is equal to the average of the squared distances between x and its k -nearest neighbors within $\{x_1, \dots, x_n\}$, and this is less than the expected distance to the k -nearest neighbor, for which the two following lemmas (taken from [23, Theorem 2.4]) 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 $\mathcal{X} \subset \mathbb{R}^d$. Consider $n+1$ 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 $4 \frac{\text{diam}(\mathcal{X})^2}{n^{2/d}}$ for $d \geq 2$, and less than $\frac{2}{n} \text{diam}(\mathcal{X})^2$ for $d = 1$.*

Proof By symmetry we aim at computing $\frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{E}[\|x_i - x_{(i)}\|_\infty^2]$, where $x_{(i)}$ is a nearest neighbor of x_i among the other n points. Denoting by $R_i = \|x_i - x_{(i)}\|_\infty$, then the sets $B_i = \{x \in \mathbb{R}^d, \|x - x_i\|_\infty < \frac{R_i}{2}\}$ are disjoint.



Moreover, their union has diameter less than $\text{diam}(X) + \text{diam}(X) = 2\text{diam}(X)$. Thus by comparing volumes, we have: $\sum_{i=1}^{n+1} R_i^d \leq (2\text{diam}(X))^d$. Therefore, by Jensen's inequality, for $d \geq 2$,

$$\left(\frac{1}{n+1} \sum_{i=1}^{n+1} R_i^2 \right)^{d/2} \leq \frac{1}{n+1} \sum_{i=1}^{n+1} (R_i)^d \leq \frac{2^d \text{diam}(\mathcal{X})^d}{n+1},$$

leading to the desired result. For $d = 1$, we simply have $\left(\frac{1}{n+1} \sum_{i=1}^{n+1} R_i^2 \right) \leq \text{diam}(\mathcal{X}) \left(\frac{1}{n+1} \sum_{i=1}^{n+1} R_i \right) \leq \frac{2}{n+1} \text{diam}(\mathcal{X})^2$. ■

Lemma 6.2 (distance to k -nearest-neighbor) *Let $k \geq 1$. Consider a probability distribution with compact support in $\mathcal{X} \subset \mathbb{R}^d$. Consider $n+1$ points x_1, \dots, x_n, x_{n+1} sampled i.i.d. from \mathcal{X} . Then the*

expected squared ℓ_∞ -distance between x_{n+1} and its k -nearest-neighbor is less than $8\text{diam}(\mathcal{X})^2 \left(\frac{2k}{n}\right)^{2/d}$ for $d \geq 2$, and less than $\frac{2k}{n}\text{diam}(\mathcal{X})^2$ for $d = 1$.

Proof Without loss of generality, we assume $2k \leq n$ (otherwise, the bound is trivial). We can then divide randomly (and independently) the n first points into $2k$ sets of size approximately $\frac{n}{2k}$. We denote $x_{(k)}^j$ a 1-nearest neighbor of x_{n+1} within the j -th set. The squared distance from x_{n+1} to the k -nearest neighbor among all first n points is less than the k -th smallest of the distances $\|x_{n+1} - x_{(k)}^j\|_\infty^2$, $j \in \{1, \dots, 2k\}$, because we take a k -nearest neighbor over a smaller set. This k -th smallest distance is less than $\frac{1}{k} \sum_{j=1}^{2k} \|x_{n+1} - x_{(k)}^j\|_\infty^2$ (this is a general fact that the k -smallest element among non-negative p elements, is less than their sum divided by $p - k$).

Thus, using the lemma above, we get that the desired averaged distance is less than

$$\frac{1}{k} \sum_{j=1}^{2k} 4 \frac{\text{diam}(\mathcal{X})^2}{\left(\frac{n}{2k}\right)^{2/d}} = 8 \frac{\text{diam}(\mathcal{X})^2}{n^{2/d}} (2k)^{2/d}.$$

A similar argument can be extended to $d = 1$. ■

Putting things together, we get the following result for the consistency of k -nearest-neighbors.

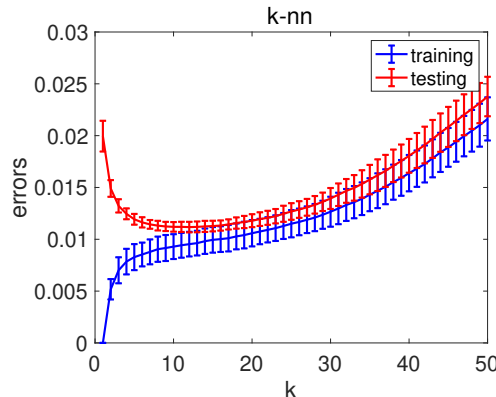
Proposition 6.3 (Convergence rate for k -nearest-neighbors) *Assume bounded noise (H1) and a Lipschitz-continuous target function (H2). Then for the k -nearest-neighbor estimate \hat{f} with the ℓ_∞ -norm, we have, for $d \geq 2$:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leq \frac{\sigma^2}{k} + 8B^2 \text{diam}(\mathcal{X})^2 \left(\frac{2k}{n}\right)^{2/d}. \quad (6.5)$$

Balancing the two terms above is obtained with $k \propto n^{2/(2+d)}$, and we obtain the same result as for the other local averaging schemes. See more details in [34] and [23].

Exercise 6.1 *Show that if the Bayes rate is 0 (that is, $\sigma = 0$), then 1-nearest-neighbor is consistent.*

Experiments. For the problem shown in Section 6.2, below, we plot training and testing errors averaged over 32 replications (with error bar showing the standard deviations), where we clearly see the trade-off in the choice of k .

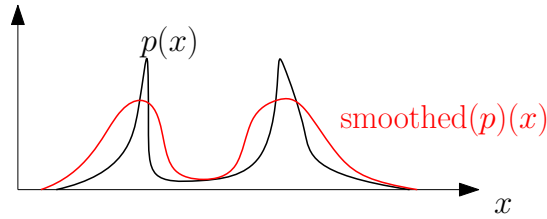


6.3.3 Kernel regression (Nadaraya-Watson) (♦)

In this section, we assume that $\mathcal{X} = \mathbb{R}^d$, and for simplicity, we assume that $dp(x)$ has a density p with respect to the Lebesgue measure. We also assume that $k(x, x') = q_h(x - x') = h^{-d}q(\frac{1}{h}(x - x'))$ for a probability density $q : \mathbb{R}^d \rightarrow \mathbb{R}_+$. The function q_h is also a density, which is the density of hz when z has density $q(z)$ (it is thus gets more concentrated around 0 as h tends to zero). With these notations, the weights can be written:

$$\hat{w}_i(x) = \frac{q_h(x - x_i)}{\sum_{j=1}^n q_h(x - x_j)}.$$

Smoothing by convolution. When performing kernel smoothing, quantities like $\frac{1}{n} \sum_{i=1}^n q_h(x - x_i)g(x_i)$ naturally appear. When the number n of observations goes to infinity, by the law of large numbers, it tends almost surely to $\int_{\mathbb{R}^d} q_h(x - z)g(z)p(z)dz$, which is exactly the convolution between the function q_h and the function $x \mapsto p(x)g(x)$, which we can denote $(pg) * q_h(x)$. The function q_h is a probability density that is putting all most its weights at range of values which are of order h , e.g., for kernels like the Gaussian kernel or the box kernel. Thus convolution will smooth the function pg by averaging values which are at range h . Thus, when h goes to zero, it converges to the function pg itself. See an example below for $g = 1$.



Note that for this limit to hold, we need to make sure the factors in n and h^d are present.

We can now look at the generalization bound from Eq. (6.3) and see how it applies to kernel regression. We now consider the ℓ_2 -distance for simplicity, and consider the variance and bias terms separately, first with an asymptotic result and then a formal result.

Variance term. We have, for a fixed $x \in \mathcal{X}$:

$$n \sum_{i=1}^n \hat{w}_i(x)^2 = \frac{\frac{1}{n} \sum_{i=1}^n q_h(x - x_i)^2}{\left(\frac{1}{n} \sum_{i=1}^n q_h(x - x_i)\right)^2}.$$

Using the law of large numbers and the smoothing reasoning above, this sum $n \sum_{i=1}^n \hat{w}_i(x)^2$ is converging almost surely to

$$\frac{\int_{\mathbb{R}^d} q_h(x - z)^2 p(z) dz}{\left(\int_{\mathbb{R}^d} q_h(x - z) p(z) dz\right)^2} = \frac{q_h^2 * p(x)}{(q_h * p(x))^2}.$$

When h tends to zero, then the denominator above $(q_h * p(x))^2$ tends to $p(x)^2$ because the bandwidth of the smoothing goes to zero. The numerator above corresponds to the smoothing of p by the density $x \mapsto \frac{q_h(x)^2}{\int_{\mathbb{R}^d} q_h(u)^2 du}$, and is thus asymptotically equivalent to $p(x) \int_{\mathbb{R}^d} q_h(u)^2 du = p(x) h^{-d} \int_{\mathbb{R}^d} q(u)^2 du$.

Overall, when n tends to infinity, and h tends to zero, we get:

$$\sum_{i=1}^n \hat{w}_i(x)^2 \sim \frac{1}{nh^d} \frac{1}{p(x)} \int_{\mathbb{R}^d} q(u)^2 du,$$

and thus

$$\int_{\mathcal{X}} \left[\sum_{i=1}^n \hat{w}_i(x)^2 \right] p(x) dx \sim \frac{1}{nh^d} \text{vol}(\text{supp}(dp)) \int_{\mathbb{R}^d} q(u)^2 du.$$

Bias. With the same intuitive reasoning, we get, when n tends to infinity:

$$\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \rightarrow \frac{\int_{\mathbb{R}^d} q_h(x-z) \|x-z\|_2^2 p(z) dz}{\int_{\mathbb{R}^d} q_h(x-z) p(z) dz}.$$

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:

$$\int_{\mathcal{X}} \left[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \right] p(x) dx \sim h^2 \int_{\mathbb{R}^d} q(u) \|u\|_2^2 du.$$

Therefore, overall we get an *asymptotic* bound proportional to (up to constants depending on q):

$$\frac{\sigma^2}{nh^d} + B^2 h^2,$$

leading to the same upper-bound as for partitioning estimates, by setting $h \propto n^{-1/(d+2)}$.

Formal reasoning (♦♦). We can make the informal reasoning above more formal using concentration inequalities, leading to non-asymptotic bounds of the same nature (simply more complicated), that make explicit the joint dependence on n and h . We will prove the following result:

Proposition 6.4 (Convergence rate for Nadaraya-Watson estimation) *Assume bounded noise (H1) and a Lipschitz-continuous target function (H2), and a function $q : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\int_{\mathbb{R}^d} q(z) dz = 1$, and $\|q\|_\infty = \sup_{z \in \mathbb{R}^d} q(z)$ is finite. We also assume that $p(x) \in [p_{\min}, p_{\max}]$ for all $x \in \mathcal{X}$. Then for the Nadaraya-Watson estimate \hat{f} , we have:*

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leq \frac{4\|q\|_\infty}{p_{\min}} \frac{2\sigma^2 + B \text{diam}(\mathcal{X})^2}{nh^d} + 2h^2 \cdot \frac{p_{\max}}{p_{\min}} \int_{\mathbb{R}^d} q(u) \|u\|_2^2 du. \quad (6.6)$$

Before giving the proof, we note that the optimal bandwidth parameter is indeed proportional to $h \propto n^{-1/(d+2)}$, with an overall excess risk proportional to $n^{-2/(d+2)}$.

Proof of Proposition 6.4 (♦) In order to deal with the denominator in the definition of the weights, we can first use Bernstein’s inequality, applied to the random variables $q_h(x - x_i)$ which is almost surely in $[0, h^{-d}\|q\|_\infty]$, to bound

$$\mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n q_h(x - x_i) \leq \mathbb{E}[q_h(x - z)] - \varepsilon\right) \leq \exp\left(-\frac{n\varepsilon^2}{2\mathbb{E}[q_h^2(x - z)] + 2\|q\|_\infty h^{-d}\varepsilon/3}\right).$$

We get with $\varepsilon = \frac{1}{2}\mathbb{E}[q_h(x-z)]$, using $\mathbb{E}[q_h^2(x-z)] \leq \|q\|_\infty h^{-d}\mathbb{E}[q_h(x-z)]$:

$$\begin{aligned} \mathbb{P}(\mathcal{A}(x)) &\leq \exp\left(-\frac{\frac{n}{4}(\mathbb{E}[q_h(x-z)])^2}{2\mathbb{E}[q_h^2(x-z)] + \mathbb{E}[q_h(x-z)]h^{-d}\|q\|_\infty/3}\right) \\ &\leq \exp\left(-\frac{\frac{n}{4}\mathbb{E}[q_h(x-z)]}{(7/3)h^{-d}\|q\|_\infty}\right) \leq \frac{\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]} \times \frac{1}{e} \frac{28}{3} \leq \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}, \end{aligned}$$

where $\mathcal{A}(x)$ is the event $\mathcal{A} = \{\frac{1}{n}\sum_{i=1}^n q_h(x-x_i) \leq \frac{1}{2}\mathbb{E}[q_h(x-z)]\}$. We can now bound bias and variance.

Variance. For a fixed $x \in \mathcal{X}$, we get

$$\begin{aligned} \mathbb{E}\left[\sum_{i=1}^n \hat{w}_i(x)^2\right] &= \mathbb{E}\left[1_{\mathcal{A}(x)} \sum_{i=1}^n \hat{w}_i(x)^2\right] + \mathbb{E}\left[1_{\mathcal{A}(x)^c} \sum_{i=1}^n \hat{w}_i(x)^2\right] \\ &\leq \mathbb{P}(\mathcal{A}(x)) + \frac{4}{(n\mathbb{E}[q_h(x-z)])^2} \mathbb{E}\left[\sum_{i=1}^n q\left(\frac{1}{h}(x-x_i)\right)^2\right] \\ &\leq \frac{4\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]} + \frac{4\mathbb{E}[q_h(x-z)^2]}{n[\mathbb{E}q_h(x-z)]^2} \leq \frac{8\|q\|_\infty}{nh^d\mathbb{E}[q_h(x-z)]}. \end{aligned}$$

Moreover, we have $\mathbb{E}[q_h(x-z)] = \int_{\mathbb{R}^d} dp(x-hu)q(u)du = p * q_h(x)$. This leads to an overall bound on the variance term as $\int_{\mathcal{X}} \mathbb{E}\left[\sum_{i=1}^n \hat{w}_i(x)^2\right] p(x)dx \leq \frac{8\|q\|_\infty}{nh^d} \int_{\mathcal{X}} \frac{p(x)}{p * q_h(x)} dx$.

Bias term. We have a similar reasoning for the bias term. Indeed, we get for a given $x \in \mathcal{X}$:

$$\begin{aligned} &\mathbb{E}\left[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\right] \\ &= \mathbb{E}\left[1_{\mathcal{A}(x)} \sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\right] + \mathbb{E}\left[1_{\mathcal{A}(x)^c} \sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\right] \\ &\leq \mathbb{P}(\mathcal{A}(x)) \cdot \text{diam}(\mathcal{X})^2 + \frac{2}{n\mathbb{E}[q_h(x-z)]} \cdot n\mathbb{E}[q_h(x-z)\|x-z\|_2^2] \\ &\leq \frac{4\|q\|_\infty}{nh^d q_h * p(x)} \cdot \text{diam}(\mathcal{X})^2 + \frac{2h^2}{q_h * p(x)} \cdot \int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du. \end{aligned}$$

This leads to an overall bound on the bias term as $\int_{\mathcal{X}} \mathbb{E}\left[\sum_{i=1}^n \hat{w}_i(x)\|x-x_i\|_2^2\right] p(x)dx \leq \frac{4\|q\|_\infty}{nh^d} \int_{\mathcal{X}} \frac{p(x)}{p * q_h(x)} dx$.

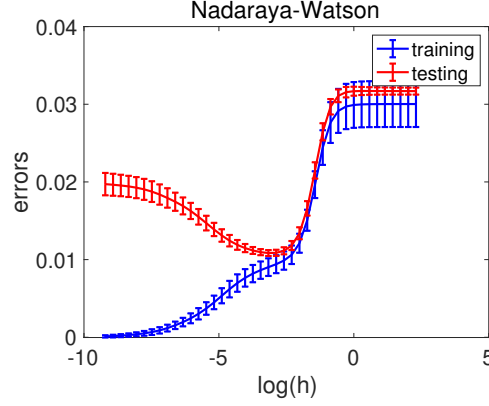
$$\text{diam}(\mathcal{X})^2 + h^2 \int_{\mathcal{X}} \frac{2p(x)}{q_h * p(x)} \cdot \left(\int_{\mathbb{R}^d} q(u)\|u\|_2^2 p(x-uh)du\right) dx.$$

Putting things together, and using $p(x) \in [p_{\min}, p_{\max}]$, such that $p * q_h(x) \geq p_{\min}$, we get

$$\int_{\mathcal{X}} \mathbb{E}[(\hat{f}(x) - f^*(x))^2] dp(x) \leq \frac{4\|q\|_\infty}{p_{\min}} \frac{2\sigma^2 + B\text{diam}(\mathcal{X})^2}{nh^d} + 2h^2 \cdot \frac{p_{\max}}{p_{\min}} \int_{\mathbb{R}^d} q(u)\|u\|_2^2 du.$$

■

Experiments. For the problem shown in Section 6.2, below, we plot training and testing errors averaged over 32 replications (and with error bars showing standard deviations), where we clearly see the trade-off in the choice of h .



6.4 Universal consistency (◆)

Above, we have required the following conditions on the weights:

- $\int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x) d(x_i, x)^2 \right] dp(x) \rightarrow 0$ when n tends to infinity, to ensure that the bias goes to zero.
- $\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E} [\hat{w}_i(x)^2] dp(x) \rightarrow 0$ 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 turned out to be optimal).

In order to show universal consistency, that is consistency for any square-integrable functions, we need an extra (technical) assumption, which was first outlined in Stone's theorem [100], namely that there exists $c > 0$ such that for any non-negative integrable function $h : \mathcal{X} \rightarrow \mathbb{R}$, then

$$\int_{\mathcal{X}} \sum_{i=1}^n \mathbb{E} [\hat{w}_i(x) h(x_i)] dp(x) \leq c \cdot \int_{\mathcal{X}} h(x) dp(x). \quad (6.7)$$

Below, h will be the squared deviation between two functions.

⚠ Above, we only take the expectation with respect to the training data, while we use the integral notation to take the expectation with respect to the training distribution.

Then for any $\varepsilon > 0$, and for any $f^* \in L_2(dp(x))$, we can find a function g which is $B(\varepsilon)$ -Lipschitz-continuous and such that $\|f - g\|_{L_2(dp(x))} \leq \varepsilon$, because the set of Lipschitz-continuous functions is dense in $L_2(dp(x))$ (see, e.g., [4]).

Then we have, for a given $x \in \mathcal{X}$:

$$\begin{aligned}
& \mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) [f^*(x_i) - f^*(x)] \right]^2 \right) \\
& \leq \mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) (|f^*(x_i) - g(x_i)| + |g(x_i) - g(x)| + |g(x) - f^*(x)|) \right]^2 \right) \\
& \leq 3\mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) |f^*(x_i) - g(x_i)| \right]^2 \right) + 3\mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) |g(x_i) - g(x)| \right]^2 \right) + 3\mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) |g(x) - f^*(x)| \right]^2 \right) \\
& \quad \text{using the inequality } (a + b + c)^2 \leq 3a^2 + 3b^2 + 3c^2, \\
& \leq 3\mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) |f^*(x_i) - g(x_i)| \right]^2 \right) + 3\mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) B(\varepsilon) d(x, x_i) \right]^2 \right) + 3\mathbb{E} \left(|g(x) - f^*(x)|^2 \right) \\
& \quad \text{since weights sum to one, and } g \text{ is Lipschitz-continuous,} \\
& \leq 3\mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x) |f^*(x_i) - g(x_i)|^2 \right] + 3B(\varepsilon)^2 \mathbb{E} \left(\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \right) + 3\mathbb{E} \left(|g(x) - f^*(x)|^2 \right) \\
& \quad \text{using Jensen's inequality on the second term,} \\
& \leq 3c \cdot \mathbb{E} [|f^*(x) - g(x)|^2] + 3B(\varepsilon)^2 \mathbb{E} \left(\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \right) + 3\mathbb{E} \left(|g(x) - f^*(x)|^2 \right) \text{ using Eq. (6.7).}
\end{aligned}$$

We can now integrate with respect to x , to get

$$\int_{\mathcal{X}} \mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) [f^*(x_i) - f^*(x)] \right]^2 \right) dp(x) \leq 3c \cdot \varepsilon^2 + 3B(\varepsilon)^2 \int_{\mathcal{X}} \mathbb{E} \left(\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \right) dp(x) + 3\varepsilon^2. \quad (6.8)$$

Proving universal consistency. We can then combine the bound above (which gives a bound on the bias) with Eq. (6.2), starting from:

$$\int_{\mathcal{X}} \mathbb{E} [(\hat{f}(x) - f^*(x))^2] dp(x) \leq \int_{\mathcal{X}} \mathbb{E} \left(\left[\sum_{i=1}^n \hat{w}_i(x) |f^*(x_i) - f^*(x)| \right]^2 \right) dp(x) + \sigma^2 \int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x)^2 \right] dp(x),$$

which is the sum of a bias term and a variance term, and for which, together with Eq. (6.8), we can use the same tools for consistency as for Eq. (6.3).

In order to prove universal consistency, we fix a certain ε , from which we obtain some $B(\varepsilon)$. For such a $B(\varepsilon)$, we know how to obtain an overall term $B(\varepsilon)^2 \int_{\mathcal{X}} \mathbb{E} \left(\sum_{i=1}^n \hat{w}_i(x) d(x, x_i)^2 \right) dp(x) + \sigma^2 \int_{\mathcal{X}} \mathbb{E} \left[\sum_{i=1}^n \hat{w}_i(x)^2 \right] dp(x)$, for a well chosen hyperparameter and number of observations n (see previous sections). Thus, if the extra condition in Eq. (6.7) is satisfied, these three methods are universally consistent.

We can now look at the three cases:

- Partitioning: We have then $c = 2$, and we get universal consistency. Indeed, we have:

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)f(x_i)] &= \sum_{j \in J} \sum_{i=1}^n \mathbb{E}[\hat{w}_i(x)1_{x \in A_j}f(x_i)] \\
&= \sum_{j \in J} \mathbb{E} \left(1_{x \in A_j} \left[1_{n_{A_j} > 0} \frac{1}{n_{A_j}} \sum_{i \in B_j} f(x_i) + 1_{n_{A_j} = 0} \frac{1}{n} \sum_{i=1}^n f(x_i) \right] \right) \\
&\leq \sum_{j \in J} \mathbb{E} \left(1_{x \in A_j} [\mathbb{E}[f(z)|z \in A_j]] + 1_{x \in A_j} \frac{1}{n} \sum_{i=1}^n f(x_i) \right) \\
&\leq 2 \mathbb{E}[f(x)].
\end{aligned}$$

- Kernel regression: it can be shown using the same type of techniques outlined for consistency for Lipschitz-continuous functions.
- k -nearest neighbor: the condition in Eq. (6.7) is not easy to show, and is often referred to as Stone's lemma. See [23, Lemma 10.7].

6.5 Adaptivity (◆◆)

As shown above, all local averaging techniques achieve the same performance on Lipschitz-continuous functions, which is a bad unavoidable performance when d grows (curse of dimensionality). Moreover, higher smoothness of the target function does not seem to be easy to leverage.

Positive definite kernel methods will provide simple ways in Chapter 7, as well as neural networks in Chapter 9. Among local averaging techniques, there are ways to do it. For example, using locally linear regression, where one solves for any test point x ,

$$\inf_{\beta_1 \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \sum_{i=1}^n \hat{w}_i(x)(y_i - \beta_1^\top x - \beta_0)^2.$$

(note that the regular regressogram corresponds to setting $\beta_1 = 0$ above). In other words we solve

$$\inf_{\beta_1 \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \int_{\mathcal{Y}} (y - \beta_1^\top x - \beta_0)^2 d\hat{p}(y|x).$$

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 [102]) improves. See an example with the regressogram weights below.

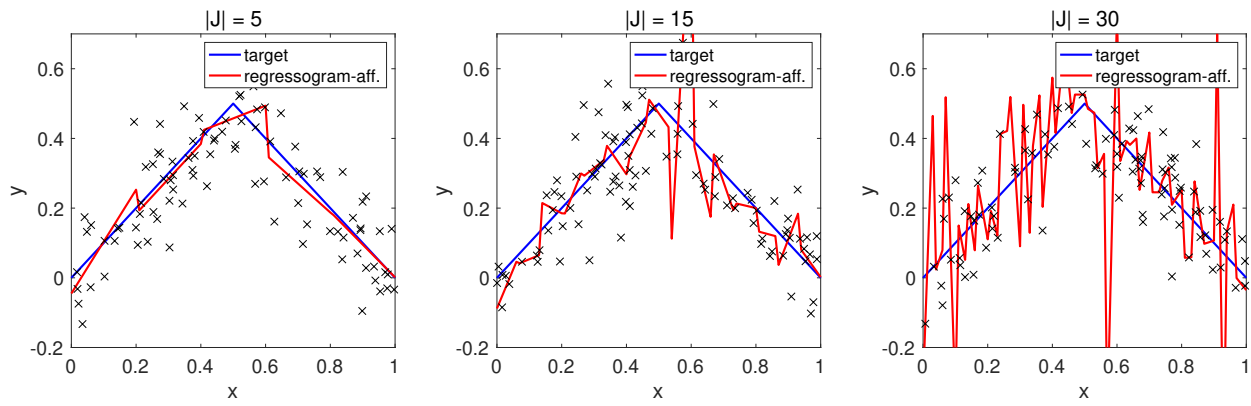


Figure 6.5: Locally linear regression

Chapter 7

Kernel methods

Chapter summary

- Kernels and representer theorems: learning with infinite-dimensional linear models can be done in time that depend on the number of observations by using a kernel function.
- Kernels on \mathbb{R}^d : such models include polynomials and classical Sobolev spaces (functions with square-integrable partial derivatives).
- Algorithms: convex optimization algorithms can be applied with theoretical guarantees and many dedicated developments to avoid the quadratic complexity of computing the kernel matrix.
- Analysis of well-specified models: When the target function is in the associated function space, learning can be done with rates that are independent of dimension.
- Analysis of mis-specified models: if the target is not in the RKHS, the curse of dimensionality cannot be avoided in the worst case situations of few existing derivatives of the target function, but the methods are adaptive to any amount of intermediate smoothness.
- Sharp analysis of ridge regression: for the square loss, a more involved analysis leads to optimal rates in a variety of situations in \mathbb{R}^d .

In this chapter, we consider positive-definite kernel methods. For more details, see [95, 97, 38].

7.1 Introduction

In this chapter, we study empirical risk minimization for linear models, that is, prediction functions $f_\theta : \mathcal{X} \rightarrow \mathbb{R}$ which are linear in their parameters θ , that is, of the form $f_\theta(x) = \langle \theta, \varphi(x) \rangle_{\mathcal{H}}$, where $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ and \mathcal{H} is a Hilbert space (essentially a Euclidean space with potentially infinite dimension), and $\theta \in \mathcal{H}$. We will often use the notation $\langle \theta, \varphi(x) \rangle$ in this chapter instead of $\langle \theta, \varphi(x) \rangle_{\mathcal{H}}$ when this is not ambiguous.

The key difference with Chapter 3 on least-squares estimation is that, (1) we are not restricted to the square loss (although many of the same concepts will play a role, in particular 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* $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$ will be particularly fruitful.

Why is this relevant? The study of infinite-dimensional linear methods is important for several reasons:

- Understanding linear models in finite but very large input dimensions requires tools from infinite-dimensional analysis.
- Kernel methods lead to simple and stable algorithms, with theoretical guarantees, and adaptivity to smoothness of the target function (as opposed to local averaging techniques). They can be applied in high dimensions, with good practical performance (note that for supervised learning problems with many observations in domains such as computer vision and natural language processing, they do not achieve the state of the art anymore, which is achieved by neural networks presented in Chapter 9).
- They can be easily applied when input observations are not vectors.
- They are useful to understand other models such as neural networks (see Chapter 9).



The type of kernel we consider here is different from the ones in Chapter 6. The ones here are “positive definite”; the ones from Chapter 6 are “non-negative”. See more details in <https://francisbach.com/cursed-kernels/>.

7.2 Representer theorem

Dealing with infinite-dimensional models seems impossible at first because algorithms cannot be run in infinite dimensions. In this section, we show how the kernel function plays a crucial role to achieve lower-dimensional algorithms.

As a motivation, we consider the optimization problem coming from machine learning with linear models, with data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$:

$$\min_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \langle \varphi(x_i), \theta \rangle) + \frac{\lambda}{2} \|\theta\|^2, \quad (7.1)$$

assuming the loss function ℓ is already from $\mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}$ and not from $\mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ (e.g., hinge loss, logistic loss or least-squares, see Chapter 4).

The key property of the objective function in Eq. (7.1) is that it accesses the input observations $x_1, \dots, x_n \in \mathcal{X}$, only through dot-products $\langle \theta, \varphi(x_i) \rangle$, $i = 1, \dots, n$, and that we penalize using the Hilbert norm $\|\theta\|$. The following theorem is crucial and has a particularly simple proof.

Theorem 7.1 (Representer theorem [62]) *Let $\varphi : \mathcal{X} \rightarrow \mathcal{H}$. Let $(x_1, \dots, x_n) \in \mathcal{X}^n$, and assume that the functional $\Psi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ is strictly increasing with respect to the last variable, then the infimum of*

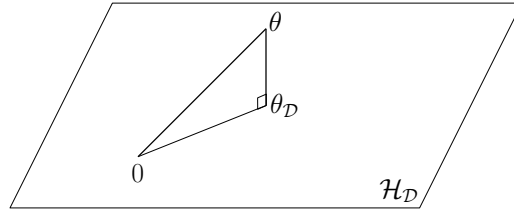
$\Psi(\langle\theta, \varphi(x_1)\rangle, \dots, \langle\theta, \varphi(x_n)\rangle, \|\theta\|^2)$ can be obtained by restricting to a vector θ of the form

$$\theta = \sum_{i=1}^n \alpha_i \varphi(x_i),$$

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 feature vectors.

Let $\theta_{\mathcal{D}} \in \mathcal{H}_{\mathcal{D}}$ and $\theta_{\perp} \in \mathcal{H}_{\mathcal{D}}^{\perp}$ be such that $\theta = \theta_{\mathcal{D}} + \theta_{\perp}$, a decomposition which is using the Hilbertian structure of \mathcal{H} . Then $\forall i \in \{1, \dots, n\}$, $\langle\theta, \varphi(x_i)\rangle = \langle\theta_{\mathcal{D}}, \varphi(x_i)\rangle + \langle\theta_{\perp}, \varphi(x_i)\rangle$ with $\langle\theta_{\perp}, \varphi(x_i)\rangle = 0$.



From Pythagoreas theorem, we get: $\|\theta\|^2 = \|\theta_{\mathcal{D}}\|^2 + \|\theta_{\perp}\|^2$. Therefore we have:

$$\begin{aligned} \Psi(\langle\theta, \varphi(x_1)\rangle, \dots, \langle\theta, \varphi(x_n)\rangle, \|\theta\|^2) &= \Psi(\langle\theta_{\mathcal{D}}, \varphi(x_1)\rangle, \dots, \langle\theta_{\mathcal{D}}, \varphi(x_n)\rangle, \|\theta_{\mathcal{D}}\|^2 + \|\theta_{\perp}\|^2) \\ &\geq \Psi(\langle\theta_{\mathcal{D}}, \varphi(x_1)\rangle, \dots, \langle\theta_{\mathcal{D}}, \varphi(x_n)\rangle, \|\theta_{\mathcal{D}}\|^2). \end{aligned}$$

Thus

$$\inf_{\theta \in \mathcal{H}} \Psi(\langle\theta, \varphi(x_1)\rangle, \dots, \langle\theta, \varphi(x_n)\rangle, \|\theta\|^2) = \inf_{\theta \in \mathcal{H}_{\mathcal{D}}} \Psi(\langle\theta, \varphi(x_1)\rangle, \dots, \langle\theta, \varphi(x_n)\rangle, \|\theta\|^2),$$

which is exactly the desired result. ■

This implies that the minimizer of Eq. (7.1) can be looked among the vectors of the form $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$:

Corollary 7.1 (Representer theorem for supervised learning) For $\lambda > 0$,

$$\inf_{\theta \in \mathcal{H}} \frac{1}{n} \sum \ell(y_i, \langle\theta, \varphi(x_i)\rangle) + \frac{\lambda}{2} \|\theta\|^2 = \inf_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum \ell(y_i, \langle\theta, \varphi(x_i)\rangle) + \frac{\lambda}{2} \|\theta\|^2 \text{ such that } \theta = \sum_{i=1}^n \alpha_i \varphi(x_i).$$

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, 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 which is the dot product between feature vectors:

$$k(x, x') = \langle\varphi(x), \varphi(x')\rangle.$$

We have:

$$\forall j \in \{1, \dots, n\}, \langle \theta, \varphi(x_j) \rangle = \sum_{i=1}^n \alpha_i k(x_i, x_j) = (K\alpha)_j$$

where $K \in \mathbb{R}^{n \times n}$ is the *kernel matrix*, such that $K_{ij} = \langle \varphi(x_i), \varphi(x_j) \rangle = k(x_i, x_j)$, and

$$\|\theta\|^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K_{ij} = \alpha^\top K \alpha.$$

We can then write:

$$\inf_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \langle \theta, \varphi(x_i) \rangle) + \frac{\lambda}{2} \|\theta\|^2 = \inf_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i) + \frac{\lambda}{2} \alpha^\top K \alpha.$$

For a test point $x \in \mathcal{X}$, we have $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$.

Thus, the input observations are summarized in the kernel matrix and the kernel function, regardless of the dimension of \mathcal{H} . Moreover, computing the feature vector $\varphi(x)$ explicitly is never needed! This is the *kernel trick*. The kernel trick allows to:

- replace \mathcal{H} by \mathbb{R}^n ; this is interesting computationally when the dimension of \mathcal{H} is very large (see more details in Section 7.4),
- separate the representation problem (design of kernels on a set \mathcal{X}) and algorithms and 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).

7.3 Kernels

In the section above, we have introduced the kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ as obtained from a dot product $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$. The associated kernel matrix is then a matrix of dot-products (often called a “Gram matrix”), and is thus symmetric positive semi-definite, that is, all of its eigenvalues are non-negative, or $\forall \alpha \in \mathbb{R}^n, \alpha^\top K \alpha \geq 0$. It turns out that this simple property is enough to impose the existence of a feature function.

⚠ If $\mathcal{H} = \mathbb{R}^d$, and $\Phi \in \mathbb{R}^{n \times d}$ is the matrix of features (design matrix in the context of regression) with i -th row composed of $\varphi(x_i)$, then $K = \Phi \Phi^\top \in \mathbb{R}^{n \times n}$ is the kernel matrix, while $\frac{1}{n} \Phi^\top \Phi \in \mathbb{R}^{d \times d}$ is the empirical covariance matrix.

Definition 7.1 *a function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a positive definite kernel if and only if all kernel matrices are symmetric positive semi-definite.*

The important following theorem that dated back to Aronszajn (1950), with an elegant constructive proof. Note the total absence of assumptions on the set \mathcal{X} .

Theorem 7.2 ([5]) *k is a positive definite kernel if and only if there exists a Hilbert space \mathcal{H} , and a function $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ such that $\forall x, x', k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$.*

Partial proof One direction is straightforward. For the other direction we consider a positive-definite kernel, and we will construct explicitly a space of functions from \mathcal{X} to \mathbb{R} with a dot-product. We define the set $\mathcal{H}' \subset \mathbb{R}^{\mathcal{X}}$ as the set of linear combinations of kernel functions $\sum_{i=1}^n \alpha_i k(\cdot, x_i)$ for any integer n , any set of n points and any $\alpha \in \mathbb{R}^n$. This is a vector space, on which we can define a dot-product through

$$\left\langle \sum_{i=1}^n \alpha_i k(\cdot, x_i), \sum_{j=1}^m \beta_j k(\cdot, x'_j) \right\rangle = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \beta_j k(x_i, x'_j). \quad (7.2)$$

One can check that this is a well-defined function on $\mathcal{H}' \times \mathcal{H}'$ (the value does not depend on the chosen representation as linear combination of kernel functions), that it is a dot-product on \mathcal{H}' (indeed, in Eq. (7.2) above, when $\alpha = \beta$ and the x 's and the y 's are the same, we get a positive number because of the positivity of the kernel k), which satisfies the two properties for any $f \in \mathcal{H}'$, $x, x' \in \mathcal{X}$:

$$\langle k(\cdot, x), f \rangle = f(x) \text{ and } \langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x').$$

These are called reproducing properties, and corresponds to an explicit construction $\varphi(x) = k(\cdot, x)$.

The space \mathcal{H}' is called “pre-Hilbertian”, because it is not complete. It can be “completed” into a Hilbert space \mathcal{H} with the same reproducing property. See [5, 21] for more details. ■

We can make the following observations:

- \mathcal{H} is called the “feature space”, and φ the “feature map”, that goes from the “input space” \mathcal{X} to the feature space \mathcal{H} .
- No assumptions are needed about the input space \mathcal{X} , and no regularity assumptions are needed for k . Up to isomorphisms, the feature map and space happen to be unique. The particular space of functions, we built is called the *reproducing kernel Hilbert space (RKHS)*, associated to \mathcal{H} , for which $\varphi(x) = k(\cdot, x)$.
- A classical intuitive interpretation of the identity $\langle k(\cdot, x), f \rangle = f(x)$ is that the function evaluation is the dot-product with a function (this in fact another characterization). If $L_2(\mathbb{R}^d)$ was an RKHS, this would mean that there exists a function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that $\int_{\mathbb{R}^d} k(x, x') f(x') dx' = f(x)$. In other words, $k(x, x') dx'$ would be a Dirac measure at x , which is impossible (as Dirac measures have no density with respect to the Lebesgue measure). Thus $L_2(\mathbb{R}^d)$ is a Hilbert space which is too large to be an RKHS.
- Given a positive-definite kernel k , we can thus associate it to some feature map φ such that $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$, but also to a *space of functions on \mathcal{X} with a given norm*, either directly through the RKHS above, 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 $\|\theta\|_{\mathcal{H}}^2$.

⚠ From now on, we will denote elements of the Hilbert space \mathcal{H} through the notation $f \in \mathcal{H}$ to highlight the fact that we are considering a space of functions from \mathcal{X} to \mathbb{R} , except for optimization algorithms in Section 7.4, where will use the notation $\langle \theta, \varphi(x) \rangle_{\mathcal{H}}$ instead of $f(x)$.

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, sometimes it is not. In the next section, we will look at the main examples, and describe the associated spaces of functions (and the corresponding norms).

Exercise 7.1 *The sum and (pointwise) product of kernels are kernels. What are their associated feature spaces and feature maps?*

We now look at different ways of building the kernels, by starting first from the feature vector (e.g., linear kernels), from the kernel and explicit feature map (polynomial kernel), from the norm (translation-invariant kernel on $[0, 1]$), or from the kernel without explicit features (translation-invariant kernel on \mathbb{R}^d).

7.3.1 Linear and polynomial kernels

We start with the most obvious kernels on $\mathcal{X} = \mathbb{R}^d$, for which feature maps are easily found.

Linear kernel. $k(x, x') = x^\top x'$. It corresponds to 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 (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 r a positive integer, the kernel $k(x, x') = (x^\top x')^r$ can be expanded as (with the binomial theorem¹):

$$k(x, x') = \left(\sum_{i=1}^d x_i x'_i \right)^r = \sum_{\alpha_1 + \dots + \alpha_d = r} \binom{r}{\alpha_1, \dots, \alpha_d} \underbrace{(x_1 x'_1)^{\alpha_1} \dots (x_d x'_d)^{\alpha_d}}_{(x_1^{\alpha_1} \dots x_d^{\alpha_d})(x'_1)^{\alpha_1} \dots (x'_d)^{\alpha_d}},$$

where the sum is over all non-negative integer vectors $(\alpha_1, \dots, \alpha_d)$. We have an explicit feature map: $\varphi(x) = \left(\binom{r}{\alpha_1, \dots, \alpha_d}^{\frac{1}{2}} x_1^{\alpha_1} \dots x_d^{\alpha_d} \right)_{\alpha_1 + \dots + \alpha_d = r}$, and the set of functions is the set of homogeneous polynomials on \mathbb{R}^d , which has dimension $\binom{d+r-1}{r}$.

When d and r grows, the dimension of the feature space grows as d^r , an explicit representation is not desirable, and the kernel trick can be advantageous. Note however, that the associated norm (which penalizes coefficients of the polynomials), is hard to interpret (as a small change in a single high-order coefficient can lead to significant changes).

Exercise 7.2 *Show that the kernels $k(x, y) = (1 + x^\top y)^r$ corresponds to the set of all monomials $x_1^{\alpha_1} \dots x_d^{\alpha_d}$ such that $\alpha_1 + \dots + \alpha_d \leq r$.*

7.3.2 Translation-invariant kernels on $[0, 1]$

We consider $\mathcal{X} = [0, 1]$, and kernels of the form $k(x, x') = q(x - x')$ with a function $q : [0, 1] \rightarrow \mathbb{R}$, which is 1-periodic. We will show how they emerge from penalties on the Fourier coefficients of functions. We will use the fact that squared integrable functions which are 1-periodic can be expanded in Fourier series, that is, $q(x) = \sum_{m \in \mathbb{Z}} e^{2im\pi x} \hat{q}_m$, with $\hat{q}_m = \int_0^1 q(x) e^{-2im\pi x} dx$, for $m \in \mathbb{Z}$.

¹https://en.wikipedia.org/wiki/Binomial_theorem

When presenting translation-invariant kernels, we can choose to start from the kernel or from the associated squared norm. In this section, we start from the squared norm, while in the next one, we start from the kernel.

Given a 1-periodic function f decomposed into its Fourier series $f(x) = \sum_{m \in \mathbb{Z}} e^{2im\pi x} \hat{f}_m$, we consider the penalty

$$\sum_{m \in \mathbb{Z}} c_m |\hat{f}_m|^2,$$

with $c \in \mathbb{R}_+^{\mathbb{Z}}$; this penalty can be interpreted through a feature map and a standard Euclidean norm. Indeed, it corresponds to the feature vector $\varphi(x)_m = \frac{e^{2im\pi x}}{\sqrt{c_m}}$, and $\theta \in \mathbb{C}^{\mathbb{Z}}$, such that $\theta_m = \hat{f}_m \sqrt{c_m}$ (we can easily consider complex-valued features instead of real-valued features if Hermitian dot-products are considered), so that $f(x) = \langle \theta, \varphi(x) \rangle$ and $\sum_{m \in \mathbb{Z}} |\theta_m|^2$ is equal to the norm $\sum_{m \in \mathbb{Z}} c_m |\hat{f}_m|^2$.

Thus the associated kernel is

$$k(x, x') = \sum_{m \in \mathbb{Z}} \varphi(x)_m \varphi(x')_m^* = \sum_{m \in \mathbb{Z}} \frac{e^{2im\pi x}}{\sqrt{c_m}} \frac{e^{-2im\pi x'}}{\sqrt{c_m}} = \sum_{m \in \mathbb{Z}} \frac{1}{c_m} e^{2im\pi(x-x')} = q(x - x').$$

What we showed above is that any penalty of the form $\sum_{m \in \mathbb{Z}} c_m |\hat{f}_m|^2$ defines a squared RKHS norm as soon as c_m is positive and $\sum_{m \in \mathbb{Z}} \frac{1}{c_m}$ is finite. The kernel function is then of the form $k(x, y) = q(x - y)$ with q being 1-periodic, and such that the Fourier series has non-negative real values $\hat{q}_m = c_m^{-1}$.

Penalization of derivatives. For certain penalties based on c , there is a natural link with penalties on derivatives, as, if f is s -times differentiable with squared integrable derivative, we have $f^{(s)}(x) = \sum_{m \in \mathbb{Z}} (2im\pi)^s e^{2im\pi x} \hat{f}_m$, and thus, from Parseval's theorem:

$$\int_0^1 |f^{(s)}(x)|^2 dx = (2\pi)^{2s} \sum_{m \in \mathbb{Z}} m^{2s} |\hat{f}_m|^2.$$

In this chapter we will consider penalizing such derivatives, leading to Sobolev spaces on $[0, 1]$. The following examples are often considered:

- **Bernoulli polynomials:** we can consider $c_0 = (2\pi)^{-2s}$ and $c_m = |m|^{2s}$ for $m \neq 0$, for which the associated norm is $\|f\|_{\mathcal{H}}^2 = \frac{1}{(2\pi)^{2s}} \int_0^1 |f^{(s)}(x)|^2 dx + \frac{1}{(2\pi)^{2s}} \left(\int_0^1 |f(x)|^2 dx \right)$. The corresponding kernel $k(x, x')$ can then be written as

$$k(x, x') = \sum_{m \in \mathbb{Z}} c_m^{-1} e^{2im\pi(x-x')} = (2\pi)^{2s} + \sum_{m \geq 1} \frac{2 \cos[2\pi m(x - x')]}{m^{2s}}.$$

In order to have an expression for q in closed form we notice that if we define $\{x\} = x - [x] \in [0, 1]$ the fractional part of x , the function $x \mapsto \{x\}$ has (by integration by part) an m -th Fourier coefficient equal to $\int_0^1 e^{-2im\pi x} x dx = \frac{i}{2m\pi}$. Similarly, the s -th power of $\{x\}$ has similarly an m -th Fourier coefficient which is an order s polynomial in m^{-1} . This implies that $k(x, x')$ has to be an order s polynomial in $\{x - x'\}$.

For $s = 1$, we have $k(x, x) = (2\pi)^2 + 2 \sum_{m \geq 1} m^{-2} = (2\pi)^2 + \pi^2/3$; moreover by using the Fourier series expansion $\{t\} = \frac{1}{2} - \frac{1}{2\pi} \sum_{m \geq 1} \frac{2 \sin[2\pi mt]}{m}$, and integrating, we get

$$k(x, x') = 2\pi^2 \{x - x'\}^2 - 2\pi^2 \{x - x'\} + \pi^2/3 + (2\pi)^2.$$

For $s \geq 1$, we have the closed-form expression $k(x, x') = (2\pi)^{2s} + (-1)^{s-1} \frac{(2\pi)^{2s}}{(2s)!} B_{2s}(\{x - x'\})$, where B_{2s} the $(2s)$ -th Bernoulli polynomial², from which we can “check” the computation above since $B_2(t) = t^2 - t + 1/6$.

- **Periodic exponential kernel:** we can consider $c_m = 1 + \alpha^2 |m|^2$, for which we have also a closed-form formula, with the penalty $\|f\|_{\mathcal{H}}^2 = \frac{\alpha^2}{(2\pi)^2} \int_0^1 |f^{(s)}(x)|^2 dx + \int_0^1 |f(x)|^2 dx$.

Exercise 7.3 (◆◆◆) Give a closed-form for the kernel $k(x, x') = \sum_{m \in \mathbb{Z}} \frac{e^{2im\pi(x-x')}}{1 + \alpha^2 |m|^2}$. Hint: use the Cauchy residue formula (see <https://francisbach.com/cauchy-residue-formula/>).

These kernels are mostly used for their simplicity and their explicit feature map, which are simpler than the kernels which are most used below (with similar links with Sobolev spaces). Note also, that for the uniform distribution on $[0, 1]$, the Fourier basis will be an orthogonal eigenbasis of the covariance operator with eigenvalues c_m^{-1} (see Section 7.6.5).

We saw that for the kernel $q(x - x')$ with Fourier series \hat{q}_m for q , the associated norm is $\sum_{m \in \mathbb{Z}} \frac{|\hat{f}_m|^2}{\hat{q}_m}$. We now extend this to Fourier transforms (instead of Fourier series).

7.3.3 Translation-invariant kernels on \mathbb{R}^d

We consider $\mathcal{X} = \mathbb{R}^d$, and a kernel of the form $k(x, x') = q(x - x')$ with a function $q : \mathbb{R}^d \rightarrow \mathbb{R}$. The following theorem gives conditions under which we obtain a positive definite kernel.

Theorem 7.3 (Böchner [90]) *The kernel k is positive definite if and only if q is the Fourier transform of a non-negative Borel measure. As a consequence, if $q \in L^1(dx)$ and its Fourier transform only has non-negative real values, then k is positive definite.*

Partial proof We only give the proof of the consequence, which is the only one that we need. Since q is integrable, $\hat{q}(\omega) = \int_{\mathbb{R}^d} e^{-i\omega^\top x} q(x) dx$ is defined on \mathbb{R}^d and continuous, and we have through the inverse Fourier transform formula:

$$q(x - x') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{q}(\omega) e^{i(x-x')^\top \omega} d\omega.$$

²See https://en.wikipedia.org/wiki/Bernoulli_polynomials.

Let $x_1, \dots, x_n \in \mathbb{R}^d$, let $\alpha_1, \dots, \alpha_n \in \mathbb{R}$. We have:

$$\begin{aligned} \sum_{s,j=1}^n \alpha_s \alpha_j k(x_s, x_j) &= \sum_{s,j=1}^n \alpha_s \alpha_j q(x_s - x_j) = \frac{1}{(2\pi)^d} \sum_{s,j=1}^n \alpha_s \alpha_j \int_{\mathbb{R}^d} e^{i\omega^\top (x_s - x_j)} \hat{q}(\omega) d\omega \\ &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left(\sum_{s,j=1}^n \alpha_s \alpha_j e^{i\omega^\top x_s} (e^{i\omega^\top x_j})^* \right) \hat{q}(\omega) d\omega \\ &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left| \sum_{s=1}^n \alpha_s e^{i\omega^\top x_s} \right|^2 \hat{q}(\omega) d\omega \geq 0, \end{aligned}$$

which shows the positive-definiteness. ■

Construction of the associated norm. We give an intuitive (non-rigorous) reasoning: if q is in $L^1(dx)$, then $\hat{q}(\omega)$ exists and, we have an explicit representation as

$$k(x, x') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \langle \sqrt{\hat{q}(\omega)} e^{i\omega^\top x}, \sqrt{\hat{q}(\omega)} e^{i\omega^\top x'} \rangle d\omega = \int_{\mathbb{R}^d} \langle \varphi(x)_\omega, \varphi(x')_\omega \rangle d\omega,$$

which is of the form $\langle \varphi(x), \varphi(y) \rangle$, with $\varphi(x)_\omega = \frac{1}{(2\pi)^{d/2}} \sqrt{\hat{q}(\omega)} e^{i\omega^\top x}$. If we consider $f(x) = \int_{\mathbb{R}^d} \varphi(x)_\omega \theta_\omega d\omega = \langle \varphi(x), \theta \rangle$, then $\theta_\omega = \frac{1}{(2\pi)^{d/2}} \hat{f}(\omega) / \sqrt{\hat{q}(\omega)}$, and the squared norm of θ is equal to $\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} d\omega$, where \hat{f} denotes the Fourier transform of f . Therefore, the norm of a function $f \in \mathcal{H}$ is (for a formal proof, see [95]):

$$\|f\|_{\mathcal{H}}^2 = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} d\omega.$$

Note the similarity with the penalty for the kernel on $[0, 1]$ (see more similarity below).

Link with derivatives. When f has partial derivatives, then the Fourier transform of $\frac{\partial f}{\partial x_j}$ is equal to $i\omega_j$ times the Fourier transform of f . This leads to, using Parseval's theorem, $\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\omega_j|^2 |\hat{f}(\omega)|^2 d\omega = \int_{\mathbb{R}^d} \left| \frac{\partial f}{\partial x_j}(x) \right|^2 dx$, which extends to higher order derivatives:

$$\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\omega_1^{\alpha_1} \dots \omega_d^{\alpha_d}|^2 |\hat{f}(\omega)|^2 d\omega = \int_{\mathbb{R}^d} \left| \frac{\partial^\alpha f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}(x) \right|^2 dx.$$

This will allow us to find corresponding norms, by expanding $\hat{q}(\omega)^{-1}$ as sums of monomials. We now consider the main classical examples.

Exponential kernel. This is the kernel $q(x - x') = \exp(-\alpha \|x - x'\|_2)$, for which the Fourier transform can be computed as $\hat{q}(\omega) = 2^d \pi^{(d-1)/2} \Gamma((d+1)/2) \frac{\alpha}{(\alpha^2 + \|\omega\|_2^2)^{(d+1)/2}}$. See [106, page 84]. Thus, $\hat{q}(\omega)^{-1}$ is

a sum of monomials, and looking at their orders, we see that the corresponding RKHS norm is penalizing all derivatives up to total order $(d+1)/2$, that is for all $\alpha \in \mathbb{N}^d$ such that $\alpha_1 + \dots + \alpha_d = (d+1)/2$, which is a Sobolev space (fractional for d even).

In particular, for $d = 1$, we have $\hat{q}(\omega) = \frac{2\alpha}{\alpha^2 + \omega^2}$, and thus

$$\|f\|_{\mathcal{H}}^2 = \frac{1}{2\pi} \int_{\mathbb{R}} \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} d\omega = \frac{\alpha}{2} \frac{1}{2\pi} \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega + \frac{1}{2\alpha} \frac{1}{2\pi} \int_{\mathbb{R}} |\omega \hat{f}(\omega)|^2 d\omega = \frac{\alpha}{2} \int_{\mathbb{R}} |f(x)|^2 dx + \frac{1}{2\alpha} \int_{\mathbb{R}} |f'(x)|^2 dx,$$

and we recover the Sobolev space of functions with squared-integrable derivatives.

Gaussian kernel. This is the kernel $q(x - x') = \exp(-\alpha\|x - x'\|^2)$, for which the Fourier transform can be computed as $\hat{q}(\omega) = (\frac{\pi}{\alpha})^{d/2} \exp(-\|\omega\|_2^2/(4\alpha))$. By expanding $\hat{q}(\omega)^{-1}$ through its entire series as $\hat{q}(\omega)^{-1} = (\frac{\pi}{\alpha})^{d/2} \sum_{s=0}^{\infty} (-1)^s \frac{\|\omega\|_2^{2s}}{(4\alpha)^s s!}$, this corresponds to an RKHS norm which is penalizing all derivatives. Note that all members of this RKHS are infinitely differentiable, and thus much smoother than functions coming from the exponential kernel (the RKHS is smaller).

Matern kernels. More generally, one can define a series of kernels so that $\hat{q}(\omega) \propto \frac{1}{(\alpha^2 + \|\omega\|_2^2)^s}$ for $s > d/2$, to ensure integrability of the Fourier transform. These so-called Matern kernels all correspond to Sobolev spaces of order s . See [106, page 84]. A key fact is that to be an RKHS, a Sobolev space has to have many derivatives when d grows; in particular, having only first-order derivatives ($s = 1$) only leads to an RKHS for $d = 1$.

For $s = \frac{d+3}{2}$, we have $k(x, x') \propto (1 + \sqrt{3}\alpha\|x - x'\|_2) \exp(-\sqrt{3}\alpha\|x - x'\|_2)$, and for $s = \frac{d+5}{2}$, we have $k(x, x') \propto (1 + \sqrt{5}\alpha\|x - x'\|_2 + \frac{5}{3}\alpha^2\|x - x'\|_2^2) \exp(-\sqrt{5}\alpha\|x - x'\|_2)$. General values s also lead to closed-form formulas (through Bessel functions).

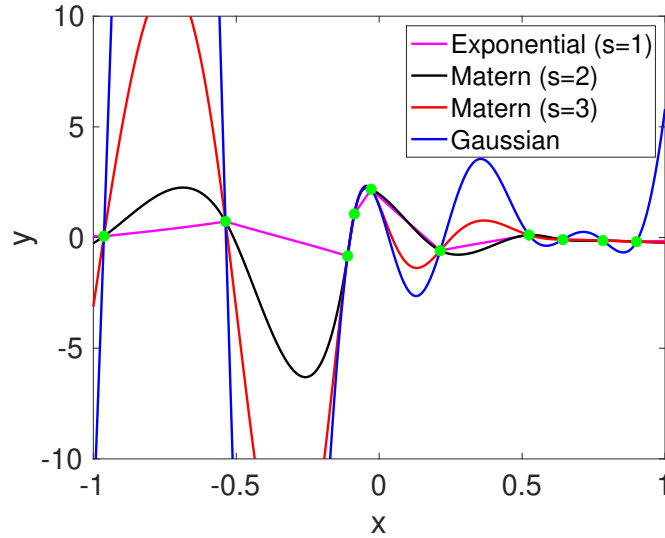
Density in $L_2(dx)$. For all the kernels below, the set \mathcal{H} is dense in $L_2(dx)$, meaning that all functions in $L_2(dx)$ can be approached (with respect to their corresponding norm) by a function in \mathcal{H} . This is made quantitative in Section 7.5.2.

⚠ In this chapter, we will consider two spaces of integrable functions, with respect to the Lebesgue measure dx (which is not a probability measure), which we denote $L_2(dx)$, and with respect to the probability measure of the input data, which we denote $L_2(dp(x))$. If $\frac{dp}{dx}(x)$ is uniformly bounded, then $L_2(dx) \subset L_2(dp(x))$; more precisely, $\|f\|_{L_2(dp(x))} \leq \left\| \frac{dp}{dx} \right\|_{\infty}^{1/2} \|f\|_{L_2(dx)}$.

but the converse is not true, simply because being an element of $L_2(dx)$ imposes a zero limit at infinity, which being an element of $L_2(dp(x))$ does not impose.

Examples of members of RKHS. Below, we sampled $n = 10$ random points in $[-1, 1]$ with 10 random responses, and we look for the function $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for all $i \in \{1 \dots, n\}$ and with minimum norm. Given the representer theorem, we can write $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$, and the interpolation condition implies that $K\alpha = y$, and thus $y = K^{-1}\alpha$.

We consider several kernels below, going from close to piecewise affine interpolation to infinitely differentiable functions (for the Gaussian kernel).



7.3.4 Beyond (♦)

While the theoretical analysis of kernel methods focuses a lot on kernels on \mathbb{R}^d and their link with differentiability properties of the target function, kernels can be applied to a wide variety of problems, with various input types. We give below classical examples (see more details in [97]).

- Set of subsets of a given set V : for example, the function k defined as $k(A, B) = \frac{|A \cap B|}{|A \cup B|}$ is a positive definite kernel.
- 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 group of letters, single original words, or groups of words), and counting the number of occurrences of this word in the corresponding document. This gives a typically a high-dimensional feature vector $\varphi(x)$ (with dimension the size of the vocabulary). Using linear functions on this feature provide a cheap and stable predictors on such data types (better models that take into account the word order can be obtained, such as neural networks, at the expense of significantly more computational resources). See, e.g., [58] for examples.
- Sequences: given some finite alphabet \mathcal{A} , we consider the set \mathcal{X} of finite sequences in \mathcal{A} with arbitrary length. A classical infinite-dimensional feature space is indexed by \mathcal{X} itself, and for $y \in \mathcal{X}$, $\varphi(x)_y$ is equal to 1 if y is a subsequence of x (we could also count the number of times the subsequence y appears in x , or we could add a weight that depend on y , e.g., to penalize longer subsequences). This kernel has an infinite-dimensional feature space, but for two sequences x and x' , we can enumerate all subsequences of x and x' and compare them in polynomial time (there exist much faster algorithms, see [51]). These kernels have many applications in bioinformatics.

The same techniques can be extended to more general combinatorial objects such as trees, graphs (see [97]).

- Images: before neural networks took over in the years 2010s with the use of large amounts of data, several kernels were designed for images, with often a “bag-of-words” assumption which provides for free invariance by translation. The key is what to consider as “words”, i.e., presence of certain local patterns in the image, as well as the regions under which this assumption is made. See [108] for details.

7.4 Algorithms

In this section, we briefly mention algorithms aimed at solving

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2, \quad (7.3)$$

for ℓ being convex with respect to its second variable. We assume that for all $i \in \{1, \dots, n\}$, $k(x_i, x_i) = \|\varphi(x_i)\|^2 \leq R^2$.

Representer theorem. We can directly apply the representer theorem 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.

In the special 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 $\alpha = (K + n\lambda I)^{-1}y$.

However, in general (for the square loss and beyond), it is a ill-conditioned optimization problem because K has often very small eigenvalues (more on this later), and when the loss is smooth, the Hessians are equal to $\frac{1}{n}K \text{Diag}(h)K + \lambda K$, where $h \in \mathbb{R}^n$ is a vector of second-order derivatives of ℓ , so that the Hessians are ill-conditioned.

A better alternative is to first compute a square root of K as $K = \Phi\Phi^\top$, where $\Phi \in \mathbb{R}^{n \times m}$, and m the rank of K , and solve

$$\min_{\beta \in \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (\Phi\beta)_i) + \frac{\lambda}{2} \|\beta\|_2^2,$$

with $\alpha = \Phi^\top \beta$. Note that this corresponds to an explicit feature space representation (that is, the rows of Φ corresponds to features in \mathbb{R}^n for the corresponding data point). For ridge regression, the Hessian of the objective function is then 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 SVD) [47], in running time $O(m^2n)$.

Column sampling. Approximate square roots are a very useful tool, and among various algorithms, approximating $K \in \mathbb{R}^{n \times n}$ from a subset of its columns can be done as $K \approx K(V, I)K(I, I)^{-1}K(I, V)$, where $K(A, B)$ is the sub-matrix of K obtained by taking rows from the set $A \subset \{1, \dots, n\}$ and columns from $B \subset \{1, \dots, n\}$, and $V = \{1, \dots, n\}$. See below for an illustration when $I = \{1, \dots, m\}$ and a partition of the kernel matrix.

$K(I, I)$	$K(I, J)$
$K(J, I)$	$K(J, J)$

This corresponds to an approximate square root $\Phi = K(V, I)K(I, I)^{-1/2} \in \mathbb{R}^{n \times m}$, with $m = |I|$, and it can be computed in time $O(m^2n)$ (computing the entire kernel matrix is not even needed). Then, the complexity is typically $O(m^2n)$ instead of $O(n^3)$ (e.g., when using matrix inversion for ridge regression, for faster algorithms, see below), and is thus linear in n .

Exercise 7.4 (♦) *Show that this corresponds to approximating optimally each $\varphi(x_j)$, $j \notin I$, by a linear combination of $\varphi(x_i)$, $i \in I$.*

This approximation technique, often called “Nyström approximation”, can be analyzed when the columns are chosen randomly [92].

Random features. Some kernels have a special form that leads to specific approximation schemes, that is,

$$k(x, x') = \int_{\mathcal{V}} \varphi(x, v) \varphi(x', v) d\mu(v),$$

where $d\mu$ is a probability distribution on some space \mathcal{V} and $\varphi(x, v) \in \mathbb{R}$. We can then approximate the expectation by an empirical average

$$\hat{k}(x, x') = \frac{1}{m} \sum_{i=1}^m \varphi(x, v_i) \varphi(x', v_i),$$

where the v_i ’s are sampled i.i.d. from $d\mu$. We can thus use an explicit feature representation $\hat{\varphi}(x) = (\frac{1}{\sqrt{m}} \varphi(x, v_i))_{i \in \{1, \dots, m\}}$, and solve

$$\min_{\beta \in \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \hat{\varphi}(x_i)^\top \beta) + \frac{\lambda}{2} \|\beta\|_2^2.$$

For this scheme to make sense, the number m of random features has to be significantly smaller than n , which is often sufficient in practice (see an analysis in [93]).

⚠ Note that dimension reduction is performed independently of the input data (that is the random feature functions $\varphi(\cdot, v_i)$ are selected before the data are observed, as opposed to column sampling which is a data-dependent dimension reduction scheme.

The two classical examples are:

- **Translation-invariant kernels:** $k(x, y) = q(x - y) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{q}(\omega) e^{i\omega^\top(x-y)} d\omega$, for which we can take $\varphi(x, \omega) = \sqrt{q(0)} e^{i\omega^\top x} \in \mathbb{C}$, where ω is sampled from the distribution with density $\frac{1}{(2\pi)^d} \frac{q(\omega)}{q(0)}$, 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 $[0, 2\pi]$ [89].
- **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 $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. For the “rectified linear unit”, that is, $\sigma(\alpha) = \max\{0, \alpha\}$, and for v sampled uniformly on the sphere, we have $k(x, x') = \frac{\|x\|_2 \|x'\|_2}{2(d+1)\pi} [(\pi - \eta) \cos \eta + \sin \eta]$, where $\cos \eta = \frac{x^\top x'}{\|x\|_2 \|x'\|_2}$ [65]. Therefore, we can see a neural network with a large number of hidden neurons, with input weights which are random and not optimized as a kernel method. See a thorough discussion in Chapter 9.

Dual algorithms (♦). For the next two algorithms, we go back to the notation $f(x) = \langle \varphi(x), \theta \rangle$ with $\theta \in \mathcal{H}$ because it is more adapted (and is a direct infinite-dimensional extension of the algorithms from Chapter 5). To solve $\min_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \langle \varphi(x_i), \theta \rangle) + \frac{\lambda}{2} \|\theta\|^2$, for a loss which is convex with respect to the second variable, we can derive a Lagrange dual in the following way (for an introduction to Lagrange duality, see [28]). We start by reformulating the problem as a constrained problem:

$$\begin{aligned} & \min_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \langle \varphi(x_i), \theta \rangle) + \frac{\lambda}{2} \|\theta\|^2 \\ &= \min_{\theta \in \mathcal{H}, u \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, u_i) + \frac{\lambda}{2} \|\theta\|^2 \text{ such that } \forall i \in \{1, \dots, n\}, \langle \varphi(x_i), \theta \rangle = u_i \end{aligned}$$

By Lagrange duality, this is equal to

$$\begin{aligned} & \max_{\alpha \in \mathbb{R}^n} \min_{\theta \in \mathcal{H}, u \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, u_i) + \frac{\lambda}{2} \|\theta\|^2 + \lambda \sum_{i=1}^n \alpha_i (u_i - \langle \varphi(x_i), \theta \rangle) \\ &= \max_{\alpha \in \mathbb{R}^n} \left\{ \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \} + \min_{\theta \in \mathcal{H}} \left\{ \frac{\lambda}{2} \|\theta\|^2 - \lambda \sum_{i=1}^n \alpha_i \langle \varphi(x_i), \theta \rangle \right\} \right\} \text{ by reordering,} \\ &= \max_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \} - \frac{1}{2\lambda} \left\| \sum_{i=1}^n \alpha_i \varphi(x_i) \right\|^2 \text{ with } \theta = \sum_{i=1}^n \alpha_i \varphi(x_i), \\ &= \max_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \min_{u_i \in \mathbb{R}} \{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \} - \frac{1}{2\lambda} \alpha^\top K \alpha, \end{aligned}$$

with $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$ at optimum. Since the functions $\alpha_i \mapsto \min_{u_i \in \mathbb{R}} \{ \ell(y_i, u_i) + n\lambda \alpha_i u_i \}$ are concave (as minima of affine functions), this is a concave maximization problem.

Note the similarity with the representer theorem (existence of $\alpha \in \mathbb{R}^n$ such that $\theta = \sum_{i=1}^n \alpha_i \varphi(x_i)$) and the dissimilarity (one is a minimization problem, one is maximization problem). Moreover, when the loss is smooth, one can show that the function $\min_{u_i \in \mathbb{R}} \{\ell(y_i, u_i) + n\lambda \alpha_i u_i\}$ is a strongly concave function, and thus relatively easy to optimize (in other words, the associated condition numbers are smaller),

Exercise 7.5 (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 using normal equations like in Chapter 3, and relate the two using the matrix inversion lemma $(\Phi\Phi^\top + n\lambda I)^{-1}\Phi = \Phi(\Phi^\top\Phi + n\lambda I)^{-1}$.

SGD (♦). When minimizing an expectation

$$\min_{\theta \in \mathcal{H}} \mathbb{E}[\ell(y, \langle \varphi(x), \theta \rangle)] + \frac{\lambda}{2} \|\theta\|^2$$

like in Chapter 5, the stochastic gradient algorithm leads to the recursion

$$\theta_t = \theta_{t-1} - \gamma_t [\ell'(y_t, \langle \varphi(x_t), \theta_{t-1} \rangle) \varphi(x_t) + \lambda \theta_{t-1}],$$

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

$$\alpha_i^{(t)} = (1 - \gamma_t \lambda) \alpha_i^{(t-1)} \text{ for } i \in \{1, \dots, t-1\}, \text{ and } \alpha_t^{(t)} = -\gamma_t \ell'(y_t, \sum_{i=1}^{t-1} \alpha_i^{(t-1)} \varphi(x_i)).$$

The complexity after t iterations is $O(t^2)$ kernel evaluations. The convergence rates from Chapter 5 apply. More precisely, if the loss is G -Lipschitz continuous, then, for $F(\theta) = \mathbb{E}[\ell(y, \langle \varphi(x), \theta \rangle)] + \frac{\lambda}{2} \|\theta\|^2$, we have, for the averaged iterate $\bar{\theta}_t$,

$$\mathbb{E}[F(\bar{\theta}_t)] - \inf_{\theta \in \mathcal{H}} F(\theta) \leq \frac{G^2 R^2}{\lambda t}.$$



When doing a single pass with $t = n$, then $F(\theta)$ is the regularized expected risk, and we obtain a generalization bound, leading to $\mathbb{E}[\mathcal{R}(f_{\bar{\theta}_t})] \leq \frac{G^2 R^2}{\lambda n} + \inf_{f \in \mathcal{H}} \{\mathcal{R}(f) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2\}$. These bounds are similar than the ones below (which assume a regularized empirical risk minimizer is available).

“Kernelization” of linear algorithms. Beyond supervised learning, many unsupervised learning algorithms can be “kernelized”, such as principal component analysis or canonical correlation analysis. See [95, 97] for details.

7.5 Generalization guarantees - Lipschitz-continuous losses

In this section, we consider a G -Lipschitz-continuous loss function, and consider a minimizer $\hat{f}_D^{(c)}$ of the constrained problem

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) \text{ such that } \|f\|_{\mathcal{H}} \leq D, \quad (7.4)$$

and the unique minimizer $\hat{f}_\lambda^{(r)}$ of the regularized problem

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2. \quad (7.5)$$

We denote by $\mathcal{R}(f) = \mathbb{E}[\ell(y, f(x))]$ the expected risk, and by f^* one of its minimizers (which we assume to be square integrable). We assume $k(x, x) \leq R^2$ almost surely.

We can first relate the excess risk to the L_2 -norm of $f - f^*$, as

$$\begin{aligned} \mathcal{R}(f) - \mathcal{R}(f^*) &\leq \mathbb{E}[|\ell(y, f(x)) - \ell(y, f^*(x))|] \leq G \mathbb{E}[|f(x) - f^*(x)|] \\ &\leq G \sqrt{\mathbb{E}[|f(x) - f^*(x)|^2]} = G \|f - f^*\|_{L_2(dp(x))}, \end{aligned}$$

that is, the excess risk is dominated by the $L_2(dp(x))$ -norm of $f - f^*$. For $\mathcal{X} = \mathbb{R}^d$, and probability measures with bounded density with respect to the Lebesgue measure, we had shown that $\|f\|_{L_2(dp(x))} \leq \left\| \frac{dp}{dx} \right\|_\infty^{1/2} \|f\|_{L_2(dx)}$, so we can replace $G \|f - f^*\|_{L_2(dp(x))}$ by $G \left\| \frac{dp}{dx} \right\|_\infty^{1/2} \|f - f^*\|_{L_2(dx)}$.

7.5.1 Risk decomposition

Constrained problem. Dimension-free results from Chapter 4 (Prop. 4.4), based on Rademacher complexities immediately apply, and we obtain that the estimation error is bounded from above by $\frac{2GDR}{\sqrt{n}}$, leading to:

$$\mathbb{E}[\mathcal{R}(\hat{f}_D^{(c)})] - \mathcal{R}(f^*) \leq \frac{2GDR}{\sqrt{n}} + G \inf_{\|f\|_{\mathcal{H}} \leq D} \|f - f^*\|_{L_2(dp(x))},$$

(the first term is the **estimation error**, the second term is the **approximation error**).

In order to find the optimal D (to balance estimation and approximation error), we can minimize the bound with respect to D , leading to, using Lagrange duality:

$$\begin{aligned}
& \inf_{D \geq 0} \frac{2GRD}{\sqrt{n}} + G \inf_{\|f\|_{\mathcal{H}} \leq D} \|f - f^*\|_{L_2(dp)} \\
&= \inf_{D \geq 0} \frac{2GBD}{\sqrt{n}} + G \sup_{\lambda \geq 0} \inf_{f \in \mathcal{H}} \|f - f^*\|_{L_2(dp(x))} + \sqrt{\lambda}(\|f\|_{\mathcal{H}} - D) \text{ using duality,} \\
&\leq \sup_{D \geq 0} \inf_{D \geq 0} GD \left[\frac{2R}{\sqrt{n}} - \sqrt{\lambda} \right] + 2G \sqrt{\inf_{f \in \mathcal{H}} \{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|_{\mathcal{H}}^2 \}} \text{ using } a + b \leq 2\sqrt{a^2 + b^2}, \\
&= \sup_{\lambda \geq 0} G \sqrt{\inf_{f \in \mathcal{H}} \{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|_{\mathcal{H}}^2 \}} \text{ such that } \sqrt{\lambda} \leq \frac{2R}{\sqrt{n}} \text{ by solving with respect to } D, \\
&\leq 2G \sqrt{\inf_{f \in \mathcal{H}} \{ \|f - f^*\|_{L_2(dp(x))}^2 + \frac{4R^2}{n} \|f\|_{\mathcal{H}}^2 \}} \text{ with } \lambda^* = \frac{4R^2}{n}.
\end{aligned}$$

Note that the value $\lambda^* = \frac{4R^2}{n}$ is a priori not a regularization parameter to be used in an algorithm that would lead to the rate we are going to describe below. From such a λ^* , and the corresponding optimal f , the suggested D is $\|f\|_{\mathcal{H}}$ (as shown below, a good regularization parameter to achieve this bound is proportional to $1/\sqrt{n}$).

Overall, we need to understand how the deterministic quantity

$$A(\lambda, f^*) = \inf_{f \in \mathcal{H}} \{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|_{\mathcal{H}}^2 \}$$

goes to zero when λ goes to zero. A few situations are possible:

- If the target function f^* happens to be in \mathcal{H} , then $A(\lambda, f^*) = \lambda \|f^*\|_{\mathcal{H}}^2$, and thus it tends to zero as $O(\lambda)$. This is the best case scenario, and requires that the target function is sufficiently regular (with at least $d/2$ derivatives for $\mathcal{X} = \mathbb{R}^d$). Then, using it with $\lambda = \frac{4R^2}{n}$ above, the overall excess risk goes to zero as $O(1/\sqrt{n})$.
- The target function f^* is not in \mathcal{H} , but can be approached arbitrary closely in $L_2(dp(x))$ -norm by a function in \mathcal{H} ; in other words, f^* is in the closure of \mathcal{H} in $L_2(dp(x))$. In this situation, then $A(\lambda, f^*)$ goes to zero as λ goes to zero, but without an explicit rate if no further assumptions are made.

For $\mathcal{X} = \mathbb{R}^d$, and $dp(x)$ 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 $L_2(dx)$, so this case includes most potential functions. See Section 7.5.2 for explicit rates.

- Otherwise, denoting $\Pi_{\mathcal{H}}(f^*)$ the orthogonal projection in $L_2(dp(x))$ of f^* on the closure of \mathcal{H} , by Pythagoras theorem, $A(\lambda, f^*) = A(\lambda, \Pi_{\mathcal{H}}(f^*)) + \|f^* - \Pi_{\mathcal{H}}(f^*)\|_{L_2(dp(x))}^2$, that is, there is an incompressible error due to a choice of function space which is not large enough.

Regularized problem (♦). For the regularized problem, we can use the bound from Chapter 4:

$$\mathbb{E}[\mathcal{R}(\hat{f}_\lambda^{(r)})] - \mathcal{R}(f^*) \leq \frac{32G^2R^2}{\lambda n} + \inf_{f \in \mathcal{H}} \{ G \|f - f^*\|_{L_2(dp(x))} + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2 \}.$$

We can now minimize the bound with respect to λ as $\lambda^* = \frac{8RG}{\sqrt{n}}$, to obtain the bound:

$$G \inf_{f \in \mathcal{H}} \left\{ \|f - f^*\|_{L_2(dp(x))} + \frac{8R}{\sqrt{n}} \|f\|_{\mathcal{H}} \right\} \leq 2G \sqrt{\inf_{f \in \mathcal{H}} \left\{ \|f - f^*\|_{L_2(dp)}^2 + \frac{64R^2}{n} \|f\|_{\mathcal{H}}^2 \right\}},$$

which is the same bound as for constrained problem, but on a more commonly used optimization problem in practice. This also suggests to use a regularization parameter proportional to R^2/n .

7.5.2 Approximation error for translation-invariant kernels on \mathbb{R}^d

We first start with the analysis of the approximation error of kernel methods for translation invariant kernels. Given a distribution $dp(x)$, the goal is to compute

$$A(\lambda, f^*) = \inf_{f \in \mathcal{H}} \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|_{\mathcal{H}}^2,$$

where f^* is the target function (e.g., the minimizer of the test risk), which we assume squared-integrable. If $A(\lambda, f^*)$ tends to zero when λ tends to zero for any fixed f^* , then kernel-based supervised learning leads to universally consistent algorithms.

We assume that $\|f - f^*\|_{L_2(dp(x))}^2 \leq C \|f - f^*\|_{L_2(dx)}^2$ (e.g., with $C = \|dp/dx\|_{\infty}$ where dp/dx is the density of $dp(x)$). Moreover, for simplicity, we assume that $\|f^*\|_{L_2(dx)}$ is finite (that is, f^* is not allowed to explode at infinity). We now give bounds on

$$\tilde{A}(\lambda, f^*) = \inf_{f \in \mathcal{H}} \|f - f^*\|_{L_2(dx)}^2 + \lambda \|f\|_{\mathcal{H}}^2,$$

Remember from Section 7.5.1 that if $f^* \in \mathcal{H}$ (best case scenario), then $\tilde{A}(\lambda, f^*) = \lambda \|f^*\|_{\mathcal{H}}^2$.

Explicit approximation. We have, for translation-invariant kernels, $\|f\|_{\mathcal{H}}^2 = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} d\omega$, and thus

$$\tilde{A}(\lambda, f^*) = \inf_{\hat{f} \in L_2(d\omega)} \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left[|\hat{f}(\omega) - \hat{f}^*(\omega)|^2 + \lambda \frac{|\hat{f}(\omega)|^2}{\hat{q}(\omega)} \right] d\omega.$$

The optimization can be performed independently for each ω , and this is the quadratic problem, setting the derivative with respect to $\hat{f}(\omega)$ to zero leads to $0 = 2(\hat{f}(\omega) - \hat{f}^*(\omega)) + 2\lambda \frac{\hat{f}(\omega)}{\hat{q}(\omega)}$, and thus $\hat{f}_{\lambda}(\omega) = \frac{\hat{f}^*(\omega)}{1 + \lambda \hat{q}(\omega)^{-1}}$. In terms of objective function, we get:

$$\tilde{A}(\lambda, f^*) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left[|\hat{f}^*(\omega)|^2 \left(1 - \frac{1}{1 + \lambda \hat{q}(\omega)^{-1}} \right) \right] d\omega = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \left[|\hat{f}^*(\omega)|^2 \frac{\lambda}{\hat{q}(\omega) + \lambda} \right] d\omega.$$

When λ goes to zero, we see that for each ω , $\hat{f}_{\lambda}(\omega)$ tends to $\hat{f}(\omega)$. By the dominated convergence theorem, $\tilde{A}(\lambda, f^*)$ goes to zero, when λ goes to zero.

Without further assumptions it is not possible to obtain a rate of convergence (otherwise the no-free lunch theorem from Chapter 2 would be invalidated). However, this is possible when assuming regularity properties for f^* .

Sobolev spaces (♦). If we assume that

$$\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} (1 + \|\omega\|_2^2)^t |\hat{f}^*(\omega)|^2 d\omega \quad (7.6)$$

is finite for some $t > 0$, that is, for f^* with squared integrable partial derivatives up to order t , then we can further bound:

$$\tilde{A}(\lambda, f^*) \leq \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} (1 + \|\omega\|_2^2)^t |\hat{f}^*(\omega)|^2 d\omega \times \sup_{\omega \in \mathbb{R}^d} \left\{ \frac{\lambda}{\hat{q}(\omega) + \lambda} \frac{1}{(1 + \|\omega\|_2^2)^t} \right\}.$$

If we now assume $\hat{q}(\omega) \propto (1 + \|\omega\|_2^2)^{-s}$ (Matern kernels), with $s > d/2$ to get an RKHS, then with $t \geq s$, $f^* \in \mathcal{H}$, and have $\tilde{A}(\lambda, f^*) = \lambda \|f^*\|_{\mathcal{H}}^2$. With $t < s$, that is the function is not inside the RKHS \mathcal{H} , then we get a bound proportional to (using $a + b \geq a^{t/s} b^{1-t/s}$):

$$\sup_{\omega \in \mathbb{R}^d} \left\{ \frac{\lambda}{\hat{q}(\omega) + \lambda} \frac{1}{(1 + \|\omega\|_2^2)^t} \right\} \leq \sup_{\omega \in \mathbb{R}^d} \left\{ \frac{\lambda}{\hat{q}(\omega)^{t/s} \lambda^{1-t/s}} \frac{1}{(1 + \|\omega\|_2^2)^t} \right\} = O(\lambda^{t/s}).$$

Exercise 7.6 (♦) Find an upper-bound of $\tilde{A}(\lambda, f^*)$ for the same assumption on f^* but with the Gaussian kernel.



There are two regularities: $t \geq 0$ for the target function, and $s > d/2$ for the kernel.

Putting things together. Thus, for Lipschitz-continuous losses and target functions that satisfy Eq. (7.6), we get an expected excess risk of the order $\sqrt{\tilde{A}(R^2/n, f^*)} = O(\frac{1}{n^{t/(2s)}})$, when $t \leq s$. For example, when $t = 1$, that is only first order derivative are assumed to be squared integrable, then for $s = d/2 + 1/2$ (exponential kernel), we obtain a rate of $O(\frac{1}{n^{1/(d+1)}})$, which is similar to the rate obtained with local averaging techniques in Chapter 6 (note here that we are in Lipschitz-loss set-up, which leads to worse rates, see square loss in Section 7.6). Thus kernel methods do not escape the curse of dimensionality (which is unavoidable anyway). *But*, with the proper choice of regularization parameter, they can benefit from extra smoothness of the target function: in the very favorable case, where $f^* \in \mathcal{H}$, that is $t \geq s$, then we obtain a dimension independent rate of $1/\sqrt{n}$. In intermediate scenarios, the rates are inbetween. This is why kernel methods are said to be *adaptive to the smoothness* of the target function.

Approximation bounds (♦). In some analysis set-ups (like explored in Chapter 9), 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 $A(\lambda, f^*) = \inf_{f \in \mathcal{H}} \{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|_{\mathcal{H}}^2 \}$ of the form $c\lambda^\alpha$ for $\alpha \in (0, 1)$ leads to the following bound:

$$\begin{aligned} & \inf_{f \in \mathcal{H}} \|f\|_{\mathcal{H}}^2 \text{ such that } \|f - f^*\|_{L_2(dp(x))} \leq \varepsilon \\ &= \inf_{f \in \mathcal{H}} \sup_{\mu \geq 0} \|f\|_{\mathcal{H}}^2 + \mu (\|f - f^*\|_{L_2(dp(x))}^2 - \varepsilon^2) \text{ using Lagrangian duality,} \\ &= \sup_{\mu \geq 0} \mu A(\mu^{-1}, f^*) - \mu \varepsilon^2 \leq \sup_{\mu \geq 0} \mu c \mu^{-\alpha} - \mu \varepsilon^2. \end{aligned}$$

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 $\alpha = t/s$ like before, this leads to an RKHS norm proportional to $\varepsilon^{-(1-\alpha)/\alpha}$ to get an error less than $\|f - f^*\|_{L_2(dx)}$. So where $t = 1$ (single derivative for the target function), and $s > d/2$ (for the Sobolev kernel), we get a norm of the order $\varepsilon^{-(1/\alpha-1)} = \varepsilon^{-(s-1)} \geq \varepsilon^{-d/2+1}$, which explodes exponentially in dimension, which is another way of formulating the curse of dimensionality.

7.6 Theoretical analysis of ridge regression (◆◆)

In this section, we provide finer results for ridge regression used within kernel methods. Compared to the analysis performed in Section 3.6, there are three difficulties:

- (1) we go from fixed design to random design: this will require finer probabilistic arguments (similar to the ones in Section 3.8.2),
- (2) we need to go infinite-dimensional: in terms of notations, this will mean not using transposes of matrices, but dot-products, which is a minor modification,
- (3) the infimum of the expected risk over linear functions parameterized by $\theta \in \mathcal{H}$ may not be attained by an element of \mathcal{H} , but by an element of its closure in $L_2(dp(x))$. This is important, as this allows to access a potentially large set of functions, and requires more care.

7.6.1 Kernel ridge regression

Beyond fixed-design finite-dimensional analysis. In Chapter 3, we considered ridge regression in the fixed design setting (where the input data are assumed deterministic) and a finite-dimensional feature space \mathcal{H} , and obtained in Prop. 3.7 the following *exact* expression on the excess risk of the ridge regression estimator $\hat{\theta}_\lambda$, assuming $y_i = \langle \theta_*, \varphi(x_i) \rangle + \varepsilon_i$, with ε_i independent from x_i , and where $\mathbb{E}[\varepsilon_i^2] = \sigma^2$:

$$\mathbb{E}[(\hat{\theta}_\lambda - \theta_*)^\top \hat{\Sigma}(\hat{\theta}_\lambda - \theta_*)] = \lambda^2 \theta_*^\top (\hat{\Sigma} + \lambda I)^{-2} \hat{\Sigma} \theta_* + \frac{\sigma^2}{n} \text{tr} [\hat{\Sigma}^2 (\hat{\Sigma} + \lambda I)^{-2}]. \quad (7.7)$$

For the random design assumption (which is the usual machine learning setting), we first need to obtain a value for the expected risk. Moreover, in order to apply to infinite dimensional \mathcal{H} where the minimizer has potentially infinite norm, we need to replace the matrix notation.

Modeling assumptions. We assume that

$$y_i = f_*(x_i) + \varepsilon_i,$$

with for simplicity $\mathbb{E}(\varepsilon_i|x_i) = 0$, and $\mathbb{E}(\varepsilon_i^2|x_i) \leq \sigma^2$ almost surely, for some target function $f_* \in L_2(dp(x))$, so that $f^*(x) = \mathbb{E}[y|x]$ is exactly the conditional expectation of $y|x$. Moreover, for simplicity we will assume that $\|f_*\|_\infty$ is bounded, that is the target function is uniformly bounded.



The target function f^* may not be in \mathcal{H} . All dot-products will always be in \mathcal{H} , while for norms we will specify the corresponding space.

We thus consider the optimization problem:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2, \quad (7.8)$$

with solution found with algorithms in Section 7.4. We have, with $\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^n \varphi(x_i) \otimes \varphi(x_i)$ a self-adjoint operator from \mathcal{H} to \mathcal{H} (the empirical covariance operator), a cost function equal to

$$\frac{1}{n} \sum_{i=1}^n y_i^2 + \langle f, \widehat{\Sigma} f \rangle - 2 \left\langle \frac{1}{n} \sum_{i=1}^n y_i \varphi(x_i), \theta \right\rangle + \lambda \|f\|_{\mathcal{H}}^2,$$

leading to the minimizer \hat{f}_λ of Eq. (7.8) 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 excess risk equal to $\mathbb{E}[\|\hat{f}_\lambda - f^*\|_{L_2(dp(x))}^2]$ as (and using that $\mathbb{E}(\varepsilon_i | x_i) = 0$):

$$\begin{aligned} & \mathbb{E}[\|\hat{f}_\lambda - f^*\|_{L_2(dp(x))}^2] \\ = & \mathbb{E}\left[\left\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i)\right\|_{L_2(dp(x))}^2\right] + \mathbb{E}\left[\left\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n f^*(x_i) \varphi(x_i) - f^*\right\|_{L_2(dp(x))}^2\right]. \end{aligned}$$

The first term is the usual **variance** term (that depends on the noise on top of the optimal predictions), while the second is the **bias** term (which depends on the regularity of the target function). Before developing the probabilistic argument, we give a simplified upper-bounds of the two terms.

On top of the non-centered empirical covariance operator $\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^n \varphi(x_i) \otimes \varphi(x_i)$, we will need its expectation, the covariance operator (from \mathcal{H} to \mathcal{H})

$$\Sigma = \mathbb{E}[\varphi(x) \otimes \varphi(x)]$$

for the corresponding distribution of the x_i 's. A key property is that for $g \in \mathcal{H}$,

$$\|g\|_{L_2(dp(x))}^2 = \int_{\mathcal{X}} g(x)^2 dp(x) = \int_{\mathcal{X}} \langle g, \varphi(x) \rangle^2 dp(x) = \int_{\mathcal{X}} \langle g, \varphi(x) \otimes \varphi(x) g \rangle dp(x) = \langle g, \Sigma g \rangle = \|\Sigma^{1/2} g\|_{\mathcal{H}}^2.$$

Variance term. Starting from **variance** = $\mathbb{E}\left[\left\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i)\right\|_{L_2(dp(x))}^2\right]$, the variance term is less than (first using independence and zero means of the variables ε_i):

$$\begin{aligned} \mathbb{E}\left[\left\|(\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \varphi(x_i)\right\|_{L_2(dp(x))}^2\right] &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}\left[\text{tr}\left((\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} \varepsilon_i^2 \varphi(x_i) \otimes \varphi(x_i)\right)\right] \\ &\leq \frac{\sigma^2}{n} \mathbb{E}\left[\text{tr}\left((\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma}\right)\right] \text{ using } \mathbb{E}(\varepsilon_i | x_i) \leq \sigma^2, \\ &\leq \frac{\sigma^2}{n} \mathbb{E}\left[\text{tr}\left((\widehat{\Sigma} + \lambda I)^{-1} \Sigma\right)\right] \text{ using } (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} \preceq I. \end{aligned} \quad (7.9)$$

This will be the main expression we will bound later. We note that the quantity above, *before* the expectation is almost surely less than $\frac{\sigma^2}{n} \frac{R^2}{\lambda}$. This will be useful for the probabilistic argument.

Bias term. We first assume that $f_* \in \mathcal{H}$, that is, the model is well-specified. Then, writing $f_*(x_i) = \langle f_*, \varphi(x_i) \rangle$ (which is possible because $f_* \in \mathcal{H}$), the bias term is equal to

$$\begin{aligned} \text{bias} &= \mathbb{E} \left[\left\| (\widehat{\Sigma} + \lambda I)^{-1} \frac{1}{n} \sum_{i=1}^n \langle f_*, \varphi(x_i) \rangle \varphi(x_i) - f_* \right\|_{L_2(dp(x))}^2 \right] \\ &= \mathbb{E} \left[\left\| (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} f_* - f_* \right\|_{L_2(dp(x))}^2 \right] \\ &= \mathbb{E} \left[\left\| \lambda \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} f_* \right\|_{\mathcal{H}}^2 \right] = \lambda^2 \mathbb{E} \left[\langle f_*, (\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} f_* \rangle \right]. \end{aligned} \quad (7.10)$$

This will be the main expression we will bound later. Note that the expression above is only valid for $f_* \in \mathcal{H}$. We note that the quantity above, *before* the expectation is almost surely less than $2\|f_*\|_{L_2(dp(x))}^2 + 2\|(\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} f_*\|_{L_2(dp(x))}^2 \leq 2\|f_*\|_{L_2(dp(x))}^2 + 2\frac{R^2}{\lambda} \|\widehat{\Sigma}^{1/2} f_*\|_{\mathcal{H}}^2 \leq 2\|f_*\|_{L_2(dp(x))}^2 + 2\frac{R^2}{\lambda} \|f_*\|_{L_\infty(dp(x))}^2 \leq 2(1 + \frac{R^2}{\lambda}) \|f_*\|_{L_\infty(dp(x))}^2$. This will be useful for the probabilistic argument.

Given the expression of the expected variance in Eq. (7.9) and of the expected bias in Eq. (7.10), we notice that both the empirical and expected covariance operators appear, and that it would be important to replace the empirical one by the expected one. This is possible if λ is large enough, which we now show. Then we will bound the two terms separately and show how balancing them leads to interesting learning bounds.

7.6.2 Relationship between covariance operators

We first start with the following lemma relating $\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^n \varphi(x_i) \otimes \varphi(x_i)$ and $\Sigma = \mathbb{E}[\varphi(x) \otimes \varphi(x)]$, which are the non-centered empirical and population covariance operators. This concentration result relies on a dimension-independent version of matrix concentration inequalities presented in Section 1.2.5, which applies to operators [74, Eq. (3.9)]. It will allow to replace $\widehat{\Sigma}$ by Σ in many inequalities.

Lemma 7.1 (Concentration for covariance operators) *If $\|\varphi(x)\| \leq R$ almost surely, then for $n \geq \frac{5R^2}{\lambda}$, with probability greater than $1 - 14 \operatorname{tr}[(\Sigma + \lambda I)^{-1} \Sigma] \exp(-\frac{2n\lambda}{R^2})$, we have*

$$-\frac{1}{2}I \preceq (\Sigma + \lambda I)^{-1/2} (\Sigma - \widehat{\Sigma}) (\Sigma + \lambda I)^{-1/2} \preceq \frac{1}{2}I. \quad (7.11)$$

Proof Let $M_i = \frac{1}{n} (\Sigma + \lambda I)^{-1/2} (\varphi(x_i) \otimes \varphi(x_i) - \Sigma) (\Sigma + \lambda I)^{-1/2}$ be a self-adjoint operator from \mathcal{H} to \mathcal{H} . We have $\mathbb{E}[M_i] = 0$, $\|M_i\|_{\text{op}} \leq \frac{R^2}{\lambda}$, and

$$\begin{aligned} \mathbb{E}[M_i^2] &= \frac{1}{n^2} (\Sigma + \lambda I)^{-1/2} \mathbb{E}[\varphi(x_i) \otimes \varphi(x_i) (\Sigma + \lambda I)^{-1} \varphi(x_i) \otimes \varphi(x_i)] (\Sigma + \lambda I)^{-1/2} - \frac{1}{n^2} (\Sigma + \lambda I)^{-2} \Sigma^2 \\ &\preceq \frac{1}{n^2} \frac{R^2}{\lambda} (\Sigma + \lambda I)^{-1} \Sigma, \text{ by using } \langle \varphi(x_i), (\Sigma + \lambda I)^{-1} \varphi(x_i) \rangle \leq \frac{R^2}{\lambda}. \end{aligned}$$

Thus $\text{tr } \mathbb{E}[M_i^2] \leq \frac{1}{n^2} \frac{R^2}{\lambda} \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma]$, and $\mathbb{E}[M_i^2] \preceq \frac{1}{n^2} \frac{R^2}{\lambda} I$. Using the bound from [74, Eq. (3.9)], we get that

$$\begin{aligned} \mathbb{P}\left(\|(\Sigma + \lambda I)^{-1/2}(\Sigma - \widehat{\Sigma})(\Sigma + \lambda I)^{-1/2}\|_{\text{op}} > t\right) &= \mathbb{P}\left(\left\|\sum_{i=1}^n M_i\right\|_{\text{op}} > t\right) \\ &\leq 14 \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma] \exp\left(-\frac{t^2/2}{\frac{R^2}{n\lambda}(1+t/3)}\right), \end{aligned}$$

if $t^2 \geq \frac{R^2}{\lambda n}(1+t/3)$. With $t = 1/2$, it is sufficient that $n \geq 5 \frac{R^2}{\lambda} \geq \frac{R^2}{\lambda} \frac{1+1/6}{1/4}$, and $\delta \leq 14 \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma] \exp\left(-\frac{2n\lambda}{R^2}\right)$. This leads to the desired result. Note that this provides also interesting result when t is smaller, but we will not use it, since we want to obtain a result in expectation (but we could use it for results in high probability). ■

The inequality in Eq. (7.11) has some interesting consequences. First $(\Sigma + \lambda I)^{-1/2}(\Sigma - \widehat{\Sigma})(\Sigma + \lambda I)^{-1/2} \leq tI$ leads to $(\Sigma - \widehat{\Sigma}) \preceq t(\Sigma + \lambda I)$, and thus, $(1-t)(\Sigma - \widehat{\Sigma}) \preceq t(\widehat{\Sigma} + \lambda I)$, which leads to $(\widehat{\Sigma} + \lambda I)^{-1/2}(\Sigma - \widehat{\Sigma})(\widehat{\Sigma} + \lambda I)^{-1/2} \leq \frac{t}{1-t}I$ and also $\widehat{\Sigma} + \lambda I \succcurlyeq (1-t)(\Sigma + \lambda I)$.

7.6.3 Analysis for well-specified problems

In this section, we assume that $f_* \in \mathcal{H}$. We have the following result for the excess risk.

Proposition 7.1 (Convergence rate for 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 $\mathbb{E}(\varepsilon_i|x_i) = 0$ and $\mathbb{E}(\varepsilon_i^2|x_i) \leq \sigma^2$, and $f_* \in \mathcal{H}$. Assume $\|\varphi(x)\|_2 \leq R$ almost surely and $\lambda \leq R^2$. Then, if $n \geq \frac{5R^2}{\lambda}(1 + \log \frac{R^2}{\lambda})$, we have:*

$$\mathbb{E}[\mathcal{R}(\hat{f}_\lambda) - \mathcal{R}^*] \leq 16 \frac{\sigma^2}{n} \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma] + 16\lambda \langle f_*, (\Sigma + \lambda I)^{-1} \Sigma f_* \rangle + \frac{24}{n^2} \|f_*\|_{L_\infty(dp(x))}^2. \quad (7.12)$$

This is to be contrasted with Eq. (7.7): we obtain a similar result with $\widehat{\Sigma}$ replaced by Σ , but with some extra constants and an additional negligible term.

Proof for the variance term. (◆◆) We can bound the variance term from Eq. (7.9), using the event $\mathcal{A} = \left\{-\frac{1}{2}I \preceq (\Sigma + \lambda I)^{-1/2}(\Sigma - \widehat{\Sigma})(\Sigma + \lambda I)^{-1/2} \preceq \frac{1}{2}I\right\}$ from Lemma 7.1:

$$\begin{aligned} \text{variance} &= \frac{\sigma^2}{n} \mathbb{E}\left[\text{tr} \left[(\widehat{\Sigma} + \lambda I)^{-1} \Sigma\right]\right] \\ &\leq \frac{\sigma^2}{n} \mathbb{E}\left[1_{\mathcal{A}} \text{tr} \left[(\widehat{\Sigma} + \lambda I)^{-1} \Sigma\right]\right] + \frac{\sigma^2}{n} \mathbb{E}\left[1_{\mathcal{A}^c} \text{tr} \left[(\widehat{\Sigma} + \lambda I)^{-1} \Sigma\right]\right] \\ &\leq 2 \frac{\sigma^2}{n} \mathbb{E}\left[\text{tr} \left[(\Sigma + \lambda I)^{-1} \Sigma\right]\right] + \frac{\sigma^2}{n} \mathbb{P}(\mathcal{A}^c) \mathbb{E}\left[\text{tr} \left[(\Sigma + \lambda I)^{-1} \Sigma\right]\right] \\ &\leq 2 \frac{\sigma^2}{n} \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma] + \frac{\sigma^2}{n} \frac{R^2}{\lambda} 14 \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma] \exp\left(-\frac{2n\lambda}{R^2}\right) \\ &= 2 \frac{\sigma^2}{n} \text{tr} [(\Sigma + \lambda I)^{-1} \Sigma] \left(1 + 7 \frac{R^2}{\lambda} \exp\left(-\frac{2n\lambda}{R^2}\right)\right). \end{aligned}$$

We thus need that $\exp\left(\frac{2n\lambda}{R^2}\right) \geq \frac{R^2}{\lambda}$, that is, $n \geq \frac{R^2}{2\lambda} \log \frac{R^2}{\lambda}$ (which we have), to get the desired variance term less than $16 \frac{\sigma^2}{n} \text{tr}[(\Sigma + \lambda I)^{-1} \Sigma]$. \blacksquare

Proof for the bias term. (◆◆) We start from Eq. (7.10):

$$\text{bias} \leq \lambda^2 \mathbb{E} \left[\langle f_*, (\widehat{\Sigma} + \lambda I)^{-1} \Sigma (\widehat{\Sigma} + \lambda I)^{-1} f_* \rangle \right] = \mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} f_* - f_* \right\|^2 \right].$$

We can now introduce $f_\lambda = (\Sigma + \lambda I)^{-1} \Sigma f_*$ the smoothing of f_* (which is a deterministic function), and bound

$$\begin{aligned} \text{bias} &\leq 2\mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} f_* - f_\lambda \right\|_{\mathcal{H}}^2 \right] + 2\mathbb{E} \left[\left\| \Sigma^{1/2} (f_\lambda - f_*) \right\|_{\mathcal{H}}^2 \right] \\ &= 2\mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} f_* - f_\lambda \right\|_{\mathcal{H}}^2 \right] + 2\mathbb{E} \left[\left\| \lambda \Sigma^{1/2} (\Sigma + \lambda I)^{-1} f_* \right\|_{\mathcal{H}}^2 \right] \\ &\leq 4\mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} (f_* - f_\lambda) \right\|_{\mathcal{H}}^2 \right] + 4\mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^{-1} - I) f_\lambda \right\|_{\mathcal{H}}^2 \right] + 2\mathbb{E} \left[\left\| \lambda \Sigma^{1/2} (\Sigma + \lambda I)^{-1} f_* \right\|_{\mathcal{H}}^2 \right] \end{aligned}$$

The third term is simply

$$2\lambda^2 \langle f_*, \Sigma (\Sigma + \lambda I)^{-2} f_* \rangle \leq 2\lambda \langle f_*, \Sigma (\Sigma + \lambda I)^{-1} f_* \rangle. \quad (7.13)$$

For the second term, we have

$$\begin{aligned} 4\mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} (\widehat{\Sigma} + \lambda I)^{-1} - I) f_\lambda \right\|_{\mathcal{H}}^2 \right] &= 4\mathbb{E} \left[\left\| \lambda \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} f_\lambda \right\|_{\mathcal{H}}^2 \right] \\ &\leq 4\lambda^2 \mathbb{E} \left[\left\| \Sigma^{1/2} (\Sigma + \lambda I)^{-1/2} (\Sigma + \lambda I)^{1/2} (\widehat{\Sigma} + \lambda I)^{-1/2} (\widehat{\Sigma} + \lambda I)^{-1/2} f_\lambda \right\|_{\mathcal{H}}^2 \right] \\ &\leq 4\lambda \mathbb{E} \left[\left\| (\Sigma + \lambda I)^{1/2} (\widehat{\Sigma} + \lambda I)^{-1/2} \right\|_{\text{op}}^2 \right] \|f_\lambda\|_{\mathcal{H}}^2 \\ &= 4\mathbb{E} \left[\left\| (\Sigma + \lambda I)^{1/2} (\widehat{\Sigma} + \lambda I)^{-1/2} \right\|_{\text{op}}^2 \right] \times \lambda \langle f_*, (\Sigma + \lambda I)^{-2} \Sigma^2 f_* \rangle. \quad (7.14) \end{aligned}$$

For the first term, we have:

$$\begin{aligned} &4\mathbb{E} \left[\left\| \Sigma^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} \widehat{\Sigma} (f_* - f_\lambda) \right\|^2 \right] \\ &\leq 4\mathbb{E} \left[\left\| \Sigma^{1/2} (\Sigma + \lambda I)^{-1/2} (\Sigma + \lambda I)^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} (\Sigma + \lambda I)^{1/2} (\Sigma + \lambda I)^{-1/2} \widehat{\Sigma} (f_* - f_\lambda) \right\|^2 \right] \\ &\leq 4\mathbb{E} \left[\left\| (\Sigma + \lambda I)^{1/2} (\widehat{\Sigma} + \lambda I)^{-1} (\Sigma + \lambda I)^{1/2} \right\|_{\text{op}}^2 \cdot \left\| (\Sigma + \lambda I)^{-1/2} \widehat{\Sigma} (f_* - f_\lambda) \right\|^2 \right]. \quad (7.15) \end{aligned}$$

We can apply the same reasoning as for the variance term and introduce the event \mathcal{A} , which leads to a bound (when \mathcal{A} is true, from Eq. (7.13), Eq. (7.14) and Eq. (7.15)):

$$2\lambda \langle f_*, \Sigma (\Sigma + \lambda I)^{-1} f_* \rangle + 8\lambda \langle f_*, (\Sigma + \lambda I)^{-2} \Sigma^2 f_* \rangle + 16\mathbb{E} \left[\left\| (\Sigma + \lambda I)^{-1/2} \widehat{\Sigma} (f_* - f_\lambda) \right\|^2 \right].$$

For the last term, we can use

$$\begin{aligned}
\mathbb{E}[\widehat{\Sigma}(\Sigma + \lambda I)^{-1}\widehat{\Sigma}] &= \frac{1}{n^2} \sum_{i,j=1}^n \mathbb{E}[\varphi(x_i) \otimes \varphi(x_i)(\Sigma + \lambda I)^{-1}\varphi(x_j) \otimes \varphi(x_j)] \\
&= \frac{1}{n^2} \sum_{i \neq j}^n \Sigma(\Sigma + \lambda I)^{-1}\Sigma + \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}[\varphi(x_i) \otimes \varphi(x_i)(\Sigma + \lambda I)^{-1}\varphi(x_i) \otimes \varphi(x_i)] \\
&\preceq \Sigma(\Sigma + \lambda I)^{-1}\Sigma + \frac{R^2}{n}\Sigma,
\end{aligned}$$

to get the bound

$$\begin{aligned}
&10\lambda \langle f_*, (\Sigma + \lambda I)^{-1}\Sigma f_* \rangle + 16 \left[\lambda^2 \langle f_*, (\Sigma + \lambda I)^{-3}\Sigma^2 f_* \rangle + \frac{\lambda R^2}{n} \langle f_*, (\Sigma + \lambda I)^{-2}\Sigma f_* \rangle \right] \\
&\leq \lambda \langle f_*, (\Sigma + \lambda I)^{-1}\Sigma f_* \rangle \left(10 + 16 \frac{R^2}{\lambda n} \right) \leq 16\lambda \langle f_*, (\Sigma + \lambda I)^{-1}\Sigma f_* \rangle \text{ using the constraint on } n.
\end{aligned}$$

We can now compute the term coming from $\mathbb{P}(\mathcal{A}^c)$, which is less than

$$\begin{aligned}
4 \frac{R^2}{\lambda} \|f_*\|_{L_\infty(dp(x))}^2 \times 14 \frac{R^2}{\lambda} \exp\left(-\frac{2n\lambda}{R^2}\right) &= 4 \frac{R^2}{\lambda} \|f_*\|_{L_\infty(dp(x))}^2 \times 14 \frac{R^2}{\lambda} \exp\left(-\frac{4n\lambda}{5R^2} - \frac{6n\lambda}{5R^2}\right) \\
&\leq 4 \frac{R^2}{\lambda} \|f_*\|_{L_\infty(dp(x))}^2 \times 14 \frac{R^2}{\lambda} \left(\frac{\lambda}{R^2}\right)^4 \left(\frac{5R^2}{6n\lambda}\right)^2 \max_{\alpha} \alpha^2 e^{-\alpha} \\
&\leq \frac{24}{n^2} \|f_*\|_{L_\infty(dp(x))}^2.
\end{aligned}$$

■

Before analyzing the last proposition, and balancing bias and variance, we show how this can be applied beyond well-specified models.

7.6.4 Analysis beyond well-specified problems

In the bound in Eq. (7.12), the only term that requires potentially that $f_* \in \mathcal{H}$ is the bias term $16\lambda \langle f_*, (\Sigma + \lambda I)^{-1}\Sigma f_* \rangle$. The key to an extension to all potential functions f_* is the following simple lemma.

Lemma 7.2 *Given the covariance operator Σ and any function $f_* \in \mathcal{H}$, then*

$$\lambda \langle f_*, (\Sigma + \lambda I)^{-1}\Sigma f_* \rangle = \inf_{f \in \mathcal{H}} \left\{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|^2 \right\}.$$

Proof The optimization problem above can be written as $\inf_{f \in \mathcal{H}} \left\{ \|\Sigma^{1/2}(f - f^*)\|^2 + \lambda \|f\|^2 \right\}$, 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, this shows we can extend Prop. 7.1 to the general case of $f^* \in L_2(dp(x))$, but in the closure of \mathcal{H} in $L_2(dp(x))$ (because all functions in the closure can be approached by a function in \mathcal{H}). For translation-invariant kernels in \mathbb{R}^d (which are dense in $L_2(dx)$), this will allow to estimate any target function. We will also give below a more general result when f_* is not in the closure of \mathcal{H} .

Proposition 7.2 (Convergence rate for kernel ridge regression - mis-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 $\mathbb{E}(\varepsilon_i | x_i) = 0$ and $\mathbb{E}(\varepsilon_i^2 | x_i) \leq \sigma^2$, and f_* in the closure of \mathcal{H} . Assume $\|\varphi(x)\|_2 \leq R$ almost surely and $\lambda \leq R^2$. Then, if $n \geq \frac{5R^2}{\lambda}(1 + \log \frac{R^2}{\lambda})$, we have:*

$$\mathbb{E}[\mathcal{R}(\hat{f}_\lambda) - \mathcal{R}^*] \leq 16 \frac{\sigma^2}{n} \text{tr}[(\Sigma + \lambda I)^{-1} \Sigma] + 16 \inf_{f \in \mathcal{H}} \left\{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|^2 \right\} + \frac{24}{n^2} \|f_*\|_{L_\infty(dp(x))}^2. \quad (7.16)$$



Be careful with homogeneity.

General case. if f_* is not in the closure, we denote by $f_*^{\mathcal{H}}$ the projection of f_* for the $L_2(dp(x))$ -norm onto the closure of \mathcal{H} . The result from Eq. (7.16) has to be updated to

$$\mathbb{E}[\mathcal{R}(\hat{f}_\lambda) - \mathcal{R}(f_*^{\mathcal{H}})] \leq 16 \frac{\sigma^2}{n} \text{tr}[(\Sigma + \lambda I)^{-1} \Sigma] + 16 \inf_{f \in \mathcal{H}} \left\{ \|f - f_*^{\mathcal{H}}\|_{L_2(dp(x))}^2 + \lambda \|f\|^2 \right\} + \frac{24}{n^2} \|f_*\|_{L_\infty(dp(x))}^2.$$

Since for $f \in \mathcal{H}$, by Pythagoreas' theorem, $\|f - f_*\|_{L_2(dp(x))}^2 = \|f - f_*^{\mathcal{H}}\|_{L_2(dp(x))}^2 + \mathcal{R}(f_*^{\mathcal{H}}) - \mathcal{R}^*$, the equation above implies Eq. (7.16), which remains true in all situations.

7.6.5 Balancing bias and variance (◆◆)

We can now balance the bias and variance term in the following upper-bound on the expected excess risk, valid if $n \geq \frac{5R^2}{\lambda}(1 + \log \frac{R^2}{\lambda})$:

$$16 \frac{\sigma^2}{n} \text{tr}[(\Sigma + \lambda I)^{-1} \Sigma] + 16 \inf_{f \in \mathcal{H}} \left\{ \|f - f^*\|_{L_2(dp(x))}^2 + \lambda \|f\|^2 \right\},$$

plus negligible terms.

For this section, we will assume that $\mathcal{X} = \mathbb{R}^d$, and that the target function belongs to a Sobolev kernel of order $t > 0$, while the RKHS is a Sobolev space of order $s > d/2$.

We have seen in Section 7.5.2 that the bias term is of order $\lambda^{t/s}$ when $s \geq t$. For the variance term, we need to study the so-called “degrees of freedom”.

Degrees of freedom. This is the quantity $\text{tr}[\Sigma(\Sigma + \lambda I)^{-1}]$, which is decreasing in λ , from $+\infty$ for $\lambda = 0$ to 0 for $\lambda = +\infty$. If we know that the eigenvalues $(\lambda_m)_{m \geq 0}$ of the covariance operator satisfy

$$\lambda_m \leq C(m+1)^{-\alpha},$$

for $\alpha > 1$, then one has:

$$\begin{aligned} \text{tr} [\Sigma(\Sigma + \lambda I)^{-1}] &= \sum_{m \geq 0} \frac{\lambda_m}{\lambda_m + \lambda} \leq \sum_{m \geq 0} \frac{1}{1 + \lambda C^{-1}(m+1)^\alpha} \leq \int_0^\infty \frac{1}{1 + \lambda C^{-1}t^\alpha} \\ &\leq \int_0^\infty \lambda^{-1/\alpha} C^{1/\alpha} \frac{1}{\alpha} u^{1/\alpha-1} \frac{du}{1+u} \text{ with the change of variable } u = \lambda C^{-1}t^\alpha, \\ &\leq O(\lambda^{-1/\alpha}). \end{aligned}$$

It turns out that if $dp(x)$ has a bounded density with respect to the Lebesgue measure, then for our chosen Sobolev space, then we have $\alpha = 2s/d$ (see, e.g., [53, Appendix D]).

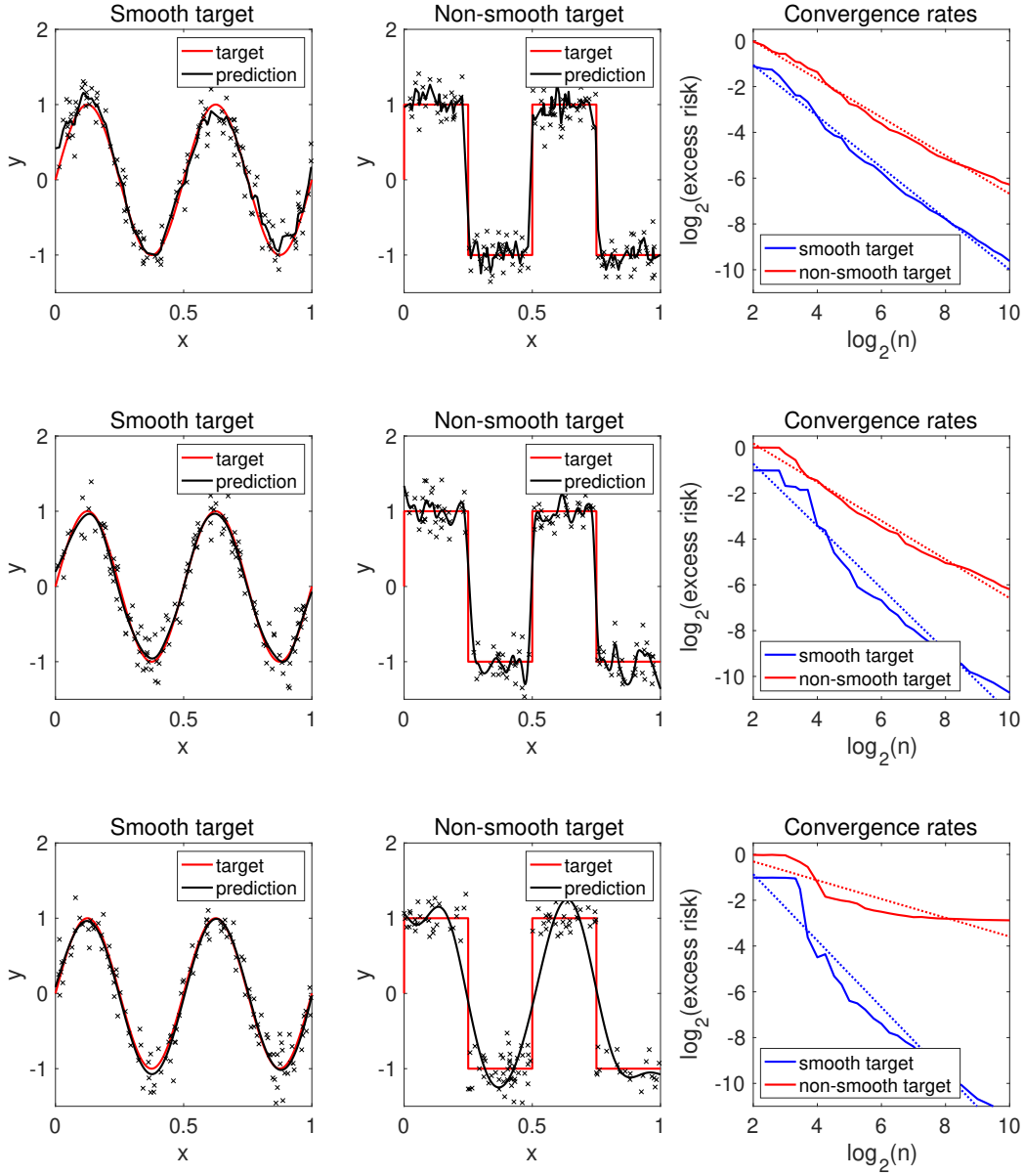
Balancing terms (Sobolev spaces). We thus need to balance $\lambda^{t/s}$ with $\frac{1}{n}\lambda^{-1/\alpha}$, leading to an optimal λ proportional to $n^{-(1/\alpha+t/s)^{-1}}$, and a rate proportional to $\frac{1}{n^{\alpha t/(\alpha t+s)}}$. This rate is only achievable through our analysis when the bound $n \geq \frac{5R^2}{\lambda}(1 + \log \frac{R^2}{\lambda})$ is true, that is, up to logarithmic terms, $\lambda \geq R^2/n$, thus, $\frac{1}{\alpha} + \frac{t}{s} \geq 1$.

For $\alpha = 2s/d$, we obtain the rate $\frac{1}{n^{2t/(2t+d)}}$, which is valid as long as $\frac{d}{2} + t \geq s \geq t$. We can make the following observations:

- Except for the constraint $\frac{d}{2} + t \geq s \geq t$, the upper-bound on the rate obtained after optimizing over λ does not depend on the kernel.
- We obtain some form of adaptivity, that is, the rate improves with the regularity of the target function, from the slow rate $\frac{1}{n^{2/(2+d)}}$ when $t = 1$ (recovering the same rate as for local averaging methods in Chapter 6, and can only be achieved when $s \leq d/2 + 1$, e.g., with the exponential kernel), to the rate $\frac{1}{n^{2s/(2s+d)}}$ when $t = s$, the rate is then always better than $1/\sqrt{n}$ because of the constraint $s > d/2$.
- In order to allow for regularization parameters λ which are less than $1/n$, other assumptions are needed. See, e.g., [88] 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 non-smooth target, and three kernels: exponential kernel corresponding to the Sobolev space of order 1 (top), Matern kernel corresponding to the Sobolev space of order 3 (middle), and Gaussian kernel (bottom). In the right plots, dotted lines are affine fits to the log-log learning curves. The regularization parameter for ridge regression is selected to minimize expected risk, and learning curves are obtained by averaging over 20 replications.



We observe adaptivity for the three kernels: learning is possible even with irregular function, and the rates are better for the smooth target function. We also note that for kernels with smaller feature spaces (Matern and Gaussian), the performance on the non-smooth target function is worse than for the large feature space (exponential kernel). As highlighted in [9], this drop in performance is mostly due to a numerical issue (the eigenvalues of the kernel matrix decay exponentially fast, and finite precision arithmetic prevents the use of regularization parameters which are too small).

Chapter 8

Sparse methods

Chapter summary

$-\ell_0$ penalty: For linear regression, if the optimal predictor has k non-zeros, then we can replace the rate $\sigma^2 d/n$ by $\sigma^2 k \log(d)/n$ with an ℓ_0 -penalty on the square loss (which is computationally hard).
 $-\ell_1$ penalty: With few assumptions, we can get a slow rate proportional to $\sqrt{\log(d)/n}$ with an ℓ_1 -penalty and efficient algorithms, while fast rates require very strong assumptions on the design matrix.

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 $\mathcal{X} = \mathbb{R}^d$, we saw that the excess risk for k -nearest-neighbors, 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 was scaling as d/n .

In these two situations, when d is large (of course much larger in the linear case), efficient learning is not possible in general.

In order to improve upon these rates, we study two techniques in this course. The first one is regularization, e.g., by the ℓ_2 -norm, that allows to obtain dimension-independent bounds that cannot improve over the bounds above in the worst-case, but are typically adaptive to additional regularity (see Chapter 3 and Chapter 7).

In this chapter, we consider another framework, namely *variable selection*, whose aim is to build predictors that depend only on a small number of variables. The key difficulty is that the identity of the selected variables is not known in advance.

In practice, variable selection is used in mainly two ways:

- The original set of features is already large (for example in text or web data).
- Given some input $x \in \mathcal{X}$, a large-dimensional feature vector $\varphi(x)$ is built where features are added that could potentially help predicting the response, but from which we expect only a small number to be relevant.

⚠ If no good predictor with small number of active variables exists, these methods are not supposed to work better.

In this chapter, we focus on linear methods, where we assume that we have a feature vector $\varphi(x) \in \mathbb{R}^d$, and we aim to minimize

$$\mathbb{E}[\ell(y, \varphi(x)^\top \theta)]$$

with respect to $\theta \in \mathbb{R}^d$, for some loss function $\ell : \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}$. We will consider two variable selection techniques, namely the penalization by $\|\theta\|_0$ the number of non-zeros in θ (often called abusively the “ ℓ_0 -norm”), or the ℓ_1 -norm.

Main focus on least-squares. These two types of penalties can be applied to all losses, but in this chapter, for simplicity we will mostly consider the square loss, and in most cases, the fixed design setting (see the classical set-up in Chapter 3), and assume that we have n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, such that there exists $\theta_* \in \mathbb{R}^d$ for which for $i \in \{1, \dots, n\}$,

$$y_i = \varphi(x_i)^\top \theta_* + \varepsilon_i,$$

where x_i is assumed deterministic, and ε_i has zero mean and variance σ^2 (we also assume independence from x_i , and sometimes stronger regularity, such as bounded almost surely, or Gaussian). The goal is then to find $\theta \in \mathbb{R}^d$, such that

$$\frac{1}{n} \|\Phi(\theta - \theta_*)\|_2^2 = (\theta - \theta_*)^\top \widehat{\Sigma} (\theta - \theta_*)$$

is as small as possible, where $\Phi \in \mathbb{R}^{n \times d}$ is the design matrix and $\widehat{\Sigma} = \frac{1}{n} \Phi^\top \Phi$ the non-centered empirical covariance matrix. We recall from Chapter 3 that for the ordinary least-squares estimator, the expectation of this excess risk is less than $\sigma^2 d/n$. This is the best possible performance if we make no assumption on θ_* . In this chapter, we assume that θ_* is sparse, that is, only a few of its components are non-zero, or in other words, $\|\theta_*\|_0 = k$ is small compared to d .

8.1.1 Dedicated proof technique for constrained least-squares

In this chapter, we consider a more refined proof technique¹ that can extend to constrained versions of least-squares (while our technique in Chapter 3 heavily relies on having a closed form for the estimator, which is not possible in constrained or regularized cases except in few instances, such as ridge regression).

We denote by $\hat{\theta}$ a minimizer of $\frac{1}{n} \|y - \Phi\theta\|_2^2$ with the constraint that $\theta \in \Theta$. If $\theta_* \in \Theta$, then we have, by optimality of $\hat{\theta}$:

$$\|y - \Phi\hat{\theta}\|_2^2 \leq \|y - \Phi\theta_*\|_2^2.$$

¹Taken from Philippe Rigollet’s lecture notes, see <http://www-math.mit.edu/~rigollet/PDFs/RigNotes17.pdf>. See also [91] for an example of application.

By expanding with $y = \Phi\theta_* + \varepsilon$, we get $\|\varepsilon - \Phi(\hat{\theta} - \theta_*)\|_2^2 \leq \|\varepsilon\|_2^2$, leading to, by expanding the norms:

$$\|\varepsilon\|_2^2 - 2\varepsilon^\top \Phi(\hat{\theta} - \theta_*) + \|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq \|\varepsilon\|_2^2,$$

and thus

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 2\varepsilon^\top \Phi(\hat{\theta} - \theta_*).$$

We can write it as

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 2\|\Phi(\hat{\theta} - \theta_*)\|_2 \cdot \varepsilon^\top \left(\frac{\Phi(\hat{\theta} - \theta_*)}{\|\Phi(\hat{\theta} - \theta_*)\|_2} \right).$$

This reformulation is difficult to deal with because $\hat{\theta}$ also appears on the right side of the equation. Like done for upper-bounding estimation errors in Chapter 4, we can maximize with respect to $\theta \in \Theta$, which leads to

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 2\|\Phi(\hat{\theta} - \theta_*)\|_2 \cdot \sup_{\theta \in \Theta} \varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right),$$

and finally

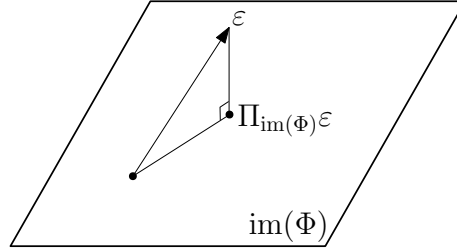
$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 4 \sup_{\theta \in \Theta} \left[\varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right) \right]^2. \quad (8.1)$$

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 quadratic forms in ε .

For example, when $\Theta = \mathbb{R}^d$ (no constraints), we get, by taking $z = \frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2}$, with $\Pi_\Phi = \Pi_{\text{im}(\Phi)}$ the orthogonal projector on the image space $\text{im}(\Phi)$:

$$\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 4\mathbb{E} \left[\sup_{z \in \text{im}(\Phi), \|z\|_2=1} [\varepsilon^\top z]^2 \right].$$

By a simple geometric argument (see below)



we have

$$\sup_{z \in \text{im}(\Phi), \|z\|_2=1} [\varepsilon^\top z]^2 = \sup_{z \in \text{im}(\Phi), \|z\|_2=1} [(\Pi_\Phi \varepsilon)^\top z + (\varepsilon - \Pi_\Phi \varepsilon)^\top z]^2 = \sup_{z \in \text{im}(\Phi), \|z\|_2=1} [(\Pi_\Phi \varepsilon)^\top z]^2 = \|\Pi_\Phi \varepsilon\|^2,$$

leading to

$$\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 4\mathbb{E}[\|\Pi_\Phi \varepsilon\|^2] = 4\sigma^2 \mathbb{E} \text{tr}(\Pi_\Phi^2) = 4\sigma^2 \text{rank}(\Phi).$$

We thus, get up to a constant 4, the excess risk as $\sigma^2 d/n$, which is worse than the direct computation from Chapter 3, but allows extensions to more complex situations.

This reasoning also allows to get high probability bounds by adding assumptions on the noise ε . Finally, this also extends to penalized problems (see Section 8.2.2).

8.1.2 Probabilistic and combinatorial lemmas

We start with two small probabilistic lemmas:

Lemma 8.1 *If $z \in \mathbb{R}^n$ is normally distributed with mean 0 and covariance matrix $\sigma^2 I$, then, if $s < \frac{1}{2\sigma^2}$, $\mathbb{E}[e^{s\|z\|_2^2}] = (1 - 2\sigma^2 s)^{-n/2}$.*

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):

$$\begin{aligned} \mathbb{E}[e^{s\|z\|_2^2}] &= \mathbb{E}[e^{s\sum_{i=1}^n z_i^2}] = \prod_{i=1}^n \mathbb{E}[e^{s z_i^2}] = \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^n \int_{-\infty}^{\infty} e^{(s-\frac{1}{2})z_i^2} dz_i \\ &= \frac{1}{(2\pi)^{n/2}} \prod_{i=1}^n \sqrt{2\pi} (1 - 2s)^{-1/2} = (1 - 2s)^{-n/2}. \end{aligned}$$

■

Lemma 8.2 *Let u_1, \dots, u_m be m random variables which are **potentially dependent**, and $s > 0, v > 0$ such that for each $i \in \{1, \dots, m\}$, $\mathbb{E}[e^{s u_i}] \leq v$. Then, $\mathbb{E}[\max\{u_1, \dots, u_m\}] \leq \frac{1}{s} \log(mv)$.*

Proof Following the reasoning from Section 1.2.4 in Chapter 1, for any $s \in \mathbb{R}$,

$$\mathbb{E}[\max\{u_1, \dots, u_m\}] \leq \frac{1}{s} \log \left(\sum_{i=1}^m \mathbb{E}[e^{s u_i}] \right) \leq \frac{1}{s} \log(mv).$$

■

The previous two lemmas can be combined to upper-bound the expectation of squared norms of Gaussian random variables: if $z_1, \dots, z_m \in \mathbb{R}^n$ are centered (that is, zero-mean) Gaussian random vectors which are potentially dependent, but for which the covariance matrix of z_i has eigenvalues less than σ^2 , we have for $s = \frac{1}{4\sigma^2}$, and Lemma 8.1, $\mathbb{E}[e^{s\|z\|_2^2}] \leq 2^{n/2}$, and from Lemma 8.2,

$$\mathbb{E}[\max\{\|z_1\|_2^2, \dots, \|z_m\|_2^2\}] \leq 4\sigma^2 \log(m2^{n/2}) = 2n\sigma^2 \log(2) + 4\sigma^2 \log(m),$$

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 extra term in $\sigma^2 k \log(d)$ for methods based on the ℓ_0 -penalty. The term in d^k comes from the following lemma.

Lemma 8.3 *Let $d > 0$ and $k \in \{1, \dots, d\}$. Then $\log \binom{d}{k} \leq k(1 + \log \frac{d}{k})$.*

Proof By recursion on k , the inequality is trivial for $k = 1$, and if $\binom{d}{k-1} \leq (\frac{ed}{k-1})^{k-1}$, then

$$\binom{d}{k} = \binom{d}{k-1} \frac{d-k+1}{k} \leq \left(\frac{ed}{k-1}\right)^{k-1} \frac{d}{k} \leq \left(\frac{ed}{k}\right)^{k-1} \left(1 + \frac{1}{k-1}\right)^{k-1} \frac{d}{k} \leq \left(\frac{ed}{k}\right)^{k-1} e \frac{d}{k} = \left(\frac{ed}{k}\right)^k,$$

where we use for $\alpha > 0$, $(1 + \frac{1}{\alpha})^\alpha = \exp(\alpha \log(1 + \frac{1}{\alpha})) \leq \exp(1) = e$. ■

We now consider two types of variable selection frameworks, one based on ℓ_0 -penalties, one based on ℓ_1 -penalties.

8.2 Variable selection by ℓ_0 penalty

In this section, we assume that the target vector θ_* has k non-zero components, that is, $\|\theta_*\|_0 = k$. We denote by $A = \text{supp}(\theta_*)$ the “support” of θ_* , that is, the subset of $\{1, \dots, d\}$ composed of j such that $(\theta_*)_j \neq 0$. We have $|A| = k$.

Price of adaptivity. If we knew the set A , then we could simply perform least-squares with the design matrix $\Phi_A \in \mathbb{R}^{n \times |A|}$, where Φ_B denotes the sub-matrix of Φ obtained by keeping only the columns from B , with an excess risk proportional to $\sigma^2 k/n$ (this is what we called the “oracle” in Section 8.4). Thus, as 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 have to estimate it. We will see that this will lead to an extra factor of $\log\left(\frac{d}{k}\right) \leq \log d$, due to the potentially large number of models with k variables.

8.2.1 Assuming k is known

We start by assuming that the cardinality k is known in advance, and we consider Gaussian noise for simplicity (this extends to sub-Gaussian noise as well, see note below).

Proposition 8.1 (Model selection - known k) *Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance σ^2 , with $\|\theta_*\|_0 \leq k$, for $k \leq d/2$. Let $\hat{\theta}$ be the minimizer of $\|y - \Phi\theta\|_2^2$ with the constraint that $\|\theta\|_0 \leq k$. Then, the (fixed design) excess risk is:*

$$\mathbb{E}[(\hat{\theta} - \theta_*)^\top \widehat{\Sigma}(\hat{\theta} - \theta_*)] = \mathbb{E}\left[\frac{1}{n} \|\Phi(\hat{\theta} - \theta_*)\|_2^2\right] \leq 32\sigma^2 \frac{k}{n} \left(\log\left(\frac{d}{k}\right) + 1 \right).$$

Proof Starting from Eq. (8.1), we see that for any θ such that $\|\theta\|_0 \leq k$, we have $\|\theta - \theta_*\|_0 \leq 2k$, and thus we have, from Section 8.1.1:

$$\begin{aligned}
\|\Phi(\hat{\theta} - \theta_*)\|_2^2 &\leq 4 \sup_{\theta \in \mathbb{R}^d, \|\theta\|_0 \leq k} \left[\varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right) \right]^2 \text{ from Eq. (8.1),} \\
&\leq 4 \sup_{\theta \in \mathbb{R}^d, \|\theta - \theta_*\|_0 \leq 2k} \left[\varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right) \right]^2 \text{ from the discussion above,} \\
&= 4 \sup_{B \subset \{1, \dots, n\}, |B| \leq 2k} 4 \sup_{\text{supp}(\theta - \theta_*) = B} \left[\varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right) \right]^2 \text{ by separating by supports,} \\
&\leq 4 \sup_{B \subset \{1, \dots, n\}, |B| \leq 2k} \sup_{z \in \text{im}(\Phi_B), \|z\|_2 = 1} [\varepsilon^\top z]^2 \\
&\leq 4 \sup_{B \subset \{1, \dots, n\}, |B| \leq 2k} \|\Pi_{\Phi_B} \varepsilon\|^2 \text{ using the same argument as in Section 8.1.1,} \\
&\leq 4 \sup_{B \subset \{1, \dots, n\}, |B| = 2k} \|\Pi_{\Phi_B} \varepsilon\|^2,
\end{aligned}$$

because $\|\Pi_{\Phi_B} \varepsilon\|^2$ is non-decreasing in B .

The random variable $\|\Pi_{\Phi_B} \varepsilon\|^2$ has an expectation which is less than $2k$, given that there are $\binom{d}{2k} \leq \left(\frac{ed}{2k}\right)^{2k}$ sets B of cardinality $2k$ (bound from Lemma 8.3), we should expect, with concentration inequalities from Section 8.1.2, that we pay a price of $\log \left(\frac{ed}{2k}\right)^{2k} \approx k \log \frac{d}{k}$. We will make this reasoning formal.

Indeed, $\Pi_{\Phi_B} \varepsilon$ is normally distributed with isotropic covariance matrix of dimension $|B| \leq 2k$, and thus we have for $s\sigma^2 < 1/2$ small enough, from Lemma 8.1:

$$\mathbb{E}[e^{s\|\Pi_{\Phi_B} \varepsilon\|^2}] \leq (1 - 2\sigma^2 s)^{-k}.$$

Thus, with $s = 1/(4\sigma^2)$, for which $(1 - 2\sigma^2 s)^{-k} = 2^k$, we get, from Lemma 8.2:

$$\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 16\sigma^2 \log \left(\binom{d}{2k} 2^k \right) \leq 16\sigma^2 \log \left(\left(\frac{ed}{2k}\right)^{2k} 2^k \right) = 16\sigma^2 \left(2k \log \left(\frac{d}{k}\right) + (2 - \log 2)k \right).$$

This leads to the desired result. ■

We can make the following observations:

- The result extends beyond Gaussian noise, that is, for all sub-Gaussian ε_i , for which $\mathbb{E}[e^{s\varepsilon_i}] \leq e^{s^2\tau^2}$ 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 minimisation is only done approximately.
- This result is not improvable by any algorithm (polynomial time or not), see, e.g., [45, Theorem 2.3].

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 when such fast rates are available for ℓ_1 -regularization (see Section 8.3.3).

- **Greedy algorithm:** starting from the empty set, variables are added one by one that maximizing the resulting cost reduction. This is often referred to as orthogonal matching pursuit.
- **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):

$$\frac{1}{n}\|y - \Phi\theta_{t-1}\|_2^2 - \frac{2}{n}(y - \Phi\theta_{t-1})^\top \Phi(\theta - \theta_{t-1}) + \lambda_{\max}(\frac{1}{n}\Phi^\top\Phi)\|\theta - \theta_{t-1}\|_2^2$$

on the cost function $\frac{1}{n}\|y - \Phi\theta\|_2^2$ is built and minimized with respect to $\|\theta\|_0 \leq k$ to obtain θ_t , which is done (checked as an exercise) by computing the unconstrained minimizer $\theta_{t-1} + \frac{1}{\lambda_{\max}(\frac{1}{n}\Phi^\top\Phi)} \frac{1}{n}\Phi^\top(y - \Phi\theta_{t-1})$, and selecting the k largest components.

8.2.2 Estimating k (♦)

In practice, regardless of the computational cost, one also needs to estimate k . A classical idea to consider penalized least-squares and minimize

$$\frac{1}{n}\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_0. \quad (8.2)$$

This is known to be a hard problem to solve, which essentially requires to look 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 $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance σ^2 , with $\|\theta_*\|_0 \leq k$. Let $\hat{\theta}$ be the minimizer of Eq. (8.2). Then, for $\lambda = \frac{2\sigma^2}{n}(3 + 2\log d)$, we have:*

$$\mathbb{E}\left[\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2\right] \leq \frac{16\sigma^2 k}{n}(3 + 2\log d) + \frac{5\sigma^2}{n}.$$

Proof We follow the same proof technique than in Section 8.1.1, but now for regularized problems. We have by optimality of $\hat{\theta}$:

$$\|y - \Phi\hat{\theta}\|_2^2 + n\lambda\|\hat{\theta}\|_0 \leq \|y - \Phi\theta_*\|_2^2 + n\lambda\|\theta_*\|_0,$$

which leads to, using the inequality $2ab \leq 2a^2 + \frac{1}{2}b^2$:

$$\begin{aligned} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 &\leq 2\|\Phi(\hat{\theta} - \theta_*)\|_2 \cdot \varepsilon^\top \left(\frac{\Phi(\hat{\theta} - \theta_*)}{\|\Phi(\hat{\theta} - \theta_*)\|_2} \right) + n\lambda\|\theta_*\|_0 - n\lambda\|\hat{\theta}\|_0 \\ &\leq 2\left(\varepsilon^\top \left(\frac{\Phi(\hat{\theta} - \theta_*)}{\|\Phi(\hat{\theta} - \theta_*)\|_2} \right)\right)^2 + \frac{1}{2}\|\Phi(\hat{\theta} - \theta_*)\|_2^2 + n\lambda\|\theta_*\|_0 - n\lambda\|\hat{\theta}\|_0, \end{aligned}$$

leading to, by taking the supremum over $\theta \in \mathbb{R}^d$:

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq \sup_{\theta \in \mathbb{R}^d} \left\{ 4\left(\varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right)\right)^2 + 2n\lambda\|\theta_*\|_0 - 2n\lambda\|\theta\|_0 \right\}.$$

We then take the supremum by layers, as $\sup_{\theta \in \mathbb{R}^d} = \sup_{k' \in \{1, \dots, d\}} \sup_{|B|=k'} \sup_{\text{supp}(\theta)=B}$, that is, and using the same derivations as for Prop. 8.1 (A is the support of θ_*):

$$\begin{aligned} \mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] &\leq \mathbb{E} \left[\sup_{k' \in \{1, \dots, d\}} \sup_{|B|=k'} \sup_{\text{supp}(\theta)=B} \left\{ 4 \left(\varepsilon^\top \left(\frac{\Phi(\theta - \theta_*)}{\|\Phi(\theta - \theta_*)\|_2} \right) \right)^2 + 2n\lambda \|\theta_*\|_0 - 2n\lambda k' \right\} \right]^2 \\ &\leq 4\mathbb{E} \left[\sup_{k' \in \{1, \dots, d\}} \sup_{|B|=k'} \left\{ \|\Pi_{\Phi_{A \cup B}} \varepsilon\|^2 + \frac{n\lambda}{2} \|\theta_*\|_0 - \frac{n\lambda}{2} k' \right\} \right]^2. \end{aligned}$$

We thus get with the same reasoning as in Section 8.2.1 (based on the probabilistic lemmas from Section 8.1.2):

$$\begin{aligned} \mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] &\leq 16\sigma^2 \log \left(\sum_{k'=1}^d \binom{d}{2k'} 2^{2k'} \exp\left(\frac{n}{2} \frac{\lambda}{\sigma^2} \|\theta_*\|_0 - \frac{n}{2} \frac{\lambda}{\sigma^2} k'\right) \right) \\ &\leq 8n\lambda \|\theta_*\|_0 + 16\sigma^2 \log \left(\sum_{k'=1}^d \binom{d}{2k'} 2^{2k'} \exp\left(-\frac{n}{2} \frac{\lambda}{\sigma^2} k'\right) \right) \\ &\leq 8n\lambda \|\theta_*\|_0 + 16\sigma^2 \log \left(\sum_{k'=1}^d \left(\frac{ed}{2k'}\right)^{2k'} 2^{2k'} \exp\left(-\frac{n}{2} \frac{\lambda}{\sigma^2} k'\right) \right) \\ &\leq 8n\lambda \|\theta_*\|_0 + 16\sigma^2 \log \left(\sum_{k'=1}^d \left(\exp(k'(2 \log(d) + 2 - \frac{n}{2} \frac{\lambda}{\sigma^2})) \right) \right). \end{aligned}$$

We thus simply impose that $2 \log(d) + 2 - \frac{n}{2} \frac{\lambda}{\sigma^2} \leq -\log 2$, to get

$$\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 8n\lambda \|\theta_*\|_0 + 16\sigma^2 \log(2).$$

We can thus choose: $\lambda = \frac{2\sigma^2}{n}(3 + 2 \log d) \geq \frac{2\sigma^2}{n}(2 + \log 2)$, and get the desired result. \blacksquare

We can make the following observations:

- The penalty proportional to $\|\theta\|_0 \log d$ is often referred to as the “BIC penalty”.
- Note that we need to know σ^2 in advance, which can be a problem in practice. See [46] for more details and alternative formulations.
- The three most important aspects are that: (1) the bound does not require any assumption on the design matrix Φ , (2) that we observe a positive high-dimensional phenomenon, where d only appears as $\frac{\log d}{n}$, but (3) only exponential-time algorithms are possible for solving the problem with guarantees (see algorithms below).

Exercise 8.1 (\blacklozenge) *With a penalty proportional to $\|\theta\|_0 \log \frac{d}{\|\theta_0\|}$, show the same bound than for k known.*

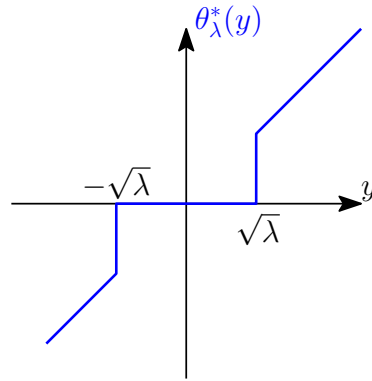
Algorithms. We can extend the two algorithms from Section 8.2.1 for the penalized case:

- **Forward-backward algorithm to minimize a function of a set B :** Starting from the empty set $B = \emptyset$, at every step of the algorithm, one tries both a forward algorithm (adding a node to B) and a backward algorithm (removing a node from B), and only perform a step if it decreases the overall cost function.

- **Iterative soft-thresholding:** compared to the constrained case, we minimize

$$\frac{1}{n}\|y - \Phi\theta_{t-1}\|_2^2 - \frac{2}{n}(y - \Phi\theta_{t-1})^\top \Phi(\theta - \theta_{t-1}) + \lambda_{\max}\left(\frac{1}{n}\Phi^\top \Phi\right)\|\theta - \theta_{t-1}\|_2^2 + \lambda\|\theta\|_0,$$

which can also be computed in closed form (by iterative hard thresholding). That is, with $\theta_t = \theta_{t-1} + \frac{1}{\lambda_{\max}(\Phi^\top \Phi)}\Phi^\top(y - \Phi\theta_{t-1})$, all components $(\theta_t)_j$ such that $|(\theta_t)_j|^2 \geq \frac{\lambda}{\frac{1}{n}\lambda_{\max}(\Phi^\top \Phi)}$, 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 $|y|^2 \leq \lambda$ and $\theta_\lambda^*(y) = y$ otherwise (see below).



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.

8.3 High-dimensional estimation through ℓ_1 -regularization

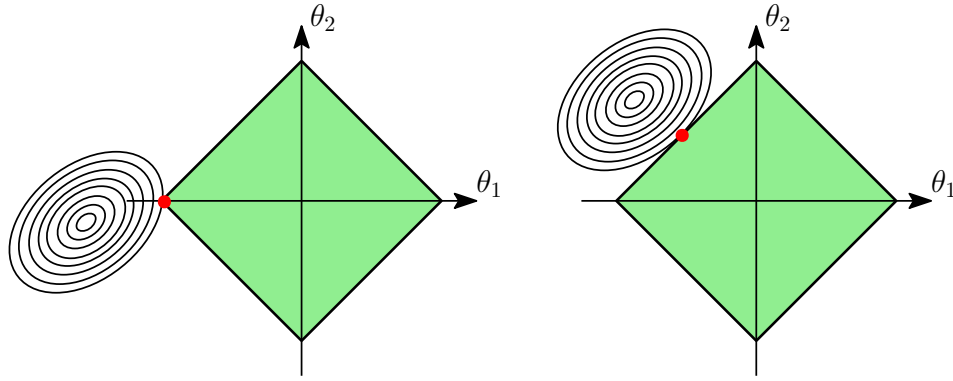
We now consider a computationally efficient alternative to ℓ_0 -penalties, namely using ℓ_1 -penalties, by minimizing, for the square loss:

$$\frac{1}{2n}\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1. \quad (8.3)$$

This is a convex optimization problem on which algorithms from Chapter 5 can be applied (see instances below). It is often referred to as the “Lasso” problem, for “least absolute shrinkage and selection operator”.

8.3.1 Intuition and algorithms

Sparsity-inducing effect. As opposed to the squared ℓ_2 -norm used in ridge regression, the ℓ_1 -norm is non differentiable, and its non-differentiability is not limited to $\theta = 0$, but in many other points. To see this, we can look at the ℓ_1 -ball and its different geometry compared to the ℓ_2 -ball. This is directly relevant to situations where we constrain the value of the norm instead of penalizing by it.

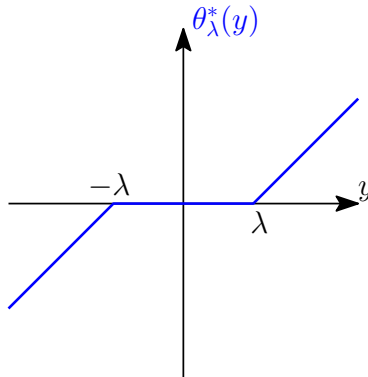


As shown above, where we represent the level set of a potential loss function, the solution of the minimization of the loss subject to the ℓ_1 -constraint (in green), is obtained when level sets are “tangent” to the constraint set. In the right part, this is obtained in a point away from the axes, but on the left part, this is achieved at one of the corners of the ℓ_1 -ball, which are points where one of the components of θ is equal to zero. Such corners are attractive and thus typically lead to sparse solutions.

One-dimensional problem. Another classical way to understand the sparsity-inducing effect is to consider the one-dimensional problem:

$$\min_{\theta \in \mathbb{R}} F(\theta) = \frac{1}{2}(y - \theta)^2 + \lambda|\theta|.$$

Since F is strongly-convex, it has a unique minimizer $\theta_\lambda^*(y)$. For $\lambda = 0$ (no regularization), we have $\theta_0^*(y) = y$, while for $\lambda > 0$, by computing left and right derivatives at zero (to be done as an exercise), one can check that $\theta_\lambda^*(y) = 0$ if $|y| \leq \lambda$, and $\theta_\lambda^*(y) = y - \lambda$ for $y > \lambda$, and $\theta_\lambda^*(y) = y + \lambda$ for $y < -\lambda$, which can be put all together as $\theta_\lambda^*(y) = \max\{|y| - \lambda, 0\} \text{sign}(y)$, which is depicted below. This referred to as iterative soft thresholding (this will be useful for proximal methods below).



Note that the minimizer is either sent to zero, or shrunk towards zero.

Optimization algorithms. We can adapt algorithms from Chapter 5 to the problem in Eq. (8.3).

- **Iterative hard-thresholding:** We can apply proximal methods to the objective function of the form $F(\theta) + \lambda\|\theta\|_1$ for $F(\theta) = \frac{1}{2n}\|y - \Phi\theta\|_2^2$, for which $F'(\theta) = -\frac{1}{2n}\Phi^\top(y - \Phi\theta)$. The plain (non-accelerated) proximal method recursion is

$$\theta_t = \arg \min_{\theta \in \mathbb{R}^d} F(\theta_{t-1}) + F'(\theta_{t-1})^\top(\theta - \theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2 + \lambda\|\theta\|_1,$$

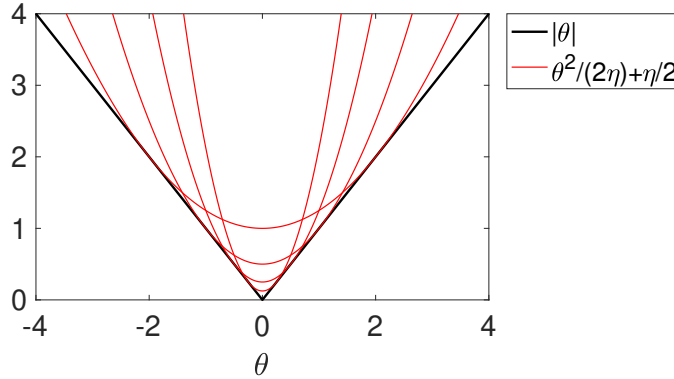
with $L = \lambda_{\max}(\frac{1}{n}\Phi^\top\Phi)$. This leads to $(\theta_t)_j = \max\{ |(\eta_t)_j| - \lambda, 0 \} \text{sign}((\eta_t)_j)$, for $\eta_t = \theta_{t-1} - \frac{1}{L}F'(\theta_{t-1})$. This simple algorithm can also be accelerated. The convergence rate then depends on invertibility of $\frac{1}{n}\Phi^\top\Phi$.

- **Coordinate descent:** Although the ℓ_1 -norm is a non-differentiable function, coordinate descent can be applied (because the ℓ_1 -norm is “separable”). At each iteration, we select a coordinate to update (at random or by cycling), and optimize with respect to this coordinate, which is a one-dimensional problem which can be solved in closed form. The convergence properties are similar to proximal methods [43].

η -trick. The non-differentiability 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 $\eta_j = |\theta_j|$. See below an example in one dimension, with $|\theta|$ and several quadratic upper bounds.



This leads to the reformulation of Eq. (8.3) as

$$\inf_{\theta \in \mathbb{R}^d} \frac{1}{2n}\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1 = \inf_{\eta \in \mathbb{R}_+^d} \inf_{\theta \in \mathbb{R}^d} \frac{1}{2n}\|y - \Phi\theta\|_2^2 + \frac{\lambda}{2} \sum_{j=1}^d \frac{\theta_j^2}{\eta_j} + \frac{\lambda}{2} \sum_{j=1}^d \eta_j,$$

and alternating optimization algorithms can be used: (a) minimizing with respect to 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 which can be solved by a linear system. See more details in <https://francisbach.com/the-%ce%b7-trick-or-the-effectiveness-of-reweighted-least-squares/>.

Optimality conditions (♦). In order to study the estimator defined by Eq. (8.3), 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 to zero. The gradient looks only at d directions (along the coordinate axis), while, in the non-smooth context, we need to look at all directions, that is, for all $\Delta \in \mathbb{R}^d$, we need that the directional derivative

$$\partial H(\theta, \Delta) = \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} [H(\theta + \varepsilon \Delta) - H(\theta)],$$

is non-negative. 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:

$$\partial H(\theta, \Delta) = F'(\theta)^\top \Delta + \lambda \sum_{j, \theta_j \neq 0} \text{sign}(\theta_j) \Delta_j + \lambda \sum_{j, \theta_j = 0} |\Delta_j|.$$

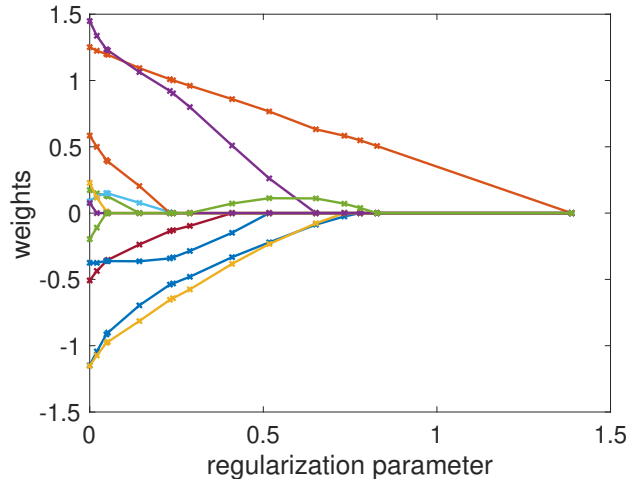
It is separable in Δ_j , $j = 1, \dots, d$, and it is non-negative for all j , if and only if, all components that depend on Δ_j are non-negative.

When $\theta_j \neq 0$, then this requires $F'(\theta)_j + \lambda \text{sign}(\theta_j) = 0$, while when $\theta_j = 0$, then we need $F'(\theta)_j \Delta_j + \lambda |\Delta_j| \geq 0$ for all Δ_j , which is equivalent to $|F'(\theta)_j| \leq \lambda$. This leads to the set of conditions:

$$\begin{cases} F'(\theta)_j + \lambda \text{sign}(\theta_j) = 0, & \forall j \in \{1, \dots, d\} \text{ such that } \theta_j \neq 0, \\ |F'(\theta)_j| \leq \lambda, & \forall j \in \{1, \dots, d\} \text{ such that } \theta_j = 0. \end{cases}$$

See [45] for more details.

Homotopy method (♦♦). We assume for simplicity that $\Phi^\top \Phi$ is invertible so that the minimizer $\theta(\lambda)$ is unique. Given a certain sign pattern for θ , optimality conditions are all convex in λ and thus define an interval in λ where the sign is constant. Given the sign, then the solution $\theta(\lambda)$ is affine in λ , leading to a piecewise affine function in λ (see an example of a regularization path below).



If we know the break points in λ and the associated signs, then we can compute all solutions for all λ . This is the source of the homotopy algorithm for Eq. (8.3), which starts with large λ and builds the path of solutions by computing break points one by one. See more details in [86].

8.3.2 Slow rates

We first consider an analysis based on simple tools and with no assumptions on the design matrix Φ . We will see that we can deal with high-dimensional inference problems where d can be large, but it will be rates in $1/\sqrt{n}$ and not $1/n$, hence the denomination “slow”.

We study the penalization by a general norm $\Omega : \mathbb{R}^d \rightarrow \mathbb{R}$ with dual norm Ω^* defined as $\Omega^*(z) = \sup_{\Omega(\theta) \leq 1} z^\top \theta$. We thus denote by $\hat{\theta}$ a minimizer of

$$\frac{1}{2n} \|y - \Phi\theta\|_2^2 + \lambda\Omega(\theta). \quad (8.4)$$

We first start by a lemma characterizing the excess risk in two situations: (a) where λ is large enough, and (b) in the general case.

Lemma 8.4 *Let $\hat{\theta}$ be a minimizer of Eq. (8.4).*

(a) *If $\Omega^*(\Phi^\top \varepsilon) \leq \frac{n\lambda}{2}$, then we have $\Omega(\hat{\theta}) \leq 3\Omega(\theta_*)$ and $\frac{1}{n} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 3\lambda\Omega(\theta_*)$.*

(b) *In all cases, $\frac{1}{n} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq \frac{4}{n} \|\varepsilon\|_2^2 + 4\lambda\Omega(\theta_*)$.*

Proof We have, like in Section 8.1.1, by optimality of $\hat{\theta}$ for Eq. (8.4):

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 2\varepsilon^\top \Phi(\hat{\theta} - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}).$$

Then, with the dual norm $\Omega^*(z) = \sup_{\Omega(\theta) \leq 1} z^\top \theta$, assuming that $\Omega^*(\Phi^\top \varepsilon) \leq \frac{n\lambda}{2}$, and using the triangle inequality:

$$\begin{aligned} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 &\leq 2\Omega^*(\Phi^\top \varepsilon)\Omega(\hat{\theta} - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}) \\ &\leq n\lambda\Omega(\hat{\theta} - \theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}) \\ &\leq n\lambda\Omega(\hat{\theta}) + n\lambda\Omega(\theta_*) + 2n\lambda\Omega(\theta_*) - 2n\lambda\Omega(\hat{\theta}) \leq 3n\lambda\Omega(\theta_*) - n\lambda\Omega(\hat{\theta}). \end{aligned}$$

This implies that $\Omega(\hat{\theta}) \leq 3\Omega(\theta_*)$ and $\frac{1}{n} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 3\lambda\Omega(\theta_*)$.

We also have a general bound through:

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq 2\|\varepsilon\|_2 \|\Phi(\hat{\theta} - \theta_*)\|_2 + 2n\lambda\Omega(\theta_*),$$

which leads to, using the identity $2ab \leq \frac{1}{2}a^2 + 2b^2$,

$$\|\Phi(\hat{\theta} - \theta_*)\|_2^2 \leq \frac{1}{2} \|\Phi(\hat{\theta} - \theta_*)\|_2^2 + 2\|\varepsilon\|_2^2 + 2n\lambda\Omega(\theta_*),$$

which leads to the desired bound. ■

We can now use the lemma above to compute the excess risk of the Lasso, for which $\Omega = \|\cdot\|_1$ and $\Omega^*(\Phi^\top \varepsilon) = \|\Phi^\top \varepsilon\|_\infty$. The key is to note that since $\|\Phi^\top \varepsilon\|_\infty$ is a maximum of $2d$ terms that scales as \sqrt{n} , according to Section 1.2.4, its maximum scales as $\sqrt{n \log(d)}$, and we will apply the lemma above when λ is larger than $\sqrt{\frac{\log d}{n}}$. We denote by $\|\widehat{\Sigma}\|_\infty$ the largest element of the matrix $\widehat{\Sigma}$ in absolute value.

Proposition 8.3 (Lasso - slow rate) *Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance σ^2 . Let $\hat{\theta}$ be the minimizer of Eq. (8.3). Then, for $\lambda = 4\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat{\Sigma}\|_\infty}$, we have:*

$$\mathbb{E}\left[\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2\right] \leq 28\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat{\Sigma}\|_\infty}\|\theta_*\|_1 + \frac{24}{n}\sigma^2.$$

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 , $\mathbb{P}(|z| \geq t) \leq 2\exp(-t^2/2)$:

$$\mathbb{P}(\|\Phi^\top \varepsilon\|_\infty > \frac{n\lambda}{2}) \leq \sum_{j=1}^d \mathbb{P}(|\Phi^\top \varepsilon|_j > \frac{n\lambda}{2}) \leq 2 \sum_{j=1}^d \exp\left(-\frac{n\lambda^2}{8\sigma^2\widehat{\Sigma}_{jj}}\right) \leq 2d \exp\left(-\frac{n\lambda^2}{8\sigma^2\|\widehat{\Sigma}\|_\infty}\right) = \delta.$$

Thus, with probability greater than $1 - \delta$, we can apply the first part of Lemma 8.4, and thus the error is less than $3\lambda\|\theta^*\|_1$. This would be the end of the proof if a high-probability result was desired. For a result in expectation, we need also the second part.

Overall, we get, denoting \mathcal{A} the event $\mathcal{A} = \{\Omega^*(\Phi^\top \varepsilon) \leq \frac{n\lambda}{2}\}$, and the previous lemma:

$$\begin{aligned} \mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] &= \mathbb{E}[1_{\mathcal{A}}\|\Phi(\hat{\theta} - \theta_*)\|_2^2] + \mathbb{E}[1_{\mathcal{A}^c}\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \\ &\leq 3n\lambda\|\theta_*\|_1 + \mathbb{E}[1_{\mathcal{A}^c}(4\|\varepsilon\|_2^2 + 4n\lambda\|\theta_*\|_1)] \end{aligned}$$

Using Cauchy-Schwarz inequality, this leads to

$$\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 7n\lambda\|\theta_*\|_1 + 4\mathbb{P}(\mathcal{A}^c)^{1/2} \left(\mathbb{E}[\|\varepsilon\|_2^4]\right)^{1/2}.$$

With Gaussian noise, we have: $\sqrt{\mathbb{E}[\|\varepsilon\|_2^4]} \leq 3n\sigma^2$, leading to:

$$\frac{1}{n}\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 7\lambda\|\theta_*\|_1 + 24d\sigma^2 \exp\left(-\frac{n\lambda^2}{16\sigma^2\|\widehat{\Sigma}\|_\infty}\right).$$

With $\frac{n\lambda^2}{16\sigma^2\|\widehat{\Sigma}\|_\infty} = \log(dn)$, we get

$$\frac{1}{n}\mathbb{E}[\|\Phi(\hat{\theta} - \theta_*)\|_2^2] \leq 7\lambda\|\theta_*\|_1 + \frac{24}{n}\sigma^2 \leq 28\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\widehat{\Sigma}\|_\infty}\|\theta_*\|_1 + \frac{24}{n}\sigma^2.$$



Check homogeneity!

■

We already observe some high-dimensional phenomenon with the term $\sqrt{\frac{\log d}{n}}$, where n can be much larger than d (if of course we assume that the optimal predictor θ_* is sparse).

Exercise 8.2 (♦) *Using Rademacher complexities from Chapter 4, show a similar slow rate for ℓ_1 -constrained optimization with Lipschitz-continuous losses.*

8.3.3 Fast rates (♦)

We now consider conditions to obtain a fast rate with leading term proportional to $\sigma^2 \frac{k \log d}{n}$, which is the same as for ℓ_0 -penalty, but with tractable algorithms. This will come with extra (very) strong conditions on the design matrix Φ .

We start with a simple (but crucial) lemma, characterizing the solution of Eq. (8.3) in terms of the support A of θ_* .

Lemma 8.5 *Let $\hat{\theta}$ be a minimizer of Eq. (8.4). Assume $\|\Phi^\top \varepsilon\|_\infty \leq \frac{n\lambda}{2}$. If $\Delta = \hat{\theta} - \theta_*$, then $\|\Delta_{A^c}\|_1 \leq 3\|\Delta_A\|_1$ and $\|\Phi\Delta\|_2^2 \leq 3n\lambda\|\Delta_A\|_1$.*

Proof We have, like in previous proofs, with $\Delta = \hat{\theta} - \theta_*$, and A the support of θ_* :

$$\|\Phi\Delta\|_2^2 \leq 2\varepsilon^\top \Phi\Delta + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat{\theta}\|_1.$$

Then, assuming that $\|\Phi^\top \varepsilon\|_\infty \leq \frac{n\lambda}{2}$,

$$\begin{aligned} \|\Phi\Delta\|_2^2 &\leq 2\|\Phi^\top \varepsilon\|_\infty \|\Delta\|_1 + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat{\theta}\|_1 \\ \|\Phi\Delta\|_2^2 &\leq n\lambda\|\Delta\|_1 + 2n\lambda\|\theta_*\|_1 - 2n\lambda\|\hat{\theta}\|_1. \end{aligned}$$

We then use, by using the decomposability of the ℓ_1 -norm and the [triangle inequality](#):

$$\|\theta_*\|_1 - \|\hat{\theta}\|_1 = \|(\theta_*)_A\|_1 - \|\theta_* + \Delta\|_1 = \|(\theta_*)_A\|_1 - \|(\theta_* + \Delta)_A\|_1 - \|\Delta_{A^c}\|_1 \leq \|\Delta_A\|_1 - \|\Delta_{A^c}\|_1,$$

to get

$$\begin{aligned} \|\Phi\Delta\|_2^2 &\leq n\lambda\|\Delta\|_1 + 2n\lambda(\|\theta_*\|_1 - \|\hat{\theta}\|_1) \leq n\lambda\|\Delta\|_1 + 2n\lambda(\|\Delta_A\|_1 - \|\Delta_{A^c}\|_1) \\ &\leq n\lambda(\|\Delta_A\|_1 + \|\Delta_{A^c}\|_1) + 2n\lambda(\|\Delta_A\|_1 - \|\Delta_{A^c}\|_1) = 3n\lambda\|\Delta_A\|_1 - n\lambda\|\Delta_{A^c}\|_1. \end{aligned}$$

This leads to $\|\Delta_{A^c}\|_1 \leq 3\|\Delta_A\|_1$ and the other desired inequality. ■

We can now add an extra assumption that will make the proof go through, namely

$$\frac{1}{n}\|\Phi\Delta\|_2^2 \geq \kappa\|\Delta_A\|_2^2 \tag{8.5}$$

for all Δ that satisfies the condition $\|\Delta_{A^c}\|_1 \leq 3\|\Delta_A\|_1$. This is called the “restrictive eigenvalue property”, because if the smallest eigenvalue of $\frac{1}{n}\Phi^\top \Phi$ is greater than κ , the condition is satisfied (but this is only possible if $n \geq d$). The relevance of this assumption is discussed in Section 8.3.4.

This leads to the following proposition.

Proposition 8.4 (Lasso - fast rate) Assume $y = \Phi\theta_* + \varepsilon$, with $\varepsilon \in \mathbb{R}^n$ a vector with independent Gaussian components of zero mean and variance σ^2 . Let $\hat{\theta}$ be the minimizer of Eq. (8.3). Then, for $\lambda = 4\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\hat{\Sigma}\|_\infty}$, we have, *if Eq. (8.5) is satisfied*:

$$\mathbb{E}\left[\frac{1}{n}\|\Phi(\hat{\theta} - \theta_*)\|_2^2\right] \leq \frac{144|A|\sigma^2\|\hat{\Sigma}\|_\infty}{\kappa} \frac{\log(dn)}{n} + \frac{24}{n}\sigma^2 + \frac{32}{dn^2}\|\theta_*\|_1\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\hat{\Sigma}\|_\infty}.$$

Proof (◆) We have, when λ is large enough, and by application of Lemma 8.5, and using Eq. (8.5):

$$\|\Delta_A\|_1 \leq |A|^{1/2}\|\Delta_A\|_2 \leq \frac{|A|^{1/2}}{\sqrt{n\kappa}}\|\Phi\Delta\|_2 \leq \frac{|A|^{1/2}}{\sqrt{n\kappa}}\sqrt{3n\lambda\|\Delta_A\|_1},$$

which leads to $\|\Delta_A\|_1 \leq \frac{3|A|\lambda}{\kappa}$. We then get $\frac{1}{n}\|\Phi\Delta\|_2^2 \leq \frac{9|A|\lambda^2}{\kappa}$, and we can reuse the same reasoning as for the slow rate, to get

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n}\|\Phi\Delta\|_2^2\right] &\leq \frac{9|A|\lambda^2}{\kappa} + 8d\lambda\|\theta_*\|_1 \exp\left(-\frac{n\lambda^2}{8\sigma^2\|\hat{\Sigma}\|_\infty}\right) + \frac{24}{n}\sigma^2 \\ &\leq \frac{9|A|\lambda^2}{\kappa} + 8d\lambda\|\theta_*\|_1 \frac{1}{d^2n^2} + \frac{24}{n}\sigma^2 \\ &\leq \frac{144|A|\sigma^2\|\hat{\Sigma}\|_\infty}{\kappa} \frac{\log(dn)}{n} + \frac{24}{n}\sigma^2 + \frac{32}{dn^2}\|\theta_*\|_1\sigma\sqrt{\frac{\log(dn)}{n}}\sqrt{\|\hat{\Sigma}\|_\infty}. \end{aligned}$$

■

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.

8.3.4 Zoo of conditions (◆◆)

Conditions to obtain fast rates are plentiful: they all assume that there is low-correlation among predictors, which is rarely the case in practice (in particular, if there are two features which are equal, they are never satisfied).

Restricted eigenvalue property (REP). The most direct condition is the so-called restricted eigenvalue property (REP), which is exactly Eq. (8.5), with the supremum taken over the unknown set A of cardinality less than k :

$$\inf_{|A| \leq k} \inf_{\|\Delta_{A^c}\|_1 \leq 3\|\Delta_A\|_1} \frac{\|\Phi\Delta\|_2^2}{n\|\Delta_A\|_2^2} \geq \kappa > 0.$$

Mutual incoherence condition. A simpler one to check, but weaker, is the mutual incoherence condition:

$$\sup_{i \neq j} |\hat{\Sigma}_{ij}| \leq \frac{\min_{j \in \{1, \dots, d\}} \hat{\Sigma}_{jj}}{14k}, \quad (8.6)$$

which states that all cross-correlation coefficients are small (pure decorrelation would set them to zero).

This is weaker than the REP condition above. Indeed, by expanding, we have:

$$\|\Phi\Delta\|_2^2 = \|\Phi_A\Delta_A + \Phi_{A^c}\Delta_{A^c}\|_2^2 = \|\Phi_A\Delta_A\|_2^2 + 2\Delta_A^\top \Phi_A^\top \Phi_{A^c} \Delta_{A^c} + \|\Phi_{A^c}\Delta_{A^c}\|_2^2 \geq \|\Phi_A\Delta_A\|_2^2 + 2\Delta_A^\top \Phi_A^\top \Phi_{A^c} \Delta_{A^c}.$$

Moreover, we have:

$$\Delta_A^\top \widehat{\Sigma}_{AA} \Delta_A = \Delta_A^\top \text{Diag}(\text{diag}(\widehat{\Sigma}_{AA})) \Delta_A + \Delta_A^\top (\widehat{\Sigma}_{AA} - \text{Diag}(\text{diag}(\widehat{\Sigma}_{AA}))) \Delta_A \geq \min_{j \in \{1, \dots, d\}} \widehat{\Sigma}_{jj} (\|\Delta_A\|_2^2 - \frac{1}{14k} \|\Delta_A\|_1^2),$$

and

$$|\Delta_A^\top \Phi_A^\top \Phi_{A^c} \Delta_{A^c}| \leq \frac{\min_{j \in \{1, \dots, d\}} \widehat{\Sigma}_{jj}}{14k} \|\Delta_{A^c}\|_1 \|\Delta_A\|_1 \leq \frac{3 \min_{j \in \{1, \dots, d\}} \widehat{\Sigma}_{jj}}{14k} \|\Delta_A\|_1^2.$$

This leads to $\frac{1}{n} \|\Phi\Delta\|_2^2 \geq \min_{j \in \{1, \dots, d\}} \widehat{\Sigma}_{jj} \left(\|\Delta_A\|_2^2 - \frac{7}{14k} \|\Delta_A\|_1^2 \right) \geq \min_{j \in \{1, \dots, d\}} \widehat{\Sigma}_{jj} \left(\|\Delta_A\|_2^2 - \frac{7k}{14k} \|\Delta_A\|_2^2 \right)$, thus leading to $\kappa = \min_{j \in \{1, \dots, d\}} \widehat{\Sigma}_{jj}/2$ for the REP condition.

Restricted isometry property. One of the earlier conditions was the restricted isometry property: all eigenvalues of submatrices of $\widehat{\Sigma}$ of size less than $2k$, are between $1 - \delta$ and $1 + \delta$ for δ small enough. See [45, 105] for details.

Gaussian designs (♦). It is not obvious that the conditions above are non-trivial (that is, there may exist no matrix with good sizes d and n for k large enough). In order for our results to be non-trivial, we need that $k \frac{\log d}{n}$ is small but not too small. We show in this paragraph that when sampling from Gaussian distributions, then assumptions above are satisfied. This is a first step towards a random design assumption.

Theorem 8.1 ([105], Theorem 7.16) *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 property is satisfied with $\kappa = \frac{c_1}{2} \lambda_{\min}(\Sigma)$ as soon as $k \frac{\log d}{n} \leq \frac{c_1}{8c_2} \frac{\lambda_{\min}(\Sigma)}{\|\Sigma\|_\infty}$, with $c_1 = 1/8$ and $c_2 = 50$.*

The theorem above is hard to prove, the following exercise proposes to prove a weaker result, showing that the guarantees for the maximal cardinality k of the support has to be smaller.

Exercise 8.3 (♦♦♦) *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 $k^2 \frac{\log d}{n}$, then mutual incoherence property in Eq. (8.6) is satisfied.*

Model selection and irrepresentable condition (♦). Given that the Lasso aims at performing variable selection, it is natural to study its capacity to find the support of θ_* , that is, the set of non-zero variables. It turns out that it also depends on some conditions on the design matrix, which are stronger

than the REP conditions, and called the “irrepresentable condition”, and also valid for Gaussian random matrices with similar scalings between n , d and k . See [45, 105] for details.



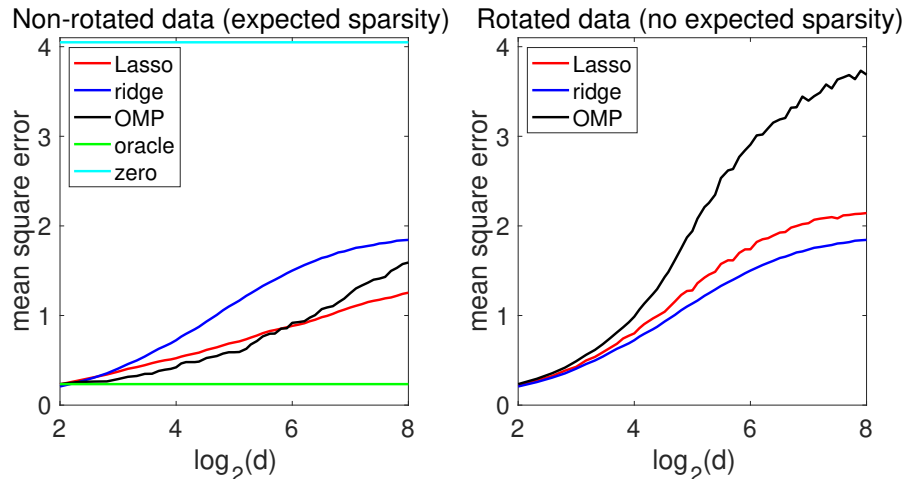
Algorithmic and theoretical tools are similar to “compressed sensing”, where the design matrix represents a set of measurements, which can be chosen by the user/theoretician. In this context, sampling from i.i.d. Gaussians make sense. For machine learning and statistics, the design matrix is the data, and comes **as it is**, often with strong correlations.

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 that remains constant when k varies). We perform 128 replications. For each method and each value of its hyperparameter, we averaged the test risk over the 128 replications and report the minimum value (with respect to the hyperparameter). We compare the following three methods:

- Ridge regression: penalty by $\lambda \|\theta\|_2^2$.
- Lasso regression: penalty by $\lambda \|\theta\|_1$.
- Orthogonal matching pursuit (greedy forward method), with hyperparameter k (the number of included variables).

We compare two situations: (1) non-rotated data (exactly the model above), and (2) rotated data, where we replace Φ by ΦR and θ_* by $R^\top \theta_*$, where R is a rotation matrix. For the rotated data, we do not expect sparse solutions, and hence sparse methods are not expected to work better than ridge regression (and OMP performs significantly worse because once the support is chosen, there is no regularization). Note that the two curves for ridge regression are exactly the same (as expected from rotation invariance of the 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 a number of ways:

- **Group penalties:** in many cases, $\{1, \dots, d\}$ is partitioned into m subsets A_1, \dots, A_m , and the goal is to consider “group sparsity”, that is, if we select one variable within a group A_j , the entire group should be selected. Such behavior can be obtained using the penalty $\sum_{i=1}^m \|\theta_{A_i}\|_2$ or $\sum_{i=1}^m \|\theta_{A_i}\|_\infty$. See, e.g., [45] for details.
- **Structured sparsity:** it is also possible to favor other specific patterns for the selected variables, such as blocks, trees, etc. See [13] for details.
- **Nuclear norm:** when learning on matrices, a natural form of sparsity is for a matrix to have low rank. This can be achieved by penalizing by the sum of singular values of a matrix, which is a norm called the nuclear norm or the trace norm. See [8] and references therein.
- **Multiple kernel learning:** the group penalty can be extended when the groups have an infinite dimension and ℓ_2 -norms are replaced by RKHS norms defined in Chapter 7. This becomes a tool to learn the kernel matrix from data. See [12] for details.
- **Elastic net:** often, when both effects of the ℓ_1 -norm (sparsity) and of the squared ℓ_2 -norm (strong-convexity) are desired, we can sum the two, which is referred to as the “elastic net” penalty.
- **Concave penalization and debiasing:** in order 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 on debiasing the Lasso, an elegant one being to use a “concave” penalty. That is, we use $\sum_{i=1}^d a(|\theta_i|)$ where a is a concave increasing function on \mathbb{R}^+ , such as $a(u) = u^\alpha$ for $\alpha \in (0, 1)$. This leads to a non-convex optimization problem, where iterative weighted ℓ_1 -minimization provides natural algorithms (see [71] and references therein).

Chapter 9

Neural networks

Chapter summary

- Single hidden layer neural networks: Using combinations of simple affine functions with additional non-linearity.
- Estimation error: the number of parameters is not the driver of the estimation error, the norms of the various weights play an important role.
- Approximation properties and universality: for the “ReLU” activation function, the approximation properties can be characterized and are superior to kernel methods because they are adaptive to linear structures.

9.1 Introduction

In supervised learning, the main focus has been 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 $f : \mathcal{X} \rightarrow \mathbb{R}$ where the following cost function is minimized:

$$\frac{1}{n} \sum_{i=1}^n \ell(y_i, f(x_i)) + \Omega(f),$$

where $\ell : \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}$ 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, e.g., $\Phi(u) = \max\{1 - u, 0\}$ (hinge loss leading to the support vector machine) or $\Phi(u) = \log(1 + \exp(-u))$ (leading to logistic regression).

The class of functions we have considered so far were (with their “pros” and “cons”):

- **Linear functions in some explicit features:** given a feature map $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$, we consider $f(x) = \theta^\top \varphi(x)$, with parameters $\theta \in \mathbb{R}^d$, as analyzed in Chapter 3 (for least-squares) and Chapter 4.

() *Pros:* Simple to implement, convex optimization with gradient descent algorithms, with running time complexity in $O(nd)$, and theoretical guarantees.

() *Cons:* Only applies to linear functions on explicit (and fixed feature spaces), so they can underfit the data.

- **Linear functions in some implicit features through kernel methods:** the feature map can have arbitrarily large dimension, that is, $\varphi(x) \in \mathcal{H}$ where \mathcal{H} is a Hilbert space, accessed through a kernel $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$, as presented in Chapter 7.

() *Pros:* Non-linear flexible predictions, simple to implement, convex optimization algorithms with strong guarantees. Provides adaptivity to regularity of the target function.

() *Cons:* Running-time complexity up to $O(n^2)$. May still suffer from the curse of dimensionality for non-smooth target functions.

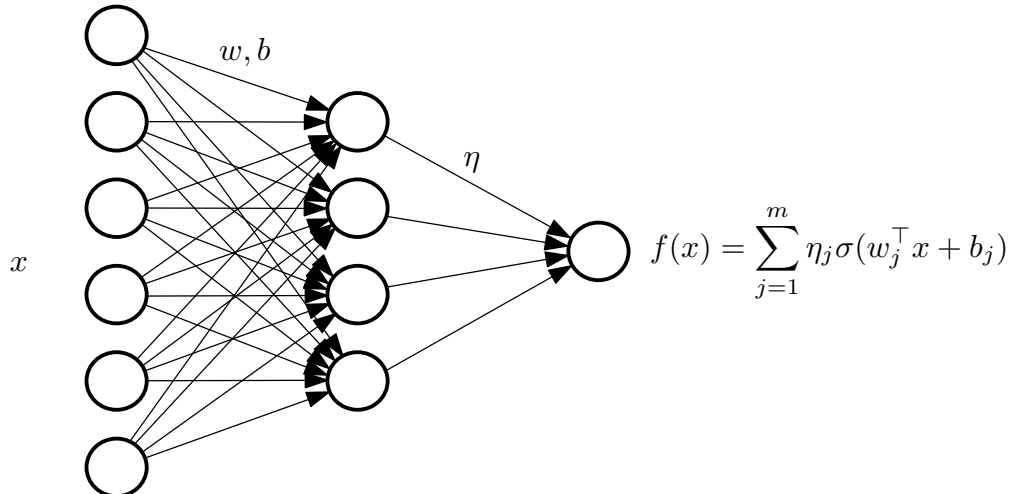
The goal of this chapter is to explore another class of functions for non-linear predictions, namely neural networks, that come with additional benefits, such as more “adaptivity for linear structures”, but comes with some potential drawbacks, such as a harder optimization problem.

9.2 Single hidden layer neural network

We consider $\mathcal{X} = \mathbb{R}^d$ and the set of functions that can be written as

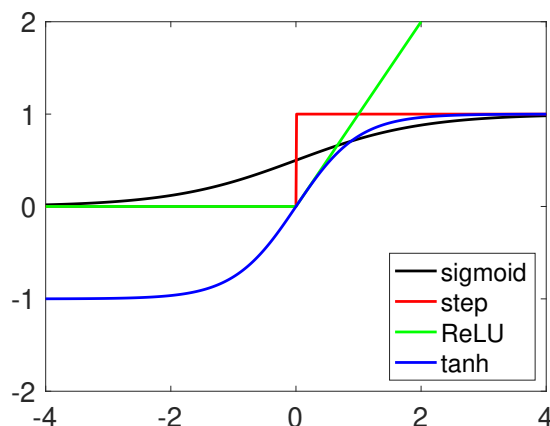
$$f(x) = \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j), \quad (9.1)$$

where $w_j \in \mathbb{R}^d$, $b_j \in \mathbb{R}$ and $\eta_j \in \mathbb{R}$, $j = 1, \dots, m$, and σ is an activation function. This is often represented as a graph (see below).



The activation function is typically from one of the following examples (see plot below):

- sigmoid $\sigma(u) = \frac{1}{1+e^{-u}}$,
- step $\sigma(u) = 1_{u>0}$,
- rectified linear unit (ReLU) $\sigma(u) = (u)_+ = \max\{u, 0\}$,
- hyperbolic tangent $\sigma(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$.



The function f is defined as the linear combination of m functions $x \mapsto \sigma(w_j^\top x + b_j)$, which are the “hidden neurons”.



The constant terms b_j are sometimes referred to as “biases”, which is unfortunate in a statistical context.



Do not get confused by the name “neural network” and its biological inspiration. This inspiration is not a proper justification of its behavior on machine learning problems.



Following standard practice, we are not adding a non-linearity for the last layer; note that if we were to use an additional sigmoid activation and using the cross-entropy loss for binary classification, we would exactly be using the logistic loss on the output without an extra activation function.

As any method based on empirical risk minimization, we have to study the three classical aspects: (1) optimization (convergence properties of algorithms for minimizing the risk), (2) estimation error (effect of having a finite amount of data on the prediction performance), and (3) approximation error (effect of having a finite number of parameters).

9.2.1 Optimization

In order 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:

$$\min_{\theta \in \mathbb{R}^{m(d+2)}} \frac{1}{n} \sum_{i=1}^n \ell\left(y_i, \sum_{j=1}^m \eta_j \sigma(w_j^\top x_i + b_j)\right).$$

⚠ Note that in the true objective is to perform well on unseen data, and the optimization problem is just a mean to an end. See Chapter 4 and Chapter 5.

This is a non-convex optimization problem where the gradient descent algorithms from Chapter 5 can be applied without guarantees (see Section 9.5 for recent results on providing some qualitative global convergence guarantees when m is large). Sometimes regularization is added on the parameters.

While stochastic gradient descent remains an algorithm of choice, several tricks have been observed to lead to better stability and performance: specific step-size decay schedules, momentum, batch-normalization, etc. But overall, the objective function is non-convex, and it remains difficult to understand why gradient-based methods perform well in practice (some elements are presented in Section 9.5).

See <https://playground.tensorflow.org/> for a nice interactive illustration.

9.2.2 Estimation error

In order to study the estimation error, we will consider that the parameters of the network are constrained, that is, $\Omega(\theta) \leq D$ for a certain norm Ω that we will define below. We can then compute the Rademacher complexity of the associated class \mathcal{F} of function we just defined, using tools from Chapter 4 (Section 4.5).

We consider an ℓ_1 -bound $\|\eta\|_1 \leq D_\eta$, as this will be our main tool for approximation theory in later sections.

We have, by definition of the Rademacher complexity $R_n(\mathcal{F})$ of \mathcal{F} , and taking expectations with respect to the data (x_i, y_i) , $i = 1, \dots, n$ (which is assumed i.i.d.) and the independent Rademacher random variables $\varepsilon_i \in \{-1, 1\}$:

$$R_n(\mathcal{F}) = \mathbb{E} \left[\sup_{\theta \in \mathbb{R}^{m(d+2)}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \ell(y_i, f_\theta(x_i)) \right].$$

Assuming the loss is almost surely G_ℓ -Lipschitz-continuous with respect to the second variable, using Proposition 4.3 from Chapter 4 that allows to get rid of the loss, we get the bound:

$$R_n(\mathcal{F}) \leq G_\ell \mathbb{E} \left[\sup_{\theta \in \mathbb{R}^{m(d+2)}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i f_\theta(x_i) \right] = G_\ell \mathbb{E} \left[\sup_{\theta \in \mathbb{R}^{m(d+2)}} \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m \eta_j \varepsilon_i \sigma(w_j^\top x_i + b_j) \right].$$

Using the ℓ_1 -constraint on η and using $\sup_{\|\eta\|_1 \leq D_\eta} z^\top \eta = D_\eta \|z\|_\infty$, we can directly maximize with respect to η , leading to (note that another ℓ_p -constraint on η would be harder to deal with):

$$R_n(\mathcal{F}) \leq G_\ell \mathbb{E} \left[\sup_{(w,b) \in \mathbb{R}^{m(d+1)}} \sup_{s \in \{-1,1\}} \sup_{j \in \{1,\dots,m\}} D_\eta s \frac{1}{n} \sum_{i=1}^n \varepsilon_i \sigma(w_j^\top x_i + b_j) \right].$$

Assuming the activation function σ is G_σ -Lipschitz continuous, we get, again using Proposition 4.3 from Chapter 4 :

$$R_n(\mathcal{F}) \leq G_\ell D_\eta G_\sigma \mathbb{E} \left[\sup_{(w,b) \in \mathbb{R}^{m(d+1)}} \sup_{j \in \{1,\dots,m\}} \sup_{s \in \{-1,1\}} s \left\{ w_j^\top \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i x_i \right) + b_j \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i \right) \right\} \right].$$

If we assume that we bound $\Theta(w_j, b_j) \leq D_{w,b}$, for each $j \in \{1, \dots, m\}$, we get, with the usual definition of the dual norm $\Theta^*(u, v) = \sup_{\Theta(w,b) \leq 1} \begin{pmatrix} w \\ b \end{pmatrix}^\top \begin{pmatrix} u \\ v \end{pmatrix}$:

$$R_n(\mathcal{F}) \leq G_\ell D_\eta G_\sigma D_{w,b} \mathbb{E} \left[\Theta^* \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i x_i, \frac{1}{n} \sum_{i=1}^n \varepsilon_i \right) \right].$$

Using $\Theta(w, b) = \max\{\|w\|_2, |b|/\sqrt{\mathbb{E}\|x\|_2^2}\}$, with $\Theta^*(u, v) = \|u\|_2 + |v|\sqrt{\mathbb{E}\|x\|_2^2}$, we get, using Jensen's inequality (of the form $\mathbb{E}[Z] \leq \sqrt{\mathbb{E}[Z^2]}$):

$$\begin{aligned} \mathbb{E} \left[\Theta^* \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i x_i, \frac{1}{n} \sum_{i=1}^n \varepsilon_i \right) \right] &= \mathbb{E} \left[\left\| \frac{1}{n} \sum_{i=1}^n \varepsilon_i x_i \right\|_2 \right] + \sqrt{\mathbb{E}\|x\|_2^2} \mathbb{E} \left[\left| \frac{1}{n} \sum_{i=1}^n \varepsilon_i \right| \right] \\ &\leq \sqrt{\mathbb{E} \left[\left\| \frac{1}{n} \sum_{i=1}^n \varepsilon_i x_i \right\|_2^2 \right]} + \sqrt{\mathbb{E}\|x\|_2^2} \sqrt{\mathbb{E} \left[\left| \frac{1}{n} \sum_{i=1}^n \varepsilon_i \right|^2 \right]}. \end{aligned}$$

Then using independence of all ε_i and their zero means, we get

$$\mathbb{E} \left[\Theta^* \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i x_i, \frac{1}{n} \sum_{i=1}^n \varepsilon_i \right) \right] \leq 2 \sqrt{\frac{\mathbb{E}\|x\|_2^2}{n}}.$$

Thus, we get the following proposition, with a bound proportional to $1/\sqrt{n}$ with no explicit dependence in the number of parameters.

Proposition 9.1 *Let \mathcal{F} be the class of functions $(y, x) \mapsto \ell(y, f(x))$ where f is a neural network defined in Eq. (9.1), with the constraint that $\|\eta\|_1 \leq D_\eta$, $\max\{\|w_j\|_2, |b_j|/\sqrt{\mathbb{E}\|x\|_2^2}\} \leq D_{w,b}$ for all $j \in \{1, \dots, m\}$. If the loss function is G_ℓ -Lipschitz-continuous and the activation function σ is G_σ -Lipschitz-continuous, the Rademacher complexity is upperbounded as*

$$R_n(\mathcal{F}) \leq 2G_\ell G_\sigma D_{w,b} D_\eta \frac{\sqrt{\mathbb{E}\|x\|_2^2}}{\sqrt{n}}.$$

The proposition above allows to bound the estimation error for neural networks, as the maximal deviation between expected risk and empirical risk over all potential networks with bounded parameters is bounded in expectation by twice the Rademacher complexity above.

For the ReLU activation function, where $G_\sigma = 1$, this will be combined with a study of the approximation properties in Section 9.3.



The number of parameters is irrelevant!!!!!!
What counts is the overall norm of the weights.



Check homogeneity.

When the norm of weights is not explicitly penalized or contained, we will see in Chapter 10 some recent results showing how optimization algorithms add an implicit regularization that leads to provable generalization in over-parameterized neural networks (that is, networks with many hidden units).

Exercise 9.1 (♦) *Provide the bound for $\Omega(w, b) = \max\{\|w\|_1, |b|/\sup \|x\|_\infty\}$, where $\sup \|x\|_\infty$ denotes the supremum of $\|x\|_\infty$ over all x in the support of its distribution.*

9.3 Approximation properties of single-hidden layer neural networks

As seen above, the estimation error grows as $\frac{\|\eta\|_1}{\sqrt{n}}$, and is independent of the number m of neurons. Two important questions will be tackled in this section:

- What is the associated approximation error so that we can derive generalization bounds?
- What will be the number of neurons required to reach such a behavior?

For this, we need to understand the space of functions that neural networks span, and how they relate to smoothness properties of the function. We first draw a link with kernel methods from Chapter 7.

In this chapter, we focus primarily on the ReLU activation function, noting that universal approximation results exist as soon as σ is not a polynomial [67].

9.3.1 Link with kernel methods

Learning features and kernels. A one-hidden layer neural network corresponds to a linear classifier with feature vector of dimension m

$$\varphi(x)_j = \frac{1}{\sqrt{m}} \sigma(w_j^\top x + b_j)$$

parameterized by all weights w_j, b_j , with kernel

$$\hat{k}(x, x') = \frac{1}{m} \sum_{j=1}^m \sigma(w_j^\top x + b_j) \sigma(w_j^\top x' + b_j).$$

This corresponds to penalizing the output weights η_j , $j \in \{1, \dots, m\}$, by $m \sum_{j=1}^m \eta_j^2$, and keeping the input weights (w_j, b_j) fixed, for $j = 1, \dots, m$. Thus, a neural networks, can be seen a learning from data a feature representations $\varphi(x)$ (with parameters $\{(w_j), (b_j)\}$), and thus, equivalently a kernel function.

Random input weights. With random independent and identically distributed weights $w_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$, when m tends to infinity (a set-up often referred to as the “over-parameterized” set-up), by the law of large numbers, we get

$$\hat{k}(x, x') \rightarrow k(x, x') = \mathbb{E} \left[\sigma(w^\top x + b) \sigma(w^\top x' + b) \right].$$

Therefore, infinite width networks where input weights are random and only output weights are learned are in fact kernel methods in disguise [77, 89].

This kernel can be computed in closed form for simple activations and distributions of weights [37, 11], and thus the same regularization properties may be achieved with algorithms from Chapter 7 (which are based on convex optimization, and thus come with guarantees). Note that as shown in Section 7.4, a common strategy for kernels defined as expectations is to use the a *random feature* approximation $\hat{k}(x, x')$, that is, here, use explicitly the neural network representation.



The kernel approximation corresponds to input weights w_j, b_j sampled randomly and *held fixed*. Only the output weights η_j are optimized.

Exercise 9.2 For $\binom{w}{b/R}$ uniform on the sphere, and for the ReLU activation, compute the associated kernel as a function of the cosine between the vectors $\binom{x}{R}$ and $\binom{x'}{R}$.

Integral representations of functions in the RKHS. When using a slightly different normalization and writing instead $f(x) = \frac{1}{m} \sum_{i=1}^m \tilde{\eta}_j \sigma(w_j^\top x + b_j)$, with $\tilde{\eta}_j = m \eta_j$, the penalty becomes $\frac{1}{m} \sum_{j=1}^m \tilde{\eta}_j^2$, and expressions of the form

$$\frac{1}{m} \sum_{j=1}^m \tilde{\eta}_j F(w_j, b_j)$$

can be seen (by the law of large numbers) as the integral

$$\int_{\mathbb{R}^{d+1}} F(w, b) \eta(w, b) d\tau(w, b)$$

where $(w, b) \mapsto \eta(w, b)$ is a function such that $\tilde{\eta}_j = \eta(w_j, b_j)$, and $d\tau(w, b)$ is the probability measure on \mathbb{R}^{d+1} generating the weights (w_j, b_j) .

Thus, when m tends to infinity, we can represent any function f within the RKHS associated to $k(x, x') = \int_{\mathbb{R}^{d+1}} \sigma(w^\top x + b) \sigma(w^\top x' + b) d\tau(w, b)$ as

$$f(x) = \int_{\mathbb{R}^{d+1}} \eta(w, b) \sigma(w^\top x + b) d\tau(w, b),$$

where $\eta : \mathbb{R}^{d+1} \rightarrow \mathbb{R}$ is chosen as to minimize

$$\int_{\mathbb{R}^{d+1}} |\eta(w, b)|^2 d\tau(w, b),$$

the minimum value being equal to the squared RKHS norm of f .

We assume the support of $d\tau$ is compact (bounded and closed). Then the minimum achievable norm is exactly the squared RKHS norm of f , which we denote as $\gamma_2(f)^2$. We denote by \mathcal{H}_2 this RKHS, that is, the set of functions f such that $\gamma_2(f)$ is finite. See [11, Section 2.3] for more details.



Because Dirac measures are not squared integrable, the function $x \mapsto \sigma(w^\top x + b)$, that is, a single neuron, is typically not in the RKHS, which is typically composed of smooth functions. See examples below.

9.3.2 From L_2 -norms to L_1 -norms

Another function space can be defined, where

$$f(x) = \int_{\mathbb{R}^{d+1}} \eta(w, b) \sigma(w^\top x + b) d\tau(w, b),$$

where η is chosen as to minimize


$$\int_{\mathbb{R}^{d+1}} |\eta(w, b)| d\tau(w, b),$$

and $d\tau(w, b)$ is a probability measure on \mathbb{R}^{d+1} . The only difference with the squared RKHS norm above is that we consider the L_1 -norm instead of the squared L_2 -norm of η (with respect to the probability measure $d\tau$). The minimum achievable norm is a specific norm of f , which we denote as $\gamma_1(f)$.

Note that typically, the infimum over all η is not achieved, as, because we use L_1 -norms and the measures $d\mu(w, b) = \eta(w, b) d\tau(w, b)$ can span all measures $d\mu(w, b)$ with finite total variation $\int_{\mathbb{R}^{d+1}} |d\mu(\eta, b)| = \int_{\mathbb{R}^{d+1}} |\eta(w, b)| d\tau(w, b)$, we can reformulate the integral representation of f as

$$f(x) = \int_{\mathbb{R}^{d+1}} \sigma(w^\top x + b) d\mu(w, b),$$

with $d\mu$ a non-negative measure such that the *total variation* $\int_{\mathbb{R}^{d+1}} |d\mu(\eta, b)|$ is minimized. The norm γ_1 is often referred to as the variation norm (see [11] and references therein). We denote by \mathcal{H}_1 the set of functions f such that $\gamma_1(f)$ is finite. We have the following properties (see Table 9.1 for a summary):

- Because of Jensen's inequality, we have $\gamma_1(f) \leq \gamma_2(f)$, and thus $\mathcal{H}_2 \subset \mathcal{H}_1$, that is the space \mathcal{H}_1 contains many more functions.
-  A single neuron is in \mathcal{H}_1 with γ_1 -norm less than one, as the mass of a Dirac is equal to one.

\mathcal{H}_2	\mathcal{H}_1
Hilbert space	Banach space
$\gamma_2(f)^2 = \inf \int_{\mathbb{R}^{d+1}} \eta(w, b) ^2 d\tau(w, b)$ such that $f(x) = \int_{\mathbb{R}^{d+1}} \eta(w, b) \sigma(w^\top x + b) d\tau(w, b)$	$\gamma_1(f) = \inf \int_{\mathbb{R}^{d+1}} \eta(w, b) d\tau(w, b)$ such that $f(x) = \int_{\mathbb{R}^{d+1}} \eta(w, b) \sigma(w^\top x + b) d\tau(w, b)$
Smooth functions Single neurons $\notin \mathcal{H}_2$	Potentially non-smooth functions Single neurons $\in \mathcal{H}_1$

Table 9.1: Summary of properties of the norms γ_1 and γ_2 .

Goals. In this chapter, to describe more precisely the spaces of functions \mathcal{H}_1 and \mathcal{H}_2 , we will consider measures supported on the set $\{(w, b), \|w\|_2 = 1, |b| \leq R\}$ for R such that almost surely $\|x\|_2 \leq R$, and $\sigma(u) = \max\{u, 0\} = (u)_+$ the ReLU activation function, which leads to a reasonably simple analysis.

First, with the assumptions above, if $f(x) = \sum_{j=1}^m \eta_j(w_j^\top x + b_j)_+$, for neurons such that $(w_j, b_j) \in \{(w, b), \|w\|_2 = 1, |b| \leq R\}$ for all $j \in \{1, \dots, m\}$, then $\gamma_1(f) \leq \|\eta\|_1$, and $\gamma_2(f) = \infty$.

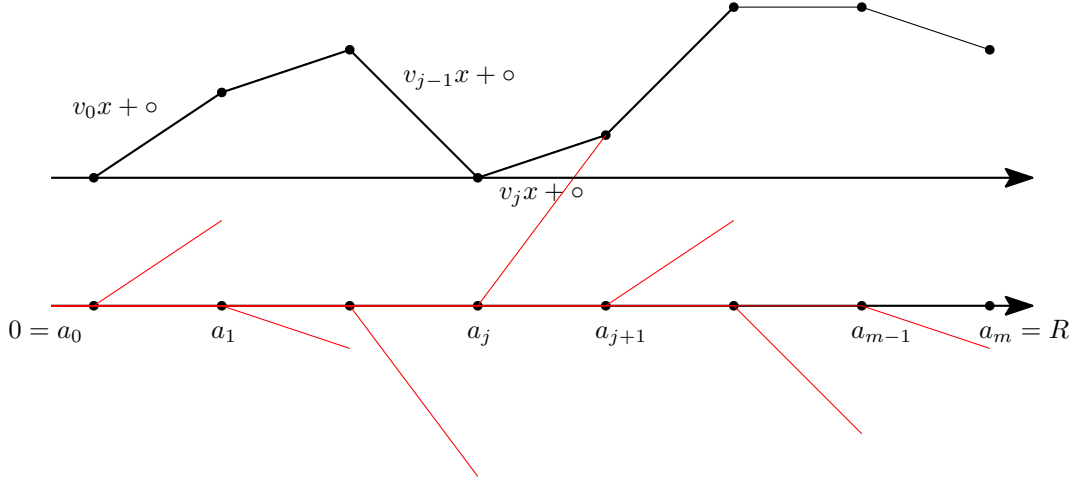
We will show in Section 9.3.5 how the norm γ_1 controls the number of neurons needed to approximate a function from \mathcal{H}_1 , but we now study which functions have finite γ_1 -norm and how functions outside of \mathcal{H}_1 can be approximated by functions in \mathcal{H}_1 .

9.3.3 Variation norm in one dimension

The ReLU activation function is specific and leads to simple approximation properties in the interval $[-R, R]$ for functions $g : [-R, R] \rightarrow \mathbb{R}$. We start by piecewise affine functions, which, given the shape of the ReLU activation should be easy to approximate (and immediately lead to an universal approximation results as all “reasonable” functions can be approximated by piecewise affine functions). See more details in [29, 16].

Piecewise affine functions. We first assume that $g(0) = 0$.

We consider a continuous piecewise affine function on $[-R, R]$ with knots at each $a_j = \frac{j}{m}R$ for $j \in [-m, m] \cap \mathbb{Z}$, so that on $[a_j, a_{j+1}]$, g is affine with slope v_j , for $j \in \{-m, m+1\}$.



Since $g(0) = 0$, we can directly approximate on $[0, R]$, by first starting to fit the function on $[a_0, a_1] = [0, \frac{1}{m}]$, as $\hat{g}_0(x) = v_0(x - a_0)_+$. For $x > a_0$, this approximation has slope v_0 . In order to be correct it on $[a_1, a_2]$ (while not modifying the function on $[a_0, a_1]$, we consider $\hat{g}_1(x) = \hat{g}_0(x) + (v_1 - v_0)(x - a_1)_+$, which is now exact on $[a_0, a_2]$, we can pursue recursively by considering, for $j \in \{1, \dots, m-1\}$

$$\hat{g}_j(x) = \hat{g}_{j-1}(x) + (v_j - v_{j-1})(x - a_j)_+,$$

which is equal to $g(x)$ for $x \in [a_0, a_{j+1}]$. We can thus represent $g(x)$ on $[0, R]$ exactly with $\hat{g}_{m-1}(x)$, which itself is zero on $[-R, 0]$. We have

$$\hat{g}_{m-1}(x) = v_0(x - a_0)_+ + \sum_{j=1}^m (v_j - v_{j-1})(x - a_j)_+,$$

and thus, by construction of the norm γ_1 , we have $\gamma_1(\hat{g}_{m-1}) \leq |v_0| + \sum_{j=1}^{m-1} |v_j - v_{j-1}|$. On the set $[-R, 0]$, we can obtain the same type of approximation with γ_1 -norm less than $|v_{-1}| + \sum_{j=2}^m |v_{-j} - v_{-j+1}|$.

Therefore by summing these two approximations and by the triangular inequality, overall, we get:

$$\gamma_1(g) \leq |v_0| + \sum_{j=1}^{m-1} |v_j - v_{j-1}| + |v_{-1}| + \sum_{j=2}^m |v_{-j} - v_{-j+1}|.$$

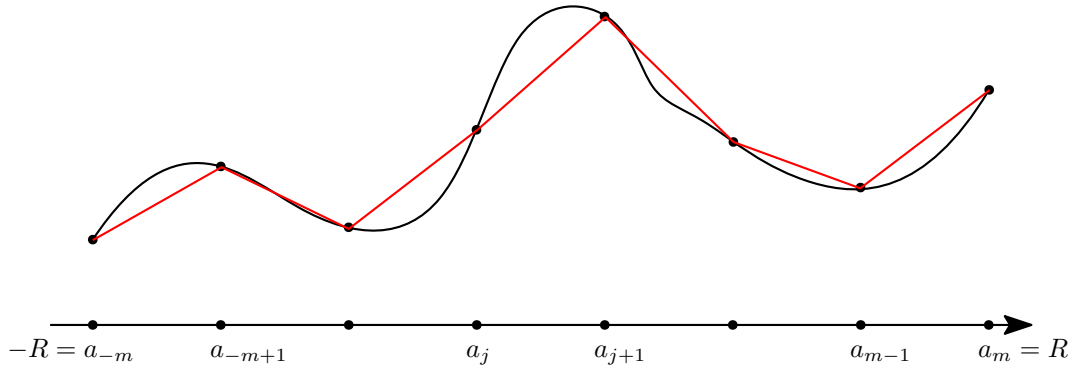
In order to consider functions g without the constraint $g(0) = 0$, we notice that the constant function has norm $\gamma(1) \leq \frac{1}{R}$, by using, for $x \in [-R, R]$, $2R = (x + R)_+ + (-x + R)_+$, and we apply the result above to $g(x) - g(0)$ (which is zero at zero), thus leading to

$$\begin{aligned} \gamma_1(g) &\leq \frac{|g(0)|}{R} + |v_0| + \sum_{j=1}^{m-1} |v_j - v_{j-1}| + |v_{-1}| + \sum_{j=2}^m |v_{-j} - v_{-j+1}| \\ &\leq \frac{|g(0)|}{R} + |v_0 + v_{-1}| + \sum_{j=-m+1}^{m-1} |v_j - v_{j-1}|, \text{ using } |v_0| + |v_{-1}| \leq |v_0 + v_{-1}| + |v_0 - v_{-1}|. \end{aligned}$$

We can then use that g is piecewise-affine with knots at each a_j , to get $v_j = \frac{m}{R}(g(\frac{j+1}{m}R) - g(\frac{j}{m}R))$, and thus:

$$\gamma_1(g) \leq \frac{|g(0)|}{R} + \frac{m}{R}|g(\frac{R}{m}) - g(-\frac{R}{m})| + \frac{m}{R} \sum_{j=-m+1}^{m-1} |g(\frac{j+1}{m}R) - 2g(\frac{j}{m}R) + g(\frac{j-1}{m}R)|.$$

Twice continuously differentiable functions. We consider a twice differentiable function g on $[-R, R]$, it is then the limit of its piecewise interpolation (see illustration below).



Thus, when m tends to infinity, $\frac{m}{R}|g(\frac{R}{m}) - g(-\frac{R}{m})|$ tends to $2|g'(0)|$ while $|g(\frac{j+1}{m}R) - 2g(\frac{j}{m}R) + g(\frac{j-1}{m}R)|$ is asymptotically equivalent to

$$|g(\frac{j}{m}R) + \frac{R}{m}g'(\frac{j}{m}R) + \frac{1}{2}\frac{R^2}{m^2}g''(\frac{j}{m}R) - 2g(\frac{j}{m}R) + g(\frac{j}{m}R) - \frac{R}{m}g'(\frac{j}{m}R) + \frac{1}{2}\frac{R^2}{m^2}g''(\frac{j}{m}R)| \sim |\frac{R^2}{m^2}g''(\frac{j}{m}R)|,$$

and thus we get:

$$\gamma_1(g) \leq \limsup_{m \rightarrow +\infty} \frac{|g(0)|}{R} + 2|g'(0)| + \frac{R}{m} \sum_{j=-m+1}^{m-1} |g''(\frac{j}{m}R)|,$$

which thus leads to using approximations of integral by Riemannian sums:

$$\gamma_1(g) \leq \frac{|g(0)|}{R} + 2|g'(0)| + \int_{-R}^R |g''(x)|dx.$$

In order to allow an extension for non-continuously differentiable functions at 0, we can further use that

$$|g'(0)| \leq |g'(y)| + \int_0^y |g''(x)|dx \leq |g'(y)| + \int_0^R |g''(x)|dx \text{ for any } y \in [0, R],$$

$$\text{leading to } |g'(0)| \leq \frac{1}{R} \int_0^R |g'(y)|dy + \int_0^R |g''(x)|dx \text{ by integration,}$$

$$\text{and } |g'(0)| \leq \frac{1}{2R} \int_{-R}^R |g'(x)|dx + \frac{1}{2} \int_{-R}^R |g''(x)|dx \text{ by symmetry.}$$

Overall, we get the expression

$$\gamma_1(g) \leq \tilde{\gamma}_1(g) = \frac{|g(0)|}{R} + \frac{1}{R} \int_{-R}^R |g'(x)| dx + 2 \int_{-R}^R |g''(x)| dx, \quad (9.2)$$

which shows that if the number of neurons is allowed to grow then the ℓ_1 -norm of the weights remain bounded by the quantity above to exactly represent the function g .

This can be extended to continuous functions which are only twice differentiable almost everywhere with integrable first and second-order derivatives; thus $\tilde{\mathcal{H}}_1 \subset \mathcal{H}_1$ (which corresponds to the norm $\tilde{\gamma}_1$ defined above). Since this space is dense in L_2 (see more general argument below in higher dimension), we obtain that neural networks are universal approximators.

RKHS norm γ_2 in one dimension (♦♦). In one dimension, with w uniform on the unit sphere, that is, $w \in \{-1, 1\}$, and with b uniform on $[-R, R]$, we have the following kernel

$$k(x, x') = \frac{1}{4R} \int_{-R}^R \left((x-b)_+(x'-b)_+ + (-x-b)_+(-x'-b)_+ \right) db.$$

Using the same reasoning as the end of Section 9.3.1, we can get an upper-bound on $\gamma_2(f)$ by decomposing f as

$$f(x) = \int_{-R}^R \eta_+(b)(x-b)_+ \frac{db}{4R} + \int_{-R}^R \eta_-(b)(-x-b)_+ \frac{db}{4R},$$

$$\text{with } \gamma_2(f)^2 \leq \int_{-R}^R \eta_+(b)^2 \frac{db}{4R} + \int_{-R}^R \eta_-(b)^2 \frac{db}{4R}.$$

By using Taylor expansion with integral remainder, we get, for any twice differentiable function f on $[-R, R]$, such that $f(0) = f'(0) = 0$,

$$f(x) = \int_0^R f''(b)(x-b)_+ db + \int_0^R f''(-b)(-x-b)_+ db.$$

Thus, for this function, $\gamma_2(f)^2 \leq 4R \int_{-R}^R f''(b)^2 db$. We can now use

$$\int_{-R}^R \frac{(x-b)_+ - (-x-b)_+}{2R} db = \int_{-R}^R \frac{(x-b)_+ - (b-x)_+}{2R} db = \int_{-R}^R \frac{x}{2R} db = x$$

to get that that $\gamma_2(x \mapsto x)^2 \leq 4$, and use

$$\int_{-R}^R [(x-b)_+ + (-x-b)_+] db = \int_{-R}^x (x-b) db + \int_{-R}^{-x} (-x-b) db = \frac{(x-R)^2}{2} + \frac{(x+R)^2}{2} = x^2 + R^2,$$

to get that $\gamma_2(x \mapsto x^2 + R^2)^2 \leq 16R^2$.

Thus by considering $\tilde{f}(x) = f(x) - f'(0)x - \frac{f(0)}{R^2}(x^2 + R^2)$, we have:

$$\begin{aligned}
\gamma_2(f) &\leq \sqrt{4R \int_{-R}^R \tilde{f}''(b)^2 db} + 2|f'(0)| + \frac{|f(0)|}{R} \\
&= \sqrt{4R \int_{-R}^R |f''(b) - 2f(0)/R^2|^2 db} + 2|f'(0)| + \frac{|f(0)|}{R} \\
&\leq \sqrt{4R \int_{-R}^R |f''(b)|^2 db} + \sqrt{4R \int_{-R}^R |2f(0)/R^2|^2 db} + 2|f'(0)| + \frac{|f(0)|}{R} \\
&= \sqrt{4R \int_{-R}^R |f''(b)|^2 db} + 4\sqrt{2} \frac{|f(0)|}{R} + 2|f'(0)| + \frac{|f(0)|}{R},
\end{aligned}$$

leading to the upper-bound

$$\gamma_2(g)^2 \leq \tilde{\gamma}_2(g)^2 = 36 \frac{f(0)^2}{R^2} + 16f'(0)^2 + 16R \int_{-R}^R f''(x)^2 dx. \quad (9.3)$$

The main difference with $\tilde{\gamma}_1$ is that the second-derivative is penalized by an L_2 -norm and not by and L_1 -norm, and that this L_2 -norm can be infinite when the L_1 -norm is finite, the classical example being for the hidden neuron functions $(x - b)_+$. Note that we only derive an upper-bound on γ_2 , but similar lower bounds could also be derived.

⚠ The RKHS is combining infinitely many hidden neuron functions $(x - b)_+$, none of them are inside the RKHS,

⚠ This smoothness penalty does not allow the ReLU to be part of the RKHS. However, this is still an universal penalty (as the set of functions with squared integrable second derivative is dense in L_2).

9.3.4 Variation norm in arbitrary dimension

If we assume that f is continuous on the ball of center zero and radius R , then the Fourier transform $\hat{f}(\omega) = \int_{\mathbb{R}^d} f(x) e^{-i\omega^\top x} dx$ is defined everywhere, and we can write

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^\top x} d\omega.$$

In order to compute an upper-bound on $\gamma_1(f)$, it suffices to upper-bound for each $\omega \in \mathbb{R}^d$, $\gamma_1(x \mapsto e^{i\omega^\top x})$, which is easy because we have the representation from Section 9.3.3 and Eq. (9.2) applied to $g : u \mapsto e^{iu\|\omega\|_2}$: for $u \in [-R, R]$,

$$e^{iu\|\omega\|_2} = \int_{-R}^R \eta_+(b)(u - b)_+ db + \int_{-R}^R \eta_-(b)(-u - b)_+ db,$$

with $\int_{-R}^R |\eta_+(b)| db + \int_{-R}^R |\eta_-(b)| db \leq \frac{|g(0)|}{R} + \frac{1}{R} \int_{-R}^R |g'(x)| dx + 2 \int_{-R}^R |g''(x)| dx = \frac{1}{R} + 2\|\omega\|_2 + 4R\|\omega\|_2^2$ (which is the norm defined in Eq. (9.2)). We can therefore decompose

$$e^{i\omega^\top x} = e^{i(x^\top \omega / \|\omega\|_2) \|\omega\|_2} = \int_{-R}^R \eta_+(b)(x^\top (\omega / \|\omega\|_2) - b)_+ db + \int_{-R}^R \eta_-(b)(x^\top (-\omega / \|\omega\|_2) - b)_+ db,$$

with weights being in the correct constraint set (unit norm for w 's and $|b| \leq R$, leading to

$$\gamma_1(x \mapsto e^{i\omega^\top x}) \leq \tilde{\gamma}_1(x \mapsto e^{i\omega^\top x}) \leq \frac{1}{R} + 2\|\omega\|_2 + 4R\|\omega\|_2^2 = \frac{1}{R}(1 + 2R\|\omega\|_2)^2.$$

Thus, we obtain

$$\gamma_1(f) \leq \frac{1}{(2\pi)^d} \frac{1}{R} \int_{\mathbb{R}^d} |\hat{f}(\omega)|(1 + 2R^2\|\omega\|_2^2) d\omega.$$

Given a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $\int_{\mathbb{R}^d} |\hat{f}(\omega)| d\omega$ is a measure of smoothness of f , and so $\gamma_1(f)$ being finite imposes that f and all second-order derivatives of f have this form of smoothness. See [63] for more details and below for a relationship with Sobolev spaces.

Precise rates of approximation (♦). In this section, we will relate the space \mathcal{H}_1 to Sobolev spaces, by considering $s > d/2$ (to make sure the integral below exists), and bounding using Cauchy-Schwarz inequality:

$$\begin{aligned} \gamma_1(f) &\leq \frac{1}{(2\pi)^d} \frac{1}{R} \int_{\mathbb{R}^d} |\hat{f}(\omega)|(1 + 2R^2\|\omega\|_2^2) d\omega = \frac{1}{(2\pi)^d} \frac{1}{R} \int_{\mathbb{R}^d} |\hat{f}(\omega)|(1 + 2R^2\|\omega\|_2^2)^{1+s/2} \frac{d\omega}{(1 + 2R^2\|\omega\|_2^2)^{s/2}} \\ &\leq \frac{1}{(2\pi)^d} \frac{1}{R} \sqrt{\int_{\mathbb{R}^d} |\hat{f}(\omega)|^2 (1 + 2R^2\|\omega\|_2^2)^{2+s} d\omega} \sqrt{\int_{\mathbb{R}^d} \frac{d\omega}{(1 + 2R^2\|\omega\|_2^2)^s}}, \end{aligned}$$

which is a constant times $\sqrt{\int_{\mathbb{R}^d} |\hat{f}(\omega)|^2 (1 + 2R^2\|\omega\|_2^2)^{2+s} d\omega}$, which is exactly the Sobolev norm from Chapter 7, with $s + 2$ derivatives (which is an RKHS).

Thus, all approximation properties from Chapter 7 apply. See Chapter 7 for precise rates. Note however, that, *using this reasoning*, if we start from a Lipschitz-continuous function then to approximate it up to $L_2(dx)$ -norm ε requires a γ_1 -norm exploding as $\varepsilon^{-(s+1)} \geq \varepsilon^{-(d/2+1)}$ (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{H}_1 (neural networks) is not really more advantageous than using kernel methods (such as functions in \mathcal{H}_2). This changes drastically when such linear structures are present, as we show below.

Adaptivity to linear structures (♦). We consider a target function f depends only a r -dimensional projection of the data, that is, f is of the form $f(x) = g(V^\top x)$, where $V \in \mathbb{R}^{d \times r}$ is full rank and has all singular values less than 1, and $g : \mathbb{R}^r \rightarrow \mathbb{R}$. Without loss of generality we can assume that V is a rotation matrix. Then if $\gamma_1(g)$ is finite, it can be written as

$$g(z) = \int_{\mathbb{R}^{r+1}} (w^\top z + b)_+ d\mu(w, b),$$

with $d\mu$ supported on $\{(w, b) \in \mathbb{R}^{r+1}, \|w\|_2 = 1, |b| \leq R\}$, and $\gamma_1(g) = \int_{\mathbb{R}^{r+1}} |d\mu(w, b)|$. We then have:

$$f(x) = g(V^\top x) = \int_{\mathbb{R}^{r+1}} ((Vw)^\top x + b)_+ d\mu(w, b),$$

leading to $\gamma_1(f) \leq \int_{\mathbb{R}^{r+1}} |d\mu(w, b)| = \gamma_1(g)$ (because $\|Vw\|_2 = 1$). Thus the approximation properties of g translate to f , and thus we pay only the price of these r dimensions and not of all d variables, *without* the need to know V in advance. For example, (a) if g has more than $r/2 + 2$ squared integrable derivatives, then $\gamma_1(g)$ and thus $\gamma_1(f)$ is finite, or (b) if g is Lipschitz-continuous, then both g and f can be approached in $L_2(dx)$ with error ε with a function with γ_1 -norm of order $\varepsilon^{-(r/2+1)}$, thus escaping the curse of dimensionality. See [11] for more details.



Kernel methods do not have such adaptivity. In other words, using the ℓ_2 -norm instead of the ℓ_1 -norm on the output weights, leads to worse performance.

9.3.5 From the variation norm to a finite number of neurons

Given a measure $d\mu$ on \mathbb{R}^d , and a function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\gamma_1(g)$ is finite, we would like to find a set of m neurons $(w_j, b_j) \in \mathcal{V} \subset \mathbb{R}^{d+1}$ (which is the compact support of all measures that we consider), such that the associated function defined through

$$f(x) = \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j)$$

is close to g .

If the input weights are fixed, then the bound on $\gamma_1(g)$ translates to a bound $\|\eta\|_1 \leq \gamma_1(g)$. The set of such functions f is the convex hull of functions $s_j \gamma_1(g) \sigma(w_j^\top x + b_j)$, for $s_j \in \{-1, 1\}$. Thus, we are faced with the problem of approximating an elements of a convex hull as an explicit linear combination of extreme points, if possible with as few extreme points as possible.

In finite dimension, Carathéodory's theorem tells that the number of such extreme points can be taken to be 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 $\min_{g \in \mathcal{H}_1} \|f - g\|_{L_2(dx)}^2$ such that $\gamma_1(f) \leq \gamma_1(g)$, whose solution is $f = g$, with an algorithm that constructs an approximate solution from extreme points. This will be achieved by the Frank-Wolfe algorithm (a.k.a. conditional gradient algorithm). This algorithm is applicable more generally, for more details, see [56, 10].

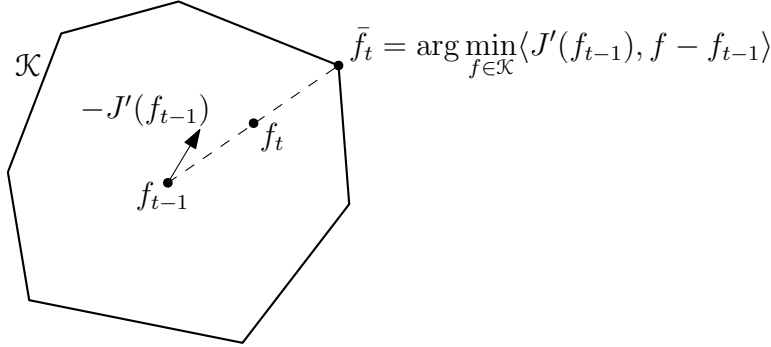
Frank-Wolfe algorithm. We thus make a detour by considering an algorithm defined in a Hilbert space \mathcal{H} , such that \mathcal{K} is a bounded convex set, and J a convex smooth function from \mathcal{H} to \mathbb{R} , that is such that there exists a gradient function $J' : \mathcal{H} \rightarrow \mathcal{H}$ such that for all elements f, g of \mathcal{H} :

$$J(g) + \langle J'(g), h - g \rangle_{\mathcal{H}} \leq J(f) \leq J(g) + \langle J'(g), h - g \rangle_{\mathcal{H}} + \frac{L}{2} \|h - g\|_{\mathcal{H}}^2.$$

The goal is to minimize J on the bounded convex set \mathcal{K} , with an algorithm that only requires to access the set \mathcal{K} through a “linear minimization” oracle (i.e., through maximizing linear functions), as opposed to the projection oracle that we required in Chapter 5.

We consider the following recursive algorithm, started from a vector $f_0 \in \mathcal{K}$:

$$\begin{aligned}\bar{f}_t &\in \arg \min_{f \in \mathcal{K}} \langle J'(f_{t-1}), f - f_{t-1} \rangle_{\mathcal{H}}, \\ f_t &= \frac{t-1}{t+1} f_{t-1} + \frac{2}{t+1} \bar{f}_t = f_{t-1} + \frac{2}{t+1} (\bar{f}_t - f_{t-1}).\end{aligned}$$



Because \bar{f}_t is obtained by minimizing a linear function on a bounded convex set, we can restrict the minimizer \bar{f}_t to be extreme points of \mathcal{K} , so that, f_t is the convex combination of t such extreme points $\bar{f}_1, \dots, \bar{f}_t$ (note that the first point f_0 disappears). We now show that

$$J(f_t) - \inf_{f \in \mathcal{K}} J(f) \leq \frac{2L}{t+1} \text{diam}_{\mathcal{H}}(\mathcal{K})^2.$$

Proof of convergence rate (♦). This is simply obtained by using smoothness:

$$\begin{aligned}J(f_t) &\leq J(f_{t-1}) + \langle J'(f_{t-1}), f_t - f_{t-1} \rangle_{\mathcal{H}} + \frac{L}{2} \|f_t - f_{t-1}\|_{\mathcal{H}}^2 \\ &= J(f_{t-1}) + \frac{2}{t+1} \langle J'(f_{t-1}), \bar{f}_t - f_{t-1} \rangle_{\mathcal{H}} + \frac{4}{(t+1)^2} \frac{L}{2} \|\bar{f}_t - f_{t-1}\|_{\mathcal{H}}^2 \\ &\leq J(f_{t-1}) + \frac{2}{t+1} \min_{f \in \mathcal{K}} \langle J'(f_{t-1}), f - f_{t-1} \rangle_{\mathcal{H}} + \frac{4}{(t+1)^2} \frac{L}{2} \text{diam}_{\mathcal{H}}(\mathcal{K})^2.\end{aligned}$$

By convexity of J , we have for all $f \in \mathcal{K}$, $J(f) \geq J(f_{t-1}) + \langle J'(f_{t-1}), f - f_{t-1} \rangle_{\mathcal{H}}$, leading to $\inf_{f \in \mathcal{K}} J(f) \geq J(f_{t-1}) + \inf_{f \in \mathcal{K}} \langle J'(f_{t-1}), f - f_{t-1} \rangle_{\mathcal{H}}$. Thus, we get

$$\begin{aligned}J(f_t) - \inf_{f \in \mathcal{K}} J(f) &\leq [J(f_{t-1}) - \inf_{f \in \mathcal{K}} J(f)] \frac{t-1}{t+1} + \frac{4}{(t+1)^2} \frac{L}{2} \text{diam}_{\mathcal{H}}(\mathcal{K})^2, \text{ leading to} \\ t(t+1)[J(f_t) - \inf_{f \in \mathcal{K}} J(f)] &\leq (t-1)t[J(f_{t-1}) - \inf_{f \in \mathcal{K}} J(f)] + 2L \text{diam}_{\mathcal{H}}(\mathcal{K})^2 \\ &\leq 2Lt \text{diam}_{\mathcal{H}}(\mathcal{K})^2 \text{ by using a telescoping sum,}\end{aligned}$$

and thus $J(f_t) - \inf_{f \in \mathcal{K}} J(f) \leq \frac{2L}{t+1} \text{diam}_{\mathcal{H}}(\mathcal{K})^2$, as claimed earlier.

Application to approximate representations with a finite number of neurons. We can apply this to $\mathcal{H} = L_2(dx)$ and $J(f) = \|f - g\|_{L_2(dx)}^2$, leading to $L = 2$, with $\mathcal{K} = \{f \in L_2(dx), \gamma_1(f) \leq \gamma_1(g)\}$ for which the set of extreme points are exactly single neurons $s\sigma(w^\top \cdot + b)$ scaled by $\gamma_1(g)$, and with an extra sign $s \in \{-1, 1\}$.

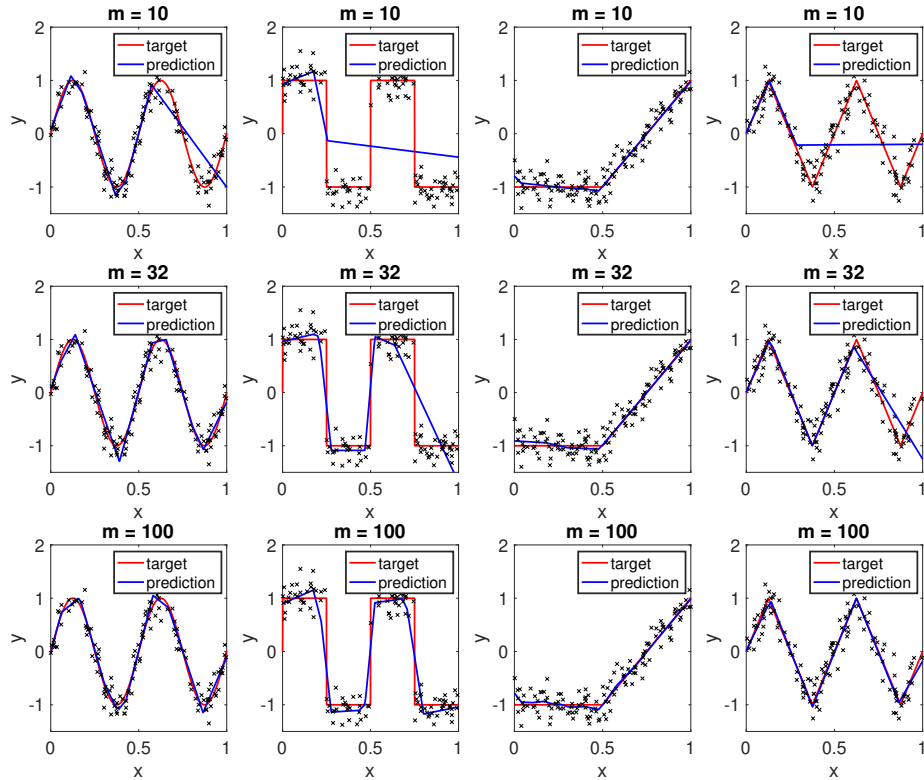
We thus obtain after t steps a representation of f with t neurons for which

$$\|f - g\|_{L_2(dx)}^2 \leq \frac{4L\gamma_1(g)^2}{t+1} \sup_{(w,b) \in \mathcal{K}} \|\sigma(w^\top \cdot + b)\|_{L_2(dx)}^2.$$

Thus, it is sufficient to have t of order $O(\gamma_1(g)^2/\varepsilon^2)$ to achieve $\|f - g\|_{L_2(dx)} \leq \varepsilon$. Therefore the norm $\gamma_1(g)$ directly controls the approximability of the function g by a finite number of neurons, and tell us how many neurons should be used for a given target function.

9.4 Experiments

We consider the same experimental set-up as Section 7.7, that is, one-dimensional problems to highlight the adaptivity of neural networks methods to the regularity of the target function, with smooth targets and non-smooth targets. We consider several values for the number m of hidden neurons, and we consider a neural network with ReLU activation functions and an additional global constant term. Training is done by stochastic gradient descent with a small constant step-size and a random initialization.

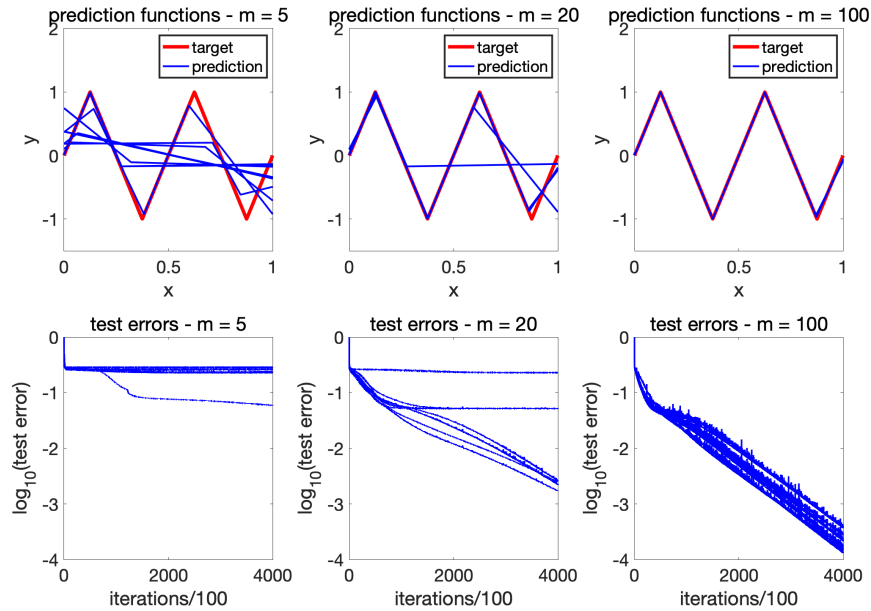


Note that for small m , while a neural network with the same number of hidden neurons could fit the data better, optimization is not successful (that is, SGD gets trapped in a bad local minimum).

9.5 Global convergence of gradient descent for infinite widths (◆◆)

In this section, we will provide intuitive arguments of the proof of global convergence of gradient descent algorithms for one-hidden layer when the number of hidden neurons is infinite (without any convergence rates, hence it is only a “qualitative” result). Precise results with all regularity assumptions can be found in [35].

The goal of this section is to explain the empirical observation already made in Section 9.4 that gradient descent can be trapped in local minima. We show an additional experiments below for the same one-dimensional set-up, where we compare several runs of stochastic gradient descent (SGD) where observations are only seen once (so no overfitting is possible) and with random initializations. We show the estimated predictors, as well as the testing errors for problems with zero label noise (that is, the Bayes rate is zero), with 10 different initializations.



We see that when $m = 5$ (which is sufficient to attain zero testing errors), small errors are never achieved. With $m = 20$ neurons, then SGD finds the optimal predictor for most restarts. When $m = 100$, all restarts have the desired behaviors. In this section, we essentially show that this is true for $m = +\infty$.

See <https://francisbach.com/gradient-descent-neural-networks-global-convergence/>.

9.6 Extensions

The fully-connected single-hidden layer neural networks is far from what is being used in practice. 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 networks 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:

$$\begin{aligned} y^{(k)} &= (W^{(k)})^\top x^{(k-1)} \\ x^{(k)} &= \sigma(y^{(k)}), \end{aligned}$$

where $W^{(\ell)}$ is the matrix of weights for layer k . For these models, obtaining simple and powerful theoretical results is still an active area of research. See, e.g., [68, 70].

- **Convolutional neural networks:** In order to be able to tackle data of large size and to improve performances, it is important to leverage the prior knowledge about the structure of the typical data to process. For instance, for signal, images or videos, it is important to take into account the translation invariance (up to boundary issues) of the domain. This is done by constraining the linear operators involved in the linear part of neural networks to respect some form of translation invariance, and thus to use convolutions. See [48] for details.

Part III

Special topics

Chapter 10

Implicit bias of gradient descent

Chapter summary

- Implicit regularization of gradient descent: for linear models, when there are several minimizers, gradient descent techniques tend to converge to the one with minimum Euclidean norm.
- Double descent: for unregularized models learned with gradient descent techniques, when the number of parameters grows, the performance can exhibit a second descent after the test error blows up after the number of parameters goes beyond the number of observations.
- Global convergence of gradient descent for two-layer neural networks: in the infinite width limit, gradient descent exhibits some globally convergent behavior for a non-convex problem.

In this chapter, we will cover three recent topics within learning theory, all partially related to high-dimensional models (such as neural networks) in the “over-parameterized” regime, where the number of parameters is larger than the number of observations.



The number of parameters is not what characterizes in general the generalization capabilities of regularized learning methods.

10.1 Implicit bias of gradient descent

Given an optimization problem whose aim is to minimize some function $F(\theta)$ over some $\theta \in \mathbb{R}^d$, if there is a unique global minimizer θ_* , then the goal of optimization algorithms is to find this minimizer, that is, we want that the t -th iterate θ_t converges to that θ_* . When there are multiple minimizers (thus for a function which cannot be strongly convex), we showed only that $F(\theta_t) - \inf_{\theta \in \mathbb{R}^d} F(\theta)$ is converging to zero (and only if a minimizer exists, see Chapter 5).

With some extra assumptions, we can show that the algorithm is converging to one of the multiple minimizers of F (note that when F is convex, this set is also convex). But which one? This is what is referred to as the implicit regularization properties of optimization algorithms, and here gradient descent and its variants.

This is interesting in machine learning because, when $F(\theta)$ is the empirical loss on n observations, and d is much larger than n , **and no regularization is used**, there are multiple minimizers, and an arbitrary empirical risk minimizer is not expected to work well on unseen data. A classical solution is to use explicit regularization (e.g., ℓ_2 -norms like in Chapter 3 and Chapter 7, or ℓ_1 -norms like in Chapter 8). In this section, we show that optimization algorithms have a similar regularizing effect. In a nutshell, gradient descent usually leads to minimum ℓ_2 -norm solutions. 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.

10.1.1 Least-squares

We consider $F(\theta) = \frac{1}{2n} \|y - \Phi\theta\|_2^2$, with $\Phi \in \mathbb{R}^{n \times d}$ such that $d > n$ and (for simplicity) $\Phi\Phi^\top \in \mathbb{R}^{n \times n}$ invertible (this is the kernel matrix). There are thus infinitely many (a whole affine subspace) solutions such that $y = \Phi\theta$, since the column space of Φ is the entire space \mathbb{R}^n and θ has dimension $d > n$. We apply gradient descent with step-size $\gamma \leq \frac{1}{\lambda_{\max}(\frac{1}{n}\Phi^\top\Phi)} = \frac{1}{\lambda_{\max}(\frac{1}{n}\Phi\Phi^\top)}$ starting from $\theta_0 = 0$. Thus, for any θ solution of $y = \Phi\theta$, we have, as shown in Chapter 5:

$$\theta_t - \theta = \left(I - \frac{\gamma}{n}\Phi^\top\Phi\right)^t(\theta_0 - \theta) = -\left(I - \frac{\gamma}{n}\Phi^\top\Phi\right)^t\theta,$$

leading to

$$\theta_t = \left[I - \left(I - \frac{\gamma}{n}\Phi^\top\Phi\right)^t\right]\theta.$$

Note that it is not entirely obvious that the formula above is independent of the choice of θ (but it is).

If $\Phi = U \text{Diag}(s)V^\top$ is the SVD decomposition of Φ , where $U \in \mathbb{R}^{n \times n}$ is orthonormal, and $V \in \mathbb{R}^{d \times n}$ has orthonormal columns and $s \in (\mathbb{R}_+^*)^n$, we can take $\theta = V \text{Diag}(s)^{-1}U^\top y$ as one of the solutions (since then $\Phi\theta = U \text{Diag}(s)V^\top V \text{Diag}(s)^{-1}U^\top y = U \text{Diag}(s) \text{Diag}(s)^{-1}U^\top y = UU^\top y = y$) and get:

$$\theta_t = V \text{Diag}((1 - (1 - \gamma s_i^2/n)^t)s_i^{-1})U^\top y.$$

Since each $s_i > 0$, and $\gamma_i \leq \frac{n}{\max_i s_i^2}$, we have

$$s_i^{-1} \leq (1 - (1 - \gamma s_i^2/n)^t)s_i^{-1} \leq s_i^{-1}(1 - (1 - \gamma \min_i s_i^2/n)^t),$$

and thus

$$\|\theta_t - V \text{Diag}(s)^{-1}U^\top y\|_2 \leq (1 - \gamma \min_i s_i^2/n)^t \|V \text{Diag}(s)^{-1}U^\top y\|_2.$$

We thus get linear convergence to $V \text{Diag}(s)^{-1}U^\top y$, which happens to be the minimum ℓ_2 -norm solution, because all solutions to $y = \Phi\theta$ can be written as $V \text{Diag}(s)^{-1}U^\top y$ plus a vector which is orthogonal to the column space of V .

Moreover, with $\gamma_i = \frac{n}{\max_i s_i^2}$ (largest allowed step-size), we get a rate of $\left(1 - \gamma \frac{\min_i s_i^2}{\max_i s_i^2}\right)^t$.

Lojasiewicz's inequality (♦). It turns out that linear convergence here can be shown directly for any L -smooth function, for which we have the so-called Lojasiewicz's inequality:

$$\forall \theta \in \mathbb{R}^d, \quad F(\theta) - F(\theta_*) \leq \frac{1}{2\mu} \|F'(\theta)\|_2^2, \quad (10.1)$$

for some $\mu > 0$.

We have seen in Chapter 5 that this is a consequence of μ -strong-convexity, but this can be satisfied without strong convexity. For example, for any least-squares example, we have, for any minimizer θ_* :

$$\|F'(\theta)\|_2^2 = \left\| \frac{1}{n} \Phi^\top \Phi (\theta - \theta_*) \right\|_2^2 = \frac{1}{n^2} (\theta - \theta_*)^\top \Phi^\top \Phi \Phi^\top \Phi (\theta - \theta_*) \geq \frac{\lambda_{\min}^+(\Phi \Phi^\top)}{n^2} (\theta - \theta_*)^\top \Phi^\top \Phi (\theta - \theta_*),$$

where $\lambda_{\min}^+(\Phi \Phi^\top) = \lambda_{\min}^+(\Phi^\top \Phi)$ is the smallest non-zero eigenvalue of $\Phi \Phi^\top$ (which is also the one of $\Phi^\top \Phi$). Thus, we have

$$\|F'(\theta)\|_2^2 \geq \frac{\lambda_{\min}^+(K)}{n^2} \|\Phi(\theta - \theta_*)\|_2^2 = \frac{2\lambda_{\min}^+(K)}{n} [F(\theta) - F(\theta_*)].$$

Thus, Eq. (10.1) is satisfied with $\mu = \frac{1}{n} \lambda_{\min}^+(K)$, where $K = \Phi \Phi^\top \in \mathbb{R}^{n \times n}$ is the kernel matrix. Note that this includes also the strongly-convex case since $\lambda_{\min}^+(\Phi^\top \Phi) \geq \lambda_{\min}(\Phi^\top \Phi)$.

When Eq. (10.1) is satisfied, we have for the t -th iterate of gradient descent with step-size $\gamma = 1/L$, following the analysis of Chapter 5:

$$F(\theta_t) - F(\theta_*) \leq F(\theta_{t-1}) - F(\theta_*) - \frac{1}{2L} \|F'(\theta_{t-1})\|_2^2 \leq \left(1 - \frac{\mu}{L}\right) [F(\theta_{t-1}) - F(\theta_*)].$$

Moreover, we can then show that the iterates x_t are also converging to a minimizer of F (see, [24, 61] for more details).

Alternative proof. If started at $\theta_0 = 0$, gradient descent techniques (stochastic or not) will always have iterates θ_t which are linear combinations of row of Φ , that is, of the form $\theta_t = \Phi^\top \alpha_t$ for some $\alpha_t \in \mathbb{R}^n$. This is an alternative algorithmic version of the representer theorem from Chapter 7.

If the method is converging, then we must have $\Phi \theta_t$ converging to y (because the standard squared Euclidean norm is strongly-convex, and $\Phi \theta$ is unique while θ may not be), and thus $\Phi \Phi^\top \alpha_t$ is converging to y . If $K = \Phi \Phi^\top$ is invertible, this means that α_t is converging to $K^{-1}y$, and thus $\theta_t = \Phi^\top \alpha_t$ is converging to $\Phi^\top K^{-1}y$.

It turns out that this is exactly the minimum ℓ_2 -norm solution as, by standard Lagrangian duality:

$$\begin{aligned} \inf_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\theta\|_2^2 \text{ such that } y = \Phi \theta &= \inf_{\theta \in \mathbb{R}^d} \sup_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|\theta\|_2^2 + \alpha^\top (y - \Phi \theta) \\ &= \sup_{\alpha \in \mathbb{R}^n} \alpha^\top y - \frac{1}{2} \|\Phi^\top \alpha\|_2^2 \text{ with } \theta = \Phi^\top \alpha \text{ at optimum,} \\ &= \sup_{\alpha \in \mathbb{R}^n} \alpha^\top y - \frac{1}{2} \alpha^\top K \alpha. \end{aligned}$$

The last problem is exactly solved for $\alpha = K^{-1}y$. Note that in Chapter 7, we used this formula for function interpolation to compare different RKHSs.

10.1.2 Separable classification

We now consider logistic regression, that is,

$$F(\theta) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \varphi(x_i)^\top \theta)),$$

with $\Phi \in \mathbb{R}^{n \times d}$ the design matrix such that $d > n$ and $\Phi \Phi^\top$ invertible.

Maximum margin classifier. Since $\Phi \Phi^\top$ is invertible, there exists $\eta \in \mathbb{R}^d$ of unit norm such that $\forall i \in \{1, \dots, n\}$, $y_i \varphi(x_i)^\top \eta > 0$. We denote by η_* the one such that

$$\min_{i \in \{1, \dots, n\}} y_i \varphi(x_i)^\top \eta$$

is maximal (and thus strictly positive). That is, η_* solves the following problem, which can be rewritten as, using Lagrange duality:

$$\begin{aligned} \sup_{\|\eta\|_2 \leq 1} \min_{i \in \{1, \dots, n\}} y_i \varphi(x_i)^\top \eta &= \sup_{\|\eta\|_2 \leq 1, t \in \mathbb{R}} t \text{ such that } \forall i \in \{1, \dots, n\}, y_i \varphi(x_i)^\top \eta \geq t \\ &= \inf_{\alpha \in \mathbb{R}_+^n} \sup_{\|\eta\|_2 \leq 1, t \in \mathbb{R}} t + \sum_{i=1}^n \alpha_i (y_i \varphi(x_i)^\top \eta - t) \\ &= \inf_{\alpha \in \mathbb{R}_+^n} \left\| \sum_{i=1}^n \alpha_i y_i \varphi(x_i) \right\|_2 \text{ such that } \sum_{i=1}^n \alpha_i = 1, \end{aligned}$$

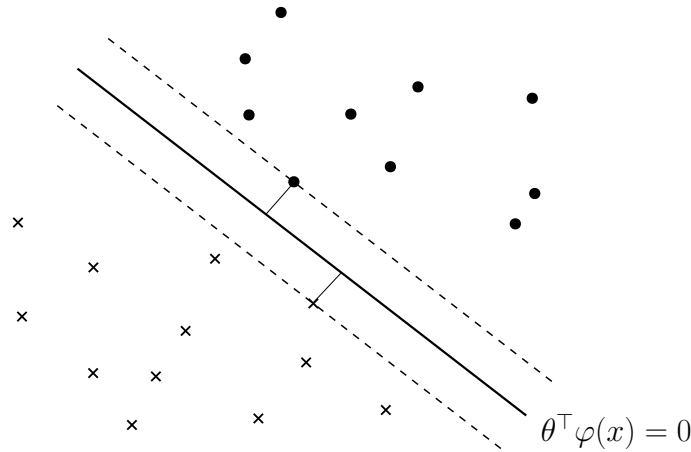
with $\eta \propto \sum_{i=1}^n \alpha_i y_i \varphi(x_i)$ at optimum. Moreover, by complementary slackness non-negative α_i is non zero only for i attaining the minimum in $\min_{i \in \{1, \dots, n\}} y_i \varphi(x_i)^\top \eta$.

Moreover, because of homogeneity, we want $\min_{i \in \{1, \dots, n\}} y_i \varphi(x_i)^\top \eta$ large and $\|\eta\|_2$ small, and we can decide to constrain the first and minimize the second one. In other words, we can see η_* as the direction of the solution θ_* of:

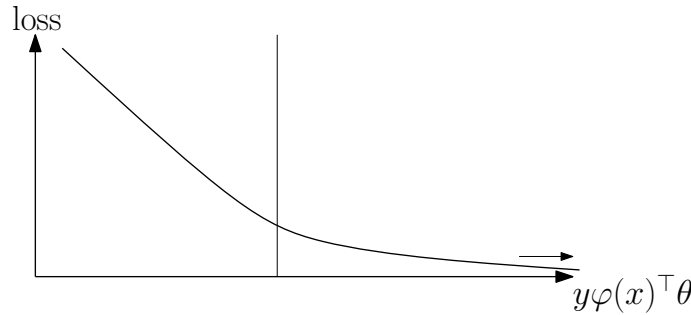
$$\begin{aligned} \inf_{\theta \in \mathbb{R}^d} \frac{1}{2} \|\theta\|_2^2 \text{ such that } \text{Diag}(y) \Phi \theta \geq 1_n &= \inf_{\theta \in \mathbb{R}^d} \sup_{\alpha \in \mathbb{R}_+^n} \frac{1}{2} \|\theta\|_2^2 + \alpha^\top (1_n - \text{Diag}(y) \Phi \theta) \\ &= \sup_{\alpha \in \mathbb{R}_+^n} \alpha^\top 1_n - \frac{1}{2} \|\Phi^\top \text{Diag}(y) \alpha\|_2^2 \text{ with } \theta = \Phi^\top \text{Diag}(y) \alpha \text{ at optimum.} \end{aligned}$$

Note that above, $\text{Diag}(y) \Phi \theta \geq 1_n$ is the compact formulation of $\forall i \in \{1, \dots, n\}, y_i \varphi(x_i)^\top \theta \geq 1$.

The θ_* above is the solution of the separable SVM with vanishing regularization parameter, that is, of $\frac{1}{2} \|\theta\|_2^2 + C \sum_{i=1}^n (1 - y_i \varphi(x_i)^\top \theta)_+$ for C large enough.



Divergence and convergence of directions. The function F has an infimum equal to zero, which is not attained. However, for any sequence θ_t such that all $y_i \varphi(x_i)^\top \theta_t$ tend to infinity, we have $F(\theta_t) \rightarrow \inf_{\theta \in \mathbb{R}^d} F(\theta) = 0$.



In such a situation, gradient descent cannot converge to a point, and, to achieve small values of F , it has to diverge. It turns out that it diverges along a direction, that is, $\|\theta_t\|_2 \rightarrow +\infty$, with $\frac{1}{\|\theta_t\|_2} \theta_t \rightarrow \eta$ for some $\eta \in \mathbb{R}^d$ of unit ℓ_2 -norm. See [50] for a proof. Here, we just show what the vector η is.

The gradient $F'(\theta)$ is equal to $F'(\theta) = -\frac{1}{n} \sum_{i=1}^n \frac{\exp(-y_i \varphi(x_i)^\top \theta)}{1 + \exp(-y_i \varphi(x_i)^\top \theta)} y_i \varphi(x_i)$.

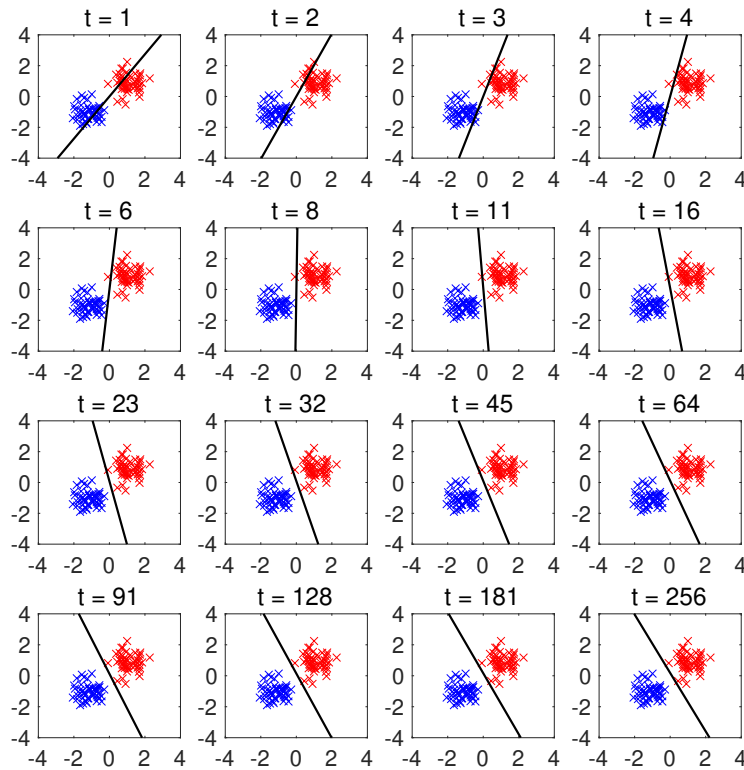
Asymptotically, θ_t will behave as $\|\theta_t\| \eta$, with $\|\theta_t\|_2$ tending to infinity. Thus, because we have a sum of exponentials with scale that goes to infinity, the dominant term in $F'(\theta_t)$ corresponds to the indices i for which $-y_i \varphi(x_i)^\top \eta$ is largest. Moreover, all of these values have to be negative (indeed we can only attain zero loss for well-classified training data). We denote by I this set. Thus,

$$F'(\theta_t) \sim -\frac{1}{n} \sum_{i \in I} y_i \exp(-\|\theta_t\|_2 y_i \varphi(x_i)^\top \eta) \varphi(x_i).$$

Moreover, we must have $F'(\theta_t)$ along $-u$ to diverge in the direction u , thus u has to be proportional to a vector $\sum_{i \in I} \alpha_i y_i \varphi(x_i)$, where $\alpha \geq 0$, and $\alpha_i = 0$ as soon as i is not among the minimizers of $y_i \varphi(x_i)^\top \eta$.

This is exactly the optimality condition for η_* above. Thus $\eta = \eta_*$.

Overall, we obtain a classifier corresponding to a minimum ℓ_2 -norm. See examples in two dimensions below.



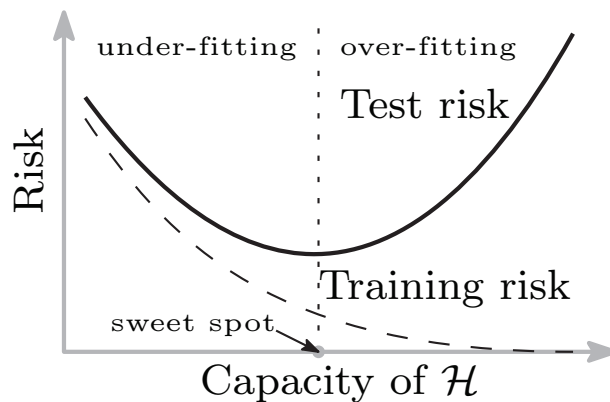
General result. The result above extends to more general situation beyond linear classification. See [69].

10.2 Double descent

In this section, we consider a recent and interesting phenomenon described in several recent works [20, 72, 44, 54].

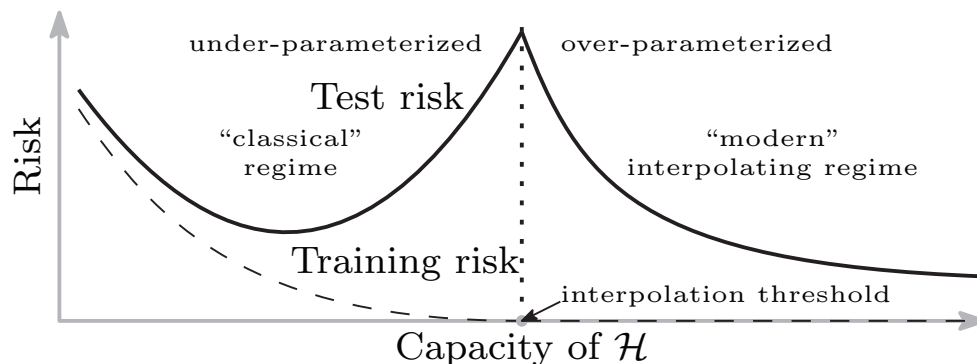
10.2.1 The double descent phenomenon

As seen in Chapter 2 and Chapter 4, typical learning curves look like the one below (figure taken from [20]):



Typically the “capacity” of the space of functions \mathcal{H} is controlled either by the number of parameters, either by some norms of its parameters. In particular, at the extreme right of the curve, when there is zero training error, the testing error may be arbitrarily bad, and the bound that we have used in Chapter 4, such as Rademacher averages for \mathcal{H} controlled by the ℓ_2 -norm of some parameters (with a bound D), grows as D/\sqrt{n} , which can typically be quite large. These bounds were true for *all* empirical risk minimizers. In this section we will focus on a particular one, namely **the one obtained by unconstrained gradient descent**.

When the model is over-parameterized (in other words, the capacity gets very large), that is, when the number of parameters is large or the norm constraint allows for exact fitting, a new phenomenon occurs, where after the test error explodes as the capacity grows, it goes down again (figure also taken from [20]):



The goal of this section is to understand why. But before this let’s present some empirical evidence, from toy examples and research papers.



There may be no double descent phenomenon if other empirical risk minimizers are used (instead of the one obtained by (stochastic) gradient descent).

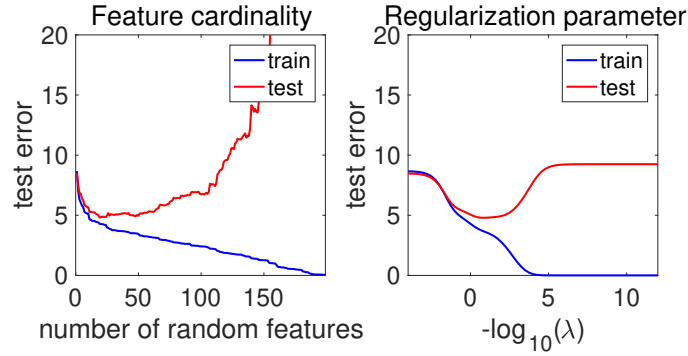
10.2.2 Empirical evidence

Toy example with random feature. We consider a random feature models like in Chapter 7 and Chapter 9, with the features $(v^\top x)_+$, for neurons v sampled uniformly on the unit spheres. We consider $n = 200$, $d = 5$ with input data distributed uniformly on the unit sphere, and we consider $y = \left(\frac{1}{4} + (v_*^\top x)^2\right)^{-1} + \mathcal{N}(0, \sigma^2)$, $\sigma = 2$, for some random v_* .

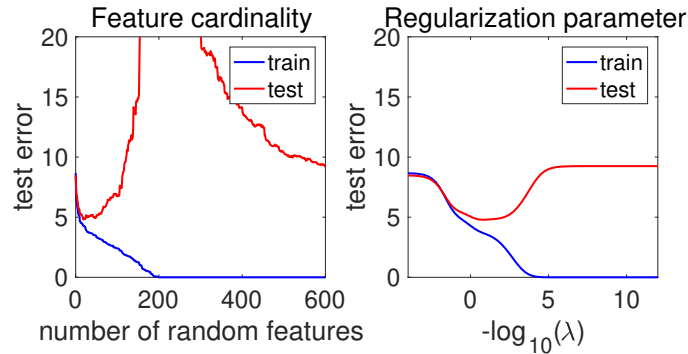
We sample m random features v_1, \dots, v_m uniformly on the sphere, and we learn parameters $\theta \in \mathbb{R}^m$ by minimizing

$$\frac{1}{n} \sum_{i=1}^n \left(y_i - \sum_{j=1}^m \theta_j (v_j^\top x_i)_+ \right)^2 + \lambda \|\theta\|_2^2. \quad (10.2)$$

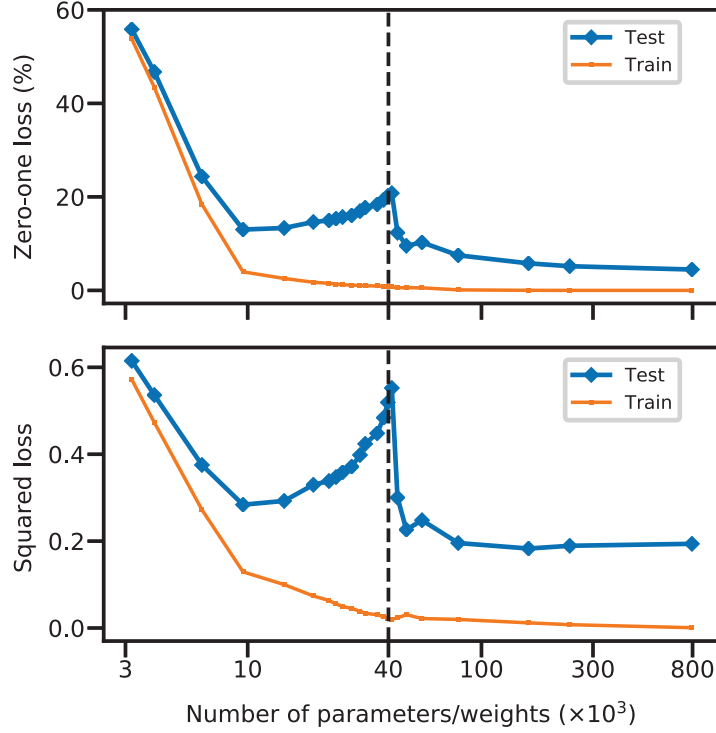
Below we report test errors after learning with gradient descent until convergence: (Left) varying m with $\lambda = 0$, (Right) varying λ with $m = +\infty$.



In the left curve above, the number of random features m is left less than n , as the test error diverges. But, when this number m is allowed to grow past n , we see the double descent phenomenon below (the right curve does not move). Similar experiments are shown in [20, 72].



Neural networks. We consider here a single hidden-layer fully connected network, on the MNIST dataset of handwritten digits, trained by stochastic gradient descent. As shown below (figure taken from [20]), we see a similar spike in errors around $n = 40000$ which is the number of training data points.



No phenomem when using regularization. When an extra regularizer is used, that is $\lambda \neq 0$ in Eq. (10.2), then the double descent phenomenon is reduced (see [72], in particular the right one). In particular, if the regularization parameter λ is adapted for each m , then the phenomem totally disappears (See [72] for more details).

10.2.3 Simplest analysis

We consider a Gaussian random variable with mean 0 and covariance matrix identity, with n observations x_1, \dots, x_n , and responses $y_i = x_i^\top \theta_* + \varepsilon_i$, with ε_i normal with mean zero and variance $\sigma^2 I$. We will compute an exact expectation of the risk of the minimum norm empirical risk minimizer (as detailed in Section 10.1.1), 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.

The excess risk is $R(\hat{\theta}) = (\hat{\theta} - \theta_*)^\top \Sigma (\hat{\theta} - \theta_*) = \|\hat{\theta} - \theta_*\|_2^2$.

Underparameterized regime. In the underparameterized regime, then the minimum norm empirical risk minimizer is simply the ordinary least-squares estimator, which is unbiased, that is $\mathbb{E}[\hat{\theta}] = \theta_*$, and we have an expected excess risk equal to (see the random design analysis from Chapter 3):

$$\mathbb{E}[R(\hat{\theta})] = \frac{\sigma^2}{n} \mathbb{E}[\text{tr}(\Sigma \hat{\Sigma}^{-1})].$$

As seen in Chapter 3, the expected risk is equal to

$$\sigma^2 \mathbb{E}[\text{tr}((X^\top X)^{-1})],$$

where $X \in \mathbb{R}^{n \times d}$ is the associated design matrix. The matrix $X^\top X \in \mathbb{R}^{d \times d}$ has a Wishart distribution with n degrees of freedom. It is almost surely invertible if $n \geq d$, and is such that $\mathbb{E}[\text{tr}((X^\top X)^{-1})] = \frac{d}{n-d-1}$ if $n \geq d+2$. The expectation is infinite for $n = d$ and $n = d+1$.

Therefore, we have for $n \geq d+2$:

$$\mathbb{E}[R(\hat{\theta})] = \sigma^2 \frac{d}{n-d-1}.$$

Overparameterized regime. In the overparameterized regime, when $n \leq d$, then the kernel matrix is almost surely invertible, and the minimum ℓ_2 -norm interpolator $\hat{\theta}$ is equal to (using the formulas above) $\hat{\theta} = X^\top (XX^\top)^{-1} y = X^\top (XX^\top)^{-1} X \theta_* + X^\top (XX^\top)^{-1} \varepsilon$. The expected excess risk decomposes into a bias and a variance term.

The *variance* term is equal to, since $\Sigma = I$,

$$\mathbb{E}[\varepsilon^\top (XX^\top)^{-1} X \Sigma X^\top (XX^\top)^{-1} \varepsilon] = \sigma^2 \mathbb{E}[\text{tr}((XX^\top)^{-1} X X^\top (XX^\top)^{-1})] = \sigma^2 \mathbb{E}[\text{tr}((XX^\top)^{-1})],$$

which is now a Wishart related expectation with the order of n and d reversed, that is, $\sigma^2 \frac{n}{d-n-1}$ for $d \geq n+2$.

The *bias term* is equal to

$$\mathbb{E}[\|\Sigma^{1/2} (X^\top (XX^\top)^{-1} X \theta_* - \theta_*)\|_2^2].$$

Since $\Sigma = I$, then we get a bias term equal to

$$\mathbb{E}[\theta_*^\top (I - X^\top (XX^\top)^{-1} X) \theta_*].$$

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 . By rotational invariance of the Gaussian distribution, this random subspace is uniformly distributed among all subspaces, and therefore, by rotational invariance, we can replace θ_* by $\|\theta_*\|_2 e_j$, that is,

$$\mathbb{E}[\theta_*^\top X^\top (XX^\top)^{-1} X \theta_*] = \|\theta_*\|_2^2 \cdot \mathbb{E}[e_j^\top X^\top (XX^\top)^{-1} e_j]$$

for any of the d canonical basis vectors e_j , $j = 1, \dots, d$, and thus

$$\mathbb{E}[\theta_*^\top X^\top (XX^\top)^{-1} X \theta_*] = \frac{\|\theta_*\|_2^2}{d} \sum_{j=1}^d \mathbb{E}[e_j^\top X^\top (XX^\top)^{-1} X e_j] = \frac{\|\theta_*\|_2^2}{d} \mathbb{E}[\text{tr}[X^\top (XX^\top)^{-1} X]] = \frac{\|\theta_*\|_2^2 n}{d}.$$

Thus the bias term is equal to $\frac{d-n}{d} \|\theta_*\|_2^2$.

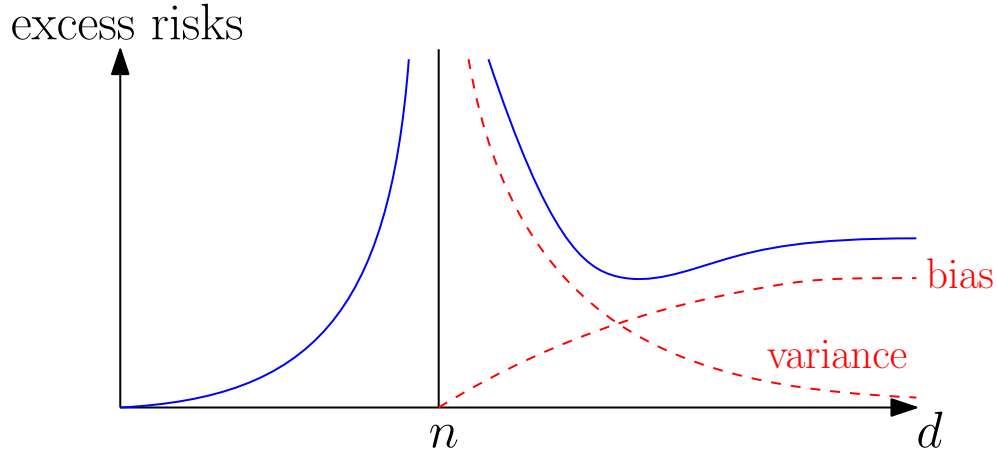
Therefore the overall expected risk is

$$\frac{\sigma^2 n}{d-n-1} + \|\theta_*\|_2^2 \frac{d-n}{d}.$$

Summary. We get

$$\begin{aligned} \text{if } d \leq n - 2, \quad \mathbb{E}[R(\hat{\theta})] &= \sigma^2 \frac{d}{n - d - 1} \\ \text{if } d \geq n + 2, \quad \mathbb{E}[R(\hat{\theta})] &= \frac{\sigma^2 n}{d - n - 1} + \|\theta_*\|_2^2 \frac{d - n}{d}. \end{aligned}$$

This leads to the following picture.



This extends to more general sampling models, see [54], and to random non-linear features [72].

10.3 Global convergence of gradient descent for two-layer neural networks

In Section 9.5, arguments were presented, highlighting that gradient descent neural networks with a single hidden layer and infinite widths could be shown to converge to a global minimum. This was based on [35], and taken from <https://francisbach.com/gradient-descent-neural-networks-global-convergence/>.

When applied to logistic regression, then combining these results with Section 10.1, we also obtain that in the infinite width limit, we obtain a predictor that interpolates the data, with a minimum norm, for norms which are exactly the ones obtained in Section 9.3[36]: See <https://francisbach.com/gradient-descent/> for more details.

Chapter 11

Lower bounds on performance

Chapter summary

-Statistical lower bounds: for least-squares regression, the optimal performance of supervised learning with target functions which are linear in some feature vector, or in Sobolev spaces on \mathbb{R}^d , happens to be achieved by several algorithms presented earlier in the book. The lower bounds can be obtained through information theory or Bayesian analysis.

-Optimization lower bounds: for the classical problem classes from Chapter 5, hard functions can be designed so that gradient-descent based algorithms that linearly combine gradients are shown to be optimal.

-Lower bounds for stochastic gradient descent: The rates proportional to $O(1/\sqrt{n})$ for convex functions and $O(1/n\mu)$ for 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 k goes to infinity. Most of them were non-asymptotic upper-bounds on the error measures, with a precise dependence on the problem parameters (e.g., smoothness of the target function or the objective function).

In this chapter, we are looking at lower-bounds on performance, that is, we aim to show that for a certain problem class and a certain class of algorithms, the error measures cannot go to zero too quickly. Lower bounds are useful, in particular when they match upper-bounds up to constants (we can then claim that we have an “optimal” method). They sometimes provide hard problems (like for optimization), sometimes not (when they are based on information theory such as for prediction performance).



Lower bounds will be obtained in a “minimax” setting where we look at the worst-case performance over the entire problem class. As for upper-bounds, looking at worst-case performance is by essence pessimistic, and algorithms often behaved 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 contains interesting problems.

11.1 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 $f^* : \mathcal{X} \rightarrow \mathbb{R}$ (here the conditional expectation of y given x) is in a particular set, such as:

- linear function of some d -dimensional features, that is, $f_*(x) = \langle \theta_*, \varphi(x) \rangle$, for $\theta_* \in \mathbb{R}^d$, potentially in a ℓ_2 -ball, and/or with less than k non-zero elements,
- functions with all partial derivatives up to order s bounded in L_2 -norm (e.g., Sobolev spaces).

Since we are looking for lower-bounds, we are free to make extra assumptions (that can only make the problem simpler) and lower the lower-bounds. For example, we will focus on Gaussian noise with constant variance σ^2 which is independent from x .

We can either consider fixed design assumptions or random designs with the simplest input distributions (that can only make the problem simpler)/

Classification. Lower bounds for classification problems are more delicate and out of scope (see, e.g., [107]). We can however get lower-bounds for the convex surrogates which are typically used (but note that this does not translate to lower-bounds for the 0-1 loss), see for example Section 11.3 for Lipschitz-continuous loss functions.

11.1.1 Minimax lower bounds

We consider a set of probability distributions indexed by some set Θ (that can be characterizing input distributions, smoothness of the target function). We consider some data \mathcal{D} , generated from this distribution, and we denote \mathbb{E}_θ expectations with respect to data coming from the distribution indexed by θ .

We consider an estimator $\mathcal{A}(\mathcal{D})$ of $\theta \in \Theta$, with some squared distance d^2 between two elements of Θ , so that $d(\theta, \theta')^2$ measures the performance of θ' when the true estimator is θ . The performance of \mathcal{A} when the data come from θ_* is

$$\mathbb{E}_{\theta_*} [d(\theta_*, \mathcal{A}(\mathcal{D}))^2].$$

The goal is to find an algorithm so that $\sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*} [d(\theta_*, \mathcal{A}(\mathcal{D}))^2]$ is as small as possible, and the lower bound of performance is thus:

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*} [d(\theta_*, \mathcal{A}(\mathcal{D}))^2]. \quad (11.1)$$

This is often referred to as “minimax” lower bounds.

Since by Markov’s inequality, $\mathbb{E}_{\theta_*} [d(\theta_*, \mathcal{A}(\mathcal{D}))^2] \geq A \mathbb{P}_{\theta_*} (d(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A)$, it is sufficient to lower bound

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{P}_{\theta_*} (d(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A),$$

for some $A > 0$. This will be useful for techniques based on information theory.

We will see two principles for obtaining statistical minimax lower-bounds:

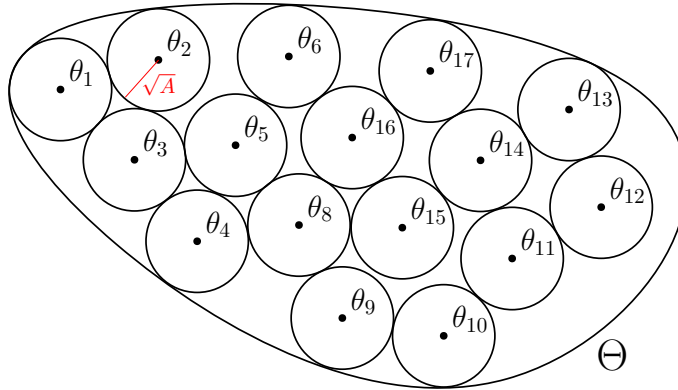
- **Reduction to an hypothesis test:** by selecting a finite subset $\{\theta_1, \dots, \theta_M\}$ of distributions Θ which is maximally spread, a good estimator leads to a good hypothesis test that can identify which θ_j was used to generate the data. We can then use information theory to lower-bound the probability of error of such a test. This is a very versatile technique that can deal with most situations, 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 we used to compute the Bayes risk in Chapter 4, e.g., for Hilbertian or Euclidean performance measures, the optimal estimator is the conditional expectation $\mathbb{E}[\theta_*|\mathcal{D}]$. The key is then to choose distributions so that it can be computed in closed form. This approach is less flexible, but the simplest in situations where it can be applied (fixed design regression on balls, with potentially sparse assumptions).

11.1.2 Reduction to an hypothesis test

The principle is simple: pack the set Θ with “balls” of some radius $4A$, that is find $\theta_1, \dots, \theta_M \in \Theta$ such that

$$\forall i \neq j, d(\theta_i, \theta_j)^2 \geq 4A, \quad (11.2)$$

and transform the estimation problem into a hypothesis test, that is, an algorithm going from the data \mathcal{D} to one out of M potential outcomes.



Then, because we take the supremum over a smaller set:

$$\sup_{\theta_* \in \Theta} \mathbb{P}_{\theta_*}(d(\theta_*, \mathcal{A}(\mathcal{D}))^2 > A) \geq \max_{j \in \{1, \dots, M\}} \mathbb{P}_{\theta_j}(d(\theta_j, \mathcal{A}(\mathcal{D}))^2 > A).$$

Any algorithm $\mathcal{A}(\mathcal{D}) \in \Theta$ gives a test

$$g(\mathcal{A}(\mathcal{D})) = \arg \min_{j \in \{1, \dots, m\}} d(\theta_j, \mathcal{A}(\mathcal{D})) \in \{1, \dots, m\},$$

where ties are broken arbitrarily (e.g., by selecting the minimal index). Because of the packing condition in Eq. (11.2), the performance of \mathcal{A} can be lower-bounded by the classification performance of $g \circ \mathcal{A}$.

Indeed, if, for some $j \in \{1, \dots, M\}$, $g(\mathcal{A}(\mathcal{D})) \neq j$, there exists $k \neq j$, such that $d(\theta_k, \mathcal{A}(\mathcal{D})) < d(\theta_j, \mathcal{A}(\mathcal{D}))$. Moreover, using the triangle inequality for d , we get:

$$d(\theta_j, \theta_k)^2 \leq 2[d(\theta_j, \mathcal{A}(\mathcal{D}))^2 + d(\mathcal{A}(\mathcal{D}), \theta_k)^2],$$

then,

$$\begin{aligned} d(\theta_j, \mathcal{A}(\mathcal{D}))^2 &\geq \frac{1}{2}d(\theta_j, \theta_k)^2 - d(\mathcal{A}(\mathcal{D}), \theta_k)^2 \\ &\geq \frac{1}{2}d(\theta_j, \theta_k)^2 - d(\mathcal{A}(\mathcal{D}), \theta_j)^2 \text{ using the optimal } k, \end{aligned}$$

which implies $d(\theta_j, \mathcal{A}(\mathcal{D}))^2 \geq \frac{1}{4}d(\theta_j, \theta_k)^2 \geq A$. Thus, we have

$$\mathbb{P}_{\theta_j}(d(\theta_j, \mathcal{A}(\mathcal{D}))^2 > A) \geq \mathbb{P}_{\theta_j}(g(\mathcal{A}(\mathcal{D})) \neq j),$$

leading to

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*}[d(\theta_*, \mathcal{A}(\mathcal{D}))^2] \geq A \cdot \inf_g \max_{j \in \{1, \dots, M\}} \mathbb{P}_{\theta_j}(g(\mathcal{D}) \neq j) \geq A \cdot \inf_g \frac{1}{M} \sum_{j=1}^M \mathbb{P}_{\theta_j}(g(\mathcal{D}) \neq j), \quad (11.3)$$

where g is any function from \mathcal{D} to $\{1, \dots, M\}$. We have lower-bounded the minimax statistical performance by the minimax performance of an hypothesis test $g : \mathcal{D} \rightarrow \{1, \dots, M\}$. Information theory can be then used to lower-bound this minimax error. We first provide a quick review of information theory (see [39] for more details).

11.1.3 Information theory

Entropy. Given a random variable y taking finitely many values in \mathcal{Y} , its entropy is equal to

$$H(y) = - \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \log \mathbb{P}(y = y').$$

Since $\mathbb{P}(y = y') \in [0, 1]$, the entropy is always non-negative. Moreover, using Jensen's inequality for the logarithm, we have $H(y) = \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \log \frac{1}{\mathbb{P}(y=y')} \leq \log \left(\sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') \frac{1}{\mathbb{P}(y=y')} \right) = \log |\mathcal{Y}|$.

The entropy $H(y)$ represents the uncertainty associated with the random variable y , going from $H(y) = 0$ if y is deterministic (that is $\mathbb{P}(y = y') = 1$ for some $y' \in \mathcal{Y}$), to $\log |\mathcal{Y}|$ when y has a uniform distribution.

Joint and conditional entropies. Given two random variables x, y with finitely many values in \mathcal{X} and \mathcal{Y} , we can define the joint entropy

$$H(x, y) = - \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x', y = y') \log \mathbb{P}(x = x', y = y').$$

It can be decomposed as

$$\begin{aligned}
 H(x, y) &= - \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log [\mathbb{P}(y = y' | x = x') \mathbb{P}(x = x')] \\
 &= - \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \mathbb{P}(y = y' | x = x') - \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y', x = x') \log \mathbb{P}(x = x') \\
 &= \sum_{x' \in \mathcal{X}} \mathbb{P}(x = x') \log H(y | x = x') + H(x),
 \end{aligned}$$

where $H(y | x = x')$ is the entropy of the conditional distribution of y given $x = x'$. By defining the conditional entropy $H(y | x)$ as $H(y | x) = \sum_{x' \in \mathcal{X}} \mathbb{P}(x = x') H(y | x = x')$, we exactly have:

$$H(x, y) = H(y | x) + H(x).$$

This leads to a first version of Fano's inequality, that lower bounds the probability that $y \neq \hat{y}$ from the conditional entropy $H(y | \hat{y})$, the main idea is that if y remains very uncertain given \hat{y} , then the probability that they are equal cannot be too large.

Proposition 11.1 (Fano's inequality) *If the random variable y and \hat{y} have values in the same finite set \mathcal{Y} , then*

$$\mathbb{P}(\hat{y} \neq y) \geq \frac{H(y | \hat{y}) - \log 2}{\log |\mathcal{Y}|}.$$

Proof Let $e = 1_{y \neq \hat{y}} \in \{0, 1\}$ be the indicator function of errors, then, by decomposing the joint entropy through conditional and marginal entropies in the two different ways, we get:

$$H(e | \hat{y}) + H(y | e, \hat{y}) = H(e, y | \hat{y}) = H(y | \hat{y}) + H(e | y, \hat{y}).$$

We then have $H(e | y, \hat{y}) = 0$ (because e is deterministic given y and \hat{y}), $H(e | \hat{y}) \leq H(e) \leq \log 2$ (because $e \in \{0, 1\}$), and $H(y | e, \hat{y}) = \mathbb{P}(e = 1) H(y | \hat{y}, e = 1) + \mathbb{P}(e = 0) H(y | \hat{y}, e = 0) = \mathbb{P}(e = 1) H(y | \hat{y}, e = 1) + 0 \leq \mathbb{P}(\hat{y} \neq y) \log |\mathcal{Y}|$. Expressing $\mathbb{P}(\hat{y} \neq y)$ in function of other quantities leads to the desired result. ■

Data processing inequality. A fundamental result in information theory allows to lower bound conditional entropies where conditional independencies are present. That is, if we have three random variables x, y, z , such that z and x are conditionally independent given y , then $H(x | z) \geq H(x | y)$: in words, the uncertainty of x given z has to be larger than the uncertainty of $x | y$, which is “normal” because the statistical dependence between x and z is entirely through y .

The data processing inequality is simple application of the concavity of the entropy as a function of the probability mass function; indeed, we have, using that by conditional independence $\mathbb{P}(x = x' | z =$

$$z') = \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x' | y = y') \mathbb{P}(y = y' | z = z'):$$

$$\begin{aligned} H(x|z) &= \sum_{z' \in \mathcal{Z}} \mathbb{P}(z = z') H(x|z = z') \\ &\geq \sum_{z' \in \mathcal{X}} \mathbb{P}(z = z') \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y' | z = z') H(x|y = y') \\ &= \sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y') H(x|y = y') = H(x|y). \end{aligned}$$

This leads immediately to the following full version of Fano's inequality:

Proposition 11.2 (Fano's inequality) *If the random variable y and \hat{y} have values in the same finite set \mathcal{Y} , and if we have a Markov chain $y \rightarrow z \rightarrow \hat{y}$, then*

$$\mathbb{P}(\hat{y} \neq y) \geq \frac{H(y|\hat{y}) - \log 2}{\log |\mathcal{Y}|} \geq \frac{H(y|z) - \log 2}{\log |\mathcal{Y}|}.$$

We need a last concept from information theory, namely mutual information and Kullback-Leibler divergence, both discrete-valued random variables, and for continuous-valued random variables.

Mutual information. Given two random variables x and y , then we can define their mutual information as

$$I(x, y) = H(x) - H(x|y) = H(x) + H(y) - H(x, y) = H(y) - H(y|x).$$

This can be seen as the reduction of uncertainty in x when observing y . It is symmetric, always less than $\log |\mathcal{X}|$ and $\log |\mathcal{Y}|$. Moreover, it can be written as:

$$\begin{aligned} I(x, y) &= H(x) + H(y) - H(x, y) \\ &= \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}} \mathbb{P}(x = x', y = y') \log \frac{\mathbb{P}(x = x', y = y')}{\mathbb{P}(x = x') \mathbb{P}(y = y')}, \end{aligned}$$

which can be seen as the Kullback-Leibler (KL) divergence between the distribution of (x, y) and the product of marginals of x and y . Indeed, given two distribution on \mathcal{Z} , p and q (which are non-negative functions on \mathcal{Z} that sum to one), then

$$D_{\text{KL}}(p||q) = \sum_{x \in \mathcal{Z}} p(x) \log \frac{p(x)}{q(x)}.$$

The KL divergence is always non-negative by convexity of the function $t \mapsto t \log t$, and equal to zero, if and only if $p = q$. Moreover, the KL divergence is jointly convex in (p, q) . Thus, one can see the mutual information between the KL divergences between the joint distribution of (x, y) and the corresponding product of marginals (which is thus non-negative).

From discrete to continuous distributions. Many of the information theory concepts can be extended to continuous random variables on \mathbb{R}^d , by replacing the probability mass function by the probability density with respect to some base measures. Then many properties (which were obtained through convex arguments) extend. In particular, the data processing inequality and Fano's inequality when z is continuous-valued.

Moreover, the KL divergence between two distributions can be defined as

$$D_{\text{KL}}(dp||qq) = \mathbb{E}_{dp(x)} \log \frac{dp(x)}{dq(x)}.$$

A short calculation shows that for two normal distributions of means μ_1, μ_2 and equal covariance matrices Σ , the KL divergence is equal to

$$\frac{1}{2}(\mu_1 - \mu_2)^\top \Sigma^{-1}(\mu_1 - \mu_2).$$

11.1.4 Lower-bound on hypothesis testing based on information theory

We consider a joint random variable (y, \mathcal{D}) distributed as y uniform in $\{1, \dots, M\}$, and, given $y = j$, \mathcal{D} distributed as the distribution associated with θ_j . We consider $\hat{y} = g(\mathcal{D})$. This defines a Markov chain: $y \rightarrow \mathcal{D} \rightarrow g(\mathcal{D})$, that is, even for a randomized test, $g(\mathcal{D})$ is independent of y given \mathcal{D} . The last term in Eq. (11.3) is exactly the probability that $\hat{y} \neq y$. This is exactly what Fano's inequality from information theory gives us, leading to the following corollary.

Corollary 11.1 (Fano's inequality for multiple hypothesis testing) *Given M probability distributions dp_j on \mathcal{D} , then*

$$\inf_g \frac{1}{M} \sum_{j=1}^M \mathbb{P}_j(g(\mathcal{D}) \neq j) \geq 1 - \frac{1}{M^2 \log M} \sum_{j,j'=1}^M D_{\text{KL}}(dp_j||dp_{j'}) - \frac{\log 2}{\log M}.$$

Proof We consider a joint random variable (y, \mathcal{D}) distributed as y uniform in $\{1, \dots, M\}$, and, given $y = j$, \mathcal{D} distributed as the distribution dp_j . Starting from Prop. 11.2, we get:

$$\begin{aligned} H(y|z) &= H(y) - I(y, z) = \log M - \frac{1}{M} \sum_{j=1}^M D_{\text{KL}}(dp_j||\frac{1}{M} \sum_{j'=1}^M dp_{j'}) \\ &\geq H(y) - I(y, z) = \log M - \frac{1}{M^2} \sum_{j,j'=1}^M D_{\text{KL}}(dp_j||dp_{j'}), \end{aligned}$$

by convexity of the Kullback-Leibler divergence. ■

Using Gaussian noise to compute KL divergences. For regression with Gaussian errors such as $y_i = f_\theta(x_i) + \varepsilon_i$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, then, for fixed designs (all x_i 's deterministic), we exactly get

$$D_{\text{KL}}(dp_{\theta_j}||dp_{\theta_{j'}}) = \frac{1}{2\sigma^2} \sum_{i=1}^n [f_{\theta_j}(x_i) - f_{\theta_{j'}}(x_i)]^2 = \frac{n}{2\sigma^2} d(\theta_j, \theta_{j'})^2,$$

where $d(\theta, \theta')^2 = \frac{1}{n} \sum_{i=1}^n [f_\theta(x_i) - f_{\theta'}(x_i)]^2$.

For random designs, we consider distributions on $(x_i, y_i)_{i=1, \dots, n}$. If we consider a single distribution for x , then

$$D_{\text{KL}}(dp_{\theta_j} || dp_{\theta_{j'}}) = \frac{1}{2\sigma^2} \int_{\mathcal{X}} [f_{\theta_j}(x) - f_{\theta_{j'}}(x)]^2 dp(x) = \frac{1}{\sigma^2} \|f_{\theta_j} - f_{\theta_{j'}}\|_{L_2(dp(x))}^2 = \frac{1}{2\sigma^2} d(\theta_j, \theta_{j'})^2.$$

In order to obtain a lower bound with Gaussian noise, we need to find $\theta_1, \dots, \theta_M$ in Θ such that:

- $\frac{1}{M^2} \sum_{j, j'=1}^M \frac{n}{2\sigma^2} d(\theta_j, \theta_{j'})^2 \leq \log(M)/4$.
- $\log 2 / \log M \leq 1/4$ (that is $M \geq 16$)
- $\min_{j \neq k} d(\theta_j, \theta_k)^2 \geq 4A$.

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) using the Varshamov-Gilbert's lemma. We present them before going over examples.

Volume argument. The following lemma provides the simplest argument.

Lemma 11.1 (Packing ℓ_2 -balls) *Let M be the maximal number of elements of the Euclidean ball of radius 1, which are at least 2ε -apart in ℓ_2 -norm. Then $(2\varepsilon)^{-d} \leq M \leq (1 + \varepsilon^{-1})^d$.*

Proof Let $\theta_1, \dots, \theta_M$ be the corresponding M points.

(a) All balls of center θ_j and radius ε are disjoint and included in the ball of radius $1 + \varepsilon$. Thus, the sum of volumes of the small balls is smaller than the volume of the large balls, that is, $M\varepsilon^d \leq (1 + \varepsilon)^d$.

(b) Since M is maximal, for any θ such that $\|\theta\|_2 \leq 1$, there exists a $j \in \{1, \dots, M\}$ such that $\|\theta_j - \theta\|_2 \leq 2\varepsilon$ (otherwise, we can add a new point to $\{\theta_1, \dots, \theta_M\}$ and M is not maximal). Thus the ball of radius 1 is covered by the M balls of radius θ_j and radius 2ε . Thus, by using volumes, we get $1 \leq M(2\varepsilon)^d$. ■

Packing with Varshamov-Gilbert lemma. The maximal number of points in the hypercube $\{0, 1\}^d$ that are at least $d/4$ -apart in Hamming loss (i.e., ℓ_1 -distance) is greater than $\exp(d/8)$.

Lemma 11.2 (Varshamov-Gilbert's lemma) *For any $\alpha \in (0, 1)$, there exists a subset A of the hypercube $\{0, 1\}^d$ such that*

- (a) *for all $x, x' \in A$ such that $x \neq x'$, $\|x - x'\|_1 \geq (1 - \alpha)\frac{d}{2}$,*
- (b) $|A| \geq \exp(d\alpha^2/2)$.

Proof We consider the largest family satisfying (a). By maximality, the union of ℓ_1 -ball of radius $(1 - \alpha)\frac{d}{2}$ includes all of $\{0, 1\}^d$. Therefore,

$$2^d \leq \sum_{x \in A} |\{y \in \{0, 1\}^d, \|y - x\|_1 \leq (1 - \alpha)\frac{d}{2}\}|.$$

Consider a random variable z which is binomial with parameter d and $1/2$. Then,

$$2^{-d} |\{y \in \{0, 1\}^d, \|y - x\|_2^2 = \|y - x\|_1 \leq (1 - \alpha)\frac{d}{2}\}| = \mathbb{P}(z \leq (1 - \alpha)\frac{d}{2}) = \mathbb{P}(z \geq (1 + \alpha)\frac{d}{2}).$$

Using Hoeffding's inequality, we get $\mathbb{P}(z \geq (1 + \alpha)\frac{d}{2}) = \mathbb{P}(z - \mathbb{E}[z] \geq \alpha\frac{d}{2}) \leq \exp(-2d(\alpha/2)^2) = \exp(-d\alpha^2/2)$. This leads to the result. \blacksquare

11.1.5 Examples

Fixed design linear regression. We consider linear regression with $\Phi \in \mathbb{R}^{n \times d}$ a design matrix with $\frac{1}{n}\Phi^\top \Phi = I$ (which imposes $n \geq d$). We consider the ball $\Theta = \{\theta \in \mathbb{R}^d, \|\theta\|_2 \leq D\}$. By rotational invariance of the Gaussian distribution of the noise variable ε , we can assume that the first d rows are equal to $\sqrt{n}I$ and the rest of the rows are equal to zero, and thus we can assume the model $y = \theta_* + \frac{1}{\sqrt{n}}\varepsilon$, where $\varepsilon \in \mathbb{R}^d$ with normal distribution with mean zero and covariance $\sigma^2 I$, and $y \in \mathbb{R}^d$. We are thus in the situation where $d(\theta, \theta')^2 = \|\theta - \theta'\|_2^2$.

In order to find M points in $\Theta = \{\theta \in \mathbb{R}^d, \|\theta\|_2 \leq D\}$, we consider the $M \geq \exp(d/8)$ elements x_1, \dots, x_M of $\{0, 1\}^d$ from Lemma 11.2, and define $\theta_i = \beta(2x_i - 1_d)$. Thus $\|\theta_i\|_2^2 = \beta^2 d$, and, for $i \neq j$,

$$\|\theta_i - \theta_j\|_2^2 \leq 4\beta^2 d \leq 32\beta^2 \log(M) \text{ and } \|\theta_i - \theta_j\|_2^2 \geq \beta^2 d.$$

We thus need, $\beta^2 d \leq D^2$, and $32\beta^2 \log(M) \frac{n}{2\sigma^2} \leq \frac{\log M}{4}$, that is, $64\beta^2 \frac{n}{\sigma^2} \leq 1$. Thus, the optimal rate is greater than

$$\frac{1}{8}\beta^2 d \geq \frac{1}{8} \min\{D^2, \frac{\sigma^2 d}{64n}\}.$$

Therefore, when $D^2 \geq \frac{\sigma^2 d}{64n}$, we get a lower bound of $\frac{\sigma^2 d}{512n}$, which is the upper-bound obtained in Chapter 3 (note that in Section 3.7 we provided a sharper lower-bound using similar tools as Section 11.1.6).

The sparse regression setting could be considered as well with the same tool, but the proof is simpler with the Bayesian arguments from Section 11.1.6. We now turn to the random design setting.

Exercise 11.1 Use Lemma 11.1 instead of Lemma 11.2 to obtain the same result.

Random design linear regression. We consider the same model as above, but with (x_i, y_i) sampled i.i.d. from a given distribution such that $\mathbb{E}[\varphi(x)\varphi(x)^\top] = I$, so that $d(\theta, \theta')^2 = \|\theta - \theta'\|_2^2$. Thus the result above for fixed design regression also applies to the random design setting.

Non parametric estimation with Hilbert spaces. We consider random design regression with a fixed distribution for the inputs, with Gaussian independent noise and target functions which are in certain ellipsoid of $L_2(dp(x))$. That is, we assume that there exists a compact self adjoint operator T on $L_2(dp(x))$ such that $\langle \theta, T^{-1}\theta \rangle_{L_2(dp(x))} \leq D^2$. We denote by $(\lambda_m)_{m \geq 1}$ the non-increasing sequence of eigenvalues of T , with the associated eigenvectors ψ_m in $L_2(dp(x))$.

We consider a certain integer K , then consider $M \geq \exp(K/8)$ elements x_1, \dots, x_M of $\{0, 1\}^K$. We then define $\theta_i = \beta \sum_{m=1}^K (2(x_i)_m - 1) \psi_m$. Then $\langle \theta, T^{-1} \theta \rangle_{L_2(dp(x))} = \beta^2 \sum_{m=1}^K \lambda_m^{-1} \leq K \beta^2 \lambda_K^{-1}$, and, for $i \neq j$,

$$\|\theta_i - \theta_j\|_{L_2(dp(x))}^2 \leq 4\beta^2 K \leq 32\beta^2 \log(M) \text{ and } \|\theta_i - \theta_j\|_{L_2(dp(x))}^2 \geq \beta^2 K.$$

We thus need, $\beta^2 K \leq D^2 \lambda_K$, and $32\beta^2 \log(M) \frac{n}{2\sigma^2} \leq \frac{\log M}{4}$, that is, $64\beta^2 \frac{n}{\sigma^2} \leq 1$. Thus, the minimax lower bound is greater than

$$\frac{1}{8} \beta^2 K \geq \frac{1}{8} \min\{D^2 \lambda_K, \frac{\sigma^2 K}{64n}\}.$$

We can now specialized to Sobolev spaces where it can be shown that for compact supports with piecewise smooth boundaries, then the sum of all L_2 -norms of partial derivatives correspond to an operator for which $\lambda_K \geq K^{-\alpha}$, with $\alpha = 2s/d$. The lower bound becomes

$$\max_{K \geq 1} \frac{1}{8} \min\{D^2 K^{-\alpha}, \frac{\sigma^2 K}{64n}\},$$

which can be balanced to obtain $K \propto \left(\frac{nD^2}{\sigma^2}\right)^{1/(1+\alpha)}$, leading to lower bound proportional to

$$D^{2/(1+\alpha)} \left(\frac{\sigma^2}{n}\right)^{\alpha/(1+\alpha)}.$$

For $\alpha = 2s/d$, we get $\alpha/(1+\alpha) = \frac{2s}{2s+d}$, and the lower matches the upper-bound obtained with kernel ridge regression in Chapter 7. It turns out that the lower bound on the minimax rate for Lipschitz-continuous function is the same as for $s = 1$ [102, Section 2.6].

11.1.6 Minimax lower bounds through Bayesian analysis

As outlined for least-square in Section 3.7, we can use a Bayesian analysis as follows. We consider a certain probability distribution $dp(\theta_*)$ whose support is included in Θ . Then we have:

$$\inf_{\mathcal{A}} \sup_{\theta_* \in \Theta} \mathbb{E}_{\theta_*} [d(\theta_*, \mathcal{A}(\mathcal{D}))^2] \geq \inf_{\mathcal{A}} \mathbb{E}_{dp(\theta_*)} \mathbb{E}_{\theta_*} [d(\theta_*, \mathcal{A}(\mathcal{D}))^2].$$

This reasoning is particularly simple when the optimal algorithm \mathcal{A} is simple to estimate, which is the case in particular where d is an Euclidean norm, so that $\mathcal{A}^*(\mathcal{D}) = \mathbb{E}[\theta_* | \mathcal{D}]$. If the prior $dp(\theta_*)$ and the likelihood $dp(\mathcal{D} | \theta_*)$ are simple enough, then the conditional expectation can be done 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 in [42].

Least-squares on an Euclidean ball. We consider linear regression with fixed design like in the previous section (with a bound $\|\theta_*\|_2 \leq D$), we corresponds to the model $y = \theta_* + \frac{1}{\sqrt{n}} \varepsilon$, where $\varepsilon \in \mathbb{R}^d$ with normal distribution with mean zero and covariance $\sigma^2 I$, and $y, \theta_* \in \mathbb{R}^d$.

We then consider a prior distribution on θ_* as $\theta_* = \beta x$, where $x \in \{-1, 1\}^d$ are independent Rademacher random variables. We need $\beta^2 d \leq D^2$ to be in the correct set. We then need to compute $\mathbb{E}[\theta_* | y]$. The posterior probability of θ_* is supported on $\beta\{-1, 1\}^n$. Moreover, given the independence

by component, we can treat each of them separately. Then, by keeping only terms that depends on the posterior value, we get:

$$\mathbb{P}((\theta_*)_i = \pm\beta|y_i) \propto \exp(-\frac{n}{2\sigma^2}(y_i - \pm\beta)^2) \propto \exp(\pm\frac{n}{\sigma^2}y_i\beta).$$

Thus,

$$\mathbb{E}[(\theta_*)_i|y_i] = \beta \frac{\exp(\frac{n}{\sigma^2}y_i\beta) - \exp(-\frac{n}{\sigma^2}y_i\beta)}{\exp(\frac{n}{\sigma^2}y_i\beta) + \exp(-\frac{n}{\sigma^2}y_i\beta)} = \beta \frac{1 - \exp(-2\frac{n}{\sigma^2}y_i\beta)}{1 + \exp(-2\frac{n}{\sigma^2}y_i\beta)} = \beta [2\text{sigmoid}(2\frac{n}{\sigma^2}y_i\beta) - 1],$$

where $\text{sigmoid}(\alpha) = 1/(1 + \exp(-\alpha))$.

The posterior variance for the i -th component is equal to

$$\begin{aligned} \mathbb{E}[(\theta_*)_i - \mathbb{E}[(\theta_*)_i|y_i]]^2 &= \frac{1}{2}\mathbb{E}_{\varepsilon_i}(\beta - \beta[2\text{sigmoid}(2\frac{n}{\sigma^2}\beta(\beta + \varepsilon_i/\sqrt{n})) - 1])^2 \\ &\quad + \frac{1}{2}\mathbb{E}_{\varepsilon_i}(-\beta - \beta[2\text{sigmoid}(2\frac{n}{\sigma^2}\beta(-\beta + \varepsilon_i/\sqrt{n})) - 1])^2 \\ &= 4\beta^2\mathbb{E}_{\varepsilon_i \sim \mathcal{N}(0, \sigma^2)}\left[\left(\text{sigmoid}\left(-2\frac{n}{\sigma^2}\beta^2 + 2\frac{\sqrt{n}}{\sigma^2}\beta\varepsilon_i\right)\right)^2\right] \\ &= 4\beta^2\mathbb{E}_{\tilde{\varepsilon}_i \sim \mathcal{N}(0, 1)}\left[\left(\text{sigmoid}\left(-2\frac{n}{\sigma^2}\beta^2 + 2\frac{\beta\sqrt{n}}{\sigma}\tilde{\varepsilon}_i\right)\right)^2\right] \end{aligned}$$

We consider the function $\psi : \alpha \mapsto \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, 1)}\left[\left(\text{sigmoid}(-2\alpha^2 + 2\alpha\varepsilon)\right)^2\right]$. We have $\psi(0) = 1/4$, and $\psi(\alpha) \rightarrow 0$ when $\alpha \rightarrow +\infty$, and we have $\psi(\alpha) \geq \frac{1}{4}\mathbb{P}_{\varepsilon \sim \mathcal{N}(0, 1)}(\varepsilon > \alpha) \geq \frac{1}{8}\exp(-\alpha^2)$, by using simple Gaussian tail bounds.

Thus, the posterior variance is greater than

$$\frac{\beta^2 d}{2} \exp(-n\beta^2/\sigma^2) = \frac{\sigma^2 d}{n} \times \frac{\beta^2 n}{2\sigma^2} \exp(-n\beta^2/\sigma^2),$$

which is maximized for $\beta^2 \propto \sigma^2/n$, and thus if $\sigma^2 d/n$ is smaller than D^2 , we obtain the usual $\sigma^2 d/n$, while if it is greater then D^2 , we take $\beta^2 = D^2/d$, to obtain the lower bound

$$D^2 \exp(-4nD^2/(\sigma^2 d)) \geq D^2 \exp(-4),$$

which leads to the same bound as the previous section, but with a more direct argument.

Sparse case. In order to deal with the sparse case, we could consider a prior on θ_* that is only selecting k non-zero elements out of d , and perform an analysis based on the posterior probability of θ_* . Following [42], it is easier to divide the set of d variables into k blocks of size d/k (for simplicity we assume that d/k is an integer). We then consider a prior probability defined independently on each of the k blocks by selecting one of the d/k variables uniformly at random and setting its value to β , while all others are set to zero.

In order 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 j -th one, which is proportional to

$$\exp(-n/(2\sigma^2)(y_j - \beta)^2) \prod_{i \neq j} \exp(-n/(2\sigma^2)(y_i)^2) \propto \exp(n\beta y_j/\sigma^2).$$

The conditional expectation of θ_* then satisfies

$$\mathbb{E}[(\theta_*)_i|y] = \beta \frac{\exp(n\beta y_i/\sigma^2)}{\sum_{j=1}^{d/k} \exp(n\beta y_j/\sigma^2)}.$$

In order to compute the posterior variance, we need to sample from the prior θ_* . By symmetry, we may consider that $\theta_1 = \beta$. If $y_1 \leq \max_{j \neq 1} y_j$, then

$$\mathbb{E}[(\theta_*)_1|y] = \beta \frac{\exp(n\beta y_1/\sigma^2)}{\sum_{j=1}^{d/k} \exp(n\beta y_j/\sigma^2)} \leq \beta \frac{\exp(n\beta y_1/\sigma^2)}{\exp(n\beta y_1/\sigma^2) + \exp(n\beta \max_{j \neq 1} y_j/\sigma^2)} \leq \beta/2,$$

and then the risk is at least $(\beta - \mathbb{E}[(\theta_*)_1|y])^2 \geq \beta^2/4$.

In order to lower-bound the probability that $y_1 \leq \max_{j \neq 1} y_j$ we can then consider the events $\{y_1 \leq \beta\}$ and $\{\beta \leq \max_{j \neq 1} y_j\}$. The probability that $y_1 = \beta + \varepsilon_1$ is less than β is greater than $1/2$. Moreover,

$$\mathbb{P}(\{\beta \leq \max_{j \neq 1} y_j\}) \geq 1 - (1 - \mathbb{P}_{t \sim \mathcal{N}(0,1)}(t \geq \beta\sqrt{n}/\sigma))^{d/k-1}.$$

Thus, the lower bound is greater than

$$k \frac{\beta^2}{4} \left[1 - (1 - \mathbb{P}_{t \sim \mathcal{N}(0,1)}(t \geq \beta\sqrt{n}/\sigma))^{d/k-1} \right] \geq k \frac{\beta^2}{4} \left[1 - (1 - \frac{1}{2} \exp(-\beta^2 n/\sigma^2))^{d/k-1} \right],$$

using the Gaussian tail bound $\mathbb{P}_{t \sim \mathcal{N}(0,1)}(t \geq z) \geq \frac{1}{2} \exp(-z^2)$. We can then consider $\beta^2 = \frac{\sigma^2}{n} \sqrt{2 \log(d/k)}$, leading to a lower bound

$$\frac{\sigma^2 k}{4n} \log(d/k) \left[(1 - (1 - \frac{1}{2}(k/d))^{d/k-1}) \right]$$

which is greater than $\frac{\sigma^2 k}{8n} \log(d/k)$ if $k \leq 2d$. We obtain the same lower-bound as the upper-bound for ℓ_0 -penalty-based methods in Chapter 8.

11.2 Optimization lower bounds

In this section, we consider ways of obtaining lower-bounds of performance for optimization algorithms. While the statistical lower-bounds from the previous section were not explicitly giving hard problems, the algorithmic lower bounds of this section will explicitly build such hard problems.

11.2.1 Convex optimization

In order to obtain computational lower bounds for convex optimization, which is notoriously hard in general in computer science, we will rely on a very 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 [80, Section 2.1.2] and [31, Section 3.5], and assume that we want to minimize a convex function F defined on \mathbb{R}^d . The algorithm starts from $\theta_0 = 0$, and can only query points in the span of the observed gradients or some sub-gradients of F at the previous observed points.

The key is to find 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 only achieve the minimum on the first k variables.

Non-smooth functions. We consider the following function, which we will dedicated to a given number of iterations k :

$$F(\theta) = \eta \max_{i \in \{1, \dots, k+1\}} \theta_i + \frac{\mu}{2} \|\theta\|_2^2,$$

for $k < d$, and η, μ positive parameters that will be set later.

The subdifferential of $F(\theta)$ is equal to

$$\mu\theta + \eta \cdot \text{hull}(\{e_i, \theta_i = \max_{i' \in \{1, \dots, k+1\}} \theta_{i'}\}),$$

which is bounded in ℓ_2 -norm on the ball of radius R , by $\mu R + \eta$ (here e_i denotes the i -th basis vector). We consider the oracle where the gradient which is output is $\mu\theta + \eta e_i$, where i is the smallest index within maximizers of $\theta_{i'}$.


Starting from $\theta_0 = 0$, θ_1 is supported on the first variable, and by recursion, after $k \leq d$ steps of subgradient descent, θ_k is supported on the first k variables. Since $k < d$, then $(\theta_k)_{k+1} = 0$, so $F(\theta_k) \geq 0$. Minimizing on the span of the first k variables leads to, by symmetry, $\theta_* = \kappa \sum_{i=1}^{k+1} e_i$, for a certain κ which minimizes $\eta\kappa + \frac{(k+1)\mu}{2}\kappa^2$, so that $\kappa = -\frac{\eta}{\mu(k+1)}$, and thus $\theta_* = -\frac{\eta}{\mu(k+1)} \sum_{i=1}^{k+1} e_i$, with value $F(\theta_*) = -\frac{\eta^2}{2\mu(k+1)}$. Thus

$$F(\theta_k) - F(\theta_*) \geq 0 - F(\theta_*) = \frac{\eta^2}{2\mu(k+1)},$$

with $\|\theta_*\|_2^2 = \frac{\eta^2}{\mu^2(k+1)}$.

In order to build a B -Lipschitz-continuous function on a ball of center 0 and radius D , we can take $\eta = B/2$, and $D = B/(2\mu)$, and we get a lower bound of $\frac{B^2}{8\mu k}$.

With $\mu = \frac{B}{D} \frac{1}{1+\sqrt{k+1}}$ and $\eta = B \frac{\sqrt{k+1}}{1+\sqrt{k+1}}$, we also get a B -Lipschitz continuous function, and we get the lower bound $\frac{DB}{2(1+\sqrt{k+1})}$, which is valid as long as $k < d$.

 The lower bounds are only valid for $k < d$, because there exists algorithms which are linearly convergence in this setting with a constant that depends on d , such as the ellipsoid method or the center of mass method (see [31] for details).

Smooth functions. We consider a sequence of quadratic function on \mathbb{R}^d . We need that the gradient for iterates supported on the first i components is supported on the first $i+1$ components. We consider the example from [80, Section 2.1.2]:

$$F_k(\theta) = \frac{L}{4} \left\{ \frac{1}{2} \left[\theta_1^2 + \theta_k^2 + \sum_{i=1}^{k-1} (\theta_i - \theta_{i+1})^2 \right] - \theta_1 \right\}.$$

The function F_k is convex, and smooth, with a smoothness constant which 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


$$\|\theta_*^{(k)}\|_2^2 = \sum_{i=1}^k \left(1 - \frac{i}{k+1}\right)^2 \leq \frac{k+1}{3}.$$

By construction, if θ is supported in the first i components for $i < k$, then $F'_k(\theta)$ is supported on the first $i+1$ components. Thus, the i -th iterate is supported on the the first i components, and thus the lowest attainable value is $F_i(\theta_*^{(i)})$.

Given this set of functions, for a given k such that $k \leq \frac{d-1}{2}$, and 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:

$$\frac{F_{2k+1}(\theta_k) - F_{2k+1}^*}{\|\theta_0 - \theta_*\|_2^2} \geq \frac{L}{8} \frac{\frac{1}{k+1} - \frac{1}{2k+2}}{\frac{2k+2}{3}} \geq \frac{3L}{32} \frac{1}{(k+1)^2}.$$

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 [80], 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, we have:

$$\|\theta_k - \theta_*\|_2^2 \geq \sum_{i=k+1}^{\infty} q^{2i} = q^{2k} \|\theta_*\|_2^2.$$

This leads to $F(\theta_k) - F_* \geq \frac{\mu}{2} \|\theta_k - \theta_*\|_2^2 \geq q^{2k} \|\theta_0 - \theta_*\|_2^2$.

11.2.2 Non-convex optimization

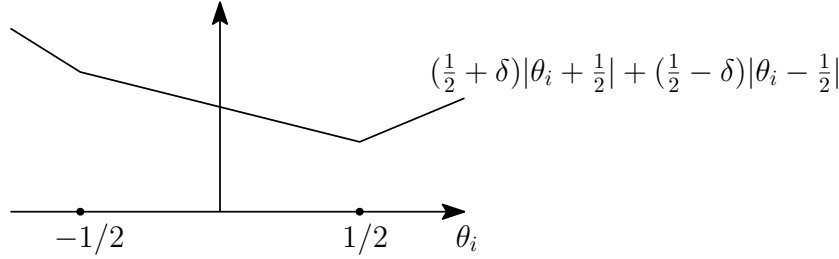
See <https://francisbach.com/optimization-is-as-hard-as-approximation/>

11.3 Lower bounds for stochastic gradient descent (◆)

We follow the exposition from [1] and consider a function

$$F_\alpha(\theta) = \frac{B}{2d} \sum_{i=1}^d \left\{ \left(\frac{1}{2} + \alpha_i \delta \right) \cdot \left| \theta_i + \frac{1}{2} \right| + \left(\frac{1}{2} - \alpha_i \delta \right) \cdot \left| \theta_i - \frac{1}{2} \right| \right\}, \quad (11.4)$$

with $\alpha \in \{-1, 1\}^d$ a well chosen vector and $\delta \in (0, 1/4]$, and $B > 0$. One element of the sum is plotted below.



The function F_α is convex and Lipschitz-continuous with gradients bounded in L_2 -norm by B/\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 lead to identification of α among a set of α 's which are sufficiently different.

Lemma 11.3 *If $\alpha, \beta \in \{-1, 1\}^d$, and $F_\alpha(\theta) - F_\alpha^* \leq \varepsilon$, then $F_\beta(\theta) - F_\beta^* \geq \frac{B\delta}{2d} \|\alpha - \beta\|_1 - \varepsilon$.*

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

$$F_\beta(\theta) + F_\alpha(\theta) - F_\beta^* - F_\alpha^* = \frac{B}{2d} \sum_{i, \alpha_i \neq \beta_i} \left\{ \left| \theta_i + \frac{1}{2} \right| + \left| \theta_i - \frac{1}{2} \right| + 2\delta - 1 \right\} \geq \frac{B\delta}{2d} \|\alpha - \beta\|_1.$$

■

Thus, if we consider M points $\alpha^{(1)}, \dots, \alpha^{(M)} \in \{-1, 1\}^d$ such that $\|\alpha^{(i)} - \alpha^{(j)}\|_1 \geq \frac{d}{2}$ (with potentially $M \geq \exp(d/8)$ such points from Lemma 11.2), then, if $\varepsilon < \frac{B\delta}{4}$, minimizing up to ε exactly identifies which of the functions $F_{\alpha^{(i)}}$ was being minimized.

Moreover, if $\hat{\theta}$ is random then, denoting $\mathcal{A} = \{\alpha^{(1)}, \dots, \alpha^{(M)}\}$,

$$\sup_{\alpha \in \mathcal{A}} \mathbb{E}_\alpha [F_\alpha(\hat{\theta}) - F_\alpha^*] \geq \varepsilon \cdot \sup_{\alpha \in \mathcal{A}} \mathbb{P}_\alpha (F_\alpha(\hat{\theta}) - F_\alpha^* > \varepsilon) \geq \varepsilon \cdot \frac{1}{|\mathcal{A}|} \sum_{\alpha \in \mathcal{A}} \mathbb{P}_\alpha (F_\alpha(\hat{\theta}) - F_\alpha^* > \varepsilon).$$

From an estimate $\hat{\theta}$, we can build a test $g(\hat{\theta}) \in \mathcal{A}$ by selecting the (unique if $\varepsilon < \frac{B\delta}{4}$) $\alpha \in \mathcal{A}$ such that $F_\alpha(\hat{\theta}) - F_\alpha^* \leq \varepsilon$ if it exists, and uniformly at random in \mathcal{A} otherwise. Therefore, the minimax performance is greater than ε times the probability of mistake of the best possible test.

We consider the following stochastic oracle:

- (1) pick some coordinate $i \in \{1, \dots, d\}$ uniformly at random,
- (2) draw a Bernoulli random variable $b_i \in \{0, 1\}$ with parameter $\frac{1}{2} + \alpha_i \delta$,
- (3) consider $\hat{F}(\theta) = cb_i|\theta_i + \frac{1}{2}| + c(1 - b_i)|\theta_i - \frac{1}{2}|$, with gradient

$$\hat{F}'_\alpha(\theta) = \frac{B}{2} [b_i \text{sign}(\theta_i + 1/2) + (1 - b_i) \text{sign}(\theta_i - 1/2)].$$

The stochastic gradients have an ℓ_2 -norm bounded by B . Moreover, observation of the gradient for $\theta \in [-1/2, 1/2]^d$ reveals the outcome of the Bernoulli random variable b_i .

Therefore, after k steps, we can apply Fano's inequality to the following set-up: the random variable $\alpha \in \mathcal{A}$ is uniform, and given α , we sample independently k 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 k times because each of the k gradients are sampled independently.

The mutual information can be decomposed as

$$I(\alpha, (i, b)) = I(\alpha, i) + I(\alpha, b|i) = 0 + \mathbb{E}_i \mathbb{E}_\alpha [D_{\text{KL}}(p(b|i, \alpha) || p(b|i))]$$

where $p(b|i, \alpha)$ and $p(b|i)$ denotes the probability distribution of b . Thus

$$\begin{aligned} I(\alpha, (i, b)) &= \mathbb{E}_i \mathbb{E}_\alpha [D_{\text{KL}}(p(b|i, \alpha) || \frac{1}{|\mathcal{A}|} \sum_{\alpha' \in \mathcal{A}} p(b|i, \alpha'))] \\ &\leq \frac{1}{|\mathcal{A}|} \sum_{\alpha' \in \mathcal{A}} \mathbb{E}_i \mathbb{E}_\alpha [D_{\text{KL}}(p(b|i, \alpha) || p(b|i, \alpha'))] \end{aligned}$$

Since $p(b|i, \alpha)$ is Bernoulli random variable with parameter $\frac{1}{2} + \delta$ or $\frac{1}{2} - \delta$, the KL divergences above are bounded by the KL between two Bernoulli random variables with the two different parameters, that is,

$$\begin{aligned} I(\alpha, (i, b)) &\leq (\frac{1}{2} + \delta) \log \frac{\frac{1}{2} + \delta}{\frac{1}{2} - \delta} + (\frac{1}{2} - \delta) \log \frac{\frac{1}{2} - \delta}{\frac{1}{2} + \delta} = 2\delta \log \frac{1 + 2\delta}{1 - 2\delta} \\ &= 2\delta \log \left(1 + \frac{4\delta}{1 - 2\delta} \right) \leq \frac{8\delta^2}{1 - 2\delta} \leq 16\delta^2 \text{ if } \delta \in [0, 1/4]. \end{aligned}$$

Therefore the minimax lower bound is greater than

$$\varepsilon \left(1 - \frac{16k\delta^2 - \log 2}{\log M} \right) \geq \varepsilon \left(1 - \frac{16k\delta^2 - \log 2}{d/8} \right).$$

Thus, we need $256k\delta^2 \geq d$, and then $B\delta/4$ is the lower bound on the rate, so that the lower bound is

$$\frac{1}{16} \sqrt{\frac{d}{k}},$$

which is the desired lower-bound in $O(DB/\sqrt{k})$ where D is the diameter of the set of θ . The lower-bound is the same as the upper-bound achieved by stochastic gradient descent in Chapter 5.

The result above can be extended to strongly-convex problems (see [1]).

Bibliography

- [1] Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.
- [2] Alekh Agarwal, Martin J. Wainwright, Peter L. Bartlett, and Pradeep K. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, 2009.
- [3] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2020.
- [4] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. Density of Lipschitz functions and equivalence of weak gradients in metric measure spaces. *Revista Matemática Iberoamericana*, 29(3):969–996, 2013.
- [5] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337 – 404, 1950.
- [6] Jean-Yves Audibert, Alexandre B Tsybakov, et al. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2):608–633, 2007.
- [7] Chloé-Agathe Azencott. *Introduction au Machine Learning*. Dunod, 2019.
- [8] Francis Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(Jun):1019–1048, 2008.
- [9] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209. PMLR, 2013.
- [10] Francis Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015.
- [11] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.
- [12] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.
- [13] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

- [14] Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems*, 2013.
- [15] Nikhil Bansal and Anupam Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(1):1–32, 2019.
- [16] Andrew R. Barron and Jason M. Klusowski. Approximation and estimation for high-dimensional deep learning networks. Technical Report 1809.03090, arXiv, 2018.
- [17] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- [18] Peter L. Bartlett and Shahrar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
- [19] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- [20] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.
- [21] Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, volume 3. Springer, 2004.
- [22] Rajendra Bhatia. *Positive Definite Matrices*, volume 24. Princeton University Press, 2009.
- [23] Gérard Biau and Luc Devroye. *Lectures on the Nearest Neighbor Method*, volume 246. Springer, 2015.
- [24] Jérôme Bolte, Aris Daniilidis, Olivier Ley, and Laurent Mazet. Characterizations of lojasiewicz inequalities and applications. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.
- [25] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.
- [26] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.
- [27] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.
- [28] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [29] Leo Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993.
- [30] Leo Breiman and David Freedman. How many variables should be entered in a regression equation? *Journal of the American Statistical Association*, 78(381):131–136, 1983.

- [31] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.
- [32] Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56. Institute of Mathematical Statistics, 2007.
- [33] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 3437–3445, 2014.
- [34] George H. Chen and Devavrat Shah. *Explaining the Success of Nearest Neighbor Methods in Prediction*. Now Publishers, 2018.
- [35] Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, pages 3036–3046, 2018.
- [36] Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Proceedings of the Conference on Learning Theory*, 2020.
- [37] Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, 2009.
- [38] Andreas Christmann and Ingo Steinwart. *Support Vector Machines*. Springer, 2008.
- [39] Thomas M. Cover and Joy A. Thomas. *Elements of information Theory*. John Wiley & Sons, 1999.
- [40] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 2014.
- [41] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic Theory of Pattern Recognition*, volume 31. Springer Science & Business Media, 1996.
- [42] David L. Donoho and Iain M. Johnstone. Minimax risk over ℓ_p -balls for ℓ_q -error. *Probability Theory and Related Fields*, 99(2):277–303, 1994.
- [43] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.
- [44] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d’Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *arXiv preprint arXiv:1901.01608*, 2019.
- [45] Christophe Giraud. *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC, 2014.
- [46] Christophe Giraud, Sylvie Huet, and Nicolas Verzelen. High-dimensional regression with unknown variance. *Statistical Science*, 27(4):500–518, 2012.
- [47] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

- [48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [49] Robert M. Gower, Mark Schmidt, Francis Bach, and Peter Richtarik. Variance-reduced methods for machine learning. Technical Report 2010.00892, arXiv, 2020.
- [50] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *In International Conference on Machine Learning*, 2018.
- [51] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [52] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-free Theory of Nonparametric Regression*. Springer Science & Business Media, 2006.
- [53] Zaid Harchaoui, Francis Bach, and Eric Moulines. Testing for homogeneity with kernel fisher discriminant analysis. Technical Report 0804.1026, arXiv, 2008.
- [54] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. Technical Report 903.08560, arXiv, 2019.
- [55] Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on Learning Theory*, 2012.
- [56] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, 2013.
- [57] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, volume 26, 2013.
- [58] Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017.
- [59] Anatoli Juditsky and Arkadi Nemirovski. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods. *Optimization for Machine Learning*, pages 121–148, 2011.
- [60] Anatoli Juditsky and Arkadi Nemirovski. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, 30(9):149–183, 2011.
- [61] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.
- [62] George Kimeldorf and Grace Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.
- [63] Jason M. Klusowski and Andrew R. Barron. 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, 2018.

- [64] Vladimir Koltchinskii and Olexandra Beznosova. Exponential convergence rates in classification. In *International Conference on Computational Learning Theory*, pages 295–307. Springer, 2005.
- [65] Nicolas Le Roux and Yoshua Bengio. Continuous neural networks. In *Artificial Intelligence and Statistics*, pages 404–411, 2007.
- [66] Guillaume Lecu   and Shahar Mendelson. Performance of empirical risk minimization in linear aggregation. *Bernoulli*, 22(3):1520–1534, 2016.
- [67] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.
- [68] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. Technical Report 2001.03040, arXiv, 2020.
- [69] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2019.
- [70] Chao Ma, Stephan Wojtowytsch, and Lei Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don’t. Technical Report 2009.10713, arXiv, 2020.
- [71] Julien Mairal, Francis Bach, Jean Ponce, et al. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.
- [72] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.
- [73] Ron Meir and Tong Zhang. Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research*, 4(Oct):839–860, 2003.
- [74] Stanislav Minsker. On some extensions of Bernstein’s inequality for self-adjoint operators. *Statistics & Probability Letters*, 127:111–119, 2017.
- [75] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.
- [76] Jaouad Mourtada. Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. Technical Report 1912.10754, arXiv, 2019.
- [77] Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.
- [78] Yurii Nesterov. *Introductory Lectures on Convex Optimization: a Basic Course*. Kluwer, 2004.
- [79] Yurii Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep.*, 76, 2007.
- [80] Yurii Nesterov. *Lectures on Convex Optimization*, volume 137. Springer, 2018.

- [81] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.
- [82] Alex Nowak, Francis Bach, and Alessandro Rudi. Sharp analysis of learning with discrete losses. In *International Conference on Artificial Intelligence and Statistics*, pages 1920–1929, 2019.
- [83] Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. A general theory for structured prediction with smooth convex surrogates. Technical Report 1902.01958, arXiv, 2019.
- [84] Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. Consistent structured prediction with Max-Min Margin Markov Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.
- [85] Roberto Imbuzeiro Oliveira. The lower tail of random quadratic forms, with applications to ordinary least squares and restricted eigenvalue properties. Technical Report 1312.2903, arXiv, 2013.
- [86] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the Lasso and its dual. *Journal of Computational and Graphical statistics*, 9(2):319–337, 2000.
- [87] Dmitrii Ostrovskii and Francis Bach. Finite-sample analysis of M-estimators using self-concordance. *Electronic Journal of Statistics*, 15(1):326–391, 2021.
- [88] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. In *Advances in Neural Information Processing Systems*, pages 8114–8124, 2018.
- [89] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.
- [90] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics, Volume 2*. Academic press, 1978.
- [91] Philippe Rigollet and Alexander B. Tsybakov. Linear and convex aggregation of density estimators. *Mathematical Methods of Statistics*, 16(3):260–280, 2007.
- [92] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.
- [93] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.
- [94] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.
- [95] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2001.
- [96] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

- [97] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [98] Karthik Sridharan, Shai Shalev-Shwartz, and Nathan Srebro. Fast rates for regularized objectives. In *Advances in Neural Information Processing Systems*, 2009.
- [99] George W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.
- [100] Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, pages 595–620, 1977.
- [101] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.
- [102] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Science & Business Media, 2008.
- [103] Aad W. Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.
- [104] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47. Cambridge University Press, 2018.
- [105] Martin J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.
- [106] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT press Cambridge, MA, 2006.
- [107] Yuhong Yang. Minimax nonparametric classification. i. rates of convergence. *IEEE Transactions on Information Theory*, 45(7):2271–2284, 1999.
- [108] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.
- [109] Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, volume 26, 2013.
- [110] Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006.