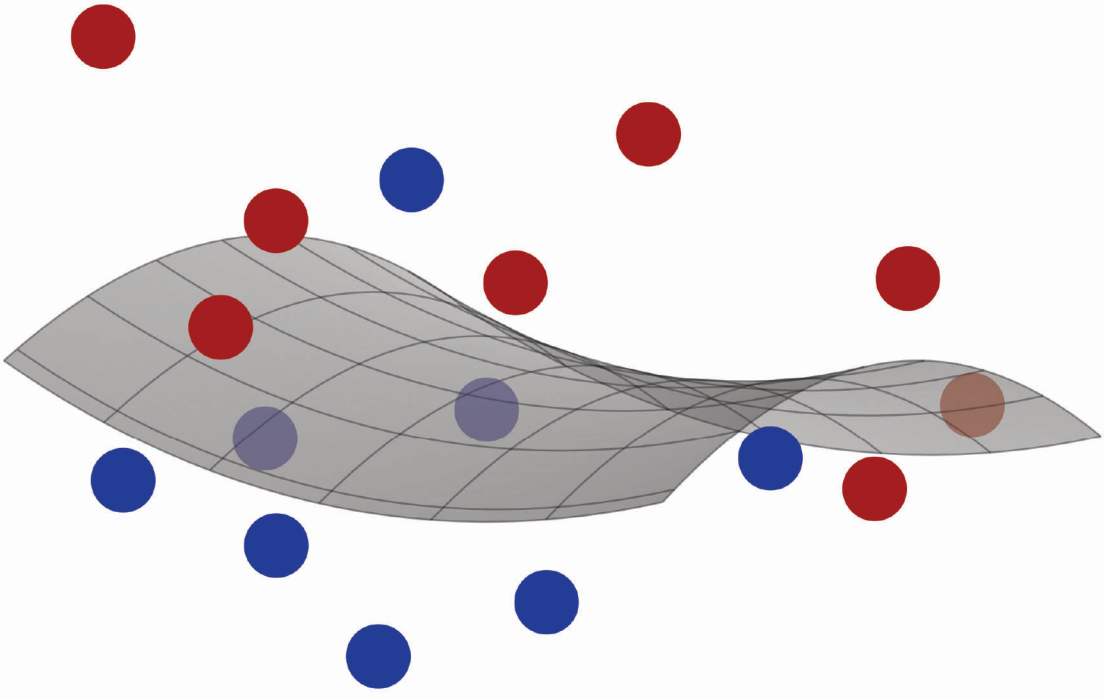
Foundations of Machine Learning

second edition



Mehryar Mohri,
Afshin Rostamizadeh,
and Ameet Talwalkar

Foundations of Machine Learning

second edition

Adaptive Computation and Machine Learning

Francis Bach, Editor

A complete list of books published in The Adaptive Computations and Machine Learning series appears at the back of this book.

Foundations of Machine Learning

second edition

Mehryar Mohri

Afshin Rostamizadeh

Ameet Talwalkar

The MIT Press
Cambridge, Massachusetts
London, England

This book was set in LaTeX by the authors. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Names: Mohri, Mehryar, author. | Rostamizadeh, Afshin, author. | Talwalkar, Ameet, author.
Title: Foundations of machine learning / Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar.
Description: Second edition. | Cambridge, MA : The MIT Press, [2018] | Series: Adaptive computation and machine learning series | Includes bibliographical references and index.
Identifiers: LCCN 2018022812 | ISBN 9780262039406 (hardcover : alk. paper)
Subjects: LCSH: Machine learning. | Computer algorithms.
Classification: LCC Q325.5 .M64 2018 | DDC 006.3/1--dc23 LC record available at https://lccn.loc.gov/2018022812

10 9 8 7 6 5 4 3 2

Contents

Preface

This book is a general introduction to machine learning that can serve as a reference book for researchers and a textbook for students. It covers fundamental modern topics in machine learning while providing the theoretical basis and conceptual tools needed for the discussion and justification of algorithms. It also describes several key aspects of the application of these algorithms.

We have aimed to present the most novel theoretical tools and concepts while giving concise proofs, even for relatively advanced results. In general, whenever possible, we have chosen to favor succinctness. Nevertheless, we discuss some crucial complex topics arising in machine learning and highlight several open research questions. Certain topics often merged with others or treated with insufficient attention are discussed separately here and with more emphasis: for example, a different chapter is reserved for multi-class classification, ranking, and regression.

Although we cover a very wide variety of important topics in machine learning, we have chosen to omit a few important ones, including graphical models and neural networks, both for the sake of brevity and because of the current lack of solid theoretical guarantees for some methods.

The book is intended for students and researchers in machine learning, statistics and other related areas. It can be used as a textbook for both graduate and advanced undergraduate classes in machine learning or as a reference text for a research seminar. The first three or four chapters of the book lay the theoretical foundation for the subsequent material. Other chapters are mostly self-contained, with the exception of chapter 6 which introduces some concepts that are extensively used in later ones and chapter 13, which is closely related to chapter 12. Each chapter concludes with a series of exercises, with full solutions presented separately.

The reader is assumed to be familiar with basic concepts in linear algebra, probability, and analysis of algorithms. However, to further help, we have included an extensive appendix presenting a concise review of linear algebra, an introduction to convex optimization, a brief probability review, a collection of concentration

inequalities useful to the analyses and discussions in this book, and a short introduction to information theory.

Our goal has been to give a unified presentation of multiple topics and areas, as opposed to a more specialized presentation adopted by some books which favor a particular viewpoint, such as for example a Bayesian view, or a particular topic, such as for example kernel methods. The theoretical foundation of this book and its deliberate emphasis on proofs and analysis make it also very distinct from many other presentations.

In this second edition, we have updated the entire book. The changes include a different writing style in most chapters, new figures and illustrations, many simplifications, some additions to existing chapters, in particular chapter 6 and chapter 17, and several new chapters. We have added a full chapter on model selection (chapter 4), which is an important topic that was only briefly discussed in the previous edition. We have also added a new chapter on Maximum Entropy models (chapter 12) and a new chapter on Conditional Maximum Entropy models (chapter 13) which are both essential topics in machine learning. We have also significantly changed the appendix. In particular, we have added a full section on Fenchel duality to appendix B on convex optimization, made a number of changes and additions to appendix D dealing with concentration inequalities, added appendix E on information theory, and updated most of the material. Additionally, we have included a number of new exercises and their solutions for existing and new chapters.

Most of the material presented here takes its origins in a machine learning graduate course (*Foundations of Machine Learning*) taught by the first author at the Courant Institute of Mathematical Sciences in New York University over the last fourteen years. This book has considerably benefited from the comments and suggestions from students in these classes, along with those of many friends, colleagues and researchers to whom we are deeply indebted.

We are particularly grateful to Corinna Cortes and Yishay Mansour who made a number of key suggestions for the design and organization of the material presented in the first edition, with detailed comments that we have fully taken into account and that have greatly improved the presentation. We are also grateful to Yishay Mansour for using a preliminary version of the first edition of the book for teaching, and for reporting his feedback to us.

We also thank for discussions, suggested improvement, and contributions of many kinds the following colleagues and friends from academic and corporate research laboratories: Jacob Abernethy, Cyril Allauzen, Kareem Amin, Stephen Boyd, Aldo Corbisiero, Giulia DeSalvo, Claudio Gentile, Spencer Greenberg, Lisa Hellerstein, Sanjiv Kumar, Vitaly Kuznetsov, Ryan McDonald, Andrès Muñoz Medina, Tyler Neylon, Peter Norvig, Fernando Pereira, Maria Pershina, Borja de Balle Pigem,

Ashish Rastogi, Michael Riley, Dmitry Storcheus, Ananda Theertha Suresh, Umar Syed, Csaba Szepesvári, Toshiyuki Tanaka, Eugene Weinstein, Jason Weston, Scott Yang, and Ningshan Zhang.

Finally, we thank the MIT Press publication team for their help and support in the development of this text.

1 Introduction

This chapter presents a preliminary introduction to machine learning, including an overview of some key learning tasks and applications, basic definitions and terminology, and the discussion of some general scenarios.

1.1 What is machine learning?

Machine learning can be broadly defined as computational methods using experience to improve performance or to make accurate predictions. Here, *experience* refers to the past information available to the learner, which typically takes the form of electronic data collected and made available for analysis. This data could be in the form of digitized human-labeled training sets, or other types of information obtained via interaction with the environment. In all cases, its quality and size are crucial to the success of the predictions made by the learner.

An example of a learning problem is how to use a finite sample of randomly selected documents, each labeled with a topic, to accurately predict the topic of unseen documents. Clearly, the larger is the sample, the easier is the task. But the difficulty of the task also depends on the quality of the labels assigned to the documents in the sample, since the labels may not be all correct, and on the number of possible topics.

Machine learning consists of designing efficient and accurate prediction *algorithms*. As in other areas of computer science, some critical measures of the quality of these algorithms are their time and space complexity. But, in machine learning, we will need additionally a notion of *sample complexity* to evaluate the sample size required for the algorithm to learn a family of concepts. More generally, theoretical learning guarantees for an algorithm depend on the complexity of the concept classes considered and the size of the training sample.

Since the success of a learning algorithm depends on the data used, machine learning is inherently related to data analysis and statistics. More generally, learning

techniques are data-driven methods combining fundamental concepts in computer science with ideas from statistics, probability and optimization.

1.2 What kind of problems can be tackled using machine learning?

Predicting the label of a document, also known as document classification, is by no means the only learning task. Machine learning admits a very broad set of practical applications, which include the following:

- Text or document classification. This includes problems such as assigning a topic to a text or a document, or determining automatically if the content of a web page is inappropriate or too explicit; it also includes spam detection.
- Natural language processing (NLP). Most tasks in this field, including part-of-speech tagging, named-entity recognition, context-free parsing, or dependency parsing, are cast as learning problems. In these problems, predictions admit some structure. For example, in part-of-speech tagging, the prediction for a sentence is a sequence of part-of-speech tags labeling each word. In context-free parsing the prediction is a tree. These are instances of richer learning problems known as *structured prediction problems*.
- Speech processing applications. This includes speech recognition, speech synthesis, speaker verification, speaker identification, as well as sub-problems such as language modeling and acoustic modeling.
- Computer vision applications. This includes object recognition, object identification, face detection, Optical character recognition (OCR), content-based image retrieval, or pose estimation.
- Computational biology applications. This includes protein function prediction, identification of key sites, or the analysis of gene and protein networks.
- Many other problems such as fraud detection for credit card, telephone or insurance companies, network intrusion, learning to play games such as chess, backgammon, or Go, unassisted control of vehicles such as robots or cars, medical diagnosis, the design of recommendation systems, search engines, or information extraction systems, are tackled using machine learning techniques.

This list is by no means comprehensive. Most prediction problems found in practice can be cast as learning problems and the practical application area of machine learning keeps expanding. The algorithms and techniques discussed in this book can be used to derive solutions for all of these problems, though we will not discuss in detail these applications.

1.3 Some standard learning tasks

The following are some standard machine learning tasks that have been extensively studied:

- *Classification*: this is the problem of assigning a category to each item. For example, document classification consists of assigning a category such as *politics*, *business*, *sports*, or *weather* to each document, while image classification consists of assigning to each image a category such as *car*, *train*, or *plane*. The number of categories in such tasks is often less than a few hundreds, but it can be much larger in some difficult tasks and even unbounded as in OCR, text classification, or speech recognition.
- *Regression*: this is the problem of predicting a real value for each item. Examples of regression include prediction of stock values or that of variations of economic variables. In regression, the penalty for an incorrect prediction depends on the magnitude of the difference between the true and predicted values, in contrast with the classification problem, where there is typically no notion of closeness between various categories.
- *Ranking*: this is the problem of learning to order items according to some criterion. Web search, e.g., returning web pages relevant to a search query, is the canonical ranking example. Many other similar ranking problems arise in the context of the design of information extraction or natural language processing systems.
- *Clustering*: this is the problem of partitioning a set of items into homogeneous subsets. Clustering is often used to analyze very large data sets. For example, in the context of social network analysis, clustering algorithms attempt to identify natural *communities* within large groups of people.
- *Dimensionality reduction* or *manifold learning*: this problem consists of transforming an initial representation of items into a lower-dimensional representation while preserving some properties of the initial representation. A common example involves preprocessing digital images in computer vision tasks.

The main practical objectives of machine learning consist of generating accurate predictions for unseen items and of designing efficient and robust algorithms to produce these predictions, even for large-scale problems. To do so, a number of algorithmic and theoretical questions arise. Some fundamental questions include: Which concept families can actually be learned, and under what conditions? How well can these concepts be learned computationally?

1.4 Learning stages

Here, we will use the canonical problem of spam detection as a running example to illustrate some basic definitions and describe the use and evaluation of machine learning algorithms in practice, including their different stages.

Spam detection is the problem of learning to automatically classify email messages as either SPAM or non-SPAM. The following is a list of definitions and terminology commonly used in machine learning:

- *Examples*: Items or instances of data used for learning or evaluation. In our spam problem, these examples correspond to the collection of email messages we will use for learning and testing.
- *Features*: The set of attributes, often represented as a vector, associated to an example. In the case of email messages, some relevant features may include the length of the message, the name of the sender, various characteristics of the header, the presence of certain keywords in the body of the message, and so on.
- *Labels*: Values or categories assigned to examples. In classification problems, examples are assigned specific categories, for instance, the SPAM and non-SPAM categories in our binary classification problem. In regression, items are assigned real-valued labels.
- *Hyperparameters*: Free parameters that are not determined by the learning algorithm, but rather specified as inputs to the learning algorithm.
- *Training sample*: Examples used to train a learning algorithm. In our spam problem, the training sample consists of a set of email examples along with their associated labels. The training sample varies for different learning scenarios, as described in section 1.5.
- *Validation sample*: Examples used to tune the parameters of a learning algorithm when working with labeled data. The validation sample is used to select appropriate values for the learning algorithm's free parameters (hyperparameters).
- *Test sample*: Examples used to evaluate the performance of a learning algorithm. The test sample is separate from the training and validation data and is not made available in the learning stage. In the spam problem, the test sample consists of a collection of email examples for which the learning algorithm must predict labels based on features. These predictions are then compared with the labels of the test sample to measure the performance of the algorithm.
- *Loss function*: A function that measures the difference, or loss, between a predicted label and a true label. Denoting the set of all labels as $\mathcal{Y}$ and the set of possible predictions as $\mathcal{Y}'$, a loss function L is a mapping $L\colon \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}_+$. In most cases, $\mathcal{Y}' = \mathcal{Y}$ and the loss function is bounded, but these conditions do not always hold. Common examples of loss functions include the zero-one (or

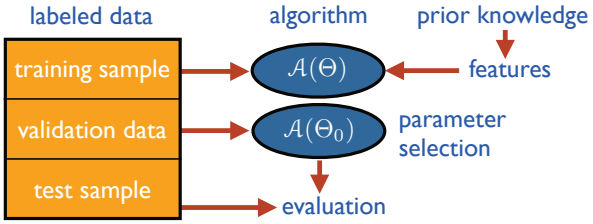


Figure 1.1
Illustration of the typical stages of a learning process.

misclassification) loss defined over $\{-1,+1\} \times \{-1,+1\}$ by $L(y,y') = 1_{y' \neq y}$ and the squared loss defined over $\mathcal{I} \times \mathcal{I}$ by $L(y,y') = (y'-y)^2$, where $\mathcal{I} \subseteq \mathbb{R}$ is typically a bounded interval.

- *Hypothesis set*: A set of functions mapping features (feature vectors) to the set of labels $\mathcal{Y}$. In our example, these may be a set of functions mapping email features to $\mathcal{Y} = \{\text{SPAM}, \text{non-SPAM}\}$. More generally, hypotheses may be functions mapping features to a different set $\mathcal{Y}'$. They could be linear functions mapping email feature vectors to real numbers interpreted as *scores* ($\mathcal{Y}' = \mathbb{R}$), with higher score values more indicative of SPAM than lower ones.

We now define the learning stages of our spam problem (see figure 1.1). We start with a given collection of labeled examples. We first randomly partition the data into a training sample, a validation sample, and a test sample. The size of each of these samples depends on a number of different considerations. For example, the amount of data reserved for validation depends on the number of hyperparameters of the algorithm, which are represented here by the vector Θ. Also, when the labeled sample is relatively small, the amount of training data is often chosen to be larger than that of the test data since the learning performance directly depends on the training sample.

Next, we associate relevant features to the examples. This is a critical step in the design of machine learning solutions. Useful features can effectively guide the learning algorithm, while poor or uninformative ones can be misleading. Although it is critical, to a large extent, the choice of the features is left to the user. This choice reflects the user's *prior knowledge* about the learning task which in practice can have a dramatic effect on the performance results.

Now, we use the features selected to train our learning algorithm $\mathcal{A}$ by tuning the values of its free parameters Θ (also called *hyperparameters*). For each value of these parameters, the algorithm selects a different hypothesis out of the hypothesis set. We choose the one resulting in the best performance on the validation sample (Θ_0). Finally, using that hypothesis, we predict the labels of the examples in the test sample. The performance of the algorithm is evaluated by using the loss

function associated to the task, e.g., the zero-one loss in our spam detection task, to compare the predicted and true labels. Thus, the performance of an algorithm is of course evaluated based on its test error and not its error on the training sample.

1.5 Learning scenarios

We next briefly describe some common machine learning scenarios. These scenarios differ in the types of training data available to the learner, the order and method by which training data is received and the test data used to evaluate the learning algorithm.

- *Supervised learning*: The learner receives a set of labeled examples as training data and makes predictions for all unseen points. This is the most common scenario associated with classification, regression, and ranking problems. The spam detection problem discussed in the previous section is an instance of supervised learning.
- *Unsupervised learning*: The learner exclusively receives unlabeled training data, and makes predictions for all unseen points. Since in general no labeled example is available in that setting, it can be difficult to quantitatively evaluate the performance of a learner. Clustering and dimensionality reduction are example of unsupervised learning problems.
- *Semi-supervised learning*: The learner receives a training sample consisting of both labeled and unlabeled data, and makes predictions for all unseen points. Semi-supervised learning is common in settings where unlabeled data is easily accessible but labels are expensive to obtain. Various types of problems arising in applications, including classification, regression, or ranking tasks, can be framed as instances of semi-supervised learning. The hope is that the distribution of unlabeled data accessible to the learner can help him achieve a better performance than in the supervised setting. The analysis of the conditions under which this can indeed be realized is the topic of much modern theoretical and applied machine learning research.
- *Transductive inference*: As in the semi-supervised scenario, the learner receives a labeled training sample along with a set of unlabeled test points. However, the objective of transductive inference is to predict labels only for these particular test points. Transductive inference appears to be an easier task and matches the scenario encountered in a variety of modern applications. However, as in the semi-supervised setting, the assumptions under which a better performance can be achieved in this setting are research questions that have not been fully resolved.

- *On-line learning*: In contrast with the previous scenarios, the online scenario involves multiple rounds where training and testing phases are intermixed. At each round, the learner receives an unlabeled training point, makes a prediction, receives the true label, and incurs a loss. The objective in the on-line setting is to minimize the cumulative loss over all rounds or to minimize the *regret*, that is the difference of the cumulative loss incurred and that of the best expert in hindsight. Unlike the previous settings just discussed, no distributional assumption is made in on-line learning. In fact, instances and their labels may be chosen adversarially within this scenario.
- *Reinforcement learning*: The training and testing phases are also intermixed in reinforcement learning. To collect information, the learner actively interacts with the environment and in some cases affects the environment, and receives an immediate reward for each action. The object of the learner is to maximize his reward over a course of actions and iterations with the environment. However, no long-term reward feedback is provided by the environment, and the learner is faced with the *exploration versus exploitation* dilemma, since he must choose between exploring unknown actions to gain more information versus exploiting the information already collected.
- *Active learning*: The learner adaptively or interactively collects training examples, typically by querying an oracle to request labels for new points. The goal in active learning is to achieve a performance comparable to the standard supervised learning scenario (or *passive learning* scenario), but with fewer labeled examples. Active learning is often used in applications where labels are expensive to obtain, for example computational biology applications.

In practice, many other intermediate and somewhat more complex learning scenarios may be encountered.

1.6 Generalization

Machine learning is fundamentally about *generalization*. As an example, the standard supervised learning scenario consists of using a finite sample of labeled examples to make accurate predictions about unseen examples. The problem is typically formulated as that of selecting a function out of a *hypothesis set*, that is a subset of the family of all functions. The function selected is subsequently used to label all instances, including unseen examples.

How should a hypothesis set be chosen? With a rich or *complex* hypothesis set, the learner may choose a function or predictor that is *consistent* with the training sample, that is one that commits no error on the training sample. With a less complex family, incurring some errors on the training sample may be unavoidable. But,

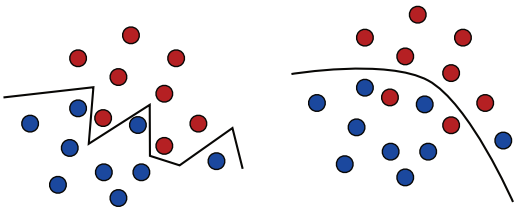


Figure 1.2
The zig-zag line on the left panel is consistent over the blue and red training sample, but it is a complex separation surface that is not likely to generalize well to unseen data. In contrast, the decision surface on the right panel is simpler and might generalize better in spite of its misclassification of a few points of the training sample.

which will lead to a better generalization? How should we define the complexity of a hypothesis set?

Figure 1.2 illustrates these two types of solution: one is a zig-zag line that perfectly separates the two populations of blue and red points and that is chosen from a complex family; the other one is a smoother line chosen from a simpler family that only imperfectly discriminates between the two sets. We will see that, in general, the best predictor on the training sample may not be the best overall. A predictor chosen from a very complex family can essentially memorize the data, but generalization is distinct from the memorization of the training labels.

We will see that the trade-off between the sample size and complexity plays a critical role in generalization. When the sample size is relatively small, choosing from a too complex a family may lead to poor generalization, which is also known as *overfitting*. On the other hand, with a too simple a family it may not be possible to achieve a sufficient accuracy, which is known as *underfitting*.

In the next chapters, we will analyze more in detail the problem of generalization and will seek to derive theoretical guarantees for learning. This will depend on different notions of complexity that we will thoroughly discuss.

2 The PAC Learning Framework

Several fundamental questions arise when designing and analyzing algorithms that learn from examples: What can be learned efficiently? What is inherently hard to learn? How many examples are needed to learn successfully? Is there a general model of learning? In this chapter, we begin to formalize and address these questions by introducing the *Probably Approximately Correct* (PAC) learning framework. The PAC framework helps define the class of learnable concepts in terms of the number of sample points needed to achieve an approximate solution, *sample complexity*, and the time and space complexity of the learning algorithm, which depends on the cost of the computational representation of the concepts.

We first describe the PAC framework and illustrate it, then present some general learning guarantees within this framework when the hypothesis set used is finite, both for the *consistent* case where the hypothesis set used contains the concept to learn and for the opposite *inconsistent* case.

2.1 The PAC learning model

We first introduce several definitions and the notation needed to present the PAC model, which will also be used throughout much of this book.

We denote by $\mathcal{X}$ the set of all possible *examples* or *instances*. $\mathcal{X}$ is also sometimes referred to as the *input space*. The set of all possible *labels* or *target values* is denoted by $\mathcal{Y}$. For the purpose of this introductory chapter, we will limit ourselves to the case where $\mathcal{Y}$ is reduced to two labels, $\mathcal{Y} = \{0, 1\}$, which corresponds to the so-called *binary classification*. Later chapters will extend these results to more general settings.

A *concept* $c\colon \mathcal{X} \to \mathcal{Y}$ is a mapping from $\mathcal{X}$ to $\mathcal{Y}$. Since $\mathcal{Y} = \{0, 1\}$, we can identify c with the subset of $\mathcal{X}$ over which it takes the value 1. Thus, in the following, we equivalently refer to a concept to learn as a mapping from $\mathcal{X}$ to $\{0, 1\}$, or as a subset of $\mathcal{X}$. As an example, a concept may be the set of points inside a triangle

or the indicator function of these points. In such cases, we will say in short that the concept to learn is a triangle. A *concept class* is a set of concepts we may wish to learn and is denoted by $\mathcal{C}$. This could, for example, be the set of all triangles in the plane.

We assume that examples are independently and identically distributed (i.i.d.) according to some fixed but unknown distribution $\mathcal{D}$. The learning problem is then formulated as follows. The learner considers a fixed set of possible concepts $\mathcal{H}$, called a *hypothesis set*, which might not necessarily coincide with $\mathcal{C}$. It receives a sample $S = (x_1, \ldots, x_m)$ drawn i.i.d. according to $\mathcal{D}$ as well as the labels $(c(x_1), \ldots, c(x_m))$, which are based on a specific target concept $c \in \mathcal{C}$ to learn. The task is then to use the labeled sample S to select a hypothesis $h_S \in \mathcal{H}$ that has a small *generalization error* with respect to the concept c. The generalization error of a hypothesis $h \in \mathcal{H}$, also referred to as the *risk* or *true error* (or simply *error*) of h is denoted by $R(h)$ and defined as follows.[1]

Definition 2.1 (Generalization error) *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying distribution $\mathcal{D}$, the* generalization error *or* risk *of h is defined by*

$$R(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq c(x)] = \mathop{\mathbb{E}}_{x \sim \mathcal{D}}\left[1_{h(x) \neq c(x)}\right], \tag{2.1}$$

where 1_ω is the indicator function of the event ω.[2]

The generalization error of a hypothesis is not directly accessible to the learner since both the distribution $\mathcal{D}$ and the target concept c are unknown. However, the learner can measure the *empirical error* of a hypothesis on the labeled sample S.

Definition 2.2 (Empirical error) *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and a sample $S = (x_1, \ldots, x_m)$, the* empirical error *or* empirical risk *of h is defined by*

$$\widehat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}. \tag{2.2}$$

Thus, the empirical error of $h \in \mathcal{H}$ is its average error over the sample S, while the generalization error is its expected error based on the distribution $\mathcal{D}$. We will see in this chapter and the following chapters a number of guarantees relating these two quantities with high probability, under some general assumptions. We can already note that for a fixed $h \in \mathcal{H}$, the expectation of the empirical error based on an i.i.d.

[1] The choice of R instead of E to denote an error avoids possible confusions with the notation for expectations and is further justified by the fact that the term *risk* is also used in machine learning and statistics to refer to an error.

[2] For this and other related definitions, the family of functions $\mathcal{H}$ and the target concept c must be measurable. The function classes we consider in this book all have this property.

sample S is equal to the generalization error:

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{R}_S(h)] = R(h). \tag{2.3}$$

Indeed, by the linearity of the expectation and the fact that the sample is drawn i.i.d., we can write

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{R}_S(h)] = \frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[1_{h(x_i)\neq c(x_i)}] = \frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[1_{h(x)\neq c(x)}],$$

for any x in sample S. Thus,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{R}_S(h)] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[1_{h(x)\neq c(x)}] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[1_{h(x)\neq c(x)}] = R(h).$$

The following introduces the *Probably Approximately Correct* (PAC) learning framework. Let n be a number such that the computational cost of representing any element $x \in \mathcal{X}$ is at most $O(n)$ and denote by size(c) the maximal cost of the computational representation of $c \in \mathcal{C}$. For example, x may be a vector in $\mathbb{R}^n$, for which the cost of an array-based representation would be in $O(n)$. In addition, let h_S denote the hypothesis returned by algorithm $\mathcal{A}$ after receiving a labeled sample S. To keep notation simple, the dependency of h_S on $\mathcal{A}$ is not explicitly indicated.

Definition 2.3 (PAC-learning) *A concept class $\mathcal{C}$ is said to be* PAC-learnable *if there exists an algorithm $\mathcal{A}$ and a polynomial function poly$(\cdot,\cdot,\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions $\mathcal{D}$ on $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $m \geq$ poly$(1/\epsilon, 1/\delta, n,$ size$(c))$:*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[R(h_S) \leq \epsilon] \geq 1-\delta. \tag{2.4}$$

If $\mathcal{A}$ further runs in poly$(1/\epsilon, 1/\delta, n,$ size$(c))$, then $\mathcal{C}$ is said to be efficiently PAC-learnable. *When such an algorithm $\mathcal{A}$ exists, it is called a* PAC-learning algorithm *for $\mathcal{C}$.*

A concept class $\mathcal{C}$ is thus PAC-learnable if the hypothesis returned by the algorithm after observing a number of points polynomial in $1/\epsilon$ and $1/\delta$ is *approximately correct* (error at most ϵ) with high *probability* (at least $1-\delta$), which justifies the PAC terminology. The parameter $\delta > 0$ is used to define the *confidence* $1-\delta$ and $\epsilon > 0$ the *accuracy* $1-\epsilon$. Note that if the running time of the algorithm is polynomial in $1/\epsilon$ and $1/\delta$, then the sample size m must also be polynomial if the full sample is received by the algorithm.

Several key points of the PAC definition are worth emphasizing. First, the PAC framework is a *distribution-free model*: no particular assumption is made about the distribution $\mathcal{D}$ from which examples are drawn. Second, the training sample and the test examples used to define the error are drawn according to the same distribution $\mathcal{D}$. This is a natural and necessary assumption for generalization to

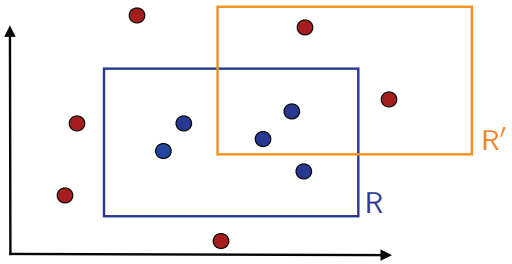


Figure 2.1
Target concept R and possible hypothesis R′. Circles represent training instances. A blue circle is a point labeled with 1, since it falls within the rectangle R. Others are red and labeled with 0.

be possible in general. It can be relaxed to include favorable *domain adaptation* problems. Finally, the PAC framework deals with the question of learnability for a concept class $\mathcal{C}$ and not a particular concept. Note that the concept class $\mathcal{C}$ is known to the algorithm, but of course the target concept $c \in \mathcal{C}$ is unknown.

In many cases, in particular when the computational representation of the concepts is not explicitly discussed or is straightforward, we may omit the polynomial dependency on n and size(c) in the PAC definition and focus only on the sample complexity.

We now illustrate PAC-learning with a specific learning problem.

Example 2.4 (Learning axis-aligned rectangles) Consider the case where the set of instances are points in the plane, $\mathcal{X} = \mathbb{R}^2$, and the concept class $\mathcal{C}$ is the set of all axis-aligned rectangles lying in $\mathbb{R}^2$. Thus, each concept c is the set of points inside a particular axis-aligned rectangle. The learning problem consists of determining with small error a target axis-aligned rectangle using the labeled training sample. We will show that the concept class of axis-aligned rectangles is PAC-learnable.

Figure 2.1 illustrates the problem. R represents a target axis-aligned rectangle and R′ a hypothesis. As can be seen from the figure, the error regions of R′ are formed by the area within the rectangle R but outside the rectangle R′ and the area within R′ but outside the rectangle R. The first area corresponds to *false negatives*, that is, points that are labeled as 0 or *negatively* by R′, which are in fact *positive* or labeled with 1. The second area corresponds to *false positives*, that is, points labeled positively by R′ which are in fact negatively labeled.

To show that the concept class is PAC-learnable, we describe a simple PAC-learning algorithm $\mathcal{A}$. Given a labeled sample S, the algorithm consists of returning the tightest axis-aligned rectangle $\mathsf{R}' = \mathsf{R}_S$ containing the points labeled with 1. Figure 2.2 illustrates the hypothesis returned by the algorithm. By definition, R_S does not produce any false positives, since its points must be included in the target concept R. Thus, the error region of R_S is included in R.

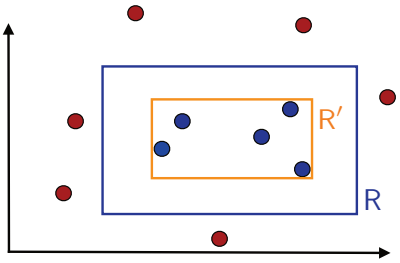


Figure 2.2
Illustration of the hypothesis $\mathsf{R}' = \mathsf{R}_S$ returned by the algorithm.

Let $\mathsf{R} \in \mathcal{C}$ be a target concept. Fix $\epsilon > 0$. Let $\mathbb{P}[\mathsf{R}]$ denote the probability mass of the region defined by R, that is the probability that a point randomly drawn according to $\mathcal{D}$ falls within R. Since errors made by our algorithm can be due only to points falling inside R, we can assume that $\mathbb{P}[\mathsf{R}] > \epsilon$; otherwise, the error of R_S is less than or equal to ϵ regardless of the training sample S received.

Now, since $\mathbb{P}[\mathsf{R}] > \epsilon$, we can define four rectangular regions r_1, r_2, r_3, and r_4 along the sides of R, each with probability at least $\epsilon/4$. These regions can be constructed by starting with the full rectangle R and then decreasing the size by moving one side as much as possible while keeping a distribution mass of at least $\epsilon/4$. Figure 2.3 illustrates the definition of these regions.

Let l, r, b, and t be the four real values defining R: $\mathsf{R} = [l, r] \times [b, t]$. Then, for example, the left rectangle r_4 is defined by $r_4 = [l, s_4] \times [b, t]$, with $s_4 = \inf\{s\colon \mathbb{P}[[l, s] \times [b, t]] \geq \epsilon/4\}$. It is not hard to see that the probability of the region $\overline{r}_4 = [l, s_4[\times[b, t]$ obtained from r_4 by excluding the rightmost side is at most $\epsilon/4$. r_1, r_2, r_3 and $\overline{r}_1$, $\overline{r}_2$, $\overline{r}_3$ are defined in a similar way.

Observe that if R_S meets all of these four regions r_i, $i \in [4]$, then, because it is a rectangle, it will have one side in each of these regions (geometric argument). Its error area, which is the part of R that it does not cover, is thus included in the union of the regions $\overline{r}_i$, $i \in [4]$, and cannot have probability mass more than ϵ. By contraposition, if $R(\mathsf{R}_S) > \epsilon$, then R_S must miss at least one of the regions r_i, $i \in [4]$. As a result, we can write

$$\begin{aligned}
\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[R(\mathsf{R}_S) > \epsilon] &\leq \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\cup_{i=1}^4 \{\mathsf{R}_S \cap r_i = \emptyset\}] && (2.5)\\
&\leq \sum_{i=1}^4 \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\{\mathsf{R}_S \cap r_i = \emptyset\}] && \text{(by the union bound)}\\
&\leq 4(1-\epsilon/4)^m && \text{(since } \mathbb{P}[r_i] \geq \epsilon/4)\\
&\leq 4\exp(-m\epsilon/4),
\end{aligned}$$

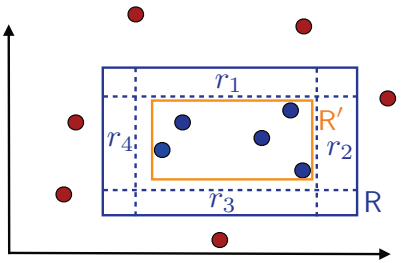


Figure 2.3
Illustration of the regions $r_1, \ldots, r_4$.

where for the last step we used the general inequality $1 - x \leq e^{-x}$ valid for all $x \in \mathbb{R}$. For any $\delta > 0$, to ensure that $\mathbb{P}_{S \sim \mathcal{D}^m}[R(\mathsf{R_S}) > \epsilon] \leq \delta$, we can impose

$$4 \exp(-\epsilon m/4) \leq \delta \Leftrightarrow m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}. \tag{2.6}$$

Thus, for any $\epsilon > 0$ and $\delta > 0$, if the sample size m is greater than $\frac{4}{\epsilon} \log \frac{4}{\delta}$, then $\mathbb{P}_{S \sim \mathcal{D}^m}[R(\mathsf{R_S}) > \epsilon] \leq \delta$. Furthermore, the computational cost of the representation of points in $\mathbb{R}^2$ and axis-aligned rectangles, which can be defined by their four corners, is constant. This proves that the concept class of axis-aligned rectangles is PAC-learnable and that the sample complexity of PAC-learning axis-aligned rectangles is in $O(\frac{1}{\epsilon} \log \frac{1}{\delta})$.

An equivalent way to present sample complexity results like (2.6), which we will often see throughout this book, is to give a *generalization bound*. A generalization bound states that with probability at least $1 - \delta$, $R(\mathsf{R_S})$ is upper bounded by some quantity that depends on the sample size m and δ. To obtain this, it suffices to set δ to be equal to the upper bound derived in (2.5), that is $\delta = 4 \exp(-m\epsilon/4)$ and solve for ϵ. This yields that with probability at least $1 - \delta$, the error of the algorithm is bounded as follows:

$$R(\mathsf{R_S}) \leq \frac{4}{m} \log \frac{4}{\delta}. \tag{2.7}$$

Other PAC-learning algorithms could be considered for this example. One alternative is to return the largest axis-aligned rectangle not containing the negative points, for example. The proof of PAC-learning just presented for the tightest axis-aligned rectangle can be easily adapted to the analysis of other such algorithms.

Note that the hypothesis set $\mathcal{H}$ we considered in this example coincided with the concept class $\mathcal{C}$ and that its cardinality was infinite. Nevertheless, the problem admitted a simple proof of PAC-learning. We may then ask if a similar proof can readily apply to other similar concept classes. This is not as straightforward because the specific geometric argument used in the proof is key. It is non-trivial to extend the proof to other concept classes such as that of non-concentric circles

(see exercise 2.4). Thus, we need a more general proof technique and more general results. The next two sections provide us with such tools in the case of a finite hypothesis set.

2.2 Guarantees for finite hypothesis sets — consistent case

In the example of axis-aligned rectangles that we examined, the hypothesis h_S returned by the algorithm was always *consistent*, that is, it admitted no error on the training sample S. In this section, we present a general sample complexity bound, or equivalently, a generalization bound, for consistent hypotheses, in the case where the cardinality $|\mathcal{H}|$ of the hypothesis set is finite. Since we consider consistent hypotheses, we will assume that the target concept c is in $\mathcal{H}$.

Theorem 2.5 (Learning bound — finite $\mathcal{H}$, consistent case) *Let $\mathcal{H}$ be a finite set of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$. Let $\mathcal{A}$ be an algorithm that for any target concept $c \in \mathcal{H}$ and i.i.d. sample S returns a consistent hypothesis h_S: $\widehat{R}_S(h_S) = 0$. Then, for any $\epsilon, \delta > 0$, the inequality $\mathbb{P}_{S\sim\mathcal{D}^m}[R(h_S) \leq \epsilon] \geq 1 - \delta$ holds if*

$$m \geq \frac{1}{\epsilon}\Big(\log|\mathcal{H}| + \log\frac{1}{\delta}\Big). \tag{2.8}$$

This sample complexity result admits the following equivalent statement as a generalization bound: for any $\epsilon, \delta > 0$, with probability at least $1 - \delta$,

$$R(h_S) \leq \frac{1}{m}\Big(\log|\mathcal{H}| + \log\frac{1}{\delta}\Big). \tag{2.9}$$

Proof: Fix $\epsilon > 0$. We do not know which consistent hypothesis $h_S \in \mathcal{H}$ is selected by the algorithm $\mathcal{A}$. This hypothesis further depends on the training sample S. Therefore, we need to give a *uniform convergence bound*, that is, a bound that holds for the set of all consistent hypotheses, which a fortiori includes h_S. Thus, we will bound the probability that some $h \in \mathcal{H}$ would be consistent and have error more than ϵ. For any $\epsilon > 0$, define $\mathcal{H}_\epsilon$ by $\mathcal{H}_\epsilon = \{h \in \mathcal{H} \colon R(h) > \epsilon\}$. The probability that a hypothesis h in $\mathcal{H}_\epsilon$ is consistent on a training sample S drawn i.i.d., that is, that it would have no error on any point in S, can be bounded as follows:

$$\mathbb{P}[\widehat{R}_S(h) = 0] \leq (1-\epsilon)^m.$$

Thus, by the union bound, the following holds:

$$\begin{aligned}
\mathbb{P}\Big[\exists h \in \mathcal{H}_\epsilon \colon \widehat{R}_S(h) = 0\Big] &= \mathbb{P}\Big[\widehat{R}_S(h_1) = 0 \vee \cdots \vee \widehat{R}_S(h_{|\mathcal{H}_\epsilon|}) = 0\Big] \\
&\leq \sum_{h\in\mathcal{H}_\epsilon} \mathbb{P}\Big[\widehat{R}_S(h) = 0\Big] \qquad \text{(union bound)} \\
&\leq \sum_{h\in\mathcal{H}_\epsilon} (1-\epsilon)^m \leq |\mathcal{H}|(1-\epsilon)^m \leq |\mathcal{H}|e^{-m\epsilon}.
\end{aligned}$$

Setting the right-hand side to be equal to δ and solving for ϵ concludes the proof. $\square$

The theorem shows that when the hypothesis set $\mathcal{H}$ is finite, a consistent algorithm $\mathcal{A}$ is a PAC-learning algorithm, since the sample complexity given by (2.8) is dominated by a polynomial in $1/\epsilon$ and $1/\delta$. As shown by (2.9), the generalization error of consistent hypotheses is upper bounded by a term that decreases as a function of the sample size m. This is a general fact: as expected, learning algorithms benefit from larger labeled training samples. The decrease rate of $O(1/m)$ guaranteed by this theorem, however, is particularly favorable.

The price to pay for coming up with a consistent algorithm is the use of a larger hypothesis set $\mathcal{H}$ containing target concepts. Of course, the upper bound (2.9) increases with $|\mathcal{H}|$. However, that dependency is only logarithmic. Note that the term $\log |\mathcal{H}|$, or the related term $\log_2 |\mathcal{H}|$ from which it differs by a constant factor, can be interpreted as the number of bits needed to represent $\mathcal{H}$. Thus, the generalization guarantee of the theorem is controlled by the ratio of this number of bits, $\log_2 |\mathcal{H}|$, and the sample size m.

We now use theorem 2.5 to analyze PAC-learning with various concept classes.

Example 2.6 (Conjunction of Boolean literals) Consider learning the concept class $\mathcal{C}_n$ of conjunctions of at most n Boolean literals $x_1, \ldots, x_n$. A Boolean literal is either a variable x_i, $i \in [n]$, or its negation $\overline{x}_i$. For $n = 4$, an example is the conjunction: $x_1 \wedge \overline{x}_2 \wedge x_4$, where $\overline{x}_2$ denotes the negation of the Boolean literal x_2. $(1, 0, 0, 1)$ is a positive example for this concept while $(1, 0, 0, 0)$ is a negative example.

Observe that for $n = 4$, a positive example $(1, 0, 1, 0)$ implies that the target concept cannot contain the literals $\overline{x}_1$ and $\overline{x}_3$ and that it cannot contain the literals x_2 and x_4. In contrast, a negative example is not as informative since it is not known which of its n bits are incorrect. A simple algorithm for finding a consistent hypothesis is thus based on positive examples and consists of the following: for each positive example $(b_1, \ldots, b_n)$ and $i \in [n]$, if $b_i = 1$ then $\overline{x}_i$ is ruled out as a possible literal in the concept class and if $b_i = 0$ then x_i is ruled out. The conjunction of all the literals not ruled out is thus a hypothesis consistent with the target. Figure 2.4 shows an example training sample as well as a consistent hypothesis for the case $n = 6$.

We have $|\mathcal{H}| = |\mathcal{C}_n| = 3^n$, since each literal can be included positively, with negation, or not included. Plugging this into the sample complexity bound for consistent hypotheses yields the following sample complexity bound for any $\epsilon > 0$ and $\delta > 0$:

$$m \geq \frac{1}{\epsilon}\Big((\log 3)n + \log \frac{1}{\delta}\Big). \tag{2.10}$$

Thus, the class of conjunctions of at most n Boolean literals is PAC-learnable. Note that the computational complexity is also polynomial, since the training cost per example is in $O(n)$. For $\delta = 0.02$, $\epsilon = 0.1$, and $n = 10$, the bound becomes

0	1	1	0	1	1	+
0	1	1	1	1	1	+
0	0	1	1	0	1	-
0	1	1	1	1	1	+
1	0	0	1	1	0	-
0	1	0	0	1	1	+
0	1	?	?	1	1	

Figure 2.4
Each of the first six rows of the table represents a training example with its label, + or −, indicated in the last column. The last row contains 0 (respectively 1) in column $i \in [6]$ if the ith entry is 0 (respectively 1) for all the positive examples. It contains "?" if both 0 and 1 appear as an ith entry for some positive example. Thus, for this training sample, the hypothesis returned by the consistent algorithm described in the text is $\overline{x}_1 \wedge x_2 \wedge x_5 \wedge x_6$.

$m \geq 149$. Thus, for a labeled sample of at least 149 examples, the bound guarantees 90% accuracy with a confidence of at least 98%.

Example 2.7 (Universal concept class) Consider the set $\mathcal{X} = \{0,1\}^n$ of all Boolean vectors with n components, and let $\mathcal{U}_n$ be the concept class formed by all subsets of $\mathcal{X}$. Is this concept class PAC-learnable? To guarantee a consistent hypothesis the hypothesis class must include the concept class, thus $|\mathcal{H}| \geq |\mathcal{U}_n| = 2^{(2^n)}$. Theorem 2.5 gives the following sample complexity bound:

$$m \geq \frac{1}{\epsilon}\left((\log 2)2^n + \log\frac{1}{\delta}\right). \tag{2.11}$$

Here, the number of training samples required is exponential in n, which is the cost of the representation of a point in $\mathcal{X}$. Thus, PAC-learning is not guaranteed by the theorem. In fact, it is not hard to show that this universal concept class is not PAC-learnable.

Example 2.8 (k-term DNF formulae) A disjunctive normal form (DNF) formula is a formula written as the disjunction of several terms, each term being a conjunction of Boolean literals. A k-term DNF is a DNF formula defined by the disjunction of k terms, each term being a conjunction of at most n Boolean literals. Thus, for $k = 2$ and $n = 3$, an example of a k-term DNF is $(x_1 \wedge \overline{x}_2 \wedge x_3) \vee (\overline{x}_1 \wedge x_3)$.

Is the class $\mathcal{C}$ of k-term DNF formulae PAC-learnable? The cardinality of the class is 3^{nk}, since each term is a conjunction of at most n variables and there are 3^n such conjunctions, as seen previously. The hypothesis set $\mathcal{H}$ must contain $\mathcal{C}$ for

consistency to be possible, thus $|\mathcal{H}| \geq 3^{nk}$. Theorem 2.5 gives the following sample complexity bound:

$$m \geq \frac{1}{\epsilon}\Big((\log 3)nk + \log\frac{1}{\delta}\Big), \tag{2.12}$$

which is polynomial. However, it can be shown by a reduction from the graph 3-coloring problem that the problem of learning k-term DNF, even for $k = 3$, is not efficiently PAC-learnable, unless RP, the complexity class of problems that admit a randomized polynomial-time decision solution, coincides with NP(RP = NP), which is commonly conjectured not to be the case. Thus, while the sample size needed for learning k-term DNF formulae is only polynomial, efficient PAC-learning of this class is not possible if RP $\neq$ NP.

Example 2.9 (k-CNF formulae) A conjunctive normal form (CNF) formula is a conjunction of disjunctions. A k-CNF formula is an expression of the form $T_1 \wedge \ldots \wedge T_j$ with arbitrary length $j \in \mathbb{N}$ and with each term T_i being a disjunction of at most k Boolean attributes.

The problem of learning k-CNF formulae can be reduced to that of learning conjunctions of Boolean literals, which, as seen previously, is a PAC-learnable concept class. This can be done at the cost of introducing $(2n)^k$ new variables $Y_{u_1,\ldots,u_k}$ using the following bijection:

$$(u_1, \ldots, u_k) \to Y_{u_1,\ldots,u_k}, \tag{2.13}$$

where $u_1, \ldots, u_k$ are Boolean literals over the original variables $x_1, \ldots, x_n$. The value of $Y_{u_1,\ldots,u_k}$ is determined by $Y_{u_1,\ldots,u_k} = u_1 \vee \cdots \vee u_k$. Using this mapping, the original training sample can be transformed into one defined in terms of the new variables and any k-CNF formula over the original variables can be written as a conjunction over the variables $Y_{u_1,\ldots,u_k}$. This reduction to PAC-learning of conjunctions of Boolean literals can affect the original distribution of examples, but this is not an issue since in the PAC framework no assumption is made about the distribution. Thus, using this transformation, the PAC-learnability of conjunctions of Boolean literals implies that of k-CNF formulae.

This is a surprising result, however, since any k-term DNF formula can be written as a k-CNF formula. Indeed, using associativity, a k-term DNF $T_1 \vee \cdots \vee T_k$ with $T_i = u_{i,1} \wedge \cdots \wedge u_{i,n_i}$ for $i \in [k]$ can be rewritten as a k-CNF formula via

$$\bigvee_{i=1}^{k} u_{i,1} \wedge \cdots \wedge u_{i,n_i} = \bigwedge_{j_1 \in [n_1], \ldots, j_k \in [n_k]} u_{1,j_1} \vee \cdots \vee u_{k,j_k},$$

To illustrate this rewriting in a specific case, observe, for example, that

$$(u_1 \wedge u_2 \wedge u_3) \vee (v_1 \wedge v_2 \wedge v_3) = \bigwedge_{i,j=1}^{3} (u_i \vee v_j).$$

But, as we previously saw, k-term DNF formulae are not efficiently PAC-learnable if RP $\neq$ NP! What can explain this apparent inconsistency? The issue is that converting into a k-term DNF a k-CNF formula we have learned (which is equivalent to a k-term DNF) is in general intractable if RP $\neq$ NP.

This example reveals some key aspects of PAC-learning, which include the cost of the representation of a concept and the choice of the hypothesis set. For a fixed concept class, learning can be intractable or not depending on the choice of the representation.

2.3 Guarantees for finite hypothesis sets — inconsistent case

In the most general case, there may be no hypothesis in $\mathcal{H}$ consistent with the labeled training sample. This, in fact, is the typical case in practice, where the learning problems may be somewhat difficult or the concept classes more complex than the hypothesis set used by the learning algorithm. However, inconsistent hypotheses with a small number of errors on the training sample can be useful and, as we shall see, can benefit from favorable guarantees under some assumptions. This section presents learning guarantees precisely for this inconsistent case and finite hypothesis sets.

To derive learning guarantees in this more general setting, we will use Hoeffding's inequality (theorem D.2) or the following corollary, which relates the generalization error and empirical error of a single hypothesis.

Corollary 2.10 *Fix $\epsilon > 0$. Then, for any hypothesis $h\colon X \to \{0,1\}$, the following inequalities hold:*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\widehat{R}_S(h) - R(h) \geq \epsilon\right] \leq \exp(-2m\epsilon^2) \tag{2.14}$$

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\widehat{R}_S(h) - R(h) \leq -\epsilon\right] \leq \exp(-2m\epsilon^2). \tag{2.15}$$

By the union bound, this implies the following two-sided inequality:

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\left|\widehat{R}_S(h) - R(h)\right| \geq \epsilon\right] \leq 2\exp(-2m\epsilon^2). \tag{2.16}$$

Proof: The result follows immediately from theorem D.2. □

Setting the right-hand side of (2.16) to be equal to δ and solving for ϵ yields immediately the following bound for a single hypothesis.

Corollary 2.11 (Generalization bound — single hypothesis) *Fix a hypothesis $h\colon \mathcal{X} \to \{0,1\}$. Then, for any $\delta > 0$, the following inequality holds with probability at least $1-\delta$:*

$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{2.17}$$

The following example illustrates this corollary in a simple case.

Example 2.12 (Tossing a coin) Imagine tossing a biased coin that lands heads with probability p, and let our hypothesis be the one that always guesses tails. Then the true error rate is $R(h) = p$ and the empirical error rate $\widehat{R}_S(h) = \widehat{p}$, where $\widehat{p}$ is the empirical probability of heads based on the training sample drawn i.i.d. Thus, corollary 2.11 guarantees with probability at least $1 - \delta$ that

$$|p - \widehat{p}| \leq \sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \tag{2.18}$$

Therefore, if we choose $\delta = 0.02$ and use a sample of size 500, with probability at least 98%, the following approximation quality is guaranteed for $\widehat{p}$:

$$|p - \widehat{p}| \leq \sqrt{\frac{\log(10)}{1000}} \approx 0.048. \tag{2.19}$$

Can we readily apply corollary 2.11 to bound the generalization error of the hypothesis h_S returned by a learning algorithm when training on a sample S? No, since h_S is not a fixed hypothesis, but a random variable depending on the training sample S drawn. Note also that unlike the case of a fixed hypothesis for which the expectation of the empirical error is the generalization error (equation (2.3)), the generalization error $R(h_S)$ is a random variable and in general distinct from the expectation $\mathbb{E}[\widehat{R}_S(h_S)]$, which is a constant.

Thus, as in the proof for the consistent case, we need to derive a uniform convergence bound, that is a bound that holds with high probability for all hypotheses $h \in \mathcal{H}$.

Theorem 2.13 (Learning bound — finite $\mathcal{H}$, inconsistent case) *Let $\mathcal{H}$ be a finite hypothesis set. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following inequality holds:*

$$\forall h \in \mathcal{H}, \quad R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{2}{\delta}}{2m}}. \tag{2.20}$$

Proof: Let $h_1, \ldots, h_{|\mathcal{H}|}$ be the elements of $\mathcal{H}$. Using the union bound and applying corollary 2.11 to each hypothesis yield:

$$\begin{aligned}
&\mathbb{P}\left[\exists h \in \mathcal{H} \middle| \widehat{R}_S(h) - R(h)\middle| > \epsilon\right] \\
&\qquad = \mathbb{P}\left[\left(\middle|\widehat{R}_S(h_1) - R(h_1)\middle| > \epsilon\right) \vee \ldots \vee \left(\middle|\widehat{R}_S(h_{|\mathcal{H}|}) - R(h_{|\mathcal{H}|})\middle| > \epsilon\right)\right] \\
&\qquad \leq \sum_{h \in \mathcal{H}} \mathbb{P}\left[\middle|\widehat{R}_S(h) - R(h)\middle| > \epsilon\right] \\
&\qquad \leq 2|\mathcal{H}| \exp(-2m\epsilon^2).
\end{aligned}$$

Setting the right-hand side to be equal to δ completes the proof. □

Thus, for a finite hypothesis set $\mathcal{H}$,

$$R(h) \leq \widehat{R}_S(h) + O\left(\sqrt{\frac{\log_2 |\mathcal{H}|}{m}}\right).$$

As already pointed out, $\log_2 |\mathcal{H}|$ can be interpreted as the number of bits needed to represent $\mathcal{H}$. Several other remarks similar to those made on the generalization bound in the consistent case can be made here: a larger sample size m guarantees better generalization, and the bound increases with $|\mathcal{H}|$, but only logarithmically. But, here, the bound is a less favorable function of $\frac{\log_2 |\mathcal{H}|}{m}$; it varies as the square root of this term. This is not a minor price to pay: for a fixed $|\mathcal{H}|$, to attain the same guarantee as in the consistent case, a quadratically larger labeled sample is needed.

Note that the bound suggests seeking a trade-off between reducing the empirical error versus controlling the size of the hypothesis set: a larger hypothesis set is penalized by the second term but could help reduce the empirical error, that is the first term. But, for a similar empirical error, it suggests using a smaller hypothesis set. This can be viewed as an instance of the so-called *Occam's Razor principle* named after the theologian William of Occam: *Plurality should not be posited without necessity*, also rephrased as, *the simplest explanation is best*. In this context, it could be expressed as follows: All other things being equal, a simpler (smaller) hypothesis set is better.

2.4 Generalities

In this section we will discuss some general aspects of the learning scenario, which, for simplicity, we left out of the discussion of the earlier sections.

2.4.1 Deterministic versus stochastic scenarios

In the most general scenario of supervised learning, the distribution $\mathcal{D}$ is defined over $\mathcal{X} \times \mathcal{Y}$, and the training data is a labeled sample S drawn i.i.d. according to $\mathcal{D}$:

$$S = ((x_1, y_1), \ldots, (x_m, y_m)).$$

The learning problem is to find a hypothesis $h \in \mathcal{H}$ with small generalization error

$$R(h) = \mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[1_{h(x)\neq y}].$$

This more general scenario is referred to as the *stochastic scenario*. Within this setting, the output label is a probabilistic function of the input. The stochastic scenario captures many real-world problems where the label of an input point is not unique. For example, if we seek to predict gender based on input pairs formed by the height and weight of a person, then the label will typically not be unique.

For most pairs, both male and female are possible genders. For each fixed pair, there would be a probability distribution of the label being male.

The natural extension of the PAC-learning framework to this setting is known as the *agnostic PAC-learning*.

Definition 2.14 (Agnostic PAC-learning) *Let $\mathcal{H}$ be a hypothesis set. $\mathcal{A}$ is an* agnostic PAC-learning *algorithm if there exists a polynomial function $poly(\cdot,\cdot,\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the following holds for any sample size $m \geq poly(1/\epsilon, 1/\delta, n, size(c))$:*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[R(h_S) - \min_{h\in\mathcal{H}} R(h) \leq \epsilon] \geq 1 - \delta. \tag{2.21}$$

If $\mathcal{A}$ further runs in $poly(1/\epsilon, 1/\delta, n)$, then it is said to be an efficient agnostic PAC-learning algorithm.

When the label of a point can be uniquely determined by some measurable function $f\colon \mathcal{X} \to \mathcal{Y}$ (with probability one), then the scenario is said to be *deterministic*. In that case, it suffices to consider a distribution $\mathcal{D}$ over the input space. The training sample is obtained by drawing $(x_1, \ldots, x_m)$ according to $\mathcal{D}$ and the labels are obtained via f: $y_i = f(x_i)$ for all $i \in [m]$. Many learning problems can be formulated within this deterministic scenario.

In the previous sections, as well as in most of the material presented in this book, we have restricted our presentation to the deterministic scenario in the interest of simplicity. However, for all of this material, the extension to the stochastic scenario should be straightforward for the reader.

2.4.2 Bayes error and noise

In the deterministic case, by definition, there exists a target function f with no generalization error: $R(h) = 0$. In the stochastic case, there is a minimal non-zero error for any hypothesis.

Definition 2.15 (Bayes error) *Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the* Bayes error R^* *is defined as the infimum of the errors achieved by measurable functions $h\colon \mathcal{X} \to \mathcal{Y}$:*

$$R^\star = \inf_{\substack{h \\ h\ measurable}} R(h). \tag{2.22}$$

A hypothesis h with $R(h) = R^*$ is called a Bayes hypothesis *or* Bayes classifier.

By definition, in the deterministic case, we have $R^* = 0$, but, in the stochastic case, $R^* \neq 0$. Clearly, the Bayes classifier h_{Bayes} can be defined in terms of the conditional probabilities as:

$$\forall x \in \mathcal{X}, \quad h_{\text{Bayes}}(x) = \operatorname*{argmax}_{y\in\{0,1\}} \mathbb{P}[y|x]. \tag{2.23}$$

The average error made by h_{Bayes} on $x \in \mathcal{X}$ is thus $\min\{\mathbb{P}[0|x], \mathbb{P}[1|x]\}$, and this is the minimum possible error. This leads to the following definition of *noise*.

Definition 2.16 (Noise) *Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the* noise *at point $x \in \mathcal{X}$ is defined by*

$$\mathit{noise}(x) = \min\{\mathbb{P}[1|x], \mathbb{P}[0|x]\}. \tag{2.24}$$

The average noise *or the* noise *associated to $\mathcal{D}$ is $\mathbb{E}[\mathit{noise}(x)]$.*

Thus, the average noise is precisely the Bayes error: $\text{noise} = \mathbb{E}[\text{noise}(x)] = R^*$. The noise is a characteristic of the learning task indicative of its level of difficulty. A point $x \in \mathcal{X}$, for which $\text{noise}(x)$ is close to $1/2$, is sometimes referred to as *noisy* and is of course a challenge for accurate prediction.

2.5 Chapter notes

The PAC learning framework was introduced by Valiant [1984]. The book of Kearns and Vazirani [1994] is an excellent reference dealing with most aspects of PAC-learning and several other foundational questions in machine learning. Our example of learning axis-aligned rectangles, also discussed in that reference, is originally due to Blumer et al. [1989].

The PAC learning framework is a computational framework since it takes into account the cost of the computational representations and the time complexity of the learning algorithm. If we omit the computational aspects, it is similar to the learning framework considered earlier by Vapnik and Chervonenkis [see Vapnik, 2000]. The definition of noise presented in this chapter can be generalized to arbitrary loss functions (see exercise 2.14).

Occam's razor principle is invoked in a variety of contexts, such as in linguistics to justify the superiority of a set of rules or syntax. The Kolmogorov complexity can be viewed as the corresponding framework in information theory. In the context of the learning guarantees presented in this chapter, the principle suggests selecting the most parsimonious explanation (the hypothesis set with the smallest cardinality). We will see in the next sections other applications of this principle with different notions of simplicity or complexity.

2.6 Exercises

2.1 Two-oracle variant of the PAC model. Assume that positive and negative examples are now drawn from two separate distributions $\mathcal{D}_+$ and $\mathcal{D}_-$. For an accuracy $(1-\epsilon)$, the learning algorithm must find a hypothesis h such that:

$$\mathop{\mathbb{P}}_{x \sim \mathcal{D}_+}[h(x) = 0] \leq \epsilon \text{ and } \mathop{\mathbb{P}}_{x \sim \mathcal{D}_-}[h(x) = 1] \leq \epsilon\,. \tag{2.25}$$

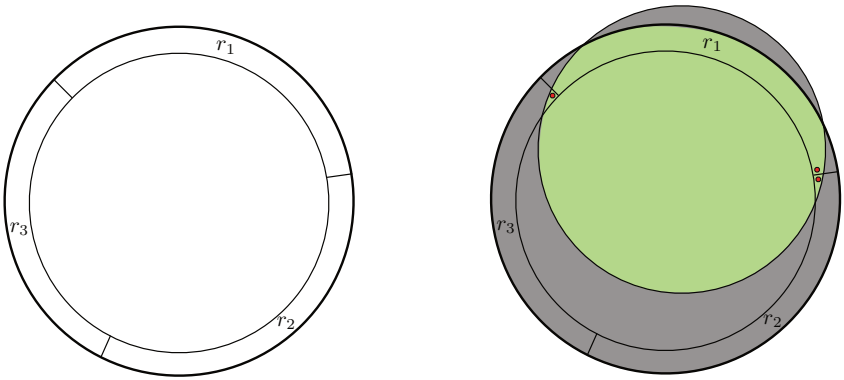


Figure 2.5
(a) Gertrude's regions r_1, r_2, r_3. (b) Hint for solution.

Thus, the hypothesis must have a small error on both distributions. Let $\mathcal{C}$ be any concept class and $\mathcal{H}$ be any hypothesis space. Let h_0 and h_1 represent the identically 0 and identically 1 functions, respectively. Prove that $\mathcal{C}$ is efficiently PAC-learnable using $\mathcal{H}$ in the standard (one-oracle) PAC model if and only if it is efficiently PAC-learnable using $\mathcal{H} \cup \{h_0, h_1\}$ in this two-oracle PAC model.

2.2 PAC learning of hyper-rectangles. An axis-aligned hyper-rectangle in $\mathbb{R}^n$ is a set of the form $[a_1, b_1] \times \ldots \times [a_n, b_n]$. Show that axis-aligned hyper-rectangles are PAC-learnable by extending the proof given in Example 2.4 for the case $n = 2$.

2.3 Concentric circles. Let $\mathcal{X} = \mathbb{R}^2$ and consider the set of concepts of the form $c = \{(x, y) \colon x^2 + y^2 \leq r^2\}$ for some real number r. Show that this class can be (ϵ, δ)-PAC-learned from training data of size $m \geq (1/\epsilon)\log(1/\delta)$.

2.4 Non-concentric circles. Let $\mathcal{X} = \mathbb{R}^2$ and consider the set of concepts of the form $c = \{x \in \mathbb{R}^2 \colon ||x - x_0|| \leq r\}$ for some point $x_0 \in \mathbb{R}^2$ and real number r. Gertrude, an aspiring machine learning researcher, attempts to show that this class of concepts may be (ϵ, δ)-PAC-learned with sample complexity $m \geq (3/\epsilon)\log(3/\delta)$, but she is having trouble with her proof. Her idea is that the learning algorithm would select the smallest circle consistent with the training data. She has drawn three regions r_1, r_2, r_3 around the edge of concept c, with each region having probability $\epsilon/3$ (see figure 2.5(a)). She wants to argue that if the generalization error is greater than or equal to ϵ, then one of these regions must have been missed by the training data, and hence this event will occur with probability at most δ. Can you tell Gertrude if her approach works? (*Hint*: You may wish to use figure 2.5(b) in your solution).

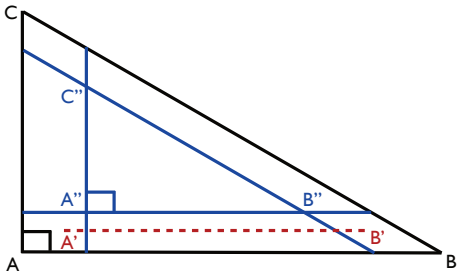


Figure 2.6
Axis-aligned right triangles.

2.5 Triangles. Let $\mathcal{X} = \mathbb{R}^2$ with orthonormal basis $(\mathbf{e}_1, \mathbf{e}_2)$, and consider the set of concepts defined by the area inside a right triangle ABC with two sides parallel to the axes, with $\overrightarrow{AB}/\|\overrightarrow{AB}\| = \mathbf{e}_1$ and $\overrightarrow{AC}/\|\overrightarrow{AC}\| = \mathbf{e}_2$, and $\|\overrightarrow{AB}\|/\|\overrightarrow{AC}\| = \alpha$ for some positive real $\alpha \in \mathbb{R}_+$. Show, using similar methods to those used in the chapter for the axis-aligned rectangles, that this class can be (ϵ, δ)-PAC-learned from training data of size $m \geq (3/\epsilon)\log(3/\delta)$. (*Hint*: You may consider using figure 2.6 in your solution).

2.6 Learning in the presence of noise — rectangles. In example 2.4, we showed that the concept class of axis-aligned rectangles is PAC-learnable. Consider now the case where the training points received by the learner are subject to the following noise: points negatively labeled are unaffected by noise but the label of a positive training point is randomly flipped to negative with probability $\eta \in (0, \frac{1}{2})$. The exact value of the noise rate η is not known to the learner but an upper bound η' is supplied to him with $\eta \leq \eta' < 1/2$. Show that the algorithm returning the tightest rectangle containing positive points can still PAC-learn axis-aligned rectangles in the presence of this noise. To do so, you can proceed using the following steps:

(a) Using the same notation as in example 2.4, assume that $\mathbb{P}[\mathsf{R}] > \epsilon$. Suppose that $R(\mathsf{R}') > \epsilon$. Give an upper bound on the probability that R' misses a region r_j, $j \in [4]$ in terms of ϵ and η'?

(b) Use that to give an upper bound on $\mathbb{P}[R(\mathsf{R}') > \epsilon]$ in terms of ϵ and η' and conclude by giving a sample complexity bound.

2.7 Learning in the presence of noise — general case. In this question, we will seek a result that is more general than in the previous question. We consider a finite hypothesis set $\mathcal{H}$, assume that the target concept is in $\mathcal{H}$, and adopt the following noise model: the label of a training point received by the learner is

randomly changed with probability $\eta \in (0, \frac{1}{2})$. The exact value of the noise rate η is not known to the learner but an upper bound η' is supplied to him with $\eta \leq \eta' < 1/2$.

(a) For any $h \in \mathcal{H}$, let $d(h)$ denote the probability that the label of a training point received by the learner disagrees with the one given by h. Let h^* be the target hypothesis, show that $d(h^*) = \eta$.

(b) More generally, show that for any $h \in \mathcal{H}$, $d(h) = \eta + (1 - 2\eta)\, R(h)$, where $R(h)$ denotes the generalization error of h.

(c) Fix $\epsilon > 0$ for this and all the following questions. Use the previous questions to show that if $R(h) > \epsilon$, then $d(h) - d(h^*) \geq \epsilon'$, where $\epsilon' = \epsilon(1 - 2\eta')$.

(d) For any hypothesis $h \in \mathcal{H}$ and sample S of size m, let $\widehat{d}(h)$ denote the fraction of the points in S whose labels disagree with those given by h. We will consider the algorithm L which, after receiving S, returns the hypothesis h_S with the smallest number of disagreements (thus $\widehat{d}(h_S)$ is minimal). To show PAC-learning for L, we will show that for any h, if $R(h) > \epsilon$, then with high probability $\widehat{d}(h) \geq \widehat{d}(h^*)$. First, show that for any $\delta > 0$, with probability at least $1 - \delta/2$, for $m \geq \frac{2}{\epsilon'^2} \log \frac{2}{\delta}$, the following holds:

$$\widehat{d}(h^*) - d(h^*) \leq \epsilon'/2$$

(e) Second, show that for any $\delta > 0$, with probability at least $1 - \delta/2$, for $m \geq \frac{2}{\epsilon'^2}(\log |\mathcal{H}| + \log \frac{2}{\delta})$, the following holds for all $h \in \mathcal{H}$:

$$d(h) - \widehat{d}(h) \leq \epsilon'/2$$

(f) Finally, show that for any $\delta > 0$, with probability at least $1 - \delta$, for $m \geq \frac{2}{\epsilon^2(1-2\eta')^2}(\log |\mathcal{H}| + \log \frac{2}{\delta})$, the following holds for all $h \in \mathcal{H}$ with $R(h) > \epsilon$:

$$\widehat{d}(h) - \widehat{d}(h^*) \geq 0.$$

(*Hint*: use $\widehat{d}(h) - \widehat{d}(h^*) = [\widehat{d}(h) - d(h)] + [d(h) - d(h^*)] + [d(h^*) - \widehat{d}(h^*)]$ and use previous questions to lower bound each of these three terms).

2.8 Learning intervals. Give a PAC-learning algorithm for the concept class $\mathcal{C}$ formed by closed intervals $[a, b]$ with $a, b \in \mathbb{R}$.

2.9 Learning union of intervals. Give a PAC-learning algorithm for the concept class $\mathcal{C}_2$ formed by unions of two closed intervals, that is $[a, b] \cup [c, d]$, with $a, b, c, d \in \mathbb{R}$. Extend your result to derive a PAC-learning algorithm for the concept class $\mathcal{C}_p$ formed by unions of $p \geq 1$ closed intervals, thus $[a_1, b_1] \cup \cdots \cup [a_p, b_p]$, with $a_k, b_k \in \mathbb{R}$ for $k \in [p]$. What are the time and sample complexities of your algorithm as a function of p?

2.10 Consistent hypotheses. In this chapter, we showed that for a finite hypothesis set $\mathcal{H}$, a consistent learning algorithm $\mathcal{A}$ is a PAC-learning algorithm. Here, we consider a converse question. Let $\mathcal{Z}$ be a finite set of m labeled points. Suppose that you are given a PAC-learning algorithm $\mathcal{A}$. Show that you can use $\mathcal{A}$ and a finite training sample S to find in polynomial time a hypothesis $h \in \mathcal{H}$ that is consistent with $\mathcal{Z}$, with high probability. (*Hint*: you can select an appropriate distribution $\mathcal{D}$ over $\mathcal{Z}$ and give a condition on $R(h)$ for h to be consistent.)

2.11 Senate laws. For important questions, President Mouth relies on expert advice. He selects an appropriate advisor from a collection of $\mathcal{H} = 2{,}800$ experts.

(a) Assume that laws are proposed in a random fashion independently and identically according to some distribution $\mathcal{D}$ determined by an unknown group of senators. Assume that President Mouth can find and select an expert senator out of $\mathcal{H}$ who has consistently voted with the majority for the last $m = 200$ laws. Give a bound on the probability that such a senator incorrectly predicts the global vote for a future law. What is the value of the bound with 95% confidence?

(b) Assume now that President Mouth can find and select an expert senator out of $\mathcal{H}$ who has consistently voted with the majority for all but $m' = 20$ of the last $m = 200$ laws. What is the value of the new bound?

2.12 Bayesian bound. Let $\mathcal{H}$ be a countable hypothesis set of functions mapping $\mathcal{X}$ to $\{0, 1\}$ and let p be a probability measure over $\mathcal{H}$. This probability measure represents the *prior probability* over the hypothesis class, i.e. the probability that a particular hypothesis is selected by the learning algorithm. Use Hoeffding's inequality to show that for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:

$$\forall h \in \mathcal{H}, R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log \frac{1}{p(h)} + \log \frac{1}{\delta}}{2m}}. \tag{2.26}$$

Compare this result with the bound given in the inconsistent case for finite hypothesis sets (*Hint*: you could use $\delta' = p(h)\delta$ as confidence parameter in Hoeffding's inequality).

2.13 Learning with an unknown parameter. In example 2.9, we showed that the concept class of k-CNF is PAC-learnable. Note, however, that the learning algorithm is given k as input. Is PAC-learning possible even when k is not provided? More generally, consider a family of concept classes $\{\mathcal{C}_s\}_s$ where $\mathcal{C}_s$ is the set of concepts in $\mathcal{C}$ with size at most s. Suppose we have a PAC-learning algorithm $\mathcal{A}$ that can be used for learning any concept class $\mathcal{C}_s$ when s is given.

Can we convert $\mathcal{A}$ into a PAC-learning algorithm $\mathcal{B}$ that does not require the knowledge of s? This is the main objective of this problem.

To do this, we first introduce a method for testing a hypothesis h, with high probability. Fix $\epsilon > 0$, $\delta > 0$, and $i \geq 1$ and define the sample size n by $n = \frac{32}{\epsilon}[i \log 2 + \log \frac{2}{\delta}]$. Suppose we draw an i.i.d. sample S of size n according to some unknown distribution $\mathcal{D}$. We will say that a hypothesis h is *accepted* if it makes at most $3/4\epsilon$ errors on S and that it is *rejected* otherwise. Thus, h is accepted iff $\widehat{R}(h) \leq 3/4\epsilon$.

(a) Assume that $R(h) \geq \epsilon$. Use the (multiplicative) Chernoff bound to show that in that case $\mathbb{P}_{S \sim \mathcal{D}^n}[h \text{ is accepted}] \leq \frac{\delta}{2^{i+1}}$.

(b) Assume that $R(h) \leq \epsilon/2$. Use the (multiplicative) Chernoff bounds to show that in that case $\mathbb{P}_{S \sim \mathcal{D}^n}[h \text{ is rejected}] \leq \frac{\delta}{2^{i+1}}$.

(c) Algorithm $\mathcal{B}$ is defined as follows: we start with $i = 1$ and, at each round $i \geq 1$, we guess the parameter size s to be $\widetilde{s} = \lfloor 2^{(i-1)/\log \frac{2}{\delta}} \rfloor$. We draw a sample S of size n (which depends on i) to test the hypothesis h_i returned by $\mathcal{A}$ when it is trained with a sample of size $S_{\mathcal{A}}(\epsilon/2, 1/2, \widetilde{s})$, that is the sample complexity of $\mathcal{A}$ for a required precision $\epsilon/2$, confidence $1/2$, and size $\widetilde{s}$ (we ignore the size of the representation of each example here). If h_i is accepted, the algorithm stops and returns h_i, otherwise it proceeds to the next iteration. Show that if at iteration i, the estimate $\widetilde{s}$ is larger than or equal to s, then $\mathbb{P}[h_i \text{ is accepted}] \geq 3/8$.

(d) Show that the probability that $\mathcal{B}$ does not halt after $j = \lceil \log \frac{2}{\delta} / \log \frac{8}{5} \rceil$ iterations with $\widetilde{s} \geq s$ is at most $\delta/2$.

(e) Show that for $i \geq \lceil 1 + (\log_2 s) \log \frac{2}{\delta} \rceil$, the inequality $\widetilde{s} \geq s$ holds.

(f) Show that with probability at least $1 - \delta$, algorithm $\mathcal{B}$ halts after at most $j' = \lceil 1 + (\log_2 s) \log \frac{2}{\delta} \rceil + j$ iterations and returns a hypothesis with error at most ϵ.

2.14 In this exercise, we generalize the notion of noise to the case of an arbitrary loss function $L \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$.

(a) Justify the following definition of the noise at point $x \in \mathcal{X}$:

$$\text{noise}(x) = \min_{y' \in \mathcal{Y}} \mathop{\mathbb{E}}_{y}[L(y, y')|x].$$

What is the value of noise(x) in a deterministic scenario? Does the definition match the one given in this chapter for binary classification?

(b) Show that the average noise coincides with the Bayes error (minimum loss achieved by a measurable function).

3 Rademacher Complexity and VC-Dimension

The hypothesis sets typically used in machine learning are infinite. But the sample complexity bounds of the previous chapter are uninformative when dealing with infinite hypothesis sets. One could ask whether efficient learning from a finite sample is even possible when the hypothesis set $\mathcal{H}$ is infinite. Our analysis of the family of axis-aligned rectangles (Example 2.4) indicates that this is indeed possible at least in some cases, since we proved that that infinite concept class was PAC-learnable. Our goal in this chapter will be to generalize that result and derive general learning guarantees for infinite hypothesis sets.

A general idea for doing so consists of reducing the infinite case to the analysis of finite sets of hypotheses and then proceed as in the previous chapter. There are different techniques for that reduction, each relying on a different notion of complexity for the family of hypotheses. The first complexity notion we will use is that of *Rademacher complexity*. This will help us derive learning guarantees using relatively simple proofs based on McDiarmid's inequality, while obtaining high-quality bounds, including data-dependent ones, which we will frequently make use of in future chapters. However, the computation of the empirical Rademacher complexity is NP-hard for some hypothesis sets. Thus, we subsequently introduce two other purely combinatorial notions, the *growth function* and the *VC-dimension*. We first relate the Rademacher complexity to the growth function and then bound the growth function in terms of the VC-dimension. The VC-dimension is often easier to bound or estimate. We will review a series of examples showing how to compute or bound it, then relate the growth function and the VC-dimensions. This leads to generalization bounds based on the VC-dimension. Finally, we present lower bounds based on the VC-dimension for two different settings: The *realizable* setting, where there is at least one hypothesis in the hypothesis set under consideration that achieves zero expected error, as well as the *non-realizable* setting, where no hypothesis in the set achieves zero expected error.

3.1 Rademacher complexity

We will continue to use $\mathcal{H}$ to denote a hypothesis set as in the previous chapters. Many of the results of this section are general and hold for an arbitrary loss function $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. In what follows, $\mathcal{G}$ will generally be interpreted as *the family of loss functions associated to* $\mathcal{H}$ mapping from $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$:

$$\mathcal{G} = \{g\colon (x, y) \mapsto L(h(x), y)\colon h \in \mathcal{H}\}.$$

However, the definitions are given in the general case of a family of functions $\mathcal{G}$ mapping from an arbitrary input space $\mathcal{Z}$ to $\mathbb{R}$.

The Rademacher complexity captures the richness of a family of functions by measuring the degree to which a hypothesis set can fit random noise. The following states the formal definitions of the empirical and average Rademacher complexity.

Definition 3.1 (Empirical Rademacher complexity) *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[a, b]$ and $S = (z_1, \ldots, z_m)$ a fixed sample of size m with elements in $\mathcal{Z}$. Then, the* empirical Rademacher complexity *of $\mathcal{G}$ with respect to the sample S is defined as:*

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)\right], \tag{3.1}$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)^\top$, with σ_is independent uniform random variables taking values in $\{-1, +1\}$.[3] *The random variables σ_i are called* Rademacher variables.

Let $\mathbf{g}_S$ denote the vector of values taken by function g over the sample S: $\mathbf{g}_S = (g(z_1), \ldots, g(z_m))^\top$. Then, the empirical Rademacher complexity can be rewritten as

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{g \in \mathcal{G}} \frac{\boldsymbol{\sigma} \cdot \mathbf{g}_S}{m}\right].$$

The inner product $\boldsymbol{\sigma} \cdot \mathbf{g}_S$ measures the correlation of $\mathbf{g}_S$ with the vector of random noise $\boldsymbol{\sigma}$. The supremum $\sup_{g \in \mathcal{G}} \frac{\boldsymbol{\sigma} \cdot \mathbf{g}_S}{m}$ is a measure of how well the function class $\mathcal{G}$ correlates with $\boldsymbol{\sigma}$ over the sample S. Thus, the empirical Rademacher complexity measures on average how well the function class $\mathcal{G}$ correlates with random noise on S. This describes the richness of the family $\mathcal{G}$: richer or more complex families $\mathcal{G}$ can generate more vectors $\mathbf{g}_S$ and thus better correlate with random noise, on average.

[3] We assume implicitly that the supremum over the family $\mathcal{G}$ in this definition is measurable and in general will adopt the same assumption throughout this book for other suprema over a class of functions. This assumption does not hold for arbitrary function classes but it is valid for the hypotheses sets typically considered in practice in machine learning, and the instances discussed in this book.

Definition 3.2 (Rademacher complexity) *Let $\mathcal{D}$ denote the distribution according to which samples are drawn. For any integer $m \geq 1$, the* Rademacher complexity *of $\mathcal{G}$ is the expectation of the empirical Rademacher complexity over all samples of size m drawn according to $\mathcal{D}$:*

$$\mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{\mathfrak{R}}_S(\mathcal{G})]. \tag{3.2}$$

We are now ready to present our first generalization bounds based on Rademacher complexity.

Theorem 3.3 *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[0,1]$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of an i.i.d. sample S of size m, each of the following holds for all $g \in \mathcal{G}$:*

$$\mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \tag{3.3}$$

$$\textit{and} \quad \mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}\,. \tag{3.4}$$

Proof: For any sample $S = (z_1, \ldots, z_m)$ and any $g \in \mathcal{G}$, we denote by $\widehat{\mathbb{E}}_S[g]$ the empirical average of g over S: $\widehat{\mathbb{E}}_S[g] = \frac{1}{m}\sum_{i=1}^{m} g(z_i)$. The proof consists of applying McDiarmid's inequality to function Φ defined for any sample S by

$$\Phi(S) = \sup_{g\in\mathcal{G}}\Big(\mathbb{E}[g] - \widehat{\mathbb{E}}_S[g]\Big). \tag{3.5}$$

Let S and S' be two samples differing by exactly one point, say z_m in S and z'_m in S'. Then, since the difference of suprema does not exceed the supremum of the difference, we have

$$\Phi(S') - \Phi(S) \leq \sup_{g\in\mathcal{G}}\Big(\widehat{\mathbb{E}}_S[g] - \widehat{\mathbb{E}}_{S'}[g]\Big) = \sup_{g\in\mathcal{G}}\frac{g(z_m) - g(z'_m)}{m} \leq \frac{1}{m}. \tag{3.6}$$

Similarly, we can obtain $\Phi(S) - \Phi(S') \leq 1/m$, thus $|\Phi(S) - \Phi(S')| \leq 1/m$. Then, by McDiarmid's inequality, for any $\delta > 0$, with probability at least $1 - \delta/2$, the following holds:

$$\Phi(S) \leq \mathop{\mathbb{E}}_{S}[\Phi(S)] + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{3.7}$$

We next bound the expectation of the right-hand side as follows:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S}[\Phi(S)] &= \mathop{\mathbb{E}}_{S}\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}[g]-\widehat{\mathbb{E}}_S(g)\right)\right] \\
&= \mathop{\mathbb{E}}_{S}\left[\sup_{g\in\mathcal{G}}\mathop{\mathbb{E}}_{S'}\left[\widehat{\mathbb{E}}_{S'}(g)-\widehat{\mathbb{E}}_S(g)\right]\right] && (3.8)\\
&\leq \mathop{\mathbb{E}}_{S,S'}\left[\sup_{g\in\mathcal{G}}\left(\widehat{\mathbb{E}}_{S'}(g)-\widehat{\mathbb{E}}_S(g)\right)\right] && (3.9)\\
&= \mathop{\mathbb{E}}_{S,S'}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}(g(z_i')-g(z_i))\right] && (3.10)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{\sigma},S,S'}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i(g(z_i')-g(z_i))\right] && (3.11)\\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma},S'}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i')\right]+\mathop{\mathbb{E}}_{\boldsymbol{\sigma},S}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}-\sigma_i g(z_i)\right] && (3.12)\\
&= 2\mathop{\mathbb{E}}_{\boldsymbol{\sigma},S}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i)\right]=2\mathfrak{R}_m(\mathcal{G}). && (3.13)
\end{aligned}$$

Equation (3.8) uses the fact that points in S' are sampled in an i.i.d. fashion and thus $\mathbb{E}[g]=\mathbb{E}_{S'}[\widehat{\mathbb{E}}_{S'}(g)]$, as in (2.3). Inequality 3.9 holds due to the sub-additivity of the supremum function.

In equation (3.11), we introduce Rademacher variables σ_i, which are uniformly distributed independent random variables taking values in $\{-1,+1\}$ as in definition 3.2. This does not change the expectation appearing in (3.10): when $\sigma_i=1$, the associated summand remains unchanged; when $\sigma_i=-1$, the associated summand flips signs, which is equivalent to swapping z_i and z_i' between S and S'. Since we are taking the expectation over all possible S and S', this swap does not affect the overall expectation; we are simply changing the order of the summands within the expectation.

Equation (3.12) holds by the sub-additivity of the supremum function, that is the inequality $\sup(U+V)\leq\sup(U)+\sup(V)$. Finally, (3.13) stems from the definition of Rademacher complexity and the fact that the variables σ_i and $-\sigma_i$ are distributed in the same way.

The reduction to $\mathfrak{R}_m(\mathcal{G})$ in equation (3.13) yields the bound in equation (3.3), using δ instead of $\delta/2$. To derive a bound in terms of $\widehat{\mathfrak{R}}_S(\mathcal{G})$, we observe that, by definition 3.1, changing one point in S changes $\widehat{\mathfrak{R}}_S(\mathcal{G})$ by at most $1/m$. Then, using again McDiarmid's inequality, with probability $1-\delta/2$ the following holds:

$$\mathfrak{R}_m(\mathcal{G})\leq\widehat{\mathfrak{R}}_S(\mathcal{G})+\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{3.14}$$

Finally, we use the union bound to combine inequalities 3.7 and 3.14, which yields with probability at least $1-\delta$:

$$\Phi(S) \leq 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}, \tag{3.15}$$

which matches (3.4). □

The following result relates the empirical Rademacher complexities of a hypothesis set $\mathcal{H}$ and to the family of loss functions $\mathcal{G}$ associated to $\mathcal{H}$ in the case of binary loss (zero-one loss).

Lemma 3.4 *Let $\mathcal{H}$ be a family of functions taking values in $\{-1,+1\}$ and let $\mathcal{G}$ be the family of loss functions associated to $\mathcal{H}$ for the zero-one loss: $\mathcal{G} = \{(x,y) \mapsto 1_{h(x)\neq y} \colon h \in \mathcal{H}\}$. For any sample $S = ((x_1,y_1),\ldots,(x_m,y_m))$ of elements in $\mathcal{X}\times\{-1,+1\}$, let $S_{\mathcal{X}}$ denote its projection over $\mathcal{X}$: $S_{\mathcal{X}} = (x_1,\ldots,x_m)$. Then, the following relation holds between the empirical Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$:*

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \frac{1}{2}\widehat{\mathfrak{R}}_{S_{\mathcal{X}}}(\mathcal{H}). \tag{3.16}$$

Proof: For any sample $S = ((x_1,y_1),\ldots,(x_m,y_m))$ of elements in $\mathcal{X}\times\{-1,+1\}$, by definition, the empirical Rademacher complexity of $\mathcal{G}$ can be written as:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\mathcal{G}) &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i 1_{h(x_i)\neq y_i}\Big] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \tfrac{1-y_i h(x_i)}{2}\Big] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} -\sigma_i y_i h(x_i)\Big] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)\Big] = \frac{1}{2}\widehat{\mathfrak{R}}_{S_{\mathcal{X}}}(\mathcal{H}),
\end{aligned}$$

where we used the fact that $1_{h(x_i)\neq y_i} = (1-y_i h(x_i))/2$ and the fact that for a fixed $y_i \in \{-1,+1\}$, σ_i and $-y_i\sigma_i$ are distributed in the same way. □

Note that the lemma implies, by taking expectations, that for any $m \geq 1$, $\mathfrak{R}_m(\mathcal{G}) = \frac{1}{2}\mathfrak{R}_m(\mathcal{H})$. These connections between the empirical and average Rademacher complexities can be used to derive generalization bounds for binary classification in terms of the Rademacher complexity of the hypothesis set $\mathcal{H}$.

Theorem 3.5 (Rademacher complexity bounds – binary classification) *Let $\mathcal{H}$ be a family of functions taking values in $\{-1,+1\}$ and let $\mathcal{D}$ be the distribution over the input space $\mathcal{X}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample S of*

size m drawn according to $\mathcal{D}$, each of the following holds for any $h \in \mathcal{H}$:

$$R(h) \leq \widehat{R}_S(h) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \tag{3.17}$$

$$\textit{and} \quad R(h) \leq \widehat{R}_S(h) + \widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \tag{3.18}$$

Proof: The result follows immediately by theorem 3.3 and lemma 3.4. □

The theorem provides two generalization bounds for binary classification based on the Rademacher complexity. Note that the second bound, (3.18), is data-dependent: the empirical Rademacher complexity $\widehat{\mathfrak{R}}_S(\mathcal{H})$ is a function of the specific sample S drawn. Thus, this bound could be particularly informative if we could compute $\widehat{\mathfrak{R}}_S(\mathcal{H})$. But, how can we compute the empirical Rademacher complexity? Using again the fact that σ_i and $-\sigma_i$ are distributed in the same way, we can write

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} -\sigma_i h(x_i)\Big] = -\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\inf_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)\Big].$$

Now, for a fixed value of $\boldsymbol{\sigma}$, computing $\inf_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)$ is equivalent to an *empirical risk minimization* problem, which is known to be computationally hard for some hypothesis sets. Thus, in some cases, computing $\widehat{\mathfrak{R}}_S(\mathcal{H})$ could be computationally hard. In the next sections, we will relate the Rademacher complexity to combinatorial measures that are easier to compute and also of independent interest for their usefulness in the analysis of learning in many contexts.

3.2 Growth function

Here we will show how the Rademacher complexity can be bounded in terms of the *growth function*.

Definition 3.6 (Growth function) *The* growth function $\Pi_{\mathcal{H}}\colon \mathbb{N} \to \mathbb{N}$ *for a hypothesis set $\mathcal{H}$ is defined by:*

$$\forall m \in \mathbb{N},\ \Pi_{\mathcal{H}}(m) = \max_{\{x_1,\ldots,x_m\}\subseteq X} \Big|\big\{\big(h(x_1),\ldots,h(x_m)\big)\colon h \in \mathcal{H}\big\}\Big|. \tag{3.19}$$

In other words, $\Pi_{\mathcal{H}}(m)$ is the maximum number of distinct ways in which m points can be classified using hypotheses in $\mathcal{H}$. Each one of these distinct classifications is called a *dichotomy* and, thus, the growth function counts the number of dichotomies that are realized by the hypothesis. This provides another measure of the richness of the hypothesis set $\mathcal{H}$. However, unlike the Rademacher complexity, this measure does not depend on the distribution, it is purely combinatorial.

To relate the Rademacher complexity to the growth function, we will use Massart's lemma.

Theorem 3.7 (Massart's lemma) *Let $\mathcal{A} \subseteq \mathbb{R}^m$ be a finite set, with $r = \max_{\mathbf{x}\in\mathcal{A}} \|\mathbf{x}\|_2$, then the following holds:*

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\frac{1}{m}\sup_{\mathbf{x}\in\mathcal{A}}\sum_{i=1}^{m}\sigma_i x_i\right] \leq \frac{r\sqrt{2\log|\mathcal{A}|}}{m}\,, \tag{3.20}$$

where σ_is are independent uniform random variables taking values in $\{-1,+1\}$ and $x_1,\ldots,x_m$ are the components of vector $\mathbf{x}$.

Proof: The result follows immediately from the bound on the expectation of a maximum given by Corollary D.11 since the random variables $\sigma_i x_i$ are independent and each $\sigma_i x_i$ takes values in $[-|x_i|,|x_i|]$ with $\sqrt{\sum_{i=1}^m x_i^2} \leq r^2$. □

Using this result, we can now bound the Rademacher complexity in terms of the growth function.

Corollary 3.8 *Let $\mathcal{G}$ be a family of functions taking values in $\{-1,+1\}$. Then the following holds:*

$$\mathfrak{R}_m(\mathcal{G}) \leq \sqrt{\frac{2\log\Pi_{\mathcal{G}}(m)}{m}}. \tag{3.21}$$

Proof: For a fixed sample $S = (x_1,\ldots,x_m)$, we denote by $\mathcal{G}_{|S}$ the set of vectors of function values $(g(x_1),\ldots,g(x_m))^\top$ where g is in $\mathcal{G}$. Since $g \in \mathcal{G}$ takes values in $\{-1,+1\}$, the norm of these vectors is bounded by $\sqrt{m}$. We can then apply Massart's lemma as follows:

$$\mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S}\left[\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{u\in\mathcal{G}_{|S}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i u_i\right]\right] \leq \mathop{\mathbb{E}}_{S}\left[\frac{\sqrt{m}\sqrt{2\log|\mathcal{G}_{|S}|}}{m}\right].$$

By definition, $|\mathcal{G}_{|S}|$ is bounded by the growth function, thus,

$$\mathfrak{R}_m(\mathcal{G}) \leq \mathop{\mathbb{E}}_{S}\left[\frac{\sqrt{m}\sqrt{2\log\Pi_{\mathcal{G}}(m)}}{m}\right] = \sqrt{\frac{2\log\Pi_{\mathcal{G}}(m)}{m}},$$

which concludes the proof. □

Combining the generalization bound (3.17) of theorem 3.5 with corollary 3.8 yields immediately the following generalization bound in terms of the growth function.

Corollary 3.9 (Growth function generalization bound) *Let $\mathcal{H}$ be a family of functions taking values in $\{-1,+1\}$. Then, for any $\delta > 0$, with probability at least $1-\delta$, for any $h \in \mathcal{H}$,*

$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{2\log\Pi_{\mathcal{H}}(m)}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{3.22}$$

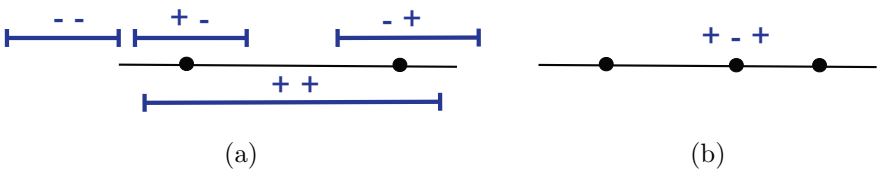


Figure 3.1
VC-dimension of intervals on the real line. (a) Any two points can be shattered. (b) No sample of three points can be shattered as the $(+,-,+)$ labeling cannot be realized.

Growth function bounds can be also derived directly (without using Rademacher complexity bounds first). The resulting bound is then the following:

$$\mathbb{P}\left[\left|R(h)-\widehat{R}_S(h)\right|>\epsilon\right]\leq 4\Pi_{\mathcal{H}}(2m)\exp\left(-\frac{m\epsilon^2}{8}\right), \tag{3.23}$$

which only differs from (3.22) by constants.

The computation of the growth function may not be always convenient since, by definition, it requires computing $\Pi_{\mathcal{H}}(m)$ for all $m \geq 1$. The next section introduces an alternative measure of the complexity of a hypothesis set $\mathcal{H}$ that is based instead on a single scalar, which will turn out to be in fact deeply related to the behavior of the growth function.

3.3 VC-dimension

Here, we introduce the notion of *VC-dimension* (Vapnik-Chervonenkis dimension). The VC-dimension is also a purely combinatorial notion but it is often easier to compute than the growth function (or the Rademacher Complexity). As we shall see, the VC-dimension is a key quantity in learning and is directly related to the growth function.

To define the VC-dimension of a hypothesis set $\mathcal{H}$, we first introduce the concept of *shattering*. Recall from the previous section, that given a hypothesis set $\mathcal{H}$, a dichotomy of a set S is one of the possible ways of labeling the points of S using a hypothesis in $\mathcal{H}$. A set S of $m \geq 1$ points is said to be shattered by a hypothesis set $\mathcal{H}$ when $\mathcal{H}$ realizes all possible dichotomies of S, that is when $\Pi_{\mathcal{H}}(m) = 2^m$.

Definition 3.10 (VC-dimension) *The VC-dimension of a hypothesis set $\mathcal{H}$ is the size of the largest set that can be shattered by $\mathcal{H}$:*

$$\mathrm{VCdim}(\mathcal{H}) = \max\{m\colon \Pi_{\mathcal{H}}(m) = 2^m\}. \tag{3.24}$$

Note that, by definition, if $\mathrm{VCdim}(\mathcal{H}) = d$, there exists a set of size d that can be shattered. However, this does not imply that all sets of size d or less are shattered and, in fact, this is typically not the case.

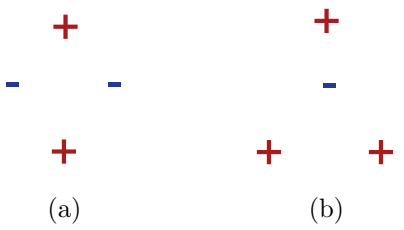


Figure 3.2
Unrealizable dichotomies for four points using hyperplanes in $\mathbb{R}^2$. (a) All four points lie on the convex hull. (b) Three points lie on the convex hull while the remaining point is interior.

To further illustrate this notion, we will examine a series of examples of hypothesis sets and will determine the VC-dimension in each case. To compute the VC-dimension we will typically show a lower bound for its value and then a matching upper bound. To give a lower bound d for $\text{VCdim}(\mathcal{H})$, it suffices to show that a set S of cardinality d can be shattered by $\mathcal{H}$. To give an upper bound, we need to prove that no set S of cardinality $d+1$ can be shattered by $\mathcal{H}$, which is typically more difficult.

Example 3.11 (Intervals on the real line) Our first example involves the hypothesis class of intervals on the real line. It is clear that the VC-dimension is at least two, since all four dichotomies $(+,+),(-,-),(+,-),(-,+)$ can be realized, as illustrated in figure 3.1(a). In contrast, by the definition of intervals, no set of three points can be shattered since the $(+,-,+)$ labeling cannot be realized. Hence, $\text{VCdim}(\text{intervals in } \mathbb{R}) = 2$.

Example 3.12 (Hyperplanes) Consider the set of hyperplanes in $\mathbb{R}^2$. We first observe that any three non-collinear points in $\mathbb{R}^2$ can be shattered. To obtain the first three dichotomies, we choose a hyperplane that has two points on one side and the third point on the opposite side. To obtain the fourth dichotomy we have all three points on the same side of the hyperplane. The remaining four dichotomies are realized by simply switching signs. Next, we show that four points cannot be shattered by considering two cases: (i) the four points lie on the convex hull defined by the four points, and (ii) three of the four points lie on the convex hull and the remaining point is internal. In the first case, a positive labeling for one diagonal pair and a negative labeling for the other diagonal pair cannot be realized, as illustrated in figure 3.2(a). In the second case, a labeling which is positive for the points on the convex hull and negative for the interior point cannot be realized, as illustrated in figure 3.2(b). Hence, $\text{VCdim}(\text{hyperplanes in } \mathbb{R}^2) = 3$.

More generally in $\mathbb{R}^d$, we derive a lower bound by starting with a set of $d+1$ points in $\mathbb{R}^d$, setting $\mathbf{x}_0$ to be the origin and defining $\mathbf{x}_i$, for $i \in \{1,\ldots,d\}$, as the point whose ith coordinate is 1 and all others are 0. Let $y_0, y_1, \ldots, y_d \in \{-1,+1\}$ be an

arbitrary set of labels for $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_d$. Let $\mathbf{w}$ be the vector whose ith coordinate is y_i. Then the classifier defined by the hyperplane of equation $\mathbf{w} \cdot \mathbf{x} + \frac{y_0}{2} = 0$ shatters $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_d$ since for any $i \in \{0, \ldots, d\}$,

$$\operatorname{sgn}\left(\mathbf{w} \cdot \mathbf{x}_i + \frac{y_0}{2}\right) = \operatorname{sgn}\left(y_i + \frac{y_0}{2}\right) = y_i. \tag{3.25}$$

To obtain an upper bound, it suffices to show that no set of $d+2$ points can be shattered by halfspaces. To prove this, we will use the following general theorem.

Theorem 3.13 (Radon's theorem) Any set $\mathcal{X}$ of $d+2$ points in $\mathbb{R}^d$ can be partitioned into two subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ such that the convex hulls of $\mathcal{X}_1$ and $\mathcal{X}_2$ intersect.

Proof: Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{d+2}\} \subset \mathbb{R}^d$. The following is a system of $d+1$ linear equations in $\alpha_1, \ldots, \alpha_{d+2}$:

$$\sum_{i=1}^{d+2} \alpha_i \mathbf{x}_i = 0 \qquad \text{and} \qquad \sum_{i=1}^{d+2} \alpha_i = 0, \tag{3.26}$$

since the first equality leads to d equations, one for each component. The number of unknowns, $d+2$, is larger than the number of equations, $d+1$, therefore the system admits a non-zero solution $\beta_1, \ldots, \beta_{d+2}$. Since $\sum_{i=1}^{d+2} \beta_i = 0$, both $\mathcal{I}_1 = \{i \in [d+2] \colon \beta_i > 0\}$ and $\mathcal{I}_2 = \{i \in [d+2] \colon \beta_i \leq 0\}$ are non-empty sets and $\mathcal{X}_1 = \{\mathbf{x}_i \colon i \in \mathcal{I}_1\}$ and $\mathcal{X}_2 = \{\mathbf{x}_i \colon i \in \mathcal{I}_2\}$ form a partition of $\mathcal{X}$. By the last equation of (3.26), $\sum_{i \in \mathcal{I}_1} \beta_i = -\sum_{i \in \mathcal{I}_2} \beta_i$. Let $\beta = \sum_{i \in \mathcal{I}_1} \beta_i$. Then, the first part of (3.26) implies

$$\sum_{i \in \mathcal{I}_1} \frac{\beta_i}{\beta} \mathbf{x}_i = \sum_{i \in \mathcal{I}_2} \frac{-\beta_i}{\beta} \mathbf{x}_i,$$

with $\sum_{i \in \mathcal{I}_1} \frac{\beta_i}{\beta} = \sum_{i \in \mathcal{I}_2} \frac{-\beta_i}{\beta} = 1$, and $\frac{\beta_i}{\beta} \geq 0$ for $i \in \mathcal{I}_1$ and $\frac{-\beta_i}{\beta} \geq 0$ for $i \in \mathcal{I}_2$. By definition of the convex hulls (B.6), this implies that $\sum_{i \in \mathcal{I}_1} \frac{\beta_i}{\beta} \mathbf{x}_i$ belongs both to the convex hull of $\mathcal{X}_1$ and to that of $\mathcal{X}_2$. □

Now, let $\mathcal{X}$ be a set of $d+2$ points. By Radon's theorem, it can be partitioned into two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ such that their convex hulls intersect. Observe that when two sets of points $\mathcal{X}_1$ and $\mathcal{X}_2$ are separated by a hyperplane, their convex hulls are also separated by that hyperplane. Thus, $\mathcal{X}_1$ and $\mathcal{X}_2$ cannot be separated by a hyperplane and $\mathcal{X}$ is not shattered. Combining our lower and upper bounds, we have proven that $\operatorname{VCdim}(\text{hyperplanes in } \mathbb{R}^d) = d+1$.

Example 3.14 (Axis-aligned Rectangles) We first show that the VC-dimension is at least four, by considering four points in a diamond pattern. Then, it is clear that all 16 dichotomies can be realized, some of which are illustrated in figure 3.3(a). In contrast, for any set of five distinct points, if we construct the minimal axis-aligned rectangle containing these points, one of the five points is in the interior of

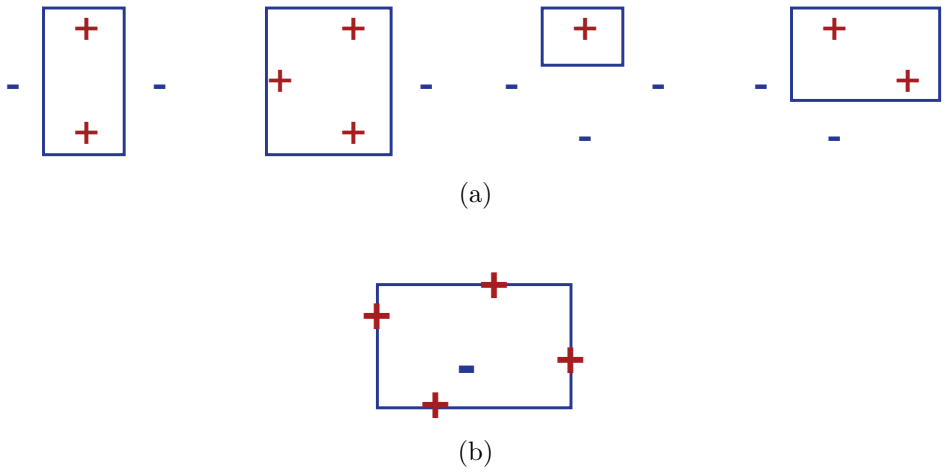


Figure 3.3
VC-dimension of axis-aligned rectangles. (a) Examples of realizable dichotomies for four points in a diamond pattern. (b) No sample of five points can be realized if the interior point and the remaining points have opposite labels.

this rectangle. Imagine that we assign a negative label to this interior point and a positive label to each of the remaining four points, as illustrated in figure 3.3(b). There is no axis-aligned rectangle that can realize this labeling. Hence, no set of five distinct points can be shattered and VCdim(axis-aligned rectangles) = 4.

Example 3.15 (Convex Polygons) We focus on the class of convex d-gons in the plane. To get a lower bound, we show that any set of $2d+1$ points can be shattered. To do this, we select $2d+1$ points that lie on a circle, and for a particular labeling, if there are more negative than positive labels, then the points with the positive labels are used as the polygon's vertices, as in figure 3.4(a). Otherwise, the tangents of the negative points serve as the edges of the polygon, as shown in (3.4)(b). To derive an upper bound, it can be shown that choosing points on the circle maximizes the number of possible dichotomies, and thus VCdim(convex d-gons) $= 2d + 1$. Note also that VCdim(convex polygons) $= +\infty$.

Example 3.16 (Sine Functions) The previous examples could suggest that the VC-dimension of $\mathcal{H}$ coincides with the number of free parameters defining $\mathcal{H}$. For example, the number of parameters defining hyperplanes matches their VC-dimension. However, this does not hold in general. Several of the exercises in this chapter illustrate this fact. The following provides a striking example from this point of view. Consider the following family of sine functions: $\{t \mapsto \sin(\omega t) : \omega \in \mathbb{R}\}$. One instance of this function class is shown in figure 3.5. These sine functions can be

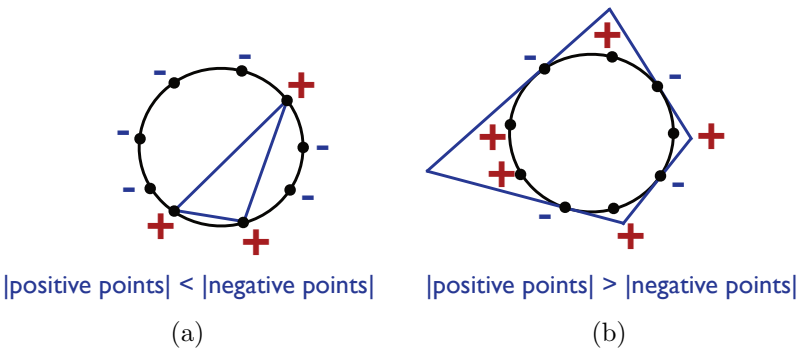


Figure 3.4
Convex d-gons in the plane can shatter $2d+1$ points. (a) d-gon construction when there are more negative labels. (b) d-gon construction when there are more positive labels.

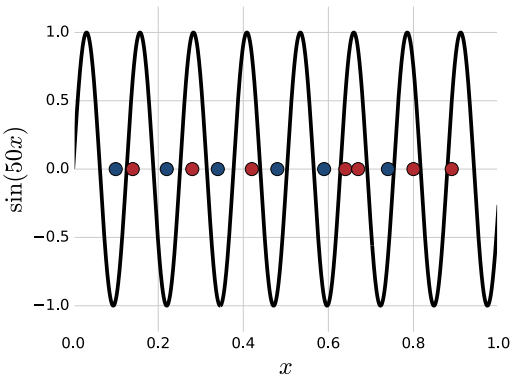


Figure 3.5
An example of a sine function (with $\omega = 50$) used for classification.

used to classify the points on the real line: a point is labeled positively if it is above the curve, negatively otherwise. Although this family of sine functions is defined via a single parameter, ω, it can be shown that VCdim(sine functions) $= +\infty$ (exercise 3.20).

The VC-dimension of many other hypothesis sets can be determined or upper-bounded in a similar way (see this chapter's exercises). In particular, the VC-dimension of any vector space of dimension $r < \infty$ can be shown to be at most r (exercise 3.19). The next result, known as *Sauer's lemma*, clarifies the connection between the notions of growth function and VC-dimension.

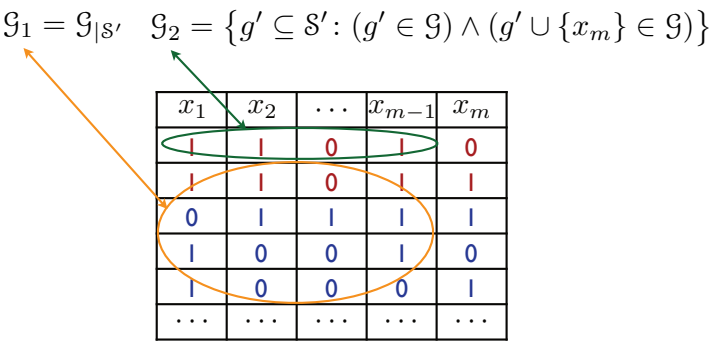


Figure 3.6
Illustration of how $\mathcal{G}_1$ and $\mathcal{G}_2$ are constructed in the proof of Sauer's lemma.

Theorem 3.17 (Sauer's lemma) *Let $\mathcal{H}$ be a hypothesis set with* $\mathrm{VCdim}(\mathcal{H}) = d$. *Then, for all $m \in \mathbb{N}$, the following inequality holds:*

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}. \tag{3.27}$$

Proof: The proof is by induction on $m + d$. The statement clearly holds for $m = 1$ and $d = 0$ or $d = 1$. Now, assume that it holds for $(m-1, d-1)$ and $(m-1, d)$. Fix a set $\mathcal{S} = \{x_1, \ldots, x_m\}$ with $\Pi_{\mathcal{H}}(m)$ dichotomies and let $\mathcal{G} = \mathcal{H}_{|\mathcal{S}}$ be the set of concepts $\mathcal{H}$ induced by restriction to $\mathcal{S}$.

Now consider the following families over $\mathcal{S}' = \{x_1, \ldots, x_{m-1}\}$. We define $\mathcal{G}_1 = \mathcal{G}_{|\mathcal{S}'}$ as the set of concepts $\mathcal{H}$ induced by restriction to S'. Next, by identifying each concept as the set of points (in $\mathcal{S}'$ or $\mathcal{S}$) for which it is non-zero, we can define $_2$ as

$$\mathcal{G}_2 = \{g' \subseteq \mathcal{S}' : (g' \in \mathcal{G}) \wedge (g' \cup \{x_m\} \in \mathcal{G})\}.$$

Since $g' \subseteq \mathcal{S}'$, $g' \in \mathcal{G}$ means that without adding x_m it is a concept of $\mathcal{G}$. Further, the constraint $g' \cup \{x_m\} \in \mathcal{G}$ means that adding x_m to g' also makes it a concept of $\mathcal{G}$. The construction of $\mathcal{G}_1$ and $\mathcal{G}_2$ is illustrated pictorially in figure 3.6. Given our definitions of $\mathcal{G}_1$ and $\mathcal{G}_2$, observe that $|\mathcal{G}_1| + |\mathcal{G}_2| = |\mathcal{G}|$.

Since $\mathrm{VCdim}(\mathcal{G}_1) \leq \mathrm{VCdim}(\mathcal{G}) \leq d$, then by definition of the growth function and using the induction hypothesis,

$$|\mathcal{G}_1| \leq \Pi_{\mathcal{G}_1}(m-1) \leq \sum_{i=0}^{d} \binom{m-1}{i}.$$

Further, by definition of $\mathcal{G}_2$, if a set $\mathcal{Z} \subseteq \mathcal{S}'$ is shattered by $\mathcal{G}_2$, then the set $\mathcal{Z} \cup \{x_m\}$ is shattered by $\mathcal{G}$. Hence,

$$\mathrm{VCdim}(\mathcal{G}_2) \leq \mathrm{VCdim}(\mathcal{G}) - 1 = d - 1,$$

and by definition of the growth function and using the induction hypothesis,

$$|\mathcal{G}_2| \leq \Pi_{\mathcal{G}_2}(m-1) \leq \sum_{i=0}^{d-1} \binom{m-1}{i}.$$

Thus,

$$|\mathcal{G}| = |\mathcal{G}_1| + |\mathcal{G}_2| \leq \sum_{i=0}^{d} \tbinom{m-1}{i} + \sum_{i=0}^{d-1} \tbinom{m-1}{i} = \sum_{i=0}^{d} \tbinom{m-1}{i} + \tbinom{m-1}{i-1} = \sum_{i=0}^{d} \tbinom{m}{i},$$

which completes the inductive proof. □

The significance of Sauer's lemma can be seen by corollary 3.18, which remarkably shows that growth function only exhibits two types of behavior: either $\mathrm{VCdim}(\mathcal{H}) = d < +\infty$, in which case $\Pi_{\mathcal{H}}(m) = O(m^d)$, or $\mathrm{VCdim}(\mathcal{H}) = +\infty$, in which case $\Pi_{\mathcal{H}}(m) = 2^m$.

Corollary 3.18 *Let $\mathcal{H}$ be a hypothesis set with $\mathrm{VCdim}(\mathcal{H}) = d$. Then for all $m \geq d$,*

$$\Pi_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d = O(m^d). \tag{3.28}$$

Proof: The proof begins by using Sauer's lemma. The first inequality multiplies each summand by a factor that is greater than or equal to one since $m \geq d$, while the second inequality adds non-negative summands to the summation.

$$\begin{aligned}
\Pi_{\mathcal{H}}(m) &\leq \sum_{i=0}^{d} \binom{m}{i} \\
&\leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&\leq \sum_{i=0}^{m} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&= \left(\frac{m}{d}\right)^d \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^i \\
&= \left(\frac{m}{d}\right)^d \left(1 + \frac{d}{m}\right)^m \leq \left(\frac{m}{d}\right)^d e^d.
\end{aligned}$$

After simplifying the expression using the binomial theorem, the final inequality follows using the general inequality $(1-x) \leq e^{-x}$. □

The explicit relationship just formulated between VC-dimension and the growth function combined with corollary 3.9 leads immediately to the following generalization bounds based on the VC-dimension.

Corollary 3.19 (VC-dimension generalization bounds) *Let $\mathcal{H}$ be a family of functions taking values in $\{-1, +1\}$ with VC-dimension d. Then, for any $\delta > 0$, with proba-*

bility at least $1-\delta$*, the following holds for all* $h \in \mathcal{H}$*:*

$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}. \tag{3.29}$$

Thus, the form of this generalization bound is

$$R(h) \leq \widehat{R}_S(h) + O\left(\sqrt{\frac{\log(m/d)}{(m/d)}}\right), \tag{3.30}$$

which emphasizes the importance of the ratio m/d for generalization. The theorem provides another instance of Occam's razor principle where simplicity is measured in terms of smaller VC-dimension.

VC-dimension bounds can be derived directly without using an intermediate Rademacher complexity bound, as for (3.23): combining Sauer's lemma with (3.23) leads to the following high-probability bound

$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{8d \log \frac{2em}{d} + 8 \log \frac{4}{\delta}}{m}},$$

which has the general form of (3.30). The log factor plays only a minor role in these bounds. A finer analysis can be used in fact to eliminate that factor.

3.4 Lower bounds

In the previous section, we presented several upper bounds on the generalization error. In contrast, this section provides lower bounds on the generalization error of any learning algorithm in terms of the VC-dimension of the hypothesis set used.

These lower bounds are shown by finding for any algorithm a 'bad' distribution. Since the learning algorithm is arbitrary, it will be difficult to specify that particular distribution. Instead, it suffices to prove its existence non-constructively. At a high level, the proof technique used to achieve this is the *probabilistic method* of Paul Erdös. In the context of the following proofs, first a lower bound is given on the expected error over the parameters defining the distributions. From that, the lower bound is shown to hold for at least one set of parameters, that is one distribution.

Theorem 3.20 (Lower bound, realizable case) *Let* $\mathcal{H}$ *be a hypothesis set with VC-dimension* $d > 1$*. Then, for any* $m \geq 1$ *and any learning algorithm* $\mathcal{A}$*, there exist a distribution* $\mathcal{D}$ *over* $\mathcal{X}$ *and a target function* $f \in \mathcal{H}$ *such that*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left[R_{\mathcal{D}}(h_S, f) > \frac{d-1}{32m}\right] \geq 1/100. \tag{3.31}$$

Proof: Let $\overline{\mathcal{X}} = \{x_0, x_1, \ldots, x_{d-1}\} \subseteq \mathcal{X}$ be a set that is shattered by $\mathcal{H}$. For any $\epsilon > 0$, we choose $\mathcal{D}$ such that its support is reduced to $\overline{\mathcal{X}}$ and so that one point (x_0)

has very high probability $(1-8\epsilon)$, with the rest of the probability mass distributed uniformly among the other points:

$$\mathop{\mathbb{P}}_{\mathcal{D}}[x_0] = 1-8\epsilon \quad \text{and} \quad \forall i \in [d-1], \mathop{\mathbb{P}}_{\mathcal{D}}[x_i] = \frac{8\epsilon}{d-1}. \tag{3.32}$$

With this definition, most samples would contain x_0 and, since $\mathcal{X}$ is shattered, $\mathcal{A}$ can essentially do no better than tossing a coin when determining the label of a point x_i not falling in the training set.

We assume without loss of generality that $\mathcal{A}$ makes no error on x_0. For a sample S, we let $\overline{S}$ denote the set of its elements falling in $\{x_1, \ldots, x_{d-1}\}$, and let $\mathcal{S}$ be the set of samples S of size m such that $|\overline{S}| \leq (d-1)/2$. Now, fix a sample $S \in \mathcal{S}$, and consider the uniform distribution $\mathcal{U}$ over all labelings $f\colon \overline{\mathcal{X}} \to \{0,1\}$, which are all in $\mathcal{H}$ since the set is shattered. Then, the following lower bound holds:

$$\begin{aligned}
\mathop{\mathbb{E}}_{f\sim\mathcal{U}}[R_{\mathcal{D}}(h_S, f)] &= \sum_f \sum_{x\in\overline{\mathcal{X}}} 1_{h_S(x)\neq f(x)}\, \mathbb{P}[x]\, \mathbb{P}[f] \\
&\geq \sum_f \sum_{x\notin\overline{S}} 1_{h_S(x)\neq f(x)}\, \mathbb{P}[x]\, \mathbb{P}[f] \\
&= \sum_{x\notin\overline{S}} \Big(\sum_f 1_{h_S(x)\neq f(x)}\, \mathbb{P}[f] \Big)\, \mathbb{P}[x] \\
&= \frac{1}{2} \sum_{x\notin\overline{S}} \mathbb{P}[x] \geq \frac{1}{2}\frac{d-1}{2}\frac{8\epsilon}{d-1} = 2\epsilon.
\end{aligned} \tag{3.33}$$

The first lower bound holds because we remove non-negative terms from the summation when we only consider $x \notin \overline{S}$ instead of all x in $\overline{\mathcal{X}}$. After rearranging terms, the subsequent equality holds since we are taking an expectation over $f \in \mathcal{H}$ with uniform weight on each f and $\mathcal{H}$ shatters $\overline{\mathcal{X}}$. The final lower bound holds due to the definitions of $\mathcal{D}$ and $\overline{S}$, the latter which implies that $|\overline{\mathcal{X}} - \overline{S}| \geq (d-1)/2$.

Since (3.33) holds for all $S \in \mathcal{S}$, it also holds in expectation over all $S \in \mathcal{S}$: $\mathbb{E}_{S\in\mathcal{S}}\big[\mathbb{E}_{f\sim\mathcal{U}}[R_{\mathcal{D}}(h_S, f)]\big] \geq 2\epsilon$. By Fubini's theorem, the expectations can be permuted, thus,

$$\mathop{\mathbb{E}}_{f\sim\mathcal{U}}\Big[\mathop{\mathbb{E}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f)]\Big] \geq 2\epsilon. \tag{3.34}$$

This implies that $\mathbb{E}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0)] \geq 2\epsilon$ for at least one labeling $f_0 \in \mathcal{H}$. Decomposing this expectation into two parts and using $R_{\mathcal{D}}(h_S, f_0) \leq \mathbb{P}_{\mathcal{D}}[\overline{\mathcal{X}} - \{x_0\}]$, we obtain:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0)] &= \sum_{S:R_{\mathcal{D}}(h_S,f_0)\geq\epsilon} R_{\mathcal{D}}(h_S, f_0)\, \mathbb{P}[R_{\mathcal{D}}(h_S, f_0)] + \sum_{S:R_{\mathcal{D}}(h_S,f_0)<\epsilon} R_{\mathcal{D}}(h_S, f_0)\, \mathbb{P}[R_{\mathcal{D}}(h_S, f_0)] \\
&\leq \mathop{\mathbb{P}}_{\mathcal{D}}[\overline{\mathcal{X}} - \{x_0\}]\, \mathop{\mathbb{P}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon] + \epsilon \mathop{\mathbb{P}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) < \epsilon] \\
&\leq 8\epsilon \mathop{\mathbb{P}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon] + \epsilon\big(1 - \mathop{\mathbb{P}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon]\big).
\end{aligned}$$

Collecting terms in $\mathbb{P}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon]$ yields

$$\mathop{\mathbb{P}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon] \geq \frac{1}{7\epsilon}(2\epsilon - \epsilon) = \frac{1}{7}. \tag{3.35}$$

Thus, the probability over all samples S (not necessarily in $\mathcal{S}$) can be lower bounded as

$$\mathop{\mathbb{P}}_{S}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon] \geq \mathop{\mathbb{P}}_{S\in\mathcal{S}}[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon]\,\mathbb{P}[\mathcal{S}] \geq \frac{1}{7}\,\mathbb{P}[\mathcal{S}]. \tag{3.36}$$

This leads us to find a lower bound for $\mathbb{P}[\mathcal{S}]$. By the multiplicative Chernoff bound (Theorem D.4), for any $\gamma > 0$, the probability that more than $(d-1)/2$ points are drawn in a sample of size m verifies:

$$1 - \mathbb{P}[\mathcal{S}] = \mathbb{P}[S_m \geq 8\epsilon m(1+\gamma)] \leq e^{-8\epsilon m\frac{\gamma^2}{3}}. \tag{3.37}$$

Therefore, for $\epsilon = (d-1)/(32m)$ and $\gamma = 1$,

$$\mathbb{P}[S_m \geq \tfrac{d-1}{2}] \leq e^{-(d-1)/12} \leq e^{-1/12} \leq 1 - 7\delta, \tag{3.38}$$

for $\delta \leq .01$. Thus $\mathbb{P}[\mathcal{S}] \geq 7\delta$ and $\mathbb{P}_S[R_{\mathcal{D}}(h_S, f_0) \geq \epsilon] \geq \delta$. □

The theorem shows that for any algorithm $\mathcal{A}$, there exists a 'bad' distribution over $\mathcal{X}$ and a target function f for which the error of the hypothesis returned by $\mathcal{A}$ is a constant times $\frac{d}{m}$ with some constant probability. This further demonstrates the key role played by the VC-dimension in learning. The result implies in particular that PAC-learning in the realizable case is not possible when the VC-dimension is infinite.

Note that the proof shows a stronger result than the statement of the theorem: the distribution $\mathcal{D}$ is selected independently of the algorithm $\mathcal{A}$. We now present a theorem giving a lower bound in the non-realizable case. The following two lemmas will be needed for the proof.

Lemma 3.21 *Let α be a uniformly distributed random variable taking values in $\{\alpha_-, \alpha_+\}$, where $\alpha_- = \frac{1}{2} - \frac{\epsilon}{2}$ and $\alpha_+ = \frac{1}{2} + \frac{\epsilon}{2}$, and let S be a sample of $m \geq 1$ random variables $X_1, \ldots, X_m$ taking values in $\{0,1\}$ and drawn i.i.d. according to the distribution $\mathcal{D}_\alpha$ defined by $\mathbb{P}_{\mathcal{D}_\alpha}[X = 1] = \alpha$. Let h be a function from $\mathcal{X}^m$ to $\{\alpha_-, \alpha_+\}$, then the following holds:*

$$\mathop{\mathbb{E}}_{\alpha}\Big[\mathop{\mathbb{P}}_{S\sim\mathcal{D}_\alpha^m}[h(S) \neq \alpha]\Big] \geq \Phi(2\lceil m/2\rceil, \epsilon), \tag{3.39}$$

where $\Phi(m,\epsilon) = \frac{1}{4}\Big(1 - \sqrt{1 - \exp\big(-\frac{m\epsilon^2}{1-\epsilon^2}\big)}\Big)$ for all m and ϵ.

Proof: The lemma can be interpreted in terms of an experiment with two coins with biases α_- and α_+. It implies that for a discriminant rule $h(S)$ based on a sample S drawn from $\mathcal{D}_{\alpha_-}$ or $\mathcal{D}_{\alpha_+}$, to determine which coin was tossed, the sample size m must be at least $\Omega(1/\epsilon^2)$. The proof is left as an exercise (exercise D.3). □

We will make use of the fact that for any fixed ϵ the function $m \mapsto \Phi(m, x)$ is convex, which is not hard to establish.

Lemma 3.22 *Let Z be a random variable taking values in $[0, 1]$. Then, for any $\gamma \in [0, 1)$,*

$$\mathbb{P}[z > \gamma] \geq \frac{\mathbb{E}[Z] - \gamma}{1 - \gamma} > \mathbb{E}[Z] - \gamma. \tag{3.40}$$

Proof: Since the values taken by Z are in $[0, 1]$,

$$\begin{aligned}
\mathbb{E}[Z] &= \sum_{z \leq \gamma} \mathbb{P}[Z = z]z + \sum_{z > \gamma} \mathbb{P}[Z = z]z \\
&\leq \sum_{z \leq \gamma} \mathbb{P}[Z = z]\gamma + \sum_{z > \gamma} \mathbb{P}[Z = z] \\
&= \gamma\, \mathbb{P}[Z \leq \gamma] + \mathbb{P}[Z > \gamma] \\
&= \gamma(1 - \mathbb{P}[Z > \gamma]) + \mathbb{P}[Z > \gamma] \\
&= (1 - \gamma)\, \mathbb{P}[Z > \gamma] + \gamma,
\end{aligned}$$

which concludes the proof. □

Theorem 3.23 (Lower bound, non-realizable case) *Let $\mathcal{H}$ be a hypothesis set with VC-dimension $d > 1$. Then, for any $m \geq 1$ and any learning algorithm $\mathcal{A}$, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$ such that:*

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left[R_{\mathcal{D}}(h_S) - \inf_{h \in \mathcal{H}} R_{\mathcal{D}}(h) > \sqrt{\frac{d}{320m}}\right] \geq 1/64. \tag{3.41}$$

Equivalently, for any learning algorithm, the sample complexity verifies

$$m \geq \frac{d}{320\epsilon^2}. \tag{3.42}$$

Proof: Let $\overline{\mathcal{X}} = \{x_1, \ldots, x_d\} \subseteq \mathcal{X}$ be a set shattered by $\mathcal{H}$. For any $\alpha \in [0, 1]$ and any vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)^\top \in \{-1, +1\}^d$, we define a distribution $\mathcal{D}_{\boldsymbol{\sigma}}$ with support $\overline{\mathcal{X}} \times \{0, 1\}$ as follows:

$$\forall i \in [d], \quad \mathop{\mathbb{P}}_{\mathcal{D}_{\boldsymbol{\sigma}}}[(x_i, 1)] = \frac{1}{d}\Big(\frac{1}{2} + \frac{\sigma_i \alpha}{2}\Big). \tag{3.43}$$

Thus, the label of each point x_i, $i \in [d]$, follows the distribution $\mathbb{P}_{\mathcal{D}_{\boldsymbol{\sigma}}}[\cdot|x_i]$, that of a biased coin where the bias is determined by the sign of σ_i and the magnitude of α. To determine the most likely label of each point x_i, the learning algorithm will therefore need to estimate $\mathbb{P}_{\mathcal{D}_{\boldsymbol{\sigma}}}[1|x_i]$ with an accuracy better than α. To make this further difficult, α and $\boldsymbol{\sigma}$ will be selected based on the algorithm, requiring, as in lemma 3.21, $\Omega(1/\alpha^2)$ instances of each point x_i in the training sample.

Clearly, the Bayes classifier $h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}$ is defined by $h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}(x_i) = \operatorname{argmax}_{y\in\{0,1\}} \mathbb{P}[y|x_i] = 1_{\sigma_i>0}$ for all $i \in [d]$. $h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}$ is in $\mathcal{H}$ since $\overline{\mathcal{X}}$ is shattered. For all $h \in \mathcal{H}$,

$$R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h) - R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}) = \frac{1}{d}\sum_{x\in\overline{\mathcal{X}}}\Big(\frac{\alpha}{2}+\frac{\alpha}{2}\Big)1_{h(x)\neq h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}(x)} = \frac{\alpha}{d}\sum_{x\in\overline{\mathcal{X}}} 1_{h(x)\neq h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}(x)}. \quad (3.44)$$

Let h_S denote the hypothesis returned by the learning algorithm $\mathcal{A}$ after receiving a labeled sample S drawn according to $\mathcal{D}_{\boldsymbol{\sigma}}$. We will denote by $|S|_x$ the number of occurrences of a point x in S. Let $\mathcal{U}$ denote the uniform distribution over $\{-1,+1\}^d$. Then, in view of (3.44), the following holds:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{\boldsymbol{\sigma}\sim\mathcal{U}\\ S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}}\left[\frac{1}{\alpha}\big[R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h_S) - R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h^*_{\mathcal{D}_{\boldsymbol{\sigma}}})\big]\right]\\
&= \frac{1}{d}\sum_{x\in\overline{\mathcal{X}}}\mathop{\mathbb{E}}_{\substack{\boldsymbol{\sigma}\sim\mathcal{U}\\ S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}}\Big[1_{h_S(x)\neq h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}(x)}\Big]\\
&= \frac{1}{d}\sum_{x\in\overline{\mathcal{X}}}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}\sim\mathcal{U}}\Big[\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}\big[h_S(x)\neq h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}(x)\big]\Big]\\
&= \frac{1}{d}\sum_{x\in\overline{\mathcal{X}}}\sum_{n=0}^{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}\sim\mathcal{U}}\Big[\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}\big[h_S(x)\neq h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}(x)\,\big|\,|S|_x=n\big]\,\mathbb{P}[|S|_x=n]\Big]\\
&\geq \frac{1}{d}\sum_{x\in\overline{\mathcal{X}}}\sum_{n=0}^{m}\Phi(n+1,\alpha)\,\mathbb{P}[|S|_x=n] \qquad \text{(lemma 3.21)}\\
&\geq \frac{1}{d}\sum_{x\in\overline{\mathcal{X}}}\Phi(m/d+1,\alpha) \qquad \text{(convexity of } \Phi(\cdot,\alpha) \text{ and Jensen's ineq.)}\\
&= \Phi(m/d+1,\alpha).
\end{aligned}$$

Since the expectation over $\boldsymbol{\sigma}$ is lower-bounded by $\Phi(m/d+1,\alpha)$, there must exist some $\boldsymbol{\sigma} \in \{-1,+1\}^d$ for which

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}\left[\frac{1}{\alpha}\big[R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h_S) - R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h^*_{\mathcal{D}_{\boldsymbol{\sigma}}})\big]\right] > \Phi(m/d+1,\alpha). \quad (3.45)$$

Then, by lemma 3.22, for that $\boldsymbol{\sigma}$, for any $\gamma \in [0,1]$,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}\left[\frac{1}{\alpha}\big[R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h_S) - R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h^*_{\mathcal{D}_{\boldsymbol{\sigma}}})\big] > \gamma u\right] > (1-\gamma)u, \quad (3.46)$$

where $u = \Phi(m/d+1,\alpha)$. Selecting δ and ϵ such that $\delta \leq (1-\gamma)u$ and $\epsilon \leq \gamma\alpha u$ gives

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m_{\boldsymbol{\sigma}}}\big[R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h_S) - R_{\mathcal{D}_{\boldsymbol{\sigma}}}(h^*_{\mathcal{D}_{\boldsymbol{\sigma}}}) > \epsilon\big] > \delta. \quad (3.47)$$

To satisfy the inequalities defining ϵ and δ, let $\gamma = 1 - 8\delta$. Then,

$$\delta \leq (1-\gamma)u \iff u \geq \frac{1}{8} \tag{3.48}$$

$$\iff \frac{1}{4}\left(1 - \sqrt{1 - \exp\left(-\frac{(m/d+1)\alpha^2}{1-\alpha^2}\right)}\right) \geq \frac{1}{8} \tag{3.49}$$

$$\iff \frac{(m/d+1)\alpha^2}{1-\alpha^2} \leq \log\frac{4}{3} \tag{3.50}$$

$$\iff \frac{m}{d} \leq \left(\frac{1}{\alpha^2} - 1\right)\log\frac{4}{3} - 1. \tag{3.51}$$

Selecting $\alpha = 8\epsilon/(1-8\delta)$ gives $\epsilon = \gamma\alpha/8$ and the condition

$$\frac{m}{d} \leq \left(\frac{(1-8\delta)^2}{64\epsilon^2} - 1\right)\log\frac{4}{3} - 1. \tag{3.52}$$

Let $f(1/\epsilon^2)$ denote the right-hand side. We are seeking a sufficient condition of the form $m/d \leq \omega/\epsilon^2$. Since $\epsilon \leq 1/64$, to ensure that $\omega/\epsilon^2 \leq f(1/\epsilon^2)$, it suffices to impose $\frac{\omega}{(1/64)^2} = f\left(\frac{1}{(1/64)^2}\right)$. This condition gives

$$\omega = (7/64)^2\log(4/3) - (1/64)^2(\log(4/3)+1) \approx .003127 \geq 1/320 = .003125.$$

Thus, $\epsilon^2 \leq \frac{1}{320(m/d)}$ is sufficient to ensure the inequalities. □

The theorem shows that for any algorithm $\mathcal{A}$, in the non-realizable case, there exists a 'bad' distribution over $\mathcal{X} \times \{0,1\}$ such that the error of the hypothesis returned by $\mathcal{A}$ is a constant times $\sqrt{\frac{d}{m}}$ with some constant probability. The VC-dimension appears as a critical quantity in learning in this general setting as well. In particular, with an infinite VC-dimension, agnostic PAC-learning is not possible.

3.5 Chapter notes

The use of Rademacher complexity for deriving generalization bounds in learning was first advocated by Koltchinskii [2001], Koltchinskii and Panchenko [2000], and Bartlett, Boucheron, and Lugosi [2002a], see also [Koltchinskii and Panchenko, 2002, Bartlett and Mendelson, 2002]. Bartlett, Bousquet, and Mendelson [2002b] introduced the notion of *local Rademacher complexity*, that is the Rademacher complexity restricted to a subset of the hypothesis set limited by a bound on the variance. This can be used to derive better guarantees under some regularity assumptions about the noise.

Theorem 3.7 is due to Massart [2000]. The notion of VC-dimension was introduced by Vapnik and Chervonenkis [1971] and has been since extensively studied [Vapnik, 2006, Vapnik and Chervonenkis, 1974, Blumer et al., 1989, Assouad, 1983, Dudley,

1999]. In addition to the key role it plays in machine learning, the VC-dimension is also widely used in a variety of other areas of computer science and mathematics (e.g., see Shelah [1972], Chazelle [2000]). Theorem 3.17 is known as *Sauer's lemma* in the learning community, however the result was first given by Vapnik and Chervonenkis [1971] (in a somewhat different version) and later independently by Sauer [1972] and Shelah [1972].

In the realizable case, lower bounds for the expected error in terms of the VC-dimension were given by Vapnik and Chervonenkis [1974] and Haussler et al. [1988]. Later, a lower bound for the probability of error such as that of theorem 3.20 was given by Blumer et al. [1989]. Theorem 3.20 and its proof, which improves upon this previous result, are due to Ehrenfeucht, Haussler, Kearns, and Valiant [1988]. Devroye and Lugosi [1995] gave slightly tighter bounds for the same problem with a more complex expression. Theorem 3.23 giving a lower bound in the non-realizable case and the proof presented are due to Anthony and Bartlett [1999]. For other examples of application of the probabilistic method demonstrating its full power, consult the reference book of Alon and Spencer [1992].

There are several other measures of the complexity of a family of functions used in machine learning, including *covering numbers*, *packing numbers*, and some other complexity measures discussed in chapter 11. A covering number $\mathcal{N}_p(\mathcal{G}, \epsilon)$ is the minimal number of L_p balls of radius $\epsilon > 0$ needed to cover a family of loss functions $\mathcal{G}$. A packing number $\mathcal{M}_p(\mathcal{G}, \epsilon)$ is the maximum number of non-overlapping L_p balls of radius ϵ centered in $\mathcal{G}$. The two notions are closely related, in particular it can be shown straightforwardly that $\mathcal{M}_p(\mathcal{G}, 2\epsilon) \leq \mathcal{N}_p(\mathcal{G}, \epsilon) \leq \mathcal{M}_p(\mathcal{G}, \epsilon)$ for $\mathcal{G}$ and $\epsilon > 0$. Each complexity measure naturally induces a different reduction of infinite hypothesis sets to finite ones, thereby resulting in generalization bounds for infinite hypothesis sets. Exercise 3.31 illustrates the use of covering numbers for deriving generalization bounds using a very simple proof. There are also close relationships between these complexity measures: for example, by Dudley's theorem, the empirical Rademacher complexity can be bounded in terms of $\mathcal{N}_2(\mathcal{G}, \epsilon)$ [Dudley, 1967, 1987] and the covering and packing numbers can be bounded in terms of the VC-dimension [Haussler, 1995]. See also [Ledoux and Talagrand, 1991, Alon et al., 1997, Anthony and Bartlett, 1999, Cucker and Smale, 2001, Vidyasagar, 1997] for a number of upper bounds on the covering number in terms of other complexity measures.

3.6 Exercises

3.1 Growth function of intervals in $\mathbb{R}$. Let $\mathcal{H}$ be the set of intervals in $\mathbb{R}$. The VC-dimension of $\mathcal{H}$ is 2. Compute its shattering coefficient $\Pi_{\mathcal{H}}(m)$, $m \geq 0$. Compare your result with the general bound for growth functions.

3.2 Growth function and Rademacher complexity of thresholds in $\mathbb{R}$. Let $\mathcal{H}$ be the family of threshold functions over the real line: $\mathcal{H} = \{x \mapsto 1_{x \leq \theta} \colon \theta \in \mathbb{R}\} \cup \{x \mapsto 1_{x \geq \theta} \colon \theta \in \mathbb{R}\}$. Give an upper bound on the growth function $\Pi_m(\mathcal{H})$. Use that to derive an upper bound on $\mathfrak{R}_m(\mathcal{H})$.

3.3 Growth function of linear combinations. A *linearly separable labeling* of a set $\mathcal{X}$ of vectors in $\mathbb{R}^d$ is a classification of $\mathcal{X}$ into two sets $\mathcal{X}^+$ and $\mathcal{X}^-$ with $\mathcal{X}^+ = \{\mathbf{x} \in \mathcal{X} \colon \mathbf{w} \cdot \mathbf{x} > 0\}$ and $\mathcal{X}^- = \{\mathbf{x} \in \mathcal{X} \colon \mathbf{w} \cdot \mathbf{x} < 0\}$ for some $\mathbf{w} \in \mathbb{R}^d$.

Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a subset of $\mathbb{R}^d$.

(a) Let $\{\mathcal{X}^+, \mathcal{X}^-\}$ be a dichotomy of $\mathcal{X}$ and let $\mathbf{x}_{m+1} \in \mathbb{R}^d$. Show that $\{\mathcal{X}^+ \cup \{\mathbf{x}_{m+1}\}, \mathcal{X}^-\}$ and $\{\mathcal{X}^+, \mathcal{X}^- \cup \{\mathbf{x}_{m+1}\}\}$ are linearly separable by a hyperplane going through the origin if and only if $\{\mathcal{X}^+, \mathcal{X}^-\}$ is linearly separable by a hyperplane going through the origin and $\mathbf{x}_{m+1}$.

(b) Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a subset of $\mathbb{R}^d$ such that any k-element subset of $\mathcal{X}$ with $k \leq d$ is linearly independent. Then, show that the number of linearly separable labelings of $\mathcal{X}$ is $C(m, d) = 2 \sum_{k=0}^{d-1} \binom{m-1}{k}$. (*Hint*: prove by induction that $C(m+1, d) = C(m, d) + C(m, d-1)$.

(c) Let $f_1, \ldots, f_p$ be p functions mapping $\mathbb{R}^d$ to $\mathbb{R}$. Define $\mathcal{F}$ as the family of classifiers based on linear combinations of these functions:

$$\mathcal{F} = \left\{ x \mapsto \operatorname{sgn}\left(\sum_{k=1}^{p} a_k f_k(x) \right) \colon a_1, \ldots, a_p \in \mathbb{R} \right\}.$$

Define Ψ by $\Psi(x) = (f_1(x), \ldots, f_p(x))$. Assume that there exists $x_1, \ldots, x_m \in \mathbb{R}^d$ such that every p-subset of $\{\Psi(x_1), \ldots, \Psi(x_m)\}$ is linearly independent. Then, show that

$$\Pi_{\mathcal{F}}(m) = 2 \sum_{i=0}^{p-1} \binom{m-1}{i}.$$

3.4 Lower bound on growth function. Prove that Sauer's lemma (theorem 3.17) is tight, i.e., for any set $\mathcal{X}$ of $m > d$ elements, show that there exists a hypothesis class $\mathcal{H}$ of VC-dimension d such that $\Pi_{\mathcal{H}}(m) = \sum_{i=0}^{d} \binom{m}{i}$.

3.5 Finer Rademacher upper bound. Show that a finer upper bound on the Rademacher complexity of the family $\mathcal{G}$ can be given in terms of $\mathbb{E}_S[\Pi(\mathcal{G}, S)]$, where $\Pi(\mathcal{G}, S)$ is the number of ways to label the points in sample S.

3.6 Singleton hypothesis class. Consider the trivial hypothesis set $\mathcal{H} = \{h_0\}$.

(a) Show that $\mathfrak{R}_m(\mathcal{H}) = 0$ for any $m > 0$.

(b) Use a similar construction to show that Massart's lemma (theorem 3.7) is tight.

3.7 Two function hypothesis class. Let $\mathcal{H}$ be a hypothesis set reduced to two functions: $\mathcal{H} = \{h_{-1}, h_{+1}\}$ and let $S = (x_1, \ldots, x_m) \subseteq \mathcal{X}$ be a sample of size m.

(a) Assume that h_{-1} is the constant function taking value -1 and h_{+1} the constant function taking the value $+1$. What is the VC-dimension d of $\mathcal{H}$? Upper bound the empirical Rademacher complexity $\widehat{\mathfrak{R}}_S(\mathcal{H})$ (*Hint*: express $\widehat{\mathfrak{R}}_S(\mathcal{H})$ in terms of the absolute value of a sum of Rademacher variables and apply Jensen's inequality) and compare your bound with $\sqrt{d/m}$.

(b) Assume that h_{-1} is the constant function taking value -1 and h_{+1} the function taking value -1 everywhere except at x_1 where it takes the value $+1$. What is the VC-dimension d of $\mathcal{H}$? Compute the empirical Rademacher complexity $\widehat{\mathfrak{R}}_S(\mathcal{H})$.

3.8 Rademacher identities. Fix $m \geq 1$. Prove the following identities for any $\alpha \in \mathbb{R}$ and any two hypothesis sets $\mathcal{H}$ and $\mathcal{H}'$ of functions mapping from $\mathcal{X}$ to $\mathbb{R}$:

(a) $\mathfrak{R}_m(\alpha\mathcal{H}) = |\alpha|\mathfrak{R}_m(\mathcal{H})$.

(b) $\mathfrak{R}_m(\mathcal{H} + \mathcal{H}') = \mathfrak{R}_m(\mathcal{H}) + \mathfrak{R}_m(\mathcal{H}')$.

(c) $\mathfrak{R}_m(\{\max(h, h') \colon h \in \mathcal{H}, h' \in \mathcal{H}'\}) \leq \mathfrak{R}_m(\mathcal{H}) + \mathfrak{R}_m(\mathcal{H}')$,
where $\max(h, h')$ denotes the function $x \mapsto \max_{x \in \mathcal{X}}(h(x), h'(x))$ (*Hint*: you could use the identity $\max(a, b) = \frac{1}{2}[a + b + |a - b|]$ valid for all $a, b \in \mathbb{R}$ and Talagrand's contraction lemma (see lemma 5.7)).

3.9 Rademacher complexity of intersection of concepts. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two families of functions mapping $\mathcal{X}$ to $\{0, 1\}$ and let $\mathcal{H} = \{h_1 h_2 \colon h_1 \in \mathcal{H}_1, h_2 \in \mathcal{H}_2\}$. Show that the empirical Rademacher complexity of $\mathcal{H}$ for any sample S of size m can be bounded as follows:

$$\widehat{\mathfrak{R}}_S(H) \leq \widehat{\mathfrak{R}}_S(\mathcal{H}_1) + \widehat{\mathfrak{R}}_S(\mathcal{H}_2).$$

Hint: use the Lipschitz function $x \mapsto \max(0, x-1)$ and Talagrand's contraction lemma.

Use that to bound the Rademacher complexity $\mathfrak{R}_m(\mathcal{U})$ of the family $\mathcal{U}$ of intersections of two concepts c_1 and c_2 with $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$ in terms of the Rademacher complexities of $\mathcal{C}_1$ and $\mathcal{C}_2$.

3.10 Rademacher complexity of prediction vector. Let $S = (x_1, \ldots, x_m)$ be a sample of size m and fix $h\colon \mathcal{X} \to \mathbb{R}$.

(a) Denote by $\mathbf{u}$ the vector of predictions of h for S: $\mathbf{u} = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_m) \end{bmatrix}$. Give an upper bound on the empirical Rademacher complexity $\widehat{\mathfrak{R}}_S(\mathcal{H})$ of $\mathcal{H} = \{h, -h\}$ in terms of $\|\mathbf{u}\|_2$ (*Hint*: express $\widehat{\mathfrak{R}}_S(\mathcal{H})$ in terms of the expectation of an absolute value and apply Jensen's inequality). Suppose that $h(x_i) \in \{0, -1, +1\}$ for all $i \in [m]$. Express the bound on the Rademacher complexity in terms of the sparsity measure $n = |\{i \mid h(x_i) \neq 0\}|$. What is that upper bound for the extreme values of the sparsity measure?

(b) Let $\mathcal{F}$ be a family of functions mapping $\mathcal{X}$ to $\mathbb{R}$. Give an upper bound on the empirical Rademacher complexity of $\mathcal{F} + h = \{f + h\colon f \in \mathcal{F}\}$ and that of $\mathcal{F} \pm h = (\mathcal{F} + h) \cup (\mathcal{F} - h)$ in terms of $\widehat{\mathfrak{R}}_S(\mathcal{F})$ and $\|\mathbf{u}\|_2$.

3.11 Rademacher complexity of regularized neural networks. Let the input space be $\mathcal{X} = \mathbb{R}^{n_1}$. In this problem, we consider the family of regularized neural networks defined by the following set of functions mapping $\mathcal{X}$ to $\mathbb{R}$:

$$\mathcal{H} = \left\{ \mathbf{x} \mapsto \sum_{j=1}^{n_2} w_j \sigma(\mathbf{u}_j \cdot \mathbf{x})\colon \|\mathbf{w}\|_1 \leq \Lambda', \|\mathbf{u}_j\|_2 \leq \Lambda, \forall j \in [n_2] \right\},$$

where σ is an L-Lipschitz function. As an example, σ could be the sigmoid function which is 1-Lipschitz.

(a) Show that $\widehat{\mathfrak{R}}_S(\mathcal{H}) = \frac{\Lambda'}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\|\mathbf{u}\|_2 \leq \Lambda} \left| \sum_{i=1}^m \sigma_i \sigma(\mathbf{u} \cdot \mathbf{x}_i) \right| \right]$.

(b) Use the following form of Talagrand's lemma valid for all hypothesis sets $\mathcal{H}$ and L-Lipschitz function Φ:

$$\frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \left| \sum_{i=1}^m \sigma_i (\Phi \circ h)(x_i) \right| \right] \leq \frac{L}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \left| \sum_{i=1}^m \sigma_i h(x_i) \right| \right],$$

to upper bound $\widehat{\mathfrak{R}}_S(\mathcal{H})$ in terms of the empirical Rademacher complexity of $\mathcal{H}'$, where $\mathcal{H}'$ is defined by

$$\mathcal{H}' = \{\mathbf{x} \mapsto s(\mathbf{u} \cdot \mathbf{x}) \colon \|\mathbf{u}\|_2 \leq \Lambda, s \in \{-1, +1\}\}.$$

(c) Use the Cauchy-Schwarz inequality to show that

$$\widehat{\mathfrak{R}}_S(\mathcal{H}') = \frac{\Lambda}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2 \right].$$

(d) Use the inequality $\mathbb{E}_{\mathbf{v}}[\|\mathbf{v}\|_2] \leq \sqrt{\mathbb{E}_{\mathbf{v}}[\|\mathbf{v}\|_2^2]}$, which holds by Jensen's inequality to upper bound $\widehat{\mathfrak{R}}_S(\mathcal{H}')$.

(e) Assume that for all $\mathbf{x} \in S$, $\|\mathbf{x}\|_2 \leq r$ for some $r > 0$. Use the previous questions to derive an upper bound on the Rademacher complexity of $\mathcal{H}$ in terms of r.

3.12 Rademacher complexity. Professor Jesetoo claims to have found a better bound on the Rademacher complexity of any hypothesis set $\mathcal{H}$ of functions taking values in $\{-1, +1\}$, in terms of its VC-dimension $\mathrm{VCdim}(\mathcal{H})$. His bound is of the form $\mathfrak{R}_m(\mathcal{H}) \leq O\big(\frac{\mathrm{VCdim}(\mathcal{H})}{m}\big)$. Can you show that Professor Jesetoo's claim cannot be correct? (*Hint*: consider a hypothesis set $\mathcal{H}$ reduced to just two simple functions.)

3.13 VC-dimension of union of k intervals. What is the VC-dimension of subsets of the real line formed by the union of k intervals?

3.14 VC-dimension of finite hypothesis sets. Show that the VC-dimension of a finite hypothesis set $\mathcal{H}$ is at most $\log_2 |\mathcal{H}|$.

3.15 VC-dimension of subsets. What is the VC-dimension of the set of subsets I_α of the real line parameterized by a single parameter α: $I_\alpha = [\alpha, \alpha+1] \cup [\alpha+2, +\infty)$?

3.16 VC-dimension of axis-aligned squares and triangles.

(a) What is the VC-dimension of axis-aligned squares in the plane?

(b) Consider right triangles in the plane with the sides adjacent to the right angle both parallel to the axes and with the right angle in the lower left corner. What is the VC-dimension of this family?

3.17 VC-dimension of closed balls in $\mathbb{R}^n$. Show that the VC-dimension of the set of all closed balls in $\mathbb{R}^n$, i.e., sets of the form $\{x \in \mathbb{R}^n \colon \|x - x_0\|^2 \leq r\}$ for some $x_0 \in \mathbb{R}^n$ and $r \geq 0$, is less than or equal to $n + 2$.

3.18 VC-dimension of ellipsoids. What is the VC-dimension of the set of all ellipsoids in $\mathbb{R}^n$?

3.19 VC-dimension of a vector space of real functions. Let F be a finite-dimensional vector space of real functions on $\mathbb{R}^n$, $\dim(F) = r < \infty$. Let $\mathcal{H}$ be the set of hypotheses:

$$\mathcal{H} = \{\{x \colon f(x) \geq 0\} \colon f \in F\}.$$

Show that d, the VC-dimension of $\mathcal{H}$, is finite and that $d \leq r$. (*Hint*: select an arbitrary set of $m = r + 1$ points and consider linear mapping $u \colon F \to \mathbb{R}^m$ defined by: $u(f) = (f(x_1), \ldots, f(x_m))$.)

3.20 VC-dimension of sine functions. Consider the hypothesis family of sine functions (Example 3.16): $\{x \to \sin(\omega x) \colon \omega \in \mathbb{R}\}$.

(a) Show that for any $x \in \mathbb{R}$ the points $x, 2x, 3x$ and $4x$ cannot be shattered by this family of sine functions.

(b) Show that the VC-dimension of the family of sine functions is infinite. (*Hint*: show that $\{2^{-i} \colon i \leq m\}$ can be shattered for any $m > 0$.)

3.21 VC-dimension of union of halfspaces. Provide an upper bound on the VC-dimension of the class of hypotheses described by the unions of k halfspaces.

3.22 VC-dimension of intersection of halfspaces. Consider the class $\mathcal{C}_k$ of convex intersections of k halfspaces. Give lower and upper bound estimates for $\mathrm{VCdim}(\mathcal{C}_k)$.

3.23 VC-dimension of intersection concepts.

(a) Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two concept classes. Show that for any concept class $\mathcal{C} = \{c_1 \cap c_2 \colon c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\}$,

$$\Pi_{\mathcal{C}}(m) \leq \Pi_{\mathcal{C}_1}(m)\, \Pi_{\mathcal{C}_2}(m). \tag{3.53}$$

(b) Let $\mathcal{C}$ be a concept class with VC-dimension d and let $\mathcal{C}_s$ be the concept class formed by all intersections of s concepts from $\mathcal{C}$, $s \geq 1$. Show that the VC-dimension of $\mathcal{C}_s$ is bounded by $2ds \log_2(3s)$. (*Hint*: show that $\log_2(3x) < 9x/(2e)$ for any $x \geq 2$.)

3.24 VC-dimension of union of concepts. Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of functions mapping from $\mathcal{X}$ into $\{0,1\}$, and assume that both $\mathcal{A}$ and $\mathcal{B}$ have finite VC-dimension, with $\mathrm{VCdim}(\mathcal{A}) = d_{\mathcal{A}}$ and $\mathrm{VCdim}(\mathcal{B}) = d_{\mathcal{B}}$. Let $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ be the union of $\mathcal{A}$ and $\mathcal{B}$.

(a) Prove that for all m, $\Pi_{\mathcal{C}}(m) \leq \Pi_{\mathcal{A}}(m) + \Pi_{\mathcal{B}}(m)$.

(b) Use Sauer's lemma to show that for $m \geq d_{\mathcal{A}} + d_{\mathcal{B}} + 2$, $\Pi_{\mathcal{C}}(m) < 2^m$, and give a bound on the VC-dimension of $\mathcal{C}$.

3.25 VC-dimension of symmetric difference of concepts. For two sets $\mathcal{A}$ and $\mathcal{B}$, let $\mathcal{A}\Delta\mathcal{B}$ denote the symmetric difference of $\mathcal{A}$ and $\mathcal{B}$, i.e., $\mathcal{A}\Delta\mathcal{B} = (\mathcal{A}\cup\mathcal{B}) - (\mathcal{A}\cap\mathcal{B})$. Let $\mathcal{H}$ be a non-empty family of subsets of $\mathcal{X}$ with finite VC-dimension. Let $\mathcal{A}$ be an element of $\mathcal{H}$ and define $\mathcal{H}\Delta\mathcal{A} = \{X\Delta\mathcal{A} \colon X \in \mathcal{H}\}$. Show that

$$\mathrm{VCdim}(\mathcal{H}\Delta\mathcal{A}) = \mathrm{VCdim}(\mathcal{H}).$$

3.26 Symmetric functions. A function $h \colon \{0,1\}^n \to \{0,1\}$ is *symmetric* if its value is uniquely determined by the number of 1's in the input. Let $\mathcal{C}$ denote the set of all symmetric functions.

(a) Determine the VC-dimension of $\mathcal{C}$.

(b) Give lower and upper bounds on the sample complexity of any consistent PAC learning algorithm for $\mathcal{C}$.

(c) Note that any hypothesis $h \in \mathcal{C}$ can be represented by a vector $(y_0, y_1, \ldots, y_n) \in \{0,1\}^{n+1}$, where y_i is the value of h on examples having precisely i 1's. Devise a consistent learning algorithm for $\mathcal{C}$ based on this representation.

3.27 VC-dimension of neural networks.

Let $\mathcal{C}$ be a concept class over $\mathbb{R}^r$ with VC-dimension d. A $\mathcal{C}$-neural network with one intermediate layer is a concept defined over $\mathbb{R}^n$ that can be represented by a directed acyclic graph such as that of Figure 3.7, in which the input nodes are those at the bottom and in which each other node is labeled with a concept $c \in \mathcal{C}$.

The output of the neural network for a given input vector $(x_1, \ldots, x_n)$ is obtained as follows. First, each of the n input nodes is labeled with the corresponding value $x_i \in \mathbb{R}$. Next, the value at a node u in the higher layer and labeled with c is obtained by applying c to the values of the input nodes admitting an

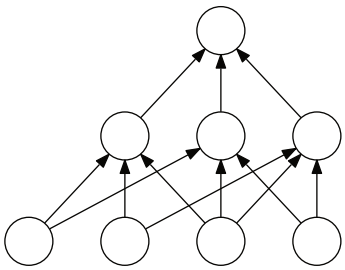


Figure 3.7
A neural network with one intermediate layer.

edge ending in u. Note that since c takes values in $\{0,1\}$, the value at u is in $\{0,1\}$. The value at the top or output node is obtained similarly by applying the corresponding concept to the values of the nodes admitting an edge to the output node.

(a) Let $\mathcal{H}$ denote the set of all neural networks defined as above with $k \geq 2$ internal nodes. Show that the growth function $\Pi_{\mathcal{H}}(m)$ can be upper bounded in terms of the product of the growth functions of the hypothesis sets defined at each intermediate layer.

(b) Use that to upper bound the VC-dimension of the $\mathcal{C}$-neural networks (*Hint*: you can use the implication $m = 2x\log_2(xy) \Rightarrow m > x\log_2(ym)$ valid for $m \geq 1$, and $x, y > 0$ with $xy > 4$).

(c) Let $\mathcal{C}$ be the family of concept classes defined by threshold functions $\mathcal{C} = \{\text{sgn}(\sum_{j=1}^{r} w_j x_j) \colon \mathbf{w} \in \mathbb{R}^r\}$. Give an upper bound on the VC-dimension of $\mathcal{H}$ in terms of k and r.

3.28 VC-dimension of convex combinations. Let $\mathcal{H}$ be a family of functions mapping from an input space $\mathcal{X}$ to $\{-1,+1\}$ and let T be a positive integer. Give an upper bound on the VC-dimension of the family of functions $\mathcal{F}_T$ defined by

$$\mathcal{F} = \left\{ \text{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t \right) : h_t \in \mathcal{H}, \alpha_t \geq 0, \sum_{t=1}^{T} \alpha_t \leq 1 \right\}.$$

(*Hint*: you can use exercise 3.27 and its solution).

3.29 Infinite VC-dimension.

(a) Show that if a concept class $\mathcal{C}$ has infinite VC-dimension, then it is not PAC-learnable.

(b) In the standard PAC-learning scenario, the learning algorithm receives all examples first and then computes its hypothesis. Within that setting, PAC-learning of concept classes with infinite VC-dimension is not possible as seen in the previous question.

Imagine now a different scenario where the learning algorithm can alternate between drawing more examples and computation. The objective of this problem is to prove that PAC-learning can then be possible for some concept classes with infinite VC-dimension.

Consider for example the special case of the concept class $\mathcal{C}$ of all subsets of natural numbers. Professor Vitres has an idea for the first stage of a learning algorithm L PAC-learning $\mathcal{C}$. In the first stage, L draws a sufficient number of points m such that the probability of drawing a point beyond the maximum value M observed be small with high confidence. Can you complete Professor Vitres' idea by describing the second stage of the algorithm so that it PAC-learns $\mathcal{C}$? The description should be augmented with the proof that L can PAC-learn $\mathcal{C}$.

3.30 VC-dimension generalization bound – realizable case. In this exercise we show that the bound given in corollary 3.19 can be improved to $O(\frac{d\log(m/d)}{m})$ in the realizable setting. Assume we are in the realizable scenario, i.e. the target concept is included in our hypothesis class $\mathcal{H}$. We will show that if a hypothesis h is consistent with a sample $S \sim \mathcal{D}^m$ then for any $\epsilon > 0$ such that $m\epsilon \geq 8$

$$\mathbb{P}[R(h) > \epsilon] \leq 2\left[\frac{2em}{d}\right]^d 2^{-m\epsilon/2}\,. \tag{3.54}$$

(a) Let $\mathcal{H}_S \subseteq \mathcal{H}$ be the subset of hypotheses consistent with the sample S, let $\widehat{R}_S(h)$ denote the empirical error with respect to the sample S and define S' as another independent sample drawn from $\mathcal{D}^m$. Show that the following inequality holds for any $h_0 \in \mathcal{H}_S$:

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}_S} |\widehat{R}_S(h) - \widehat{R}_{S'}(h)| > \frac{\epsilon}{2}\right] \geq \mathbb{P}\left[B(m,\epsilon) > \frac{m\epsilon}{2}\right]\mathbb{P}[R(h_0) > \epsilon]\,,$$

where $B(m,\epsilon)$ is a binomial random variable with parameters (m,ϵ). (*Hint*: prove and use the fact that $\mathbb{P}[\widehat{R}_S(h) \geq \frac{\epsilon}{2}] \geq \mathbb{P}[\widehat{R}_S(h) > \frac{\epsilon}{2} \wedge R(h) > \epsilon]$.)

(b) Prove that $\mathbb{P}\left[B(m,\epsilon) > \frac{m\epsilon}{2}\right] \geq \frac{1}{2}$. Use this inequality along with the result from (a) to show that for any $h_0 \in \mathcal{H}_S$

$$\mathbb{P}\left[R(h_0) > \epsilon\right] \leq 2\,\mathbb{P}\left[\sup_{h\in\mathcal{H}_S} |\widehat{R}_S(h) - \widehat{R}_{S'}(h)| > \frac{\epsilon}{2}\right].$$

(c) Instead of drawing two samples, we can draw one sample T of size $2m$ then uniformly at random split it into S and S'. The right hand side of part (b) can then be rewritten as:

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}_S} |\widehat{R}_S(h) - \widehat{R}_{S'}(h)| > \frac{\epsilon}{2}\right] = \mathop{\mathbb{P}}_{\substack{T\sim\mathcal{D}^{2m}:\\ T\to[S,S']}}\left[\exists h\in\mathcal{H}\colon \widehat{R}_S(h) = 0 \wedge \widehat{R}_{S'}(h) > \frac{\epsilon}{2}\right].$$

Let h_0 be a hypothesis such that $\widehat{R}_T(h_0) > \frac{\epsilon}{2}$ and let $l > \frac{m\epsilon}{2}$ be the total number of errors h_0 makes on T. Show that the probability of all l errors falling into S' is upper bounded by 2^{-l}.

(d) Part (b) implies that for any $h \in \mathcal{H}$

$$\mathop{\mathbb{P}}_{\substack{T\sim\mathcal{D}^{2m}:\\ T\to(S,S')}}\left[\widehat{R}_S(h) = 0 \ \wedge\ \widehat{R}_{S'}(h) > \frac{\epsilon}{2} \ \middle|\ \widehat{R}_T(h_0) > \frac{\epsilon}{2}\right] \leq 2^{-l}.$$

Use this bound to show that for any $h \in \mathcal{H}$

$$\mathop{\mathbb{P}}_{\substack{T\sim\mathcal{D}^{2m}:\\ T\to(S,S')}}\left[\widehat{R}_S(h) = 0 \ \wedge\ \widehat{R}_{S'}(h) > \frac{\epsilon}{2}\right] \leq 2^{-\frac{\epsilon m}{2}}.$$

(e) Complete the proof of inequality (3.54) by using the union bound to upper bound $\mathbb{P}_{\substack{T\sim\mathcal{D}^{2m}:\\ T\to(S,S')}}\left[\exists h \in \mathcal{H}\colon \widehat{R}_S(h) = 0 \ \wedge\ \widehat{R}_{S'}(h) > \frac{\epsilon}{2}\right]$. Show that we can achieve a high probability generalization bound that is of the order $O(\frac{d\log(m/d)}{m})$.

3.31 Generalization bound based on covering numbers. Let $\mathcal{H}$ be a family of functions mapping $\mathcal{X}$ to a subset of real numbers $\mathcal{Y} \subseteq \mathbb{R}$. For any $\epsilon > 0$, the *covering number* $\mathcal{N}(\mathcal{H},\epsilon)$ of $\mathcal{H}$ for the L_∞ norm is the minimal $k \in \mathbb{N}$ such that $\mathcal{H}$ can be covered with k balls of radius ϵ, that is, there exists $\{h_1,\ldots,h_k\} \subseteq \mathcal{H}$ such that, for all $h \in \mathcal{H}$, there exists $i \leq k$ with $\|h - h_i\|_\infty = \max_{x\in\mathcal{X}} |h(x) - h_i(x)| \leq \epsilon$. In particular, when $\mathcal{H}$ is a compact set, a finite covering can be extracted from a covering of $\mathcal{H}$ with balls of radius ϵ and thus $\mathcal{N}(\mathcal{H},\epsilon)$ is finite.

Covering numbers provide a measure of the complexity of a class of functions: the larger the covering number, the richer is the family of functions. The objective of this problem is to illustrate this by proving a learning bound in the case of the squared loss. Let $\mathcal{D}$ denote a distribution over $\mathcal{X}\times\mathcal{Y}$ according to which

labeled examples are drawn. Then, the generalization error of $h \in \mathcal{H}$ for the squared loss is defined by $R(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[(h(x)-y)^2]$ and its empirical error for a labeled sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ by $\widehat{R}_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2$. We will assume that $\mathcal{H}$ is bounded, that is there exists $M > 0$ such that $|h(x) - y| \leq M$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The following is the generalization bound proven in this problem:

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}} |R(h) - \widehat{R}_S(h)| \geq \epsilon\right] \leq \mathcal{N}\Big(\mathcal{H}, \frac{\epsilon}{8M}\Big) 2\exp\Big(\frac{-m\epsilon^2}{2M^4}\Big). \tag{3.55}$$

The proof is based on the following steps.

(a) Let $L_S = R(h) - \widehat{R}_S(h)$, then show that for all $h_1, h_2 \in \mathcal{H}$ and any labeled sample S, the following inequality holds:

$$|L_S(h_1) - L_S(h_2)| \leq 4M\|h_1 - h_2\|_\infty .$$

(b) Assume that $\mathcal{H}$ can be covered by k subsets $\mathcal{B}_1, \ldots, \mathcal{B}_k$, that is $\mathcal{H} = \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_k$. Then, show that, for any $\epsilon > 0$, the following upper bound holds:

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}} |L_S(h)| \geq \epsilon\right] \leq \sum_{i=1}^{k} \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{B}_i} |L_S(h)| \geq \epsilon\right].$$

(c) Finally, let $k = \mathcal{N}(\mathcal{H}, \frac{\epsilon}{8M})$ and let $\mathcal{B}_1, \ldots, \mathcal{B}_k$ be balls of radius $\epsilon/(8M)$ centered at $h_1, \ldots, h_k$ covering $\mathcal{H}$. Use part (a) to show that for all $i \in [k]$,

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{B}_i} |L_S(h)| \geq \epsilon\right] \leq \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[|L_S(h_i)| \geq \frac{\epsilon}{2}\right],$$

and apply Hoeffding's inequality (theorem D.2) to prove (3.55).

4 Model Selection

A key problem in the design of learning algorithms is the choice of the hypothesis set $\mathcal{H}$. This is known as the *model selection* problem. How should the hypothesis set $\mathcal{H}$ be chosen? A rich or complex enough hypothesis set could contain the ideal Bayes classifier. On the other hand, learning with such a complex family becomes a very difficult task. More generally, the choice of $\mathcal{H}$ is subject to a trade-off that can be analyzed in terms of the *estimation* and *approximation errors*.

Our discussion will focus on the particular case of binary classification but much of what is discussed can be straightforwardly extended to different tasks and loss functions.

4.1 Estimation and approximation errors

Let $\mathcal{H}$ be a family of functions mapping $\mathcal{X}$ to $\{-1, +1\}$. The *excess error* of a hypothesis h chosen from $\mathcal{H}$, that is the difference between its error $R(h)$ and the Bayes error R^*, can be decomposed as follows:

$$R(h) - R^* = \underbrace{\left(R(h) - \inf_{h \in \mathcal{H}} R(h)\right)}_{\text{estimation}} + \underbrace{\left(\inf_{h \in \mathcal{H}} R(h) - R^*\right)}_{\text{approximation}}. \tag{4.1}$$

The first term is called the *estimation error*, the second term the *approximation error*. The estimation error depends on the hypothesis h selected. It measures the error of h with respect to the infimum of the errors achieved by hypotheses in $\mathcal{H}$, or that of the best-in-class hypothesis h^* when that infimum is reached. Note that the definition of agnostic PAC-learning is precisely based on the estimation error.

The approximation error measures how well the Bayes error can be approximated using $\mathcal{H}$. It is a property of the hypothesis set $\mathcal{H}$, a measure of its richness. For a more complex or richer hypothesis $\mathcal{H}$, the approximation error tends to be smaller at the price of a larger estimation error. This is illustrated by Figure 4.1.

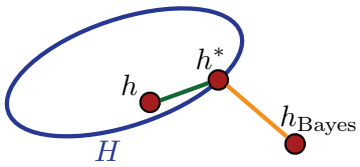


Figure 4.1
Illustration of the estimation error (in green) and approximation error (in orange). Here, it is assumed that there exists a best-in-class hypothesis, that is h^* such that $R(h^*) = \inf_{h \in \mathcal{H}} R(h)$.

Model selection consists of choosing $\mathcal{H}$ with a favorable trade-off between the approximation and estimation errors. Note, however, that the approximation error is not accessible, since in general the underlying distribution $\mathcal{D}$ needed to determine R^* is not known. Even with various noise assumptions, estimating the approximation error is difficult. In contrast, the *estimation error of an algorithm* $\mathcal{A}$, that is, the estimation error of the hypothesis h_S returned after training on a sample S, can sometimes be bounded using generalization bounds as shown in the next section.

4.2 Empirical risk minimization (ERM)

A standard algorithm for which the estimation error can be bounded is *Empirical Risk Minimization* (ERM). ERM seeks to minimize the error on the training sample:[4]

$$h_S^{\text{ERM}} = \operatorname*{argmin}_{h \in \mathcal{H}} \widehat{R}_S(h). \tag{4.2}$$

Proposition 4.1 *For any sample S, the following inequality holds for the hypothesis returned by ERM:*

$$\mathbb{P}\left[R(h_S^{\text{ERM}}) - \inf_{h \in \mathcal{H}} R(h) > \epsilon\right] \leq \mathbb{P}\left[\sup_{h \in \mathcal{H}} |R(h) - \widehat{R}_S(h)| > \frac{\epsilon}{2}\right]. \tag{4.3}$$

Proof: By definition of $\inf_{h \in \mathcal{H}} R(h)$, for any $\epsilon > 0$, there exists h_ϵ such that $R(h_\epsilon) \leq \inf_{h \in \mathcal{H}} R(h) + \epsilon$. Thus, using $\widehat{R}_S(h_S^{\text{ERM}}) \leq \widehat{R}_S(h_\epsilon)$, which holds by the

[4] Note that, if there exists multiple hypotheses with minimal error on the training sample, then ERM returns an arbitrary one.

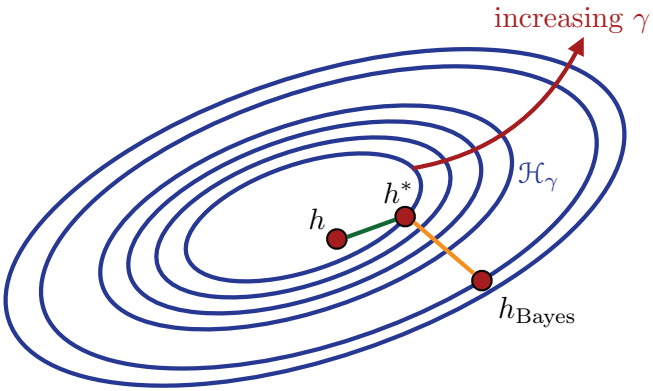


Figure 4.2
Illustration of the decomposition of a rich family $\mathcal{H} = \bigcup_{\gamma \in \Gamma} \mathcal{H}_\gamma$.

definition of the algorithm, we can write

$$\begin{aligned} R(h_S^{\text{ERM}}) - \inf_{h \in \mathcal{H}} R(h) &= R(h_S^{\text{ERM}}) - R(h_\epsilon) + R(h_\epsilon) - \inf_{h \in \mathcal{H}} R(h) \\ &\leq R(h_S^{\text{ERM}}) - R(h_\epsilon) + \epsilon \\ &= R(h_S^{\text{ERM}}) - \widehat{R}_S(h_S^{\text{ERM}}) + \widehat{R}_S(h_S^{\text{ERM}}) - R(h_\epsilon) + \epsilon \\ &\leq R(h_S^{\text{ERM}}) - \widehat{R}_S(h_S^{\text{ERM}}) + \widehat{R}_S(h_\epsilon) - R(h_\epsilon) + \epsilon \\ &\leq 2 \sup_{h \in \mathcal{H}} |R(h) - \widehat{R}_S(h)| + \epsilon. \end{aligned}$$

Since the inequality holds for all $\epsilon > 0$, it implies the following:

$$R(h_S^{\text{ERM}}) - \inf_{h \in \mathcal{H}} R(h) \leq 2 \sup_{h \in \mathcal{H}} |R(h) - \widehat{R}_S(h)|,$$

which concludes the proof. □

The right-hand side of (4.3) can be upper-bounded using the generalization bounds presented in the previous chapter in terms of the Rademacher complexity, the growth function, or the VC-dimension of $\mathcal{H}$. In particular, it can be bounded by $2e^{-2m[\epsilon - \mathfrak{R}_m(\mathcal{H})]^2}$. Thus, when $\mathcal{H}$ admits a favorable Rademacher complexity, for example a finite VC-dimension, for a sufficiently large sample, with high probability, the estimation error is guaranteed to be small. Nevertheless, the performance of ERM is typically very poor. This is because the algorithm disregards the complexity of the hypothesis set $\mathcal{H}$: in practice, either $\mathcal{H}$ is not complex enough, in which case the approximation error can be very large, or $\mathcal{H}$ is very rich, in which case the bound on the estimation error becomes very loose. Additionally, in many cases, determining the ERM solution is computationally intractable. For example, finding

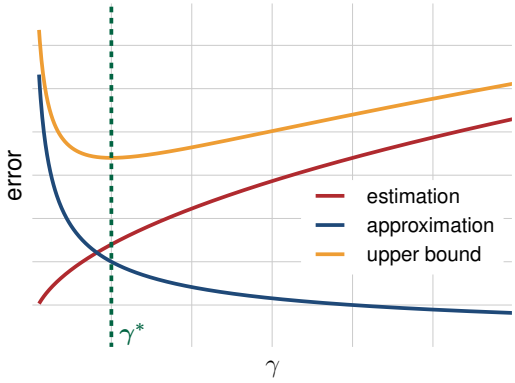


Figure 4.3
Choice of γ^* with the most favorable trade-off between estimation and approximation errors.

a linear hypothesis with the smallest error on the training sample is NP-hard, as a function of the dimension of the space.

4.3 Structural risk minimization (SRM)

In the previous section, we showed that the estimation error can be sometimes bounded or estimated. But, since the approximation error cannot be estimated, how should we choose $\mathcal{H}$? One way to proceed is to choose a very complex family $\mathcal{H}$ with no approximation error or a very small one. $\mathcal{H}$ may be too rich for generalization bounds to hold for $\mathcal{H}$, but suppose we can decompose $\mathcal{H}$ as a union of increasingly complex hypothesis sets $\mathcal{H}_\gamma$, that is $\mathcal{H} = \bigcup_{\gamma \in \Gamma} \mathcal{H}_\gamma$, with the complexity of $\mathcal{H}_\gamma$ increasing with γ, for some set Γ. Figure 4.2 illustrates this decomposition. The problem then consists of selecting the parameter $\gamma^* \in \Gamma$ and thus the hypothesis set $\mathcal{H}_{\gamma^*}$ with the most favorable trade-off between estimation and approximation errors. Since these quantities are not known, instead, as illustrated by Figure 4.3, a uniform upper bound on their sum, the excess error (also called excess risk), can be used.

This is precisely the idea behind the *Structural Risk Minimization* (SRM) method. For SRM, $\mathcal{H}$ is assumed to be decomposable into a countable set, thus, we will write its decomposition as $\mathcal{H} = \bigcup_{k \geq 1} \mathcal{H}_k$. Additionally, the hypothesis sets $\mathcal{H}_k$ are assumed to be nested: $\mathcal{H}_k \subset \mathcal{H}_{k+1}$ for all $k \geq 1$. However, many of the results presented in this section also hold for non-nested hypothesis sets. Thus, we will not make use of that assumption, unless explicitly specified. SRM consists of choosing the index $k^* \geq 1$ and the ERM hypothesis h in $\mathcal{H}_{k^*}$ that minimize an upper bound on the excess error.

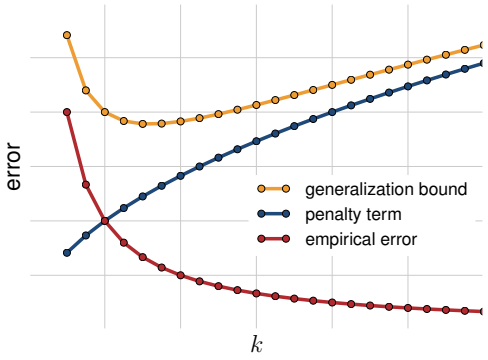


Figure 4.4
Illustration of structural risk minimization. The plots of three errors are shown as a function of the index k. Clearly, as k, or equivalently the complexity the hypothesis set $\mathcal{H}_k$, increases, the training error decreases, while the penalty term increases. SRM selects the hypothesis minimizing a bound on the generalization error, which is a sum of the empirical error and the penalty term.

As we shall see, the following learning bound holds for all $h \in \mathcal{H}$: for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample S of size m from $\mathcal{D}^m$, for all $h \in \mathcal{H}_k$ and $k \geq 1$,

$$R(h) \leq \widehat{R}_S(h) + \mathfrak{R}_m(\mathcal{H}_{k(h)}) + \sqrt{\frac{\log k}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

Thus, to minimize the resulting bound on the excess error $(R(h) - R^*)$, the index k and the hypothesis $h \in \mathcal{H}_k$ should be chosen to minimize the following objective function:

$$F_k(h) = \widehat{R}_S(h) + \mathfrak{R}_m(\mathcal{H}_k) + \sqrt{\frac{\log k}{m}}.$$

This is precisely the definition of the SRM solution h_S^{SRM}:

$$h_S^{\mathrm{SRM}} = \operatorname*{argmin}_{k \geq 1, h \in \mathcal{H}_k} F_k(h) = \operatorname*{argmin}_{k \geq 1, h \in \mathcal{H}_k} \widehat{R}_S(h) + R_m(\mathcal{H}_k) + \sqrt{\frac{\log k}{m}}. \tag{4.4}$$

Thus, SRM identifies an optimal index k^* and therefore hypothesis set $\mathcal{H}_{k^*}$, and returns the ERM solution based on that hypothesis set. Figure 4.4 further illustrates the selection of the index k^* and hypothesis set $\mathcal{H}_{k^*}$ by SRM by minimizing an upper bound on the sum of the training error and the penalty term $R_m(\mathcal{H}_k) + \sqrt{\log k/m}$. The following theorem shows that the SRM solution benefits from a strong learning guarantee. For any $h \in \mathcal{H}$, we will denote by $\mathcal{H}_{k(h)}$ the least complex hypothesis set among the $\mathcal{H}_k$s that contain h.

Theorem 4.2 (SRM Learning guarantee) *For any $\delta > 0$, with probability at least $1 - \delta$ over the draw of an i.i.d. sample S of size m from $\mathcal{D}^m$, the generalization error of the hypothesis h_S^{SRM} returned by the SRM method is bounded as follows:*

$$R(h_S^{\mathrm{SRM}}) \leq \inf_{h \in \mathcal{H}} \left(R(h) + 2\mathfrak{R}_m(\mathcal{H}_{k(h)}) + \sqrt{\frac{\log k(h)}{m}} \right) + \sqrt{\frac{2 \log \frac{3}{\delta}}{m}}.$$

Proof: Observe first that, by the union bound, the following general inequality holds:

$$\begin{aligned}
&\mathbb{P}\left[\sup_{h \in \mathcal{H}} R(h) - F_{k(h)}(h) > \epsilon\right] \\
&\qquad = \mathbb{P}\left[\sup_{k \geq 1} \sup_{h \in \mathcal{H}_k} R(h) - F_k(h) > \epsilon\right] \\
&\qquad \leq \sum_{k=1}^{\infty} \mathbb{P}\left[\sup_{h \in \mathcal{H}_k} R(h) - F_k(h) > \epsilon\right] \\
&\qquad = \sum_{k=1}^{\infty} \mathbb{P}\left[\sup_{h \in \mathcal{H}_k} R(h) - \widehat{R}_S(h) - \mathfrak{R}_m(\mathcal{H}_k) > \epsilon + \sqrt{\frac{\log k}{m}}\right] \\
&\qquad \leq \sum_{k=1}^{\infty} \exp\left(-2m\left[\epsilon + \sqrt{\tfrac{\log k}{m}}\right]^2\right) \\
&\qquad \leq \sum_{k=1}^{\infty} e^{-2m\epsilon^2} e^{-2\log k} \\
&\qquad = e^{-2m\epsilon^2} \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} e^{-2m\epsilon^2} \leq 2e^{-2m\epsilon^2}.
\end{aligned} \tag{4.5}$$

Next, for any two random variables X_1 and X_2, if $X_1 + X_2 > \epsilon$, then either X_1 or X_2 must be larger than $\epsilon/2$. In view of that, by the union bound, $\mathbb{P}[X_1 + X_2 > \epsilon] \leq \mathbb{P}[X_1 > \frac{\epsilon}{2}] + \mathbb{P}[X_2 > \frac{\epsilon}{2}]$. Using this inequality, inequality (4.5), and the inequality $F_{k(h_S^{\mathrm{SRM}})}(h_S^{\mathrm{SRM}}) \leq F_{k(h)}(h)$, which holds for all $h \in \mathcal{H}$, by definition of h_S^{SRM}, we

can write, for any $h \in \mathcal{H}$,

$$\begin{aligned}
&\mathbb{P}\left[R(h_S^{\mathrm{SRM}}) - R(h) - 2\mathfrak{R}_m(\mathcal{H}_{k(h)}) - \sqrt{\tfrac{\log k(h)}{m}} > \epsilon\right] \\
&\quad \leq \mathbb{P}\left[R(h_S^{\mathrm{SRM}}) - F_{k(h_S^{\mathrm{SRM}})}(h_S^{\mathrm{SRM}}) > \frac{\epsilon}{2}\right] \\
&\qquad + \mathbb{P}\left[F_{k(h_S^{\mathrm{SRM}})}(h_S^{\mathrm{SRM}}) - R(h) - 2\mathfrak{R}_m(\mathcal{H}_{k(h)}) - \sqrt{\tfrac{\log k(h)}{m}} > \frac{\epsilon}{2}\right] \\
&\quad \leq 2e^{-\frac{m\epsilon^2}{2}} + \mathbb{P}\left[F_{k(h)}(h) - R(h) - 2\mathfrak{R}_m(\mathcal{H}_{k(h)}) - \sqrt{\tfrac{\log k(h)}{m}} > \frac{\epsilon}{2}\right] \\
&\quad = 2e^{-\frac{m\epsilon^2}{2}} + \mathbb{P}\left[\widehat{R}_S(h) - R(h) - \mathfrak{R}_m(\mathcal{H}_{k(h)}) > \frac{\epsilon}{2}\right] \\
&\quad = 2e^{-\frac{m\epsilon^2}{2}} + e^{-\frac{m\epsilon^2}{2}} = 3e^{-\frac{m\epsilon^2}{2}}.
\end{aligned}$$

Setting the right-hand side to be equal to δ completes the proof. □

The learning guarantee just proven for SRM is remarkable. To simplify its discussion, let us assume that there exists h^* such that $R(h^*) = \inf_{h \in \mathcal{H}} R(h)$, that is, that there exists a best-in-class classifier $h^* \in \mathcal{H}$. Then, the theorem implies in particular that, with probability at least $1 - \delta$, the following inequality holds for all $h \in \mathcal{H}$:

$$R(h_S^{\mathrm{SRM}}) \leq R(h^*) + 2\mathfrak{R}_m(\mathcal{H}_{k(h^*)}) + \sqrt{\frac{\log k(h^*)}{m}} + \sqrt{\frac{2\log\frac{3}{\delta}}{m}}. \tag{4.6}$$

Observe that, remarkably, this bound is similar to the estimation error bound for $\mathcal{H}_{k(h^*)}$: it differs from it only by the term $\sqrt{\log k(h^*)/m}$. Thus, modulo that term, the guarantee for SRM is as favorable as the one we would have obtained, had an oracle informed us of the index $k(h^*)$ of the best-in-class classifier's hypothesis set.

Furthermore, observe that when $\mathcal{H}$ is rich enough that $R(h^*)$ is close to the Bayes error, the learning bound (4.6) is approximately a bound on the excess error of the SRM solution. Note that, if for some k_0, the empirical error of the ERM solution for $\mathcal{H}_{k_0}$ is zero, which holds in particular if $\mathcal{H}_{k_0}$ contains the Bayes error, then, we have $\min_{h \in \mathcal{H}_k} F_{k_0}(h) \leq \min_{h \in \mathcal{H}_k} F_k(h)$ for all $k > k_0$ and only finitely many indices need to be considered in SRM.

Assume more generally that if $\min_{h \in \mathcal{H}_k} F_k(h) \leq \min_{h \in \mathcal{H}_{k+1}} F_k(h)$ for some k, then indices beyond $k+1$ need not be inspected. This property may hold for example if the empirical error cannot be further improved after some index k. In that case, the minimizing index k^* can be determined via a binary search in the interval $[1, k_{\max}]$, given some maximum value $k_{\max}$. $k_{\max}$ itself can be found by inspecting $\min_{h \in \mathcal{H}_{2^n}} F_k(h)$ for exponentially growing indices 2^n, $n \geq 1$, and setting $k_{\max} = 2^n$ for n such that $\min_{h \in \mathcal{H}_{2^n}} F_k(h) \leq \min_{h \in \mathcal{H}_{2^{n+1}}} F_k(h)$. The number of ERM computations needed to find $k_{\max}$ is in $O(n) = O(\log k_{\max})$ and similarly the

number of ERM computations due to the binary search is in $O(\log k_{\max})$. Thus, if n is the smallest integer such that $k^* < 2^n$, the overall number of ERM computations is in $O(\log k^*)$.

While it benefits from a very favorable guarantee, SRM admits several drawbacks. First, the decomposability of $\mathcal{H}$ into countably many hypothesis sets, each with a converging Rademacher complexity, remains a strong assumption. As an example, the family of all measurable functions cannot be written as a union of countably many hypothesis sets with finite VC-dimension. Thus, the choice of $\mathcal{H}$ or that of the hypothesis sets $\mathcal{H}_k$ is a key component of SRM. Second, and this is the main disadvantage of SRM, the method is typically computationally intractable: for most hypothesis sets, finding the solution of ERM is NP-hard and in general SRM requires determining that solution for a large number of indices k.

4.4 Cross-validation

An alternative method for model selection, *cross-validation*, consists of using some fraction of the training sample as a *validation set* to select a hypothesis set $\mathcal{H}_k$. This is in contrast with the SRM model which relies on a theoretical learning bound assigning a penalty to each hypothesis set. In this section, we analyze the cross-validation method and compare its performance to that of SRM.

As in the previous section, let $(\mathcal{H}_k)_{k\geq 1}$ be a countable sequence of hypothesis sets with increasing complexities. The cross-validation (CV) solution is obtained as follows. Let S be an i.i.d. labeled sample of size m. S is divided into a sample S_1 of size $(1-\alpha)m$ and a sample S_2 of size αm, with $\alpha \in (0,1)$ typically chosen to be relatively small. S_1 is reserved for training, S_2 for validation. For any $k \in \mathbb{N}$, let $h^{\mathrm{ERM}}_{S_1,k}$ denote the solution of ERM run on S_1 using the hypothesis set $\mathcal{H}_k$. The hypothesis h^{CV}_S returned by cross-validation is the ERM solution $h^{\mathrm{ERM}}_{S_1,k}$ with the best performance on S_2:

$$h^{\mathrm{CV}}_S = \operatorname*{argmin}_{h \in \left\{h^{\mathrm{ERM}}_{S_1,k} \colon k \geq 1\right\}} \widehat{R}_{S_2}(h). \tag{4.7}$$

The following general result will help us derive learning guarantees for cross-validation.

Proposition 4.3 *For any $\alpha > 0$ and any sample size $m \geq 1$, the following general inequality holds:*

$$\mathbb{P}\left[\sup_{k\geq 1}\left|R(h^{\mathrm{ERM}}_{S_1,k}) - \widehat{R}_{S_2}(h^{\mathrm{ERM}}_{S_1,k})\right| > \epsilon + \sqrt{\frac{\log k}{\alpha m}}\right] \leq 4e^{-2\alpha m\epsilon^2}.$$

Proof: By the union bound, we can write

$$\begin{aligned}\mathbb{P}\left[\sup_{k\geq 1}\left|R(h_{S_1,k}^{\mathrm{ERM}})-\widehat{R}_{S_2}(h_{S_1,k}^{\mathrm{ERM}})\right|>\epsilon+\sqrt{\frac{\log k}{\alpha m}}\right]&\\ \leq\sum_{k=1}^{\infty}\mathbb{P}\left[\left|R(h_{S_1,k}^{\mathrm{ERM}})-\widehat{R}_{S_2}(h_{S_1,k}^{\mathrm{ERM}})\right|>\epsilon+\sqrt{\frac{\log k}{\alpha m}}\right]&\\ =\sum_{k=1}^{\infty}\mathbb{E}\left[\mathbb{P}\left[\left|R(h_{S_1,k}^{\mathrm{ERM}})-\widehat{R}_{S_2}(h_{S_1,k}^{\mathrm{ERM}})\right|>\epsilon+\sqrt{\frac{\log k}{\alpha m}}\;\middle|\;S_1\right]\right].&\end{aligned}\tag{4.8}$$

The hypothesis $h_{S_1,k}^{\mathrm{ERM}}$ is fixed conditioned on S_1. Furthermore, the sample S_2 is independent from S_1. Therefore, by Hoeffding's inequality, we can bound the conditional probability as follows:

$$\begin{aligned}\mathbb{P}\left[\left|R(h_{S_1,k}^{\mathrm{ERM}})-\widehat{R}_{S_2}(h_{S_1,k}^{\mathrm{ERM}})\right|>\epsilon+\sqrt{\frac{\log k}{\alpha m}}\;\middle|\;S_1\right]&\leq 2e^{-2\alpha m\left(\epsilon+\sqrt{\frac{\log k}{\alpha m}}\right)^2}.\\ &\leq 2e^{-2\alpha m\epsilon^2-2\log k}\\ &=\frac{2}{k^2}e^{-2\alpha m\epsilon^2}.\end{aligned}$$

Plugging in the right-hand side of this bound in (4.8) and summing over k yields

$$\mathbb{P}\left[\sup_{k\geq 1}\left|R(h_{S_1,k}^{\mathrm{ERM}})-\widehat{R}_{S_2}(h_{S_1,k}^{\mathrm{ERM}})\right|>\epsilon+\sqrt{\frac{\log k}{\alpha m}}\right]\leq\frac{\pi^2}{3}e^{-2\alpha m\epsilon^2}<4e^{-2\alpha m\epsilon^2},$$

which completes the proof. □

Let $R(h_{S_1}^{\mathrm{SRM}})$ be the generalization error of the SRM solution using a sample S_1 of size $(1-\alpha m)$ and $R(h_S^{\mathrm{CV}},S)$ the generalization error of the cross-validation solution using a sample S of size m. Then, using Proposition 4.3, the following learning guarantee can be derived which compares the error of the CV method to that of SRM.

Theorem 4.4 (Cross-validation versus SRM) *For any $\delta>0$, with probability at least $1-\delta$, the following holds:*

$$R(h_S^{\mathrm{CV}})-R(h_{S_1}^{\mathrm{SRM}})\leq 2\sqrt{\frac{\log\max(k(h_S^{\mathrm{CV}}),k(h_{S_1}^{\mathrm{SRM}}))}{\alpha m}}+2\sqrt{\frac{\log\frac{4}{\delta}}{2\alpha m}},$$

where, for any h, $k(h)$ denotes the smallest index of a hypothesis set containing h.

Proof: By Proposition 4.3 and Theorem 4.2, using the property of h_S^{CV} as a minimizer, for any $\delta>0$, with probability at least $1-\delta$, the following inequalities

hold:

$$\begin{aligned}
R(h_S^{\mathrm{CV}}) &\leq \widehat{R}_{S_2}(h_S^{\mathrm{CV}}) + \sqrt{\frac{\log(k(h_S^{\mathrm{CV}}))}{\alpha m}} + \sqrt{\frac{\log\frac{4}{\delta}}{2\alpha m}} \\
&\leq \widehat{R}_{S_2}(h_{S_1}^{\mathrm{SRM}}) + \sqrt{\frac{\log(k(h_S^{\mathrm{CV}}))}{\alpha m}} + \sqrt{\frac{\log\frac{4}{\delta}}{2\alpha m}} \\
&\leq R(h_{S_1}^{\mathrm{SRM}}) + \sqrt{\frac{\log(k(h_S^{\mathrm{CV}}))}{\alpha m}} + \sqrt{\frac{\log(k(h_{S_1}^{\mathrm{SRM}}))}{\alpha m}} + 2\sqrt{\frac{\log\frac{4}{\delta}}{2\alpha m}} \\
&\leq R(h_{S_1}^{\mathrm{SRM}}) + 2\sqrt{\frac{\log(\max(k(h_S^{\mathrm{CV}}), k(h_{S_1}^{\mathrm{SRM}}))}{\alpha m}} + 2\sqrt{\frac{\log\frac{4}{\delta}}{2\alpha m}},
\end{aligned}$$

which completes the proof. □

The learning guarantee just proven shows that, with high probability, the generalization error of the CV solution for a sample of size m is close to that of the SRM solution for a sample of size $(1-\alpha)m$. For α relatively small, this suggests a guarantee similar to that of SRM, which, as previously discussed, is very favorable. However, in some unfavorable regimes, an algorithm (here SRM) trained on $(1-\alpha)m$ points may have a significantly worse performance than when trained on m points (avoiding this phase transition issue is one of the main motivations behind the use of the n-fold cross-validation method in practice, see section 4.5). Thus, the bound suggests in fact a trade-off: α should be chosen sufficiently small to avoid the unfavorable regimes just mentioned and yet sufficiently large for the right-hand side of the bound to be small and thus informative.

The learning bound for CV can be made more explicit in some cases in practice. Assume for example that the hypothesis sets $\mathcal{H}_k$ are nested and that the empirical errors of the ERM solutions $h_{S_1,k}^{\mathrm{ERM}}$ are decreasing before reaching zero: for any k, $\widehat{R}_{S_1}(h_{S_1,k+1}^{\mathrm{ERM}}) < \widehat{R}_{S_1}(h_{S_1,k}^{\mathrm{ERM}})$ for all k such that $\widehat{R}_{S_1}(h_{S_1,k}^{\mathrm{ERM}}) > 0$ and $\widehat{R}_{S_1}(h_{S_1,k+1}^{\mathrm{ERM}}) \leq \widehat{R}_{S_1}(h_{S_1,k}^{\mathrm{ERM}})$ otherwise. Observe that $\widehat{R}_{S_1}(h_{S_1,k}^{\mathrm{ERM}}) > 0$ implies at least one error for $h_{S_1,k}^{\mathrm{ERM}}$, therefore $\widehat{R}_{S_1}(h_{S_1,k}^{\mathrm{ERM}}) > \frac{1}{m}$. In view of that, we must then have $\widehat{R}_{S_1}(h_{S_1,n}^{\mathrm{ERM}}) = 0$ for all $n \geq m+1$. Thus, we have $h_{S_1,n}^{\mathrm{ERM}} = h_{S_1,m+1}^{\mathrm{ERM}}$ for all $n \geq m+1$ and we can assume that $k(f_{CV}) \leq m+1$. Since the complexity of $\mathcal{H}_k$ increases with k we also have $k(f_{SRM}) \leq m+1$. In view of that, we obtain the following more explicit learning bound for cross-validation:

$$R(f_{CV}, S) - R(f_{SRM}, S_1) \leq 2\sqrt{\frac{\log(\frac{4}{\delta})}{2\alpha m}} + 2\sqrt{\frac{\log(m+1)}{\alpha m}}.$$

4.5 n-Fold cross-validation

In practice, the amount of labeled data available is often too small to set aside a validation sample since that would leave an insufficient amount of training data. Instead, a widely adopted method known as *n-fold cross-validation* is used to exploit the labeled data both for *model selection* and for training.

Let $\boldsymbol{\theta}$ denote the vector of free parameters of the algorithm. For a fixed value of $\boldsymbol{\theta}$, the method consists of first randomly partitioning a given sample S of m labeled examples into n subsamples, or folds. The ith fold is thus a labeled sample $((x_{i1}, y_{i1}), \ldots, (x_{im_i}, y_{im_i}))$ of size m_i. Then, for any $i \in [n]$, the learning algorithm is trained on all but the ith fold to generate a hypothesis h_i, and the performance of h_i is tested on the ith fold, as illustrated in figure 4.5a. The parameter value $\boldsymbol{\theta}$ is evaluated based on the average error of the hypotheses h_i, which is called the *cross-validation error*. This quantity is denoted by $\widehat{R}_{\mathrm{CV}}(\boldsymbol{\theta})$ and defined by

$$\widehat{R}_{\mathrm{CV}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\frac{1}{m_i} \sum_{j=1}^{m_i} L(h_i(x_{ij}), y_{ij})}_{\text{error of } h_i \text{ on the } i\text{th fold}} .$$

The folds are generally chosen to have equal size, that is $m_i = m/n$ for all $i \in [n]$. How should n be chosen? The appropriate choice is subject to a trade-off. For a large n, each training sample used in n-fold cross-validation has size $m - m/n = m(1 - 1/n)$ (illustrated by the right vertical red line in figure 4.5b), which is close to m, the size of the full sample, and also implies all training samples are quite similar. At the same time, the ith fold used to measure the error is relatively small and thus the cross-validation error tends to have a small bias but a large variance. In contrast, smaller values of n lead to more diverse training samples but their size (shown by the left vertical red line in figure 4.5b) is significantly less than m. In this regime, the ith fold is relatively large and thus the cross-validation error tends to have a smaller variance but a larger bias.

In applications, n is typically chosen to be 5 or 10. n-fold cross-validation is used as follows in model selection. The full labeled data is first split into a training and a test sample. The training sample of size m is then used to compute the n-fold cross-validation error $\widehat{R}_{\mathrm{CV}}(\boldsymbol{\theta})$ for a small number of possible values of $\boldsymbol{\theta}$. The free parameter $\boldsymbol{\theta}$ is next set to the value $\boldsymbol{\theta}_0$ for which $\widehat{R}_{\mathrm{CV}}(\boldsymbol{\theta})$ is smallest and the algorithm is trained with the parameter setting $\boldsymbol{\theta}_0$ over the full training sample of size m. Its performance is evaluated on the test sample as already described in the previous section.

The special case of n-fold cross-validation where $n = m$ is called *leave-one-out cross-validation*, since at each iteration exactly one instance is left out of the train-

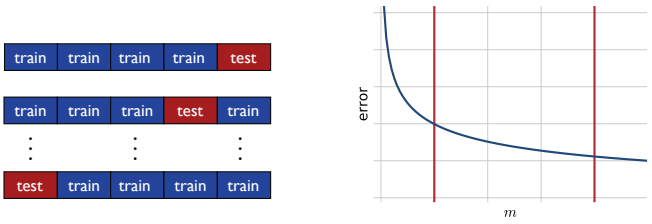


Figure 4.5
n-fold cross-validation. (a) Illustration of the partitioning of the training data into 5 folds. (b) Typical plot of a classifier's prediction error as a function of the size of the training sample m: the error decreases as a function of the number of training points. The red line on the left side marks the region for small values of n, while the red line on the right side marks the region for large values of n.

ing sample. As shown in chapter 5, the average leave-one-out error is an approximately unbiased estimate of the average error of an algorithm and can be used to derive simple guarantees for some algorithms. In general, the leave-one-out error is very costly to compute, since it requires training m times on samples of size $m-1$, but for some algorithms it admits a very efficient computation (see exercise 11.9).

In addition to model selection, n-fold cross-validation is also commonly used for performance evaluation. In that case, for a fixed parameter setting $\boldsymbol{\theta}$, the full labeled sample is divided into n random folds with no distinction between training and test samples. The performance reported is the n-fold cross-validation error on the full sample as well as the standard deviation of the errors measured on each fold.

4.6 Regularization-based algorithms

A broad family of algorithms inspired by the SRM method is that of *regularization-based algorithm*. This consists of selecting a very complex family $\mathcal{H}$ that is an uncountable union of nested hypothesis sets $\mathcal{H}_\gamma$: $\mathcal{H} = \bigcup_{\gamma>0} \mathcal{H}_\gamma$. $\mathcal{H}$ is often chosen to be dense in the space of continuous functions over $\mathcal{X}$. For example, $\mathcal{H}$ may be chosen to be the set of all linear functions in some high-dimensional space and $\mathcal{H}_\gamma$ the subset of those functions whose norm is bounded by γ: $\mathcal{H}_\gamma = \{x \mapsto \mathbf{w} \cdot \boldsymbol{\Phi}(x) \colon \|\mathbf{w}\| \leq \gamma\}$. For some choices of $\boldsymbol{\Phi}$ and the high-dimensional space, it can be shown that $\mathcal{H}$ is indeed dense in the space of continuous functions over $\mathcal{X}$.

Given a labeled sample S, the extension of the SRM method to an uncountable union would then suggest selecting h based on the following optimization problem:

$$\operatorname*{argmin}_{\gamma>0, h\in H_\gamma} \widehat{R}_S(h) + \mathfrak{R}_m(\mathcal{H}_\gamma) + \sqrt{\frac{\log \gamma}{m}},$$

where other penalty terms $\mathrm{pen}(\gamma, m)$ can be chosen in lieu of the specific choice $\mathrm{pen}(\gamma, m) = \mathfrak{R}_m(\mathcal{H}_\gamma) + \sqrt{\frac{\log \gamma}{m}}$. Often, there exists a function $\mathcal{R}\colon \mathcal{H} \to \mathbb{R}$ such that, for any $\gamma > 0$, the constrained optimization problem $\operatorname{argmin}_{\gamma>0, h\in H_\gamma} \widehat{R}_S(h) + \mathrm{pen}(\gamma, m)$ can be equivalently written as the unconstrained optimization problem

$$\operatorname*{argmin}_{h\in\mathcal{H}} \widehat{R}_S(h) + \lambda \mathcal{R}(h),$$

for some $\lambda > 0$. $\mathcal{R}(h)$ is called a *regularization term* and $\lambda > 0$ is treated as a hyperparameter since its optimal value is often not known. For most algorithms, the regularization term $\mathcal{R}(h)$ is chosen to be an increasing function of $\|h\|$ for some choice of the norm $\|\cdot\|$, when $\mathcal{H}$ is the subset of a Hilbert space. The variable λ is often called a *regularization parameter*. Larger values of λ further penalize more complex hypotheses, while, for λ close or equal to zero, the regularization term has no effect and the algorithm coincides with ERM. In practice, λ is typically selected via cross-validation or using n-fold cross-validation.

When the regularization term is chosen to be $\|h\|_p$ for some choice of the norm and $p \geq 1$, then it is a convex function of h, since any norm is convex. However, for the zero-one loss, the first term of the objective function is non-convex, thereby making the optimization problem computationally hard. In practice, most regularization-based algorithms instead use a convex upper bound on the zero-one loss and replace the empirical zero-one term with the empirical value of that convex surrogate. The resulting optimization problem is then convex and therefore admits more efficient solutions than SRM. The next section studies the properties of such convex surrogate losses.

4.7 Convex surrogate losses

The guarantees for the estimation error that we presented in previous sections hold either for ERM or for SRM, which itself is defined in terms of ERM. However, as already mentioned, for many choices of the hypothesis set $\mathcal{H}$, including that of linear functions, solving the ERM optimization problem is NP-hard mainly because the zero-one loss function is not convex. One common method for addressing this problem consists of using a convex surrogate loss function that upper bounds the zero-one loss. This section analyzes learning guarantees for such surrogate losses in terms of the original loss.

The hypotheses we consider are real-valued functions $h\colon \mathcal{X} \to \mathbb{R}$. The sign of h defines a binary classifier $f_h\colon \mathcal{X} \to \{-1, +1\}$ defined for all $x \in \mathcal{X}$ by

$$f_h(x) = \begin{cases} +1 & \text{if } h(x) \geq 0 \\ -1 & \text{if } h(x) < 0. \end{cases}$$

The loss or error of h at point $(x, y) \in \mathcal{X} \times \{-1, +1\}$ is defined as the binary classification error of f_h:

$$1_{f_h(x) \neq y} = 1_{yh(x) < 0} + 1_{h(x)=0 \wedge y=-1} \leq 1_{yh(x) \leq 0}.$$

We will denote by $R(h)$ the expected error of h: $R(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[1_{f_h(x)\neq y}\right]$. For any $x \in \mathcal{X}$, let $\eta(x)$ denote $\eta(x) = \mathbb{P}[y = +1|x]$ and let $\mathcal{D}_{\mathcal{X}}$ denote the marginal distribution over $\mathcal{X}$. Then, for any h, we can write

$$\begin{aligned} R(h) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[1_{f_h(x)\neq y}\right] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\eta(x)1_{h(x)<0} + (1-\eta(x))1_{h(x)>0} + (1-\eta(x))1_{h(x)=0}\right] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\eta(x)1_{h(x)<0} + (1-\eta(x))1_{h(x)\geq 0}\right]. \end{aligned}$$

In view of that, the Bayes classifier can be defined as assigning label $+1$ to x when $\eta(x) \geq \frac{1}{2}$, -1 otherwise. It can therefore be induced by the function h^* defined by

$$h^*(x) = \eta(x) - \tfrac{1}{2}. \tag{4.9}$$

We will refer to $h^*\colon \mathcal{X} \to \mathbb{R}$ as the *Bayes scoring function* and will denote by R^* the error of the Bayes classifier or Bayes scoring function: $R^* = R(h^*)$.

Lemma 4.5 *The excess error of any hypothesis $h\colon \mathcal{X} \to \mathbb{R}$ can be expressed as follows in terms of η and the Bayes scoring function h^*:*

$$R(h) - R^* = 2 \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[|h^*(x)|\, 1_{h(x)h^*(x)\leq 0}\right].$$

Proof: For any h, we can write

$$\begin{aligned} R(h) &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\eta(x)1_{h(x)<0} + (1-\eta(x))1_{h(x)\geq 0}\right] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\eta(x)1_{h(x)<0} + (1-\eta(x))(1-1_{h(x)<0})\right] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[[2\eta(x)-1]1_{h(x)<0} + (1-\eta(x))\right] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[2h^*(x)1_{h(x)<0} + (1-\eta(x))\right], \end{aligned}$$

where we used for the last step equation (4.9). In view of that, for any h, the following holds:

$$\begin{aligned} R(h) - R(h^*) &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\Big[2[h^*(x)](1_{h(x)\leq 0} - 1_{h^*(x)\leq 0})\Big] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\Big[2[h^*(x)]\operatorname{sgn}(h^*(x))1_{(h(x)h^*(x)\leq 0)\wedge((h(x),h^*(x))\neq(0,0))}\Big] \\ &= 2\mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\Big[|h^*(x)|\,1_{h(x)h^*(x)\leq 0}\Big], \end{aligned}$$

which completes the proof, since $R(h^*) = R^*$. □

Let $\Phi\colon \mathbb{R}\to\mathbb{R}$ be a convex and non-decreasing function so that for any $u\in\mathbb{R}$, $1_{u\leq 0}\leq \Phi(-u)$. The Φ-loss of a function $h\colon \mathcal{X}\to\mathbb{R}$ at point $(x,y)\in\mathcal{X}\times\{-1,+1\}$ is defined as $\Phi(-yh(x))$ and its expected loss given by

$$\begin{aligned} \mathcal{L}_\Phi(h) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[\Phi(-yh(x))\big] \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\Big[\eta(x)\Phi(-h(x)) + (1-\eta(x))\Phi(h(x))\Big]. \end{aligned} \tag{4.10}$$

Notice that since $1_{yh(x)\leq 0}\leq\Phi(-yh(x))$, we have $R(h)\leq\mathcal{L}_\Phi(h)$. For any $x\in\mathcal{X}$, let $u\mapsto L_\Phi(x,u)$ be the function defined for all $u\in\mathbb{R}$ by

$$L_\Phi(x,u) = \eta(x)\Phi(-u) + (1-\eta(x))\Phi(u).$$

Then, $\mathcal{L}_\Phi(h) = \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[L_\Phi(x,h(x))]$. Since Φ is convex, $u\mapsto L_\Phi(x,u)$ is convex as a sum of two convex functions. Define $h^*_\Phi\colon\mathcal{X}\to[-\infty,+\infty]$ as the *Bayes solution for the loss function* L_Φ. That is, for any x, $h^*_\Phi(x)$ is a solution of the following convex optimization problem:

$$\begin{aligned} h^*_\Phi(x) &= \mathop{\operatorname{argmin}}_{u\in[-\infty,+\infty]} L_\Phi(x,u) \\ &= \mathop{\operatorname{argmin}}_{u\in[-\infty,+\infty]} \eta(x)\Phi(-u) + (1-\eta(x))\Phi(u). \end{aligned}$$

The solution of this optimization is in general not unique. When $\eta(x) = 0$, $h^*_\Phi(x)$ is a minimizer of $u\mapsto\Phi(u)$ and since Φ is non-decreasing, we can choose $h^*_\Phi(x) = -\infty$ in that case. Similarly, when $\eta(x) = 1$, we can choose $h^*_\Phi(x) = +\infty$. When $\eta(x) = \frac{1}{2}$, $L_\Phi(x,u) = \frac{1}{2}[\Phi(-u)+\Phi(u)]$, thus, by convexity, $L_\Phi(x,u)\geq\Phi(-\frac{u}{2}+\frac{u}{2}) = \Phi(0)$. Thus, we can choose $h^*_\Phi(x) = 0$ in that case. For all other values of $\eta(x)$, in case of non-uniqueness, an arbitrary minimizer is chosen in this definition. We will denote by $\mathcal{L}^*_\Phi$ the Φ-loss of h^*_Φ: $\mathcal{L}^*_\Phi = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\Phi(-yh^*_\Phi(x))\big]$.

Proposition 4.6 *Let Φ be a convex and non-decreasing function that is differentiable at 0 with $\Phi'(0) > 0$. Then, the minimizer of Φ defines the Bayes classifier: for any $x\in\mathcal{X}$, $h^*_\Phi(x) > 0$ iff $h^*(x) > 0$ and $h^*(x) = 0$ iff $h^*_\Phi(x) = 0$, which implies $\mathcal{L}^*_\Phi = R^*$.*

Proof: Fix $x \in \mathcal{X}$. If $\eta(x) = 0$, then $h^*(x) = -\frac{1}{2}$ and $h^*_\Phi(x) = -\infty$, thus $h^*(x)$ and $h^*_\Phi(x)$ admit the same sign. Similarly, if $\eta(x) = 1$, then $h^*(x) = +\frac{1}{2}$ and $h^*_\Phi(x) = +\infty$, and $h^*(x)$ and $h^*_\Phi(x)$ admit the same sign.

Let u^* denote the minimizer defining $h^*_\Phi(x)$. u^* is a minimizer of $u \mapsto L_\Phi(x, u)$ iff the subdifferential of that function at u^* contains 0, that is, since $\partial L_\Phi(x, u^*) = -\eta(x)\partial\Phi(-u^*) + (1 - \eta(x))\partial\Phi(u^*)$, iff there exist $v_1 \in \partial\Phi(-u^*)$ and $v_2 \in \partial\Phi(u^*)$ such that

$$\eta(x)v_1 = (1 - \eta(x))v_2. \tag{4.11}$$

If $u^* = 0$, by the differentiability of Φ at 0 we have $v_1 = v_2 = \Phi'(0) > 0$ and thus $\eta(x) = \frac{1}{2}$, that is $h^*(x) = 0$. Conversely, If $h^*(x) = 0$, that is $\eta(x) = \frac{1}{2}$, then, by definition, we have $h^*_\Phi(x) = 0$. Thus, $h^*(x) = 0$ iff $h^*_\Phi(x) = 0$ iff $\eta(x) = \frac{1}{2}$.

We can assume now that $\eta(x)$ is not in $\{0, 1, \frac{1}{2}\}$. We first show that for any $u_1, u_2 \in \mathbb{R}$ with $u_1 < u_2$, and any two choices of the subgradients at u_1 and u_2, $v_1 \in \partial\Phi(u_1)$ and $v_2 \in \partial\Phi(u_2)$, we have $v_1 \leq v_2$. By definition of the subgradients at u_1 and u_2, the following inequalities hold:

$$\Phi(u_2) - \Phi(u_1) \geq v_1(u_2 - u_1) \qquad \Phi(u_1) - \Phi(u_2) \geq v_2(u_1 - u_2).$$

Summing up these inequalities yields $v_2(u_2 - u_1) \geq v_1(u_2 - u_1)$ and thus $v_2 \geq v_1$, since $u_1 < u_2$.

Now, if $u^* > 0$, then we have $-u^* < u^*$. By the property shown above, this implies $v_1 \leq v_2$. We cannot have $v_1 = v_2 \neq 0$ since (4.11) would then imply $\eta(x) = \frac{1}{2}$. We also cannot have $v_1 = v_2 = 0$ since by the property shown above, we must have $\Phi'(0) \leq v_2$ and thus $v_2 > 0$. Thus, we must have $v_1 < v_2$ with $v_2 > 0$, which, by (4.11), implies $\eta(x) > 1 - \eta(x)$, that is $h^*(x) > 0$.

Conversely, if $h^*(x) > 0$ then $\eta(x) > 1 - \eta(x)$. We cannot have $v_1 = v_2 = 0$ or $v_1 = v_2 \neq 0$ as already shown. Thus, since $\eta(x) \neq 1$, by (4.11), this implies $v_1 < v_2$. We cannot have $u^* < -u^*$ since, by the property shown above, this would imply $v_2 \leq v_1$. Thus, we must have $-u^* \leq u^*$, that is $u^* \geq 0$, and more specifically $u^* > 0$ since, as already shown above, $u^* = 0$ implies $h^*(x) = 0$. □

Theorem 4.7 *Let Φ be a convex and non-decreasing function. Assume that there exists $s \geq 1$ and $c > 0$ such that the following holds for all $x \in \mathcal{X}$:*

$$|h^*(x)|^s = \left|\eta(x) - \tfrac{1}{2}\right|^s \leq c^s \left[L_\Phi(x, 0) - L_\Phi(x, h^*_\Phi(x))\right].$$

Then, for any hypothesis h, the excess error of h is bounded as follows:

$$R(h) - R^* \leq 2c \left[\mathcal{L}_\Phi(h) - \mathcal{L}^*_\Phi\right]^{\frac{1}{s}}$$

Proof: We will use the following inequality which holds by the convexity of Φ:

$$\begin{aligned}\Phi\big(-2h^*(x)h(x)\big) &= \Phi\big((1-2\eta(x))h(x)\big)\\ &= \Phi\big(\eta(x)(-h(x)) + (1-\eta(x))h(x)\big)\\ &\leq \eta(x)\Phi((-h(x))) + (1-\eta(x))\Phi(h(x)) = L_\Phi(x, h(x)). \quad (4.12)\end{aligned}$$

By Lemma 4.5, Jensen's inequality, and $h^*(x) = \eta(x) - \frac{1}{2}$, we can write

$$\begin{aligned}&R(h) - R(h^*)\\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\left[|2\eta(x)-1|\,1_{h(x)h^*(x)\leq 0}\right]\\ &\leq \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\left[|2\eta(x)-1|^s\,1_{h(x)h^*(x)\leq 0}\right]^{\frac{1}{s}} && \text{(Jensen's ineq.)}\\ &\leq 2c \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\left[\left[\Phi(0) - L_\Phi(x, h^*_\Phi(x))\right]1_{h(x)h^*(x)\leq 0}\right]^{\frac{1}{s}} && \text{(assumption)}\\ &\leq 2c \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\left[\left[\Phi\big(-2h^*(x)h(x)\big) - L_\Phi(x, h^*_\Phi(x))\right]1_{h(x)h^*(x)\leq 0}\right]^{\frac{1}{s}} && (\Phi \text{ non-decreasing})\\ &\leq 2c \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\left[\left[L_\Phi(x, h(x)) - L_\Phi(x, h^*_\Phi(x))\right]1_{h(x)h^*(x)\leq 0}\right]^{\frac{1}{s}} && \text{(convexity ineq. (4.12))}\\ &\leq 2c \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\mathcal{X}}\left[L_\Phi(x, h(x)) - L_\Phi(x, h^*_\Phi(x))\right]^{\frac{1}{s}},\end{aligned}$$

which completes the proof, since $\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[L_\Phi(x, h^*_\Phi(x))] = L^*_\Phi$. □

The theorem shows that, when the assumption holds, the excess error of h can be upper bounded in terms of the excess Φ-loss. The assumption of the theorem holds in particular for the following convex loss functions:

- Hinge loss, where $\Phi(u) = \max(0, 1+u)$, with $s = 1$ and $c = \frac{1}{2}$.
- Exponential loss, where $\Phi(u) = \exp(u)$, with $s = 2$ and $c = \frac{1}{\sqrt{2}}$.
- Logistic loss, where $\Phi(u) = \log_2(1+e^u)$, with $s = 2$ and $c = \frac{1}{\sqrt{2}}$.

They also hold for the square loss and the squared Hinge loss (see Exercises 4.2 and 4.3).

4.8 Chapter notes

The structural risk minimization (SRM) technique is due to Vapnik [1998]. The original penalty term used by Vapnik [1998] is based on the VC-dimension of the hypothesis set. The version of SRM with Rademacher complexity-based penalties that we present here leads to finer data-dependent learning guarantees. Penalties based on alternative complexity measures can be used similarly leading to learning bounds in terms of the corresponding complexity measure [Bartlett et al., 2002a].

An alternative model selection theory of *Voted Risk Minimization* (VRM) has been recently developed by Cortes, Mohri, and Syed [2014] and other related publications [Kuznetsov et al., 2014, DeSalvo et al., 2015, Cortes et al., 2015].

Theorem 4.7 is due to Zhang [2003a]. The proof given here is somewhat different and simpler.

4.9 Exercises

4.1 For any hypothesis set $\mathcal{H}$, show that the following inequalities hold:

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\widehat{R}_S\left(h_S^{\mathrm{ERM}}\right)\right] \leq \inf_{h\in\mathcal{H}} R(h) \leq \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[R\left(h_S^{\mathrm{ERM}}\right)\right]. \tag{4.13}$$

4.2 Show that for the squared loss, $\Phi(u) = (1+u)^2$, the statement of Theorem 4.7 holds with $s = 2$ and $c = \frac{1}{2}$ and therefore that the excess error can be upper bounded as follows:

$$R(h) - R^* \leq \left[\mathcal{L}_\Phi(h) - \mathcal{L}_\Phi^*\right]^{\frac{1}{2}}.$$

4.3 Show that for the squared Hinge loss, $\Phi(u) = \max(0, 1+u)^2$, the statement of Theorem 4.7 holds with $s = 2$ and $c = \frac{1}{2}$ and therefore that the excess error can be upper bounded as follows:

$$R(h) - R^* \leq \left[\mathcal{L}_\Phi(h) - \mathcal{L}_\Phi^*\right]^{\frac{1}{2}}.$$

4.4 In this problem, the loss of $h\colon \mathcal{X} \to \mathbb{R}$ at point $(x, y) \in \mathcal{X} \times \{-1, +1\}$ is defined to be $1_{yh(x)\leq 0}$.

(a) Define the Bayes classifier and a Bayes scoring function h^* for this loss.

(b) Express the excess error of h in terms of h^* (counterpart of Lemma 4.5, for loss considered here).

(c) Give a counterpart of the result of Theorem 4.7 for this loss.

4.5 Same questions as in Exercise 4.5 with the loss of $h\colon \mathcal{X} \to \mathbb{R}$ at point $(x, y) \in \mathcal{X} \times \{-1, +1\}$ defined instead to be $1_{yh(x)<0}$.

5 Support Vector Machines

This chapter presents one of the most theoretically well motivated and practically most effective classification algorithms in modern machine learning: Support Vector Machines (SVMs). We first introduce the algorithm for separable datasets, then present its general version designed for non-separable datasets, and finally provide a theoretical foundation for SVMs based on the notion of margin. We start with the description of the problem of linear classification.

5.1 Linear classification

Consider an input space $\mathcal{X}$ that is a subset of $\mathbb{R}^N$ with $N \geq 1$, and the output or target space $\mathcal{Y} = \{-1, +1\}$, and let $f\colon \mathcal{X} \to \mathcal{Y}$ be the target function. Given a hypothesis set $\mathcal{H}$ of functions mapping $\mathcal{X}$ to $\mathcal{Y}$, the binary classification task is formulated as follows. The learner receives a training sample S of size m drawn i.i.d. from $\mathcal{X}$ according to some unknown distribution $\mathcal{D}$, $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$, with $y_i = f(x_i)$ for all $i \in [m]$. The problem consists of determining a hypothesis $h \in \mathcal{H}$, a *binary classifier*, with small generalization error:

$$R_{\mathcal{D}}(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)]. \tag{5.1}$$

Different hypothesis sets $\mathcal{H}$ can be selected for this task. In view of the results presented in chapter 3, which formalized Occam's razor principle, hypothesis sets with smaller complexity — e.g., smaller VC-dimension or Rademacher complexity — provide better learning guarantees, everything else being equal. A natural hypothesis set with relatively small complexity is that of *linear classifiers*, or hyperplanes, which can be defined as follows:

$$\mathcal{H} = \{\mathbf{x} \mapsto \operatorname{sign}(\mathbf{w} \cdot \mathbf{x} + b)\colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}. \tag{5.2}$$

The learning problem is then referred to as a *linear classification problem*. The general equation of a hyperplane in $\mathbb{R}^N$ is $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w} \in \mathbb{R}^N$ is a

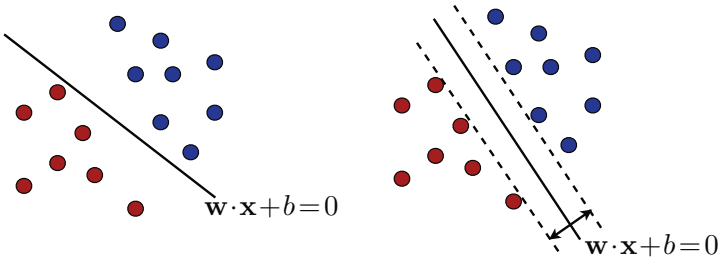


Figure 5.1
Two possible separating hyperplanes. The right-hand side figure shows a hyperplane that maximizes the margin.

non-zero vector normal to the hyperplane and $b \in \mathbb{R}$ a scalar. A hypothesis of the form $\mathbf{x} \mapsto \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$ thus labels positively all points falling on one side of the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ and negatively all others.

5.2 Separable case

In this section, we assume that the training sample S can be linearly separated, that is, we assume the existence of a hyperplane that perfectly separates the training sample into two populations of positively and negatively labeled points, as illustrated by the left panel of figure 5.1. This is equivalent to the existence of $(\mathbf{w}, b) \in (\mathbb{R}^N - \{\mathbf{0}\}) \times \mathbb{R}$ such that

$$\forall i \in [m], \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 0. \tag{5.3}$$

But, as can be seen from figure 5.1, there are then infinitely many such separating hyperplanes. Which hyperplane should a learning algorithm select? The definition of the SVM solution is based on the notion of *geometric margin*.

Definition 5.1 (Geometric margin) *The* geometric margin $\rho_h(\mathbf{x})$ *of a linear classifier* $h\colon \mathbf{x} \mapsto \mathbf{w}\cdot\mathbf{x}+b$ *at a point* $\mathbf{x}$ *is its Euclidean distance to the hyperplane* $\mathbf{w}\cdot\mathbf{x}+b = 0$*:*

$$\rho_h(x) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|_2}\,. \tag{5.4}$$

The geometric margin ρ_h *of a linear classifier* h *for a sample* $S = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ *is the minimum geometric margin over the points in the sample,* $\rho_h = \min_{i\in[m]} \rho_h(x_i)$*, that is the distance of the hyperplane defining* h *to the closest sample points.*

The SVM solution is the separating hyperplane with the maximum geometric margin and is thus known as the *maximum-margin hyperplane*. The right panel of figure 5.1 illustrates the maximum-margin hyperplane returned by the SVM

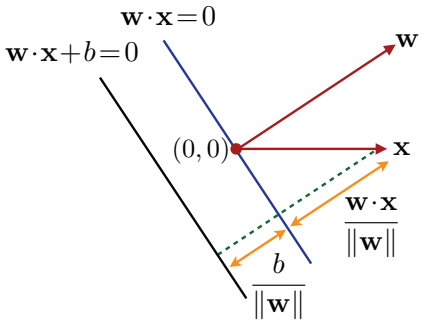


Figure 5.2
An illustration of the geometric margin of a point $\mathbf{x}$ in the case $\mathbf{w} \cdot \mathbf{x} > 0$ and $b > 0$.

algorithm in the separable case. We will present later in this chapter a theory that provides a strong justification for this solution. We can observe already, however, that the SVM solution can also be viewed as the "safest" choice in the following sense: a test point is classified correctly by a separating hyperplane with geometric margin ρ even when it falls within a distance ρ of the training samples sharing the same label; for the SVM solution, ρ is the maximum geometric margin and thus the "safest" value.

5.2.1 Primal optimization problem

We now derive the equations and optimization problem that define the SVM solution. By definition of the geometric margin (see also figure 5.2), the maximum margin ρ of a separating hyperplane is given by

$$\rho = \max_{\mathbf{w},b:\ y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0} \min_{i\in[m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i+b|}{\|\mathbf{w}\|} = \max_{\mathbf{w},b} \min_{i\in[m]} \frac{y_i\big(\mathbf{w}\cdot\mathbf{x}_i+b\big)}{\|\mathbf{w}\|}. \tag{5.5}$$

The second equality follows from the fact that, since the sample is linearly separable, for the maximizing pair $(\mathbf{w}, b)$, $y_i\big(\mathbf{w}\cdot\mathbf{x}_i+b\big)$ must be non-negative for all $i \in [m]$. Now, observe that the last expression is invariant to multiplication of $(\mathbf{w}, b)$ by a positive scalar. Thus, we can restrict ourselves to pairs $(\mathbf{w}, b)$ scaled such that $\min_{i\in[m]} y_i(\mathbf{w}\cdot\mathbf{x}_i+b) = 1$:

$$\rho = \max_{\substack{\mathbf{w},b:\\ \min_{i\in[m]} y_i(\mathbf{w}\cdot\mathbf{x}_i+b)=1}} \frac{1}{\|\mathbf{w}\|} = \max_{\substack{\mathbf{w},b:\\ \forall i\in[m], y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1}} \frac{1}{\|\mathbf{w}\|}. \tag{5.6}$$

The second equality results from the fact that for the maximizing pair $(\mathbf{w}, b)$, the minimum of $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)$ is 1.

Figure 5.3 illustrates the solution $(\mathbf{w}, b)$ of the maximization (5.6). In addition to the maximum-margin hyperplane, it also shows the *marginal hyperplanes*, which are

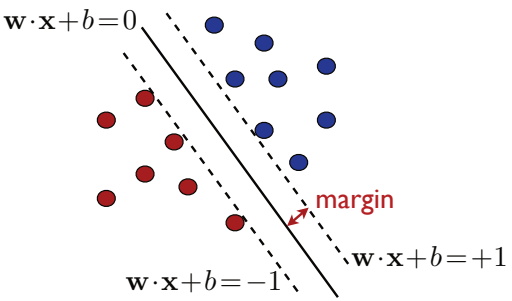


Figure 5.3
Maximum-margin hyperplane solution of (5.6). The marginal hyperplanes are represented by dashed lines on the figure.

the hyperplanes parallel to the separating hyperplane and passing through the closest points on the negative or positive sides. Since they are parallel to the separating hyperplane, they admit the same normal vector $\mathbf{w}$. Furthermore, since $|\mathbf{w}\cdot\mathbf{x}+b| = 1$ for the closest points, the equations of the marginal hyperplanes are $\mathbf{w}\cdot\mathbf{x}+b = \pm 1$.

Since maximizing $1/\|\mathbf{w}\|$ is equivalent to minimizing $\frac{1}{2}\|\mathbf{w}\|^2$, in view of (5.6), the pair $(\mathbf{w}, b)$ returned by SVM in the separable case is the solution of the following convex optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{5.7}$$
$$\text{subject to: } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, \ \forall i \in [m]\,.$$

The objective function $F\colon \mathbf{w} \mapsto \frac{1}{2}\|\mathbf{w}\|^2$ is infinitely differentiable. Its gradient is $\nabla F(\mathbf{w}) = \mathbf{w}$ and its Hessian is the identity matrix $\nabla^2 F(\mathbf{w}) = \mathbf{I}$, whose eigenvalues are strictly positive. Therefore, $\nabla^2 F(\mathbf{w}) \succ \mathbf{0}$ and F is strictly convex. The constraints are all defined by affine functions $g_i\colon (\mathbf{w}, b) \mapsto 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)$ and are therefore qualified. Thus, in view of the results known for convex optimization (see appendix B for details), the optimization problem of (5.7) admits a unique solution, an important and favorable property that does not hold for all learning algorithms.

Moreover, since the objective function is quadratic and the constraints are affine, the optimization problem of (5.7) is in fact a specific instance of *quadratic programming* (QP), a family of problems extensively studied in optimization. A variety of commercial and open-source solvers are available for solving convex QP problems. Additionally, motivated by the empirical success of SVMs along with its rich theoretical underpinnings, specialized methods have been developed to more efficiently solve this particular convex QP problem, notably the block coordinate descent algorithms with blocks of just two coordinates.

5.2.2 Support vectors

Returning to the optimization problem (5.7), we note that the constraints are affine and thus qualified. The objective function as well as the affine constraints are convex and differentiable. Thus, the requirements of theorem B.30 hold and the KKT conditions apply at the optimum. We shall use these conditions to both analyze the algorithm and demonstrate several of its crucial properties, and subsequently derive the dual optimization problem associated to SVMs in section 5.2.3.

We introduce Lagrange variables $\alpha_i \geq 0$, $i \in [m]$, associated to the m constraints and denote by $\boldsymbol{\alpha}$ the vector $(\alpha_1, \ldots, \alpha_m)^\top$. The Lagrangian can then be defined for all $\mathbf{w} \in \mathbb{R}^N$, $b \in \mathbb{R}$, and $\boldsymbol{\alpha} \in \mathbb{R}^m_+$, by

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]\,. \tag{5.8}$$

The KKT conditions are obtained by setting the gradient of the Lagrangian with respect to the primal variables $\mathbf{w}$ and b to zero and by writing the complementarity conditions:

$$\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{5.9}$$

$$\nabla_b \mathcal{L} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \qquad \Longrightarrow \qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{5.10}$$

$$\forall i, \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0 \qquad \Longrightarrow \qquad \alpha_i = 0 \vee y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1. \tag{5.11}$$

By equation (5.9), the weight vector $\mathbf{w}$ at the solution of the SVM problem is a linear combination of the training set vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$. A vector $\mathbf{x}_i$ appears in that expansion iff $\alpha_i \neq 0$. Such vectors are called *support vectors*. By the complementarity conditions (5.11), if $\alpha_i \neq 0$, then $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$. Thus, support vectors lie on the marginal hyperplanes $\mathbf{w} \cdot \mathbf{x}_i + b = \pm 1$.

Support vectors fully define the maximum-margin hyperplane or SVM solution, which justifies the name of the algorithm. By definition, vectors not lying on the marginal hyperplanes do not affect the definition of these hyperplanes — in their absence, the solution to the SVM problem remains unchanged. Note that while the solution $\mathbf{w}$ of the SVM problem is unique, the support vectors are not. In dimension N, $N + 1$ points are sufficient to define a hyperplane. Thus, when more than $N + 1$ points lie on a marginal hyperplane, different choices are possible for the $N + 1$ support vectors.

5.2.3 Dual optimization problem

To derive the dual form of the constrained optimization problem (5.7), we plug into the Lagrangian the definition of $\mathbf{w}$ in terms of the dual variables as expressed in

(5.9) and apply the constraint (5.10). This yields

$$\mathcal{L} = \underbrace{\frac{1}{2}\|\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i\|^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)}_{-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)} - \underbrace{\sum_{i=1}^{m}\alpha_i y_i b}_{0} + \sum_{i=1}^{m}\alpha_i\,, \tag{5.12}$$

which simplifies to

$$\mathcal{L} = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)\,. \tag{5.13}$$

This leads to the following dual optimization problem for SVMs in the separable case:

$$\begin{aligned}\max_{\boldsymbol{\alpha}}\ & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) \\ \text{subject to: } & \alpha_i \geq 0 \wedge \sum_{i=1}^{m}\alpha_i y_i = 0,\ \forall i\in[m]\,.\end{aligned} \tag{5.14}$$

The objective function $G\colon \boldsymbol{\alpha}\mapsto \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)$ is infinitely differentiable. Its Hessian is given by $\nabla^2 G = -\mathbf{A}$, with $\mathbf{A} = \left(y_i\mathbf{x}_i\cdot y_j\mathbf{x}_j\right)_{ij}$. $\mathbf{A}$ is the Gram matrix associated to the vectors $y_1\mathbf{x}_1,\ldots,y_m\mathbf{x}_m$ and is therefore positive semidefinite (see section A.2.3), which shows that $\nabla^2 G \preceq \mathbf{0}$ and that G is a concave function. Since the constraints are affine and convex, the maximization problem (5.14) is a convex optimization problem. Since G is a quadratic function of $\boldsymbol{\alpha}$, this dual optimization problem is also a QP problem, as in the case of the primal optimization and once again both general-purpose and specialized QP solvers can be used to obtain the solution (see exercise 5.4 for details on the SMO algorithm, which is often used to solve the dual form of the SVM problem in the more general non-separable setting).

Moreover, since the constraints are affine, they are qualified and strong duality holds (see appendix B). Thus, the primal and dual problems are equivalent, i.e., the solution $\boldsymbol{\alpha}$ of the dual problem (5.14) can be used directly to determine the hypothesis returned by SVMs, using equation (5.9):

$$h(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}\cdot\mathbf{x}+b) = \operatorname{sgn}\Big(\sum_{i=1}^{m}\alpha_i y_i(\mathbf{x}_i\cdot\mathbf{x}) + b\Big). \tag{5.15}$$

Since support vectors lie on the marginal hyperplanes, for any support vector $\mathbf{x}_i$, $\mathbf{w} \cdot \mathbf{x}_i + b = y_i$, and thus b can be obtained via

$$b = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i) \,. \tag{5.16}$$

The dual optimization problem (5.14) and the expressions (5.15) and (5.16) reveal an important property of SVMs: the hypothesis solution depends only on inner products between vectors and not directly on the vectors themselves. This observation is key and its importance will become clear in Chapter 6 where we introduce kernel methods.

Equation (5.16) can now be used to derive a simple expression of the geometric margin ρ in terms of $\boldsymbol{\alpha}$. Since (5.16) holds for all i with $\alpha_i \neq 0$, multiplying both sides by $\alpha_i y_i$ and taking the sum leads to

$$\sum_{i=1}^{m} \alpha_i y_i b = \sum_{i=1}^{m} \alpha_i y_i^2 - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \,. \tag{5.17}$$

Using the fact that $y_i^2 = 1$ along with equation (5.9) then yields

$$0 = \sum_{i=1}^{m} \alpha_i - \|\mathbf{w}\|^2. \tag{5.18}$$

Noting that $\alpha_i \geq 0$, we obtain the following expression of the margin ρ in terms of the L_1 norm of $\boldsymbol{\alpha}$:

$$\rho^2 = \frac{1}{\|\mathbf{w}\|_2^2} = \frac{1}{\sum_{i=1}^{m} \alpha_i} = \frac{1}{\|\boldsymbol{\alpha}\|_1}. \tag{5.19}$$

5.2.4 Leave-one-out analysis

We now use the notion of *leave-one-out error* to derive a first learning guarantee for SVMs based on the fraction of support vectors in the training set.

Definition 5.2 (Leave-one-out error) *Let h_S denote the hypothesis returned by a learning algorithm $\mathcal{A}$, when trained on a fixed sample S. Then, the* leave-one-out error *of $\mathcal{A}$ on a sample S of size m is defined by*

$$\widehat{R}_{\mathrm{LOO}}(\mathcal{A}) = \frac{1}{m} \sum_{i=1}^{m} 1_{h_{S-\{x_i\}}(x_i) \neq y_i}.$$

Thus, for each $i \in [m]$, $\mathcal{A}$ is trained on all the points in S except for x_i, i.e., $S - \{x_i\}$, and its error is then computed using x_i. The leave-one-out error is the average of these errors. We will use an important property of the leave-one-out error stated in the following lemma.

Lemma 5.3 *The average leave-one-out error for samples of size $m \geq 2$ is an unbiased estimate of the average generalization error for samples of size $m-1$:*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{R}_{\mathrm{LOO}}(\mathcal{A})] = \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^{m-1}}[R(h_{S'})], \tag{5.20}$$

where $\mathcal{D}$ denotes the distribution according to which points are drawn.

Proof: By the linearity of expectation, we can write

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{R}_{\mathrm{LOO}}(\mathcal{A})] &= \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[1_{h_{S-\{x_i\}}(x_i)\neq y_i}] \\
&= \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[1_{h_{S-\{x_1\}}(x_1)\neq y_1}] \\
&= \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^{m-1},x_1\sim\mathcal{D}}[1_{h_{S'}(x_1)\neq y_1}] \\
&= \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^{m-1}}[\mathop{\mathbb{E}}_{x_1\sim\mathcal{D}}[1_{h_{S'}(x_1)\neq y_1}]] \\
&= \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^{m-1}}[R(h_{S'})].
\end{aligned}$$

For the second equality, we used the fact that, since the points of S are drawn in an i.i.d. fashion, the expectation $\mathbb{E}_{S\sim\mathcal{D}^m}[1_{h_{S-\{x_i\}}(x_i)\neq y_i}]$ does not depend on the choice of $i \in [m]$ and is thus equal to $\mathbb{E}_{S\sim\mathcal{D}^m}[1_{h_{S-\{x_1\}}(x_1)\neq y_1}]$. □

In general, computing the leave-one-out error may be costly since it requires training m times on samples of size $m-1$. In some situations however, it is possible to derive the expression of $\widehat{R}_{\mathrm{LOO}}(\mathcal{A})$ much more efficiently (see exercise 11.9).

Theorem 5.4 *Let h_S be the hypothesis returned by SVMs for a sample S, and let $N_{SV}(S)$ be the number of support vectors that define h_S. Then,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[R(h_S)] \leq \mathop{\mathbb{E}}_{S\sim\mathcal{D}^{m+1}}\left[\frac{N_{\mathrm{SV}}(S)}{m+1}\right].$$

Proof: Let S be a linearly separable sample of $m+1$. If x is not a support vector for h_S, removing it does not change the SVM solution. Thus, $h_{S-\{x\}} = h_S$ and $h_{S-\{x\}}$ correctly classifies x. By contraposition, if $h_{S-\{x\}}$ misclassifies x, x must be a support vector, which implies

$$\widehat{R}_{\mathrm{LOO}}(\mathrm{SVM}) \leq \frac{N_{\mathrm{SV}}(S)}{m+1}. \tag{5.21}$$

Taking the expectation of both sides and using lemma 5.3 yields the result. □

Theorem 5.4 gives a sparsity argument in favor of SVMs: the average error of the algorithm is upper bounded by the average fraction of support vectors. One may hope that for many distributions seen in practice, a relatively small number of the training points will lie on the marginal hyperplanes. The solution will then be sparse in the sense that a small fraction of the dual variables α_i will be non-

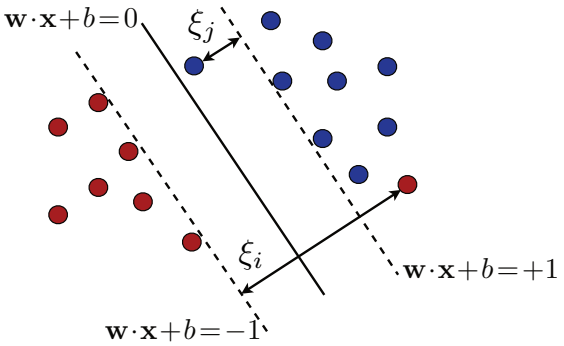


Figure 5.4
A separating hyperplane with point $\mathbf{x}_i$ classified incorrectly and point $\mathbf{x}_j$ correctly classified, but with margin less than 1.

zero. Note, however, that this bound is relatively weak since it applies only to the average generalization error of the algorithm over all samples of size m. It provides no information about the variance of the generalization error. In section 5.4, we present stronger high-probability bounds using a different argument based on the notion of margin.

5.3 Non-separable case

In most practical settings, the training data is not linearly separable, which implies that for any hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, there exists $\mathbf{x}_i \in S$ such that

$$y_i \left[\mathbf{w} \cdot \mathbf{x}_i + b\right] \ngeq 1 \,. \tag{5.22}$$

Thus, the constraints imposed in the linearly separable case discussed in section 5.2 cannot all hold simultaneously. However, a relaxed version of these constraints can indeed hold, that is, for each $i \in [m]$, there exist $\xi_i \geq 0$ such that

$$y_i \left[\mathbf{w} \cdot \mathbf{x}_i + b\right] \geq 1 - \xi_i \,. \tag{5.23}$$

The variables ξ_i are known as *slack variables* and are commonly used in optimization to define relaxed versions of constraints. Here, a slack variable ξ_i measures the distance by which vector $\mathbf{x}_i$ violates the desired inequality, $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$. Figure 5.4 illustrates the situation. For a hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, a vector $\mathbf{x}_i$ with $\xi_i > 0$ can be viewed as an *outlier*. Each $\mathbf{x}_i$ must be positioned on the correct side of the appropriate marginal hyperplane to not be considered an outlier. As a consequence, a vector $\mathbf{x}_i$ with $0 < y_i(\mathbf{w} \cdot \mathbf{x}_i + b) < 1$ is correctly classified by the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ but is nonetheless considered to be an outlier, that

is, $\xi_i > 0$. If we omit the outliers, the training data is correctly separated by $\mathbf{w} \cdot \mathbf{x} + b = 0$ with a margin $\rho = 1/\|\mathbf{w}\|$ that we refer to as the *soft margin*, as opposed to the *hard margin* in the separable case.

How should we select the hyperplane in the non-separable case? One idea consists of selecting the hyperplane that minimizes the empirical error. But, that solution will not benefit from the large-margin guarantees we will present in section 5.4. Furthermore, the problem of determining a hyperplane with the smallest zero-one loss, that is the smallest number of misclassifications, is NP-hard as a function of the dimension N of the space.

Here, there are two conflicting objectives: on one hand, we wish to limit the total amount of slack due to outliers, which can be measured by $\sum_{i=1}^m \xi_i$, or, more generally by $\sum_{i=1}^m \xi_i^p$ for some $p \geq 1$; on the other hand, we seek a hyperplane with a large margin, though a larger margin can lead to more outliers and thus larger amounts of slack.

5.3.1 Primal optimization problem

This leads to the following general optimization problem defining SVMs in the non-separable case where the parameter $C \geq 0$ determines the trade-off between margin-maximization (or minimization of $\|\mathbf{w}\|^2$) and the minimization of the slack penalty $\sum_{i=1}^m \xi_i^p$:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \xi_i^p \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \ \wedge \ \xi_i \geq 0, i \in [m]\,, \end{aligned} \tag{5.24}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m)^\top$. The parameter C is typically determined via n-fold cross-validation (see section 4.5).

As in the separable case, (5.24) is a convex optimization problem since the constraints are affine and thus convex and since the objective function is convex for any $p \geq 1$. In particular, $\boldsymbol{\xi} \mapsto \sum_{i=1}^m \xi_i^p = \|\boldsymbol{\xi}\|_p^p$ is convex in view of the convexity of the norm $\|\cdot\|_p$.

There are many possible choices for p leading to more or less aggressive penalizations of the slack terms (see exercise 5.1). The choices $p = 1$ and $p = 2$ lead to the most straightforward solutions and analyses. The loss functions associated with $p = 1$ and $p = 2$ are called the *hinge loss* and the *quadratic hinge loss*, respectively. Figure 5.5 shows the plots of these loss functions as well as that of the standard zero-one loss function. Both hinge losses are convex upper bounds on the zero-one loss, thus making them well suited for optimization. In what follows, the analysis is presented in the case of the hinge loss ($p = 1$), which is the most widely used loss function for SVMs.

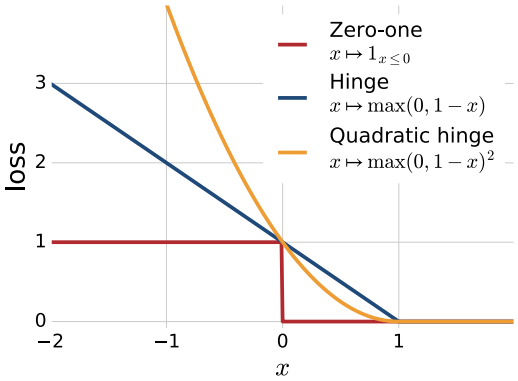


Figure 5.5
Both the hinge loss and the quadratic hinge loss provide convex upper bounds on the binary zero-one loss.

5.3.2 Support vectors

As in the separable case, the constraints are affine and thus qualified. The objective function as well as the affine constraints are convex and differentiable. Thus, the hypotheses of theorem B.30 hold and the KKT conditions apply at the optimum. We use these conditions to both analyze the algorithm and demonstrate several of its crucial properties, and subsequently derive the dual optimization problem associated to SVMs in section 5.3.3.

We introduce Lagrange variables $\alpha_i \geq 0$, $i \in [m]$, associated to the first m constraints and $\beta_i \geq 0$, $i \in [m]$ associated to the non-negativity constraints of the slack variables. We denote by $\boldsymbol{\alpha}$ the vector $(\alpha_1, \dots, \alpha_m)^\top$ and by $\boldsymbol{\beta}$ the vector $(\beta_1, \dots, \beta_m)^\top$. The Lagrangian can then be defined for all $\mathbf{w} \in \mathbb{R}^N$, $b \in \mathbb{R}$, and $\boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}_+^m$, by

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^m \beta_i \xi_i\,. \quad (5.25)$$

The KKT conditions are obtained by setting the gradient of the Lagrangian with respect to the primal variables $\mathbf{w}$, b, and ξ_is to zero and by writing the complemen-

tarity conditions:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{5.26}$$

$$\nabla_b \mathcal{L} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \qquad \Longrightarrow \qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{5.27}$$

$$\nabla_{\xi_i}\mathcal{L} = C - \alpha_i - \beta_i = 0 \qquad \Longrightarrow \qquad \alpha_i + \beta_i = C \tag{5.28}$$

$$\forall i, \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] = 0 \qquad \Longrightarrow \qquad \alpha_i = 0 \vee y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1 - \xi_i \tag{5.29}$$

$$\forall i, \beta_i \xi_i = 0 \qquad \Longrightarrow \qquad \beta_i = 0 \vee \xi_i = 0\,. \tag{5.30}$$

By equation (5.26), as in the separable case, the weight vector $\mathbf{w}$ at the solution of the SVM problem is a linear combination of the training set vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$. A vector $\mathbf{x}_i$ appears in that expansion iff $\alpha_i \neq 0$. Such vectors are called *support vectors*. Here, there are two types of support vectors. By the complementarity condition (5.29), if $\alpha_i \neq 0$, then $y_i(\mathbf{w}\cdot\mathbf{x}_i+b) = 1-\xi_i$. If $\xi_i = 0$, then $y_i(\mathbf{w}\cdot\mathbf{x}_i+b) = 1$ and $\mathbf{x}_i$ lies on a marginal hyperplane, as in the separable case. Otherwise, $\xi_i \neq 0$ and $\mathbf{x}_i$ is an outlier. In this case, (5.30) implies $\beta_i = 0$ and (5.28) then requires $\alpha_i = C$. Thus, support vectors $\mathbf{x}_i$ are either outliers, in which case $\alpha_i = C$, or vectors lying on the marginal hyperplanes. As in the separable case, note that while the weight vector $\mathbf{w}$ solution is unique, the support vectors are not.

5.3.3 Dual optimization problem

To derive the dual form of the constrained optimization problem (5.24), we plug into the Lagrangian the definition of $\mathbf{w}$ in terms of the dual variables (5.26) and apply the constraint (5.27). This yields

$$\mathcal{L} = \underbrace{\frac{1}{2}\|\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i\|^2 - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)}_{-\frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)} - \underbrace{\sum_{i=1}^{m} \alpha_i y_i b}_{0} + \sum_{i=1}^{m} \alpha_i\,. \tag{5.31}$$

Remarkably, we find that the objective function is no different than in the separable case:

$$\mathcal{L} = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)\,. \tag{5.32}$$

However, here, in addition to $\alpha_i \geq 0$, we must impose the constraint on the Lagrange variables $\beta_i \geq 0$. In view of (5.28), this is equivalent to $\alpha_i \leq C$. This leads to the following dual optimization problem for SVMs in the non-separable case, which

only differs from that of the separable case (5.14) by the constraints $\alpha_i \leq C$:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{5.33}$$
$$\text{subject to: } 0 \leq \alpha_i \leq C \wedge \sum_{i=1}^{m} \alpha_i y_i = 0, i \in [m].$$

Thus, our previous comments about the optimization problem (5.14) apply to (5.33) as well. In particular, the objective function is concave and infinitely differentiable and (5.33) is equivalent to a convex QP. The problem is equivalent to the primal problem (5.24).

The solution $\boldsymbol{\alpha}$ of the dual problem (5.33) can be used directly to determine the hypothesis returned by SVMs, using equation (5.26):

$$h(\mathbf{x}) = \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x} + b) = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b\Big). \tag{5.34}$$

Moreover, b can be obtained from any support vector $\mathbf{x}_i$ lying on a marginal hyperplane, that is any vector $\mathbf{x}_i$ with $0 < \alpha_i < C$. For such support vectors, $\mathbf{w} \cdot \mathbf{x}_i + b = y_i$ and thus

$$b = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)\,. \tag{5.35}$$

As in the separable case, the dual optimization problem (5.33) and the expressions (5.34) and (5.35) show an important property of SVMs: the hypothesis solution depends only on inner products between vectors and not directly on the vectors themselves. This fact can be used to extend SVMs to define non-linear decision boundaries, as we shall see in chapter 6.

5.4 Margin theory

This section presents generalization bounds which provide a strong theoretical justification for the SVM algorithm.

Recall that the VC-dimension of the family of hyperplanes or linear hypotheses in $\mathbb{R}^N$ is $N + 1$. Thus, the application of the VC-dimension bound (3.29) of corollary 3.19 to this hypothesis set yields the following: for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in \mathcal{H}$,

$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{2(N+1)\log \frac{em}{N+1}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}. \tag{5.36}$$

When the dimension of the feature space N is large compared to the sample size m, this bound is uninformative. Remarkably, the learning guarantees presented in this section are independent of the dimension N and thus hold regardless of its value.

The guarantees we will present hold for real-valued functions such as the function $\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b$ returned by SVMs, as opposed to classification functions returning $+1$ or -1, such as $\mathbf{x} \mapsto \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$. They are based on the notion of *confidence margin*. The confidence margin of a real-valued function h at a point x labeled with y is the quantity $yh(x)$. Thus, when $yh(x) > 0$, h classifies x correctly but we interpret the magnitude of $|h(x)|$ as the *confidence* of the prediction made by h. The notion of confidence margin is distinct from that of geometric margin and does not require a linear separability assumption. But, the two notions are related as follows in the separable case: for $h\colon \mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b$ with geometric margin ρ_{geom}, the confidence margin at any point $\mathbf{x}$ of the training sample with label y is at least $\rho_{\text{geom}}\|\mathbf{w}\|$, i.e. $|yh(\mathbf{x})| \geq \rho_{\text{geom}}\|\mathbf{w}\|$.

In view of the definition of the confidence margin, for any parameter $\rho > 0$, we will define a *ρ-margin loss function* that, as with the zero-one loss, penalizes h with the cost of 1 when it misclassifies point x ($yh(x) \leq 0$), but also penalizes h (linearly) when it correctly classifies x with confidence less than or equal to ρ ($yh(x) \leq \rho$). The main margin-based generalization bounds of this section are presented in terms of this loss function, which is formally defined as follows.

Definition 5.5 (Margin loss function) *For any $\rho > 0$, the ρ-margin loss is the function $L_\rho\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ defined for all $y, y' \in \mathbb{R}$ by $L_\rho(y, y') = \Phi_\rho(yy')$ with,*

$$\Phi_\rho(x) = \min\left(1, \max\left(0, 1 - \frac{x}{\rho}\right)\right) = \begin{cases} 1 & \text{if } x \leq 0 \\ 1 - \frac{x}{\rho} & \text{if } 0 \leq x \leq \rho \\ 0 & \text{if } \rho \leq x. \end{cases}$$

This loss function is illustrated by figure 5.6. The parameter $\rho > 0$ can be interpreted as the confidence margin demanded from a hypothesis h. The empirical margin loss is similarly defined as the margin loss over the training sample.

Definition 5.6 (Empirical margin loss) *Given a sample $S = (x_1, \ldots, x_m)$ and a hypothesis h, the empirical margin loss is defined by*

$$\widehat{R}_{S,\rho}(h) = \frac{1}{m} \sum_{i=1}^{m} \Phi_\rho(y_i h(x_i)) \,. \tag{5.37}$$

Note that, for any $i \in [m]$, $\Phi_\rho(y_i h(x_i)) \leq 1_{y_i h(x_i) \leq \rho}$. Thus, the empirical margin loss can be upper-bounded as follows:

$$\widehat{R}_{S,\rho}(h) \leq \frac{1}{m} \sum_{i=1}^{m} 1_{y_i h(x_i) \leq \rho} \,. \tag{5.38}$$

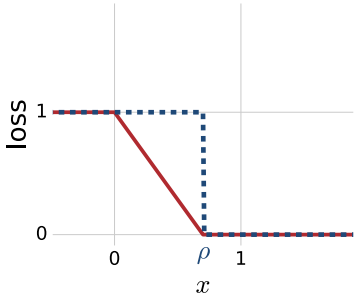


Figure 5.6
The margin loss illustrated in red, defined with respect to margin parameter $\rho = 0.7$.

In all the results that follow, the empirical margin loss can be replaced by this upper bound, which admits a simple interpretation: it is the fraction of the points in the training sample S that have been misclassified or classified with confidence less than ρ. In other words, the upper bound is then the fraction of the points in the training data with margin less than ρ. This corresponds to the loss function indicated by the blue dotted line in figure 5.6.

A key benefit of using a loss function based on Φ_ρ as opposed to the zero-one loss or the loss defined by the blue dotted line of figure 5.6 is that Φ_ρ is $1/\rho$-Lipschitz, since the absolute value of the slope of the function is at most $1/\rho$. The following lemma bounds the empirical Rademacher complexity of a hypothesis set $\mathcal{H}$ after composition with such a Lipschitz function in terms of the empirical Rademacher complexity of $\mathcal{H}$. It will be needed for the proof of the margin-based generalization bound.

Lemma 5.7 (Talagrand's lemma) *Let $\Phi_1, \ldots, \Phi_m$ be l-Lipschitz functions from $\mathbb{R}$ to $\mathbb{R}$ and $\sigma_1, \ldots, \sigma_m$ be Rademacher random variables. Then, for any hypothesis set $\mathcal{H}$ of real-valued functions, the following inequality holds:*

$$\frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i(\Phi_i\circ h)(x_i))\right] \leq \frac{l}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i h(x_i)\right] = l\,\widehat{\mathfrak{R}}_S(\mathcal{H})\,.$$

In particular, if $\Phi_i = \Phi$ for all $i \in [m]$, then the following holds:

$$\widehat{\mathfrak{R}}_S(\Phi\circ\mathcal{H}) \leq l\,\widehat{\mathfrak{R}}_S(\mathcal{H})\,.$$

Proof: First we fix a sample $S = (x_1, \ldots, x_m)$, then, by definition,

$$\frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i(\Phi_m\circ h)(x_i)\right] = \frac{1}{m}\mathop{\mathbb{E}}_{\sigma_1,\ldots,\sigma_{m-1}}\left[\mathop{\mathbb{E}}_{\sigma_m}\left[\sup_{h\in\mathcal{H}}u_{m-1}(h)+\sigma_m(\Phi_m\circ h)(x_m)\right]\right],$$

where $u_{m-1}(h) = \sum_{i=1}^{m-1} \sigma_i(\Phi_i \circ h)(x_i)$. By definition of the supremum, for any $\epsilon > 0$, there exist $h_1, h_2 \in \mathcal{H}$ such that

$$u_{m-1}(h_1) + (\Phi_m \circ h_1)(x_m) \geq (1-\epsilon)\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) + (\Phi_m \circ h)(x_m)\Big]$$
$$\text{and} \quad u_{m-1}(h_2) - (\Phi_m \circ h_2)(x_m) \geq (1-\epsilon)\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) - (\Phi_m \circ h)(x_m)\Big].$$

Thus, for any $\epsilon > 0$, by definition of $\mathbb{E}_{\sigma_m}$,

$$\begin{aligned}
&(1-\epsilon)\mathop{\mathbb{E}}_{\sigma_m}\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) + \sigma_m(\Phi_m \circ h)(x_m)\Big] \\
&= (1-\epsilon)\left[\frac{1}{2}\sup_{h\in\mathcal{H}}\Big[u_{m-1}(h) + (\Phi_m \circ h)(x_m)\Big] + \frac{1}{2}\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) - (\Phi_m \circ h)(x_m)\Big]\right] \\
&\leq \frac{1}{2}[u_{m-1}(h_1) + (\Phi_m \circ h_1)(x_m)] + \frac{1}{2}[u_{m-1}(h_2) - (\Phi_m \circ h_2)(x_m)].
\end{aligned}$$

Let $s = \operatorname{sgn}(h_1(x_m) - h_2(x_m))$. Then, the previous inequality implies

$$\begin{aligned}
&(1-\epsilon)\mathop{\mathbb{E}}_{\sigma_m}\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) + \sigma_m(\Phi_m \circ h)(x_m)\Big] \\
&\leq \frac{1}{2}[u_{m-1}(h_1) + u_{m-1}(h_2) + sl(h_1(x_m) - h_2(x_m))] && \text{(Lipschitz property)} \\
&= \frac{1}{2}[u_{m-1}(h_1) + slh_1(x_m)] + \frac{1}{2}[u_{m-1}(h_2) - slh_2(x_m)] && \text{(rearranging)} \\
&\leq \frac{1}{2}\sup_{h\in\mathcal{H}}[u_{m-1}(h) + slh(x_m)] + \frac{1}{2}\sup_{h\in\mathcal{H}}[u_{m-1}(h) - slh(x_m)] && \text{(definition of sup)} \\
&= \mathop{\mathbb{E}}_{\sigma_m}\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) + \sigma_m lh(x_m)\Big]. && \text{(definition of } \mathop{\mathbb{E}}_{\sigma_m}\text{)}
\end{aligned}$$

Since the inequality holds for all $\epsilon > 0$, we have

$$\mathop{\mathbb{E}}_{\sigma_m}\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) + \sigma_m(\Phi_m \circ h)(x_m)\Big] \leq \mathop{\mathbb{E}}_{\sigma_m}\Big[\sup_{h\in\mathcal{H}} u_{m-1}(h) + \sigma_m lh(x_m)\Big].$$

Proceeding in the same way for all other σ_i $(i \neq m)$ proves the lemma. □

The following is a general margin-based generalization bound that will be used in the analysis of several algorithms.

Theorem 5.8 (Margin bound for binary classification) *Let $\mathcal{H}$ be a set of real-valued functions. Fix $\rho > 0$, then, for any $\delta > 0$, with probability at least $1-\delta$, each of the following holds for all $h \in \mathcal{H}$:*

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \frac{2}{\rho}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \tag{5.39}$$

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \frac{2}{\rho}\widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}\,. \tag{5.40}$$

Proof: Let $\widetilde{\mathcal{H}} = \{z = (x, y) \mapsto yh(x) \colon h \in \mathcal{H}\}$. Consider the family of functions taking values in $[0, 1]$:

$$\widetilde{\mathcal{H}} = \{\Phi_\rho \circ f \colon f \in \widetilde{\mathcal{H}}\}\,.$$

By theorem 3.3, with probability at least $1 - \delta$, for all $g \in \widetilde{\mathcal{H}}$,

$$\mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\widetilde{\mathcal{H}}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\,,$$

and thus, for all $h \in \mathcal{H}$,

$$\mathbb{E}[\Phi_\rho(yh(x))] \leq \widehat{R}_{S,\rho}(h) + 2\mathfrak{R}_m\big(\Phi_\rho \circ \widetilde{\mathcal{H}}\big) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\,.$$

Since $1_{u\leq 0} \leq \Phi_\rho(u)$ for all $u \in \mathbb{R}$, we have $R(h) = \mathbb{E}[1_{yh(x)\leq 0}] \leq \mathbb{E}[\Phi_\rho(yh(x))]$, thus

$$R(h) \leq \widehat{R}_{S,\rho}(h) + 2\mathfrak{R}_m\big(\Phi_\rho \circ \widetilde{\mathcal{H}}\big) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\,.$$

Since Φ_ρ is $1/\rho$-Lipschitz, by lemma 5.7, we have $\mathfrak{R}_m\big(\Phi_\rho \circ \widetilde{\mathcal{H}}\big) \leq \frac{1}{\rho}\mathfrak{R}_m(\widetilde{\mathcal{H}})$ and $\mathfrak{R}_m(\widetilde{\mathcal{H}})$ can be rewritten as follows:

$$\mathfrak{R}_m(\widetilde{\mathcal{H}}) = \frac{1}{m}\mathop{\mathbb{E}}_{S,\sigma}\Big[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i y_i h(x_i)\Big] = \frac{1}{m}\mathop{\mathbb{E}}_{S,\sigma}\Big[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i h(x_i)\Big] = \mathfrak{R}_m\big(\mathcal{H}\big)\,.$$

This proves (5.39). The second inequality, (5.40), can be derived in the same way by using the second inequality of theorem 3.3, (3.4), instead of (3.3). □

The generalization bounds of theorem 5.8 suggest a trade-off: a larger value of ρ decreases the complexity term (second term), but tends to increase the empirical margin-loss $\widehat{R}_{S,\rho}(h)$ (first term) by requiring from a hypothesis h a higher confidence margin. Thus, if for a relatively large value of ρ the empirical margin loss of h remains relatively small, then h benefits from a very favorable guarantee on its generalization error. For theorem 5.8, the margin parameter ρ must be selected beforehand. But, the bounds of the theorem can be generalized to hold uniformly for all $\rho \in (0, 1]$ at the cost of a modest additional term $\sqrt{\frac{\log\log_2\frac{2}{\rho}}{m}}$, as shown in the following theorem (a version of this theorem with better constants can be derived, see exercise 5.2).

Theorem 5.9 *Let $\mathcal{H}$ be a set of real-valued functions. Fix $r > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following holds for all $h \in \mathcal{H}$ and $\rho \in (0, r]$:*

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \frac{4}{\rho}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log \log_2 \frac{2r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} \tag{5.41}$$

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \frac{4}{\rho}\widehat{\mathfrak{R}}_S(\mathcal{H}) + \sqrt{\frac{\log \log_2 \frac{2r}{\rho}}{m}} + 3\sqrt{\frac{\log \frac{4}{\delta}}{2m}}\,. \tag{5.42}$$

Proof: Consider two sequences $(\rho_k)_{k\geq 1}$ and $(\epsilon_k)_{k\geq 1}$, with $\epsilon_k \in (0, 1]$. By theorem 5.8, for any fixed $k \geq 1$,

$$\mathbb{P}\left[\sup_{h\in H} R(h) - \widehat{R}_{S,\rho_k}(h) > \frac{2}{\rho_k}\mathfrak{R}_m(\mathcal{H}) + \epsilon_k\right] \leq \exp(-2m\epsilon_k^2). \tag{5.43}$$

Choose $\epsilon_k = \epsilon + \sqrt{\frac{\log k}{m}}$, then, by the union bound, the following holds:

$$\begin{aligned}
&\mathbb{P}\left[\sup_{\substack{h\in H\\ k\geq 1}} R(h) - \widehat{R}_{S,\rho_k}(h) - \frac{2}{\rho_k}\mathfrak{R}_m(\mathcal{H}) - \epsilon_k > 0\right] \\
&\qquad \leq \sum_{k\geq 1} \exp(-2m\epsilon_k^2) \\
&\qquad = \sum_{k\geq 1} \exp\big[-2m(\epsilon + \sqrt{(\log k)/m})^2\big] \\
&\qquad \leq \sum_{k\geq 1} \exp(-2m\epsilon^2)\exp(-2\log k) \\
&\qquad = \Big(\sum_{k\geq 1} 1/k^2\Big)\exp(-2m\epsilon^2) \\
&\qquad = \frac{\pi^2}{6}\exp(-2m\epsilon^2) \leq 2\exp(-2m\epsilon^2).
\end{aligned}$$

We can choose $\rho_k = r/2^k$. For any $\rho \in (0, r]$, there exists $k \geq 1$ such that $\rho \in (\rho_k, \rho_{k-1}]$, with $\rho_0 = r$. For that k, $\rho \leq \rho_{k-1} = 2\rho_k$, thus $1/\rho_k \leq 2/\rho$ and $\sqrt{\log k} = \sqrt{\log \log_2(r/\rho_k)} \leq \sqrt{\log \log_2(2r/\rho)}$. Furthermore, for any $h \in \mathcal{H}$, $\widehat{R}_{S,\rho_k}(h) \leq \widehat{R}_{S,\rho}(h)$. Thus, the following inequality holds:

$$\mathbb{P}\left[\sup_{\substack{h\in H\\ \rho\in(0,r]}} R(h) - \widehat{R}_{S,\rho}(h) - \frac{4}{\rho}\mathfrak{R}_m(\mathcal{H}) - \sqrt{\frac{\log \log_2(2r/\rho)}{m}} - \epsilon > 0\right] \leq 2\exp(-2m\epsilon^2),$$

which proves the first statement. The second statement can be proven in a similar way. □

The Rademacher complexity of linear hypotheses with bounded weight vector can be bounded as follows.

Theorem 5.10 *Let $S \subseteq \{\mathbf{x} : \|\mathbf{x}\| \leq r\}$ be a sample of size m and let $\mathcal{H} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} : \|\mathbf{w}\| \leq \Lambda\}$. Then, the empirical Rademacher complexity of $\mathcal{H}$ can be bounded as follows:*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \sqrt{\frac{r^2\Lambda^2}{m}}\,.$$

Proof: The proof follows through a series of inequalities:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\mathcal{H}) &= \frac{1}{m}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|\mathbf{w}\|\leq\Lambda}\sum_{i=1}^{m}\sigma_i\mathbf{w}\cdot\mathbf{x}_i\right] = \frac{1}{m}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|\mathbf{w}\|\leq\Lambda}\mathbf{w}\cdot\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\right] \\
&\leq \frac{\Lambda}{m}\mathop{\mathbb{E}}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\right\|\right] \leq \frac{\Lambda}{m}\left[\mathop{\mathbb{E}}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\right\|^2\right]\right]^{\frac{1}{2}} \\
&= \frac{\Lambda}{m}\left[\mathop{\mathbb{E}}_{\sigma}\left[\sum_{i,j=1}^{m}\sigma_i\sigma_j(\mathbf{x}_i\cdot\mathbf{x}_j)\right]\right]^{\frac{1}{2}} \leq \frac{\Lambda}{m}\left[\sum_{i=1}^{m}\|\mathbf{x}_i\|^2\right]^{\frac{1}{2}} \leq \frac{\Lambda\sqrt{mr^2}}{m} = \sqrt{\frac{r^2\Lambda^2}{m}}\,,
\end{aligned}$$

The first inequality makes use of the Cauchy-Schwarz inequality and the bound on $\|\mathbf{w}\|$, the second follows by Jensen's inequality, the third by $\mathbb{E}[\sigma_i\sigma_j] = \mathbb{E}[\sigma_i]\,\mathbb{E}[\sigma_j] = 0$ for $i \neq j$, and the last one by $\|\mathbf{x}_i\| \leq r$. □

Combining theorem 5.10 and theorem 5.8 gives directly the following general margin bound for linear hypotheses with bounded weight vectors, presented in corollary 5.11.

Corollary 5.11 *Let $\mathcal{H} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} : \|\mathbf{w}\| \leq \Lambda\}$ and assume that $\mathcal{X} \subseteq \{\mathbf{x} : \|\mathbf{x}\| \leq r\}$. Fix $\rho > 0$, then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m, the following holds for any $h \in \mathcal{H}$:*

$$R(h) \leq \widehat{R}_{S,\rho}(h) + 2\sqrt{\frac{r^2\Lambda^2/\rho^2}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}\,. \tag{5.44}$$

As with theorem 5.8, the bound of this corollary can be generalized to hold uniformly for all $\rho \in (0, 1]$ at the cost of an additional term $\sqrt{\frac{\log\log_2\frac{2}{\rho}}{m}}$ by combining theorems 5.10 and 5.9.

This generalization bound for linear hypotheses is remarkable, since it does not depend directly on the dimension of the feature space, but only on the margin. It suggests that a small generalization error can be achieved when $\rho/(r\Lambda)$ is large (small second term) while the empirical margin loss is relatively small (first term). The latter occurs when few points are either classified incorrectly or correctly, but with margin less than ρ. When the training sample is linearly separable, for a linear hypothesis with geometric margin ρ_{geom} and the choice of the confidence margin

parameter $\rho = \rho_{\text{geom}}$, the empirical margin loss term is zero. Thus, if ρ_{geom} is relatively large, this provides a strong guarantee for the generalization error of the corresponding linear hypothesis.

The fact that the guarantee does not explicitly depend on the dimension of the feature space may seem surprising and appears to contradict the VC-dimension lower bounds of theorems 3.20 and 3.23. Those lower bounds show that for any learning algorithm $\mathcal{A}$ there exists a *bad* distribution for which the error of the hypothesis returned by the algorithm is $\Omega(\sqrt{d/m})$ with a non-zero probability. The bound of the corollary does not rule out such *bad* cases, however: for such bad distributions, the empirical margin loss would be large even for a relatively small margin ρ, and thus the bound of the corollary would be loose in that case.

Thus, in some sense, the learning guarantee of the corollary hinges upon the hope of a good margin value ρ: if there exists a relatively large margin value $\rho > 0$ for which the empirical margin loss is small, then a small generalization error is guaranteed by the corollary. This favorable margin situation depends on the distribution: while the learning bound is distribution-independent, the existence of a good margin is in fact distribution-dependent. A favorable margin seems to appear relatively often in applications.

The bound of the corollary gives a strong justification for margin-maximization algorithms such as SVMs. Choosing $\Lambda = 1$, by the generalization of corollary 5.11 to a uniform bound over $\rho \in (0, r]$, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $h \in \left\{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\| \leq 1\right\}$ and $\rho \in (0, r]$:

$$R(h) \leq \widehat{R}_{S,\rho}(h) + 4\sqrt{\frac{r^2/\rho^2}{m}} + \sqrt{\frac{\log \log_2 \frac{2r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}\,.$$

The inequality also trivially holds for ρ larger than r since in that case, by the Cauchy-Schwarz inequality, for any $\mathbf{w}$ with $\|\mathbf{w}\| \leq 1$, we have $y_i(\mathbf{w} \cdot \mathbf{x}_i) \leq r \leq \rho$ and $\widehat{R}_{S,\rho}(h)$ is equal to one for all h.

Now, for any $\rho > 0$, the ρ-margin loss function is upper bounded by the ρ-hinge loss:

$$\forall u \in \mathbb{R}, \Phi_\rho(u) = \min\left(1, \max\left(0, 1 - \frac{u}{\rho}\right)\right) \leq \max\left(0, 1 - \frac{u}{\rho}\right). \tag{5.45}$$

Thus, with probability at least $1 - \delta$, the following holds for all $h \in \left\{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\| \leq 1\right\}$ and all $\rho > 0$:

$$R(h) \leq \frac{1}{m}\sum_{i=1}^{m} \max\left(0, 1 - \frac{y_i(\mathbf{w} \cdot \mathbf{x}_i)}{\rho}\right) + 4\sqrt{\frac{r^2/\rho^2}{m}} + \sqrt{\frac{\log \log_2 \frac{2r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}\,.$$

Since for any $\rho > 0$, h/ρ admits the same generalization error as h, with probability at least $1-\delta$, the following holds for all $h \in \left\{\mathbf{x} \mapsto \mathbf{w}\cdot\mathbf{x} \colon \|\mathbf{w}\| \leq 1/\rho\right\}$ and all $\rho > 0$:

$$R(h) \leq \frac{1}{m}\sum_{i=1}^{m} \max\left(0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i)\right) + 4\sqrt{\frac{r^2/\rho^2}{m}} + \sqrt{\frac{\log\log_2 \frac{2r}{\rho}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}\,. \quad (5.46)$$

This inequality can be used to derive an algorithm that selects $\mathbf{w}$ and $\rho > 0$ to minimize the right-hand side. The minimization with respect to ρ does not lead to a convex optimization and depends on theoretical constant factors affecting the second and third terms, which may not be optimal. Thus, instead, ρ is left as a free parameter of the algorithm, typically determined via cross-validation.

Now, since only the first term of the right-hand side depends on $\mathbf{w}$, for any $\rho > 0$, the bound suggests selecting $\mathbf{w}$ as the solution of the following optimization problem:

$$\min_{\|\mathbf{w}\|^2 \leq \frac{1}{\rho^2}} \frac{1}{m}\sum_{i=1}^{m} \max\left(0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i)\right). \quad (5.47)$$

Introducing a Lagrange variable $\lambda \geq 0$, the optimization problem can be written equivalently as

$$\min_{\mathbf{w}} \lambda\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \max\left(0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i)\right). \quad (5.48)$$

Since for any choice of ρ in the constraint of (5.47) there exists an equivalent dual variable λ in the formulation of (5.48) that achieves the same optimal $\mathbf{w}$, λ can be freely selected via cross-validation.[5] The resulting algorithm precisely coincides with SVMs. Note that an alternative objective function and thus algorithm would be based on the empirical margin loss instead of the hinge loss. However, the advantage of the hinge loss is that it is convex, while the margin loss is not.

As already pointed out, the bounds just discussed do not directly depend on the dimension of the feature space but guarantee good generalization when given a favorable margin. Thus, they suggest seeking large-margin separating hyperplanes in a very high-dimensional space. In view of the form of the dual optimization problems for SVMs, determining the solution of the optimization and using it for prediction both require computing many inner products in that space. For very high-dimensional spaces, the computation of these inner products could become very costly. The next chapter provides a solution to this problem which further provides a generalization of SVMs to non-vectorial input spaces.

[5] An equivalent analysis consists of choosing $\rho = 1/\|\mathbf{w}\|$ in (5.46).

5.5 Chapter notes

The maximum-margin or *optimal hyperplane* solution described in section 5.2 was introduced by Vapnik and Chervonenkis [1964]. The algorithm had limited applications since in most tasks in practice the data is not linearly separable. In contrast, the SVM algorithm of section 5.3 for the general non-separable case, introduced by Cortes and Vapnik [1995] under the name *support-vector networks*, has been widely adopted and been shown to be effective in practice. The algorithm and its theory have had a profound impact on theoretical and applied machine learning and inspired research on a variety of topics. Several specialized algorithms have been suggested for solving the specific QP that arises when solving the SVM problem, for example the SMO algorithm of Platt [1999] (see exercise 5.4) and a variety of other decomposition methods such as those used in the LibLinear software library [Hsieh et al., 2008], and [Allauzen et al., 2010] for solving the problem when using rational kernels (see chapter 6).

Much of the theory supporting the SVM algorithm ([Cortes and Vapnik, 1995, Vapnik, 1998]), in particular the margin theory presented in section 5.4, has been adopted in the learning theory and statistics communities and applied to a variety of other problems. The margin bound on the VC-dimension of canonical hyperplanes (exercise 5.7) is by Vapnik [1998], the proof is very similar to Novikoff's margin bound on the number of updates made by the Perceptron algorithm in the separable case. Our presentation of margin guarantees based on the Rademacher complexity follows the elegant analysis of Koltchinskii and Panchenko [2002] (see also Bartlett and Mendelson [2002], Shawe-Taylor et al. [1998]). Our proof of Talagrand's lemma 5.7 is a simpler and more concise version of a more general result given by Ledoux and Talagrand [1991, pp. 112–114]. See Höffgen et al. [1995] for hardness results related to the problem of finding a hyperplane with the minimal number of errors on a training sample.

5.6 Exercises

5.1 Soft margin hyperplanes. The function of the slack variables used in the optimization problem for soft margin hyperplanes has the form: $\xi \mapsto \sum_{i=1}^{m} \xi_i$. Instead, we could use $\xi \mapsto \sum_{i=1}^{m} \xi_i^p$, with $p > 1$.

(a) Give the dual formulation of the problem in this general case.

(b) How does this more general formulation ($p > 1$) compare to the standard setting ($p = 1$)? In the case $p = 2$ is the optimization still convex?

5.2 Tighter Rademacher Bound. Derive the following tighter version of the bound of theorem 5.9: for any $\delta > 0$, with probability at least $1 - \delta$, for all $h \in \mathcal{H}$ and $\rho \in (0, 1]$ the following holds:

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \frac{2\gamma}{\rho}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\log_\gamma \frac{\gamma}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} \tag{5.49}$$

for any $\gamma > 1$.

5.3 Importance weighted SVM. Suppose you wish to use SVMs to solve a learning problem where some training data points are more important than others. More formally, assume that each training point consists of a triplet (x_i, y_i, p_i), where $0 \leq p_i \leq 1$ is the importance of the ith point. Rewrite the primal SVM constrained optimization problem so that the penalty for mis-labeling a point x_i is scaled by the priority p_i. Then carry this modification through the derivation of the dual solution.

5.4 Sequential minimal optimization (SMO). The SMO algorithm is an optimization algorithm introduced to speed up the training of SVMs. SMO reduces a (potentially) large quadratic programming (QP) optimization problem into a series of small optimizations involving only two Lagrange multipliers. SMO reduces memory requirements, bypasses the need for numerical QP optimization and is easy to implement. In this question, we will derive the update rule for the SMO algorithm in the context of the dual formulation of the SVM problem.

(a) Assume that we want to optimize equation (5.33) only over α_1 and α_2. Show that the optimization problem reduces to

$$\max_{\alpha_1,\alpha_2} \underbrace{\alpha_1 + \alpha_2 - \frac{1}{2}K_{11}\alpha_1^2 - \frac{1}{2}K_{22}\alpha_2^2 - sK_{12}\alpha_1\alpha_2 - y_1\alpha_1 v_1 - y_2\alpha_2 v_2}_{\Psi_1(\alpha_1,\alpha_2)}$$
$$\text{subject to: } 0 \leq \alpha_1, \alpha_2 \leq C \wedge \alpha_1 + s\alpha_2 = \gamma\,,$$

where $\gamma = y_1 \sum_{i=3}^m y_i\alpha_i$, $s = y_1 y_2 \in \{-1, +1\}$, $K_{ij} = (\mathbf{x}_i \cdot \mathbf{x}_j)$ and $v_i = \sum_{j=3}^m \alpha_j y_j K_{ij}$ for $i = 1, 2$.

(b) Substitute the linear constraint $\alpha_1 = \gamma - s\alpha_2$ into Ψ_1 to obtain a new objective function Ψ_2 that depends only on α_2. Show that the α_2 that maximizes Ψ_2 (without the constraints $0 \leq \alpha_1, \alpha_2 \leq C$) can be expressed as

$$\alpha_2 = \frac{s(K_{11} - K_{12})\gamma + y_2(v_1 - v_2) - s + 1}{\eta}\,,$$

where $\eta = K_{11} + K_{22} - 2K_{12}$.

(c) Show that

$$v_1 - v_2 = f(\mathbf{x}_1) - f(\mathbf{x}_2) + \alpha_2^* y_2 \eta - s y_2 \gamma (K_{11} - K_{12})$$

where $f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*$ and α_i^* are values for the Lagrange multipliers prior to optimization over α_1 and α_2 (similarly, b^* is the previous value for the offset).

(d) Show that

$$\alpha_2 = \alpha_2^* + y_2 \frac{(y_2 - f(\mathbf{x}_2)) - (y_1 - f(\mathbf{x}_1))}{\eta}.$$

(e) For $s = +1$, define $L = \max\{0, \gamma - C\}$ and $H = \min\{C, \gamma\}$ as the lower and upper bounds on α_2. Similarly, for $s = -1$, define $L = \max\{0, -\gamma\}$ and $H = \min\{C, C - \gamma\}$. The update rule for SMO involves "clipping" the value of α_2, i.e.,

$$\alpha_2^{clip} = \begin{cases} \alpha_2 & \text{if } L < \alpha_2 < H \\ L & \text{if } \alpha_2 \leq L \\ H & \text{if } \alpha_2 \geq H \end{cases}.$$

We subsequently solve for α_1 such that we satisfy the equality constraint, resulting in $\alpha_1 = \alpha_1^* + s(\alpha_2^* - \alpha_2^{clip})$. Why is "clipping" is required? How are L and H derived for the case $s = +1$?

5.5 SVMs hands-on.

(a) Download and install the `libsvm` software library from:

`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

(b) Download the `satimage` data set found at:

`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Merge the training and validation sets into one. We will refer to the resulting set as the training set from now on. Normalize both the training and test vectors.

(c) Consider the binary classification that consists of distinguishing class 6 from the rest of the data points. Use SVMs combined with polynomial kernels (see chapter 6) to solve this classification problem. To do so, randomly split the training data into ten equal-sized disjoint sets. For each value of the polynomial degree, $d = 1, 2, 3, 4$, plot the average cross-validation error plus or minus one standard deviation as a function of C (let the other parameters of polynomial kernels in `libsvm`, γ and c, be equal to their default values 1).

Report the best value of the trade-off constant C measured on the validation set.

(d) Let (C^*, d^*) be the best pair found previously. Fix C to be C^*. Plot the ten-fold cross-validation training and test errors for the hypotheses obtained as a function of d. Plot the average number of support vectors obtained as a function of d.

(e) How many of the support vectors lie on the margin hyperplanes?

(f) In the standard two-group classification, errors on positive or negative points are treated in the same manner. Suppose, however, that we wish to penalize an error on a negative point (false positive error) $k > 0$ times more than an error on a positive point. Give the dual optimization problem corresponding to SVMs modified in this way.

(g) Assume that k is an integer. Show how you can use `libsvm` without writing any additional code to find the solution of the modified SVMs just described.

(h) Apply the modified SVMs to the classification task previously examined and compare with your previous SVMs results for $k = 2, 4, 8, 16$.

5.6 Sparse SVM. One can give two types of arguments in favor of the SVM algorithm: one based on the sparsity of the support vectors, another based on the notion of margin. Suppose that instead of maximizing the margin, we choose instead to maximize sparsity by minimizing the L_p norm of the vector $\boldsymbol{\alpha}$ that defines the weight vector $\mathbf{w}$, for some $p \geq 1$. First, consider the case $p = 2$. This gives the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, b} \ & \frac{1}{2} \sum_{i=1}^{m} \alpha_i^2 + C \sum_{i=1}^{m} \xi_i \\ \text{subject to } & y_i \Big(\sum_{j=1}^{m} \alpha_j y_j \mathbf{x}_i \cdot \mathbf{x}_j + b \Big) \geq 1 - \xi_i, i \in [m] \\ & \xi_i, \alpha_i \geq 0, i \in [m]. \end{aligned} \tag{5.50}$$

(a) Show that modulo the non-negativity constraint on $\boldsymbol{\alpha}$, the problem coincides with an instance of the primal optimization problem of SVM.

(b) Derive the dual optimization of problem of (5.50).

(c) Setting $p = 1$ will induce a more sparse $\boldsymbol{\alpha}$. Derive the dual optimization in this case.

5.7 VC-dimension of canonical hyperplanes. The objective of this problem is derive a bound on the VC-dimension of canonical hyperplanes that does not depend on

the dimension of feature space. Let $S \subseteq \{\mathbf{x}\colon \|\mathbf{x}\| \leq r\}$. We will show that the VC-dimension d of the set of canonical hyperplanes $\{x \mapsto \operatorname{sgn}(\mathbf{w}\cdot\mathbf{x})\colon \min_{x\in S} |\mathbf{w}\cdot\mathbf{x}| = 1 \wedge \|\mathbf{w}\| \leq \Lambda\}$ verifies

$$d \leq r^2\Lambda^2\,. \tag{5.51}$$

(a) Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ be a set that can be shattered. Show that for all $\mathbf{y} = (y_1, \ldots, y_d) \in \{-1, +1\}^d$, $d \leq \Lambda\| \sum_{i=1}^d y_i\mathbf{x}_i\|$.

(b) Use randomization over the labels $\mathbf{y}$ and Jensen's inequality to show that $d \leq \Lambda\sqrt{\sum_{i=1}^d \|\mathbf{x}_i\|^2}$.

(c) Conclude that $d \leq r^2\Lambda^2$.