2DI70 - Statistical Learning Theory
Lecture Notes

Rui Castro

April 3, 2018

Contents

Abstract

These notes are work in progress, and are being adapted from lecture notes from a course the author taught at Columbia University. These are based on various materials, and in particular notes developed during a reading group in the University of Wisconsin - Madison (which was coordinated by Robert Nowak). Any comments and remarks are most welcome. The author wants to thank Ervin Tánczos and Sándor Kolumbán for helping in the revision of the notes and the correction of typos. Naturally, the author of the current version is the sole responsible for the errors it contains.

Chapter 1

Introduction

In this chapter we give a very short introduction of the elements of statistical learning theory, and set the stage for the subsequent chapters. We take a probabilistic approach to learning, as it provides a good framework to cope with the uncertainty inherent to any dataset.

1.1 Learning from Data

We begin with an illustrative example.

Example 1.1.1 (Classification of hand-written digits) *A classical problem in machine learning is the automated classification of handwritten digits. The main goal is to construct a classifier (that is, a function) that takes an image of an handwritten digit (e.g., a* 128×128 *pixel image) and outputs the corresponding digit (e.g.,* $0, 1, 2, \ldots, 9$*). Obviously this is not such an easy task, as different persons have different styles of writing, and there are digits that are remarkably similar (e.g., 1 and 7 or 4 and 9). The* supervised *learning approach to this problem is to take a large set of labeled example (pairs of handwritten digits and the corresponding digit, annotated by an expert) and use this dataset to automatically construct a classifier that* generalizes *well - meaning, it works well in examples outside the dataset.*

The typical architecture of such a system goes as follows. The raw data (and image, a video sequence, etc...) is first pre-processed in a smart way to extract important features (e.g., extract the salient edges from the handwritten digit images). This is then used for classification/regression. It is the last step we are mostly concerned with. In recent years, and in light of the advances in computational power, it is possible to "feed" the classifiers with the raw data, and learn what features are most useful - in what is generically known as "deep learning".

To formulate the basic learning from data problem, we must specify several basic elements: data spaces, probability measures, loss functions, and statistical risk.

1.1.1 Data Spaces

From this point on we assume the raw data has been possibly processed, and this is what we have available. Learning from data begins with a specification of two spaces:

$$\mathcal{X} \equiv \text{Input Space}$$

$$\mathcal{Y} \equiv \text{Output Space}$$

The input space is also sometimes called the "feature space" or "signal domain". In statistics and regression models this is referred to as the space of covariates. The output space is also called the "label space", "outcome space", "signal range", or in statistical regression the response space.

Example 1.1.2 *$\mathcal{X} = [0,1]^{128\times 128}$ and $\mathcal{Y} = \{0,1,\ldots,9\}$ fit the example above, where the input space indicates is that of grayscale handwritten images and the output space is that of digits.*

Example 1.1.3 *$\mathcal{X} = \mathbb{R}$ is a one-dimensional signal domain (e.g., time domain) and $\mathcal{Y} = \mathbb{R}$ corresponds to a real valued signal. A classical example is the estimation of a signal f in noise, using data*

$$Y_i = f(X_i) + W_i \ ,$$

where $\{(X_i, Y_i)\}_{i=1}^n$ is our training data, W_i corresponds to some additive noise, and f is the signal we would like to estimate.

1.1.2 Loss Functions

Since we are trying to predict/classify labels we need to measure the performance of our learner in some way. Suppose we have a true label $y \in \mathcal{Y}$ and a label prediction $\hat{y} \in \mathcal{Y}$. A loss function measures how "different" are these two quantities. Formally, a loss function is a map

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Example 1.1.4 *In binary classification problems, $\mathcal{Y} = \{0,1\}$. The $0/1$ loss function is a popular choice.*

$$\ell(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\} \ ,$$

where $\mathbf{1}\{A\}$ is the indicator function which takes a value of 1 if the logical condition A is true and zero otherwise. We typically will compare a true label y with a prediction $\hat{y}$, in which case the $0/1$ loss simply counts misclassifications. Note this is a symmetric function in the arguments, so the two possible types of errors have the same weight. Sometimes, this is not adequate, as in the next example.

Example 1.1.5 (Spam email classification) *When classifying spam emails one is much more inclined to tolerate a few misses (i.e., classifying a spam email as legit), than incurring false alarms (i.e., classifying legit emails as spam). Therefore, the typical loss function in this case is not symmetric, but rather something like*

$$\ell(\hat{y}, y) = \begin{cases} 5 & \text{if } \hat{y} = 1, y = 0 \\ 1 & \text{if } \hat{y} = 0, y = 1 \\ 0 & \text{otherwise} \end{cases} \ ,$$

where label 0 and 1 correspond to legit and spam email, respectively.

Example 1.1.6 *In regression or estimation problems, $\mathcal{Y} = \mathbb{R}$. The squared error loss function is often employed.*

$$\ell(\hat{y}, y) = (\hat{y} - y)^2 \ .$$

This function is symmetric in the arguments, and has many practical and theoretical advantages - it is continuous, infinitely differentiable and convex on both arguments. It also provides a natural way to compare an estimate $\hat{y}$ with the true value y that penalizes big differences between the two.

The loss function can be used to measure the "risk" of a learning rule. It is, however, not immediately clear how to do so. Measuring the loss incurred on the training dataset is not particularly useful if one is trying to characterize the performance in test examples that are not part of the training set. This is where probability comes to the rescue, and provides an interesting framework to cast our problem.

1.1.3 Probability Measure and Expectation

Define a joint probability distribution on $\mathcal{X} \times \mathcal{Y}$ denoted $\mathbb{P}_{XY}$. Let (X, Y) denote a pair of random variables distributed according to $\mathbb{P}_{XY}$. This distribution can alternatively be described by the marginal distribution of X and the conditional distribution of Y given X (sometimes this description is more convenient) — let $\mathbb{P}_X$ denote the marginal distribution on X, and let $\mathbb{P}_{Y|X}$ denote the conditional distribution of Y given X. For any distribution P, let p denote its density function with respect to the corresponding dominating measure; e.g., *Lebesgue measure* for continuous random variables or *counting measure* for discrete random variables.

Finally define the expectation operator as

$$\mathbb{E}[f(X, Y)] \equiv \int f(x, y) d\mathbb{P}_{XY}(x, y) = \int f(x, y) p_{XY}(x, y) dx dy \ .$$

Later we will assume our training data is an independent and identically distributed sample from such distributions. This captures the idea that features and labels are related somehow, but there is some uncertainty. Furthermore, since we assume the training examples are independent, this means that information from one training example does not help us to predict the other examples if we already know $\mathbb{P}_{XY}$.

Remark 1.1.1 *Whenever clear from the context we will drop the subscripts in this notation. For instance $\mathbb{P}_{Y|X}(Y = 1|X = x) \equiv \mathbb{P}(Y = 1|X = x)$.*

1.1.4 Statistical Risk

The basic problem in learning is to determine a mapping $f : \mathcal{X} \to \mathcal{Y}$ that takes an input $x \in \mathcal{X}$ and predicts the corresponding output $y \in \mathcal{Y}$ as well as possible. A way to measure the risk is to assume that a pair feature/label comes from the distribution $\mathbb{P}_{XY}$ and see how the learning rule does on average. This gives rise to the notion of statistical risk

Definition 1.1.1 (Statistical Risk) *For a prediction rule f and a joint distribution of features and labels $\mathbb{P}_{XY}$ the statistical risk of f is defined as*

$$R(f) \equiv \mathbb{E}\left[\ell(f(X), Y) \mid f\right] \ ,$$

where $(X, Y) \sim \mathbb{P}_{XY}$. The conditional statement is there to ensure the definition is sensible even if f is a random quantity.

The risk tells us how well, on average, the predictor f performs with respect to the chosen loss function. To judge the performance of a particular learning rule, it makes sense to compare it to the best possible learning rule. Hence the minimum risk value is a key quantity of interest. It is defined as

$$R^* = \inf_{f \text{measurable}} R(f)$$

where the infimum is taking over all measurable functions (meaning these are nice functions where you can compute the expectation).

1.1.5 The Learning Problem

We now have all the ingredients to formulate the learning problem. Suppose we are asked to construct a prediction rule that has good performance for the labeled example (X, Y) distributed according to $\mathbb{P}_{XY}$ ($(X, Y) \sim \mathbb{P}_{XY}$ for short). Our goal is to find a map f so that $f(X) \approx Y$ with high probability. Ideally, we would choose f to minimize the risk $R(f) = \mathbb{E}[\ell(f(X), Y)]$. However, in order to compute the risk (and hence optimize it) we need to know the joint distribution $\mathbb{P}_{XY}$. In many problems of practical interest, the joint distribution is unknown, and directly minimizing the risk is not possible. What can we do then? We can try to learn a good prediction rule f by using training data!

Suppose we have a training dataset, that is, a collection of labeled examples we can use to find a good prediction rule that works for examples we have not seen yet. Here we are going to make what is, perhaps, the strongest assumption of this setting. Let's assume the training examples are samples from the unknown distribution $\mathbb{P}_{XY}$. Specifically, suppose you are given n samples $\{X_i, Y_i\}_{i=1}^n$ independently and identically distributed (i.i.d.) according to the otherwise unknown distribution $\mathbb{P}_{XY}$. These are called *training data*. For simplicity, denote this collection by $D_n \equiv \{X_i, Y_i\}_{i=1}^n$.

Are these reasonable assumptions? In many cases yes. Consider for instance the example of classification of handwritten digits. Suppose you have a very big "bag" containing all the handwritten doodles of digits that you might ever observe. Obviously we do not have assess to this, but only a sample of the elements in the bag. This sample is your training set. If these digits come from different people (chosen randomly) then the i.i.d. assumption is not so far-fetched.

The independence between the samples captures the idea that these come from different sources, and that these do not carry information about each other if we have access to $\mathbb{P}_{XY}$. The identically distributed assumption captures the idea that these are somehow a representative sample of the type of instances we might encounter in the future. In the context of handwritten digit classification independence captures the notions that our dataset consists of digits written by different people at different times, and identically distributed samples means that we'll use the learned rule to classify similar types of handwritten digits. This assumption would be unreasonable if our training set consisted mainly of male writers, and we want to use the learned rule for female writers.

Suppose we have a class of candidate prediction rules $\mathcal{F}$ (e.g., linear classifiers, neural networks, etc...). Our goal is to use the training data D_n to choose a mapping out of $\mathcal{F}$ that is,

hopefully, a good one. Formally, we want to choose a map $\hat{f}_n \in \mathcal{F}$. It is important to note that the selected model $\hat{f}_n$ is a function of the training data

$$\hat{f}_n(x) \equiv f(x; D_n) \ .$$

The subscript symbol n denotes the dependence on the training set size, and the "hat" indicates this is a function of the training dataset. This is to avoid having to represent this dependence explicitly so that we avoid unnecessary notational clutter. Note that $\hat{f}_n$ is a random function (since it is a function of D_n, which is a random vector).

How do we measure the performance of this rule? Again, our assumptions come to the rescue. Suppose there is another instance $(X, Y) \sim \mathbb{P}_{XY}$, but only the feature part X is revealed to us - this is what is known as a test example. Can we predict Y well using our learned rule? In other words, is $\ell(\hat{f}_n(X), Y)$ small? We would like this to be small regardless of which test example we get. This is typically not possible, but we can maybe say the loss is small on average, or it is small "most of the times". Let's try to formalize this notion.

Recall that the statistical risk of $\hat{f}_n$ is defined as

$$R(\hat{f}_n) \equiv \mathbb{E}\left[\ell(\hat{f}_n(X), Y)\middle| \hat{f}_n\right] \ .$$

Note that, $R(\hat{f}_n)$ in this definition is still a random quantity (since it is a function of $\hat{f}_n$, which is a function of D_n). A different way to write this risk is as

$$R(\hat{f}_n) = \mathbb{E}_{XY}\left[\ell(\hat{f}_n(X), Y)\right] \ ,$$

where the expectation is taken *only* with respect to X and Y. This notation is sometimes useful to understand the concepts, but it can easily lead to problems in more complicated settings, so we will avoid it as much as possible, and use only the notation of conditional expectations[1].

Most of the material in these notes is concerned with the study of the risk as defined above. Namely we are interested in guaranteeing that $R(\hat{f}_n)$ is small with very high probability over the distribution of the training data (again, recall that $R(\hat{f}_n)$ is a random variable). Sometimes we will be content in guaranteeing the *expected risk* is small, computed over random realizations of the training data.

Definition 1.1.2 (Expected Risk) *The expected risk of a prediction rule is given by* $\mathbb{E}[R(\hat{f}_n)]$. *Note that, the expectation is over the training data and any randomized choices of the learning algorithm.*

Our hope is that $\hat{f}_n$ has a small expected risk, given enough training data. The notion of expected risk can be interpreted as follows: we would like to define an algorithm (a model selection process) that performs well on average, over any random sample of n training data. The expected risk is a measure of the expected performance of the algorithm with respect to the chosen loss function. That is, we are not gauging the risk of a particular map $f \in \mathcal{F}$, but rather we are measuring the performance of the algorithm that takes any realization of training

[1] Let X and Y be random variables. The notation $E[Y|X]$ indicates a conditional expectation, the expectation of Y given X, therefore this is a random variable, in fact $E[Y|X] = g(X)$ for some function g. A very important fact is that the expectation of a conditional expectation removes the conditioning. That is $\mathbb{E}[\mathbb{E}[Y|X]] = \mathbb{E}[Y]$. This is known as the *law of total expectation*, and we will use it often.

data $\{X_i, Y_i\}_{i=1}^n$ and selects an appropriate model in $\mathcal{F}$. In other words, we are studying the performance of algorithms, not of specific prediction rules that were derived from the data.

Perhaps surprisingly, we can make main statements in this regard without making any assumptions about $\mathbb{P}_{XY}$. So the data can be produced by ANY distribution whatsoever! The main challenges concern with the choice of "good" spaces of prediction rules $\mathcal{F}$ and how can we design useful and effective model selection algorithms, as we will see in the subsequent chapters.

Chapter 2

Binary Classification and Regression

In this chapter we will consider two specific settings that are very commonly used, namely binary classification and regression. This will help you to get a good grip of the basic concepts. Recall that the our goal is to use training data to learn a prediction rule, that is, a map from the feature space $\mathcal{X}$, to a label space $\mathcal{Y}$.

2.1 Binary Classification

This is, in a sense, the simplest type of classification problem, where we have only two possible labels. This is actually a very common setting. For instance, emails can be classified as legit or spam. A patient in an hospital is either healthy or sick, etc... In all these scenarios we can abstract the possible labels as $\mathcal{Y} = \{0, 1\}$, so this will be the label space in this section. The feature space obviously depends on the specific problem. For spam filtering it might be a bag-of-words, for the patient diagnosis it might be a collection of measurements collected by doctors (e.g., heart rate, blood pressure, presence of specific symptoms). For concreteness let's consider that the feature space is a subset of $\mathbb{R}^d$ (although this is not so important in what follows). Finally, let's consider the familiar 0/1 loss function

$$\ell(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\} = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{if } \hat{y} = y \end{cases} .$$

In this setting a prediction rule f is generally referred to as a classification rule. Recall that the risk of any prediction rule is defined simply as the expected value of the loss function for an example $(X, Y) \sim \mathbb{P}_{XY}$,

$$R(f) = \mathbb{E}\left[\ell(f(X), Y)\right] = \mathbb{E}\left[\mathbf{1}\{f(X) \neq Y\}\right] = \mathbb{P}(f(X) \neq Y) .$$

So the risk based on the 0/1 loss function is simply the probability of making errors. This is quite a natural way to measure performance of classifiers. Note that in the above we are assuming f is not random, without loss of generality (otherwise you need to include a conditioning on f, which will only overburden the notation).

Naturally we would like to compare the performance of a classifier to that of the best possible classifier (as we cannot hope to do better than this), given the knowledge of $\mathbb{P}_{XY}$. The performance of the best classification rule is known as the *Bayes Risk*.

Definition 2.1.1 *The Bayes risk is the infimum of the risk for all classifiers*

$$R^* = \inf_{f measurable} R(f) \ .$$

For the classification setting described above we can actually explicitly describe a classifier that attains the Bayes risk, known as the Bayes classifier.

Definition 2.1.2 (Bayes Classifier) *The Bayes classifier is the following mapping:*

$$f^*(x) = \mathbf{1}\{\eta(x) \geq 1/2\} = \begin{cases} 1 & \text{if } \eta(x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases} \ ,$$

where

$$\eta(x) \equiv \mathbb{P}_{Y|X}(Y = 1|X = x)$$

is called the feature conditional probability.

This is actually quite intuitive. If, for a given that $X = x$, you know that the probability that $Y = 1$ is larger than that of $Y = 0$ you should predict the label is actually 1, and vice-versa. So the Bayes classifier predicts $Y = 1$ if $\mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = 0|X = x)$, and zero otherwise. We are ready to state our first result.

Proposition 2.1.1 *The Bayes classifier attains the Bayes risk, namely*

$$R(f^*) = R^* \ .$$

Proof Let $g : \mathcal{X} \to \mathcal{Y}$ be any classifier. We will show that $R(g) - R(f^*) \geq 0$. This implies that no classifier can better than the Bayes classifier, therefore, it attains the minimum risk. Note that

$$\begin{aligned} R(g) - R(f^*) &= \mathbb{P}(g(X) \neq Y) - \mathbb{P}(f^*(X) \neq Y) \\ &= \int_{\mathcal{X}} (\mathbb{P}(g(X) \neq Y|X = x) - \mathbb{P}(f^*(X) \neq Y|X = x))\, d\mathbb{P}_X(x) \ . \end{aligned} \tag{2.1}$$

We will show that

$$\mathbb{P}(g(X) \neq Y|X = x) - \mathbb{P}(f^*(X) \neq Y|X = x) \geq 0 \ ,$$

which implies immediately that $R(g) - R(f^*) \geq 0$. For any classifier g

$$\begin{aligned} &\mathbb{P}(g(X) \neq Y|X = x) \\ &= \mathbb{P}\left(Y = 1, g(X) = 0|X = x\right) + \mathbb{P}\left(Y = 0, g(X) = 1|X = x\right) \\ &= \mathbb{E}[\mathbf{1}\{Y = 1\}\mathbf{1}\{g(X) = 0\}|X = x] + \mathbb{E}[\mathbf{1}\{Y = 0\}\mathbf{1}\{g(X) = 1\}|X = x] \\ &= \mathbf{1}\{g(x) = 0\}\mathbb{E}[\mathbf{1}\{Y = 1\}|X = x] + \mathbf{1}\{g(x) = 1\}\mathbb{E}[\mathbf{1}\{Y = 0\}|X = x] \\ &= \mathbf{1}\{g(x) = 0\}\mathbb{P}(Y = 1|X = x) + \mathbf{1}\{g(x) = 1\}\mathbb{P}(Y = 0|X = x) \\ &= \mathbf{1}\{g(x) = 0\}\eta(x) + \mathbf{1}\{g(x) = 1\}(1 - \eta(x)) \\ &= \mathbf{1}\{g(x) = 0\}(2\eta(x) - 1) + (1 - \eta(x)) \ , \end{aligned}$$

where the last equality follows by noting that $\mathbf{1}\{g(x) = 0\} = 1 - \mathbf{1}\{g(x) = 1\}$. The above equality makes it easy to write the difference between the conditional probability of error of classifier g and the Bayes classifer.

$$\begin{aligned}
&\mathbb{P}(g(X) \neq Y | X = x) - \mathbb{P}(f^*(X) \neq Y | X = x) \\
&= \mathbf{1}\{g(x) = 0\}(2\eta(x) - 1) + (1 - \eta(x)) - [\mathbf{1}\{f^*(x) = 0\}(2\eta(x) - 1) + (1 - \eta(x))] \\
&= \mathbf{1}\{g(x) = 0\}(2\eta(x) - 1) - \mathbf{1}\{f^*(x) = 0\}(2\eta(x) - 1) \\
&= \left(\mathbf{1}\{f^*(x) = 1\} - \mathbf{1}\{g(x) = 1\}\right)(2\eta(x) - 1) \ .
\end{aligned}$$

We are almost done. Recall that the Bayes classifier predicts label one if $\eta(x) \geq 1/2$ and zero otherwise. Therefore, if x is such that $\eta(x) \geq 1/2$ we have

$$\underbrace{\left(\underbrace{\mathbf{1}\{f^*(x) = 1\}}_{1} - \underbrace{\mathbf{1}\{g(x) = 1\}}_{0 \text{ or } 1}\right)}_{\geq 0} \cdot \underbrace{(2\eta(x) - 1)}_{\geq 0} \geq 0 \ .$$

Analogously, if x is such that $\eta(x) < 1/2$ we have

$$\underbrace{\left(\underbrace{\mathbf{1}\{f^*(x) = 1\}}_{0} - \underbrace{\mathbf{1}\{g(x) = 1\}}_{0 \text{ or } 1}\right)}_{\leq 0} \cdot \underbrace{(2\eta(x) - 1)}_{< 0} \geq 0 \ .$$

So, regardless of the value of x we conclude that

$$\mathbb{P}(g(X) \neq Y | X = x) - \mathbb{P}(f^*(X) \neq Y | X = x) = \left(\mathbf{1}\{f^*(x) = 1\} - \mathbf{1}\{g(x) = 1\}\right)(2\eta(x) - 1) \geq 0 \ .$$

Plugging in this in (2.1) concludes the proof. □

Note that while the Bayes classifier achieves the Bayes risk, this rule needs the knowledge of the joint distribution of (X, Y), that is not generally available to us.

It is also interesting and useful to write the *excess risk* $R(g) - R^*$ using (2.1).

Lemma 2.1.1 *Any classifier $g : \mathcal{X} \to \mathcal{Y}$ can be written as $g(x) = \mathbf{1}\{x \in G\}$ for some set G. Let $G^* = \{x \in \mathcal{X} : \eta(x) \geq 1/2\}$ be the set corresponding to the Bayes' classifier. The* excess risk *is given by*

$$R(g) - R^* = \int_{G \Delta G^*} |2\eta(x) - 1| d\mathbb{P}_X(x) \ ,$$

where $G \Delta G^* = (G \cap \bar{G}^*) \cup (\bar{G} \cap G^*)$ is the symmetric set difference between the sets G and G^*.

Proof From (2.1) we have

$$\begin{aligned}
R(g) - R^* &= \int_{\mathcal{X}} \left(\mathbf{1}\{f^*(x) = 1\} - \mathbf{1}\{g(x) = 1\}\right)(2\eta(x) - 1) d\mathbb{P}_X(x) \\
&= \int_{\mathcal{X}} |2\eta(x) - 1| \left|\mathbf{1}\{f^*(x) = 1\} - \mathbf{1}\{g(x) = 1\}\right| d\mathbb{P}_X(x) \\
&= \int_{\mathcal{X}} |2\eta(x) - 1| \mathbf{1}\{f^*(x) \neq g(x)\} d\mathbb{P}_X(x) \\
&= \int_{G \Delta G^*} |2\eta(x) - 1| d\mathbb{P}_X(x) \ .
\end{aligned}$$

□

In words, the excess risk involves only the subset of the feature space $\mathcal{X}$ where the two rules, g and f^*, disagree (in logic terms this is an exclusive-or).

Example 2.1.1 *Suppose $\mathbb{P}_X$ is the uniform distribution in the interval $[0,4]$ and that*

$$\eta(x) = \begin{cases} 0 & \text{if } 0 \leq x < 1 \\ 0.3 & \text{if } 1 \leq x < 2 \\ 0.5 & \text{if } 2 \leq x < 3 \\ 1 & \text{if } 3 \leq x \leq 4 \end{cases} .$$

The Bayes classifier is simply $f^*(x) = \mathbf{1}\{x \geq 2\}$ (note that the Bayes classifier never depends on the marginal distribution of the features $\mathbb{P}_X$). Now consider the classifier $g(x) = \mathbf{1}\{x \geq 3\}$. This classifier has actually the same risk as the Bayes classifier (why?).

Consider also the classifier $h(x) = \mathbf{1}\{x \geq 1\}$. The excess risk of this classifier is

$$\begin{aligned} R(h) - R^* &= \int_{H \Delta G^*} |2\eta(x) - 1| d\mathbb{P}_X(x) \\ &= \int_1^2 |2\eta(x) - 1| \frac{1}{4} dx \\ &= \int_1^2 0.4 \frac{1}{4} dx = 0.1 \ , \end{aligned}$$

where $H = [1,4]$ and $G^* = [2,4]$.

2.2 Regression

The goal of statistical learning is to construct a prediction mapping from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$, using training data. In regression this prediction rule is often called an estimator or regression function. For concreteness, take $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$, and consider the squared loss function

$$\ell(\hat{y}, y) = (\hat{y} - y)^2 \ .$$

Recalling that the risk is defined to be the expected value of the loss function, we have

$$R(f) = \mathbb{E}\left[\ell(f(X), Y) \middle| f\right] = \mathbb{E}\left[(f(X) - Y)^2 \middle| f\right] \ .$$

As we did for classification, it is important to understand what is the minimum risk we can hope for, that is, $R^* = \inf_f R(f)$. Below we assume f is deterministic, so that we do not need to condition on f (just as we did in the previous section). We have the following result.

Proposition 2.2.1 (Minimum Risk under Squared Error Loss (MSE))

Define $f^*(x) = \mathbb{E}[Y|X = x]$ (this is called the regression function). Then this prediction rule has the smallest possible risk, that is

$$R(f^*) = R^* \ .$$

In addition, the excess risk of any rule $f : \mathcal{X} \to \mathcal{Y}$ is given by

$$R(f) - R^* = \mathbb{E}\left[(f(X) - f^*(X))^2\right] = \mathbb{E}\left[(f(X) - \mathbb{E}[Y|X])^2\right] \ .$$

Proof Let $f : \mathcal{X} \to \mathcal{Y}$ be any prediction rule (without loss of generality assume it is not random). We have

$$\begin{aligned}
R(f) &= \mathbb{E}\left[(f(X) - Y)^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[(f(X) - Y)^2 | X\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[(f(X) - \mathbb{E}[Y|X] + \mathbb{E}[Y|X] - Y)^2 | X\right]\right] \\
&= \mathbb{E}\Big[\mathbb{E}[(f(X) - \mathbb{E}[Y|X])^2 | X] \\
&\quad + 2\mathbb{E}\left[(f(X) - E[Y|X])(\mathbb{E}[Y|X] - Y)|X\right] + \mathbb{E}[(\mathbb{E}[Y|X] - Y)^2 | X]\Big] \\
&= \mathbb{E}\Big[\mathbb{E}[(f(X) - \mathbb{E}[Y|X])^2 | X] \\
&\quad + 2(f(X) - E[Y|X]) \times 0 + \mathbb{E}[(\mathbb{E}[Y|X] - Y)^2 | X]\Big] \\
&= \underbrace{\mathbb{E}\left[(f(X) - \mathbb{E}[Y|X])^2\right]}_{>0} + R(f^*) \ .
\end{aligned}$$

Thus $R(f) \geq R(f^*)$ for any prediction rule f, and therefore $R^* = R(f^*)$. The second statement in the proposition follows naturally from the last equality. □

2.3 Empirical Risk Minimization

Now that we have a good working knowledge of the notions of risk, Bayes risk, and so on, we would like to figure out how to use data to choose a good prediction rule. A natural way to do this is to use the performance on the training data as a surrogate for the performance for unseen data, leading naturally to the notion of empirical risk.

Definition 2.3.1 (Empirical Risk) *Let $\{X_i, Y_i\}_{i=1}^n \overset{i.i.d.}{\sim} \mathbb{P}_{XY}$ be a collection of training data. Then the empirical risk is defined as*

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i) \ .$$

An interpretation of the empirical risk, is that it is the risk when considering the empirical distribution of the data as a surrogate for the true distribution. By the law of large numbers you know that, for any *fixed* prediction rule f the empirical risk $\hat{R}_n(f)$ will converge to the true risk $R(f)$ (in a stochastic sense). Therefore, we might use the empirical risk to choose a prediction rule from a pool of candidates - this is the idea of empirical risk minimization.

Definition 2.3.2 (Empirical Risk Minimizer) *Suppose you have a set of candidate prediction rules $\mathcal{F}$. Empirical risk minimization chooses the rule that minimizes the empirical risk, namely*

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \ .$$

Remark 2.3.1 *The mathematical operator* $\arg\min$ *returns a value minimizing the argument function. For instance, in the formula $x = \arg\min_{t \in \mathcal{X}} g(t)$ the value x is any value in $\mathcal{X}$ such*

that $g(x) \leq g(t)$ for any $t \in \mathcal{X}$. So this might not be uniquely defined (or it might not even exist) in general. For our purposes the latter is not really important, and it suffices to say that, if multiple minimizers exist the $\arg\min$ *operator will return one of them. Similar comments apply to* $\arg\max$.

The main idea behind this approach is that, for any fixed rule f, the empirical risk should be somewhat close to the true risk. In fact the strong law of large numbers says that for a *fixed* rule f

$$\hat{R}_n(f) \overset{\text{a.s.}}{\to} R(f) \ ,$$

as $n \to \infty$. However, this statement might not hold uniformly for all rules $f \in \mathcal{F}$, and then we will have a problem. Before going into details let us see a few examples illustrating these notions.

Example 2.3.1 (Linear Classifiers) *Let $\mathcal{Y} = \{-1, 1\}$ and consider the set of hyperplane classifiers over the feature space $\mathcal{X} = \mathbb{R}^d$ or $[0,1]^d$*

$$\mathcal{F} = \left\{ x \mapsto f(x) = \text{sign}(w^T x) : w \in \mathbb{R}^d \right\} \ ,$$

where $\text{sign}(t) = 2\mathbf{1}\{t \geq 0\} - 1$. *If we use the notation $f_w(x) \equiv \text{sign}(w^T x)$, then the set of classifiers can be alternatively represented as*

$$\mathcal{F} = \left\{ f_w : w \in \mathbb{R}^d \right\} \ .$$

In this case, the classifier which minimizes the empirical risk is

$$\begin{aligned} \hat{f}_n &= \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \\ &= \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{\text{sign}(w^T X_i) \neq Y_i\} \ . \end{aligned}$$

If there exists an hyperplane that separates the classes perfectly then it can be found efficiently (e.g., using the perceptron algorithm). Furthermore, if one desires to find the separating hyperplane that is somehow "far" from all data points than we can use what are known as Support Vector Machines (SVMs). However, if the data cannot be separated by any hyperplane than one can show that minimizing the empirical risk is actually computationally very demanding when d is large.

Example 2.3.2 (Regression) *Let the feature space be $\mathcal{X} = [-1, 1]$ and let the set of possible estimators be*

$$\mathcal{F} = \{\textit{degree } d \textit{ polynomials on } [-1,1]\} \ .$$

In this case, the classifier which minimizes the empirical risk is

$$\begin{aligned} \hat{f}_n &= \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \\ &= \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 \ . \end{aligned}$$

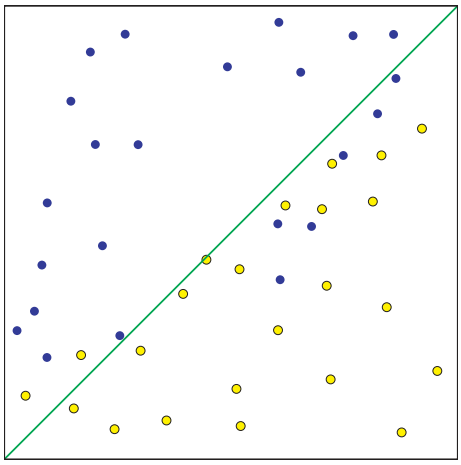


Figure 2.1: Example linear classifier for binary classification.

Alternatively, this can be expressed as

$$\begin{aligned}\hat{w} &= \arg\min_{w\in\mathbb{R}^{d+1}} \frac{1}{n}\sum_{i=1}^{n}(w_0 + w_1 X_i + \ldots + w_d X_i^d - Y_i)^2 \\ &= \arg\min_{w\in\mathbb{R}^{d+1}} \|Vw - Y\|^2\end{aligned}$$

where V is the Vandermonde matrix

$$V = \begin{bmatrix} 1 & X_1 & \ldots & X_1^d \\ 1 & X_2 & \ldots & X_2^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_n & \ldots & X_n^d \end{bmatrix} .$$

The pseudoinverse can be used to solve for $\hat{w}$ (provided V^TV is invertible)

$$\hat{w} = (V^TV)^{-1}V^TY \ .$$

A polynomial estimate is displayed in Figure 2.2. Note that in some cases, the pseudoinverse of the Vandermonde matrix can produce unstable results (see code below). This can be alleviated by using a Chebyshev Vandermonde matrix, or similar constructions. While the Vandermonde matrix contains evaluations of the polynomials $\{x^0, x^1, x^2, \cdots, x^k\}$, the Chebyshev Vandermonde matrix contains evaluations of the 0^{th} through k^{th} degree Chebyshev polynomials[1]*, which are orthogonal (in a certain sense) on the interval $[-1, 1]$.*

2.4 Model Complexity and Overfitting

Suppose $\mathcal{F}$, our collection of candidate functions, is very large. In general we can always make

$$\min_{f\in\mathcal{F}} \hat{R}_n(f)$$

[1]For a quick reference you can check http://mathworld.wolfram.com/ChebyshevPolynomialoftheFirstKind.html.

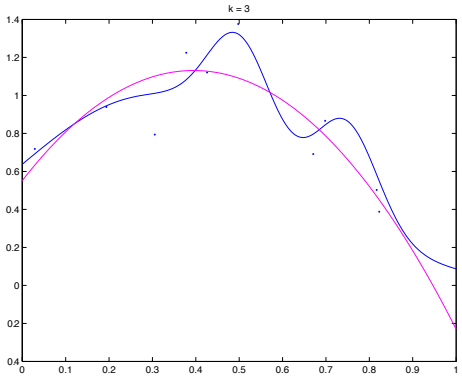


Figure 2.2: Example polynomial estimator. Blue curve denotes f^*, magenta curve is the polynomial fit to the data (denoted by dots).

small by using a large class of prediction rules $\mathcal{F}$, thereby providing more possibilities to fit to the data.

Consider this extreme example: Let $\mathcal{F}$ be all measurable functions. Then for the function f_{bad} such that

$$f_{\text{bad}}(x) = \begin{cases} Y_i & \text{if } x = X_i \text{ for } i = 1, \ldots, n \\ 0 & \text{otherwise} \end{cases},$$

the empirical risk is exactly 0. So if this function is in $\mathcal{F}$ we might take it as the empirical risk minimizer. However, this rule predicts zero for any value of the features that is not on the dataset, so it seems (and is) a terrible prediction rule. In a sense, this prediction rule overfits the data and doesn't "learn" anything outside the dataset. So we might want to have a large class of models, but not too large, or instead have a clever way to choose from a big class.

Example 2.4.1 (Classification Overfitting) *Consider the data and classifier in Figure 2.3. The data was generated by a mixture of two Gaussian distributions centered in the upper left and lower right quadrants of the feature space, and each of these two components corresponding to a different label. If we are given this information (that is, the distribution that gives rise to the data) then the optimal estimator is the linear estimator in Figure 2.1. However, we can construct another classifier that perfectly fits the training data. However, this is going to be a lousy classifier for data not in the training set.*

Example 2.4.2 (Regression Overfitting) *Consider the problem of fitting one-dimensional data with a polynomial, as in Example 2.3.2. Below is some Matlab code that simulates data and computes a polynomial fitting. The implementation has several numerical problems, so you cannot fit polynomials with high degrees. Nevertheless, this will give you a feel for the overfitting problem.*

```
%Polynomial approximation of data
close all;
clear all;

%Features live in [-1,1] and we use Chebyshev Polynomials of the first
%kind, to avoid some numerical issues...
```

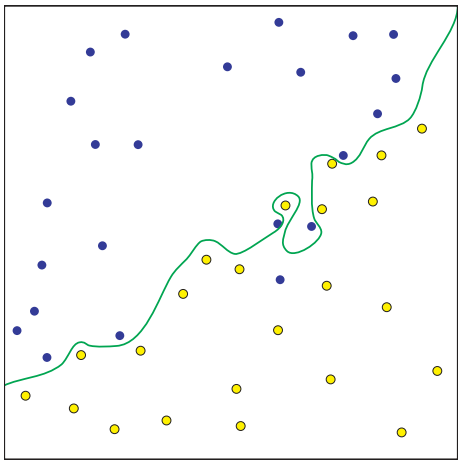


Figure 2.3: Example of overfitting classifier. The classifier's decision boundary wiggles around in order to correctly label the training data, but the optimal Bayes classifier is a straight line.

```
n=100;
x=2*rand(n,1)-1;

%y=2-3*x+5*x.^2+0.1*randn(n,1);  % a different function
y=exp(-5*(x-.3).^2)+.5*exp(-100*(x-.5).^2)+.5*exp(-100*(x-.75).^2)+0.1*randn(n,1);

t=(-1:0.0001:1)';

%pt=2-3*t+5*t.^2;  % a different function
pt=exp(-5*(t-.3).^2)+.5*exp(-100*(t-.5).^2)+.5*exp(-100*(t-.75).^2);

plot(x,y,'ro',t,pt,'b-');
xlabel('Feature');
ylabel('Label');
legend('Sample data','True regression function');

%Polynomial fit to the data with degree d

d=10;

V=zeros(n,d+1);
Vt=zeros(length(t),d+1);
for k=0:d;
    V(:,k+1)=x.^k;
    Vt(:,k+1)=t.^k;
    if k>1;
        V(:,k+1)=2*x.*V(:,k)-V(:,k-1);
        Vt(:,k+1)=2*t.*Vt(:,k)-Vt(:,k-1);
```

```
    end;
end;

w=inv(V'*V)*V'*y;
plot(x,y,'ro',t,pt,'b-',t,Vt*w,'m-');
xlabel('Feature');
ylabel('Label');
legend('Sample data','True regression function','Estimated rule');
title(['Data fitted using degree' num2str(d) 'polynomial']);
```

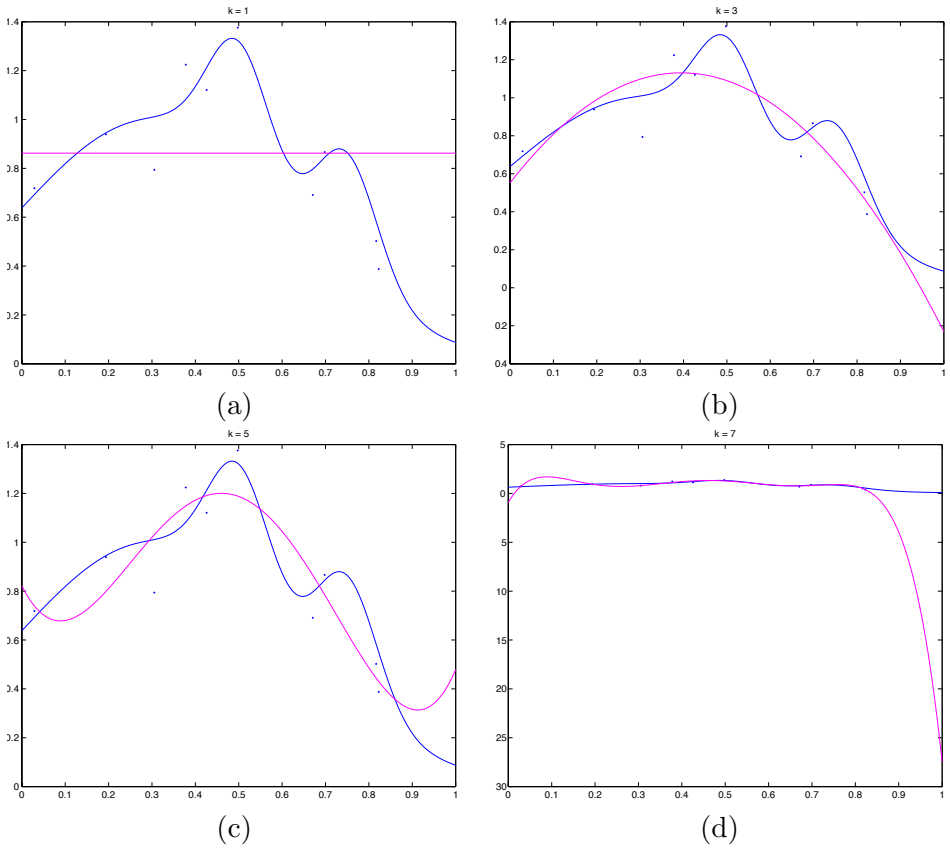


Figure 2.4: Example polynomial fitting problem. Blue curve is f^*, magenta curve is the polynomial fit to the data (dots). (a) Fitting a polynomial of degree $d = 0$ - this is an example of underfitting; (b)$d = 2$; (c) $d = 4$; (d) $d = 6$ - this is an example of overfitting. For (d) the empirical loss is zero, but clearly the estimator would not do a good job of predicting y when x is close to one.

2.5 Exercises

Exercise 2.5.1 *Consider the problem of binary classification, but instead of the* $0/1$ *loss let's consider an asymmetric loss, defined as*

$$\ell(\hat{y}, y) = \begin{cases} c & \text{if } \hat{y} = 1, y = 0 \\ 1 & \text{if } \hat{y} = 0, y = 1 \\ 0 & \text{otherwise} \end{cases},$$

where $c > 0$.

a) Show that the classifier $f^*(x) = \mathbf{1}\left\{\eta(x) \geq \frac{c}{c+1}\right\}$ *has the smallest risk among any other classifier.*

b) Give the expression for the excess risk of any classifier.

Hint: *you can mimic the steps in the proof of Proposition 2.1.1, but consider instead the loss function above.*

Exercise 2.5.2 *Consider the problem of binary classification with the* $0/1$ *loss. Suppose* $\mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 1/2$ *(i.e., both labels are equally likely) and that*

$$\mathbb{P}(X \leq x | Y = 0) = \begin{cases} 0 & \text{if } x \leq 0 \\ x/2 & \text{if } 0 < x \leq 2 \\ 1 & \text{if } x > 2 \end{cases},$$

and

$$\mathbb{P}(X \leq x | Y = 1) = \begin{cases} 0 & \text{if } x \leq 1 \\ (x-1)/3 & \text{if } 1 < x \leq 4 \\ 1 & \text{if } x > 4 \end{cases}.$$

a) Verify that the marginal distribution of the features is

$$\mathbb{P}(X \leq x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \frac{x}{4} & \text{if } 0 < x \leq 1 \\ \frac{5}{12}x - \frac{1}{6} & \text{if } 1 < x \leq 2 \\ \frac{x}{6} + \frac{2}{6} & \text{if } 2 < x \leq 4 \\ 1 & \text{if } x > 4 \end{cases}.$$

b) Show that $\eta(x) = \mathbb{P}(Y = 1 | X = x)$ *with* $x \in [0, 4]$ *is given by*

$$\eta(x) = \begin{cases} 0 & \text{if } 0 \leq x \leq 1 \\ 2/5 & \text{if } 1 < x \leq 2 \\ 1 & \text{if } 2 < x \leq 4 \end{cases}.$$

c) Use the above information to compute the risk of the Bayes classifier.

d) What is the excess risk of the classification rule $f(x) = \mathbf{1}\{x \geq 1\}$*?*

Exercise 2.5.3 *Consider a setting where the feature and label space are $\mathcal{X} = \mathcal{Y} = [0,1]$. In this exercise we will consider both the square loss and the absolute loss, namely:*

$$\ell_{square}(y_1, y_2) = (y_1 - y_2)^2 \qquad \text{and} \qquad \ell_{abs}(y_1, y_2) = |y_1 - y_2| \ .$$

Let (X, Y) be random, with the following with joint probability density

$$p_{XY}(x, y) = 2y \ , \quad \text{where } x, y \in [0,1] \ .$$

Note that in this case X and Y are independent, meaning the feature X carries no information about Y.

a) What is risk of the prediction rule $f(x) = \frac{1}{2}$ according to the two loss functions?

b) What is the risk of the prediction rule $f^*(x) = \frac{1}{\sqrt{2}}$ according to the two loss functions?

c) Show that f^* has actually the smallest absolute loss risk among all prediction rules.

Hint: *part c) is a bit harder. In general the Bayes predictor according to the absolute value loss is the median of the conditional distribution of Y given $X = x$.*

Exercise 2.5.4 *Let's consider again the problem of binary classification. We'll use as class labels $\mathcal{Y} = \{-1, 1\}$ (this is for convenience of presentation. Although the $0/1$ loss is a very natural choice in this setting, from a computational point of view it is problematic - it is not convex in its arguments, meaning that many of the optimization problems ensuing will not be convex and might be very difficult to solve in a computationally efficient manner. So, it is very useful to "convexify" the loss in a smart way so that it is "close" to the $0/1$ loss.*

In this exercise you will consider the hinge *loss, the squared loss, and the $0/1$ loss. Although the labels can take only the values -1 and 1 we allow our predictions to take any real value, so the loss functions are a map from $\mathbb{R} \times \mathcal{Y}$ to $\mathbb{R}$:*

$$\ell_{0/1}(\hat{y}, y) = \mathbf{1}\{\hat{y} \neq y\} \quad , \quad \ell_{hinge}(\hat{y}, y) = \max(1 - \hat{y}y, 0) \quad \text{and} \quad \ell_{square}(\hat{y}, y) = (\hat{y} - y)^2 \ .$$

Based on these loss functions, we can define, for any prediction rule $g : \mathcal{X} \to \mathbb{R}$ the corresponding risks:

$$R_{0/1}(g) = \mathbb{E}[\mathbf{1}\{G(X) \neq Y\}] \ , \ R_{0/1}(g) = \mathbb{E}[\ell_{hinge}(G(X), Y)] \text{ and } R_{square}(g) = \mathbb{E}[(G(X) - Y)^2] \ .$$

As in Chapter 2 let $\eta(x) = P(Y = 1 | X = x)$. Define the following prediction rules:

$$f^*_{0/1}(x) = f^*_{hinge}(x) = 2\mathbf{1}\{\eta(x) \geq 1/2\} - 1 \ ,$$

$$f^*_{square}(x) = 2\eta(x) - 1 \ .$$

In class we have shown that $f^*_{0/1}$ is optimal with respect to the $0/1$ loss. Show that the other two rules are optimal with respect to the corresponding risk definition. That is, for any mapping $g : \mathcal{X} \to \mathbb{R}$ we have $R_{loss}(g) - R_{loss}(f^*_{loss}) \geq 0$ for each specific loss.

From (a) we see already a nice feature of the hinge loss - the corresponding Bayes' prediction rule coincides with that of the $0/1$ loss. This is, however, not the case for the square loss.

Exercise 2.5.5 *In this exercise you will consider the squared loss, and a feature space* $\mathcal{X} = [0,1]$ *and label space* $\mathcal{Y} = \mathbb{R}$*. Suppose you have a dataset* $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$*, and assume* $X_i \overset{i.i.d.}{\sim}$ *Unif*$([0,1])$ *and* $Y_i = aX_i + \varepsilon_i$*, where* $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ *and you are using the following prediction rule* $\hat{f}_n(x; D_n) = \frac{2x}{n} \sum_{i=1}^n Y_i$*.*

a) What is the expected excess risk of $\hat{f}_n$*? Namely, compute* $\mathbb{E}[R(\hat{f}_n)] - R^*$*, where* R^* *is the risk of the Bayes' classifier.*

b) Suppose now the distribution of ε_i *has zero mean and finite variance, but it is otherwise arbitrary. What is the expected excess risk then?*

c) Consider instead the setting in the question with a single modification, namely $X_i \overset{i.i.d.}{\sim}$ *Unif*$([1/4, 3/4])$*. What is now the excess risk?*

Chapter 3

Competing Goals: approximation vs. estimation

In the previous chapter we saw that the size/complexity of the class of prediction rules $\mathcal{F}$ seems to play a crucial role. There are various approaches to deal with this issue. Some are more appealing from a practical point of view, while others are more appealing from a theoretical point of view. In this chapter we will see a number of ways we can manage the tradeoff between having models that fit the distribution of the data well (good approximation properties) and can be estimated well from data (good estimation properties). If one is talking about squared loss this is known as the bias-variance tradeoff. At this point this is mostly an overview, and we will not go into many technical details.

One approach to avoiding overfitting is to restrict $\mathcal{F}$ to some subset of all measurable functions. Say we used data to find a prediction rule $\hat{f}_n$. To gauge the performance of a $\hat{f}_n$ we examine the difference between the expected risk of $\hat{f}_n$ and the Bayes' risk (this difference is called the *excess risk*).

$$\mathbb{E}\left[R(\hat{f}_n)\right] - R^* = \underbrace{\left(\mathbb{E}[R(\hat{f}_n)] - \inf_{f\in\mathcal{F}} R(f)\right)}_{\text{estimation error}} + \underbrace{\left(\inf_{f\in\mathcal{F}} R(f) - R^*\right)}_{\text{approximation error}} .$$

The *approximation error* quantifies the performance hit incurred by imposing restrictions on $\mathcal{F}$. Data does not play a role in this term, and this is solely determined by the class $\mathcal{F}$ and the underlying distribution of (X, Y). The *estimation error* on the other hand, is due to the randomness of the training data, and quantifies "how well" we can use data to identify the "best" element in $\mathcal{F}$. So here the class $\mathcal{F}$ plays a role, but also how we use the training data. This decomposition into approximation and estimation error is quite natural, and, when dealing with the squared loss, it gives rise to the classical bias-variance tradeoff, where the approximation error plays the role of the squared bias, and the estimation error plays the role of the variance.

It is instructive to see what happens in very extreme cases. If $\mathcal{F}$ is very large (containing all the possible prediction rules) then the approximation can be made as small as possible. If the class if very small (say it has only one prediction rule) then the estimation error is zero (there's nothing to learn from data), but the approximation error will be terrible. So, it seems we want to choose a class $\mathcal{F}$ to be very large. However, as we will see, this will typically make the estimation error large. This tradeoff is illustrated in Figure 3.1.

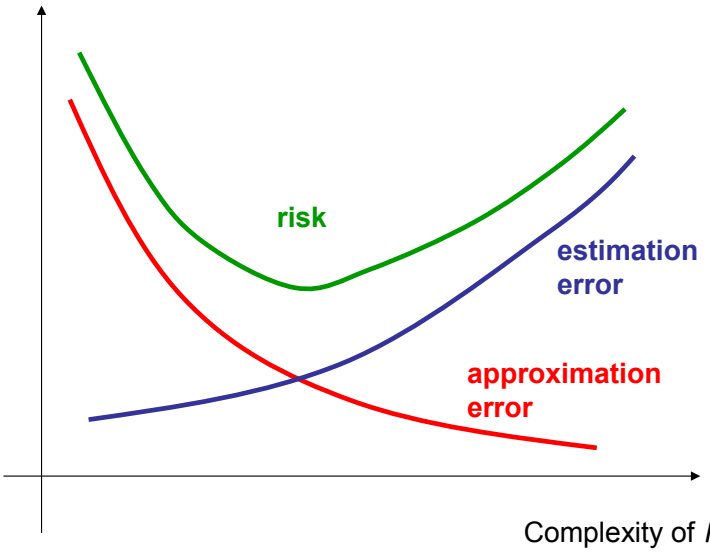


Figure 3.1: Illustration of tradeoff between estimation and approximation errors as a function of the size (complexity) of the $\mathcal{F}$.

To be more concrete, suppose you choose a prediction rule by minimizing the empirical risk

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \ , \ \text{where } \hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i) \ .$$

If $\mathcal{F}$ is very large then $\hat{R}_n(f)$ can be made arbitrarily small and the resulting $\hat{f}_n$ can "overfit" to the data since $\hat{R}_n(f)$ is not a good estimator of the true risk $R(\hat{f}_n)$.

The behavior of the true and empirical risks, as a function of the size (or *complexity*) of the space $\mathcal{F}$, is illustrated in Figure 3.2. Unfortunately, we can't easily determine whether we are over or underfitting just by looking at the empirical risk.

3.1 Strategies To Avoid Overfitting

To avoid the issues illustrated above we want to make sure we are able to choose a prediction rule from the class that is able to capture the essential information from the data, and does

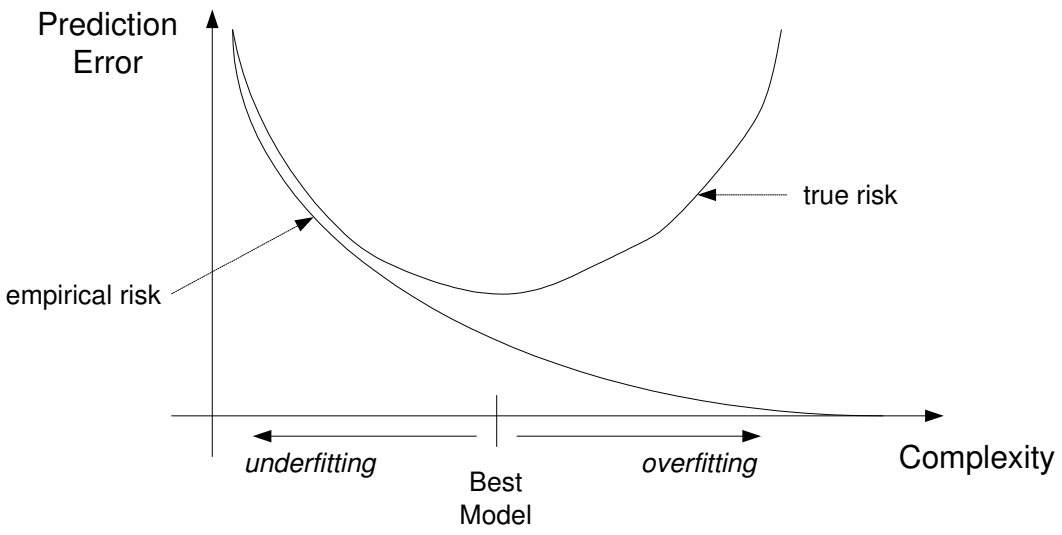


Figure 3.2: Illustration of empirical risk and the problem of overfitting to the data.

not overfit the training data. We will focus on approaches based on empirical risk and consider essentially two ways to deal with the problem:

a) Restrict the "size" of $\mathcal{F}$ so that we control the estimation error well. For instance use only linear hyperplane classifiers, or fit polynomials with degrees at most d. Effectively, this places an upper bound on the estimation error, but will also place a lower bound on the approximation error.

b) Instead of simply minimizing the empirical risk, penalize it to include the extra cost associated with a model (e.g., higher degree polynomials fit the data better, but are more "complex").

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\} ,$$

where $C(f)$ measures the complexity of f somehow.

Example 3.1.1 *Revisit the polynomial regression example of the previous chapter. We can incorporate a penalty term $C(f)$ which is proportional to the degree of f, or the norm of the derivative of f. In essence, this approach penalizes functions that are too "wiggly", and captures the intuition/assumption that the true function is somewhat smooth.*

How do we decide how to restrict or penalize the empirical risk minimization process? Approaches which have appeared in the literature include the following.

3.1.1 Method of Sieves

Perhaps the simplest approach is to try to limit the size of $\mathcal{F}$ in a way that depends on the number of training data n. The more data we have, the more complex the space of models we can entertain.

Let the size of the classes of candidate functions grow with n. That is, take

$$\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n, \ldots$$

where $|\mathcal{F}_i|$ grows as $i \to \infty$ or the elements of F_i are described by more and more parameters as i increases. In other words, consider a sequence of spaces with increasing complexity or degrees of freedom depending on the number of training data samples n.

Given samples $\{X_i, Y_i\}_{i=1}^n$ i.i.d. distributed according to $\mathbb{P}_{XY}$, select $f \in \mathcal{F}_n$ to minimize the empirical risk

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}_n} \hat{R}_n(f) \ .$$

In the next chapter we will consider an example using the method of sieves, and see exactly how it can be used. The basic idea is to design the sequence of model spaces in such a way that the excess risk decays to zero as $n \to \infty$. This sort of idea has been around for decades, but Grenander's method of sieves is often cited as a nice formalization of the idea (see for instance (Grenander, 1981)).

3.1.2 Complexity Penalization Methods

Bayesian Methods (Bayes, 1763)

In certain cases, the empirical risk happens to be a (log) likelihood function, and one can then interpret the cost $C(f)$ as reflecting prior knowledge about which models are more or less likely. In this case, $e^{-C(f)}$ is like a prior probability distribution defined on the space $\mathcal{F}$. The cost $C(f)$ is large if f is highly improbable, and $C(f)$ is small if f is highly probable.

Note that, if we assume $\mathcal{F}$ is finite, then this means empirical risk minimization is simply placing a uniform "prior" on the set $\mathcal{F}$.

Example 3.1.2 *Let $X_i \sim Unif([0,1])$, $Y_i = f(X_i) + W_i$, where $W_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. Then the joint density of the data (called the likelihood) is given by*

$$p(D_n|f) = p(x_1, y_1, \ldots, x_n, y_n|f) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(Y_i - f(X_i))^2}{2\sigma^2}\right) \ .$$

Suppose you have some prior *knowledge on f, in particular that f arises by selecting an element of $\mathcal{F}$ randomly. So, one should regard f as a random quantity. To be concrete, let's assume $p_{\mathcal{F}}$ is the density of f. We can then compute the posterior density $p(f|D_n)$, where $D_n = \{X_i, Y_i\}_{i=1}^n$.*

$$p(f|D_n) = \frac{p(D_n|f)p_{\mathcal{F}}(f)}{p(D_n)} \ .$$

For the "hard-core" Bayesian the posterior is the ultimate object of interest. Based on it you can compute anything you want about f. For instance, you may estimate f by the posterior mean (this can be seen to correspond to the minimization of a squared loss function). In some cases it is more sensible to look at the posterior mode: the maximum a posteriori *estimator is just the model that maximizes the posterior, or equivalently minimizes $-\log p(f|D_n)$.*

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} -\log p(D_n|f) - \log p_{\mathcal{F}}(f) \ ,$$

where the first term can be interpreted as the empirical risk $-\log f(D_n|f) = const + \frac{1}{2\sigma^2}\sum_{i=1}^n (Y_i - f(X_i))^2$, and the second term is a complexity penalty $C(f) = -\log p_{\mathcal{F}}(f)$.

Description Length Methods (Rissanen, 1978)

This is an information theoretical approach similar in spirit to the Bayesian methods. The idea is to measure the complexity of the model by the number of bits it takes to represent it. More complex models take more bits to represent. Let the cost $C(f)$ be proportional to the number of bits needed to describe f (the *description length*). This results in what is known as the minimum description length (MDL) approach, and gives rise to the following estimator

$$\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\} \ .$$

In certain situations the empirical risk can be interpreted as the number of bits needed to explain the dataset given the model f. Then the above criteria is simply choosing the best compression strategy for the data, by first encoding a plausible model and them encoding the differences between the prediction of the model and the actual data.

Both the Bayesian and Description Length methods have a long history and literature. They raise numerous questions (some philosophical) and we will not pursue these here. For our purposes, one of the nice features of the description length approach is that it gives a very natural way to penalize models - simply count how many "bits" do you need to describe them. This often translates in counting the number of parameters in the models (e.g., the degree of a polynomial regression functions).

3.1.3 Hold-out Methods

Hold-out methods are a very powerful way to deal with the problem of overfitting in practice. Unfortunately, the most useful hold-out procedures are typically hard to study, and therefore the amount of theoretical guarantees available is somewhat limited. Nevertheless, there are results indicating why this is a good idea (and some of them are quite deep).

The basic idea of "hold-out" methods is to split the n samples $D \equiv \{X_i, Y_i\}_{i=1}^n$ into a training set, D_T, and a test (validation) set, D_V.

$$D_T = \{X_i, Y_i\}_{i=1}^m \ , \text{ and } D_V = \{X_i, Y_i\}_{i=m+1}^n \ .$$

Now, suppose we have a collection of different model spaces $\{\mathcal{F}_\lambda\}$ indexed by $\lambda \in \Lambda$. For instance $\mathcal{F}_\lambda$ is the set of polynomials of degree d, with $\lambda = d$.

We can obtain candidate solutions using the training set as follows. Define

$$\hat{R}_m(f) = \frac{1}{m} \sum_{i=1}^m \ell(f(X_i), Y_i) \ ,$$

and take

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{F}_\lambda} \hat{R}_m(f) \ .$$

This provides us a set of candidate solutions $\{\hat{f}_\lambda\}$ indexed by λ. Note that these were obtained using only the training set. We can now try to select one of these by seeing how it performs in the validation set. Define the hold-out error estimate using the test set as

$$\hat{R}_V(f) = \frac{1}{n-m} \sum_{i=m+1}^n \ell(f(X_i), Y_i) \ ,$$

and select the "best" model to be $\hat{f}_n = \hat{f}_{\hat{\lambda}}$ where

$$\hat{\lambda} = \arg\min_{\lambda} \hat{R}_V \left(\hat{f}_\lambda \right) \ . \tag{3.1}$$

So, in retrospect, we use the training part of the data set to choose a best model for each value of λ, and then use the validation set to assess how the chosen model performs on an "independent" data set. Why is this a good idea? Think of the polynomial regression setting we have been considering. If you use a high degree polynomial it will fit the training subset very well, but it will be very "wiggly". Therefore, in the validation set it will probably be very bad. On the other hand, a moderate degree fit might ensure both are somewhat small.

The above approach can also be used in conjunction with complexity regularization approaches. Suppose you want to use a penalized empirical risk minimization approach, but do not know how much "weight" to put on the penalization. So, for any value of λ define

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_m(f) + \lambda C(f) \right\} \ .$$

We can now use the validation set to identify the "right" value of λ, using the approach in (3.1).

Hold-out procedures have many nice theoretical properties, provided both the training and test set grow with n. In practice, the above approach is not so appealing, as we don't know how to adequately split the data into training and testing sets, and there is a feeling that we are not using the information in the data to the fullest extent.

Leaving-one-out Cross-Validation

In light of the comments above there is a big collection of procedures that try to alleviate some of the issues. These are known collectively as cross-validation, and the basic idea is to randomly split the data into training and testing, repeat the procedure over and over, and average the results. In some cases this can be shown to be a very clever way to do things (Craven and Wahba, 1978/79).

A very popular hold-out method is the so call "leaving-one-out cross-validation". For each λ we compute

$$\hat{f}_\lambda^{(-k)} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{\substack{i=1 \\ i \neq k}}^{n} \ell(f(X_i), Y_i) + \lambda C(f) \ ,$$

or

$$\hat{f}_\lambda^{(-k)} = \arg\min_{f \in \mathcal{F}_\lambda} \frac{1}{n} \sum_{\substack{i=1 \\ i \neq k}}^{n} \ell(f(X_i), Y_i) \ .$$

In other words, we remove entry k from the data set, and minimize the (penalized) empirical risk. Next we define the cross-validation function

$$V(\lambda) = \frac{1}{n} \sum_{k=1}^{n} \ell(f_\lambda^{(-k)}(X_k), Y_k)$$
$$\lambda^* = \arg\min_{\lambda} V(\lambda).$$

Finally, our prediction rule is given either by $\hat{f}_{\lambda^*}$ or by $\frac{1}{n}\sum_{k=1}^{n}\hat{f}_{\lambda*}^{(-k)}$ (this choice might depend on specific properties of the loss function). In words, what we are doing is to consider n possible splits of the data into a training set (with $n-1$ points) and a test set (with 1 point). Hence the name, cross-validation. Obviously, instead of "leaving-one-out" you can consider situations where the size of the validation set is $k > 1$ - this gives rise to k-fold cross-validation (but the number of possible validation sets grows like $\binom{n}{k}$, so one might consider consider a subset of all the possible validation sets). Alternatively, one can split the data into "blocks" and do "leave-one-block-out" validation, in what is known as k-fold validation (the leave-one-out validation we described earlier is n-fold validation).

In some cases you can show theoretical guarantees for such procedures, but more often than not this is difficult. Nevertheless, in many situations the above approach can be made computationally efficient and therefore easy to implement. These methods are very commonly used in non-parametric statistics and in the practice of machine learning, but you shouldn't regard them as a panacea.

Chapter 4

Estimation of Lipschitz smooth functions

4.1 Setting

Consider the following setting. Let

$$Y = f^*(X) + W,$$

where X is a random variable (r.v.) on $\mathcal{X} = [0,1]$ and W is a random variable on $\mathcal{Y} = \mathbb{R}$ independent of X satisfying

$$\mathbb{E}[W] = 0 \text{ and } \mathbb{E}[W^2] = \sigma^2 < \infty \ .$$

Our goal is to estimate f^*. Let's make an assumption about f^*, namely, let's assume this function is somewhat smooth and does not change very quickly. We formalize this notion by assuming $f^* : [0,1] \to \mathbb{R}$ is any function satisfying

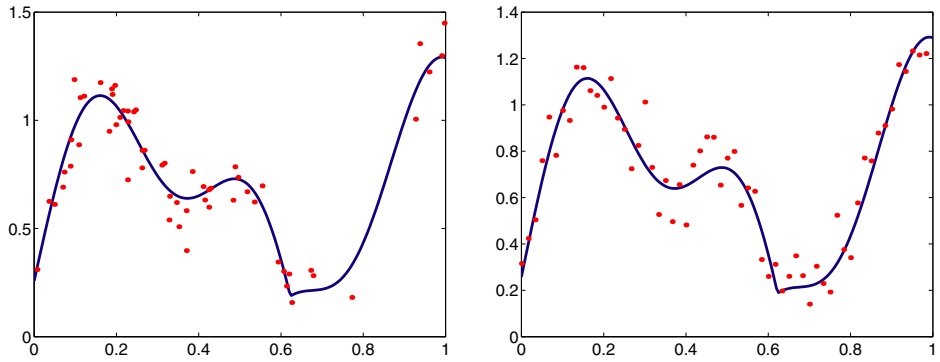
$$|f^*(t) - f^*(s)| \leq L|t-s|, \ \forall t, s \in [0,1] \ , \tag{4.1}$$

where $L > 0$ is a constant. A function satisfying condition (4.1) is said to be Lipschitz on $[0,1]$. Notice that such a function must be continuous, but it is not necessarily differentiable. An example of such a function is depicted in Figure 4.1(a).

So far we specified essentially a joint distribution of (X,Y), parameterized by f^* and by $\mathbb{P}_X$ and $\mathbb{P}_W$. We are interested in prediction, meaning we want to find a prediction rule $f : \mathcal{X} \to \mathcal{Y}$ minimizing the risk $R(f)$ defined in terms of the squared error loss function $\ell(f(X), Y) = (f(X) - Y)^2$. Recall that for this loss function the Bayes' predictor is given by the so-called regression function $\mathbb{E}[Y|X = x]$. So, taking into account the setting we are considering we have

$$\begin{aligned} \mathbb{E}[Y|X = x] &= \mathbb{E}[f^*(X) + W|X = x] \\ &= \mathbb{E}[f^*(x) + W|X = x] \\ &= f^*(x) + \mathbb{E}[W] = f^*(x) \ . \end{aligned}$$

so, in our setting f^* is actually the regression function - meaning this is the best predictor for Y given X.



(a) (b)

Figure 4.1: Example of a Lipschitz function and corresponding observations in our setting. (a) random sampling of f^*, the points correspond to $(X_i, Y_i),\ i = 1, \ldots, n$; (b) deterministic sampling of f^*, the points correspond to $(i/n, Y_i),\ i = 1, \ldots, n$.

As usual, we observe only training data

$$\{X_i, Y_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{XY} \ ,$$
$$Y_i = f^*(X_i) + W_i,\ \ i = \{1, \ldots, n\} \ .$$

Figure 4.1(a) illustrates this setup.

For simplicity we will consider a slightly different setting. In many applications we can sample $\mathcal{X} = [0, 1]$ as we like, and not necessarily at random. For example we can take n samples uniformly spaced on $[0, 1]$

$$\begin{aligned} x_i &= \frac{i}{n} \ ,\ \ i = 1, \ldots, n \ , \\ Y_i &= f^*(x_i) + W_i \\ &= f^*\left(\frac{i}{n}\right) + W_i \ . \end{aligned}$$

Considering this setup instead will facilitate the presentation and analysis. Later on you might see how to generalize this in an exercise. In Figure 4.1(b) you can see an illustration of the setup we will consider for the remainder of the chapter. Note that the above choice of sample points is somewhat related to a uniform choice over $[0, 1]$.

Suppose you have a sample X from a uniform distribution over $[0, 1]$ and would like to predict the corresponding label $Y = f^*(X) + W$, using the training data. From Chapter 2 we know that the Bayes prediction rule is f^*, and that the excess risk of any prediction rule f is given by

$$R(f) - R^* = \|f^* - f\|^2 = \int_0^1 |f^*(t) - f(t)|^2 dt \ ,$$

where we made use of the fact that $X \sim \text{Unif}([0, 1])$. The *expected excess risk* (recall that our estimator $\hat{f}_n$ is based on $\{x_i, Y_i\}$ and hence is a r.v.) is defined as

$$\mathbb{E}\left[R(\hat{f}_n) - R^*\right] = \mathbb{E}[\|f^* - \hat{f}_n\|^2] \ .$$

Finally the *empirical risk* is defined as

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(\frac{i}{n}\right) - Y_i\right)^2 .$$

In a nutshell, our goal is to construct an estimator $\hat{f}_n$ that makes

$$\mathbb{E}\left[\|f^* - \hat{f}_n\|^2\right]$$

as small as possible.

We will use empirical risk minimization together with the method of sieves to find a good estimator of f^*. The first step is to find a class of candidate models $\mathcal{F}$. Perhaps naïvely we will consider piecewise constant functions. Let $m \in \mathbb{N}$ and define the class of piecewise constant functions

$$\mathcal{F}_m = \left\{ f : f(t) = \sum_{j=1}^{m} c_j \mathbf{1}\left\{\frac{j-1}{m} < t \leq \frac{j}{m}\right\},\ c_j \in \mathbb{R} \right\} .$$

In words $\mathcal{F}_m$ is the space of functions that are constant on intervals

$$I_j \equiv \left(\frac{j-1}{m}, \frac{j}{m}\right] ,\ j = 1, \ldots, m .$$

In rigor, we should have said interval $I_{j,m}$, but, for the sake of notational clarity we drop the explicit dependence on m. Clearly if m is rather large we can approximate a Lipschitz function arbitrarily well. So it is sensible to use these classes to construct a set of sieves. Before proceeding, why not consider piecewise linear functions instead? It turns out that for Lipschitz functions there is no added value (in the worst case) when using piecewise linear functions (these are results from the field of approximation theory). So we can stick with the simplest class of functions. So, the estimator we will consider is simply

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}_m} \hat{R}_n(f) = \arg\min_{f \in \mathcal{F}_m} \frac{1}{n}\sum_{i=1}^{n}\left(f\left(\frac{i}{n}\right) - Y_i\right)^2 .$$

4.2 Analysis

Note that we are interested in the expected excess risk of the estimator $\hat{f}_n$. Our first step will be to decompose this quantity in two components, corresponding respectively to approximation error and estimation error. Let $\bar{f}(t) = \mathbb{E}[\hat{f}_n(t)]$. Then

$$\begin{aligned}
\mathbb{E}[\|f^* - \hat{f}_n\|^2] &= \mathbb{E}[\|f^* - \bar{f} + \bar{f} - \hat{f}_n\|^2] \\
&= \|f^* - \bar{f}\|^2 + \mathbb{E}[\|\bar{f} - \hat{f}_n\|^2] + 2\mathbb{E}[\langle f^* - \bar{f}, \bar{f} - \hat{f}_n\rangle] \\
&= \|f^* - \bar{f}\|^2 + \mathbb{E}[\|\bar{f} - \hat{f}_n\|^2] + 2\langle f^* - \bar{f}, \mathbb{E}[\bar{f} - \hat{f}_n]\rangle \\
&= \|f^* - \bar{f}\|^2 + \mathbb{E}[\|\bar{f} - \hat{f}_n\|^2], \qquad (4.2)
\end{aligned}$$

where it is important to note that $\langle a, b\rangle \equiv \int_0^1 a(t)b(t)dt$ is an inner product, and the final step follows from the fact that $\mathbb{E}[\hat{f}_n(t)] = \bar{f}(t)$. Alternatively, we can do the derivation directly: note

that

$$\begin{aligned}\mathbb{E}[\|f^* - \hat{f}_n\|^2] &= \mathbb{E}\left[\int_0^1 \left(f^*(t) - \hat{f}_n(t)\right)^2 dt\right] \\ &= \int_0^1 \mathbb{E}\left[\left(f^*(t) - \hat{f}_n(t)\right)^2\right] dt \ ,\end{aligned}$$

where in the second equality we used Fubini's theorem, that allows us to exchange the order the expectation with the integral. Now, the inside of the integral is simply

$$\begin{aligned}\mathbb{E}\left[\left(f^*(t) - \hat{f}_n(t)\right)^2\right] &= \mathbb{E}\left[\left(f^*(t) - \bar{f}(t) + \bar{f}(t) - \hat{f}_n(t)\right)^2\right] \\ &= \mathbb{E}\Big[\left(f^*(t) - \bar{f}(t)\right)^2 + \left(\bar{f}(t) - \hat{f}_n(t)\right)^2 \\ &\quad + 2\left(f^*(t) - \bar{f}(t)\right)\left(\bar{f}(t) - \hat{f}_n(t)\right)\Big] \\ &= \left(f^*(t) - \bar{f}(t)\right)^2 + \mathbb{E}\left[\left(\bar{f}(t) - \hat{f}_n(t)\right)^2\right] \\ &\quad + 2\left(f^*(t) - \bar{f}(t)\right)\underbrace{\left(\bar{f}(t) - \mathbb{E}\left[\hat{f}_n(t)\right]\right)}_{=0} \ .\end{aligned}$$

Therefore

$$\mathbb{E}[\|f^* - \hat{f}_n\|^2] = \int_0^1 \left(f^*(t) - \bar{f}(t)\right)^2 dt \ + \ \int_0^1 \mathbb{E}\left[\left(\bar{f}(t) - \hat{f}_n(t)\right)^2\right] dt \ .$$

Note that this is just the usual bias-variance decomposition of the mean-squared-error. A couple of important remarks pertaining the right-hand-side of equation (4.2) are in order. The first term, $\|f^* - \bar{f}\|^2$, corresponds to the approximation error and indicates how well can we approximate the function f^* with a function from $\mathcal{F}_m$. Clearly the larger the class $\mathcal{F}_m$ is the smallest we can make this term. This term is precisely the squared bias of the estimator $\hat{f}_n$. The second term, $\mathbb{E}[\|\bar{f} - \hat{f}_n\|^2]$ is the estimation error, which is the variance of our estimator. We will see that the estimation error is small if the class of possible estimators $\mathcal{F}_m$ is also small.

4.2.1 Empirical Risk Minimization

Begin by defining $N_j = \{i \in \{1, \ldots, n\} : \frac{i}{n} \in I_j\}$, and let $|N_j|$ denote the number of elements of N_j. In words, N_j is the list of data points that lie inside the interval I_j. For any $f \in \mathcal{F}_m$, $f(t) = \sum_{j=1}^m c_j \, \mathbf{1}\{t \in I_j\}$, we have

$$\begin{aligned}\hat{R}_n(f) &= \frac{1}{n}\sum_{i=1}^n \left(\left(\sum_{j=1}^m c_j \, \mathbf{1}\{\frac{i}{n} \in I_j\}\right) - Y_i\right)^2 \\ &= \frac{1}{n}\sum_{j=1}^m \left(\sum_{i \in N_j} (c_j - Y_i)^2\right) \ .\end{aligned}$$

The empirical risk minimizer is $\hat{f}_n = \arg\min_{f \in \mathcal{F}_m} \hat{R}_n(f)$. Then

$$\hat{f}_n(t) = \sum_{j=1}^{m} \hat{c}_j \, \mathbf{1}\{t \in I_j\}, \text{ where } \hat{c}_j = \frac{1}{|N_j|} \sum_{i \in N_j} Y_i \ . \tag{4.3}$$

The result in (4.3) can be easily shown, and it is left as an exercise to the reader. Note that we are assuming $|N_j| > 0$. This is true provided $m \leq n$. In fact[1] $\lfloor \frac{n}{m} \rfloor \leq |N_j| \leq \frac{n}{m}$. In what follows we will assume $m \leq n$.

Note that $\mathbb{E}[\hat{c}_j] = \frac{1}{|N_j|} \sum_{i \in N_j} f^* \left(\frac{i}{n}\right)$. Therefore, the function $\bar{f}(t) = \mathbb{E}[\hat{f}_n(t)]$ is simply

$$\bar{f}(t) = \sum_{j=1}^{m} \bar{c}_j \, \mathbf{1}\{t \in I_j\}, \text{ where } \bar{c}_j = \frac{1}{|N_j|} \sum_{i \in N_j} f^* \left(\frac{i}{n}\right) \ .$$

So, we are all set to study the approximation and estimation errors in (4.2).

4.2.2 Approximation Error

$$\begin{aligned}
\|f^* - \bar{f}\|^2 &= \int_0^1 |f^*(t) - \bar{f}(t)|^2 dt \\
&= \sum_{j=1}^{m} \int_{I_j} |f^*(t) - \bar{f}(t)|^2 dt \\
&= \sum_{j=1}^{m} \int_{I_j} |f^*(t) - \bar{c}_j|^2 dt \\
&= \sum_{j=1}^{m} \int_{I_j} \left| f^*(t) - \frac{1}{|N_j|} \sum_{i \in N_j} f^* \left(\frac{i}{n}\right) \right|^2 dt \\
&= \sum_{j=1}^{m} \int_{I_j} \left(\frac{1}{|N_j|} \left| \sum_{i \in N_j} \left(f^*(t) - f^* \left(\frac{i}{n}\right) \right) \right| \right)^2 dt \\
&\leq \sum_{j=1}^{m} \int_{I_j} \left(\frac{1}{|N_j|} \sum_{i \in N_j} \left| f^*(t) - f^* \left(\frac{i}{n}\right) \right| \right)^2 dt \\
&\leq \sum_{j=1}^{m} \int_{I_j} \left(\frac{1}{|N_j|} \sum_{i \in N_j} \frac{L}{m} \right)^2 dt \\
&= \sum_{j=1}^{m} \int_{I_j} \left(\frac{L}{m} \right)^2 dt \\
&= \sum_{j=1}^{m} \frac{1}{m} \left(\frac{L}{m} \right)^2 = \left(\frac{L}{m} \right)^2 .
\end{aligned}$$

[1] The notation $\lfloor x \rfloor$ denotes the largest integer smaller or equal to x.

The above implies that $\|f^* - \bar{f}\|^2$ can be made small by taking m large. So we can approximate Lipschitz functions arbitrarily well using these piecewise constant functions (which are not Lipschitz themselves).

4.2.3 Estimation Error

$$
\begin{aligned}
\mathbb{E}[\|\bar{f} - \hat{f}_n\|^2] &= \mathbb{E}\left[\int_0^1 |\bar{f}(t) - \hat{f}_n(t)|^2 dt\right] \\
&= \mathbb{E}\left[\int_0^1 \sum_{j=1}^m (\bar{c}_j - \hat{c}_j)^2 \mathbf{1}\{t \in I_j\} dt\right] \\
&= \mathbb{E}\left[\sum_{j=1}^m \int_{I_j} (\bar{c}_j - \hat{c}_j)^2 dt\right] \\
&= \frac{1}{m}\sum_{j=1}^m \mathbb{E}\left[(\bar{c}_j - \hat{c}_j)^2\right] \\
&= \frac{1}{m}\sum_{j=1}^m \mathbb{E}\left[\left(\frac{1}{|N_j|}\sum_{i\in N_j}(f^*(i/n) - Y_i)\right)^2\right] \\
&= \frac{1}{m}\sum_{j=1}^m \mathbb{E}\left[\left(\frac{1}{|N_j|}\sum_{i\in N_j} W_i\right)^2\right] \\
&= \frac{1}{m}\sum_{j=1}^m \frac{\sigma^2}{|N_j|} \\
&\le \frac{1}{m}\sum_{j=1}^m \frac{\sigma^2}{\lfloor n/m \rfloor} \\
&= \sigma^2 \frac{1}{\lfloor n/m \rfloor} \le 2\sigma^2 \frac{m}{n} \ ,
\end{aligned}
$$

where we assumed $m \le n$. Actually $\mathbb{E}[\|\bar{f} - \hat{f}_n\|^2] \le (1+\epsilon)\sigma^2 \frac{m}{n}$ for any $\epsilon > 0$ provided $\lfloor n/m \rfloor$ is large enough.

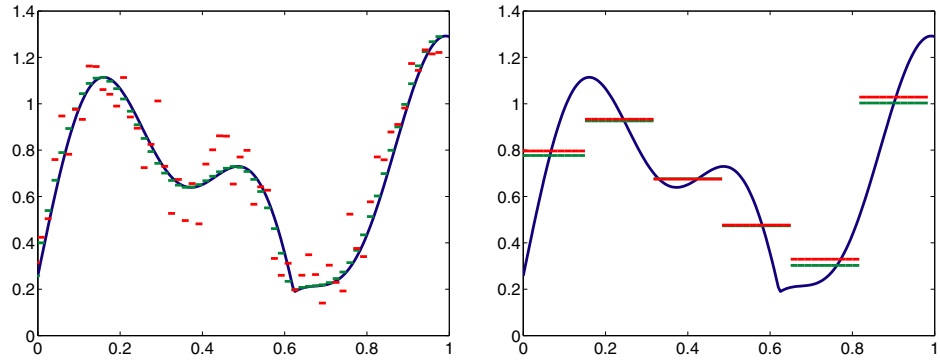
Combining all the facts derived we have

$$
\mathbb{E}[\|f^* - \hat{f}_n\|^2] \ \le \ \frac{L^2}{m^2} \ + \ \frac{2m}{n}\sigma^2 \ = \ O\left(\max\left\{\frac{1}{m^2}, \frac{m}{n}\right\}\right).^{2} \tag{4.4}
$$

What is the best choice of m? If m is small then the approximation error (i.e. $O(1/m^2)$) is going to be large, but the estimation error (i.e. $O(m/n)$) is going to be small, and vice-versa. These two conflicting goals provide a tradeoff that directs our choice of m (as a function of n). Figure 4.2 we depict this tradeoff. In Figure 4.2(a) we considered a large m_n value, and we see that the approximation of f^* by a function in the class $\mathcal{F}_{m_n}$ can be very accurate (that is, our

[2]The notation $x_n = O(y_n)$ (that reads "x_n is *big-O* y_n", or "x_n is of the order no larger than y_n as n goes to infinity") means that $x_n \le C y_n$, where C is a positive constant and y_n is a non-negative sequence.

estimate will have a small bias), but when we use the measured data our estimate looks very bad (high variance). On the other hand, as illustrated in Figure 4.2(b), using a very small m_n allows our estimator to get very close to the best approximating function in the class $\mathcal{F}_n$, so we have a low variance estimator, but the bias of our estimator (i.e. the difference between f_n and f^*) is quite considerable.



(a) (b)

Figure 4.2: Approximation and estimation of f^* (in blue) for $n = 60$. The function $\bar{f}$ is depicted in green and the function $\hat{f}_n$ is depicted in red. In (a)we have $m = 60$ and in (b) we have $m = 6$.

We need to balance the two terms in the right-hand-side of (4.4) in order to maximize the rate of decay (with n) of the expected risk. This implies that $\frac{1}{m^2} \sim \frac{m}{n}$ therefore $m_n \sim n^{1/3}$ and the Mean Squared Error (MSE) is

$$\mathbb{E}[\|f^* - \hat{f}_n\|^2] = O(n^{-2/3}) \ .$$

One can be a bit more cautious with constants, and see that the minimizer of the bound in (4.4) is actually

$$m_n \approx \left(\frac{L^2}{\sigma^2} n\right)^{1/3} \ ,$$

resulting in the MSE bound

$$\mathbb{E}[\|f^* - \hat{f}_n\|^2] \lesssim 3(L\sigma^2)^{2/3} n^{-2/3} \ .$$

In the context of the previous chapter, we just devised a sieve estimator. We are choosing an element from the sequence of classes

$$\mathcal{F}_{m_1}, \mathcal{F}_{m_2}, \ldots$$

with $m_n \approx n^{1/3}$. This choice is such that we are guaranteed to estimate f^* consistently, meaning, as we have more data the excess risk goes to zero.

It is interesting to note that the rate of decay of the MSE we obtain with this strategy cannot be further improved by using more sophisticated estimation techniques. Using information theoretical arguments one can show that

$$\inf_{\hat{f}_n} \sup_{f^* \in \text{Lipschitz}} \mathbb{E}[\|f^* - \hat{f}_n\|^2] \geq cn^{-2/3} \ ,$$

for some $c > 0$, where the infimum is taken over all possible estimators. This is a so called *minimax* lower bound. So we have, at least in terms of the speed of error decay, found the best possible estimator in this sense. Rather surprisingly we are considering classes of models $\mathcal{F}_m$ that are actually not Lipschitz, therefore our estimator of f^* is not a Lipschitz function, unlike f^* itself.

4.3 Exercises

Exercise 4.3.1 *In this chapter we constructed an estimator $\hat{f}_n$ based on the method of sieves. In particular, we used empirical risk minimization to pick a prediction rule from the class*

$$\mathcal{F}_m = \left\{ f : f(t) = \sum_{j=1}^{m} c_j \, \mathbf{1}\left\{ \frac{j-1}{m} < t \leq \frac{j}{m} \right\}, \; c_j \in \mathbb{R} \right\} \; .$$

We studied the behavior of this estimator when the true regression function f^* (which attains the Bayes' risk) belong to the class of Lipschitz functions, namely $f^* \in \mathcal{F}(L)$

$$\mathcal{F}(L) = \{ f : [0,1] \to \mathbb{R} : |f(x) - f(y)| \leq L|x-y| \; \forall x, y \in [0,1] \} \; .$$

Our analysis indicated that we should choose $m = cn^{1/3}$ to achieve small expected excess risk, where $c > 0$. In this exercise we will assume f^* belongs to a different class of functions.

In particular, we are interested in the class of piecewise Lipschitz functions. These are functions that are composed by a finite number of pieces that are Lipschitz. An example of such a function is $g(t) = f_1(t)\mathbf{1}\{t \in [0,1/3]\} + f_2(t)\mathbf{1}\{t \in (1/3,1/2)\} + f_3(t)\mathbf{1}\{t \in (1/2,1]\}$, where $f_1, f_2, f_3 \in \mathcal{F}(L)$.

Let $\mathcal{G}(M, L, R)$ denote the class of bounded piecewise Lipschitz functions. For any function in $g \in \mathcal{G}(M, L, R)$ there is a partition of $[0,1]$ into at most M intervals $A_1, A_2, \ldots, A_M$ such that the restriction of g in any of these intervals is Lipschitz, namely

$$|g(x) - g(y)| \leq L|x-y| \; \forall x, y \in A_k \; ,$$

for any $k = 1, \ldots, M$. Furthermore, assume g is bounded, in the sense that $|g(x)| \leq R$ for all $x \in [0,1]$.

Keep in mind that, although there are at most M pieces, you DO NOT KNOW where these are (that is, you do not know the endpoints of the intervals A_k). Analyze the estimator of this chapter when $f^* \in \mathcal{G}(M, L, R)$.

Hint: *Note that the only thing that changed from the setting we considered is that we are no longer assuming f^* is Lipschitz. Identify in the proof the place were we used this assumption, and see how the corresponding result changes in the setting above.*

Exercise 4.3.2 *Suppose you have a noisy image from a digital camera and want to remove the noise. An image can be thought of as a function $f : [0,1]^2 \to \mathbb{R}$. Let's suppose it satisfies a 2-dimensional Lipschitz condition*

$$|f(x_1, y_1) - f(x_2, y_2)| \leq L \max\left(|x_1 - x_2|, |y_1 - y_2|\right) \quad , \forall x_1, y_1, x_2, y_2 \in [0,1] \; .$$

a) Do you think this is a good model for images? Why or why not.

b) Assume n, the number of samples you get from the function, is a perfect square, therefore $\sqrt{n} \in \mathbb{N}$. Let f^* be a function in this class and let the observation model be

$$Y_{i,j} = f^*(i/\sqrt{n}, j/\sqrt{n}) + W_{i,j}, \quad i,j \in \{1,\dots,\sqrt{n}\} \ ,$$

where as before the noise variables are mutually independent and again $E[W_{i,j}] = 0$ and $E[W_{i,j}^2] \leq \sigma^2 < \infty$.

Using a similar approach to the one in class construct an estimator $\hat{f}_n$ for f^*. Using this procedure what is the best rate of convergence for the expected excess risk when f^* is a 2-dimensional Lipschitz function?

Exercise 4.3.3 (*) *In this chapter we consider the estimation of a smooth function using a* deterministic design *(i.e. the sample locations x_i were deterministic). In this exercise you will extend these results to a random design setting, putting us back in the general setting we introduced in chapter 1.*

Let $\mathcal{X} = [0,1]$ and $\mathcal{Y} = [-R,R]$ for some $R > 0$. Let $\mathbb{P}_{XY}$ be a joint probability distribution over $\mathcal{X} \times \mathcal{Y}$. Suppose you have n i.i.d. samples for $\mathbb{P}_{XY}$, $D_n = \{X_i, Y_i\}_{i=1}^n$. We will use D_n to construct a prediction rule $\hat{f}_n$ and characterize its expected excess risk with respect to the quadratic loss, given by

$$\mathbb{E}[R(\hat{f}_n)] - R^* = \mathbb{E}\left[(\hat{f}_n(X) - Y)^2\right] - R^* \ ,$$

where $(X,Y) \sim \mathbb{P}_{XY}$ is independent from D_n.

Let $f^*(x) = \mathbb{E}[Y|X = x]$ be the regression function (the "best" prediction rule possible). Recall that $R(f^*) = R^*$ and that the excess risk is given by

$$\mathbb{E}\left[(\hat{f}_n(X) - Y)^2\right] - R^* = \mathbb{E}\left[(\hat{f}_n(X) - f^*(X))^2\right] \ .$$

Assume f^* is a Lipschitz function with Lipschitz constant $L > 0$ (i.e. $|f^*(x) - f^*(y)| \leq L|x - y| \ \forall x,y \in [0,1]$). Finally define the estimator

$$\hat{f}_n(x) = \sum_{j=1}^m \hat{c}_j \mathbf{1}\{x \in I_j\} \ ,$$

where $I_j = (\frac{j-1}{m}, \frac{j}{m}]$, and

$$\hat{c}_j = \begin{cases} \frac{\sum_{i=1}^n Y_i \mathbf{1}\{X_i \in I_j\}}{\sum_{i=1}^n \mathbf{1}\{X_i \in I_j\}} & \text{if } \sum_{i=1}^n \mathbf{1}\{X_i \in I_j\} > 0 \\ 0 & \text{otherwise} \end{cases} \ .$$

We will proceed by carefully decomposing the excess risk $E[(\hat{f}_n(X) - f^*(X))^2]$ into an estimation and approximation error (and also a cross-term). Let $\bar{f}$ be an arbitrary prediction rule. It is easy to show that

$$\begin{aligned} \mathbb{E}\left[(\hat{f}_n(X) - f^*(X))^2\right] \ &\leq \ \mathbb{E}\left[(\hat{f}_n(X) - \bar{f}(X))^2\right] + \mathbb{E}\left[(\bar{f}(X) - f^*(X))^2\right] \\ &\quad + 2\sqrt{\mathbb{E}\left[(\hat{f}_n(X) - \bar{f}(X))^2\right] \mathbb{E}\left[(\bar{f}(X) - f^*(X))^2\right]} \ , \end{aligned} \tag{4.5}$$

where this result follows from the application of Cauchy-Schwarz's inequality. The "best" approximating function $\bar{f}$ we will use in this case is simply

$$\bar{f}(x) = \sum_{j=1}^{m} \bar{c}_j \mathbf{1}\{x \in I_j\} \ ,$$

where

$$\bar{c}_j = \begin{cases} \frac{\int_{I_j} f^*(x) dP_X(x)}{\int_{I_j} dP_X(x)} & \text{if } \int_{I_j} dP_X(x) > 0 \\ 0 & \text{otherwise} \end{cases} \ .$$

a) Prove the decomposition in (4.5).

*b) Give an upper bound on the approximation error $\mathbb{E}[(\bar{f}(X) - f^*(X))^2]$ (**Hint:** this is almost analogous to how we did it in this chapter).*

c) Give an upper bound on the estimation error $\mathbb{E}[(\hat{f}_n(X) - \bar{f}(X))^2]$. This is the trickier part of the exercise, since the number of points in each interval is also random. Start by examining $\mathbb{E}[(\hat{f}_n(x) - \bar{f}(x))^2]$ for an arbitrary $x \in [0,1]$ while conditioning on the number of points that fall in the corresponding bin first, and then removing the conditioning. Using this proceed with the bound on $\mathbb{E}[(\hat{f}_n(X) - \bar{f}(X))^2]$.

*(**Hint:** you will find the following fact[3] quite useful - for a Binomial random variable $N \sim Bin(n,p)$ we have $\mathbb{E}\left[\frac{1}{N+1}\right] \leq \frac{1}{(n+1)p}$. This implies that $\mathbb{E}\left[\frac{1}{N}\mathbf{1}\{N > 0\}\right] \leq \frac{2}{(n+1)p}$)*

d) Now put everything together in (4.5). Given your answers to the previous questions what is the proper choice of m as a function of n? What is a bound on the rate of excess risk decay of the procedure provided m is chosen appropriately? How does this compare with the results we showed for the deterministic design?

Exercise 4.3.4 *The choice of error metric used in this chapter assumed implicitly that we are interested in the case $X \sim Unif([0,1])$. In this exercise we will consider a different way to measure the error, that somehow avoids making such an assumption.*

Consider instead the following performance metric

$$AverageRisk(\hat{f}_n) \equiv \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\left(\hat{f}_n(x_i) - f^*(x_i)\right)^2\right] \ . \tag{4.6}$$

This error metric cares only about the quality of the estimator at the points where we take measurements. Sometimes this is what we want (e.g., if we want to process a digital image to remove noise). In any case, we would like this metric to be as close to zero as possible.

a) Repeat the analysis of estimator in this chapter for the average risk metric (4.6), when using $x_i = i/n$.

b) For the next question consider instead the following estimator

$$\hat{f}_n(x) = \frac{\sum_{i=1}^{n} Y_i \mathbf{1}\{|x - x_i| \leq h\}}{\sum_{i=1}^{n} \mathbf{1}\{|x - x_i| \leq h\}} \ ,$$

[3] This result can be found in the appendix of Devroye, Györfi and Lugosi (1996), for instance

where $h > 0$ is a parameter we can choose. Convention $0/0 = 0$ to ensure the $\hat{f}_n$ is always well defined.

This is a so-called kernel estimator. Give an upper bound on the average risk (4.6) in terms of n and h (for the case $x_i = i/n$). What is the choice of h that optimizes this bound, and what is the resulting average risk (4.6)?

Chapter 5

Introduction to PAC learning

Recall that our ultimate goal is to use training data to find a prediction rule that has small risk, hopefully very close to the Bayes' risk (that is, the minimal risk one can ever achieve). Let $\hat{f}_n$ be a prediction rule chosen from a collection $\mathcal{F}$ using training data $D_n = \{X_i, Y_i\}_{i=1}^n$, assumed to be an i.i.d. sample from an unknown distribution $\mathbb{P}_{XY}$. Recall that the risk of $\hat{f}_n$ is defined as

$$R(\hat{f}_n) = \mathbb{E}[\ell(\hat{f}_n(X), Y)|\hat{f}_n] \ ,$$

where (X, Y) is random with distribution $\mathbb{P}_{XY}$ and independent of D_n and ℓ is a chosen loss function. Note that the risk is a random quantity (it is a function of the data $\hat{f}_n$, which in turn depends on the training data). A simple way to deal with this randomness is to consider instead the *expected risk*

$$\mathbb{E}\left[R(\hat{f}_n)\right] = \mathbb{E}\left[\mathbb{E}[\ell(\hat{f}_n(X), Y)|\hat{f}_n]\right] = \mathbb{E}\left[\ell(\hat{f}_n(X), Y)\right] \ .$$

This is no longer a random quantity, but of course it depends on the choice of loss function, the distribution $\mathbb{P}_{XY}$, and the algorithm used to choose $\hat{f}_n$ from $\mathcal{F}$ using data.

As stated above, it is sensible to compare the expected excess risk with the risk of the Bayes' predictor. Furthermore, the difference between those two risks can be decomposed into two terms: an approximation error and an estimation error.

$$\mathbb{E}\left[R(\hat{f}_n)\right] - R^* = \underbrace{\left(\mathbb{E}[R(\hat{f}_n)] - \inf_{f\in\mathcal{F}} R(f)\right)}_{\text{estimation error}} + \underbrace{\left(\inf_{f\in\mathcal{F}} R(f) - R^*\right)}_{\text{approximation error}} \ .$$

The approximation error depends on the relation between $\mathbb{P}_{XY}$ and the chosen model class $\mathcal{F}$. In order to have any control over this error we must make some assumptions on $\mathbb{P}_{XY}$, otherwise there is nothing that can be said (e.g., in the previous chapter to control this term we assumed the Bayes prediction rule was Lipschitz smooth). Note also that the algorithm used to choose $\hat{f}_n$ from $\mathcal{F}$ using the training data D_n does not play any role in the approximation error. So the choice of learning algorithm is irrelevant for that term.

On the other hand, the estimation error has a completely different nature. At first glance it does depend on the model class $\mathcal{F}$, on $\mathbb{P}_{XY}$ and also on the algorithm used to choose $\hat{f}_n$. However, and perhaps surprisingly, we can derive very good bounds for this term without making any assumptions on $\mathbb{P}_{XY}$ (provided the loss function is mildly restricted). So, it is possible to

control the estimation error without assuming anything about the distribution of the training data! We will pursue these type of general results in the next chapters. For machine learning practitioners and learning theorists the control of the estimation error is almost the end goal. These are results characterizing the algorithm used to construct a prediction rule out of a pool of possible models. For instance, suppose we are given a dataset and are asked to use a linear hyperplane classifier to come up with a good classification procedure. Such bounds will quantify how close we are from picking to best possible hyperplane classifier had we actually known the distribution of the data. And we can make such statements without knowing anything about the true distribution of the data. On the other hand, the best hyperplane classifier might be very lousy, and there might be another type of prediction rule which is much more powerful. However, making such a statement requires some knowledge of the distribution of the data, or at least to make some strong assumptions about it. The latter is something statisticians are more prone to do, and typically causes some friction between the two communities.

5.1 PAC bounds

As noted above the risk of $\hat{f}_n$ is a random quantity. A way to deal with this randomness is to consider instead the expected excess risk, but this is somewhat limited. It makes a statement about the risk on average (over the data). We might have a very large excess risk if the risk is extremely large with a small probability (over the data), and small otherwise. This does not necessarily mean we are using a bad algorithm, but simply that, for rare occurrences of the data we will get a very bad prediction rule. A finer control over the excess risk can be stated in terms of a probabilistic statement, in what are known as PAC bounds, as introduced by Valiant (1984). The estimation error will be small if $R(\hat{f}_n)$ is close to $\inf_{f\in\mathcal{F}} R(f)$. The PAC learning framework expresses this as follows: we want $\hat{f}_n$ to be a "(P)robably (A)pproximately (C)orrect" (PAC) model from $\mathcal{F}$. Formally, we say that $\hat{f}_n$ is ε-accurate with confidence $1-\delta$, or $(\varepsilon,\delta)-$PAC for short, if

$$\mathbb{P}\left(R(\hat{f}_n) - \inf_{f\in\mathcal{F}} R(f) > \varepsilon\right) < \delta \ .$$

In other words $R(\hat{f}_n) - \inf_{f\in\mathcal{F}} R(f) \leq \varepsilon$ with probability at least $1-\delta$. We will frequently abbreviate "with probability at least $1-\delta$" by "w.p.$\geq 1-\delta$".

What does the above really mean? Let's give a frequentist interpretation. Suppose we generate many training datasets $D_n^{(1)}, D_n^{(2)}, \ldots, D_n^{(m)}$, from some arbitrary distribution $\mathbb{P}_{XY}$ where m is very very large. Now we use each one of the training datasets to construct a prediction rule $\hat{f}_n^{(i)}$, $i \in \{1,\ldots,m\}$. Then, for about $100(1-\delta)\%$ of these prediction rules we know that $R(\hat{f}_n^{(i)}) - \inf_{f\in\mathcal{F}} R(f) \leq \varepsilon$, so for most $\hat{f}_n^{(i)}$ we have a risk that is very comparable to the risk of the best predictor in $\mathcal{F}$. What does this mean for the dataset we actually have? It means that, if we assume this was generated by sampling from some distribution, then, in most cases, $R(\hat{f}_n) - \inf_{f\in\mathcal{F}} R(f) \leq \varepsilon$. The above interpretation is very close to that of a confidence interval in statistics. When we plug-in the data we observe the above statement is either true or false, but, under assumptions we are making, it will be true most of the time.

5.1.1 A Simple PAC bound in the binary classification setting

We will now construct a PAC bound for a very particular setting. The setting is quite restrictive, but nonetheless gives us a good introduction to the type of results we are seeking. Consider the usual binary classification setting, with $\mathcal{Y} = \{0, 1\}$ and the 0/1-loss function. Let $\mathcal{F}$ consist of a finite number of models, and let $|\mathcal{F}|$ denote that number. Furthermore, assume that $\min_{f \in \mathcal{F}} R(f) = 0$. It is important to notice that this is a very strong assumption, implying both that the Bayes' classifier makes absolutely no errors and that this rule is also in our model class $\mathcal{F}$.

Theorem 5.1.1 (Valiant (1984)) *Consider the 0/1 loss and suppose $\mathcal{F}$ is finite. Let $\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f)$ be an empirical risk minimizer, where $\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{f(X_i) \neq Y_i\}$ is the empirical risk. If $\min_{f \in \mathcal{F}} R(f) = 0$ then for every n and $\varepsilon > 0$,*

$$\mathbb{P}\left(R(\hat{f}_n) > \varepsilon\right) \;<\; |\mathcal{F}| e^{-n\varepsilon} \;\equiv\; \delta \;.$$

Proof Recall that the risk of f is simply $R(f) = P(f(X) \neq Y)$ for the 0/1 loss function. Note that, since $\min_{f \in \mathcal{F}} R(f) = 0$ it follows that, for such an f, $\hat{R}_n(f) = 0$ because we have $\mathbb{E}[\hat{R}_n(f)] = R(f) = 0$ and the empirical risk in non-negative. So the only possibility is that $\hat{R}_n(f) = 0$. This means that the minimizer of the empirical risk $\hat{f}_n$ satisfies $\hat{R}_n(\hat{f}_n) = 0$. In fact, there may be several $f \in \mathcal{F}$ such that $\hat{R}_n(f) = 0$, and we simply take $\hat{f}_n$ to be one of those (for this result is not important how we break ties). To make the presentation clear let $\mathcal{G} = \{f : \hat{R}_n(f) = 0\}$ denote the set of possibilities (clearly $\hat{f}_n \in \mathcal{G}$).

$$\begin{aligned}
\mathbb{P}(R(\hat{f}_n) > \varepsilon) &\leq \mathbb{P}\left(\bigcup_{f \in \mathcal{G}} \{R(f) \geq \varepsilon\}\right) \\
&= \mathbb{P}\left(\bigcup_{f \in \mathcal{F}} \{R(f) > \varepsilon,\ \hat{R}_n(f) = 0\}\right) \\
&= \mathbb{P}\left(\bigcup_{f \in \mathcal{F}: R(f) > \varepsilon} \{\hat{R}_n(f) = 0\}\right) \\
&\leq \sum_{f \in \mathcal{F}: R(f) > \varepsilon} \mathbb{P}(\hat{R}_n(f) = 0) \\
&< \sum_{f \in \mathcal{F}: R(f) > \varepsilon} (1 - \varepsilon)^n \\
&\leq |\mathcal{F}| \cdot (1 - \varepsilon)^n \\
&\leq |\mathcal{F}| e^{-n\varepsilon} \;.
\end{aligned}$$

The third inequality follows since $\hat{R}_n(f) = 0$ implies that $f(X_i) = Y_i$ for all $i \in 1, \ldots, n$, and

so

$$\begin{aligned}\mathbb{P}(\hat{R}(f)=0) &= \mathbb{P}(\cap_{i=1}^{n}\{f(X_i)=Y_i\}) \\ &= \prod_{i=1}^{n}\mathbb{P}(f(X_i)=Y_i) \\ &= (1-R(f))^n \leq (1-\varepsilon)^n \ ,\end{aligned}$$

where the second equality follows from the fact that the data are *independent* samples from $\mathbb{P}_{XY}$. For the last step of the proof we just need to recall that $1-x \leq e^{-x}$. □

Note that for n large, $\delta = |\mathcal{F}|e^{-n\varepsilon} = e^{\log|\mathcal{F}|-n\varepsilon}$ can be made arbitrarily small. To achieve a (ε,δ)-PAC bound for a desired $\varepsilon > 0$ and $\delta > 0$ we require at least $n = \frac{\log|\mathcal{F}|+\log(1/\delta)}{\varepsilon}$ training examples.

We can use the above result to get a bound on the expected excess risk. These are often easier to interpret.

Corollary 5.1.1 *Consider the setting of Theorem 5.1.1. Then*

$$\mathbb{E}[R(\hat{f}_n)] \leq \frac{1+\log|\mathcal{F}|}{n} \ .$$

Proof Recall the following useful fact.

Lemma 5.1.1 *For any non-negative random variable Z with finite mean*

$$\mathbb{E}[Z] = \int_0^\infty P(Z>t)dt \ .$$

A proof sketch of the lemma is presented below. The idea is to apply this lemma to $R(\hat{f}_n)$, but direct application of the lemma gives the bound $|\mathcal{F}|/n$, which has a terribly dependence on the class size. Fortunately, there is a simple workaround. Let $u > 0$ be an arbitrary number. Then

$$\begin{aligned}\mathbb{E}[R(\hat{f}_n)] &= \int_0^\infty \mathbb{P}(R(\hat{f}_n)>t)dt \\ &= \int_0^u \underbrace{\mathbb{P}(R(\hat{f}_n)>t)}_{\leq 1}dt + \int_u^\infty \mathbb{P}(R(\hat{f}_n)>t)dt \\ &\leq u + |\mathcal{F}|\int_u^\infty e^{-nt}dt \\ &= u + \frac{|\mathcal{F}|}{n}e^{-nu} \ .\end{aligned}$$

Minimizing with respect to u produces the smallest upper bound with $u = \frac{\log|\mathcal{F}|}{n}$. □

Sketch proof of Lemma 5.1.1 The lemma can be shown by re-writing the formula as a double integral, and interchanging the order of integration. For simplicity, assume Z has a density f_Z.

Then

$$\begin{aligned}\mathbb{E}[Z] &= \int_0^\infty P(Z > t)dt = \int_0^\infty \int_t^\infty f_Z(x)dxdt \\ &= \int_0^\infty \int_0^\infty f_Z(x)\mathbf{1}\{x \geq t\}dxdt = \int_0^\infty \int_0^\infty f_Z(x)\mathbf{1}\{x \geq t\}dtdx \\ &= \int_0^\infty f_Z(x) \int_0^\infty \mathbf{1}\{t \leq x\}dtdx = \int_0^\infty f_Z(x) \int_0^x dtdx \\ &= \int_0^\infty x f_Z(x)dx \ .\end{aligned}$$

The result can also be seen as a consequence of integration by parts. □

5.2 Agnostic Learning and general PAC bounds

The above theorem required us to make a very strong assumptions about the relation of $\mathcal{F}$ and $\mathbb{P}_{XY}$, namely, that the Bayes' classifier is in the class $\mathcal{F}$ (so the approximation error is zero). This is overly optimistic. Nevertheless, we now have an idea of the type of results we are chasing. Note also that the PAC bounds depend crucially on the algorithm used to choose the prediction rule (in the above, it was empirical risk minimization). In what follows we'll focus mostly on approaches based on empirical risk, and penalized empirical risk. The first step is to study the basic empirical risk minimization approach.

We will proceed without making any assumptions on the distribution $\mathbb{P}_{XY}$. This situation is often termed as *Agnostic Learning*. The root of the word agnostic literally means *not known*. The term agnostic learning is used to emphasize the fact that often, perhaps usually, we may have no prior knowledge about $\mathbb{P}_{XY}$ and therefore we don't know what is the Bayes' predictor, and what is the corresponding risk. The question then arises about how we can reasonably select an $f \in \mathcal{F}$ using data. Empirical risk minimization is a suitable contender as an algorithm (despite the computational issues it might entail).

5.2.1 Empirical Risk Minimization - How good is it?

Consider the Empirical Risk Minimization (ERM) selection of a classification rule from a model class $\mathcal{F}$. That is

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \ .$$

Assume for a moment that we are able to show the following result. With probability at least $1 - \delta$ we have

$$|\hat{R}_n(f) - R(f)| \leq \varepsilon, \quad \forall f \in \mathcal{F} \ , \tag{5.1}$$

for small $\varepsilon > 0$ and $\delta > 0$. In words, this means the empirical and the true risk of any predictor in $\mathcal{F}$ are close, with high probability. If this is the case, then ERM is quite a reasonable choice. In fact with probability at least $1 - \delta$

$$\begin{aligned}R(\hat{f}_n) &\leq \hat{R}_n(\hat{f}_n) + \varepsilon \\ &\leq \hat{R}_n(f) + \varepsilon, \quad \text{for any } f \in \mathcal{F} \\ &\leq R(f) + 2\varepsilon, \quad \text{for any } f \in \mathcal{F} \ ,\end{aligned} \tag{5.2}$$

were the step (5.2) follows from the definition of the empirical risk minimizer meaning that for any $f \in \mathcal{F}$ we have $\hat{R}_n(\hat{f}_n) \leq \hat{R}_n(f)$. We conclude that, with probability at least $1 - \delta$

$$R(\hat{f}_n) \leq \inf_{f \in \mathcal{F}} R(f) + 2\varepsilon \ ,$$

and so with high probability the true risk of the selected rule is only a little bit higher than the risk of the best possible rule in the class. This indicates that ERM is quite a reasonable thing to do. Of course we still need to construct a bound like (5.1). Let's first understand what that bound really means.

5.3 Constructing uniform deviation bounds

To begin, let us recall the definition of the empirical risk. Let $\{X_i, Y_i\}_{i=1}^n$ be a collection of training data. Then the empirical risk is defined as

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i) \ .$$

Note that since the training data $\{X_i, Y_i\}_{i=1}^n$ are assumed to be i.i.d. pairs, the above is a sum of independent and identically distributed random variables. Let

$$L_i = \ell(f(X_i), Y_i) \ .$$

The collection of losses $\{L_i\}_{i=1}^n$ is i.i.d. according to some unknown distribution (depending on the unknown joint distribution of (X, Y) and the loss function). The expectation of L_i is $\mathbb{E}[\ell(f(X_i), Y_i)] = \mathbb{E}[\ell(f(X), Y)] = R(f)$, the true risk of f. For now, let's assume that f is fixed. Then

$$\mathbb{E}[\hat{R}_n(f)] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[\ell(f(X_i), Y_i)] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[L_i] = R(f) \ .$$

We know from the (strong) law of large numbers that the average (or empirical mean) $\hat{R}_n(f)$ converges (almost surely) to the true mean $R(f)$ as $n \to \infty$. That is, $\hat{R}_n(f) \to R(f)$ almost surely as $n \to \infty$. This is, unfortunately, not enough to get a result like (5.1). Two issues remain to be addressed: (i) how fast is the convergence? (ii) the statement in this paragraph is for a fixed $f \in \mathcal{F}$. Can we get a statement that holds uniformly for all $f \in \mathcal{F}$? To address (i) we need what are known as concentration inequalities, that will be introduced in the next chapter. To address (ii) there are several approaches. If the class $\mathcal{F}$ is finite then it is possible to use a crude approach, called a union bound. If the class is larger, however, one must resort to more sophisticated approaches.

5.4 Exercises

Exercise 5.4.1 *In Theorem 5.1.1 above we made use of a very crude but extremely useful bound, called the union bound. This simply states that, if we have a countable collection of events $\{A_i\}$ then $P(\cup_i A_i) \leq \sum_i \mathbb{P}(A_i)$. One has equality only if the events are all mutually exclusive (i.e., disjoint sets). However, typically we use this bound even when many of the events are similar*

(and so not mutually exclusive). This is why the bound we obtain depends on the size of $\mathcal{F}$, completely ignoring any specific characteristics of the class. In this exercise we will consider a specific class of models and obtain a much sharper bound, using techniques commonly used in empirical processes.

Let $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \{0, 1\}$ and consider the 0/1 loss. Define the following class of prediction rules

$$\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y} : f(x) = \mathbf{1}\{x \geq \tau\} \text{ for some } \tau \in \mathbb{R}\} \ .$$

In words, this is the class of models that predict one when the feature value is larger than a threshold τ and zero otherwise, so each rule in this class is uniquely defined by a parameter $\tau \in \mathbb{R}$. For notational convenience we will define $R(\tau) \equiv R(f)$ where $f(x) = \mathbf{1}\{x \geq \tau\}$. The class $\mathcal{F}$ is clearly uncountable and therefore not finite (that is $|\mathcal{F}| = \infty$). Blindly applying Theorem 5.1.1 will therefore yield a completely useless bound. Let's see if we can do better.

As in Theorem 5.1.1 we will make the strong assumption that $\inf_{f \in \mathcal{F}} R(f) = 0$ (in other words $\inf_{\tau \in \mathbb{R}} R(\tau) = 0$. This means that there is a value τ_0 for which $\mathbb{P}(Y = \mathbf{1}\{X \geq \tau_0\}) = 1$. We will show that, for any $\epsilon > 0$

$$\mathbb{P}(R(\hat{f}_n) > \epsilon) < (6n + 3)e^{-n\epsilon} \ ,$$

where $\hat{f}_n$ is the empirical risk minimizer over $\mathcal{F}$.

a) Show that the risk of the prediction rule characterized by τ is given by

$$R(\tau) = \begin{cases} \mathbb{P}(\tau \leq X < \tau_0) & \text{if } \tau < \tau_0 \\ \mathbb{P}(\tau_0 \leq X < \tau) & \text{if } \tau \geq \tau_0 \end{cases} \ .$$

In particular note that the risk is non-increasing up to τ_0 and non-decreasing up after that, and is determined only by $\mathbb{P}_X$ and τ_0.

b) Let $f(x) = \mathbf{1}\{x \geq \tau\}$. Check that the empirical risk given by

$$\hat{R}_n(\tau) = \hat{R}_n(f) = \begin{cases} \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{\tau \leq X_i < \tau_0\} & \text{if } \tau < \tau_0 \\ \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{\tau_0 \leq X_i < \tau\} & \text{if } \tau \geq \tau_0 \end{cases} \ .$$

Note that the empirical risk is also non-increasing up to τ_0 and non-decreasing up after that.

At this point let us do a though experiment. Instead of finding a model in the big class $\mathcal{F}$ let us construct a much smaller class $\mathcal{F}_m$ that is somewhat representative of $\mathcal{F}$. This construction requires knowledge of the distribution $\mathbb{P}_{XY}$, so it is not practical, but it will be very useful to develop the theory. To simplify the story let us assume X is a continuous random variable, so that the risk $R(\tau)$ is also a continuous function (this is not a strictly necessary assumption - see (g)). Let $m \in \mathbb{N}$ be an arbitrary integer and for $i \in \{1, \ldots, m-1\}$ define

$$\tau_i = \inf\{\tau \in \mathbb{R} : \tau > \tau_0 \text{ and } R(\tau) > i/m\} \ ,$$

and convention $\tau_m = \infty$. Likewise define

$$\tau_{-i} = \sup\{\tau \in \mathbb{R} : \tau < \tau_0 \text{ and } R(\tau) > i/m\} \ ,$$

and convention $\tau_{-m} = -\infty$.

Note that by construction $|R(\tau_i) - R(\tau_{i-1}| \leq 1/m$ when X is a continuous random variable. Finally define $\mathcal{G} = \left\{i \in \{-m, \ldots, m\} : \hat{R}_n(\tau_i) = 0\right\}$.

c) Let $\tau > \tau_0$ be such that $\hat{R}_n(\tau) = 0$. Show that

$$R(\tau) \leq \max_{i \in \mathcal{G}} R(\tau_i) + 1/m \ .$$

d) Argue that one also has the same result for $\tau < \tau_0$.

e) Now let $\hat{\tau}_n = \arg\min_{\tau \in \mathbb{R}} \hat{R}_n(\tau)$ be the empirical risk minimizer. Show that

$$\mathbb{P}(R(\hat{\tau}_n) > \epsilon) \leq \mathbb{P}(\max_{i \in \mathcal{G}} R(\tau_i) > \epsilon - 1/m) < (2m+1)e^{-n(\epsilon - 1/m)} \ .$$

f) We are almost done. Since the choice for m is completely arbitrary we can choose a value that makes the bound small. Take $m = n$ and conclude that

$$\mathbb{P}(R(\hat{\tau}_n) > \epsilon) < (6n+3)e^{-n\epsilon} \ .$$

g) What needs to be modified in the argument to ensure the result also holds when X is not necessarily a continuous random variable?

Chapter 6

Concentration Bounds

In this chapter we take a bit of a detour from the statistical learning setting, and focus on the derivation of probabilistic bounds for the sum of independent random variables. Nevertheless, recall that the empirical risk can be seen as the (normalized) sum of random variables, so these bounds will be very useful to understand the properties of the empirical risk. We'll start from first principles with some very basic results, and we'll see that, through a clever use of these basic and crude inequalities, we can get very strong results.

6.1 Markov and Chebyshev's inequalities

Lemma 6.1.1 (Markov's inequality) *Let $Z \geq 0$ be a non-negative random variable and $t > 0$. Then*

$$\mathbb{P}(Z \geq t) \leq \frac{\mathbb{E}[Z]}{t} \ .$$

Proof

$$\begin{aligned} \mathbb{E}[Z] &\geq \mathbb{E}[Z\mathbf{1}\{Z \geq t\}] \\ &\geq \mathbb{E}[t\mathbf{1}\{Z \geq t\}] \\ &= t\mathbb{P}(Z \geq t) \ . \end{aligned}$$

Rearranging the terms give the statement in the lemma. □

This very basic result applies only to non-negative random variables, and relates the "tail" probability $\mathbb{P}(Z > t) = 1 - F(t)$ and the expected value of Z, where F is the c.d.f. of Z. This is, however, a very powerful tool to have, as we'll see next.

Lemma 6.1.2 (Chebyshev's inequality) *Let X be a random variable with finite mean $\mathbb{E}[X]$ and $t > 0$. Then*

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq t) \leq \frac{\mathbb{V}(X)}{t^2} \ ,$$

where $\mathbb{V}(X)$ denotes the variance of X.

Proof The proof is a simple application of Markov's inequality

$$\begin{aligned}\mathbb{P}(|X - \mathbb{E}[X]| \geq t) &= \mathbb{P}(|X - \mathbb{E}[X]|^2 \geq t^2) \\ &= \mathbb{P}((X - \mathbb{E}[X])^2 \geq t^2) \\ &\leq \frac{\mathbb{E}[(X - \mathbb{E}[X])^2]}{t^2} \\ &= \frac{\mathbb{V}(X)}{t^2} \ .\end{aligned}$$

□

This result seems quite sensible, and gives a nice "meaning" to the variance of a random variable. If the variance is small, it seems that the probability X deviates much from its expected value $\mathbb{E}[X]$ is also small. Actually, we can use this inequality to study what happens when we average random variables, just like in the definition of empirical risk.

6.2 A basic concentration inequality for averages

Let $X_1, \ldots, X_n$ be i.i.d. random variables with finite mean $\mathbb{E}[X_i] = \mu$ and variance $\mathbb{V}(X_i) = \sigma^2$. Define $S_n = \sum_{i=1}^n X_i$ to be the sum of these variables. The weak law of large numbers tells us that S_n/n converges to μ in probability as $n \to \infty$. This statement means that, for any $\varepsilon > 0$

$$\mathbb{P}\left(\left|\frac{S_n}{n} - \mu\right| \geq \varepsilon\right) \to 0 \ ,$$

as $n \to \infty$. This can be shown simply by using Chebyshev's inequality. Note that

$$\mathbb{E}\left[\frac{S_n}{n}\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[X_i\right] = \mu \ .$$

Also

$$\mathbb{V}\left[\frac{S_n}{n}\right] = \mathbb{V}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] = \frac{1}{n^2}\sum_{i=1}^n \mathbb{V}\left[X_i\right] = \frac{\sigma^2}{n} \ ,$$

where the second equality follows because we assume the random variables are independent. So, we can simply apply Chebyshev's inequality to the random variable S_n/n and get

$$\mathbb{P}\left(\left|\frac{S_n}{n} - \mu\right| \geq \varepsilon\right) \leq \frac{\sigma^2}{n\varepsilon^2} \ . \tag{6.1}$$

Clearly this converges to zero as $n \to \infty$. The result in (6.1) is what is known as a concentration inequality, and indicates how S_n/n concentrates around the mean μ for a finite value of n. So we should already be able to use such a result to get PAC bounds. However, we must note that this concentration inequality is generally quite loose, and the probability in the l.h.s. of (6.1) is normally much much smaller than $\frac{\sigma^2}{n\varepsilon^2}$ for any given distribution when n is large. To informally see this recall the central limit theorem. It states that the distribution of

$$\frac{1}{\sqrt{n}}\sum_{i=1}^n \frac{X_i - \mu}{\sigma}$$

approaches that of a standard normal random variable. Let Z be a standard normal random variable (that is, a continuous random variable with density $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$. Informally, this means that, for large n

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma} \overset{\text{D}}{\approx} Z \ ,$$

where the symbol $\overset{\text{D}}{\approx}$ means the two random quantities have approximately the same distribution. Note that

$$\frac{S_n}{n} - \mu = \frac{1}{n} \sum_{i=1}^{n} (X_i - \mu) = \frac{\sigma}{\sqrt{n}} \ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma} \overset{\text{D}}{\approx} \frac{\sigma}{\sqrt{n}} Z \ .$$

Therefore

$$\mathbb{P}\left(\left| \frac{S_n}{n} - \mu \right| \geq \varepsilon \right) \approx \mathbb{P}\left(\left| \frac{\sigma}{\sqrt{n}} Z \right| \geq \varepsilon \right) = \mathbb{P}\left(|Z| \geq \frac{\sqrt{n}\varepsilon}{\sigma} \right) \ .$$

The probability on the r.h.s. will be extremely small, even for moderate values of $\frac{\sqrt{n}\varepsilon}{\sigma}$, because the density of normal random variables decays exponentially fast away from the origin. The following lemma for standard normal random variables comes in handy

Lemma 6.2.1 *Let Z be a standard normal random variable. Then, for any $\gamma > 0$ we have*

$$\frac{1}{\sqrt{2\pi\gamma^2}} \left(1 - \frac{1}{\gamma^2} \right) e^{-\frac{\gamma^2}{2}} \leq \mathbb{P}(Z \geq \gamma) \leq \frac{1}{\sqrt{2\pi\gamma^2}} e^{-\frac{\gamma^2}{2}} \ ,$$

and the bounds are very close to each other if γ is large. A slightly simpler upper bound is

$$\mathbb{P}(Z \geq \gamma) \leq \frac{1}{2} e^{-\frac{\gamma^2}{2}} \ .$$

Using this result we conclude that

$$\mathbb{P}\left(|Z| \geq \frac{\sqrt{n}\varepsilon}{\sigma} \right) = 2\mathbb{P}\left(Z \geq \frac{\sqrt{n}\varepsilon}{\sigma} \right) \leq e^{-\frac{n\varepsilon^2}{2\sigma^2}} \ .$$

So, if n is large enough so that the central limit theorem is approximately valid we conclude we should have something approximately like

$$\mathbb{P}\left(\left| \frac{S_n}{n} - \mu \right| \geq \varepsilon \right) \lesssim e^{-\frac{n\varepsilon^2}{2\sigma^2}} \ . \tag{6.2}$$

We see that this indicates that S_n/n concentrates exponentially fast around μ, much faster than (6.1) indicates. So we are certain that we are off by quite a bit. This calls for a much sharper concentration inequality.

6.3 Chernoff bounding and Hoeffding's inequality

Recall that, in the proof of Chebyshev's inequality, we wanted to bound $\mathbb{P}(|X - \mathbb{E}[X]| \geq t)$ and we simply squared both sides of the relation to get $\mathbb{P}(|X - \mathbb{E}[X]|^2 \geq t^2)$, so that we could apply Markov's inequality. Instead of squaring, we can use a different monotone transformation. Actually, we'll use a transformation that will allow us to deal with the sum of independent

random variables in a nice way. The technique we are going to use is called Chernoff bounding (in reference to Herman Chernoff) that made it popular. However, it seems it's origins can be traced to the work of Sergei Bernstein published in the 1920-1930 (in Russian) and rediscovered several times after that.

Theorem 6.3.1 (Chernoff Bound) *Let $X_1, \ldots, X_n$ be independent random variables, and define $S_n = \sum_{i=1}^n X_i$. Then*

$$\begin{aligned}\mathbb{P}(S_n \geq t) &\leq \inf_{s>0} \left\{ e^{-st} \mathbb{E}\left[e^{sS_n}\right] \right\} \\ &= \inf_{s>0} \left\{ e^{-st} \prod_{i=1}^n \mathbb{E}\left[e^{sX_i}\right] \right\} \ .\end{aligned}$$

Proof The basic idea is the same as in the proof of Chebyshev's inequality, but instead of squaring both side of the inequality inside the probability symbol we'll use the monotone increasing transformation $t \mapsto e^{st}$.

$$\begin{aligned}\mathbb{P}(S_n \geq t) &= \mathbb{P}\left(e^{sS_n} \geq e^{st}\right) \\ &\leq \frac{\mathbb{E}\left[e^{sS_n}\right]}{e^{st}} \\ &= e^{-st}\mathbb{E}\left[e^{s\sum_{i=1}^n X_i}\right] \\ &= e^{-st}\mathbb{E}\left[\prod_{i=1}^n e^{sX_i}\right] \\ &= e^{-st}\prod_{i=1}^n \mathbb{E}\left[e^{sX_i}\right] \ ,\end{aligned}$$

where the last equality follows simply because the random variables are independent (recall that, is X and Y are independent then $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$). Since $s > 0$ is completely arbitrary, we can take the value that minimizes the bound to obtain the best result, concluding the proof. □

The key idea was the use of an exponential function, so that the expectation becomes the product of expectations. From this point on, what we need to do is to get a good control on the terms $\mathbb{E}\left[e^{sX_i}\right]$. This requires us to make some assumptions about the random variables X_i.

Theorem 6.3.2 (Hoeffding's Inequality) *Let $X_1, X_2, \ldots, X_n$ be independent bounded random variables such that $X_i \in [a_i, b_i]$ with probability 1. Let $S_n = \sum_{i=1}^n X_i$. Then for any $t > 0$, we have*

a) $\mathbb{P}(S_n - \mathbb{E}[S_n] \geq t) \leq e^{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}$

b) $\mathbb{P}(S_n - \mathbb{E}[S_n] \leq -t) \leq e^{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}$

c) $\mathbb{P}\left(|S_n - \mathbb{E}[S_n]| \geq t\right) \leq 2e^{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}}$

Note that this theorem seems to capture much better the intuition we gained from the central limit theorem in (6.2). In particular, consider the situation where $X_i \sim \text{Ber}(1/2)$ (just like the flips of a fair coin). Then $\mathbb{V}(X_i) = 1/4$ and (6.2) tells us that

$$\mathbb{P}\left(|S_n/n - 1/2| \geq \varepsilon\right) \lesssim e^{-2n\varepsilon^2} \ .$$

Recall the above result is only an approximation that is reasonable for large n. On the other hand, Hoeffding's inequality tells us that

$$\mathbb{P}\left(|S_n/n - 1/2| \geq \varepsilon\right) \leq 2e^{-2n\varepsilon^2} \ ,$$

which means we lost only a factor of two (but the last result is not an approximation).

Before proving the theorem let's see a few particular applications of this result.

Example 6.3.1 *Suppose you are given a coin, and are told it has some bias, meaning either heads are more likely than tails, or vice versa. How many flips of the coin do you need to decide the direction of the bias (that is, to decide whether the probability of flipping heads is higher or lower than that of flipping tails)?*

Obviously, the number of flips of the coin we will need depends on the amount of bias of the coin. If it flips heads all the time we'll quickly realize this, but if it flips heads 51% of the times it might take a while until we are certain that heads are more common. Let's formalize things a bit. Identify heads with 1 and tails with 0. Suppose we flip the coin n times. The outcome of the flips can be viewed as independent Bernoulli random variables $X_1, \ldots, X_n$ with success probability p. Since the coin has a bias we know that $|p - 1/2| = \varepsilon$, where $\varepsilon > 0$ quantifies the amount of bias we have. A natural way to proceed is to estimate p using the sample mean $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n X_i$. If this is larger or equal than $1/2$ then we deem the coin to be head-biased, otherwise we deem the coin to be tail-biased.

How many flips of the coin will we need to guarantee we decide correctly with probability at least $1 - \delta$, for some prescribed small value $\delta > 0$? Suppose $p = 1/2 - \varepsilon$, for $\varepsilon > 0$. We'll make an error if $\hat{p} \geq 1/2$. What is the probability of error? We can use Hoeffding's inequality to quantify this.

$$\begin{aligned}
\mathbb{P}(\hat{p} \geq 1/2) &= \mathbb{P}(\hat{p} - p \geq 1/2 - p) \\
&= \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n (X_i - p) \geq \varepsilon\right) \\
&= \mathbb{P}\left(\sum_{i=1}^n (X_i - p) \geq n\varepsilon\right) \\
&\leq e^{-\frac{2(n\varepsilon)^2}{n}} = e^{-2n\varepsilon^2} \ .
\end{aligned}$$

So, to ensure the probability of error is no larger than δ we should take n so that $e^{-2n\varepsilon^2} \leq \delta$, which means

$$n \geq \frac{1}{2\varepsilon^2}\log\left(\frac{1}{\delta}\right) \ . \tag{6.3}$$

The exact same reasoning applies if $p = 1/2 + \varepsilon$, and so we know that, in order to determine the direction of the bias we should take n flips of the coin satisfying the relation in (6.3).

Let's use the above result in a simple setting. Suppose you are the croupier *in a blackjack table at a casino, and at your table there is a player you suspect is counting cards (this is not strictly illegal, but the casino can ban the player). Most card-counting techniques give players a nice advantage on the* house*, namely, the probability of winning each hand is about 55%. Being a mathematically inclined croupier, you let the player continue until you are 90% sure he's cheating. How many hands should you let him play? In this case, $\delta = 0.1$ and $\varepsilon = 0.05$, therefore, you should let the player gamble at least $\frac{1}{2\varepsilon^2}\log\left(\frac{1}{\delta}\right) = 460.5$ hands.*

The only problem with the above formula is that we need to know (or have a bound on) the amount of bias ε of the coin. However, there is a simple and smart way of getting around this problem and still having a very similar guarantee in terms of the number of samples.

Example 6.3.2 *Consider the setting in the example above. We can use Hoeffding's inequality to construct an* interval estimate *for p, namely write*

$$\mathbb{P}(|\hat{p}_n - p| \geq t) \leq 2e^{-2nt^2} \ .$$

Now let $\delta = 2e^{-2nt^2}$ and solve for t, so that $t = \sqrt{\frac{\log(2/\delta)}{2n}}$. We conclude that

$$\mathbb{P}\left(p \in \left[\hat{p}_n - \sqrt{\frac{\log(2/\delta)}{2n}}, \hat{p}_n + \sqrt{\frac{\log(2/\delta)}{2n}}\right]\right) \geq 1 - \delta \ .$$

This means that such an interval contains the true unknown parameter with probability at least $1-\delta$. This interval is valid for any fixed sample size n, but not for various values of n simultaneously (see exercise 6.4.3). However, for any sequence of sample sizes, it is possible to construct a sequence of intervals such that ALL of them contain the true parameter p with probability at least $1-\delta$. A way to do it is to use a simple union bound.

What we desire to do is to construct intervals I_n such that

$$\mathbb{P}\left(\forall_n \in \mathbb{N} \quad p \in I_n\right) \geq 1 - \delta \ .$$

In other words, the intervals I_n are confidence intervals that are valid for all n simultaneously. To get such a result we will use a very simple union bound argument.

$$\begin{aligned}
\mathbb{P}\left(\forall n \in \mathbb{N} \quad p \in I_n\right) &= 1 - \mathbb{P}\left(\exists n \in \mathbb{N} \ : \ p \notin I_n\right) \\
&= 1 - \mathbb{P}\left(\bigcup_{n\in\mathbb{N}} \{p \notin I_n\}\right) \\
&\geq 1 - \sum_{n=1}^{\infty} \mathbb{P}\left(p \notin I_n\right) \ .
\end{aligned}$$

Now note that the terms in the sum can be bounded as above by Hoeffding's inequality, but with a careful choice of the confidence level so that all these probabilities sum to δ. Let $\delta_n = \frac{\delta}{n(n+1)}$

and define

$$I_n = \left[\hat{p}_n - \sqrt{\frac{\log\left(\frac{2}{\delta_n}\right)}{2n}}, \hat{p}_n + \sqrt{\frac{\log\left(\frac{2}{\delta_n}\right)}{2n}}\right]$$
$$= \left[\hat{p}_n - \sqrt{\frac{\log(n(n+1)) + \log\left(\frac{2}{\delta}\right)}{2n}}, \hat{p}_n + \sqrt{\frac{\log(n(n+1)) + \log\left(\frac{2}{\delta}\right)}{2n}}\right] \tag{6.4}$$

Then

$$\mathbb{P}\left(\forall n \in \mathbb{N}\ p \in I_n\right) \geq 1 - \sum_{n=1}^{\infty} \mathbb{P}\left(p \notin I_n\right)$$
$$\geq 1 - \sum_{n=1}^{\infty} \delta_n = 1 - \delta \sum_{n=1}^{\infty} \frac{1}{n(n+1)} = 1 - \delta\ .$$

This means that we can just go on flipping the coin until $1/2$ *is not contained in one of the intervals, and decide the direction of the bias based on the final interval. It is not hard to show that with probability at least* $1 - \delta$ *you will make the correct decision, and will stop after only slightly more flips than the ones given by* (6.3)*. See also Kääriäinen (2006) for a slightly different approach to the same problem.*

Actually, in light of the law of the iterated logarithm, the $\log(n(n+1))$ *appears to be a bit of overkill and we would expect something like* $\log\log(n)$ *instead. The story is, however, a bit more complicated as we want results that are not asymptotic. See for instance Balsubramani and Ramdas (2015).*

Proof of Theorem 6.3.2. Define the random variables $Z_i = X_i - \mathbb{E}[X_i]$. The first step is to use the Chernoff bound for these variables. Take any $s > 0$. We conclude that

$$\mathbb{P}(S_n - \mathbb{E}[S_n] \geq t) = \mathbb{P}\left(\sum_{i=1}^{n} Z_i \geq t\right) \leq e^{-st} \prod_{i=1}^{n} \mathbb{E}\left[e^{sZ_i}\right]\ .$$

To complete the proof we need to find a good bound for $\mathbb{E}\left[e^{sZ_i}\right]$, and that's where most of the work will go into.

Lemma 6.3.1 *Let* X *be a r.v. such that* $\mathbb{E}[X] = 0$ *and* $a \leq X \leq b$ *with probability one. Then*

$$\mathbb{E}\left[e^{sX}\right] \leq e^{\frac{s^2(b-a)^2}{8}}\ .$$

The proof is somewhat cluttered by a number of technicalities. See Remark 6.3.1 for the proof of a particular case, that highlights all the aspects of the general proof without as much clutter.

Proof of Lemma 6.3.1 This upper bound hinges on the convexity of the exponential function, implying that

$$e^{sx} \leq \frac{x-a}{b-a} e^{sb} + \frac{b-x}{b-a} e^{sa},\ \text{for } a \leq x \leq b\ .$$

Thus,

$$\begin{aligned}\mathbb{E}[e^{sX}] &\leq \mathbb{E}\left[\frac{X-a}{b-a}\right]e^{sb} + \mathbb{E}\left[\frac{b-X}{b-a}\right]e^{sa} \\ &= \frac{b}{b-a}e^{sa} - \frac{a}{b-a}e^{sb} \ , \text{ since } \mathbb{E}[X]=0 \\ &= (1-\lambda+\lambda e^{s(b-a)})e^{-\lambda s(b-a)} \ , \text{ where } \lambda = \frac{-a}{b-a} \ .\end{aligned}$$

The last step might seem mysterious, but it is simply a way to re-parameterize the expression. See the remark below. Now let $u = s(b-a)$ and define

$$h(u) \equiv -\lambda u + \log(1-\lambda+\lambda e^u) \ ,$$

so that

$$\mathbb{E}[e^{sX}] \leq (1-\lambda+\lambda e^{s(b-a)})e^{-\lambda s(b-a)} = e^{h(u)} \ .$$

We want to find a good upper-bound for $h(u)$. We'll use a Taylor expansion around zero for that

$$h(u) = h(0) + uh'(0) + \frac{u^2}{2}h''(v) \ \text{ for some } v \in [0,u] \ .$$

$$\begin{aligned}h'(u) &= -\lambda + \frac{\lambda e^u}{1-\lambda+\lambda e^u} \ \Rightarrow h'(0) = 0 \\ h''(u) &= \frac{\lambda e^u}{1-\lambda+\lambda e^u} - \frac{(\lambda e^u)^2}{(1-\lambda+\lambda e^u)^2} \\ &= \frac{\lambda e^u}{1-\lambda+\lambda e^u}(1 - \frac{\lambda e^u}{1-\lambda+\lambda e^u}) \\ &= \rho(1-\rho) \ ,\end{aligned}$$

where $\rho = \frac{\lambda e^u}{1-\lambda+\lambda e^u}$. Now note that $\rho(1-\rho) \leq 1/4$, for any value of ρ (the maximum is attained when $\rho = 1/2$, therefore $h''(u) \leq 1/4$. So finally we have $h(u) \leq \frac{u^2}{8} = \frac{s^2(b-a)^2}{8}$, and therefore

$$\mathbb{E}[e^{sX}] \leq e^{\frac{s^2(b-a)^2}{8}} \ ,$$

concluding the proof. □

Remark 6.3.1 *The proof of the above lemma is a bit cluttered by all the technicalities. To better understand the reasoning consider a particular case: $a = -1$ and $b = 1$. By convexity of the exponential $e^{sx} \leq \frac{e^s+e^{-s}}{2} + \frac{e^s-e^{-s}}{2}x$. Therefore, taking into account that $\mathbb{E}[X]$=0 we have*

$$\mathbb{E}[e^{sX}] \leq \mathbb{E}\left[\frac{e^s+e^{-s}}{2} + \frac{e^s-e^{-s}}{2}X\right] = \frac{e^s+e^{-s}}{2} \ .$$

Since we want an exponential bound it is not useful to use directly a Taylor expansion on the above expression. First note that

$$\frac{e^s+e^{-s}}{2} = \frac{e^{-s}}{2}(1+e^{2s}) = \frac{1}{2}e^{\underbrace{-s+\log(1+e^{2s})}_{g(s)}} \ .$$

Now let's instead expand $g(s)$ around zero. Note that $g(0) = \log 2$,

$$g'(s) = \frac{2e^{2s}}{1+e^{2s}} \quad \text{and} \quad g''(s) = \frac{4e^{2s}}{(1+e^{2s})^2} \ .$$

It is not hard to see that $g'(0) = 1$ and that $g''(s) \leq 1$ regardless of the value of s. Therefore

$$g(s) = g(0) + sg'(0) + \frac{s^2}{2}g''(\tau) \leq \log 2 + s + s^2/2 \ ,$$

where $\tau \in [0, s]$. Plugging this in gives

$$\frac{e^{-s}+e^{s}}{2} = \frac{1}{2}e^{-s+\log(1+e^{2s})} \leq \frac{1}{2}e^{-s+\log 2+s+s^2/2} = e^{s^2/2} \ .$$

This is exactly the result of the lemma, for this particular case.

One we have the result of the lemma, we can finish the proof of Hoeffding's inequality.

$$\begin{aligned}
\mathbb{P}(S_n - \mathbb{E}[S_n] \geq t) &\leq e^{-st}\prod_{i=1}^{n}\mathbb{E}[e^{s(X_i-\mathbb{E}[X_i])}] \\
&\leq e^{-st}\prod_{i=1}^{n}e^{\frac{s^2(b_i-a_i)^2}{8}} \\
&= e^{-st}e^{s^2\sum_{i=1}^{n}\frac{(b_i-a_i)^2}{8}} \ .
\end{aligned}$$

All the remains to be done is to choose a good value for s. In this case this is easily done by minimizing the above expression, yielding the choice $s = \frac{4t}{\sum_{i=1}^{n}(b_i-a_i)^2}$, and giving as a result the bound

$$\mathbb{P}(S_n - \mathbb{E}[S_n] \geq t) \leq e^{\frac{-2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}} \ .$$

This concludes the proof of (a). To show (b) one just needs to apply (a) to the r.v.'s $(-X_1), \ldots, (-X_n)$. Finally (c) follows by using (a) and (a) simultaneously and the union of events bound. Namely

$$\begin{aligned}
\mathbb{P}\left(|S_n - \mathbb{E}[S_n]| \geq t\right) &= \mathbb{P}\left(\{S_n - \mathbb{E}[S_n] \geq t\} \cup \{S_n - \mathbb{E}[S_n] \leq -t\}\right) \\
&\leq \mathbb{P}\left(S_n - \mathbb{E}[S_n] \geq t\right) + \mathbb{P}\left(S_n - \mathbb{E}[S_n] \leq -t\right) \\
&\leq 2e^{\frac{-2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}} \ .
\end{aligned}$$

□

6.4 Exercises

Exercise 6.4.1 *Recall Chebyshev's inequality we proved above. It bounds the probability that $|X - \mathbb{E}[X]| \geq t$, which is a two sided event. In this exercise you will derive a one-sided version of this inequality.*

Assume first the random variable Y has zero mean, meaning $\mathbb{E}[Y] = 0$. Assume also that it has variance $\mathbb{V}(Y) = \sigma^2 < \infty$. Finally let $t > 0$.

a) Recall the Cauchy-Schwarz inequality. For any two random variables A and B we have that

$$\mathbb{E}[AB] \leq \sqrt{\mathbb{E}(A^2)\mathbb{E}(B^2)} \ .$$

Now write $t = t - \mathbb{E}[Y] = \mathbb{E}[t - Y] \leq \mathbb{E}[(t - Y)\mathbf{1}\{t > Y\}]$, and use Cauchy-Schwarz to show that

$$t^2 \leq \mathbb{E}[(t - Y)^2]\mathbb{P}(Y < t) \ .$$

b) Using the fact that $\mathbb{E}[Y] = 0$ manipulate the above expression to obtain inequality

$$\mathbb{P}(Y \geq t) \leq \frac{\sigma^2}{\sigma^2 + t^2} \ .$$

c) Now make only the assumption that X is a random variable for which $\mathbb{V}(X) = \sigma^2 < \infty$. Show that

$$\mathbb{P}(X - \mathbb{E}[X] \geq t) \leq \frac{\sigma^2}{\sigma^2 + t^2} \ . \tag{6.5}$$

Hint: *define a suitable random variable Y as a function of X that has zero mean, and apply the result in (b).*

d) Use the above result to derive a two-sided version of this inequality. Namely, use the union bound to show that

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq t) \leq \frac{2\sigma^2}{\sigma^2 + t^2} \ .$$

e) Let Z be a standard normal random variable and use the result of (c) to get an upper bound on $\mathbb{P}(Z > 1/2)$. Noting that Z is symmetric around the origin we have $\mathbb{P}(Z > 1/2) = \frac{1}{2}\mathbb{P}(|Z| > 1/2)$. Use this and the original Chebyshev inequality to get a bound on $\mathbb{P}(Z > 1/2)$. Which is a better bound? (note that we can actually compute $P(Z > 1/2)$ numerically and get approximately 0.3085).

Remark: *Note that the inequality in (6.5) has the nice feature that the r.h.s. is always smaller than 1, so the bound is never trivial.*

Exercise 6.4.2 *[Hide-and-seek] Consider the following problem. You are given two coins, one is fair, but the other one is fake and flips heads with probability $1/2 + \varepsilon$, where $\varepsilon > 0$. However, you don't know the value of ε. You would like to identify the fake coin quickly.*

Consider the following strategy. Flip both coins n times and compute the proportion of heads of each coin (say $\hat{p}_1$ and $\hat{p}_2$ for coins 1 and 2, respectively). Now deem the coin for which the proportion of heads is larger to be the fake coin. What is the probability we'll make a mistake? Suppose without loss of generality that coin 1 is the fake.

a) We'll make a mistake if $\hat{p}_1 < \hat{p}_2$. That is

$$\mathbb{P}(\hat{p}_1 - \hat{p}_2 < 0) \ .$$

Noting that $n(\hat{p}_1 - \hat{p}_2)$ is the sum of $2n$ independent random variables use Hoeffding's inequality to show that the probability of making an error is bounded by $e^{-n\varepsilon^2}$.

b) Now suppose you have m coins, where only one coin is a fake. Similar to what we have done for the two coins we can flip each coins n times, and compute the proportion of times each coin flips heads, denoted by $\hat{p}_1, \ldots, \hat{p}_m$. What is the probability of making an error then?

Suppose without loss of generality that the first coin is fake. The probability of making an error is given by

$$\mathbb{P}(\hat{p}_1 < \hat{p}_2 \text{ or } \hat{p}_1 < \hat{p}_3 \text{ or } \cdots \text{ or } \hat{p}_1 < \hat{p}_m) = \mathbb{P}\left(\bigcup_{i=2}^{m} \{\hat{p}_1 < \hat{p}_i\}\right) .$$

Use Hoeffding's inequality an the union bound to see that the probability of making an error smaller than $(m-1)e^{-n\varepsilon^2}$.

c) Implement the above procedure with $\varepsilon = 0.1$, $m = 2$ or $m = 100$, and the following values of $n = 10, 100, 500, 1000$. For each choice of parameters m and n repeat the procedure $N = 10000$ times and compute the proportion of runs where the procedure failed to identify the correct coin. Compare these with the bounds you got. How good are the bounds you derived?

***Note:** you can use any software package or language to code this. Report your results and comment on them. For completeness, attach the code in your handout.*

Exercise 6.4.3 *Let $X_1, X_2, \ldots$ be independent Bernoulli random variables with parameter p, Let $\hat{p}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ and define the intervals*

$$\left[\hat{p}_n - \sqrt{\frac{\log(2/\delta)}{2n}}, \hat{p}_n + \sqrt{\frac{\log(2/\delta)}{2n}}\right] ,$$

with $\delta > 0$. Recall from Example 6.3.2 that for every particular value of n the corresponding interval is guaranteed to contain the true parameter p with probability at least $1-\delta$. However, this does not mean the probability all these intervals simultaneously contain the true parameter is larger than $1-\delta$. Let's experimentally verify this.

a) Take $p = 1/2$, $\delta = 0.1$ and generate data $x_1, \ldots, x_{10000}$. Compute the above intervals for all the values of $n = 3, \ldots, 10000$ and verify if all of these contain the value $1/2$. Repeat the experiment many times and count the proportion of times all the intervals contain $1/2$. How does this compare with the value $1-\delta$? You will see this proportion is smaller than $1-\delta$, indicating that, very likely, the probability all the intervals will contain $1/2$ is actually smaller than $1-\delta$.

b) Repeat the above experiment, but using the intervals defined in (6.4) instead. How does this compare to what you have seen in (a)?

c) Now replace the term $\log(n(n+1))$ in the intervals by $\log(\log n)$ (inspired by the law of the iterated logarithm). What do you observe?

Chapter 7

General bounds for bounded losses

In the previous chapter we showed a number of *concentration inequalities*, that allow us to tell how fast does the average of independent random variables converge to its mean. Since the empirical risk $\hat{R}_n(f)$ is an average of random variables, this will allow us to quantify how close is the empirical risk to the true risk.

7.1 Bounded Loss Functions

In this chapter we will consider loss functions that are bounded, meaning they cannot take arbitrarily large values. Without loss of generality let $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. The zero-one loss is a particularly useful example of such a loss function (e.g. $\ell(y_1, y_2) = \mathbf{1}\{y_1 \neq y_2\}$).

Remark 7.1.1 *Sometimes the loss function itself is not bounded, but the label space is. In that case, more often than not, the problem reduces to that of a bounded loss function. For instance, consider a regression setting where we use the square loss $\ell(y_1, y_2) = (y_1 - y_2)^2$ and know that $\mathcal{Y} = [-R, R]$ for some value $R > 0$. In that case, the largest possible loss we'll observe is $4R^2$, meaning that we can assume we have a bounded loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 4R^2]$.*

Given a set of training data $\{X_i, Y_i\}_{i=1}^n$ and a finite collection of candidate functions $\mathcal{F}$, we can select $\hat{f}_n \in \mathcal{F}$ that is (hopefully) a good prediction rule for the label Y associated with feature X for a (X, Y) sampled from the same distribution as the training data. Ideally, we would like to take

$$\tilde{f} = \arg\min_{f \in \mathcal{F}} R(f) \ ,$$

that is, the element of $\mathcal{F}$ which has the smallest risk. However, we cannot evaluate the risk because we don't know the distribution $\mathbb{P}_{XY}$. A simple idea is to use the empirical risk as a surrogate for the true risk. That is, we compute instead

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \ ,$$

where $\hat{R}_n(f)$ is the empirical risk, defined as

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(X_i), Y_i) = \frac{1}{n} \sum_{i=1}^n L_i \ ,$$

where $L_i = \ell(f(X_i), Y_i)$.

Note that the empirical risk is a random quantity. For any fixed f (that is, f is taken to be deterministic) we know that

$$\mathbb{E}[\hat{R}_n(f)] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[L_i] \ ,$$

and that

$$\mathbb{E}[L_i] = \mathbb{E}[\ell(f(X_i), Y_i)] = \mathbb{E}[\ell(f(X), Y)] = R(f) \ .$$

Furthermore, the $L_i's$ are i.i.d., because the training data is assumed to be an i.i.d. sample from some distribution $\mathbb{P}_{XY}$. This, and the fact that $L_i \in [0, 1]$ (because we assume the loss function is bounded) allows us to use Hoeffding's inequality to relate the empirical risk to the true risk in a probabilistic way. In particular, we want to study if the empirical risk underestimates the true risk by a significant amount. If the empirical risk overestimates the risk of f that rule won't be chosen, so we do not need to control that probability. Let $\varepsilon > 0$, then

$$\begin{aligned}
\mathbb{P}\left(\hat{R}_n(f) \leq R(f) - \varepsilon\right) &= \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} L_i - \mathbb{E}[L_i] \leq -\varepsilon\right) \\
&= \mathbb{P}\left(\sum_{i=1}^{n} L_i - \mathbb{E}[L_i] \leq -n\varepsilon\right) \\
&\leq e^{-\frac{2(n\varepsilon)^2}{n}} = e^{-2n\varepsilon^2} \ .
\end{aligned}$$

Note that this bound applies to a *single* model $f \in \mathcal{F}$. Since our selection process involves the choice among all $f \in \mathcal{F}$ such a result is not enough. A way around this problem is to instead ensure the probability that one or more of the empirical risks significantly underestimates the corresponding true risk. This is captured by the probability

$$\mathbb{P}\left(\exists f \in \mathcal{F} : \ \hat{R}_n(f) \leq R(f) - \varepsilon\right)$$

If the class $\mathcal{F}$ is finite we can use the *union bound* (also known as the Bonferroni's bound) to obtain

$$\begin{aligned}
\mathbb{P}\left(\exists f \in \mathcal{F} : \ \hat{R}_n(f) \leq R(f) - \varepsilon\right) &= \mathbb{P}\left(\bigcup_{f\in\mathcal{F}} \left\{\hat{R}_n(f) \leq R(f) - \varepsilon\right\}\right) \\
&\leq \sum_{f\in\mathcal{F}} \mathbb{P}(\hat{R}_n(f) \leq R(f) - \varepsilon) \\
&\leq \sum_{f\in\mathcal{F}} e^{-2n\varepsilon^2} \\
&= |\mathcal{F}| e^{-2n\varepsilon^2} \ ,
\end{aligned}$$

where $|\mathcal{F}|$ is the number of elements in $\mathcal{F}$.

We can also restate the above inequality as follows. Let $\delta > 0$. Then

$$\forall f \in \mathcal{F} \quad R(f) < \hat{R}_n(f) + \sqrt{\frac{\log|\mathcal{F}| + \log(1/\delta)}{2n}}$$

with probability at least $1-\delta$. This follows by setting $\delta = |\mathcal{F}|e^{-2n\varepsilon^2}$ and solving for ε. Thus with a high probability (greater than $1-\delta$), the true risk for all $f \in \mathcal{F}$ is bounded by the empirical risk of f plus a constant that depends on $\delta > 0$, the number of training samples n, and the size of the class $\mathcal{F}$. Most importantly the bound does not depend on the unknown distribution $\mathbb{P}_{XY}$, and holds regardless of which distribution gave rise to the data. Therefore we can call this a *distribution free* bound.

As a remark, note that we have also shown a two-sided version of Hoeffding's inequality. If we use that instead we get that for $\delta > 0$

$$\forall f \in \mathcal{F} \quad \left|\hat{R}_n(f) - R(f)\right| < \sqrt{\frac{\log|\mathcal{F}| + \log(2/\delta)}{2n}}$$

with probability at least $1-\delta$. This means that the empirical risk and the true risk will be close with probability at least $1-\delta$, provided n and $\mathcal{F}$ are large. In Chapter 5 we saw that, if we had such a result then we could easily justify the rational behind empirical risk minimization (see equation (5.2)).

7.2 Expected Risk Bounds for Empirical Risk Minimization

In this section we will derive a bound for the expected risk of $\hat{f}_n$, and will also derive a PAC bound for empirical risk minimization. Recall that

$$\hat{f}_n = \arg\min_{f\in\mathcal{F}} \hat{R}_n(f) \ .$$

The bound derived above can be written as

$$\forall f \in \mathcal{F} \quad R(f) < \hat{R}_n(f) + C(\mathcal{F}, n, \delta) \ ,$$

with probability at least $1-\delta$, where

$$C(\mathcal{F}, n, \delta) = \sqrt{\frac{\log|\mathcal{F}| + \log(1/\delta)}{2n}} \ .$$

This result holds for all $f \in \mathcal{F}$ therefore it also holds in particular for $\hat{f}_n$, and so

$$R(\hat{f}_n) < \hat{R}_n(\hat{f}_n) + C(\mathcal{F}, n, \delta) \ ,$$

w.p. $\geq 1-\delta$, and for any other $f \in \mathcal{F}$

$$R(\hat{f}_n) < \hat{R}_n(f) + C(\mathcal{F}, n, \delta) \ ,$$

since $\hat{R}_n(\hat{f}_n) \leq \hat{R}_n(f)\ \forall f \in \mathcal{F}$.

This clearly motivates why empirical risk minimization might be a good idea. Up to a small "slack" $C(\mathcal{F}, n, \delta)$ minimizing the empirical risk is equivalent to minimizing the true risk, with high probability. To summarize, we have shown that

$$R(\hat{f}_n) < \hat{R}_n(\tilde{f}) + C(\mathcal{F}, n, \delta) \ , \tag{7.1}$$

w.p. at least $1-\delta$, where $\tilde{f} = \arg\min_{f\in\mathcal{F}} R(f)$.

At this point we can try to characterize how well does empirical risk minimization do. Let us start by looking at the expected risk. Start by defining Ω to be the event given by (7.1). Our application of the Hoeffding's inequality characterizes that probability, that is $\mathbb{P}(\Omega) \geq 1-\delta$. We can now bound $\mathbb{E}[R(\hat{f}_n)] - R(\tilde{f})$ as follows

$$\begin{aligned}\mathbb{E}[R(\hat{f}_n)] - R(\tilde{f}) &= \mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f}) + \hat{R}_n(\tilde{f}) - R(\tilde{f})] \\ &= E[R(\hat{f}_n) - \hat{R}_n(\tilde{f})] \ ,\end{aligned}$$

since $\mathbb{E}[\hat{R}_n(\tilde{f})] = R(\tilde{f})$. The quantity above is bounded as follows.

$$\begin{aligned}\mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f})] &= \mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f})|\Omega]\,\mathbb{P}(\Omega) + \mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f})|\bar{\Omega}]\,\mathbb{P}(\bar{\Omega}) \\ &\leq \mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f})|\Omega] + \delta \ ,\end{aligned}$$

where $\bar{\Omega}$ denotes the complement of Ω. The above inequality follows since $\mathbb{P}(\Omega) \leq 1$, $1-\mathbb{P}(\Omega) \leq \delta$ and $R(\hat{f}_n) - \hat{R}_n(\tilde{f}) \leq 1$ (the losses are bounded). Now

$$\begin{aligned}\mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f})|\Omega] &\leq \mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\hat{f}_n)|\Omega] \\ &< C(\mathcal{F}, n, \delta) \ ,\end{aligned}$$

and so we have

$$\mathbb{E}[R(\hat{f}_n) - \hat{R}_n(\tilde{f})] < C(\mathcal{F}, n, \delta) + \delta \ .$$

We have essentially shown the following general result

Proposition 7.2.1 *Let $\{X_i, Y_i\}_{i=1}^n$ be i.i.d. samples from an arbitrary distribution $\mathbb{P}_{XY}$, and let $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ be a bounded loss function. Then*

$$\mathbb{E}[R(\hat{f}_n)] - \min_{f\in\mathcal{F}} R(f) < \inf_{\delta\in(0,1)} \left\{ \sqrt{\frac{\log|\mathcal{F}| + \log(1/\delta)}{2n}} + \delta \right\} \ .$$

In particular, taking $\delta = \sqrt{1/n}$ yields the bound

$$\mathbb{E}[R(\hat{f}_n)] - \min_{f\in\mathcal{F}} R(f) < \sqrt{\frac{\log|\mathcal{F}| + \frac{1}{2}\log n + 2}{n}} \ .$$

Proof The first statement follows since the choice of $\delta > 0$ is completely arbitrary. So ideally we should take δ so the bound is as small as possible. This cannot be done analytically. However, we can get a cruder bound by noticing the first term is larger than $1/\sqrt{n}$ when δ is not too large. Therefore, this motivates that choice $\delta = \sqrt{1/n}$, yielding

$$\begin{aligned}\mathbb{E}[R(\hat{f}_n)] - \min_{f\in\mathcal{F}} R(f) &< \sqrt{\frac{\log|\mathcal{F}| + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \\ &\leq \sqrt{\frac{\log|\mathcal{F}| + \frac{1}{2}\log n + 2}{n}} \ , (\text{since} \ \ \sqrt{x} + \sqrt{y} \leq \sqrt{2}\sqrt{x+y}, \ \forall \ x, y > 0) \ .\end{aligned}$$

□

This result shows that empirical risk minimization can be a good idea, and that the excess risk of the ERM prediction rule decays roughly like $\sqrt{\log|\mathcal{F}|/n}$. Note also that this provides a characterization of the expected estimation error of ERM, but does not say anything about the approximation error.

7.3 A PAC-bound for empirical risk minimization

We can also derive a PAC bound for empirical risk minimization. Recall that we showed that

$$\mathbb{P}\left(\forall f \in \mathcal{F} \quad R(f) < \hat{R}_n(f) + \varepsilon\right) \geq 1 - |\mathcal{F}|e^{-2n\varepsilon^2} \ .$$

Let $\tilde{f} = \arg\min_{f\in\mathcal{F}} R(f)$. An application of Hoeffding's inequality tells us also that

$$\mathbb{P}\left(\hat{R}_n(\tilde{f}) < R(\tilde{f}) + \varepsilon\right) \geq 1 - e^{-2n\varepsilon^2} \ .$$

Therefore, by the union bound, we conclude that

$$\mathbb{P}\left(\forall f \in \mathcal{F} \quad R(f) < \hat{R}_n(f) + \varepsilon \quad \text{and} \quad \hat{R}_n(\tilde{f}) < R(\tilde{f}) + \varepsilon\right) \geq 1 - (1 + |\mathcal{F}|)e^{-2n\varepsilon^2} \ .$$

Now let's recall the reasoning we took in Chapter 5, if the event inside the probability above is true then

$$\begin{aligned} R(\hat{f}_n) &< \hat{R}_n(\hat{f}_n) + \varepsilon \\ &\leq \hat{R}_n(f) + \varepsilon, \quad \text{for any } f \in \mathcal{F} \qquad (7.2) \\ &\leq \hat{R}_n(\tilde{f}) + \varepsilon \ , \\ &< R(\tilde{f}) + 2\varepsilon \ , \end{aligned}$$

were the step (7.2) follows from the definition of the empirical risk minimizer meaning that for any $f \in \mathcal{F}$ we have $\hat{R}_n(\hat{f}_n) < \hat{R}_n(f)$. We conclude that, with probability at least $1 - (1 + |\mathcal{F}|)e^{-2n\varepsilon^2}$

$$R(\hat{f}_n) \leq \min_{f\in\mathcal{F}} R(f) + 2\varepsilon \ ,$$

and so with high probability the true risk of the selected rule is only a little bit higher than the risk of the best possible rule in the class. This clearly indicates that ERM is quite a reasonable thing to do.

We can re-write the above statement as follows

Proposition 7.3.1 *Under the same assumptions of Proposition 7.2.1 we have that, for any* $\delta > 0$.

$$R(\hat{f}_n) - \inf_{f\in\mathcal{F}} R(f) < 2\sqrt{\frac{\log(1 + |\mathcal{F}|) + \log(1/\delta)}{2n}}$$

with probability at least $1 - \delta$.

Again, this result is a characterization of the estimation error of ERM, but does not say anything about the approximation error.

7.4 Application: the histogram classifier

Assume that label space is $\mathcal{Y} = \{0, 1\}$ and the loss function under consideration is the usual 0/1 loss. So we are in the setting of binary classification. The histogram classifier is based on

a partition of the feature space $\mathcal{X}$ into m smaller sets, followed by a majority vote decision in each of the sets. Denote each of the partition sets by Q_j $j \in \{1, \ldots, m\}$, so that

$$\bigcup_{j=1}^{m} Q_j = \mathcal{X} \quad \text{and} \quad \forall j \neq k \; Q_j \cap Q_k = \emptyset \; .$$

A particularly interesting case is if the feature space is the unit hypercube $\mathcal{X} = [0,1]^d$ (note that by scaling and shifting any set of d-dimensional bounded features we can satisfy this assumption). A popular choice of partition is obtained by splitting $[0,1]^d$ into m smaller hypercubes of equal size. This is illustrated for $d = 2$ in Figure 7.1.

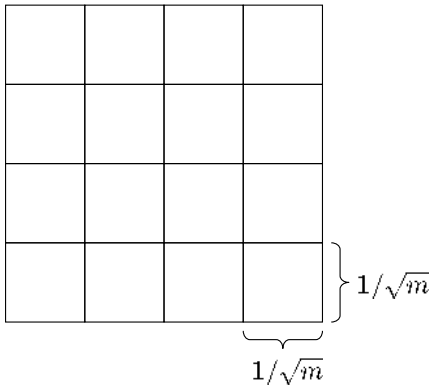


Figure 7.1: Example of hypercube $[0,1]^2$ in a m equally sized partition (here m must be a perfect square). The illustration pertains the case $m = 16$.

Now consider the class of classification rules that take either the value 0 or 1 in each of the partition sets. Formally define

$$\mathcal{F}_m = \left\{ f : \mathcal{X} \to \{0,1\} : f(x) = \sum_{j=1}^{m} c_j \mathbf{1}\{x \in Q_j\}, c_j \in \{0,1\} \right\} \; .$$

Note that this class has exactly 2^m elements. The histogram classifier is the element of this class obtained by doing a majority vote inside each partition cell. Namely

$$\hat{c}_j = \begin{cases} 1 & \text{if } \frac{\sum_{i:X_i \in Q_j} Y_i}{\sum_{i:X_i \in Q_j} 1} \geq 1/2 \\ 0 & \text{otherwise} \end{cases} \; .$$

Therefore the histogram classifier is defined as

$$\hat{f}_n(x) = \sum_{j=1}^{m} \hat{c}_j \mathbf{1}\{x \in Q_j\} \; . \tag{7.3}$$

It is easy to see that this is precisely the empirical risk minimizer (see exercise 7.5.1). Therefore we know that

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}_m} \hat{R}_n(f) \; ,$$

and we can apply the propositions above. In terms of excess risk this means that

$$\mathbb{E}[R(\hat{f}_n)] - \min_{f \in \mathcal{F}_m} R(f) \leq \sqrt{\frac{m \log 2 + 2 + \frac{1}{2} \log n}{n}} \, .$$

So, this gives us a bound on the expected estimation error of the histogram classifier, and tells us that we should not take m to be too large. Note that this result holds regardless of the distribution of the data! Essentially $m \equiv m_n$, the number of partition elements we have, should grow with the sample size, but not too fast. Our bound tells us that it must satisfy $\lim_{n \to \infty} \frac{m_n}{n} = 0$.

It is possible to show that the classification rules in $\mathcal{F}_m$ can approximate the Bayes' classifier for any distribution, provided m is large (to show this you need some knowledge of functional analysis - see Theorems 6.1 and 6.2 of Devroye et al. (1996)). This means the expected approximation error will converge to zero if m_n goes to infinity. Putting the two results together it is possible to show that

Theorem 7.4.1 (Consistency of Histogram Classifiers) *(Theorem 6.2 of Devroye et al. (1996)) If $m \to \infty$ and $\frac{n}{m} \to \infty$ as $n \to \infty$, then $\mathbb{E}[R(\hat{f}_n)] - R^* \to 0$ as $n \to \infty$, for any distribution $\mathbb{P}_{XY}$.*

Results with a similar flavor can be shown for nearest neighbor classifiers and related approaches and give rules-of-thumb on the size of the neighborhood you must choose as a function of n. Often in practice one takes $m_n = \sqrt{n}$. However, this is not necessarily a good idea and one needs good ways to choose m_n automatically. We will explore this issue in the coming chapters.

7.5 Exercises

Exercise 7.5.1 *Verify that the classification rule in (7.3) is obtained by minimizing the empirical risk over the class $\mathcal{F}_m$.*

Exercise 7.5.2 *Consider a classification problem with $\mathcal{X} = [0,1]^d$ and $\mathcal{Y} = \{0,1\}$. Let $\mathcal{F}$ be the collection of all histogram classifiers $f : \mathcal{X} \to \mathcal{Y}$ with m bins.*

a) Let $\hat{f}_n$ be one of the classifiers in $\mathcal{F}$ with the smallest training error. Assume $\min_{f \in \mathcal{F}} R(f) = 0$. Using results from a previous chapter determine, for a certain $\varepsilon, \delta > 0$, how many samples are needed to ensure that the probability $R(\hat{f}_n) \leq \varepsilon$ is larger than $1 - \delta$?

Let $m = 50$. Compute this minimal number of samples when you want to guarantee that the expected misclassification error of the chosen classifier is less than $\varepsilon = 0.01$, with probability greater than $1 - \delta = 0.95$.

b) Let $\hat{f}_n$ be one of the classifiers in $\mathcal{F}$ with the smallest training error. ***Do not assume*** *$\min_{f \in \mathcal{F}} R(f) = 0$. For a certain $\varepsilon, \delta > 0$ how many samples are needed to get ensure $R(\hat{f}_n) - \min_{f \in \mathcal{F}} R(f) \leq \varepsilon$ with probability at least $1 - \delta$. How does this compare with the answer to your previous question (plug in the same values of ε and δ in this setting and compare the minimum number of samples needed).*

Exercise 7.5.3 *Consider now a classification problem with $\mathcal{X} = [0,1]^2$ and $\mathcal{Y} = \{0,1\}$. Let $\{z_i\}_{i=1}^K \subseteq [0,1]^2$ be a collection of K points uniformly spaced around the perimeter of the unit square. Let $\mathcal{F}$ denote the set of linear classifiers obtained by using the line connecting connecting any two points $\{z_i\}$ as the decision boundary.*

a) Let $\hat{f}_n$ be one of the classifiers in $\mathcal{F}$ with the smallest training error. Assume $\min_{f\in\mathcal{F}} R(f) = 0$. For a certain $\varepsilon, \delta > 0$ how many samples are needed to ensure that the probability $R(\hat{f}_n) \leq \varepsilon$ is larger than $1 - \delta$?

Let $K = 40$. Compute this minimal number of samples when you want to guarantee that the expected misclassification error of the chosen classifier is less than $\varepsilon = 0.01$, with probability greater than $1 - \delta = 0.95$.

b) Let $\hat{f}_n$ be one of the classifiers in $\mathcal{F}$ with the smallest training error. ***Do not assume*** *$\min_{f\in\mathcal{F}} R(f) = 0$. For a certain $\varepsilon, \delta > 0$ how many samples are needed to get ensure $R(\hat{f}_n) - \min_{f\in\mathcal{F}} R(f) \leq \varepsilon$ with probability at least $1 - \delta$. How does this compare with the answer to your previous question (plug in the same values of ε and δ in this setting and compare the minimum number of samples needed).*

Chapter 8

Countably Infinite Model Spaces

In the last chapter, we characterized the performance of empirical risk minimization and obtained bounds of the form: for any $\delta > 0$, with probability at least $1 - \delta$,

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{\log|\mathcal{F}| + \log\left(\frac{1}{\delta}\right)}{2n}} \ , \quad \forall f \in \mathcal{F}$$

which led to the following upper bounds on the expected estimation error

$$\mathbb{E}[R(\hat{f}_n)] - \min_{f \in F} R(f) \leq \sqrt{\frac{\log|\mathcal{F}| + \log(n) + 2}{n}} \ .$$

The key assumptions made in deriving the error bounds were:

(i) - bounded loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$

(ii) - finite collection of candidate functions

The bounds are valid for every $\mathbb{P}_{XY}$, and therefore are called *distribution-free*. A major drawback of these bounds is that they hold only for finite classes of models, and that we cannot use these results if the class of models is infinite. In this chapter we will see a way around this issue. Furthermore, the ideas we develop in this chapter have much more profound implications, and will give us ways of selecting the complexity of a model based on the data (for instance, use the data to decide how many bins should our histogram classifier have).

To start let us suppose that $\mathcal{F}$ is a countable, possibly infinite, collection of candidate functions (as a reminder remember that $\mathbb{N}$ and $\mathbb{Q}$ are countably infinity sets). Assign a positive number $c(f)$ to each $f \in \mathcal{F}$, such that

$$\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1 \ .$$

The number $c(f)$ can be interpreted as

(i) - measure of the complexity of f

(ii) - negative log of prior probability of f

(iii) - length of a codeword describing f

The appeal of each interpretation depends on the community you ask. Interpretation (ii) is perhaps the one encountered more often: suppose you have a *prior* probability distribution $p(\cdot)$ over $\mathcal{F}$, so that $p(f) \geq 0$ for all $f \in \mathcal{F}$ and

$$\sum_{f \in \mathcal{F}} p(f) = 1 \ .$$

Then taking $c(f) = -\log p(f)$ guarantees that $\sum_{f \in \mathcal{F}} e^{-c(f)} = 1$.

Recall the steps of the derivation in the previous chapter. Hoeffding's inequality tells us that for each f and every $\varepsilon > 0$

$$\mathbb{P}\left(R(f) - \hat{R}_n(f) \geq \varepsilon \right) \leq e^{-2n\varepsilon^2} \ ,$$

or, in other words, for every $\delta > 0$

$$\mathbb{P}\left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2n}} \right) \leq \delta \ .$$

This is the expression we used for each $f \in \mathcal{F}$. So the "confidence" associated with each model f is exactly the same. Let's instead associate different levels of confidence to each model.

Let $\delta > 0$ and use the values $c(f)$ to define $\delta(f) = \delta e^{-c(f)}$. For each $f \in \mathcal{F}$ we have

$$\mathbb{P}\left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log\left(\frac{1}{\delta(f)}\right)}{2n}} \right) \leq \delta(f) \ .$$

Now let us apply the union bound, as before

$$\begin{aligned}
&\mathbb{P}\left(\exists f \in \mathcal{F} : \ R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log(1/\delta(f))}{2n}} \right) \\
&= \mathbb{P}\left(\bigcup_{f \in \mathcal{F}} \left\{ R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log(1/\delta(f))}{2n}} \right\} \right) \\
&\leq \sum_{f \in \mathcal{F}} \mathbb{P}\left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log\left(\frac{1}{\delta(f)}\right)}{2n}} \right) \\
&\leq \sum_{f \in \mathcal{F}} \delta(f) = \sum_{f \in \mathcal{F}} \delta e^{-c(f)} = \delta \ .
\end{aligned}$$

We can state the following result.

Proposition 8.0.1 *Let $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ be a bounded loss function, and suppose $\mathcal{F}$ is countable (but possibly infinite). Suppose furthermore we have a map $c : \mathcal{F} \to \mathbb{R}$ satisfying*

$$\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1 \ .$$

Let $\delta > 0$. Then

$$\forall f \in \mathcal{F} \quad R(f) \leq \hat{R}_n(f) + \sqrt{\frac{\log \frac{1}{\delta(f)}}{2n}} = \hat{R}_n(f) + \sqrt{\frac{c(f) + \log \frac{1}{\delta}}{2n}} \ ,$$

with probability at least $1 - \delta$.

The above bound also motivates us to choose a model from $\mathcal{F}$ in the following fashion

$$\hat{f}_n^{\delta} = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + \sqrt{\frac{c(f) + \log \frac{1}{\delta}}{2n}} \right\} \ . \tag{8.1}$$

In other words, one is minimizing a penalized version of the empirical risk, where complex models, those with $c(f)$ large, are penalized more heavily.

8.0.1 A Special Case - $\mathcal{F}$ finite

Suppose $\mathcal{F}$ is finite and let $c(f) = \log |\mathcal{F}| \quad \forall f \in \mathcal{F}$. Then

$$\sum_{f \in \mathcal{F}} e^{-c(f)} = \sum_{f \in \mathcal{F}} e^{-\log |\mathcal{F}|} = \sum_{f \in \mathcal{F}} \frac{1}{|\mathcal{F}|} = 1 \ ,$$

and $\delta(f) = \frac{\delta}{|\mathcal{F}|}$ which implies that for any $\delta > 0$ with probability at least $1 - \delta$, we have

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{\log |\mathcal{F}| + \log \left(\frac{1}{\delta}\right)}{2n}} \ , \ \forall f \in \mathcal{F} \ .$$

Note that this is precisely the bound we derived in the last chapter. In this case, the penalized empirical risk criterion (8.1) boils down to simple empirical risk minimization, since the penalty term is not a function of f.

8.1 Choosing the values $c(f)$

The generalized bounds allow us to handle countably infinite collections of candidate functions, but we need to choose a value $c(f)$ for all $f \in \mathcal{F}$ such that

$$\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1 \ .$$

If we have a proper prior probability distribution over $\mathcal{F}$ we already seen a way of assigning the values $c(f)$ to each model f. However, it may be difficult to design a probability distribution over an infinite class of candidates. In some cases we might be possible to do it "by hand", but in complicated scenarios it is not so clear. Another approach, that based on coding arguments, is much more practical for our purposes.

Suppose we want to encode the elements of $\mathcal{F}$ using a binary alphabet, and have assigned a uniquely decodable binary codeword to each $f \in \mathcal{F}$. Let $c(f)$ denote the codelength for f. That is, the codeword for f is $c(f)$ bits long. We want to use these codes to encode a sequence of symbols from $\mathcal{F}$, so we must concatenate the codewords (without any punctuation). A very useful class of uniquely decodable codes are called **prefix codes**, also known as instantaneous codes, as we will see next.

Definition 8.1.1 *A code is called a* prefix *or* instantaneous *code if no codeword is a prefix of any other codeword.*

Example 8.1.1 *(From Cover and Thomas (2006))*
Consider an alphabet of symbols, say A, B, C, and D and the codebooks in Figure 8.1. In the

Symbol	Singular Codebook	Nonsingular But Not Uniquely Decodable	Uniquely Decodable But Not a Prefix Code	Prefix Code
A	0	0	10	0
B	0	010	00	10
C	0	01	11	110
D	0	10	110	1110

Figure 8.1: Four possible codes for an alphabet of four symbols.

singular codebook we assign the same codeword to each symbol - a system that is obviously flawed! In the second case, the codes are not singular but the codeword 010 could represent B or CA or AD. Hence, once you concatenate codewords it becomes impossible to parse the individual symbols. This means this is not a uniquely decodable codebook.

The third and fourth cases are both examples of uniquely decodable codebooks, but the third one is not instantaneous. Consider for instance the sequence 111100... After the fourth bit all you know is that this might represent CC... or CD... After the fifth bit you know it might represent the sequence CD... or CCB... It is only after the sixth bit that you will know the first two symbols are CC. But these symbols are encoded in the first 4 bits only. So you had to "look ahead" to be able to decode the sequence. The fourth code has the added feature that no codeword is a prefix of another. Prefix codes can be decoded from left to right since each codeword is "self-punctuating" - in this case with a zero to indicate the end of each word. This is why such codes are also called instantaneous. Once you reach the end of a codeword you know it immediately.

Designing prefix codes is generally not too hard and moreover, the corresponding codeword lengths satisfy the so-called Kraft inequality, which is essentially the condition $\sum_{f\in\mathcal{F}} e^{-c(f)} \leq 1$.

8.2 The Kraft Inequality

We begin with the following important result pertaining prefix codes.

Theorem 8.2.1 *For any binary prefix code, the codeword lengths c_1, c_2, ... satisfy*

$$\sum_{i=1}^{\infty} 2^{-c_i} \leq 1 \ .$$

Conversely, given any c_1, c_2, ... satisfying the inequality above we can construct a prefix code with these codeword lengths.

The proof of this result is presented later on, but first let's see why this is useful in our learning problem.

Assume that we have assigned a binary prefix codeword to each $f \in \mathcal{F}$, and let $c(f)$ denote the bit-length of the codeword for f. Then

$$\sum_{i=1}^{n} e^{-c(f)\log 2} = \sum_{i=1}^{n} 2^{-c(f)} \leq 1 \ ,$$

by the Kraft inequality. This is precisely the condition we needed to conclude that

$$\forall f \in \mathcal{F} \quad R(f) \leq \hat{R}_n(f) + \sqrt{\frac{c(f)\log 2 + \log \frac{1}{\delta}}{2n}} \ ,$$

with probability at least $1 - \delta$.

8.2.1 Application - Structural Risk Minimization

Let $\mathcal{F}_1, \mathcal{F}_2, \ldots$, be a sequence of finite classes of candidate models with $|\mathcal{F}_1| \leq |\mathcal{F}_2| \leq \ldots$. Let $\mathcal{F} = \bigcup_{i=1}^{\infty} \mathcal{F}_i$. We can design a prefix code for all the elements of $\mathcal{F}$ in a simple way: Use the codes $0, 10, 110, 1110, \ldots$ to encode the subscript i in $\mathcal{F}_i$. For each class $\mathcal{F}_i$, construct a set of binary codewords of length $\log_2 |\mathcal{F}|$ to uniquely encode each function in $\mathcal{F}_i$. Now we just need to concatenate the two codes. For any given function $f \in \mathcal{F}$ use the first code to encode the smallest index i such that $f \in \mathcal{F}_i$, followed by a codeword of length $\log_2 |\mathcal{F}|$ identifying $f \in \mathcal{F}_i$. You can easily show this is a prefix code (this way of encoding of the specific model class is what is known as the unary code).

Example 8.2.1 *The unary code is obviously not the only way to encode the order of the model class $\mathcal{F}_k$. Consider the following telescoping sum*

$$\sum_{k=1}^{\infty} \frac{1}{k} - \frac{1}{k+1} = 1 \ .$$

This means that $\sum_{k=1}^{\infty} \frac{1}{k^2+k} = 1$, therefore $\sum_{k=1}^{\infty} 2^{-\log_2(k^2+k)} = 1$. So we can encode each element f of $\mathcal{F}$ using

$$c(f) = \log_2(k_f^2 + k_f) + \log_2 |\mathcal{F}_{k_f}|$$

bits instead. If you believe the "best" model is in $\mathcal{F}_k$ for $k \geq 5$ then this code is a better choice, since

$$c(f) = \log_2(k^2 + k) < k \ , \ \text{for } k \geq 5 \ .$$

Example 8.2.2 (Histogram Classifiers) *Recall the setting of the histogram classifier, as described in Section 7.4, where $\mathcal{Y} = \{0, 1\}$. Let $\mathcal{F}_k$, $k = 1, 2, \ldots$, denote the collection of histogram classification rules with k bins (these must be constructed before looking at the data). Note that $|\mathcal{F}_k| = 2^k$. We can use the approach just described to encode* any *histogram classifier f in the class $\mathcal{F} = \bigcup_{i=1}^{\infty} \mathcal{F}_i$. Use k bits to indicate the smallest k such that $f \in \mathcal{F}_k$, and then use $\log_2 |\mathcal{F}_k| = k$ bits to indicate which of the 2^k possible histogram rules it is. Thus we have constructed a prefix code, and for any $f \in \mathcal{F}_k$ for some $k \geq 1$ the codeword assigned to f has $c(f) = k + k = 2k$ bits. It follows that for any $\delta > 0$ with probability at least $1 - \delta$ we have*

$$\forall f \in \bigcup_{k \geq 1} \mathcal{F}_k \qquad R(f) \leq \hat{R}_n(f) + \sqrt{\frac{2k_f \log 2 + \log\left(\frac{1}{\delta}\right)}{2n}} \ ,$$

where k_f is the (smallest) the number of bins in histogram corresponding to f. Contrast this with the bound we had for the class $\mathcal{F}_k$ alone: with probability $\geq 1-\delta$, $\forall f \in \mathcal{F}_k$

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{k \log 2 + \log\left(\frac{1}{\delta}\right)}{2n}} \ .$$

Notice the bound for all histograms rules is almost as good as the bound for on only the k-bin rules. It is worse by a factor $\sqrt{2}$ only. On the other hand, the new bound is a big improvement, since it also gives us a guide for selecting the number of bins based on the data, suggesting the rule

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) + \sqrt{\frac{2k_f \log 2 + \log\left(\frac{1}{n}\right)}{2n}} \ .$$

Proof of the Kraft Inequality (Theorem 8.2.1) We will prove that for any binary prefix code, the codeword lengths $c_1, c_2, \ldots$, satisfy $\sum_{k \geq 1} 2^{-c_k} \leq 1$. The converse is also easy to prove, but it not central to our purposes here (for a proof, see Cover and Thomas (2006)). We start by proving the result when the number of codewords is finite. Consider a binary tree like the one shown in Figure 8.2.1, that we call the code-tree

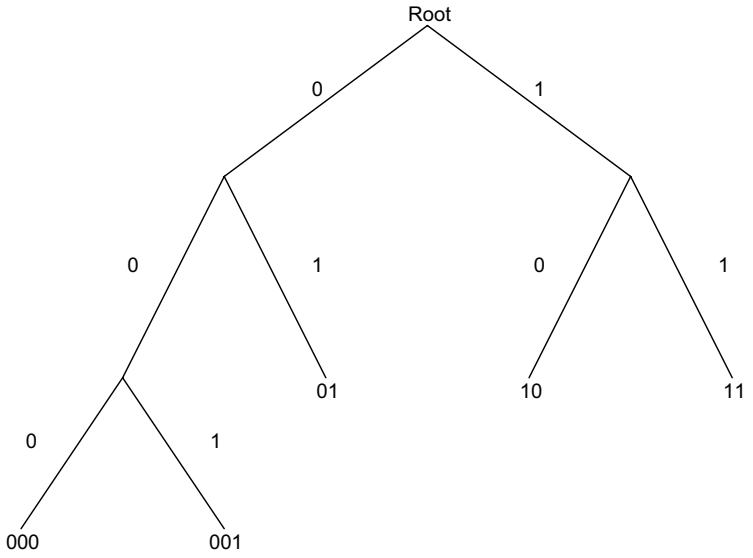


Figure 8.2: A binary tree.

The sequence of bit values leading from the root to a leaf of the tree represents a unique codeword. The prefix condition implies that no codeword is a descendant of any other codeword in the tree. Therefore each leaf of the tree represents a codeword in our code. Let $c_{\max}$ be the length of the longest codeword (also the number of branches to the deepest leaf) in the tree (since the number of codewords is assumed finite $c_{\max} < \infty$).

The tree representing the set of codewords is obviously a subtree of the tree will all the leafs at level $c_{\max}$, which we call the full tree. Consider a node in the tree at level c_i. This node in the tree can have at most $2^{c_{\max}-c_i}$ descendants at level $c_{\max}$. Furthermore, for each leaf node, the

sets of descendants at level $c_{\max}$ in the full tree are disjoint (since no codeword can be a prefix of another). Therefore, since the total number of possible leafs at level $c_{\max}$ is $2^{c_{\max}}$, we have

$$\sum_{i \in \text{leafs}} 2^{c_{\max} - c_i} \leq 2^{c_{\max}} \;\Rightarrow\; \sum_{i \in \text{leafs}} 2^{-c_i} \leq 1$$

which proves the case when the number of codewords is finite.

Suppose now that we have a countably infinite number of codewords. Let $b_1, b_2, \ldots, b_{c_i}$ be the bits of the i^{th} codeword and let

$$r_i = \sum_{j=i}^{c_i} b_j 2^{-j}$$

be the real number corresponding to the binary expansion of the codeword (in binary $r_i = 0.b_1\ b_2\ b_3 \ldots$). We can associate the interval $[r_i, r_i + 2^{-c_i})$ with the i^{th} codeword. This is the set of all real numbers whose binary expansion begins with $b_1, b_2, \ldots, b_{c_i}$. Since this is a subinterval of $[0, 1]$, and all such subintervals corresponding to prefix codewords are disjoint, the sum of their lengths must be less than or equal to 1. This generalizes the proof to the case where the number of codewords is infinite. □

8.3 Complexity Regularization Bounds

Earlier on in this chapter we proved Proposition 8.0.1, which tells us that with probability at least $1 - \delta$

$$\forall f \in \mathcal{F} \quad R(f) \leq \hat{R}_n(f) + \sqrt{\frac{c(f) + \log(1/\delta)}{2n}} \; .$$

This gives us an idea on how to pick a good model from $\mathcal{F}$, using a penalized version of the empirical risk. If we look at the result above, it is clear we want to choose f to minimize the l.h.s.. However, this is not something we can compute. On the other hand, the r.h.s. is an upper bound on that quantity, and something we can actually compute. This motivates the following choice

$$\hat{f}_n^{\delta} = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f, n, \delta) \right\} \; ,$$

where

$$C(f, n, \delta) \equiv \sqrt{\frac{c(f) + \log(1/\delta)}{2n}} \; .$$

Is this a good idea? The answer is yes, as the following result indicates.

Theorem 8.3.1 (Complexity Regularized Model Selection) *Let $\mathcal{F}$ be a countable collection of models, and assign a real number $c(f)$ to each $f \in \mathcal{F}$ such that*

$$\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1 \; .$$

Define the minimum complexity regularized risk model

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + \sqrt{\frac{c(f) + \frac{1}{2}\log n}{2n}} \right\} \; .$$

Then,

$$\mathbb{E}[R(\hat{f}_n)] \;\leq\; \inf_{f\in\mathcal{F}} \left\{ R(f) + \sqrt{\frac{c(f)+\frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \right\} \;.$$

The estimator in this result is obtained by taking $\delta = 1/\sqrt{n}$, but other choices for δ are possible.

Proof Since the statement in Proposition 8.0.1 holds for all $f \in \mathcal{F}$, it holds in particular for $\hat{f}_n^\delta$, therefore

$$R(\hat{f}_n^\delta) \leq \hat{R}_n(\hat{f}_n^\delta) + C(\widehat{f}_n^\delta, n, \delta) \;,$$

with probability at least $1-\delta$. By the definition of $\hat{f}_n^\delta$ we conclude that

$$\forall f \in \mathcal{F} \quad R(\hat{f}_n^\delta) \leq \hat{R}_n(f) + C(f, n, \delta) \;,$$

with probability at least $1-\delta$.

For any fixed model $f \in \mathcal{F}$ we know that $E[\hat{R}_n(f)] = R(f)$. Therefore

$$\mathbb{E}[R(\hat{f}_n^\delta)] - R(f) = \mathbb{E}[R(\hat{f}_n^\delta) - \hat{R}_n(f)] \;.$$

Let Ω be the event defined as

$$\forall f \in \mathcal{F} \quad R(\hat{f}_n^\delta) \leq \hat{R}_n(f) + C(f, n, \delta) \;.$$

From the proposition we know that $\mathbb{P}(\Omega) \geq 1-\delta$. Thus,

$$\begin{aligned}
\mathbb{E}[R(\hat{f}_n^\delta) - \hat{R}_n(f)] &= \mathbb{E}[R(\hat{f}_n^\delta) - \hat{R}_n(f)|\Omega]\mathbb{P}(\Omega) + \mathbb{E}[R(\hat{f}_n^\delta) - \hat{R}_n(f)|\bar{\Omega}](1-\mathbb{P}(\Omega)) \\
&\leq C(f,n,\delta) + \delta \quad (\text{since } 0 \leq R(f), \widehat{R}_n(f) \leq 1,\ \mathbb{P}(\Omega) \leq 1, \text{ and } 1-\mathbb{P}(\Omega) \leq \delta)) \\
&= \sqrt{\frac{c(f)+\log(1/\delta)}{2n}} + \delta \\
&= \sqrt{\frac{c(f)\log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} (\text{by setting } \delta = \frac{1}{\sqrt{n}}) \;,
\end{aligned}$$

concluding the proof. □

This result shows that

$$\hat{R}_n(f) + \sqrt{\frac{c(f)\log 2 + \frac{1}{2}\log n}{2n}}$$

is a reasonable surrogate for

$$R(f) + \sqrt{\frac{c(f)\log 2 + \frac{1}{2}\log n}{2n}} \;,$$

So if minimizing the latter can be considered advantageous we are in good shape. Remarkably, if we are careful about the choice of $c(f)$ this can be extremely advantageous. In the coming chapter we apply these results in a concrete setting, to get a hold of what is going on.

Chapter 9

The Histogram Classifier revisited

This chapter is devoted exclusively to the histogram classifier. We have already encountered it Section 7.4. We will see how the results of the previous chapter allow us to automatically choose the number of bins in the histogram classifier. Furthermore, we will also consider the use of leave-one-out cross validation in this setting, as an alternative way to choose the number of bins. We will see that both approaches are quite sensible, although we only have strong guarantees for the penalized empirical risk.

Recall the general setting of the histogram classifier. The label space is binary $\mathcal{Y} = \{0, 1\}$ and the loss function under consideration is the usual 0/1 loss. The histogram classifier is based on a partition of the feature space $\mathcal{X}$ into m smaller sets, followed by a majority vote decision in each of the sets. Denote each of the partition sets by $Q_{j,m}$ $j \in \{1, \ldots, m\}$, so that

$$\bigcup_{j=1}^{m} Q_{j,m} = \mathcal{X} \quad \text{and} \quad \forall i \neq j \; Q_{i,m} \cap Q_{j,m} = \emptyset \; .$$

Now consider the class of classification rules that take either the value 0 or 1 in each of the partition sets. Formally define

$$\mathcal{F}_m = \left\{ f : \mathcal{X} \to \{0, 1\} : f(x) = \sum_{j=1}^{m} c_j \mathbf{1}\{x \in Q_{j,m}\}, c_j \in \{0, 1\} \right\} \; .$$

Say we have now, for each value m a collection of histogram classifiers and define the union of these classes as $\mathcal{F} = \cup_{m=1}^{\infty}$. Note that the entire construction is done without looking at data. Now we would like to pick a model from this class based on data.

9.1 Complexity regularization

To apply the results and ideas developed in the previous chapter we need construct at map $c : \mathcal{F} \to \mathbb{R}$ satisfying

$$\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1 \; . \tag{9.1}$$

There are several ways to do this. A coding argument, as described in the previous chapter, is one possibility. Here we will explicitly construct the map and check it is sensible. To be formal,

for any $f \in \mathcal{F}$ define

$$m_f = \min_m \{m \in \mathbb{N} : f \in \mathcal{F}_m\} \ .$$

In words, this is the smallest value of m for which $f \in \mathcal{F}_m$. Now define

$$c(f) = \log(m_f) + \log(m_f + 1) + m_f \log(2) \ .$$

Let us first make sure this (9.1) is satisfied.

$$\begin{aligned}
\sum_{f \in \mathcal{F}} e^{-c(f)} &= \sum_{k=1}^{\infty} \sum_{f \in \mathcal{F}: m_f = k} e^{-c(f)} \\
&= \sum_{k=1}^{\infty} \sum_{f \in \mathcal{F}: m_f = k} \frac{1}{k(k+1)} \frac{1}{2^k} \\
&\leq \sum_{k=1}^{\infty} \frac{1}{k(k+1)} \sum_{f \in \mathcal{F}_k} \frac{1}{2^k} \\
&= \sum_{k=1}^{\infty} \frac{1}{k(k+1)} = 1 \ ,
\end{aligned}$$

where we used the fact that $|\mathcal{F}_k| = 2^k$, and the inequality comes from the fact that $\{f \in \mathcal{F} : m_f = k\} \subseteq \mathcal{F}_k$. For instance, the rule $f \equiv 0$ belongs to $\mathcal{F}_k$ for all k.

The theory we developed suggests the learning rule

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) + \sqrt{\frac{\log(m_f) + \log(m_f + 1) + m_f \log(2) + \log(1/\delta)}{2n}} \ ,$$

where $\delta > 0$ is a parameter to be chosen by us. We can write the above in a slightly different way. For each m define

$$\hat{f}_{n,m} = \arg\min_{f \in \mathcal{F}_m} \hat{R}_n(f) \ .$$

This is the empirical risk minimizer for each of the subclasses. Now define

$$\hat{m}_n = \arg\min_{m \in \mathbb{N}} \left\{ \hat{R}_n(\hat{f}_{n,m}) + \sqrt{\frac{\log(m) + \log(m+1) + m \log(2) + \log(1/\delta)}{2n}} \right\} \ ,$$

and finally define

$$\hat{f}_n = \hat{f}_{n,\hat{m}_n} \ .$$

This is entirely equivalent to the previous definition. The results in the previous chapter tell us that

$$\mathbb{E}[R(\hat{f}_n)] \leq \inf_{m \in \mathbb{N}} \left\{ \min_{f \in \mathcal{F}_m} R(f) + \sqrt{\frac{\log(m_f) + \log(m_f + 1) + m_f \log(2) + \log(1/\delta)}{2n}} + \frac{1}{\delta} \right\} \ .$$

To get a bit more insight on what this means define

$$\tilde{f}_m = \min_{f \in \mathcal{F}_m} R(f) \ ,$$

the best possible histogram classifier with m bins. The above result implies

$$\mathbb{E}[R(\hat{f}_n)] \leq \inf_{m\in\mathbb{N}} \left\{ R(\tilde{f}_m) + \sqrt{\frac{\log(m) + \log(m+1) + m\log(2) + \log(1/\delta)}{2n}} + \frac{1}{\delta} \right\} .$$

Compare this the to result when we just take a given number of bins m

$$\mathbb{E}[R(\hat{f}_{n,m}] \leq R(\tilde{f}_m) + \sqrt{\frac{m\log(2) + \log(1/\delta)}{2n}} + \frac{1}{\delta} .$$

These two bounds are on the estimation error, but the first uses the "optimal" value for the number of bins. Note that $(\log(m) + \log(m+1) + m\log 2)/(m\log 2)$ is always smaller than 2.3, and for m large this ratio is approximately $1 + \frac{2}{\log 2}\frac{\log m}{m}$. Therefore the two bounds approach each other as m grows. This means that, although $\hat{f}_n$ is choosing the number of bins automatically, it has essentially the same (worst-case) performance guarantee as $\hat{f}_{n,m_{\tilde{f}}}$, that is, the histogram classifier obtained with a clairvoyant choice of the number of bins (that we cannot do in practice).

As discussed in Chapter 3 the empirical error typically decreases as the model complexity increases. In our context, as the number of bins m increases. The empirical error is therefore, to some extent, a good surrogate for the approximation error (this is not a formal statement). So one can view penalization term in penalized empirical risk minimization as a way to account for the "missing" estimation error, so that the penalized empirical risk behaves approximately like the true risk. This is of course an informal point of view.

Example 9.1.1 *Let us consider a concrete instantiation of the above in d dimensions. Let $\mathcal{X} = [0,1]^d$ be the input space and $\mathcal{Y} = \{0,1\}$ be the output space. Let $\mathcal{F}_k$, $k = 1, 2, \ldots$ denote the collection of histogram classification rules with k^d hypercubical equal volume bins, and let $\mathcal{F} = \bigcup_{k=1}^{\infty} \mathcal{F}_k$ be the class containing all such possible histograms. Note that, unlike in the description above, the histograms do not have an arbitrary number of bins but rather k^d bins. Each class $\mathcal{F}_k$ has exactly 2^{k^d} elements, so, using the same approach as above we construct the map*

$$c(f) = \log(k_f) + \log(k_f + 1) + k_f^d \log 2 ,$$

where

$$k_f = \min_k \{k \in \mathbb{N} : f \in \mathcal{F}_k\} .$$

It is easy to see this satisfies (9.1). For each k define

$$\hat{f}_{n,k} = \arg\min_{f\in\mathcal{F}_k} \hat{R}_n(f) .$$

This is the ERM for each of the subclasses. Now define

$$\hat{k}_n = \arg\min_{k\in\mathbb{N}} \left\{ \hat{R}_n(\hat{f}_{n,k}) + \sqrt{\frac{\log(k) + \log(k+1) + k^d\log 2 + \frac{1}{2}\log n}{2n}} \right\} ,$$

and finally define

$$\hat{f}_n = \hat{f}_{n,\hat{k}_n} .$$

In the above we took the choice $\delta = 1/\sqrt{n}$ for concreteness. Our result tells us that

$$\mathbb{E}[R(\hat{f}_n)] \leq \inf_{k\in\mathbb{N}} \left\{ \min_{f\in\mathcal{F}_k} R(f) + \sqrt{\frac{\log(k) + \log(k+1) + k^d \log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \right\} .$$

Note that, for each $k \in \mathbb{N}$,

$$\mathbb{E}[R(\hat{f}_{n,k})] \leq \min_{f\in\mathcal{F}_k} R(f) + \sqrt{\frac{k^d \log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} .$$

So this means we can, at a very small price, choose the number of bins in the histogram classifier that minimizes the r.h.s. of this bound! This means that we can automatically choose the number of bins in the histogram rule in such a way that we do almost as well as if we actually knew the right number!

In Figure 9.1(b) we have an example of a dataset of size $n = 1000$ generated as i.i.d. samples from some distribution. In Figure 9.1(a) we depict the empirical risk, the true risk and the penalized empirical risk. You can see the empirical risk behaves qualitatively like the approximation error, and that the penalized empirical risk has a behavior qualitatively similar to that of the true risk. However, we see that the penalization is much more severe than deemed necessary, since the penalized empirical risk grows faster than the true risk as m increases. This means that we are a bit over-conservative. This is the price to pay for using a very general, distribution-free bound. In most situations we will overestimate the estimation error. A way around it is to construct better and data dependent bounds for the estimation error. Alternatively we can use a cross-validation approach, as explained below. Refering back to Figure 9.1(b) you can see the cross-validation risk (which is computed only based on data) mimics closely the behavior of the true risk. In Figures 9.1(c,d) you can see the resulting histogram rules based on the penalized approach and cross-validation, and that that penalized approach is underfiting the data too severely.

9.2 Leave-one-out Cross Validation

As seen in Section 7.4, $\hat{f}_{n,m}$ can be expressed as

$$\hat{f}_{n,m}(x) = \sum_{j=1}^{m} \hat{c}_{j,m} \mathbf{1}\{x \in Q_{j,m}\} ,$$

where

$$\hat{c}_{j,m} = \begin{cases} 1 & \text{if } \frac{\sum_{i:X_i\in Q_{j,m}} Y_i}{\sum_{i:X_i\in Q_{j,m}} 1} \geq 1/2 \\ 0 & \text{otherwise} \end{cases} ,$$

where we convention $0/0 = 0$. For what follows it is convenient to introduce some notation. Let

$$A_{j,m} = \sum_{i:X_i\in Q_{j,m}} Y_i \qquad \text{and} \qquad B_{j,m} = \sum_{i:X_i\in Q_{j,m}} 1 .$$

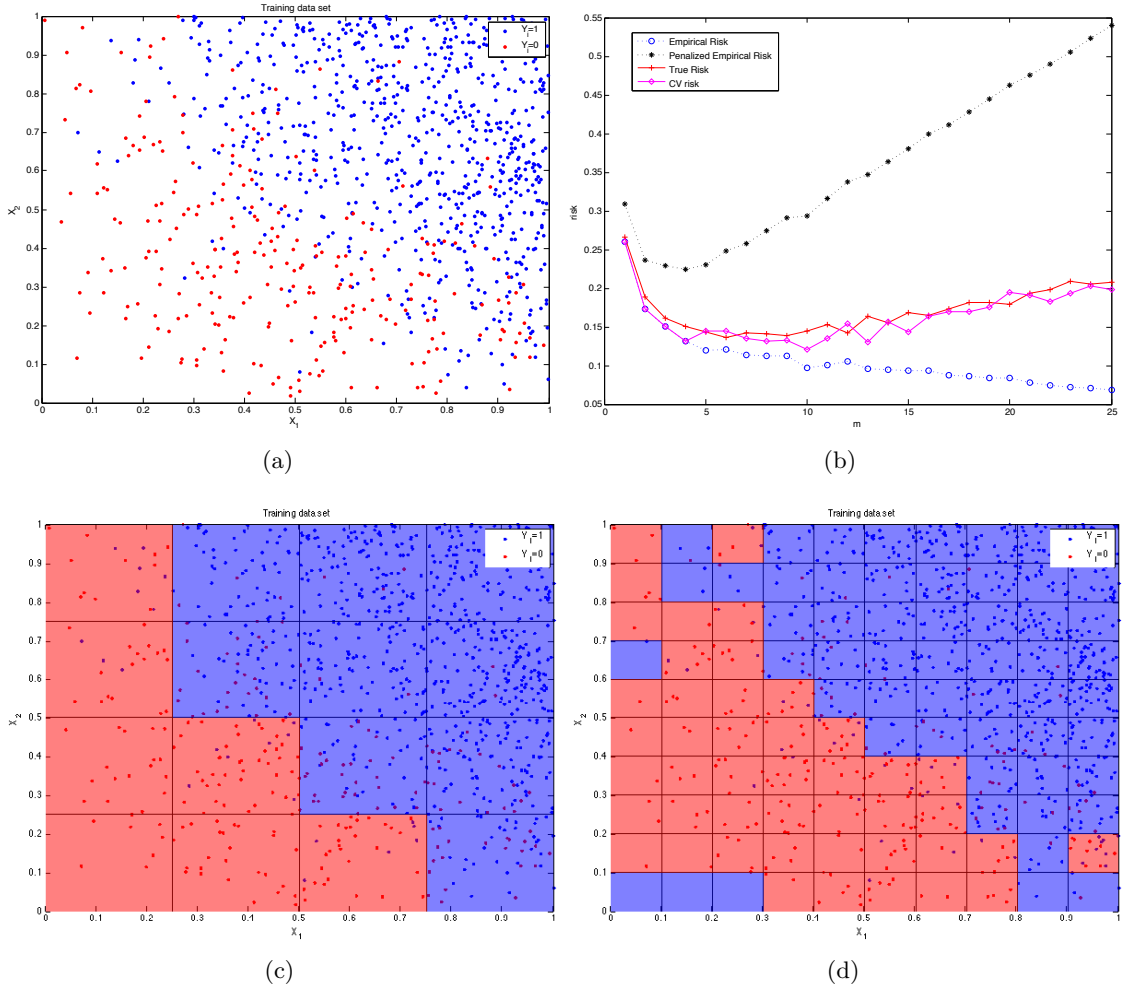


Figure 9.1: Illustration of the behavior of the histogram classifier. In panel (a) we see an example of a dataset generated as 1000 i.i.d. samples from a certain distribution. In panel (b) the various risks are plotted as a function of the number of bins. Note that the true risk cannot be computed from data - it was computed relying on the knowledge of the distribution of the data. Also, the risks are only defined for integer values of m (the lines are in the plot to aid the visualization only). In panel (c) is the resulting histogram classifier chosen by empirical risk minimization. In panel (d) is the histogram classifier as chosen using the leave-one-out cross-validation approach.

Therefore $\hat{c}_{j,m} = \mathbf{1}\{2A_{j,m} - B_{j,m} \geq 0\}$. It is easy to show that the empirical risk of $\hat{f}_{n,m}$ is given by

$$\hat{R}_n(\hat{f}_{n,m}) = \frac{1}{n}\sum_{j=1}^{m} \min\left(A_{j,m}, B_{j,m} - A_{j,m}\right) \ . \tag{9.2}$$

The idea of leave-one-our cross validation, presented in generality in Chapter 3, is to consider n splits of the data into a training set (of size $n-1$) and a validation set with only one data point. For each of the splits we can compute a classifier and unbiasedly estimate the corresponding risk. These risk estimates are lousy, but we have n of them. So the next step is to aggregate them to ensure the variance of this estimate is not too bad. Let's be more concrete. Let $i \in \{1, \ldots, n\}$ and define

$$\hat{f}_{n,m}^{(-i)}(x) = \operatorname*{arg\,min}_{f \in \mathcal{F}_m} \frac{1}{n-1} \sum_{k:k\neq i} \mathbf{1}\{f(X_k) \neq Y_k\} \ .$$

In words, this is the empirical risk minimizer when we leave the data point (X_i, Y_i) out. It is easy to check that

$$\hat{f}_{n,m}^{(-i)}(x) = \sum_{j=1}^{m} \hat{c}_{j,m}^{(-i)} \mathbf{1}\{x \in Q_{j,m}\} \ ,$$

where

$$\hat{c}_{j,m}^{(-i)} = \begin{cases} \mathbf{1}\left\{\frac{A_{j,m}}{B_{j,m}} \geq 1/2\right\} & \text{if } X_i \notin Q_{j,m} \\ \mathbf{1}\left\{\frac{A_{j,m}-Y_i}{B_{j,m}-1} \geq 1/2\right\} & \text{if } X_i \in Q_{j,m} \end{cases} \ ,$$

So, for the most part, this is almost the same as $\hat{f}_{n,m}$. Now define the cross-validation risk as

$$\mathrm{CV}_{n,m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\hat{f}_{n,m}^{(-i)}(X_i) \neq Y_i\} \ .$$

This is the average of the errors you make in the point you did not use for training. With a bit of work (see exercise 9.3.2) one can obtain a relatively simple expression for the cross-validation risk

$$\mathrm{CV}_{n,m} = \frac{1}{n}\sum_{j=1}^{m} A_{j,m}\mathbf{1}\{2A_{j,m} - B_{j,m} < 1\} + (B_{j,m} - A)\mathbf{1}\{2A_{j,m} - B_{j,m} \geq -1\} \ . \tag{9.3}$$

Why is the cross-validation risk interesting? Note that

$$\mathbb{E}[\mathrm{CV}_{n,m}] = \mathbb{E}[\mathbf{1}\{\hat{f}_{n,m}^{(-1)}(X_1) \neq Y_1\}] = R(\hat{f}_{n-1,m}) \ .$$

In other words, the expected value of the cross-validation risk is the risk of the histogram classifier with a training set of size $n-1$. If n is not too small then $R(\hat{f}_{n,m}) \approx R(\hat{f}_{n-1,m})$ therefore we have almost an unbiased estimator of the risk of the histogram classifier (unlike the empirical risk, which has potentially a strong negative bias). Therefore an idea is to use this to choose the number of bins as follows.

$$\hat{m}_{\mathrm{CV}} = \operatorname*{arg\,min}_{m} \mathrm{CV}_{n,m} \qquad \text{and} \qquad \hat{f}_n^{(CV)} = \hat{f}_{n,\hat{m}_{\mathrm{CV}}} \ .$$

It is not easy to get non-asymptotic guarantees for this methodology, but it often works very well in practice. This is illustrated in the context of the example above in Figure 9.1. You

can see that, although cross-validation is less conservative than the penalized approach, it was perhaps a bit liberal for this particular dataset, and it picks a more complex model than we would pick had we been able to evaluate the actual risk of the histogram classifiers.

Cross-validation is often used in machine-learning practice to successfully tune regularization parameters. However, not cross-validation is not the answer to all problems, and there are situations where it does not give meaningful results. So, always be critical about it when you use it and try to think if it makes sense in the specific context you are considering.

9.3 Exercises

Exercise 9.3.1 *Show* (9.2).

Exercise 9.3.2 *Show that the cross-validation risk is given by* (9.3).

Exercise 9.3.3 *Let's use the approach proposed for the histograms to help us choose the "right" number of bins. The MATLAB code below generates data from a certain distribution $\mathbb{P}_{XY}$, where $X \in \mathcal{X} = [0,1]^2$ and $Y \in \mathcal{Y} = \{0,1\}$. The function* `hist_classifier.m` *takes this data and computes the histogram classifier with $m \times m$ bins. It outputs both the predicted label for each bin (variable* `c`*) the corresponding empirical risk, and also the leave-one-out cross-validation risk. In addition, the function* `risk.m` *computes an estimate of the true risk of an histogram rule using a validation set from the same distribution (this can only be done with implicit knowledge of the distribution). Provided this validation set is very large this is quite an accurate estimate of the true risk.*

a) Take $n = 1000$ samples and compute the empirical risk of the histogram classifier with m^2 bins, where $m = 1, \ldots, 30$. Compute also the "true" risk of that same classifier and plot these as a function of m. Do you see some interesting behavior? Comment on your findings.

b) Now compute the penalized empirical risk given by

$$\hat{R}_n(\hat{f}_n^{(m)}) + \sqrt{\frac{(\log(m) + \log(m+1) + m^2 \log 2 + \frac{1}{2}\log n}{2n}} \, .$$

Compare it with the true risk. What is the value of m that minimizes the penalized empirical risk? What is the value of m that minimizes the true risk?

c) For the chosen distribution the Bayes' classifier is in the classes $\mathcal{F}_m$ provided m is a multiple of 5. Redo the above questions when $n = 10000$. Comment on your findings.

d) Let's change the distribution of the data. Replace the lines defining `y` *and* `y2` *by*

```
y=((x-ones(n,1)*[0.2 0.8])*[1;1])>0.15*randn(n,1);
y2=((x2-ones(n2,1)*[0.2 0.8])*[1;1])>0.15*randn(n2,1);
```

In this case the Bayes' classifier is not on the class of models under consideration, but can be "well" approximated provided m is growing with the sample size. Redo the above questions for this distribution of the data and $n = 1000, 10000$.

Remark: *despite its simplicity histograms are seldom a good classifier for practical settings. They can only be use for very low dimensional features, and often result in classification rules that are much more complex than those obtained by other methods, such as decision trees and their variants or SVM's.*

```
clear all;
close all;

%Sample size
n=1000;

%Training set
x=[rand(n,1).^0.5 abs(rand(n,1)-0.2).^0.7];
y=logical((sign((x-ones(n,1)*[0.5 1/5])*[0;1]).*sign(0.6+randn(n,1))+1)/2);

%Validation set
n2=1000000;
x2=[rand(n2,1).^0.5 abs(rand(n2,1)-0.2).^0.7];
y2=logical((sign((x2-ones(n2,1)*[0.5 1/5])*[0;1]).*sign(0.6+randn(n2,1))+1)/2);

plot(x(y,1),x(y,2),'b.',x(logical(1-y),1),x(logical(1-y),2),'r.');
legend('Y_i=1','Y_i=0');
xlabel('X_1');ylabel('X_2');
title('Training data set');
pause;

%The following line computes the histogram classifier with 64=8*8 bins, it
%outputs both the corresponding empirical risk, and the label corresponding
%to each cell.
[emprisk,c,cvrisk]=hist_classifier(x,y,8)
emprisk
cvrisk
c

%The following line computes an estimate for the true risk of an histogram
%rule, which is defined by the labels of each bin, given by c. It makes use
%of a validation set, as the exact computation of the risk requires the use
%of the exact data distribution.
risk(x2,y2,c)
```

```
%Input: training data (x,y).
%x must be a n by 2 matrix, with entries in
%[0,1].
%y must take values 0 or 1.
%m^2 is the number of bins in the histogram
```

```
%Output: the empirical risk of the computed rule, the leave-one-out cross-validation risk, an
%parameterizing that rule
%
%The code below is rather inefficient, and can be written in an much more
%efficient way

function [emprisk,c,cvrisk]=hist_classifier(x,y,m);

c=zeros(m,m);
emprisk=0;
cvrisk=0;
for k=1:m;
    for l=1:m;
        tmp=y((x(:,1)>(l-1)/m)&(x(:,1)<=l/m)&(x(:,2)>(k-1)/m)&(x(:,2)<=k/m));
        tmp2=sum(tmp);
        tmp3=length(tmp);
        c(k,l)=(2*tmp2>=tmp3);
        emprisk=emprisk+min(tmp2,tmp3-tmp2);
        cvrisk=cvrisk+tmp2*(tmp2<((tmp3+1)/2))+(tmp3-tmp2)*(tmp2>=((tmp3-1)/2));

    end;
end;
emprisk=emprisk/length(y);
cvrisk=cvrisk/length(y);
```

```
%Input: validation data (x,y).
%x must be a n by 2 matrix, with entries in
%[0,1].
%y must take values 0 or 1.
%One should use a very large validation set to ensure the computed
%empirical risk is close to the true risk.
%c is a matrix parameterizing the histogram rule.
%The code below is rather inefficient, and can be written in an much more
%efficient way.
function risk=est_risk(x,y,c);

m=size(c,2);
risk=0;
for k=1:m;
    for l=1:m;
        tmp=y((x(:,1)>(l-1)/m)&(x(:,1)<=l/m)&(x(:,2)>(k-1)/m)&(x(:,2)<=k/m));
        risk=risk+sum(tmp~=c(k,l));
    end;
end;
risk=risk/length(y);
```

Chapter 10

Decision Trees and Classification

10.1 Penalized Empirical Risk Minimization

Let us summarize the results of the previous chapter. Assume we are considering models from a countable model class $\mathcal{F}$. Suppose we have a map $c : \mathcal{F} \to \mathbb{R}$ such that

$$\sum_{f \in \mathcal{F}} 2^{-c(f)} \leq 1 \ .$$

If we use a prefix code to describe each element of $\mathcal{F}$ and define $c(f)$ to be the codeword length (in bits) for each $f \in \mathcal{F}$, the last inequality is automatically satisfied.

We define the *minimum penalized empirical risk predictor* as

$$\hat{f}_n \equiv \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + \sqrt{\frac{c(f) \log 2 + \frac{1}{2} \log n}{2n}} \right\} \ .$$

We have shown that

$$\mathbb{E}[R(\hat{f}_n)] \leq \inf_{f \in \mathcal{F}} \left\{ R(f) + \sqrt{\frac{c(f) \log 2 + \frac{1}{2} \log n}{2n}} + \frac{1}{\sqrt{n}} \right\} \ .$$

Note that this result tells us that the performance (risk) of $\hat{f}_n$ is on average better than

$$R(\tilde{f}_n) + \sqrt{\frac{c(\tilde{f}_n) \log 2 + \frac{1}{2} \log n}{2n}} + \frac{1}{\sqrt{n}} \ ,$$

where

$$\tilde{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ R(f) + \sqrt{\frac{c(f) \log 2 + \frac{1}{2} \log n}{2n}} \right\} \ .$$

If it happens that the best overall prediction rule

$$f^* = \arg\min_{f \text{ measurable}} R(f) \ ,$$

is close to an $f \in \mathcal{F}$ with a small $c(f)$ then $\hat{f}_n$ will perform almost as well as the best possible prediction rule.

It is frequently convenient to re-write the above bounds in terms of the excess risk $\mathbb{E}[R(\hat{f}_n)] - R^*$, where R^* is the Bayes risk,

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

By subtracting R^* (a constant) from both sides of the above inequality we get

$$\mathbb{E}[R(\widehat{f}_n)] - R^* \leq \inf_{f \in \mathcal{F}} \left\{ \underbrace{R(f) - R^*}_{\text{approximation error}} + \underbrace{\sqrt{\frac{c(f)\log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}}}_{\text{bound on estimation error}} \right\} \ .$$

Note the two terms in this upper bound: $R(f) - R^*$ is the approximation error of a model $f \in F$, and remainder is a bound on the estimation error associated with f. Thus, we see that complexity regularization automatically optimizes a balance between approximation and estimation errors (provided our bound on the estimation error is good). In other words, complexity regularization is *adaptive* to the unknown tradeoff between approximation and estimation (the exact balance depends on the unknown distribution of the data).

When is the above result useful? It is useful if the bound on the estimation error is good, and if the class of models is rich enough to ensure we can trade off approximation and estimation errors in a fair way. To say anything about the latter we need to make assumptions on $\mathbb{P}_{XY}$. The quality of the bound on the estimation error depends on the relation between $\mathbb{P}_{XY}$ and $c(f)$. Namely, the bound will be good if $c\left(\arg\min_{f \in \mathcal{F}} R(f)\right)$ is small.

10.2 Binary Classification

The above results are valid in great generality. Let's consider the particular scenario of binary classification on the unit hypercube. Let $\mathcal{X} = [0,1]^d$, $\mathcal{Y} = \{0,1\}$ and $\ell(\hat{y}, y) \equiv \mathbf{1}\{\hat{y} \neq y\}$. Then $R(f) = \mathbb{E}[\mathbf{1}\{f(X) \neq Y\}|f] = \mathbb{P}(f(X) \neq Y|f)$. As observed in Chapter 2 the Bayes' risk is attained by the Bayes' classifier, which is given by

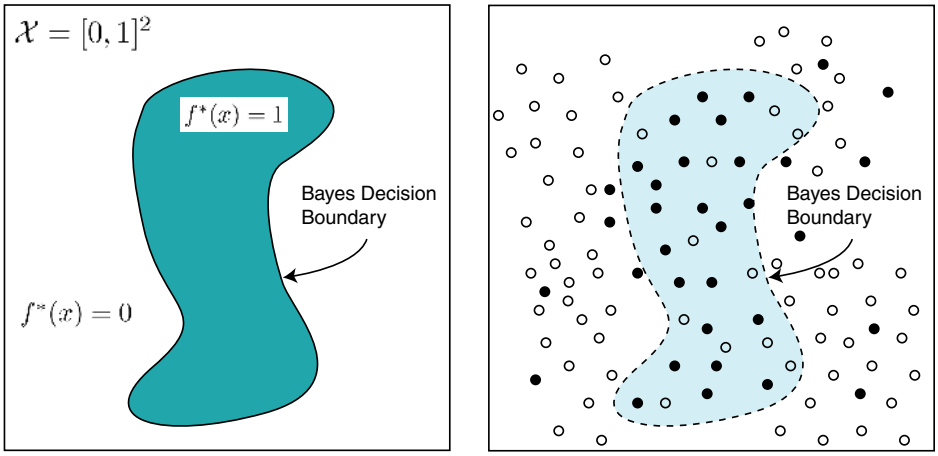
$$f^*(x) = \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1|X = x) \geq \frac{1}{2} \\ 0 & \text{if } \mathbb{P}(Y = 1|X = x) < \frac{1}{2} \end{cases} \ .$$

This classifier can be expressed in a different way. Consider the set $G^* = \{x : \ P(Y = 1|X = x) \geq 1/2\}$. The Bayes' classifier can written as $f^*(x) = \mathbf{1}\{x \in G^*\}$. Therefore the classifier is characterized entirely by the set G^*, if $X \in G^*$ then the "best" guess is that Y is one, and vice-versa. The boundary of G^* is called the *Bayes Decision Boundary*. In Figure 10.1(a) this concept is illustrated. If $\eta(x) = P(Y = 1|X = x)$ is a continuous function then the Bayes decision boundary is simply given by $\{x : \eta(x) = 1/2\}$. Clearly the structure of the decision boundary plays an important role in the difficulty of the learning problem.

Recall that any classification rule can be written as $f(x) = \mathbf{1}\{x \in G_f\}$ for some set $G_f \subseteq \mathcal{X}$. Also, we have seen in Chapter 2 that the excess risk of such a rule is simply given by

$$R(f) - R^* = \int_{G \Delta G^*} |2\eta(x) - 1| d\mathbb{P}_X \ .$$

In this expression we see two aspects of $\mathbb{P}_{XY}$ that play a role in the excess risk. The complexity of the Bayes' decision boundary (that is, the boundary of G^*) affect how small can we make the set $G\Delta G^*$. On the other hand, the behavior of η on that set affects the way we account for the size of $G\Delta G^*$. If η is very close to $1/2$ then the excess risk will be small, even if G and G^* are not very close.



(a) (b)

Figure 10.1: (a) The Bayes classifier and the Bayes decision boundary ; (b) Example of the i.i.d. training pairs.

10.2.1 Empirical Classifier Design

Given n i.i.d. training data, $\{X_i, Y_i\}_{i=1}^n$, we want to construct a classifier $\hat{f}_n$ that performs well on average, that is, we want $\mathbb{E}[R(\hat{f}_n)]$ as close to R^* as possible. In Figure 10.1(b) an example of the i.i.d. training data is depicted.

The construction of a classifier boils down to the estimation of the Bayes' decision boundary. The histogram rule, discussed in a previous lecture, approaches the problem by subdividing the feature space into small boxes and taking a majority vote of the training data in each box. A typical result is depicted in Figure 10.2(a).

The main problem with the histogram rule is that it is solving a much more complex problem than it is actually necessary, since it is indeed estimating $\eta(x)$, and then thresholding this estimate at level $1/2$. However, all we need to do is to estimate the location where $\eta(x)$ crosses the level $1/2$. If $\eta(x)$ is far away from the level $1/2$ we don't need to estimate it accurately. In principle we only need to locate the decision boundary and assign the correct label on either side (notice that the accuracy of a majority vote over a region increases with the size of the region, since a larger region will contain more training samples). The next example illustrates this.

Example 10.2.1 (Three Different Classifiers) *The pictures in Figure 10.2 correspond to the approximation of the Bayes' classifier by three different classifiers:*

The linear classifier and the tree classifier (to be defined formally later) both attack the problem of finding the boundary more directly than the histogram classifier, and therefore they tend to produce much better results in theory and practice. In the following we will demonstrate this for classification trees.

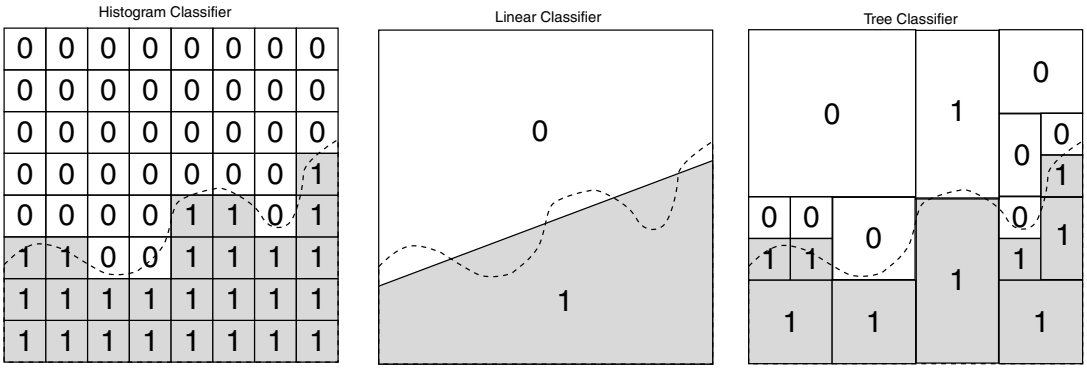


Figure 10.2: (a) Histogram classifier ; (b) Linear classifier; (c)Tree classifier.

10.3 Binary Classification Trees

The idea of binary classification trees is not unlike that of the histogram - partition the feature space into a disjoint sets, and use a majority vote the choose the predicted label of each set. However, the histogram partition is chosen *a priori*, but for the trees we will be able to learn the partition from data. It is easier to describe the binary classification trees through an algorithmic approach consisting of tree growing and tree pruning. The basic idea is to first grow a very large, complicated tree classifier, that explains the the training data very accurately, but has poor generalization characteristics, and then prune this tree, to avoid overfitting.

10.3.1 Growing Trees

The growing process is based on recursively subdividing the feature space. Usually the subdivisions are splits of existing regions into two smaller regions (i.e., binary splits). For simplicity, the splits are perpendicular to one of the feature axis. An example of such construction is depicted in Figure 10.3.

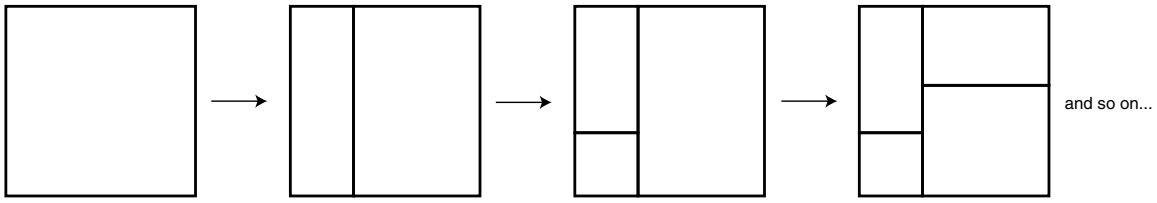


Figure 10.3: Growing a recursive binary tree ($\mathcal{X} = [0,1]^2$).

Often the splitting process is based on the training data, and is designed to separate data with different labels as much as possible. In such constructions, the location of the "split" is data dependent. This causes major difficulties for the analysis (and tuning) of these methods. Alternatively, the splitting location can be taken independent from the training data. The latter approach is the one we are going to investigate in detail, since it is more amenable to analysis, and we will consider Dyadic Decision Trees and Recursive Dyadic Partitions (depicted in Figure 10.4) in particular.

Until now we have been referring to trees, but did not made clear how do trees relate to partitions. It turns out that any decision tree can be associated with a partition of the input space $\mathcal{X}$ and vice-versa. In particular, a Recursive Dyadic Partition (RDP) can be associated with a (binary) tree. In fact, this is a very efficient way of describing a RDP. In Figure 10.4 we illustrate the procedure. Each leaf of the tree corresponds to a cell of the partition. The nodes in the tree correspond to the various partition cells that are generated in the construction of the tree. The orientation of the dyadic split alternates between the levels of the tree (for the example of Figure 10.4, at the root level the split is done in the horizontal axis, at the level below that (the level of nodes 2 and 3) the split is done in the vertical axis, and so on...). The tree is called dyadic because the splits of cells are always at the midpoint along one coordinate axis, and consequently the sidelengths of all cells are dyadic (i.e., powers of 2).

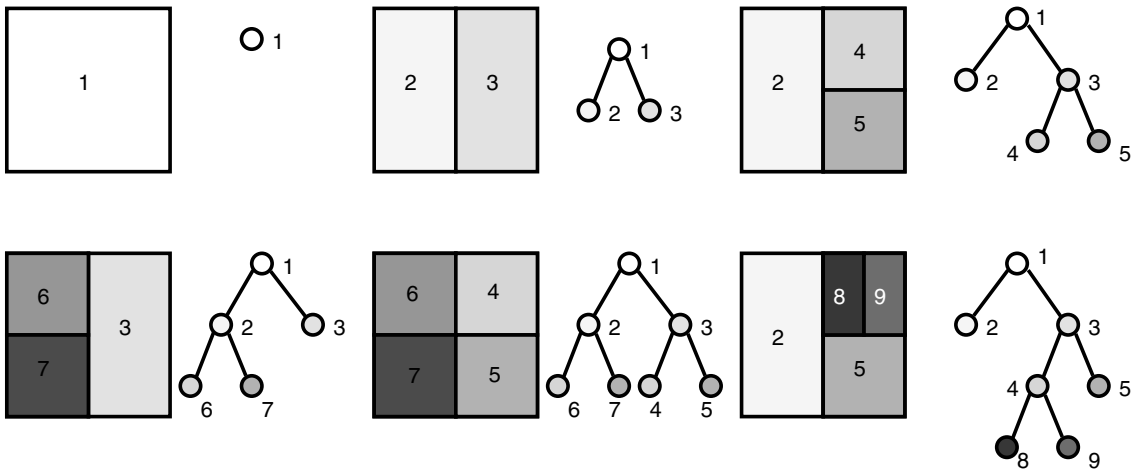


Figure 10.4: Example of Recursive Dyadic Partition (RDP) growing ($\mathcal{X} = [0,1]^2$).

In the following we are going to consider the 2-dimensional case, but all the results can be easily generalized for the d-dimensional case ($d \geq 2$), provided the dyadic tree construction is defined properly. Consider a recursive dyadic partition of the feature space into k boxes of equal size. Associated with this partition is a tree T. Minimizing the empirical risk with respect to this partition produces the histogram classifier with k equal-sized bins. Consider also all the possible partitions corresponding to pruned versions of the tree T. Minimizing the empirical risk with respect to those other partitions results in other classifiers (dyadic decision trees) that are fundamentally different than the histogram rule we analyzed earlier.

10.3.2 Pruning

Let $\mathcal{F}$ be the collection of all possible dyadic decision trees corresponding to recursive dyadic partitions of the feature space (this is a countable set, as you can enumerate all the binary trees). For the pruning we will use penalized empirical risk minimization. For this we need to come up with the complexities $c(f)$. As we've seen before, we can do this by constructing a prefix code. First note that a binary tree can be encoded is a simple way: (i) assign a zero at each internal node and a one at each leaf node (terminal node) (ii) read the code in a breadth-first fashion, top-down, left-right. Figure 10.5 exemplifies this coding strategy. Notice that, since we are considering binary trees, the total number of nodes is twice the number of leafs minus one,

that is, if the number of leafs in the tree is k then the number of nodes is $2k-1$. Therefore to encode a tree with k leafs we need $2k-1$ bits.

Since we want to use the partition associated with this tree for classification we need to assign a decision label (either zero or one) to each leaf. Hence, to encode a decision tree in this fashion we need $3k-1$ bits, where k is the number of leafs. For a tree with k leafs the first $2k-1$ bits of the codeword encode the tree structure, and the remaining k bits encode the classification labels. By construction, this is a prefix code (why?), so we can use it to define $c(f)$. To be precise for any DDT f let $k(f)$ be the number of leafs, and let $c(f) = 3k(f) - 1$. By this construction we know that

$$\sum_{f \in \mathcal{F}} 2^{-c(f)} \leq 1 \ .$$

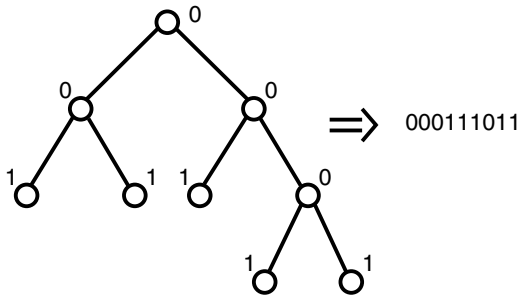


Figure 10.5: Illustration of the tree coding technique: example of a tree and corresponding prefix code.

Let

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + \sqrt{\frac{(3k(f)-1)\log 2 + \frac{1}{2}\log n}{2n}} \right\} \ .$$

This optimization can be solved through a bottom-up pruning process (starting from a very large initial tree T_0) in $O(|T_0|^2)$ operations (Scott, 2005), where $|T_0|$ is the number of leafs in the initial tree (although the typical run time is $O(|T_0|\log(|T_0|))$, as the trees approximating the Bayes' decision boundary are balanced). The complexity regularization theorem tells us that

$$\mathbb{E}[R(\hat{f}_n)] \leq \min_{f \in \mathcal{F}} \left\{ R(f) + \sqrt{\frac{(3k(f)-1)\log 2 + \frac{1}{2}\log n}{2n}} \right\} + \frac{1}{\sqrt{n}} \ . \tag{10.1}$$

10.4 Comparison between Histogram Classifiers and Classification Trees

In the following we will illustrate the idea behind complexity regularization by applying the basic theorem to histogram classifiers and classification trees (using our setup above). We will consider a very general class of distributions $\mathbb{P}_{XY}$, requiring only that the Bayes' decision boundary is not a fractal. Under this mild assumption we will see that classification trees can perform significantly better than the histogram classifier.

10.4.1 Histogram Risk Bound

Let $\mathcal{F}_k^H$ denote the class of histogram classification rules with k^2 bins. Formally

$$\mathcal{F}_k^H = \left\{ f(x) = \sum_{i=1}^{k} \sum_{i=1}^{k} c_{ij} \mathbf{1}\{x \in I_{ij}^{(k)}\}, c_{ij} \in \{0,1\} \right\} ,$$

where

$$I_{ij}^{(k)} = \left[\frac{i-1}{k}, \frac{i}{k} \right) \times \left[\frac{j-1}{k}, \frac{j}{k} \right) .$$

Then $|\mathcal{F}_k^H| = 2^{k^2}$. Let $\mathcal{F}^H = \bigcup_{k \geq 1} \mathcal{F}_k^H$. We can encode each element f of $\mathcal{F}^H$ with $c_H(f) = k^H(f) + (k^H(f))^2$ bits, where the first $k^H(f)$ bits indicate the smallest k such that $f \in \mathcal{F}_k^H$ and the following $(k^H(f))^2$ bits encode the labels of each bin. This is a prefix encoding of all the elements in $\mathcal{F}^H$.

We define our estimator as

$$\hat{f}_n^H = \hat{f}_n^{(\hat{k})} ,$$

where

$$\hat{f}_n^{(k)} = \arg\min_{f \in \mathcal{F}_k^H} \hat{R}_n(f) ,$$

and

$$\hat{k} = \arg\min_{k \geq 1} \left\{ \hat{R}_n(\hat{f}_n^{(k)}) + \sqrt{\frac{(k+k^2)\log 2 + \frac{1}{2}\log n}{2n}} \right\} .$$

Therefore $\hat{f}_n^H$ minimizes

$$\hat{R}_n(f) + \sqrt{\frac{c_H(f)\log 2 + \frac{1}{2}\log n}{2n}} ,$$

over all $f \in \mathcal{F}^H$. This means that

$$\mathbb{E}[R(\hat{f}_n^H)] - R^* \leq \inf_{f \in \mathcal{F}^H} \left\{ R(f) - R^* + \sqrt{\frac{c_H(f)\log 2 + \frac{1}{2}\log n}{2n}} \right\} + \frac{1}{\sqrt{n}} .$$

To proceed with our analysis we need to make some assumptions on the intrinsic difficulty of the problem. We will assume that the Bayes' decision boundary is a "well-behaved" 1-dimensional set, in the sense that it has box-counting dimension one (see Appendix 10.8). Let's formalize this assumption.

Assumption 10.4.1 (Box-Counting Assumption) *Take $\mathcal{X} = [0,1]^2$ and divide it in ℓ^2 squares of size $1/\ell^2$ each. These sets form a partition of $[0,1]^2$. We assume there is a number $C > 1$ such the Bayes decision boundary passes through at most $C\ell$ boxes, regardless of the value $\ell = 1, 2, \ldots$.*

Remark 10.4.1 *This assumption is essentially stating that the overall length of the Bayes' decision is assumed to be finite. More specifically, it is assumed to have at most length given by $\sqrt{2}C$.*

This implies that, for an histogram with k^2 bins, the Bayes' decision boundary intersects less than Ck bins, where C is a constant that does not depend on k. In addition, we will also make an assumption on the marginal distribution of X.

Assumption 10.4.2 (Feature density Assumption) *The distribution $\mathbb{P}_X$ is such that $\mathbb{P}_X(A) \leq p_{\max} vol(A)$, for any measurable subset $A \subseteq [0,1]^2$, where $vol(A)$ denotes the volume of set A.*

This assumption means that the samples collected do not accumulate anywhere in the unit square. What is the excess risk of the best histogram rule in this case? An arbitrary element of $\mathcal{F}_k^H$ can be written as

$$f(x) = \mathbf{1}\{x \in G_f\} = \sum_{i=1}^{k} \sum_{i=1}^{k} c_{ij} \mathbf{1}\{x \in I_{ij}^{(k)}\} \ ,$$

where

$$G_f = \bigcup_{(i,j):c_{ij}=1} I_{ij}^{(k)} \ .$$

Now, the excess risk of such a rule can be simply written as

$$\begin{aligned} R(f) - R^* &= \int_{G \Delta G^*} |2\eta(x) - 1| d\mathbb{P}_X \\ &= \sum_{i,j} \int_{(G \Delta G^*) \cap I_{ij}^{(k)}} |2\eta(x) - 1| d\mathbb{P}_X \\ &= \sum_{(i,j):c_{ij}=1} \int_{I_{ij}^{(k)} \setminus G^*} |2\eta(x) - 1| d\mathbb{P}_X \ + \sum_{(i,j):c_{ij}=0} \int_{I_{ij}^{(k)} \cap G^*} |2\eta(x) - 1| d\mathbb{P}_X \ . \end{aligned}$$

How to make this small? If $I_{ij}^{(k)}$ is contained in G^* then take $c_{ij} = 1$. If $I_{ij}^{(k)} \cap G^* = \emptyset$ take $c_{ij} = 0$. Finally if neither is true then the boundary of G^* must intersect $I_{ij}^{(k)}$, so neither term can be zero and we'll have to pay a price. Nevertheless

$$\int_{I_{ij}^{(k)} \Delta G^*} d\mathbb{P}_X \leq \int_{I_{ij}^{(k)}} d\mathbb{P}_X = \mathbb{P}(X \in I_{ij}^{(k)}) \leq p_{\max}/k^2 \ .$$

Therefore we conclude that

$$\min_{f \in \mathcal{F}_k^H} R(f) - R^* \ \leq \ \frac{p_{\max}}{k^2} Ck = \frac{C p_{\max}}{k} \ .$$

From our general analysis we know that, for any $k \in \mathbb{N}$

$$\mathbb{E}[R(\hat{f}_n^H)] - R^* \ \leq \ \frac{C p_{\max}}{k} + \sqrt{\frac{(k+k^2)\log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \ .$$

The two main terms in the above expression have the same order if we take $k = n^{1/4}$ (for n large) therefore

$$\mathbb{E}[R(\hat{f}_n^H)] - R^* \ = \ O(n^{-1/4}), \text{ as } n \to \infty \ .$$

10.4.2 Dyadic Decision Trees

Now let's consider the dyadic decision trees, under the assumptions above, and contrast these with the histogram classifier. Let

$$\mathcal{F}_k^T = \{\text{tree classifiers with } k \text{ leafs}\} \ .$$

Let $\mathcal{F}^T = \bigcup_{k \geq 1} \mathcal{F}_k^T$. We can prefix encode each element f of $\mathcal{F}^T$ with $c_T(f) = 3k(f) - 1$ bits, as described before.

Following our penalized empirical risk approach we define the estimator

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}^T} \left\{ \hat{R}_n(f) + \sqrt{\frac{(3k(f) - 1)\log 2 + \frac{1}{2}\log n}{2n}} \right\} \ .$$

Our analysis tells us that

$$\mathbb{E}[R(\hat{f}_n^T)] - R^* \ \leq \ \inf_{f \in \mathcal{F}^T} \left\{ R(f) - R^* + \sqrt{\frac{(3k(f) - 1)\log 2 + \frac{1}{2}\log n}{2n}} \right\} + \frac{1}{\sqrt{n}} \ .$$

Now assume the Bayes decision boundary satisfies the box-counting assumption stated. We have the following result

Lemma 10.4.1 *Let k be an integer power of 2. If the boundary set satisfies the box-counting assumption with $C \geq 1/3$, then there is a RDP with at most $3Ck$ leafs, and such that the boundary is contained in at most Ck partition elements each with volume $1/k^2$.*

Proof We provide the proof in a relatively informal way, but with this information it should be easy to further formalize the reasoning if desired. Consider a tree that such that the boundary is contained in at most Ck partition elements each with volume $1/k^2$ and assume it is minimal, meaning there are no nodes that can be removed without changing this property. Let's count the number of nodes at each level. Clearly we don't need any nodes at depth greater than $2\log_2 k$ (the depth of the leafs nodes containing the boundary). For even i the possible nodes in the tree correspond to square partition elements (each with volume 2^{-i}). By assumption we know that the boundary goes through at most $C2^{i/2}$ such nodes (let's call these type A nodes). So these must be kept. Their siblings (nodes with the same parent) must also be kept, because of the tree structure). Call these nodes type B nodes. All other nodes at that level can be pruned. If i is odd the corresponding partition elements will not be square, so our assumption does not tell us anything about these. In the worst case, all the children of the type A nodes in the previous level must be kept, but we don't need the children of type B nodes. Therefore, at level i odd we have at most $2C2^{(i-1)/2}$ nodes. So, for i even the combined number of nodes of level i and $i - 1$ is at most $3C2^{i/2}$.

The above reasoning can be use to give us an upper bound on the number of nodes in the tree. Namely, this tree has at most

$$\sum_{i=0:i \text{ even}}^{2\log_2 k} 3C2^{i/2} = \sum_{\ell=0}^{\log_2 k} 3C2^{\ell} = 3C(2^{1+\log_2 k} - 1) = 3C(2k - 1)$$

nodes, where we used the fact that $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$. This means this tree has at most

$$\frac{3C(2k-1)+1}{2} = 3Ck + \frac{1-3C}{2} \leq 3Ck$$

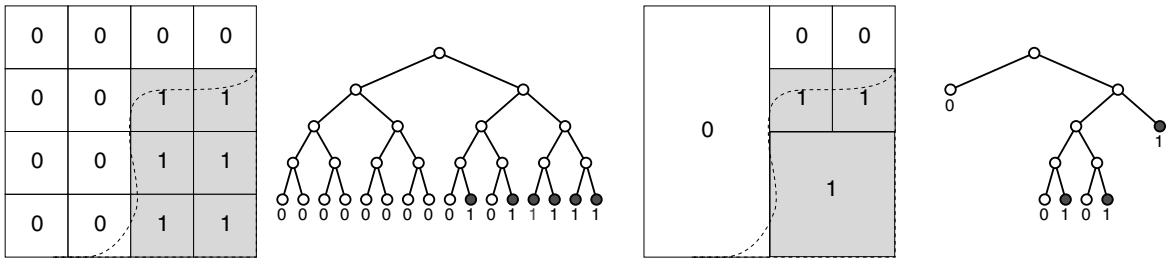
leafs, provided $C \geq 1/3$. □

The above lemma tells us that there there exists a DDT with at most $3Ck$ leafs that has the same risk as the best histogram with $O(k^2)$ bins. Therefore, using equation (10.1) we have

$$\mathbb{E}[R(\hat{f}_n^T)] - R^* \;\leq\; Cp_{\max}/k + \sqrt{\frac{(3(3Ck)-1)\log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \;.$$

We can balance the terms in the right side of the above expression using $k = n^{1/3}$ (for n large) therefore

$$\mathbb{E}[R(\hat{f}_n^T)] - R^* \;=\; O(n^{-1/3}), \quad \text{as } n \to \infty \;.$$



(a) (b)

Figure 10.6: Illustration of the tree pruning procedure: (a) Histogram classification rule, for a partition with 16 bins, and corresponding binary tree representation (with 16 leafs). (b) Pruned version of the histogram tree, yielding exactly the same classification rule, but now requiring only 6 leafs. (*Note:* The trees were constructed using the procedure of Figure 10.4)

10.5 Final Comments and Remarks for Implementation

Trees generally work much better than histogram classifiers. This is essentially because they provide much more efficient ways of approximating the Bayes' decision boundary (as we saw in our example, under reasonable assumptions on the Bayes' boundary, a tree encoded with $O(k)$ bits can describe the same classifier as an histogram that requires $O(k^2)$ bits).

The dyadic decision trees studied here are different than classical tree rules, such as CART (introduced in Breiman, Friedman, Stone and Olshen (1984)) or C4.5 introduced by Quinlan in 1993 (see also Quinlan (2014)). Those techniques select a tree according to

$$\hat{k} = \arg\min_{k \geq 1} \left\{ \widehat{R}_n(\hat{f}_n^{(k)}) + \alpha k \right\} \;,$$

for some $\alpha > 0$ whereas ours was roughly

$$\hat{k} = \arg\min_{k \geq 1} \left\{ \hat{R}_n(\hat{f}_n^{(k)}) + \alpha\sqrt{k} \right\} \;,$$

for $\alpha \sim 1/\sqrt{n}$. The square root penalty is essential for the risk bound. No such bound exists for CART or C4.5 in the classification setting. Moreover, experimental work has shown that the square root penalty often performs better in practice for classification. Finally, recent results Scott and Nowak (2006) show that a slightly tighter bounding procedure for the estimation error can be used to show that dyadic decision trees (with a slightly different pruning procedure) achieve a rate of

$$\mathbb{E}[R(\hat{f}_n^T)] - R^* \;=\; O((\log n)^{3/2} n^{-1/2}), \quad \text{as } n \to \infty \;,$$

which turns out to be the minimax optimal rate up to logarithmic factors (i.e., under the boundary assumptions above, no method can achieve a faster rate of convergence to the Bayes error). In Figure 10.7 we present the results of a small simulation of the method described in Exercise 10.6.2. The amount of penalization was chosen using a variant of cross-validation. The corresponding result when using histograms is also displayed, and one clearly sees that DDT's learn much better from data in this setting.

One can implement the above procedures exactly as prescribed, and will see they work, but are quite conservative, always choosing models that have a small number of parameters. The reason for this is that we are over-estimating the estimation error (after all, we have an upper bound on the estimation error). A way to get around this is to instead use an unbiased estimator of the estimation error. This can be achieved by using cross-validation, but then it becomes much harder to provide guarantees for the implemented procedure. Nevertheless, the above theory gives an indication about the type of penalization one should use, and therefore informs on the right algorithmic approach.

10.6 Exercises

Exercise 10.6.1 *In this exercise you'll check that lemma 10.4.1 cannot really be further improved, by studying a particular example. Suppose we Bayes decision boundary bisects* $[0,1]^2$ *along the diagonal. This means that, if you partition* $[0,1]^2$ *into* k^2 *squares then the boundary passes through exactly* k *partition elements (ignoring any issues with intersections with the boundaries of partition elements). This is illustrated in Figure 10.8. If we apply Lemma 10.4.1 (with* $C = 1$*) we conclude that there is a RDP with at most* $3k$ *elements, so that the boundary is fully contained in* k *partition elements of size* $1/k^2$*.*

a) Take $k = 8$ *and identify the smallest RDP such that the boundary is contained in* 8 *square partition elements of size* $1/64$*. How many partition elements does this RDP have? Compare this with the bound* $3k$ *from the lemma.*

b) Now consider an arbitrary value k *(note that* k *must be a power of* 2*). Give the number of partition elements (leafs) of the RDP as a function of* k*. How does this compare with the bound* $3k$*?*

Exercise 10.6.2 *In this chapter we have seen that a procedure based on Recursive Dyadic Partitions was able to achieve a rate of convergence to the Bayes risk of* $n^{-1/3}$*, when the Bayes' decision boundary is a one-dimensional well-behaved set in a two-dimensional space. It turns out this is not the best possible rate of convergence (among all classification rules). However a modification of the bounding argument can be devised such that it is possible to use dyadic trees*

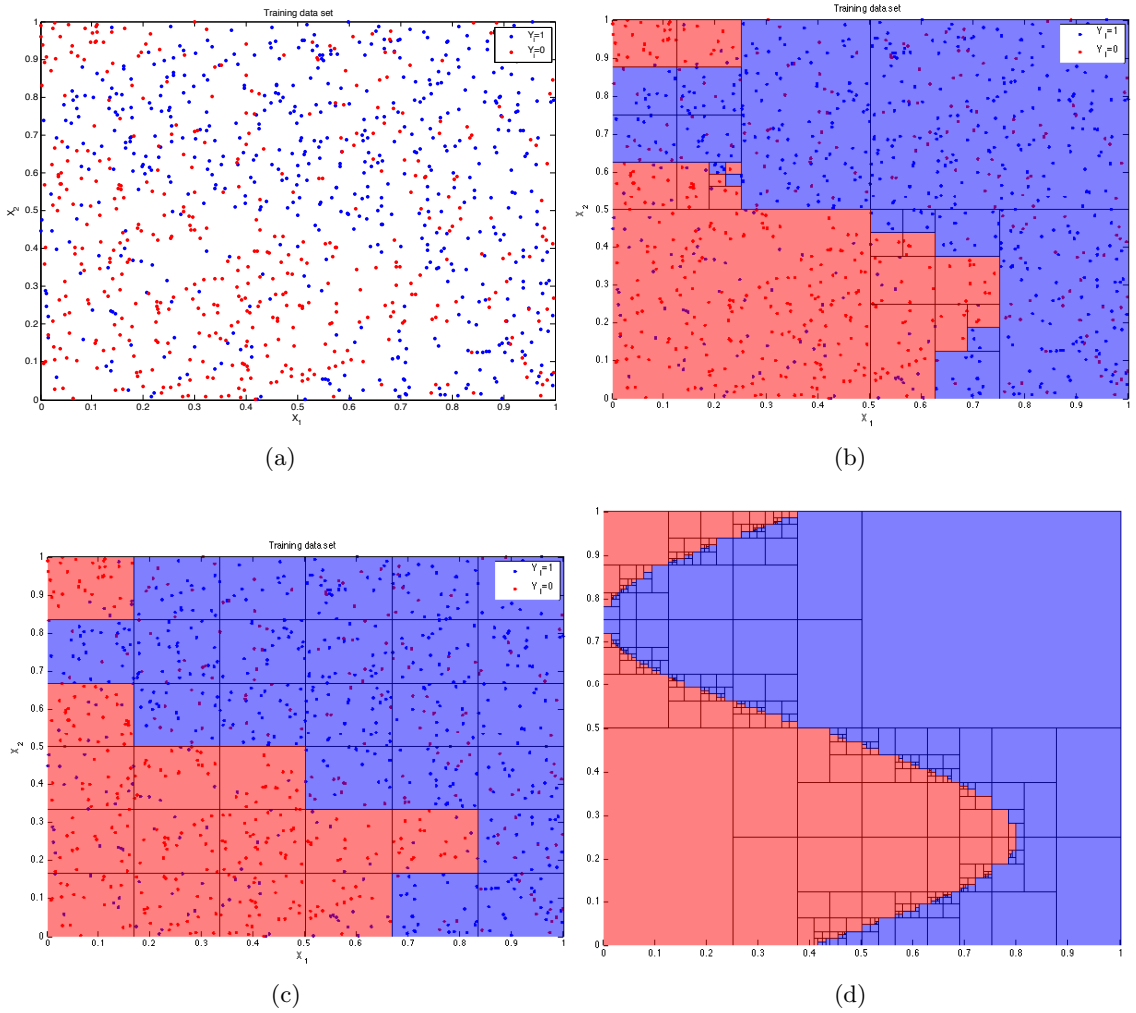
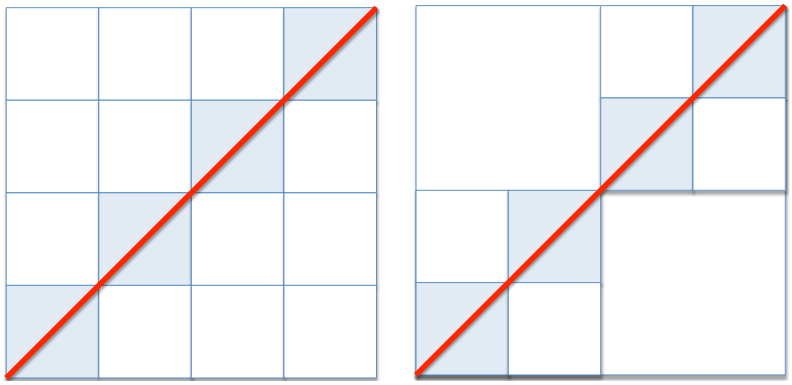


Figure 10.7: Illustration of the behavior of dyadic decision trees. In panel (a) we see an example of a dataset generated as 10000 i.i.d. samples from a certain distribution. In panel (b) a dyadic decision tree was fitted to the data using the method described in Exercise 10.6.2. The amount of penalization was chosen using a variant of cross-validation. In panel (c) is the resulting histogram classifier, where the number of bins was chosen using cross-validation. Finally in panel (d) is an high-resolution approximation of the true decision boundary (obtained using a dataset of 1000000 points).



(a) (b)

Figure 10.8: Illustration of RDPs for the exercise. The red line represents the boundary. For both panels $k = 4$. In (a) we see the histogram partition, where the shaded element correspond to those intersecting the boundary (4 out of 16). In (b) we see the best RDP, which has 10 elements (meaning the corresponding tree representation has 10 leafs).

to get the best possible rates. This was done in Scott and Nowak (2006). This exercise follows loosely the ideas and results in that paper, and will allow you to essentially replicate their results with some extra assumptions.

a) Recall that any decision tree is associated with a partition of the unit square, where each leaf of the tree corresponds to one partition element. Let f be a dyadic decision tree, and $\mathcal{P}(f)$ denote the corresponding partition of the unit square $[0,1]^2$. Note that the difference between the true and empirical risk of this decision tree can be written as the sum of the contributions of each leaf in the tree.

$$R(f) - \hat{R}_n(f) = \sum_{A \in \mathcal{P}(f)} R(A) - \hat{R}_n(A) \ ,$$

where $R(A) = \mathbb{E}[\mathbf{1}\{X \in A, f(X) \neq Y\}]$, and correspondingly $\hat{R}_n(A) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{X_i \in A, f(X_i) \neq Y_i\}$.

We can bound the error at a particular leaf using Hoeffding's inequality: for a single leaf A with probability greater or equal to $1 - \delta$

$$R(A) - \hat{R}_n(A) \leq \sqrt{\log(1/\delta)/(2n)} \ .$$

When applied to Bernoulli random variables this bound is referred to as the additive form of Chernoff's bound. Unfortunately it is not adequate for our purposes. Another version, called the relative form*, is more appropriate: Let $S_n = \sum_{i=1}^n B_i$, where B_i are i.i.d. Bernoulli random variables. Then*

$$\mathbb{P}(Sn \leq (1-\epsilon)\mathbb{E}[S_n]) \leq e^{-\mathbb{E}[S_n]\epsilon^2/2} \ ,$$

where $0 \leq \epsilon \leq 1$.

Use this inequality to obtain another upper bound on $R(A) - \hat{R}_n(A)$ valid for each set A. This bound should hold with probability $1 - \delta$ and will depend on $R(A)$.

b) We want to a bound that holds for all the possible RDPs. A way to do so is to derive a bound that holds for all the leafs simultaneously (i.e., we need to take a union bound over leafs). Devise a prefix encoding scheme for each leaf of a dyadic tree. The goal is to obtain a unique code for every possible dyadic leaf (each leaf corresponds to a dyadic rectangle of a certain size and at a certain location; you need to encode each rectangle of this form - think of the way you can encode its position in the tree). A prefix code can be devised so that the codelength for each leaf is proportional to the depth of the leaf in a dyadic tree. Use this code and the relative form of Chernoff's bound above to obtain a bound of the form: with probability at least $1-\delta$

$$R(f) - \hat{R}_n(f) \leq \sum_{A \in \mathcal{P}(f)} C(A, n, \delta, R(A)) \ , \qquad \forall f \in \mathcal{F} \ ,$$

where the bound holds for all possible dyadic decision trees.

c) In order for the upper bound derived in (b) to be of use, we must eliminate the dependence on $R(A)$*, which of course is unknown. To this end assume that* $\mathbb{P}_X$ *is such that, for any set* $G \in [0,1]^2$ *we have* $\mathbb{P}_X(G) \leq p_{\max} Vol(G)$*. This means that the marginal distribution of the features is not heavily concentrated in any area of the entire feature space. Using this assumption, derive an upper bound for* $R(A)$ *that only depends on the size of* A *and the constant* $p_{\max}$*. Use this upper bound to derive another upper bound on* $R(f) - \hat{R}_n(f)$*.* ***Hint:*** *this bound should roughly say that, with probability greater or equal to* $1-\delta$*.*

$$R(f) - \hat{R}_n(f) \leq \textit{constant} \sum_{A \in \mathcal{P}(f)} \sqrt{2^{-j(A)} \frac{j(A) + 1 + \log(1/\delta)}{n}} \ ,$$

where $j(A)$ *is the depth of leaf* A *in the dyadic tree.*

d) You are now in a position to devise a good procedure to select a dyadic tree based on training data and construct an expected excess risk bound, using essentially the ideas devised in class. This will yield an oracle bound with two terms, one corresponding to the approximation error and one corresponding to the bound on estimation error, as before. Furthermore the estimation error term should be proportional to

$$\sum_{A \in \mathcal{P}(f)} \sqrt{2^{-j(A)} \frac{j(A) + 1 + \log(n)}{n}} \ .$$

e) We can now use the bound derived to study the rate of convergence of the estimator for the class of well behaved Bayes' decision boundaries, as considered in this chapter. In particular assume that if you divide the feature space into k^2 *equal sized squares then the Bayes' decision boundary intersects at most* Ck *bins, where* $C > 1$ *is a constant that does not depend on* k *(the box-counting assumption in the chapter). Let* f_k^* *denote the dyadic decision tree obtained starting with the* k^2 *cell histogram partition and pruning back all cells that do not intersect the Bayes' decision boundary. Show that the approximation error for this tree is* $O(1/k)$*. To bound the estimation error show that* f_k^* *has* $O(2^{j/2})$ *leafs at depth* j *and use this fact.*

Noticing that the excess risk bound cannot ever be better than $1/\sqrt{n}$ we know that the maximum leaf depth in f_k^* is $O(\log n)$. Taking this into account show that the rate of convergence of the proposed estimator is, ignoring logarithmic factors $O(n^{-1/2})$ (If you proceed as above your bound should be something like $O((\log n)^{3/2}/\sqrt{n})$).

f) Note that the estimator obtained is a penalized empirical risk estimation, but now we don't have the squared root penalty anymore, and instead it is an additive penalty (sum over leafs). Consequently $\hat{f}_n$ can be obtained by starting with a histogram of very small cells (e.g., $O(n)$ cells) and pruning from the bottom-up. Briefly outline the pruning algorithm and justify why it leads to the optimal tree.

10.7 Appendix: abbreviated solution of Exercise 10.6.2

In part (a) we show that, for any set $A \in \mathcal{X}$ we have

$$\mathbb{P}\left(R(A) - \hat{R}_n(A) < \sqrt{\frac{2R(A)}{n}\log\left(\frac{1}{\delta}\right)}\right) \geq 1-\delta \ .$$

This bound does not hold for all possible leafs of a DDT simultaneously. To get such a bound we first devise a code for each possible leaf. At level j in the tree there are exactly 2^j possible leafs, so we can use $j+1$ bits to encode the depth of a leaf and j bits more to encode the exact location of the leaf. So, if we take $c(A) = 2j(A)+1$ where $j(A)$ is the depth of the leaf we ensure that

$$\sum_{A: A \text{ is a leaf in a DDT}} 2^{-c(A)} \leq 1 \ .$$

Using a union bound as in class we conclude that

$$\mathbb{P}\left(\forall A \qquad R(A) - \hat{R}_n(A) < \sqrt{\frac{2R(A)}{n}\left((2j(A)+1)\log 2 + \log\left(\frac{1}{\delta}\right)\right)}\right) \geq 1-\delta \ ,$$

which is a uniform bound over all possible leafs a DDT might have. A way to encode a DDT is to describe its leafs and the corresponding label, therefore in the end we have the bound

$$\mathbb{P}\left(\forall f \in \mathcal{F} \qquad R(f) - \hat{R}_n(f) < \underbrace{\sum_{A \in \mathcal{P}(f)} \sqrt{\frac{2R(A)}{n}\left((2j(A)+2)\log 2 + \log\left(\frac{1}{\delta}\right)\right)}}_{\text{pen}(f)}\right) \geq 1-\delta \ .$$

The $(2j(A)+2)$ term is the number of bits it takes to describe each leaf and the corresponding label (one extra bit).

Now, we are in a very similar situation as before. The only difference is the term $\text{pen}(f)$, which is a bit different (and nicer in certain regards). Using the same approach as before we can get an expected excess risk bound for a penalized estimator. Namely define

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}^T} \left\{\hat{R}_n(f) + \text{pen}(f)\right\} \ .$$

Then

$$\mathbb{E}[R(\hat{f}_n)] - R^* \leq \inf_{f \in \mathcal{F}^T} \{R(f) - R^* + \text{pen}(f) + \delta\} \ .$$

To study the performance of this classifier we are going to make use of the Assumptions in Chapter 9, namely Assumption 1 (the box counting assumption) and the assumption that, for any set A, we have $\mathbb{P}(X \in A) \leq p_{\max}|A|$, where $|A|$ denotes the volume of A. Noting that $|A| = 2^{-j(A)}$ and that $R(A) \leq \mathbb{P}(X \in A) \leq p_{\max}|A|$ we can put all this together and get concrete expressions for the proposed classifier and corresponding risk bound. For concreteness, let's take $\delta = 1/\sqrt{n}$.

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}^T} \left\{ \hat{R}_n(f) + \sum_{A \in \mathcal{P}(f)} \sqrt{\frac{2p_{\max}2^{-j(A)}}{n} \left((2j(A)+2)\log 2 + \frac{1}{2}\log n \right)} \right\} \ ,$$

and

$$\mathbb{E}[R(\hat{f}_n)] - R^* \leq \inf_{f \in \mathcal{F}^T} \left\{ R(f) - R^* + \sum_{A \in \mathcal{P}(f)} \sqrt{\frac{2p_{\max}2^{-j(A)}}{n} \left((2j(A)+2)\log 2 + \frac{1}{2}\log n \right)} + \frac{1}{\sqrt{n}} \right\} \ .$$

It is important to note that this penalty actually favors having many deep leafs (and these are good at describing the Bayes decision boundary). However, to have many deep leafs we also need to have a significant number of leafs at higher levels (this is a binary tree). So it is not immediately clear what is the "best" tree. To use the bound we need to find a DDT $f \in \mathcal{F}$ such that the r.h.s. of the above bound is small. Here we must make use of assumption 1. Let f the the best approximating DDT with the deepest leafs having volume $1/k^2$ (so these are at depth $2\log_2 k$). As we did before in the Chapter we know that

$$R(f) - R* = O(1/k) \ .$$

Also, we know from the assumption 1 that at each level j this tree has at most $O(2^{j/2})$ nodes, which means it has at most $O(2^{j/2})$ leafs at level j. Therefore

$$\begin{aligned}
\text{pen}(f) &= \sum_{A \in \mathcal{P}(f)} \sqrt{\frac{2p_{\max}2^{-j(A)}}{n} \left((2j(A)+2)\log 2 + \frac{1}{2}\log n \right)} \\
&\leq C' \sum_{j=1}^{2\log_2 k} 2^{j/2} \sqrt{\frac{2p_{\max}2^{-j}}{n} \left((2j+2)\log 2 + \frac{1}{2}\log n \right)} \\
&\leq C' \sum_{j=1}^{2\log_2 k} \sqrt{\frac{2p_{\max}}{n} \left((4\log_2 k + 2)\log 2 + \frac{1}{2}\log n \right)} \\
&\leq 2C' \log_2 k \sqrt{\frac{2p_{\max}}{n} \left((4\log_2 k + 2)\log 2 + \frac{1}{2}\log n \right)} \ ,
\end{aligned}$$

where C' is a large enough constant, so that the number of leafs at depth j is less than $C'2^{j/2}$ (this is what the big-O notation is telling you). Provided k is significantly larger than $\log n$ this means that

$$\text{pen}(f) = O\left(\log k \sqrt{\frac{\log k}{n}} \right) \ .$$

So, if we take $k = O(n)$ we conclude that $R(f) - R^* = O(1/n)$ and $\text{pen}(f) = O\left(\log^{3/2}(n)\frac{1}{\sqrt{n}}\right)$. So, overall, we conclude that

$$\mathbb{E}[R(\hat{f}_n)] - R^* = O\left(\frac{\log^{3/2} n}{\sqrt{n}}\right) \ .$$

It turn out that, up to the precise logarithmic factors, this is the best one can hope to achieve with any classification rule. So these algorithm is, in terms of error decay rates, optimal under the assumptions made. Furthermore, this is a very easy algorithm to implement, because the penalization is a sum over the leafs. This means that a bottom-up pruning approach will be able to successfully solve the computation of the classifier. In a nutshell, one will grow the tree very deep (e.g., until all the samples in the leafs are of the same label) and then prune it back, simply deciding if merging to leafs if the combined error is not larger than the corresponding local penalization price. Such an algorithm runs in $O(n)$ operations, so it is extremely fast!

10.8 Appendix: Box Counting Dimension

The notion of dimension of a sets arises in many aspects of mathematics, and it is particularly relevant to the study of fractals (that besides some important applications make really cool t-shirts). The dimension somehow indicates how we should measure the contents of a set (length, area, volume, etc...). The box-counting dimension is a simple definition of the dimension of a set. The main idea is to cover the set with boxes with sidelength r. Let $N(r)$ denote the smallest number of such boxes, then the box counting dimension is defined as

$$\lim_{r \to 0} \frac{\log N(r)}{-\log r}.$$

Although the boxes considered above do not need to be aligned on a rectangular grid (and can in fact overlap) we can usually consider them over a grid and obtain an upper bound on the box-counting dimension. To illustrate the main ideas let's consider a simple example, and connect it to the classification scenario considered before.

Let $f : [0,1] \to [0,1]$ be a Lipschitz function, with Lipschitz constant L (i.e., $|f(a) - f(b)| \leq L|a-b|, \ \forall a, b \in [0,1]$). Define the set

$$A = \{x = (x_1, x_2) : x_2 = f(x_1)\} \ ,$$

that is, the set A is the graphic of function f.

Consider a partition with k^2 squared boxes (just like the ones we used in the histograms), the points in set A intersect at most $C'k$ boxes, with $C' = (1 + \lceil L \rceil)$ (and also the number of intersected boxes is greater than k). The sidelength of the boxes is $1/k$ therefore the box-counting dimension of A satisfies

$$\begin{aligned} \dim_B(A) &\leq \lim_{1/k \to 0} \frac{\log C'k}{-\log(1/k)} \\ &= \lim_{k \to \infty} \frac{\log C' + \log(k)}{\log(k)} \\ &= 1 \ . \end{aligned}$$

The result above will hold for any "normal" set $A \subseteq [0,1]^2$ that does not occupy any area. For most sets the box-counting dimension is always going to be an integer, but for some "weird" sets (called fractal sets) it is not an integer. For example, the Koch curve[1] is such a set, and has box-counting dimension $\log(4)/\log(3) = 1.26186\ldots$. This means that it is not quite as small as a 1-dimensional curve, but not as big as a 2-dimensional set (hence occupies no area).

To connect these concepts to our classification scenario consider a simple example. Let $\eta(x) = P(Y = 1|X = x)$ and assume $\eta(x)$ has the form

$$\eta(x) = \frac{1}{2} + x_2 - f(x_1), \quad \forall x \equiv (x_1, x_2) \in \mathcal{X}, \tag{10.2}$$

where $f : [0,1] \to [0,1]$ is Lipschitz with Lipschitz constant L. The Bayes classifier is then given by

$$f^*(x) = \mathbf{1}\{\eta(x) \geq 1/2\} \equiv \mathbf{1}\{x_2 \geq f(x_1)\} \ .$$

This is depicted in Figure 10.9. Note that this is a special, restricted class of problems. That is, we are considering the subset of all classification problems such that the joint distribution $\mathbb{P}_{XY}$ satisfies $\mathbb{P}(Y = 1|X = x) = 1/2 + x_2 - f(x_1)$ for some function f that is Lipschitz. The Bayes decision boundary is therefore given by

$$A = \{x = (x_1, x_2) : x_2 = f(x_1)\} \ .$$

As we observed before this set has box-counting dimension 1.

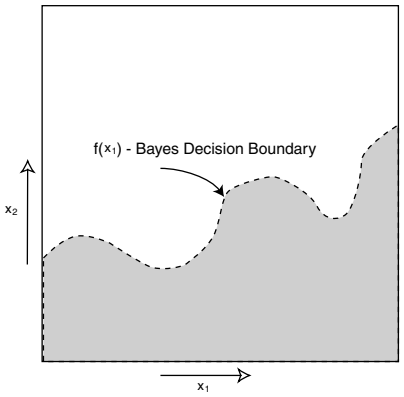


Figure 10.9: Bayes decision boundary for the setup described in Appendix 10.8.

[1] see for example http://classes.yale.edu/fractals/IntroToFrac/InitGen/InitGenKoch.html.

Chapter 11

Vapnik-Chervonenkis (VC) Bounds

In our past lectures we have derived a number of generalization bounds, but all these had one thing in common: they required the class of candidate models to be either finite or countable. In other words the class of candidate models needed to be enumerable. This was essential for the approach we took, as at some point we needed to bound the probability of a union of events. Our way around it was to use a union bound, which simply states that the probability of the union of events is at most the sum of the probabilities of each of the individual events. All this make sense if we are considering a finite or countable union of events, but it is not so sensible otherwise (e.g., how do we define the summation operator for uncountably many elements?).

In many cases of practical importance the collection of candidate models is uncountably infinite (generally it is easier to formulate the optimization problem of finding a good model in this setting too - so this is also an advantage when devising a proper algorithm). Let's see a motivational example that will help us understand what are the possible ways of dealing with this issue.

Example 11.0.1 *Let the feature space be $\mathcal{X} = \mathbb{R}$ and the label space be $\mathcal{Y} = \{0, 1\}$. Consider the following class of models*

$$\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y} \ : f(x) = \mathbf{1}\{x \geq t\},\ t \in [0, 1]\} \ .$$

Clearly this class is uncountable, therefore we cannot apply our current generalization bounds.

11.1 Two Ways to Proceed

Discretize or Quantize the Collection $\mathcal{F}$: this is a simple way around the problem. If one considers a modification of the algorithms it typically does lead to cumbersome and messy algorithms and one requires extra assumptions in order to give performance guarantees (you have encountered this approach in the exercises in Chapter 7). In some cases it is also possible to use the discretization as technical tool for the analysis, but this generally requires quite a bit of sophistication. Let's revisit our example above.

Example 11.1.1 *Let's consider a discretized version of $\mathcal{F}$. Let's assume that $\mathcal{X} = [0, 1]$ and also that $\mathbb{P}_X(A) \leq C|A|$ for all measurable sets $A \subseteq \mathcal{X}$. Let $Q \in \mathbb{N}$ be an integer, and define*

$$\mathcal{F}_Q = \{f : \mathcal{X} \to \mathcal{Y} \ : f(x) = \mathbf{1}\{x \geq t\},\ t \in \mathcal{T}_Q\} \ ,$$

where $\mathcal{T} = \left\{0, \frac{1}{Q}, \frac{2}{Q}, \ldots, \frac{Q}{Q}\right\}$. *It's not hard to show that for any* $f \in \mathcal{F}$ *there exists an* $f' \in \mathcal{F}_Q$ *such that*

$$|R(f) - R(f')| \leq \frac{C}{Q} \ ,$$

therefore if we choose Q *large enough then* $\mathcal{F}_Q$ *and* $\mathcal{F}$ *are nearly equivalent.*

This trick can be used in many situations, but often leads to to practical difficulties when trying to implement the estimators, besides making their analysis a bit more cumbersome.

Identical Empirical Errors: Recall that we are choosing a model based on a set of training data. Consider the setting in the examples above. Given a certain dataset, it is obvious that there are many elements of $\mathcal{F}$ that will give rise to exactly the same labeling of the dataset, therefore, from the point of view of a fit to the training data these are indistinguishable, and can be considered equivalent. In the example above, if we are using a dataset of size n, there will be only $O(n)$ such equivalent classifiers. So the "effective" size of $\mathcal{F}$, when looking through the lens of the training data, is only n. This is a hand-waving argument, but it can be formalized nicely. In fact, you already formalized somehow in Exercise 5.4.1 under stringent assumptions. Nevertheless, the main principle is the characterization of the complexity of the model class with respect to ANY set of training data with n samples. It turns out that, in many settings, this provides a notion of "effective size" of the model class, and allows us to again get generalization bounds. This is the key necessary for the developments below.

11.2 Vapnik-Chervonenkis Theory

For the remainder of the chapter we will consider only the binary classification setting (with 0/1 loss). However, the ideas presented here can be generalized to other settings, like regression. Such generalizations are beyond the scope of this chapter. Let $\mathcal{X}$ denote the feature space (e.g., $\mathcal{X} = \mathbb{R}^d$), and $\mathcal{Y} = \{0, 1\}$ denote the label space. Let $f : \mathcal{X} \to \mathcal{Y}$ be a classifier (also called predictor or model). Let $(X, Y) \sim \mathbb{P}_{XY}$ where the joint probability distribution $\mathbb{P}_{XY}$ is generally unknown to us. We measure the classification/prediction error using the 0/1 loss function $\ell(f(X), Y) = \mathbf{1}\{f(X) \neq Y\}$, which gives rise to the risk $R(f) = \mathbb{E}[\ell(f(X), Y)] = \mathbb{P}(f(X) \neq Y)$.

Let $\mathcal{F}$ be a class of candidate models (classification rules). We would like to choose a good model (that is, a model with small probability of error). We don't have access to $\mathbb{P}_{XY}$ but only to a training sample D_n,

$$D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} \ ,$$

where $(X_i, Y_i) \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{XY}$.

Because there are only two possible labels for each feature vector each model $f \in \mathcal{F}$ will give rise to a labeling sequence

$$(f(X_1), \ldots, f(X_n)) \in \{0, 1\}^n \ .$$

For a given training set D_n there are at most 2^n distinct such sequences. However, for large n there are often much fewer possibilities. Let $\mathcal{S}(\mathcal{F}, n)$ be the maximum number of labeling sequences the class $\mathcal{F}$ induces over n training points in $\mathcal{X}$. Formally let $x_1, \ldots, x_n \in \mathcal{X}$ and define

$$N_{\mathcal{F}}(x_1, \ldots, x_n) = \{(f(x_1), \ldots, f(x_n)) \in \{0, 1\}^n, \ f \in \mathcal{F}\} \ .$$

Definition 11.2.1 (Shatter Coefficient) *The shatter coefficient of class $\mathcal{F}$ is defined as*

$$\mathcal{S}(\mathcal{F}, n) = \max_{x_1,\ldots,x_n \in \mathcal{X}} |N_{\mathcal{F}}(x_1, \ldots, x_n)| \ ,$$

where $|\cdot|$ denotes the number of elements in the set.

Clearly $\mathcal{S}(\mathcal{F}, n) \leq 2^n$, but often it is much smaller. Let's see some examples.

Example 11.2.1 *Let's revisit Example 1. Suppose we have n feature vectors $(x_1, \ldots, x_n) \in \mathbb{R}$. Assume there are no identical points (otherwise the number of possible labelings is even smaller). Any classifier in $\mathcal{F}$ labels all points to the left of a number $t \in [0, 1]$ as 0, and points to the right as 1. For $t \in [0, x_1)$, all points are either labelled "0" or "1". For $t \in (x_1, x_2)$, x_1 is labelled "0" or "1" and $x_2 \ldots x_n$ are labeled "1" or "0" and so on. We see that there are exactly $n+1$ different possible labelings, therefore $\mathcal{S}(\mathcal{F}, n) = n + 1$. This is far less than the bound $\mathcal{S}(\mathcal{F}, n) \leq 2^n$ when $n > 1$.*

The number of different labelings that a class $\mathcal{F}$ can produce on a set of n training data is a measure of the "effective size" of $\mathcal{F}$. It is possible to define a meaningful dimension concept based the behavior of $\log \mathcal{S}(\mathcal{F}, n)$.

Definition 11.2.2 *The Vapnik-Chervonenkis (VC) dimension is defined as the largest integer k such that $\mathcal{S}(\mathcal{F}, k) = 2^k$. The VC dimension of a class $\mathcal{F}$ is denoted by VC($\mathcal{F}$).*

Note that the VC dimension is not a function of the number of training data. So, for example 1 we see that $\text{VC}(\mathcal{F}) = 1$. Let's see another example

Example 11.2.2 *Let $\mathcal{X} = \mathbb{R}^2$ and define*

$$\mathcal{F} = \{f(x) = \mathbf{1}\{x \in A\} : \ A = [a, b] \times [c, d], a, b, c, d \in \mathbb{R}\} \ .$$

In words, the classifiers in $\mathcal{F}$ label all the points inside an axis-aligned rectangle $[a, b] \times [c, d]$ one, and all the other points zero. Let's see what happens when $n = 4$. Figure 11.1 illustrates this when the points are not co-linear. We see that we can obtain all the possible labelings, and so $\mathcal{S}(\mathcal{F}, 4) = 16$. Clearly for $n \leq 4$ we have also $\mathcal{S}(\mathcal{F}, n) = 2^n$. Now if we have $n = 5$ things change a bit. Figure 11.2 illustrates this. If we have five points there is always one that stays "in the middle" of all the others. So if a rectangle contains the other four points it must necessarily contain the fifth point. The proof is quite simple: choose four points such that the one with the maximum horizontal coordinate is in the set, the one with the minimum horizontal coordinate is in the set, and the same for the vertical coordinates. Clearly the fifth point not in that set is in the "middle", and so if all the four points have label 1 the fifth point must have label one as well. Therefore $\mathcal{S}(\mathcal{F}, n) < 2^n$ for $n \geq 5$. We immediately conclude that the VC dimension of this class is VC($\mathcal{F}$) = 4. Coincidentally, this is also the number of parameters you need to define an axis-aligned rectangle.

We will see more examples later on. Typically, to prove a certain class has VC dimension v you need to show two things:

(i) For $n \leq v$ there is a set of n points that can be shattered by the class, meaning $\mathcal{S}(\mathcal{F}, n) = 2^n$. This is typically the easy part - you simply need to find one arrangement of points that is shattered.

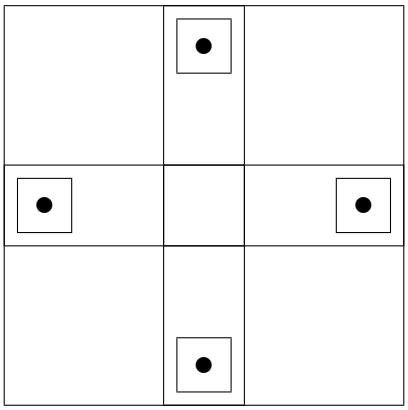


Figure 11.1: Labeling four feature vectors. If the points are not co-linear then we can obtain all possible 2^n labelings.

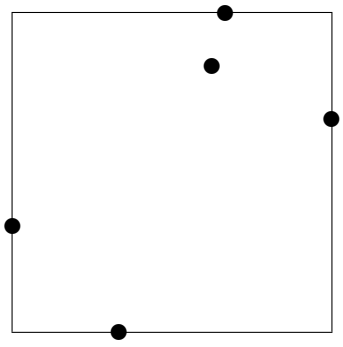


Figure 11.2: Labeling five feature vectors. Without loss of generality there is a point inside the rectangle containing all the points, that cannot have a label different than all the others, therefore $\mathcal{S}(\mathcal{F}, n) < 2^n$ for $n \geq 5$.

(ii) No set of $n > v$ points can be shattered, no matter how it is chosen. This is typically the hard part, as you need to show that there is no arrangement of the points for which $\mathcal{S}(\mathcal{F}, n) = 2^n$.

For now note that the VC dimension, and the shatter coefficient are purely combinatorial quantities. Also, note that we need to consider the set of n points that is shattered the most. This means this will typically not be a very pathological case, but rather the more general configuration of points. For now let's see what kinds of results we expect to get using this approach.

Typically, the VC dimension coincides (more or less) with the number of parameters needed to describe an element of the model class. This is, however, not true is general. The following example illustrates this in an eloquent way.

Example 11.2.3 *Consider the class*

$$\mathcal{F} = \{f : f : [0,1] \to \{0,1\}, f(x) = \mathbf{1}\{\sin(\omega x) \geq 0\}, \omega > 0\} \ .$$

One can show this class shatters n points chosen carefully, no matter how large n is (this is not immediately obvious). So the VC dimension of this class is infinite. However, the elements in the class are described by a single parameter, so, in a certain sense this class is one-dimensional.

11.3 The Shatter Coefficient and the Effective Size of a Model Class

As commented before the shatter coefficient measures the "effective" size of the class $\mathcal{F}$ when looked through the lens of n training points. Recall the generalization bound we have shown when $\mathcal{F}$ is finite

$$\begin{aligned}\mathbb{P}(\exists f \in \mathcal{F} \quad |\hat{R}_n(f) - R(f)| > \epsilon) = \mathbb{P}\left(\sup_{f\in\mathcal{F}} |\hat{R}_n(f) - R(f)| > \epsilon\right) \\ \leq 2|\mathcal{F}|e^{-2n\epsilon^2} \ .\end{aligned}$$

Our hope is that we can get something similar, but where $|\mathcal{F}|$ is replaced by $\mathcal{S}(\mathcal{F}, n)$. This is indeed the case. However, formalizing this fact is not easy, and requires very clever probabilistic arguments that allow us to show the following result.

Theorem 11.3.1 (VC inequality)

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}} |\hat{R}_n(f) - R(f)| > \epsilon\right) \leq 8\mathcal{S}(\mathcal{F}, n)e^{-n\epsilon^2/32} \ ,$$

and

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} |\hat{R}_n(f) - R(f)|\right] \leq 2\sqrt{\frac{\log \mathcal{S}(\mathcal{F}, n) + \log 2}{n}} \ .$$

The proof of the first statement is presented in the Appendix. A slightly weaker version of the second inequality can be easily derived from the first one, but a more careful and direct proof gives rise to the one in the theorem. These results are stated in terms of the shattering coefficient, but using the following result (stated without proof) one can get results in terms of the VC dimension.

Lemma 11.3.1 *Sauer's Lemma:*

$$S(\mathcal{F}, n) \leq (n+1)^{VC(\mathcal{F})} .$$

This result, together with the second inequality in the theorem, gives rise to the following important corollary that characterizes the performance of the empirical risk minimization rule.

Corollary 11.3.1 *Let*

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) ,$$

be the empirical risk minimizer (if more than one possibility is available just choose one of the possibilities). Then

$$\begin{aligned} \mathbb{E}[R(\hat{f}_n)] - \inf_{f \in \mathcal{F}} R(f) &\leq 2\sqrt{\frac{\log \mathcal{S}(\mathcal{F}, n) + \log 2}{n}} \\ &\leq 2\sqrt{\frac{VC(\mathcal{F}) \log(n+1) + \log 2}{n}} . \end{aligned}$$

Also, with probability at least $1 - \delta$ *we have*

$$R(\hat{f}_n) - \inf_{f \in \mathcal{F}} R(f) \leq 8\sqrt{2\frac{VC(\mathcal{F}) \log(n+1) + \log 8 + \log(1/\delta)}{n}} .$$

The result of the corollary is quite similar to what we have seen before, but now it applies also to uncountable classes of models.

Proof Note that, for an arbitrary (but deterministic) element of $f \in \mathcal{F}$ we have

$$\begin{aligned} \mathbb{E}\left[R(\hat{f}_n) - R(f)\right] &= \mathbb{E}\left[R(\hat{f}_n) - \hat{R}_n(f) + \hat{R}_n(f) - R(f)\right] \\ &= \mathbb{E}\left[R(\hat{f}_n) - \hat{R}_n(f)\right] \\ &\leq \mathbb{E}\left[R(\hat{f}_n) - \hat{R}_n(\hat{f}_n)\right] \\ &\leq \mathbb{E}\left[\sup_{f \in \mathcal{F}} \left|R(f) - \hat{R}_n(f)\right|\right] \\ &\leq 2\sqrt{\frac{\log \mathcal{S}(\mathcal{F}, n) + \log 2}{n}} , \end{aligned}$$

where the first inequality follows from the definition of the empirical risk minimizer, which implies that $\hat{R}_n(\hat{f}_n) \leq \hat{R}_n(f)$ for all $f \in \mathcal{F}$. Since $f \in \mathcal{F}$ is arbitrary we have just shown the first statement (together with the use of Sauer's lemma). The second part follows similarly (see exercise 11.10.6), by first showing that

$$R(\hat{f}_n) - \inf_{f \in \mathcal{F}} R(f) \leq 2 \sup_{f \in \mathcal{F}} \left|\hat{R}_n(f) - R(f)\right| .$$

□

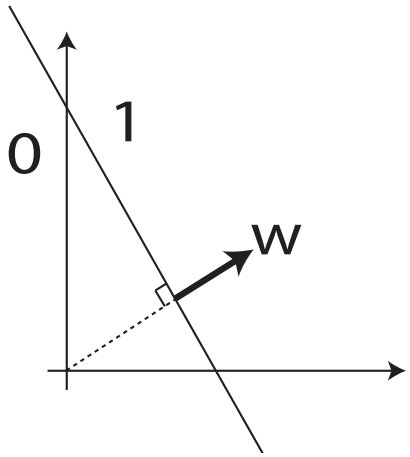


Figure 11.3: Hyperplane classifiers in two dimensions.

11.4 Linear Classifiers

Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{F}$ be the class of linear classifiers (hyperplane classifiers). Formally

$$\mathcal{F} = \left\{ f(x) = \mathbf{1}\{\boldsymbol{w}^T \boldsymbol{x} + w_0 > 0\} \; ; \; \boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^d, w_0 \in \mathbb{R} \right\} \; .$$

This class is illustrated in Figure 11.3 for $d = 2$.

Consider $d = 2$. It is clear to see that this class can shatter at least 3 points, that is, it is possible to obtain all possible labelings for a set of three points (provided these are not co-linear). Figure 11.4(a) illustrates this. Trivially then $\mathcal{S}(\mathcal{F}, n) = 2^n$ for all $n \leq 3$. Now for four points the picture changes. There is no possible arrangement of four points such that we can obtain all possible $2^4 = 16$ distinct labelings. Figure 11.4(b) illustrates this. Therefore $\mathcal{S}(\mathcal{F}, 4) < 2^4 = 16$, and so $\mathcal{S}(\mathcal{F}, n) < 2^n$ for $n > 3$. We therefore conclude that the VC dimension of $\mathcal{F}$ for $d = 2$ is $\mathrm{VC}(\mathcal{F}) = 3$.

Actually the result generalizes for d dimensions, and we have that $\mathrm{VC}(\mathcal{F}) = d+1$ (note that this is the number of parameters you need to describe such hyperplane). We can therefore apply the generalization bounds we derived before in this setting. Let $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{F}$ be the class of hyperplane classifiers. If

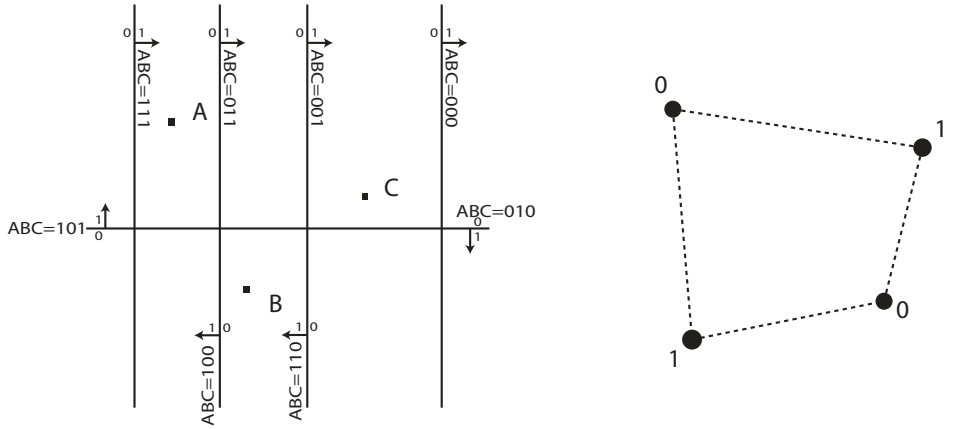
$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f) \; ,$$

then

$$\mathbb{E}\left[R(\hat{f}_n)\right] \leq \inf_{f \in \mathcal{F}} R(f) + 2\sqrt{\frac{(d+1)\log(n+1) + \log 2}{n}} \; .$$

11.5 Generalized Linear Classifiers

Normally, we have a feature vector $X \in \mathbb{R}^d$. A hyperplane in $\mathbb{R}^d$ provides a linear classifier in $\mathbb{R}^d$. Although appealing, linear classifiers are a little limited. It turns out nonlinear classifiers can be obtained using a straightforward generalization.



(a) (b)

Figure 11.4: Shattering n points using hyperplanes when $d = 2$: (a) The class $\mathcal{F}$ can shatter $n = 3$ points (in the figure you see all the hyperplanes and corresponding labelings of the points - there is a total of $2^3 = 8$ labelings); (b) Four points can never be shattered. The labeling in the figure cannot be obtained.

Let $\phi_1, \ldots, \phi_{d'}$, $d' \geq d$ be a collection of functions mapping $\mathbb{R}^d$ to $\mathbb{R}$. These functions, applied to a feature vector $X \in \mathbb{R}^d$ produce a higher dimensional feature $\phi(X) = (\phi_1(X), \phi_2(X), \ldots, \phi_{d'}(X))^T \in \mathbb{R}^{d'}$. For example, if $X = (x_1, x_2)^T$ then we could consider $d' = 5$ and $\phi = (x_1, x_2, x_1x_2, x_1^2, x_2^2)^T \in \mathbb{R}^5$. We can then construct a linear classifier in the higher dimensional generalized feature space $\mathbb{R}^{d'}$. In other words, we use an hyperplane classifier to fit the dataset

$$(\phi(X_1), Y_1), \ldots, (\phi(X_n), Y_n) \ .$$

The VC bounds immediately extend to this case, and we have for

$$\mathcal{F}' = \{f(x) = \mathbf{1}\{w^T\phi(x) + w_0 > 0\}; w \in \mathbb{R}^{d'}, w_0 \in R\} \ ,$$

$$\mathbb{E}[R(\hat{f}_n)] - \inf_{f \in \mathcal{F}'} R(f) \leq 2\sqrt{\frac{(d'+1)\log(n+1) + \log 2}{n}} \ .$$

Note that for the case of the example above (where $d = 2$ and $d' = 5$) the class $\mathcal{F}'$ consists of classification rules where the decision boundary is given by conical curves (e.g., ellipses, hyperboles, etc.).

The above way of describing the generalized linear classifiers is not particularly useful in practice. Also, solving empirical risk minimization with hyperplanes leads algorithms are too complex to be used in practice (with runtime on the order of $n^{d'}$ operations). From an algorithmic point of view, it is much better to consider the hinge-loss[1], which gives rise to Support Vector Machines (SVMs). Remarkably, when considering the corresponding optimization problem one sees that all one needs to do with the feature vectors in the algorithm is the computation of

[1]The hinge loss is the function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ where $\ell(\hat{y}, y) = \max(0, 1 - y\hat{y})$ and $\mathcal{Y} = \mathbb{R}$. This is a convex function of its arguments, which is desirable from a computational point of view. When used for binary data taking values -1 and 1 and predition rules of the form $\hat{y} = w^T x + w_0$ this gives rise to SVMs

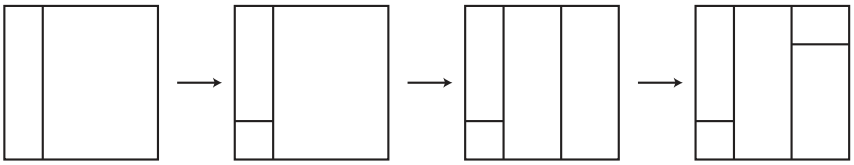


Figure 11.5: Growing trees.

inner products. It turns out one can replace the inner product by a kernel, as it is possible to show this corresponds to an inner product between transformed feature vectors in some different space (possibly high-dimensional or even infinite dimensional). So one is doing exactly the same thing as above, but without explicitly computing the higher dimensional feature space. This is a very powerful idea, known as the "kernel-trick". The type of bounds we derived above are not particularly useful in this case, but it is possible to derive more adequate results using similar ideas (these are beyond the scope of these lecture notes).

11.6 Decision Trees

Consider the following subset of the hyperplane classifiers:

$$\mathcal{F} = \{f(x) = \mathbf{1}\{x_i > b\} \text{ or } f(x) = \mathbf{1}\{x_i < b\} : i = 1, \ldots, d;\ b \in \mathbb{R}\} \ .$$

Note these are hyperplane classifiers that are aligned with the coordinate axis.

Perhaps surprisingly the VC dimension of this class is the same as the class of all hyperplanes, that is $\mathrm{VC}(\mathcal{F}) = d + 1$, although it is clear that an element of $\mathcal{F}$ can be described essentially by one real parameter and an index: b and which coordinate axis i is chosen. This further illustrates that the VC dimension might be large even for simple classes.

These kind of aligned hyperplanes can be used recursively, to create a very general class of tree classifiers. Let $k \geq 1$ and define the class of tree classifiers

$$\mathcal{T}_k = \left\{\text{classifiers based on recursive rectangular partitions of } \mathbb{R}^d \text{ with } k+1 \text{ cells}\right\} \ .$$

Let's illustrate what are the elements of $\mathcal{T}_k$. Let $T \in \mathcal{T}_k$. Each cell of T results from splitting a rectangular region into two smaller rectangles parallel to one of the coordinate axes. This process is illustrated in Figure 11.5.

Each additional split is analogous to a half-space set. Therefore each additional split can shatter at most $d+1$ points (this is not entirely obvious and a formal proof is somewhat involved). This implies that

$$\mathrm{VC}(\mathcal{T}_k) \leq (d+1)k \ .$$

This bound is a bit crude, but it is more than enough for our purposes. Notice also that if we don't restrict the number of cells in the tree then the VC dimension of the class then $\mathrm{VC}(\mathcal{T}_\infty) = \infty$ since for a set of n points we can always construct a recursive rectangular partition that has exactly one point per cell.

Let's now apply the VC bounds to trees. Let

$$\hat{f}_n = \arg\min_{f \in \mathcal{T}_k} \hat{R}_n(f) \ ,$$

then

$$\mathbb{E}\left[R(\hat{f}_n)\right] - R^* \leq \inf_{f \in \mathcal{T}_k} \{R(f) - R^*\} + 2\sqrt{\frac{k(d+1)\log(n+1) + \log 2}{n}} \ .$$

Clearly if we take k large we can make $\inf_{f \in \mathcal{T}_k}\{R(f) - R*\}$ small, but this will make the estimation error (bounded by the square-root term) large. How can we decide what dimension to choose for a generalized linear classifier? How many leafs should be used for a classification tree? The answer is complexity regularization using VC bounds! The approach is in all identical to what we have done before, but now we don't need to restrict ourselves to finite classes of models anymore.

11.7 Structural Risk Minimization (SRM)

SRM is simply complexity regularization using VC type bounds in place of the Hoeffding's inequality we used before. Let's derive a simple version of a SRM result.

Assume you have a sequence of sets of classifiers

$$\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k, \ldots \ ,$$

of increasing VC dimension

$$\mathrm{VC}(\mathcal{F}_1) \leq \mathrm{VC}(\mathcal{F}_2) \leq \ldots \ .$$

In the typical cases we want to consider the inequalities above are strict, so that is $\mathrm{VC}(\mathcal{F}_k) < \mathrm{VC}(\mathcal{F}_{k+1})$ for all $k \geq 1$. For each $k = 1, 2, \ldots$ we find the minimum empirical risk classifier

$$\hat{f}_n^{(k)} = \arg\min_{f \in \mathcal{F}_k} \hat{R}_n(f)$$

and then select the final classifier according to

$$\hat{k} = \arg\min_{k \geq 1} \left\{ \hat{R}_n(\hat{f}_n^{(k)}) + \mathrm{penalty}(k, n) \right\} \ .$$

Finally we take $\hat{f}_n \equiv \hat{f}_n^{(\hat{k})}$.

The basic rational is to choose $\mathrm{penalty}(k, n)$ as an increasing function of k, so that we choose models that are simple, but also "explain" the training data well. The choice of the penalty function can be made quite obvious by looking at our VC bounds.

Begin by writing the result of Theorem 11.3.1 in a slightly different way. Let $\delta_k > 0$. Then

$$\mathbb{P}\left(\sup_{f \in \mathcal{F}_k} \left| \hat{R}_n(f) - R(f) \right| > 8\sqrt{\frac{\mathcal{S}(\mathcal{F}_k, n) + \log 8 + \log(1/\delta_k)}{2n}} \right) \leq \delta_k \ .$$

Sauer's lemma implies that

$$\mathbb{P}\left(\sup_{f \in \mathcal{F}_k} \left| \hat{R}_n(f) - R(f) \right| > 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_k)\log(n+1) + 3 + \log(1/\delta_k)}{2n}} \right) \leq \delta_k \ .$$

Now define $\mathcal{F} = \bigcup_{k=1}^{\infty} \mathcal{F}_k$ and take $\delta_k = \delta 2^{-k}$. Let's see how big is the difference between empirical and true risk for ANY element of $\mathcal{F}$. In other words, we want to see how big can

$$|\hat{R}_n(f) - R(f)| \ ,$$

be for any $f \in \mathcal{F}$.

Let $k(f)$ be the smallest integer k such that $f \in \mathcal{F}_k$. Now note that

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left\{\left|\hat{R}_n(f) - R(f)\right| - 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_{k(f)})\log(n+1) + 3 + \log(1/\delta_{k(f)})}{2n}}\right\} > 0\right) \\
&= \mathbb{P}\left(\sup_{k}\left\{\sup_{f\in\mathcal{F}_k}\left\{\left|\hat{R}_n(f) - R(f)\right| - 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_k)\log(n+1) + 3 + \log(1/\delta_k)}{2n}}\right\}\right\} > 0\right) \\
&= \mathbb{P}\left(\bigcup_{k=1}^{\infty}\left\{\sup_{f\in\mathcal{F}_k}\left\{\left|\hat{R}_n(f) - R(f)\right| - 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_k)\log(n+1) + 3 + \log(1/\delta_k)}{2n}}\right\} > 0\right\}\right) \\
&= \mathbb{P}\left(\bigcup_{k=1}^{\infty}\left\{\sup_{f\in\mathcal{F}_k}\left|\hat{R}_n(f) - R(f)\right| > 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_k)\log(n+1) + 3 + \log(1/\delta_k)}{2n}}\right\}\right) \\
&\leq \sum_{k=1}^{\infty}\mathbb{P}\left(\sup_{f\in\mathcal{F}_k}\left|\hat{R}_n(f) - R(f)\right| > 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_k)\log(n+1) + 3 + \log(1/\delta_k)}{2n}}\right) \\
&\leq \sum_{k=1}^{\infty}\delta_k = \sum_{k=1}^{\infty}\delta 2^{-k} = \delta \ ,
\end{aligned}$$

where the first inequality is just a consequence of the union of events bound. In other words, with probability at least $1 - \delta$

$$\forall f \in \mathcal{F}, \quad \left|\hat{R}_n(f) - R(f)\right| \leq \underbrace{8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_{k(f)})\log(n+1) + 3 + \log(1/\delta_{k(f)})}{2n}}}_{C(f,n,\delta)} \ ,$$

where $k(f)$ is the smallest k such that $f \in \mathcal{F}_k$. Note that

$$C(f,n,\delta) = 8\sqrt{\frac{\mathrm{VC}(\mathcal{F}_{k(f)})\log(n+1) + 3 + k(f)\log 2 + \log(1/\delta)}{2n}} \ .$$

We are now under the setting of Chapter 8, and it seems quite sensible to take

$$\hat{f}_n \equiv \arg\min_{f\in\mathcal{F}} \hat{R}_n(f) + C(f,n,\delta) \ .$$

Using the results of that chapter we get immediately that

$$\mathbb{E}[R(\hat{f}_n)] - R^* \leq \inf_{f\in\mathcal{F}}\{R(f) - R^* + C(f,n,\delta)\} + \delta \ ,$$

so taking $\delta = 1/\sqrt{n+1}$ is a very sensible choice, yielding the bound.

$$\begin{aligned}
&\mathbb{E}[R(\hat{f}_n)] - R^* \\
&\leq \inf_{f\in\mathcal{F}}\left\{R(f) - R^* + 8\sqrt{\frac{(\mathrm{VC}(\mathcal{F}_{k(f)}) + 1/2)\log(n+1) + 3 + k(f)\log 2}{2n}}\right\} + \frac{1}{\sqrt{n+1}} \ .
\end{aligned}$$

11.8 Application to Trees

Let $\mathcal{T}_1, \mathcal{T}_2, \ldots$ be the classes of decision trees with $k+1$ leafs. Then $\mathrm{VC}(\mathcal{T}_k) \leq k(d+1)$. Now define

$$\hat{f}_n^{(k)} = \arg\min_{f \in \mathcal{T}_k} \hat{R}_n(f)$$

and then select the final classifier according to

$$\hat{k} = \arg\min_{k \geq 1} \left\{ \hat{R}_n(\hat{f}_n^{(k)}) + 8\sqrt{\frac{(k(d+1)+1/2)\log(n+1)+3+k\log 2}{2n}} \right\} .$$

Finally we take $\hat{f}_n \equiv \hat{f}_n^{(\hat{k})}$. This yields the bound

$$\begin{aligned} &\mathbb{E}[R(\hat{f}_n)] - R^* \\ &\leq \inf_k \left\{ \inf_{f \in \mathcal{F}_k} \left\{ R(f) - R^* + 8\sqrt{\frac{(k(d+1)+1/2)\log(n+1)+3+k\log 2}{2n}} \right\} \right\} + \frac{1}{\sqrt{n+1}} . \end{aligned}$$

Compare this with the bound of Chapter 9, where we considered recursive dyadic partitions with a much more stringent structure. You see that bound is has essentially the same form, although we are now considering a much richer class of classification rules, that is uncountable. These kinds of trees are used frequently for classification and regression (see for example the methods under the name of CART), and are quite useful in practice.

11.9 Appendix: Proof of Theorem 11.3.1

We will prove the first inequality in Theorem 11.3.1. A slightly weaker version of the second inequality can be easily derived from the first one, but a more careful and direct proof gives rise to the one stated in the theorem.

We will follow closely the approach presented in the book Devroye et al. (1996). The proof of this result is rather involved, but it can be broken into several important steps. The main idea is to get to a point where we can take advantage of the effective size of the class induced by the training data. There are different ways to do this (for instance, using a chaining argument), but here we will make use of a "ghost sample", that is, another sequence of data in all identical to the training data D_n. This ghost sample helps us developing the proof, but it doesn't play a role in the end result (that is, you don't need a ghost sample to apply the result). Let $D'_n = \{(X'_1, Y'_1), \ldots, (X'_n, Y'_n)\}$ be a set of random variables independent of D_n, such that $(X'_i, Y'_i) \overset{i.i.d.}{\sim} \mathbb{P}_{XY}$. This is called the ghost sample. Define also the empirical risk under this sample by

$$\hat{R}'_n(f) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{f(X'_i) \neq Y'_i\} .$$

For the rest of the proof we will assume that $n\epsilon^2 \geq 2$ without loss of generality, since otherwise the bound stated in the theorem is trivial.

Step 1: (First symmetrization by a ghost sample):

We are going to show that

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-R(f)\right|>\epsilon\right)\leq 2\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-\hat{R}'_n(f)\right|>\frac{\epsilon}{2}\right)\ . \tag{11.1}$$

Notice that the absolute value term on the right-hand-side is now symmetric, involving two different empirical risks. Begin by defining $\tilde{f}(D_n)\equiv\tilde{f}$ to be an element of $\mathcal{F}$ such that $\left|\hat{R}_n(f)-R(f)\right|>\epsilon$ if such element exists, otherwise $\tilde{f}$ is an arbitrary element of $\mathcal{F}$. You can informally think of $\tilde{f}$ as

$$\tilde{f}\approx\arg\max_{f\in\mathcal{F}}\left|\hat{R}_n(f)-R(f)\right|\ ,$$

although this is not entirely well defined. Notice that $\tilde{f}$ is a function of D_n (we dropped the explicit dependence to make the presentation cleaner).

Now let's look at the right-hand-side of (11.1).

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-\hat{R}'_n(f)\right|>\frac{\epsilon}{2}\right)\\
&\quad\geq\ \mathbb{P}\left(\left|\hat{R}_n(\tilde{f})-\hat{R}'_n(\tilde{f})\right|>\frac{\epsilon}{2}\right)\\
&\quad\geq\ \mathbb{P}\left(\left|\hat{R}_n(\tilde{f})-R(\tilde{f})\right|>\epsilon\text{ and }\left|\hat{R}'_n(\tilde{f})-R(\tilde{f})\right|<\frac{\epsilon}{2}\right)\\
&\quad=\ \mathbb{E}\left[\mathbf{1}\{\left|\hat{R}_n(\tilde{f})-R(\tilde{f})\right|>\epsilon\}\mathbf{1}\{\left|\hat{R}'_n(\tilde{f})-R(\tilde{f})\right|<\frac{\epsilon}{2}\}\right]\\
&\quad=\ \mathbb{E}\left[\mathbf{1}\{\left|\hat{R}_n(\tilde{f})-R(\tilde{f})\right|>\epsilon\}\mathbb{E}\left[\mathbf{1}\{\left|\hat{R}'_n(\tilde{f})-R(\tilde{f})\right|<\frac{\epsilon}{2}\}\Big|\,D_n\right]\right]\\
&\quad=\ \mathbb{E}\left[\mathbf{1}\{\left|\hat{R}_n(\tilde{f})-R(\tilde{f})\right|>\epsilon\}\mathbb{P}\left(\left|\hat{R}'_n(\tilde{f})-R(\tilde{f})\right|<\frac{\epsilon}{2}\Big|\,D_n\right)\right]\ ,
\end{aligned}$$

where the second inequality follows from the fact that, for any reals x, y and z,

$$|x-z|>\epsilon\text{ and }|y-z|\leq\epsilon/2\quad\Rightarrow\quad|x-y|\geq\epsilon/2\ .$$

Now, conditionally on D_n we see that

$$\hat{R}'_n(\tilde{f})-R(\tilde{f})=\frac{1}{n}\sum_{i=1}^{n}U_i\ ,$$

where $U_i=\mathbf{1}\{\tilde{f}(X'_i)\neq Y'_i\}-E\left[\mathbf{1}\tilde{f}(X'_i)\neq Y'_i\Big|\,D_n\right]$ are zero-mean i.i.d. random variables. We are in good shape to use a concentration inequality here. For our purposes Chebyshev's

inequality suffices.

$$\begin{aligned}
\mathbb{P}\left(\left|\hat{R}'_n(\tilde{f}) - R(\tilde{f})\right| < \frac{\epsilon}{2}\middle| D_n\right) &= \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n U_i\right| < \frac{\epsilon}{2}\middle| D_n\right) \\
&= \mathbb{P}\left(\left|\sum_{i=1}^n U_i\right| < \frac{n\epsilon}{2}\middle| D_n\right) \\
&\geq 1 - \frac{4}{n^2\epsilon^2}\mathrm{Var}\left(\sum_{i=1}^n U_i\middle| D_n\right) \\
&= 1 - \frac{4}{n^2\epsilon^2} n\mathrm{Var}\left(U_i| D_n\right) \\
&\geq 1 - \frac{4}{n\epsilon^2}\frac{1}{4} = 1 - \frac{1}{n\epsilon^2} \geq \frac{1}{2}\ ,
\end{aligned}$$

since we assumed that $n\epsilon^2 \geq 2$. Finally

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f) - \hat{R}'_n(f)\right| > \frac{\epsilon}{2}\right) \\
&\geq \mathbb{E}\left[\mathbf{1}\{\left|\hat{R}_n(\tilde{f}) - R(\tilde{f})\right| > \epsilon\}\mathbb{P}\left(\left|\hat{R}'_n(\tilde{f}) - R(\tilde{f})\right| < \frac{\epsilon}{2}\middle| D_n\right)\right] \\
&\geq \frac{1}{2}\mathbb{E}\left[\mathbf{1}\{\left|\hat{R}_n(\tilde{f}) - R(\tilde{f})\right| > \epsilon\}\right] \\
&= \frac{1}{2}\mathbb{P}\left(\left|\hat{R}_n(\tilde{f}) - R(\tilde{f})\right| > \epsilon\right) \\
&= \frac{1}{2}\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f) - R(f)\right| > \epsilon\right)\ ,
\end{aligned}$$

concluding the proof of (11.1), where the last step follows from the definition of $\tilde{f}$.

Step 2: (Symmetrization by random signs):

Let's rewrite the right-hand-side of (11.1).

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f) - \hat{R}'_n(f)\right| > \frac{\epsilon}{2}\right) = \mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \mathbf{1}\{f(X_i)\neq Y_i\} - \mathbf{1}\{f(X'_i)\neq Y'_i\}\right| > \frac{\epsilon}{2}\right)\ .$$

Note that $\mathbf{1}\{f(X_i)\neq Y_i\}$ and $\mathbf{1}\{f(X'_i)\neq Y'_i\}$ have the same distribution, and therefore $\mathbf{1}\{f(X_i)\neq Y_i\} - \mathbf{1}\{f(X'_i)\neq Y'_i\}$ has zero mean and a symmetric distribution[2]. So if we randomly permute the signs inside the absolute value term we won't change the probability. Let's introduce another"ghost sample"-like sequence.

Let $\sigma_1,\ldots,\sigma_n$ be i.i.d. random variables, independent of D_n and D'_n, such that $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = 1/2$. These are called Rademacher random variables. In light of our remarks

[2]A zero-mean random variable Z has a symmetric distribution if Z and $-Z$ have the same distribution.

above

$$
\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-\hat{R}'_n(f)\right|>\frac{\epsilon}{2}\right)\\
&= \mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \mathbf{1}\{f(X_i)\neq Y_i\}-\mathbf{1}\{f(X'_i)\neq Y'_i\}\right|>\frac{\epsilon}{2}\right)\\
&= \mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i\left(\mathbf{1}\{f(X_i)\neq Y_i\}-\mathbf{1}\{f(X'_i)\neq Y'_i\}\right)\right|>\frac{\epsilon}{2}\right)\\
&\leq \mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\text{ or }\right.\\
&\qquad \left.\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i\mathbf{1}\{f(X'_i)\neq Y'_i\}\right|>\frac{\epsilon}{4}\right)\\
&\leq 2\mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\right)\ ,
\end{aligned}
$$

where the last inequality follows simply from a union bound over the two events of the previous line. So in these two steps we have shown that

$$
\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-R(f)\right|>\epsilon\right)\leq 4\mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\right)\ , \tag{11.2}
$$

Note that we managed to reduce the expression in the left-hand-side of the above to a bound on the sum of i.i.d. zero-mean random variables. We also eliminated the dependence of the ghost-sample D'_n we had at the end of step 1. We are now ready to take advantage of the effective size of the $\mathcal{F}$ with respect to D_n, which will be the next step.

Step 3: (Conditioning on D_n):

This step is conceptually the same we used in all the generalization bounds we derived in the course so far. We are going to perform a union bound over all the models under consideration. The difference here is that this set in no longer the entire class $\mathcal{F}$, but instead just a finite subset of it.

Let $x_1,\ldots,x_n\in\mathcal{X}$ and $y_1,\ldots,y_n\in\mathcal{Y}$ be arbitrary sequences. Let's examine the quantity

$$
\frac{1}{n}\left|\sum_{i=1}^n \sigma_i\mathbf{1}\{f(x_i)\neq y_i\}\right|\ ,
$$

where the randomness is solely on the random signs σ_i. We have observed in the previous lecture that the sequence $(f(x_1),\ldots,f(x_n))$ can take at most $\mathcal{S}(\mathcal{F},n)$ different values, therefore

$$
(\mathbf{1}\{f(x_1)\neq y_1\},\ldots,\mathbf{1}\{f(x_n)\neq y_n\})\ ,
$$

can take at most $\mathcal{S}(\mathcal{F},n)$ different values. Let $\mathcal{F}(x_1,\ldots,x_n)\subseteq\mathcal{F}$ be the smallest subset of $\mathcal{F}$ such that

$$
N_{\mathcal{F}}(x_1,\ldots,x_n)=N_{\mathcal{F}(x_1,\ldots,x_n)}(x_1,\ldots,x_n)
$$

where as before $N_{\mathcal{F}}(x_1,\ldots,x_n) = \{(f(x_1),\ldots,f(x_n)) \in \{0,1\}^n,\ f \in \mathcal{F}\}$. In words $\mathcal{F}(x_1,\ldots,x_n)$ is the smallest subset of $\mathcal{F}$ that gives rise to all the different prediction rules for the data $(x_1,y_1),\ldots,(x_n,y_n)$, therefore $|\mathcal{F}(x_1,\ldots,x_n)| \leq \mathcal{S}(\mathcal{F},n)$.

We are essentially ready to apply our union bound.

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}} \frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right) \\
&= \mathbb{P}\left(\max_{f\in\mathcal{F}(x_1,\ldots,x_n)} \frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right) \\
&= \mathbb{P}\left(\bigcup_{f\in\mathcal{F}(x_1,\ldots,x_n)} \left\{\frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right\}\right) \\
&\leq \sum_{f\in\mathcal{F}(x_1,\ldots,x_n)} \mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right) \\
&\leq |\mathcal{F}(x_1,\ldots,x_n)| \sup_{f\in\mathcal{F}(x_1,\ldots,x_n)} \mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right) \\
&\leq \mathcal{S}(\mathcal{F},n) \sup_{f\in\mathcal{F}(x_1,\ldots,x_n)} \mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right) \\
&\leq \mathcal{S}(\mathcal{F},n) \sup_{f\in\mathcal{F}} \mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n \sigma_i \mathbf{1}\{f(x_i)\neq y_i\}\right| > \frac{\epsilon}{4}\right) \ .
\end{aligned}$$

Therefore we managed to "pull" the supremum outside the probability, and take advantage of the effective size of the class $\mathcal{F}$. We are one step away from concluding the proof.

Step 4: (Hoeffding's inequality):

Notice that

$$\frac{1}{n}\left|\sum_{i=1}^n \underbrace{\sigma_i \mathbf{1} f(x_i)\neq y_i}_{A_i}\right|$$

is the absolute value of the sum of n independent random variables A_i, with zero mean and bounded between -1 and 1. We can therefore apply Hoeffding's inequality.

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^n A_i\right| > \epsilon/4\right) &\leq \mathbb{P}\left(\left|\sum_{i=1}^n A_i\right| > n\epsilon/4\right) \\
&\leq 2e^{-\frac{2(n\epsilon/4)^2}{\sum_{i=1}^n (\max_i A_i - \min_i A_i)^2}} \\
&\leq 2e^{-\frac{n^2\epsilon^2/8}{4n}} \\
&\leq 2e^{-\frac{n\epsilon^2}{32}} \ .
\end{aligned}$$

It's time to revisit (11.2). Let's look at the right-hand-side

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}f(X_i)\neq Y_i\right|>\frac{\epsilon}{4}\right)\\
&= \mathbb{E}\left[\mathbf{1}\{\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}f(X_i)\neq Y_i\right|>\frac{\epsilon}{4}\}\right]\\
&= \mathbb{E}\left[\mathbb{E}\left[\mathbf{1}\{\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}f(X_i)\neq Y_i\right|>\frac{\epsilon}{4}\}\middle| D_n\right]\right]\\
&= \mathbb{E}\left[\mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\middle| D_n\right)\right]\\
&\leq \mathbb{E}\left[\mathcal{S}(\mathcal{F},n)\sup_{f\in\mathcal{F}}\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\middle| D_n\right)\right]\\
&\leq \mathcal{S}(\mathcal{F},n)\mathbb{E}\left[\sup_{f\in\mathcal{F}}\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\middle| D_n\right)\right]\\
&\leq \mathcal{S}(\mathcal{F},n)\mathbb{E}\left[2e^{-\frac{n\epsilon^2}{32}}\middle| D_n\right]\\
&= 2\mathcal{S}(\mathcal{F},n)e^{-\frac{n\epsilon^2}{32}}\ .
\end{aligned}$$

Using this in step (11.2) yields the desired result

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{R}_n(f)-R(f)\right|>\epsilon\right)\\
&\leq 4\mathbb{P}\left(\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\mathbf{1}\{f(X_i)\neq Y_i\}\right|>\frac{\epsilon}{4}\right)\\
&\leq 8\mathcal{S}(\mathcal{F},n)e^{-\frac{n\epsilon^2}{32}}\ ,
\end{aligned}$$

concluding the proof.

11.10 Exercises

Exercise 11.10.1 *Let $\mathcal{F}$ be a finite class of models. Show that $VC(\mathcal{F}) \leq \log_2|\mathcal{F}|$. Give an example of a class of size 8 for which this result is tight.*

Exercise 11.10.2 *Consider the setting of binary classification (with zero-one loss) and the following class of classifiers*

$$\mathcal{F}=\{f:\mathbb{R}\to\{0,1\}:f(x)=\mathbf{1}\{x\in[-a,a]\cup[b,\infty)\},0\leq a\leq b\in\mathbb{R}\}\ .$$

In other words, the feature space $\mathcal{X}$ is the real line, and this is the class of indicators of sets of the form $[-a,a]\cup[b,\infty)$ with $0\leq a\leq b$ arbitrary.

a) What is $VC(\mathcal{F})$, the VC dimension of $\mathcal{F}$? Carefully justify your answer.

b) Consider now a small modification of the above - define the class

$$\mathcal{G} = \{f : \mathbb{R} \to \{0,1\} : f(x) = \mathbf{1}\{x \in [a,b] \cup [c,\infty)\}, a \leq b \leq c \in \mathbb{R}\} \ .$$

What is $VC(\mathcal{G})$? Carefully justify your answer.

Exercise 11.10.3 *Consider the class of circles in the plane, that is*

$$\mathcal{F} = \left\{f : \ f(x,y) = \mathbf{1}\{(x-a)^2 + (y-b)^2 \leq r^2\} \ a,b,r \in \mathbb{R}\right\} \ .$$

What is the VC dimension of this class?

Remark: *Recall you must show that*

(i) For $n = VC(\mathcal{F})$ there is a set of n points that can be shattered by the class.

(ii) No set of $n > VC(\mathcal{F})$ points can be shattered.

IMPORTANT: *you may use the following geometry result without proof: two different circles can intersect in at most two distinct points. This implies that the symmetric difference of two disks consists of at most two connected components.*

Exercise 11.10.4 *In this chapter we have shown that for the class of classifiers that are indicators of axis-aligned rectangles in the plane the VD dimension is 4. In part (b) of this exercise we will generalize this a bit and consider consider now arbitrary k-sided convex polygons in the plane.*

a) Compute the VC dimension for the class of arbitrary k-sided convex polygons when $k = 3$ (this is the class of triangles).

b) Compute the VC dimension for the class of arbitrary k-sided convex polygons when $k = 4$.

c) How does the result generalize for arbitrary $k \geq 3$? Fully justify your answers.

Exercise 11.10.5 (This is a harder one - I don't the precise answer) *Consider the class of rectangles that are not necessarily aligned with the axis. What is the VC dimension of this class?*

Exercise 11.10.6 *In this exercise you'll get a feeling about the quality of the VC bounds. For the entire exercise we are considering the setting of binary classification with the $0/1$ loss.*

a) Using the first part of Theorem 11.3.1 show that, for $\delta > 0$

$$\mathbb{P}\left(R(\hat{f}_n) - \inf_{f \in \mathcal{F}} R(f) \leq 8\sqrt{2\frac{VC(\mathcal{F})\log(n+1) + \log 8 + \log(1/\delta)}{n}}\right) \geq 1 - \delta \ .$$

For the rest of the question consider the scenario where $\mathcal{X} = [0,1]$ and

$$\mathcal{F} = \{f : f(x) = \mathbf{1}\{x \geq t\}, t \in [0,1]\} \ .$$

We have seen earlier that $VC(\mathcal{F}) = 1$. Furthermore, assume that there is a value $0 < C \leq 9$ such that for any set $A \subseteq [0,1]$ we have $\mathbb{P}(X \in A) \leq C\,Vol(A)$.

b) Take $\delta = 0.1$. For which value of n does the VC bound of (a) becomes non-trivial, meaning it tells you that with probability at least $1-\delta$ $R(\hat{f}_n) - \inf_{f\in\mathcal{F}} R(f)$ is smaller than one? ***note:*** *you might need to find the answer by trial and error.*

c) Now consider instead the quantized version of $\mathcal{F}$, as defined in Example 11.1.1. This is a finite class, therefore we can use the bounds we derived earlier in class, namely

$$\mathbb{P}\left(R(\hat{f}_n^Q) \leq \min_{f\in\mathcal{F}_Q} R(f) + 2\sqrt{\frac{\log(1+|\mathcal{F}_Q|)+\log(1/\delta)}{2n}} \right) \geq 1-\delta \ ,$$

where $\hat{f}_n^Q$ is the empirical risk minimizer over $\mathcal{F}_Q$. Use this result to get a high probability upper bound on

$$R(\hat{f}_n^Q) - \inf_{f\in\mathcal{F}} R(f) \ .$$

d) Take $\delta = 0.1$ and $Q = n$. For which value of n does the bound you derived in (d) becomes non-trivial?

e) For the rest of the problem consider the following distribution $\mathbb{P}_{XY}$ of the training data. The feature X is uniformly distributed over $[0,1]$ and $\mathbb{P}(Y=1|X=x) = \eta(x)$, where

$$\eta(x) = \frac{x+2/3}{2} \ .$$

What is the Bayes' classifier f^*? Verify that this classifier is an element of $\mathcal{F}$.

f) Let $f(x) = \mathbf{1}\{x \geq t\}$. Check that the excess risk is given by

$$R(f) - R(f^*) = \frac{(t-1/3)^2}{2} \ .$$

g) Generate n datapoints for the above distribution, for various values of n. Compute the empirical risk minimizers over $\mathcal{F}$ and $\mathcal{F}_Q$ and the corresponding excess risk relative to $\mathcal{F}$, that is $R(\hat{f}_n) - R(f^*)$ and $R(\hat{f}_n^Q) - \mathbb{R}(f^*)$. How do these compare with bounds derived above?

Remark: *This exercise shows that the VC bounds are extremely loose, and cannot be used to determine how many samples to take to get a certain performance. Nevertheless they help us devise good algorithmic approaches, and give non-empirical guarantees to such algorithms.*

Chapter 12

Denoising of Piecewise Smooth Functions

From Chapter 5 onwards we've been considering general bounded loss functions, and in particular the scenario of binary classification and the 0/1 loss function later on. For the setting of binary classification with this loss function the type of results we have been producing is essentially the best one can hope for. However, if one considers the square loss function the bounds we derived turn out to be not so good. Recall that the square loss is defined as $\ell(y_1, y_2) = (y_1 - y_2)^2$. If the two arguments y_1 and y_2 are close then we incur a very small loss. For instance, if $y_1 = 0.1$ and $y_2 = 0.2$ the loss value is only 0.01. This means that we pay very little for small errors. In this chapter we will consider results for the square loss. These are going to be in a very specific setting, particularly relevant for signal and image processing, and a bit different than the learning setting we considered so far.

12.1 Noisy observations

Let $f : \mathcal{X} \to \mathbb{R}$ be a function representing a signal of interest. For instance $\mathcal{X} = [0, 1]^d$, and for $d = 1$ this might be an audio signal, and for $d = 2$ this might represent an image. This is the main object of our interest. Unfortunately we cannot observe this directly, but rather we observe only noisy samples of this function. In particular, we assume that the function is observed on a discrete grid (for $d = 2$ this might correspond precisely to the pixels in a CCD). More specifically, we observe

$$Y_i = f^*(x_i) + W_i \ , \quad i = 1, \ldots, n \ ,$$

where W_i are assumed to be i.i.d. normal random variables with zero mean and variance σ^2, and $x_i \in \mathcal{X}$ are the points where we collect the observations (at this point we won't make any assumptions about these). This is approximately the setting we considered in Chapter 4.

12.2 Estimation

We take a penalized empirical risk approach. Let $\mathcal{F}$ be a class of candidate models. Formally $\mathcal{F}$ is some collection of functions of the form $f : \mathcal{X} \to \mathbb{R}$. We assume this class is countable, and

that there is a map $c(f) : \mathcal{F} \to \mathbb{R}$ such that

$$\sum_{f \in \mathcal{F}} 2^{-c(f)} \leq 1 \ .$$

As seen before, this map can be constructed by devising a prefix code for the elements of $\mathcal{F}$ and setting $c(f)$ as the codeword length associated with the element $f \in \mathcal{F}$. With this in hand, define the following estimator of f^*.

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(x_i))^2 + \frac{8\sigma^2 c(f) \log 2}{n} \right\} \ .$$

Note that the first term in the r.h.s. is essentially the empirical risk of f, and the second term is a penalization. Furthermore, note that the form of the penalization is a bit different than that we had for the general bounded losses. In the latter case the penalization was of the form $\sim \sqrt{c(f)/n}$ instead. The reason for this penalization has to do the with fact that we are considering the square loss.

Although in principle we would like to estimate the function f^* over $\mathcal{X}$ we are going to make a small compromise, and be interested in only estimating this function over the points $\{x_i : i = 1, \ldots, n\}$. It is possible to show the following result[1]

Proposition 12.2.1 *Under the above assumptions, the penalized estimator $\hat{f}_n$ satisfies*

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[(\hat{f}_n(x_i) - f^*(x_i))^2 \right] \ \leq \ \inf_{f \in \mathcal{F}} \left\{ \frac{2}{n} \sum_{i=1}^{n} (f(x_i) - f^*(x_i))^2 \ + \ \frac{8\sigma^2 c(f) \log 2}{n} \right\} \ .$$

This is an oracle bound, similar to the ones we had derived before for the general losses. However, we see that the second term on the r.h.s., which corresponds to a bound on the estimation error, is a bit different, and behaves like $\sim c(f)/n$ instead of $\sim \sqrt{c(f)/n}$. This has to do with the nice properties of the square loss. Actually, in many cases $c(f)$ corresponds to the number of parameters in the model f that must be estimated from data - if that is the case then this term is exactly proportional to the variance of the estimator.

Note furthermore that we are not controlling $\int_{\mathcal{X}} (\hat{f}_n(t) - f^*(t))^2 dt$, as we did on Chapter 4, but are rather controlling only the deviation between $\hat{f}_n$ and f^* on the points we are sampling the function. In many settings this might be what we want to do (e.g., when removing noise from a digital image we want to obtain another digital image with the same resolution, but with less noise). There are situations, however, where we want more. For instance, if we want to construct a high-resolution image from low-resolution observations - this is very important for applications in astronomy. We won't delve on these issues further in this chapter, but there are ways to get similar results for $\int_{\mathcal{X}} (\hat{f}_n(t) - f^*(t))^2 dt$ in many settings.

12.3 Piecewise Smooth Functions

For simplicity, let's consider a one dimensional setting and take $\mathcal{X} = [0, 1]$. Furthermore let's take $x_i = i/n$, so that we are in a setting like that of Chapter 4. In that chapter we showed

[1] see example 3 in http://www.win.tue.nl/~rmcastro/6887_10/files/lecture13.pdf.

that, if f^* is a Lipschitz function, we can estimate it to a certain accuracy from these noisy observations. It can be shown that the result we showed implies that, if f^* is Lipschitz with constant $L > 0$ then

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\hat{f}_n(x_i) - f^*(x_i))^2\right] = O\left(n^{-2/3}\right) \ .$$

This is actually the best one can hope for with Lipschitz functions. In the exercises of Chapter 4 we also considered the class of piecewise Lipschitz functions, and showed that the estimator of Chapter 4 would have a "bad" performance for that class, namely $\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\hat{f}_n(x_i) - f^*(x_i))^2\right] = O\left(n^{-1/2}\right)$. It turns out we can consider a different type of estimator that does much better. Actually it performs almost as well as if the function was Lipschitz, as we will see.

Define the class $\mathcal{B}(L, M, R)$ of piecewise Lipschitz functions as

$$\mathcal{B}(L, M, R) = \{f : [0,1] \to [-R, R] : \forall x, y \in I_m \ |f(x) - f(y)| \leq L|x - y|, m = 1, \ldots, M\} \ ,$$

where $I_1, \ldots, I_M$ are intervals that partition $[0, 1]$. This is the class of functions taking values in $[-R, R]$ that are Lipschitz in at most M intervals given by $I_1, \ldots, I_M$. You can think of this class as a toy model for "one-dimensional" images, where the discontinuities correspond to edges.

The next step is to construct of suitable class of models $\mathcal{F}$, and a corresponding function $c(\cdot)$. This choice should be such that the r.h.s. of the result in Proposition 12.2.1 is somewhat small. We saw that Lipschitz functions are well approximated by piecewise constant functions, therefore it is sensible to consider in our class of models piecewise constant functions. However, we will use trees to partition $[0, 1]$, instead of the regular partition we used in Chapter 4.

12.3.1 Recursive Dyadic Partitions

We discussed this idea already in our examination of classification trees. The main idea is illustrated in Figure 12.1.

Lemma 12.3.1 *Let $f^* \in \mathcal{B}(L, M, R)$, and let $d_1, \ldots, d_{M-1}$ denote the endpoints of the corresponding Lipschitz pieces of f^* (call these boundary points). There exists a RDP with at most $2^{j_{\min}} + (M-1)(j_{\max} - j_{\min})$ elements so that all leafs are at depth at least $j_{\min}$ (so have length $2^{-j_{\max}}$) and the boundary points are contained in partition elements of length $2^{-j_{\max}}$ (these correspond to leafs at depth $j_{\max}$).*

Proof Start with the RDP described by the complete binary tree with $2^{j_{\max}}$ leafs, and prune any leaf that is at a level deeper than $j_{\min}$ and that does not contain a boundary point. At level $i \leq j_{\min}$ there are exactly 2^i nodes. At any level $i > j_{\min}$ there are at most $M - 1$ nodes containing boundary points. These must be present in the tree, as well as their siblings. So, the pruned tree has at most

$$\left(\sum_{i=0}^{j_{\min}} 2^i\right) \ + \ 2(M-1)(j_{\max} - j_{\min}) = 2^{j_{\min}+1} - 1 + 2(M-1)(j_{\max} - j_{\min})$$

nodes. This means it has at most $2^{j_{\min}} + (M-1)(j_{\max} - j_{\min})$ leafs, concluding the proof. □

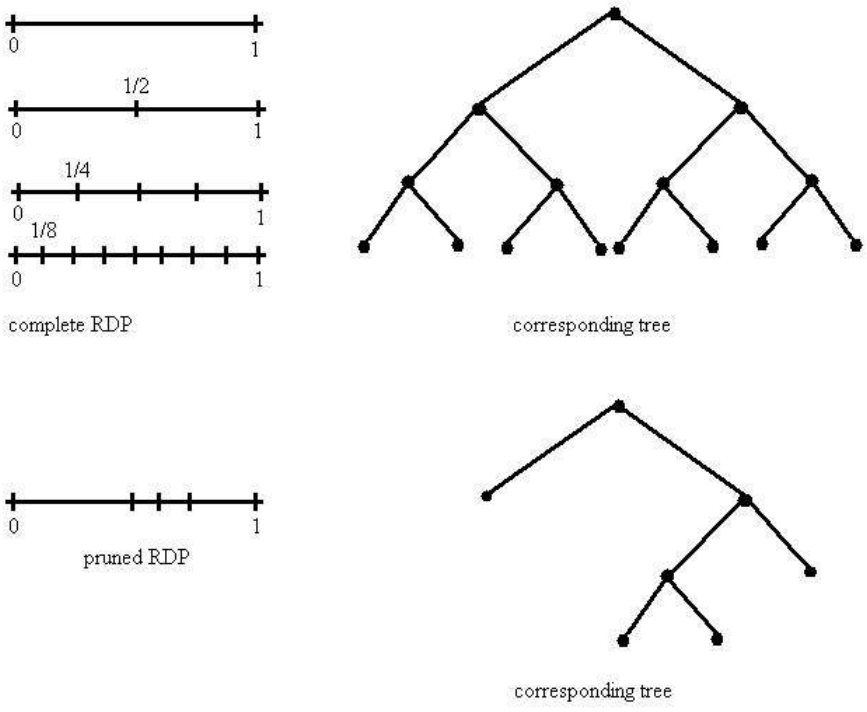


Figure 12.1: Complete and pruned RDP along with their corresponding tree structures, when $j_{\min} = 0$.

The partition given by this lemma is particularly suitable to construct an approximation of f^*. Let $\mathcal{P}(f^*, j_{\max}, j_{\min})$ denote the partition described in the lemma, and consider a function

$$\bar{f}(x) = \sum_{P \in \mathcal{P}(f^*, j_{\max}, j_{\min})} \bar{c}_P \mathbf{1}\{x \in P\} \ ,$$

where $\bar{c}_P = f^*(x_P)$ for some point $x_P \in P$ (the exact choice of x_P is not important). This function approximates f^* well, as the following result indicates.

Lemma 12.3.2 *Take $x_i = i/n$. Then*

$$\frac{1}{n} \sum_{i=1}^{n} (\bar{f}(x_i) - f^*(x_i))^2 \leq L^2 2^{-2j_{\min}} + 4R^2(M-1)2^{-j_{\max}} + \frac{4R^2(M-1)}{n} \ .$$

Proof Note that there are to types of elements in $\mathcal{P}(f^*, j_{\max}, j_{\min})$. Those containing boundary points, and those that do not contain boundary points (over the last ones f^* is Lipschitz). Let $\mathcal{P}_{\text{boundary}}$ be the collection of the first type of elements and $\mathcal{P}_{\text{Lip}}$ denote all the other ones. Therefore $\mathcal{P}(f^*, j_{\max}, j_{\min}) = \mathcal{P}_{\text{boundary}} \cup \mathcal{P}_{\text{Lip}}$.

With this in hand note that

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}(\bar{f}(x_i)-f^*(x_i))^2 \\
&=\frac{1}{n}\sum_{P\in\mathcal{P}(f^*,j_{\max},j_{\min})}\sum_{i:x_i\in P}(\bar{f}(x_i)-f^*(x_i))^2 \\
&=\frac{1}{n}\sum_{P\in\mathcal{P}_{\text{boundary}}\cup\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}(\bar{c}_P-f^*(x_i))^2 \\
&=\frac{1}{n}\sum_{P\in\mathcal{P}_{\text{boundary}}\cup\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}(f^*(x_P)-f^*(x_i))^2 \\
&=\frac{1}{n}\sum_{P\in\mathcal{P}_{\text{boundary}}}\sum_{i:x_i\in P}(f^*(x_P)-f^*(x_i))^2+\frac{1}{n}\sum_{P\in\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}(f^*(x_P)-f^*(x_i))^2\ .
\end{aligned}$$

Now, for the partitions sets $P\in\mathcal{P}_{\text{boundary}}$ we cannot take advantage of the Lipschitz assumption. But since f^* is bounded we can get the following simple bound

$$\begin{aligned}
\sum_{P\in\mathcal{P}_{\text{boundary}}}\sum_{i:x_i\in P}(f^*(x_P)-f^*(x_i))^2 &\le \sum_{P\in\mathcal{P}_{\text{boundary}}}\sum_{i:x_i\in P}4R^2 \\
&\le \sum_{P\in\mathcal{P}_{\text{boundary}}}4R^2(n\text{Vol}(P)+1) \\
&= 4R^2\sum_{P\in\mathcal{P}_{\text{boundary}}}(n2^{-j_{\max}}+1) \\
&\le 4nR^2(M-1)2^{-j_{\max}}+4R^2(M-1)\ ,
\end{aligned}$$

where the first inequality follows because f^* is bounded, the second inequality follows since $|\{i : i/n\in P\}|\le n\text{Vol}(P)+1$. Next, by the lemma above we know that the elements of the partition that cover the boundary points are at the deepest level, so have length $2^{-j_{\max}}$, and there are at most $M-1$ such sets.

For the other term we can use the Lipschitz assumption.

$$\begin{aligned}
\sum_{P\in\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}(f^*(x_P)-f^*(x_i))^2 &\le \sum_{P\in\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}(L(x_P-x_i))^2 \\
&\le \sum_{P\in\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}L^2\text{Vol}^2(P) \\
&\le \sum_{P\in\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}L^2 2^{-2j_{\min}} \\
&= L^2 2^{-2j_{\min}}\sum_{P\in\mathcal{P}_{\text{Lip}}}\sum_{i:x_i\in P}1 \\
&\le L^2 2^{-2j_{\min}}n\ ,
\end{aligned}$$

where the first two inequalities follow from the Lipschitz assumption, the third one simply uses the fact that there are there all the elements of the partition at size at most $2^{-j_{\min}}$. These two results together give the statement in the lemma. □

With these results in hand we are in good shape to come up with a good class of estimators. In particular we would like to consider the class of all functions of the form

$$\mathcal{F} = \left\{ f \ : \ f(x) = \sum_{P \in \mathcal{P}} c_P \mathbf{1}\{x \in P\} \ , \mathcal{P} \text{ is an RDP and } c_P \in [-R, R] \right\} \ .$$

There is, however, one problem with this class. It is uncountable, since c_P can take any value in the range $[-R, R]$, which is itself uncountable. A way to work around this issue is to quantize the possible values of c_P. Define

$$Q = \{-R, -R\frac{-n+1}{n}, -R\frac{-n+2}{n}, \ldots, R\frac{n-1}{n}, R\} \ .$$

This is a discretization of $[-R, R]$, and it is easy to see there are $2n+1$ elements in Q. Other discretization/quantization schemes are possible, but this one works for our purposes. So instead of $\mathcal{F}$ above we will consider

$$\mathcal{F}_Q = \left\{ f \ : \ f(x) = \sum_{P \in \mathcal{P}} \bar{c}_P \mathbf{1}\{x \in P\} \ , \mathcal{P} \text{ is an RDP and } c_P \in Q \right\} \ .$$

This class is now clearly countable, and we can apply all the theory and methods we developed.

The final step we need is to construct a good map $c : \mathcal{F}_Q \to \mathbb{R}$ satisfying the Kraft inequality. This is relatively easy though. An RDP with k leafs can be described with $2k-1$ bits as we seen in Chapter 9, and the value c_P corresponding to each leafs needs only $\log_2 |Q|$ bits to be described. Therefore for any $f \in \mathcal{F}_Q$ we set $c(f) = (2 + \log_2(2n+1))|\mathcal{P}(f)| - 1$, where $|\mathcal{P}(f)|$ is the size of the partition associated with f (that is, the number of leafs in the corresponding RDP). This is enough to guarantee that

$$\sum_{f \in \mathcal{F}_Q} 2^{-c(f)} \leq 1 \ .$$

With this in hand, our estimation algorithm is simply

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}_Q} \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(i/n))^2 + \frac{8\sigma^2 \log 2}{n} (((2 + \log_2(2n+1))|\mathcal{P}(f)| - 1) \right\} \ .$$

This is very easy to implement, as the penalty is additive in the partition elements. Therefore a bottom-up pruning strategy can be used, and it will have linear complexity in n.

12.3.2 Performance

We are in the setting of Proposition 12.2.1. To get a performance bound we will rely on Lemma 12.3.2, and clearly want to choose both $j_{\min} \to \infty$ and $j_{\max} \to \infty$ as $n \to \infty$. Let $\bar{f}$ be the approximation function prescribed by that lemma, that is

$$\bar{f}(x) = \sum_{P \in \mathcal{P}(f^*, j_{\max}, j_{\min})} \bar{c}_P \mathbf{1}\{x \in P\} \ ,$$

Clearly $\bar{f}$ is not necessarily in $\mathcal{F}_Q$. However, we can consider the closest element of $\mathcal{F}_Q$ to $\bar{f}$. This is obtained by taking $\bar{c}_P^{(Q)} \in Q$ so that $|\bar{c}_P - \bar{c}_P^{(Q)}| \leq \frac{R}{n}$. The resulting function is

$$\bar{f}_Q(x) = \sum_{P \in \mathcal{P}(f^*, j_{\max}, j_{\min})} \bar{c}_P^{(Q)} \mathbf{1}\{x \in P\} \ ,$$

is now in $\mathcal{F}_Q$. Let's use this function on the r.h.s. of Proposition 12.2.1.

$$\begin{aligned}
\frac{2}{n}\sum_{i=1}^{n}(\bar{f}_Q(i/n) - f^*(i/n))^2 &= \frac{2}{n}\sum_{i=1}^{n}(\bar{f}_Q(i/n) - \bar{f}(i/n) + \bar{f}(i/n) - f^*(i/n))^2 \\
&\leq \frac{2}{n}\sum_{i=1}^{n}(R/n + \bar{f}(i/n) - f^*(i/n))^2 \\
&= \frac{2}{n}\sum_{i=1}^{n}(\bar{f}(i/n) - f^*(i/n))^2 + \frac{2}{n}\sum_{i=1}^{n}\frac{R^2}{n^2} + \frac{2R}{n}(\bar{f}(i/n) - f^*(i/n)) \\
&\leq \frac{2}{n}\sum_{i=1}^{n}(\bar{f}(i/n) - f^*(i/n))^2 + \frac{2R^2}{n^2} + \frac{4R^2}{n} \\
&\leq 2L^2 2^{-2j_{\min}} + 8R^2(M-1)2^{-j_{\max}} + \frac{4R^2(M-1)}{n} + \frac{2R^2}{n^2} + \frac{4R^2}{n} \\
&= O\left(2^{-2j_{\min}} + 2^{-j_{\max}} + \frac{1}{n^2} + \frac{1}{n}\right) \ , \qquad (12.1)
\end{aligned}$$

as $n \to \infty$, where in the last inequality we used the result in the lemma. Note that the last two terms are the ones due to quantization, and that we controlled these is a rather crude way (which can be greatly improved if so desired). The other term in the bound of Proposition 12.2.1 corresponds to the penalty, and it is simply

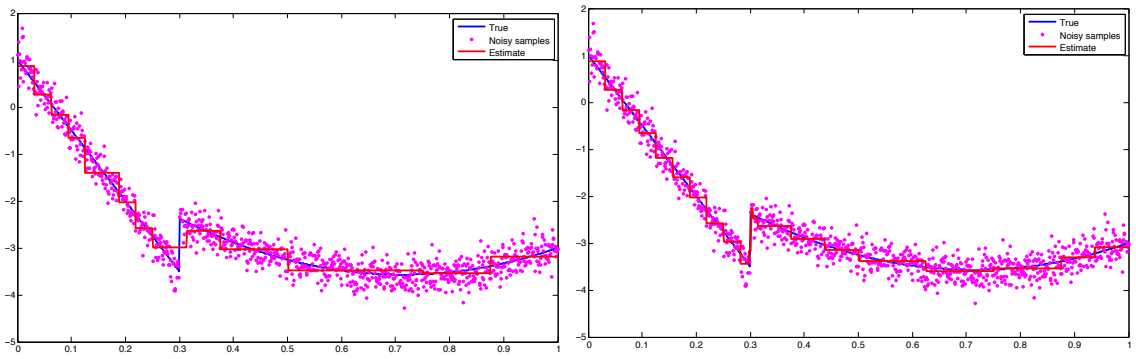
$$\begin{aligned}
&\frac{8\sigma^2 \log 2}{n}(((2 + \log_2(2n+1))|\mathcal{P}(f)| - 1) \qquad (12.2) \\
&\leq \frac{8\sigma^2 \log 2}{n}(((2 + \log_2(2n+1))(2^{j_{\min}} + (M-1)(j_{\max} - j_{\min})) - 1) \\
&= O\left(2^{j_{\min}}\frac{\log n}{n} + j_{\max}\frac{\log n}{n} + \frac{\log n}{n}\right) \ , \qquad (12.3)
\end{aligned}$$

as $n \to \infty$. So all that is left to be done is to figure out choices for $j_{\min}$ and $j_{\max}$ that result in a good bound. From the penalty term we see that there is nothing to loose if we take $j_{\max} = \log_2 n$. The value of $j_{\min}$ must be taken to balance out the first terms in (12.1) and (12.3). Solving the following equation for $j_{\min}$ gives the "right" choice

$$2^{-2j_{\min}} = 2^{j_{\min}}\frac{\log n}{n} \ .$$

This means we should take $j_{\min} = \frac{1}{3}\log_2 \frac{n}{\log n}$ Therefore both (12.1) and (12.3) are $O((n/\log n)^{-2/3})$. So, we just showed that our estimator is such that, for the class of piecewise Lipshitz functions we have

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\hat{f}_n(x_i) - f^*(x_i))^2\right] = O\left(\left(\frac{n}{\log n}\right)^{-2/3}\right) \ ,$$



(a) (b)

Figure 12.2: Illustration of the proposed estimator. In panel (a) we see an example of a dataset with $n = 1024$, the corresponding regression function and estimator using precisely the proposed methodology. In panel (b) the estimator was modified so that the penalization is only 1/5 of that prescribed by the theory.

as $n \to \infty$. Compare this to the result we have for Lipschitz functions. The only difference is that instead of n we have now $n/\log n$. This is a small price we have to pay for not knowing where the boundary pieces are. It turns out that this extra logarithmic effect is unavoidable.

12.4 Final Remarks

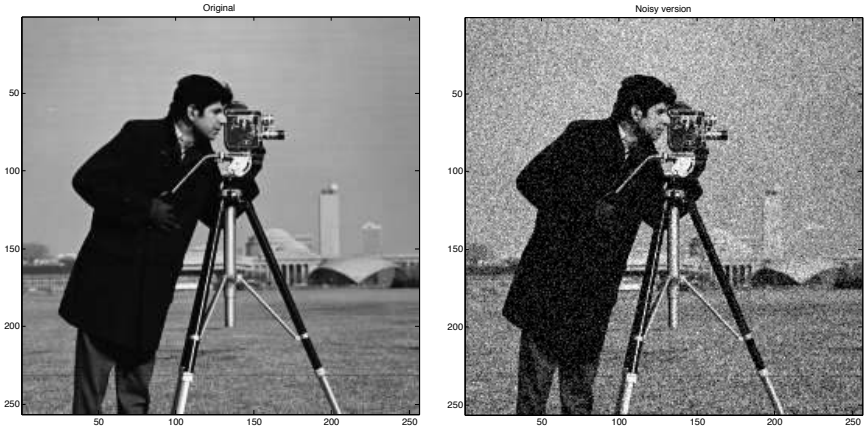
As it was the case in previous chapters, the penalization prescribed by the theory is a bit too stringent. However, the theory tells us how to property choose the form of the penalization, and this, together with the use of cross-validation and related methods, is enough to obtain very good and practical algorithms.

As an illustration in Figure 12.2(a) we have an example of data collected according to the assumed model, and the corresponding estimate with the theoretically chosen penalty. The penalization is clearly a bit too strong. In panel (b) the same estimator was used with the penalization multiplied by a factor 1/5 leading to significantly better results. A good choice of penalization can be obtained by cross-validation, for instance.

The results presented here can be generalized to higher dimensions. Dimension 2 is particularly interesting, as we can consider that f^* is a piecewise Lipschitz function, where the pieces are separated by a boundary satisfying a box-counting condition. Therefore, all the methods and analysis conducted in this chapter can be extended to that case as well, and one will conclude that

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[(\hat{f}_n(x_i) - f^*(x_i))^2\right] = O\left(\left(\frac{n}{\log n}\right)^{-1/2}\right) , \tag{12.4}$$

where x_i are samples over a square lattice in $[0, 1]$. This is actually the best result one might hope for. Furthermore, in the exercises of Chapter 4 we have seen that, in two dimensions this is the performance we can get for Lipschitz functions over the entire unit square (apart from the logarithmic factor). In Figure 12.3 you can see an example of this method applied to the denoising of a natural image (for illustration purposes only- no detailed information is given).



(a) (b)

(c)

Figure 12.3: Illustration of the proposed estimator in two-dimensions. In panel (a) we see the original regression function (in this case a natural image). In panel (b) are the observations, obtained by adding Gaussian noise to the original image. In panel (c) is the resulting regression estimator.

12.5 Exercises

Exercise 12.5.1 *Consider the class of piecewise Lipschitz functions in two dimensions, there the boundary of the various pieces satisfies a box-counting condition. Show the result in Equation* (12.4).

Bibliography

Balsubramani, A. and Ramdas, A. (2015). Sequential Nonparametric Testing with the Law of the Iterated Logarithm. *ArXiv e-prints* .

Bayes, T. (1763). An essay toward solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London* **53**, 370–418. Communicated by R. Price, in a letter to J. Canton.

Breiman, L., Friedman, J., Stone, C. and Olshen, R. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

Cover, T. and Thomas, J. (2006). *Elements of Information Theory*. A Wiley-Interscience publication. Wiley.

Craven, P. and Wahba, G. (1978/79). Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.* **31**(4), 377–403.

Devroye, L., Györfi, L. and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.

Grenander, U. (1981). *Abstract inference*. Wiley series in probability and mathematical statistics. Wiley, New York.

Kääriäinen, M. (2006). Active learning in the non-realizable case. In *Algorithmic Learning Theory*, volume 4264 of *Lecture Notes in Computer Science*, pp. 63–77. Springer Berlin Heidelberg.

Quinlan, J. (2014). *C4.5: Programs for Machine Learning*. Morgan Kaufmann series in machine learning. Elsevier Science.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica* **14**, 465–471.

Scott, C. (2005). Tree pruning with subadditive penalties. *IEEE Transactions on Signal Processing* **53**(12), 4518–4525.

Scott, C. and Nowak, R. (2006). Minimax-optimal classification with dyadic decision trees. *Information Theory, IEEE Transactions on* **52**(4), 1335–1353.

Valiant, L. G. (1984). A theory of the learnable. *Commun. ACM* **27**(11), 1134–1142.