Statistical Learning Theory

Summer 2023

Thomas Nagler

Version: July 11, 2023

Contents

1. Introduction and overview

1.1. Statistical learning theory

Advances in artificial intelligence are a key driver for our modern economy and society. Much of this success can be attributed to *machine learning (ML)*. In machine learning, an algorithm sifts through large amounts of data to find patterns that allow us to make predictions.

Machine learning is a field intersecting mathematics, statistics, and computer science. This course is about *statistical learning theory*, so what does that mean? The core of statistics is a probabilistic view of the world. Statisticians view observed data as something that's been generated through a random mechanism. Whatever we compute from the data inherits this randomness, which induces uncertainty. Understanding and controlling this uncertainty is the main problem in statistics.

Statistical learning theory is all about understanding ML algorithms from that perspective. When do algorithms (likely) work and why? When do they (likely) fail? What can we possibly learn from data in the first place? To answer these questions, we use fundamental laws of probability — similar to how a theoretical physicist uses physical laws to understand physical phenomena.

In machine learning, this uncertainty shows most prominently in the difference between an algorithm's performance on training data (*in-sample*) and test data (*out-of-sample*). Our ultimate goal is to have algorithms that predict new data well on unseen data. However, we train our models on whatever (random!) sample nature gives us. Sometimes we're lucky and sometimes we're not. When an algorithm that performs well in-sample also does so out-of-sample, we speak of *generalization*. The difference between in- and out-of-sample performance is called *generalization gap* and itself random. Sometimes it's smaller, sometimes it's larger. Understanding the factors that drive this gap and deriving mathematical bounds for it are the key objective of statistical learning theory.

1.2. Scope of this course

On a high level, this course will teach you:

- The most important results and concepts in statistical learning theory.
- The probabilistic laws and mathematical tools to derive them.

As the name suggests, the course is theoretical in nature. Part of what makes it interesting and fun is the beauty and power of the mathematics behind it. There's no need to be scared, a solid backround in real analysis and probability is enough to appreciate the maths we'll see. But we're not doing theory just for the sake of it:

"There is nothing more practical than a good theory." — Kurt Lewin

An understanding of the fundamental mechanisms governing learning and generalization is an invaluable asset in practice. The intuition we gain helps us design algorithms, identify failure modes, and improve them.

At the end of this course, you should feel relatively comfortable reading theoretical papers in statistical learning. You'll know most of the concepts they deal with, what their results mean, and have an idea on how they are derived.

1.3. Limitations

The field of machine learning is vast and so is the underlying theory. Supervised, semi-supervised, unsupervised learning, online learning, reinforcement learning, generative models, and let's not even start on the plethora of algorithms to solve these problems. To not get lost in this jungle when building our understanding, we need to limit the scope. In particular, you will **not** learn about

- tailored solutions to specific ML problems,
- new ML algorithms,
- optimization and computational techniques.

Nevertheless, the ideas and tools we develop are fundamental enough to transfer to many settings and problems beyond those discussed in this course.

1.4. These notes

These lecture notes are meant to support the oral lectures during class. The notes are more detailed than what is actually done in the latter. In class, we focus on understanding and interpreting the main results and tools. I will often omit or shorten mathematical arguments and proofs if additional detail wouldn't add much to our understanding. Reading the additional details in the notes is an optional offer to interested students. The general rule is: what we don't discuss in class isn't necessary for succeeding in this course.

1.5. Outlook

We start by introducing the formal statistical framework for our view on machine learning. There are two key quantities:

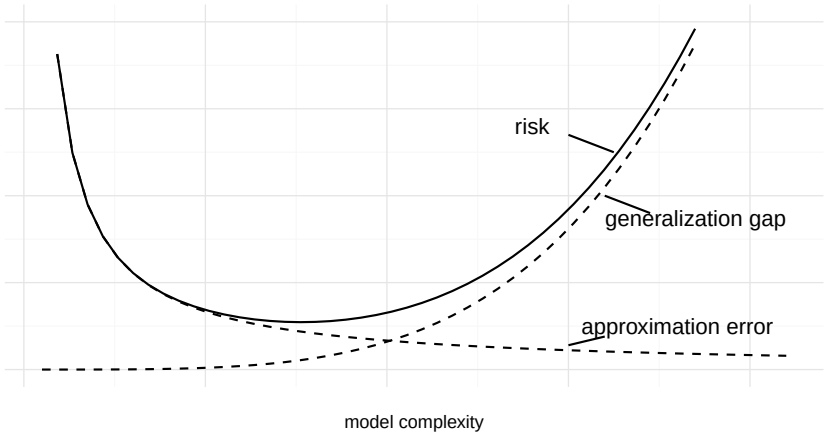

Figure 1.1.: The approximation-generalization trade-off: the more complex our model, the smaller the approximation error, but the larger the generalization error.

- The *empirical risk* or *training error* $R_n(h)$ of a predictor h quantifies its performance on a specific training data of size n.
- The *(population) risk* or *test error* $R(h)$ is its expected performance on unseen data.

The *empirical risk minimizer (ERM)* is defined as

$$\widehat{h} = \arg\min_{h \in \mathcal{H}} R_n(h),$$

where $\mathcal{H}$ is some collection of functions. Most ML algorithms can be framed as (an approximation of) an ERM. We'll see bounds of the form:[1]

$$R(\widehat{h}) - R_n(\widehat{h}) \lesssim \frac{\mathcal{C}(\mathcal{H})}{\sqrt{n}} \qquad \text{(with high probability)}, \tag{1.1}$$

where $\mathcal{C}(\mathcal{H})$ is some measure of the size or *complexity* of $\mathcal{H}$. We'll learn how to arrive at such results and how to adequately measure the complexity $\mathcal{C}(\mathcal{H})$. There are two important observations from (1.1). First, the gap decreases in n, so more data is better (unsurprisingly). Second, the bound increases with the size/complexity of $\mathcal{H}$. So the simpler our model class $\mathcal{H}$ is, the smaller the gap is. Defining $h^* = \arg\min_{h \in \mathcal{H}} R(h)$ as the best predictor in $\mathcal{H}$. From bounds on the generalization gap, we also get similar bounds on the *estimation error* $R(\widehat{h}) - R(h^*)$. The key mathematical concepts we learn are:

- Concentration of measure: Bounds on the deviations between sample average and expectation $|\overline{X}_n - \mathbb{E}[X]|$.
- Several measures $\mathcal{C}(\mathcal{H})$ for the size/complexity of $\mathcal{H}$.

[1] The notation $\lesssim$ is used as "less than a constant times".

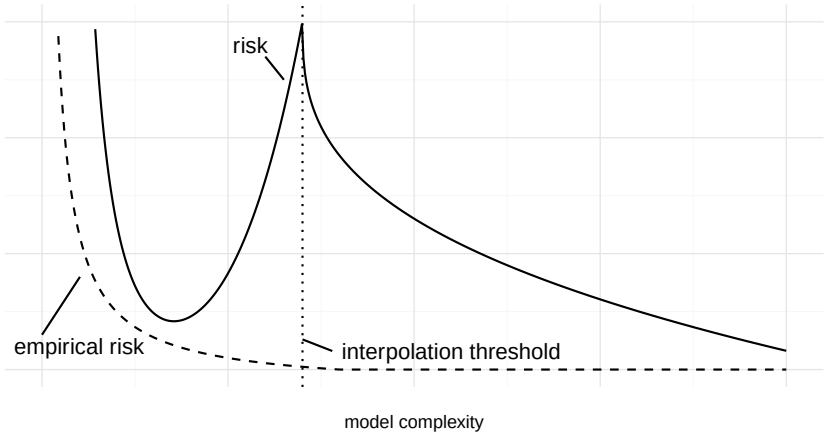

Figure 1.2.: The double descent phenomenon: after passing the interpolation threshold, performance increases with complexity.

The generalization gap and estimation error are not all we care about in practice. If the risk $R(h^*)$ is large, bounding the gap isn't all that helpful. We also want the gap between $R(h^*)$ and the risk $R(h_0)$ of the best possible predictor h_0 to be small. Because the functions h^* and h_0 are deterministic, bounding this gap is not a statistical problem but central to mathematical approximation theory, which we only touch upon. In summary: the larger the set $\mathcal{H}$, the smaller the approximation error $R(h^*) - R(h_0) = \min_{h \in \mathcal{H}} [R(h^*) - R(h_0)]$. This contrasts our bound on the generalization gap in (1.1), where enlarging $\mathcal{H}$ hurts performance. That's the infamous bias-variance trade-off you probably already heard about, see Fig. 1.1 for an illustration. We'll discuss the approximation abilities and complexity of some modern ML algorithms as well the role of regularization. Further topics are refinements of the bounds and the mathematical limits of how well/fast we can learn specific problems.

Current research in statistical learning theory is largely driven by the puzzling success of deep learning. These models are so flexible that they can achieve zero training error on almost any data set. This versatility suggests that their complexity $\mathcal{C}(\mathcal{H})$ is huge, which makes the bound (1.1) vacuous. In practice, however, adding more layers and neurons even seems to help generalization. This phenomenon is known as *double descent* and illustrated in Fig. 1.2. It appears that increasing the model size hurts performance only up until the point where the model is so flexible that it can perfectly interpolate the data (zero training error). From that point on, increasing the model size improves generalization!

Although the double descent phenomenon seemingly contradicts our previous findings, the fundamental laws of probability are inevitable. We have to be more careful with our bounds though. Double descent isn't specific to deep learning and has since been discovered (even theoretically) in algorithms as simple as linear regression. The course ends with some recent results and open questions along these lines.

2. Theoretical framework

Before we start developing our theory, we have to formalize what we mean by machine learning. On a high level:

- we want to learn about some target function $h_0 \colon \mathcal{Z} \to \mathcal{O}$,
- by running an algorithm $\mathcal{A}$ on a training data set $\mathcal{D}_n = (Z_i)_{i=1}^n \in \mathcal{Z}^n$.

We hope that the algorithm produces a function $\widehat{h} = \mathcal{A}(\mathcal{D}_n)$ that is close to h_0 in some practically meaningful sense. The output of the algorithm is often called *hypothesis* in the statistical learning literature.

2.1. Data, loss, and risk

To justify this hope, we assume that the training data $Z_1, \dots, Z_n$ are *iid* samples from some probability measure P that somehow relates to the target function h_0.

Example 2.1.1. *In supervised settings, one observation $Z_i = (Y_i, X_i)$ consists of a label $Y_i \in \mathcal{Y}$ and a feature vector $X_i \in \mathcal{X}$. The standard objective is the regression function $h_0(x) = \mathbb{E}_P[Y \mid X = x]$.*

Remark 2.1.2. *The iid assumption could be relaxed in several ways (time series structure, distribution drift, etc.), but this would unnecessarily complicate our analysis.*

The next question is how we measure whether $\widehat{h}$ is close to h_0. We use a general decision-theoretic formulation.

Definition 2.1.3.

- *The **loss function** is a map $L \colon \mathcal{Z} \times \mathcal{O} \to \mathbb{R}$ such that $L(z, h(z))$ measures how well the hypothesis h does on the sample z.*
- *The **risk** $R(h) = \mathbb{E}_P[L(Z, h(Z))]$ measures how well the hypothesis h does on average.*

Our goal is to find algorithms that produce hypotheses with small risks. The function h_0 is normally taken as the one that minimizes risk, $h_0 = \arg\min_h R(h)$, where the minimum is taken over all measurable functions $h \colon \mathcal{Z} \to \mathcal{O}$. The corresponding risk $R_0 = R(h_0)$ is called **Bayes risk** and cannot be improved

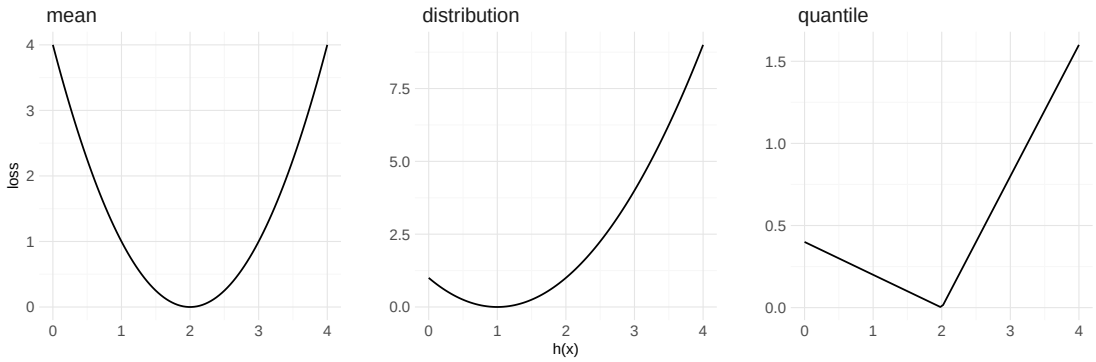

Figure 2.1.: Loss functions for regression problems with $y = 2$, $y' = 2.1$, and $\alpha = 0.8$.

upon. The **excess risk** $R(h) - R(h_0)$ measures how much worse a hypothesis h is compared to the Bayes risk.

Even though we are free to specify the loss function L, the risk $R(h)$ depends on the unknown (!) probability measure P. The risk is therefore not very useful in practice. To approximate it by something we can compute in practice, we may replace P with the empirical measure P_n.

Definition 2.1.4. *The **empirical risk** is defined as*

$$R_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(Z_i, h(Z_i)),$$

where the average is taken over the training data.

Remark 2.1.5. *In supervised settings where we predict $Y \in \mathcal{Y}$ from $X \in \mathcal{X}$, it is more common to write the loss functions directly as a map $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. This should not cause any confusion and we will use both formulations interchangeably.*

2.2. Examples

The setup so far is quite abstract, but for a good reason. It allows us to simultaneously cover many interesting problems in machine learning. Let's see some examples.

2.2.1. Regression

Consider a supervised learning setup, where $Z = (Y, X) \in \mathbb{R} \times \mathcal{X}$ and we want to predict a numeric label Y from features X. The loss functions to come are shown in Fig. 2.1

Example 2.2.1 (Mean regression)**.** *To learn the mean regression function* $h_0(x) = \mathbb{E}_P[Y \mid X = x]$*, one most commonly uses the* ***square-loss***

$$L(y, h(x)) = (y - h(x))^2.$$

The corresponding risk (also called L_2*-loss or mean squared error) is*

$$R(h) = \mathbb{E}[(Y - h(X))^2].$$

In the left panel of Fig. 2.1, we see that predictions $h(x)$ *away from the actual label* $y = 2$ *are penalized quadratically, leading to a higher risk.*

Indeed, the square loss lets us identify h_0: One can verify that it holds $h_0 = \arg\min_h R(h)$, where the minimum is taken over all functions $h\colon \mathcal{X} \to \mathbb{R}$.

We may also want to learn about other aspects of the predictive distribution.

Example 2.2.2 (Distribution regression)**.** *Suppose we want to learn the conditional distribution*

$$h_0(x) = P(Y \leq y' \mid X = x) = \mathbb{E}_P[\mathbb{1}(Y \leq y') \mid X = x],$$

where y' *is some fixed threshold. The formulation as an expectation on the far right reveals that this problem is no different from mean regression. We can use the loss*

$$L(y, h(x)) = (\mathbb{1}(y \leq y') - h(x))^2$$

and corresponding risk

$$R(h) = \mathbb{E}[(\mathbb{1}(Y \leq y') - h(X))^2].$$

The loss function is shown in the middle panel of Fig. 2.1 for $y = 2$ *and* $y' = 2.1$*, so* $\mathbb{1}(y \leq y') = 1$*. The loss penalizes every* $h(x)$ *away from 1. If* $y \leq y'$*, the loss function would shift and penalize deviations from 0.*

Example 2.2.3 (Quantile regression)**.** *We have* $Z = (Y, X) \in \mathbb{R} \to \mathcal{X}$ *and want to learn the conditional quantile function*

$$h_0(x) = Q(\alpha \mid x) = \inf\{y\colon P(Y \leq y \mid X = x) \leq \alpha\},$$

where α *is some fixed quantile level. We can use the* ***pinball loss***

$$L(y, h(x)) = (1 - \alpha)(h(x) - y)\mathbb{1}\{h(x) > y\} + \alpha(y - h(x))\mathbb{1}\{h(x) \leq y\}.$$

The quantile function $Q(\alpha \mid \cdot)$ *indeed minimizes the corresponding risk. The special case* $\alpha = 0.5$ *is* median regression*. The loss function in Fig. 2.1 penalizes* $h(x)$ *values away from* $y = 2$*, but asymmetrically. The slopes of the straight lines are exactly* $-\alpha$ *(left wing) and* $1 - \alpha$ *(right wing). The quantile level* $\alpha = 0.8$ *is*

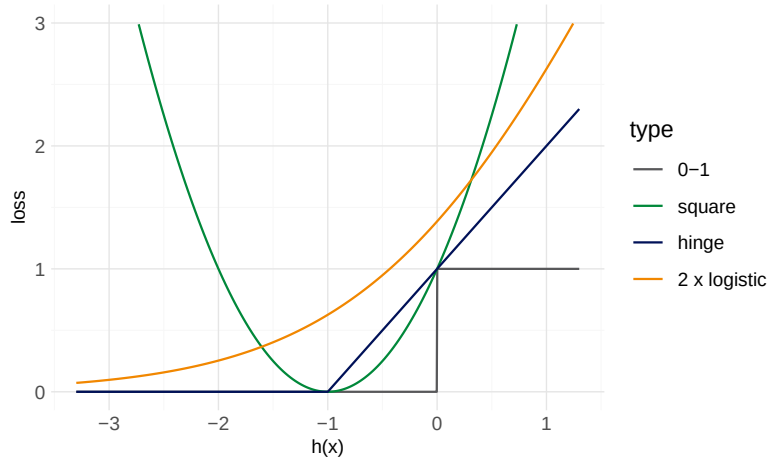

Figure 2.2.: Loss functions for classification problems with $y = -1$.

closer to 1, so we expect fewer exceedances and therefore penalize them harder.

2.2.2. Classification

In binary classification, we have $Z = (Y, X) \in \{-1, 1\} \times \mathcal{X}$ and want to learn the function that best predicts Y from X. Let's assume that $h\colon \mathcal{X} \to \mathbb{R}$ has real-valued output. To predict a label $Y \in \{-1, 1\}$ from X, we take $\operatorname{sign} h(X)$ where

$$\operatorname{sign} y = \begin{cases} 1, & \text{for } y \geq 0, \\ -1, & \text{for } y < 0. \end{cases}$$

Example 2.2.4 (0-1-loss). *The most natural is the* ***0-1-loss*** *and* ***misclassification risk***

$$L(y, h(x)) = \mathbb{1}\{y \neq \operatorname{sign} h(x)\}, \quad R(h) = \mathbb{P}\{y \neq \operatorname{sign} h(x)\}.$$

Remark 2.2.5. *The function* $h_0(x) = 2\mathbb{P}\{Y = 1 \mid X = x\} - 1 = \arg\min_h R(h)$ *is called* ***Bayes classifier****. It classifies a sample as* 1 *if* $\mathbb{P}\{Y = 1 \mid X = x\} > 1/2$ *and* -1 *otherwise. The corresponding Bayes risk* $R_0 = R(h_0)$ *can be anywhere between 0 and 1/2. If it is close to 1/2, we can't do any better than random guessing; the feature* X *doesn't provide any information about* Y *and* Y *itself has* $P(Y = 1) = 1/2$*. If* $R_0 = 0$*, we call the setting* realizable*: it is possible to find a classifier that never misclassifies.*

While the 0-1-loss is a natural choice, it has an important deficiency. It is a discontinuous function that effectively turns our hypotheses into functions with binary output. This is both theoretically and practically annoying. For example, optimizing the empirical risk becomes a combinatorial problem that is NP-hard.

In practice, we therefore also consider other, so-called **surrogate loss functions**.

Example 2.2.6 (Surrogate losses)**.** *The most popular surrogate losses are:*

- ***Square loss**: $L(y, h(x)) = (y - h(x))^2$. Treats classification as a mean regression problem, essentially ignoring the fact that we constrain output to $\{-1, 1\}$.*
- ***Hinge loss**: $L(y, h(x)) = \max(1 - yh(x), 0)$. Especially popular with Support Vector Machines, where the quantity $yh(x)$ has a geometric interpretation.*
- ***Logistic loss**: $L(y, h(x)) = \ln(1 + e^{-yh(x)})$. The risk is then proportional to the negative log-likelihood for the logistic model*

$$P(Y_i = 1 \mid X_i = x) = \frac{1}{1 + e^{-h(x)}}.$$

Note that all surrogate losses are convex upper bounds to the 0-1-loss. For multiclass classification, $\mathcal{Y} = \{1, \dots, K\}$, we can take $h \colon \mathcal{X} \to \mathbb{R}^K$ and predict the class $\widehat{k} = \arg\max_{1 \le k \le K} h_k(x)$. The loss functions above can be adapted in several ways, but let's not get too deep in the woods.

2.2.3. Unsupervised learning

Our framework also covers unsupervised problems. Here, the data $Z = X$ contain no labels and we hope to learn something useful about the distribution of X. Two popular examples are density estimation and clustering.

Example 2.2.7. *Suppose we want to learn the density $h_0 = p_Z$ of Z and $\mathcal{H}$ is some collection of density functions. We normally solve this with maximum likelihood estimation. Framed in ML lingo: we consider the **log loss***

$$L(z, h(z)) = -\ln h(z).$$

*The corresponding risk $R(h) = \mathbb{E}[-\ln h(Z)]$ is called negative **cross-entropy**. Minimizing it is equivalent to minimizing the Kullback-Leibler divergence between h and p_Z.*

In clustering problems, we want to find groups of observations that are in some way similar. Similarity has no canonical definition, so there are plausible ways to frame this. One is to assume that $p_Z = \alpha_1 p_1 + \cdots + \alpha_K p_K$ is a mixture density, each representing a cluster. Finding the mixture parameters then brings us back to density estimation. The similarity concept in K-means clustering has a more geometric flavor. To cluster observations, we partition the feature space into disjoint sets $\mathcal{Z}_1, \dots, \mathcal{Z}_K$. Our hypothesis $h \colon \mathcal{Z} \to \{1, \dots, K\}$ assigns each

observation its cluster. Different hypotheses correspond to different partitions of $\mathcal{Z}$. As a loss, we take the squared distance from z to the cluster center $\mu_h(\mathcal{Z}_k)$:

$$L(z, h(z)) = \sum_{k=1}^{K} \mathbb{1}\{h(z) = k\} \|z - \mu_h(\mathcal{Z}_k)\|^2.$$

To minimize the risk, it then suffices to find the center parameters $\mu_h(\mathcal{Z}_k)$.

2.2.4. And too many more

We stop with the examples, although there are many more. In a way, there are too many already. They served well to illustrate the generality and power of what's to come. But they are more distracting than helpful for what we want to achieve. Most of the time, we shall stick to the general formulation in Section 2.1. This allows us to discover and develop fundamental principles that apply to all the examples.

2.3. The hypothesis class

Let's finally talk about the algorithm $\mathcal{A}$. Formally, it is a map[1] $\mathcal{A} : \mathcal{Z}^n \to \mathcal{O}^{\mathcal{Z}}$ that takes a data set $\mathcal{D}_n$ of arbitrary size n and maps it to a hypothesis $\widehat{h} \in \mathcal{O}^{\mathcal{Z}}$. Here, $\mathcal{O}^{\mathcal{Z}}$ is the set of all function $h\colon \mathcal{Z} \to \mathcal{O}$.

Let $\mathcal{H}$ be the collection of all hypotheses the algorithm can produce. It will play an important role. Some examples for $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{O} = \mathbb{R}$ are:

Example 2.3.1. ***Linear functions:***

$$\mathcal{H} = \left\{ h\colon \mathcal{X} \to \mathbb{R}\colon x \mapsto \beta^\top x,\ \beta \in \mathcal{B} \subset \mathbb{R}^d \right\}.$$

Example 2.3.2. ***Basis expansions:***

$$\mathcal{H} = \left\{ h\colon \mathcal{X} \to \mathbb{R}\colon x \mapsto \sum_{k=1}^{K} \beta_k \phi_k(x),\ \beta \in \mathcal{B} \subset \mathbb{R}^K \right\},$$

where $(\phi_1, \dots, \phi_K)$ is a basis for $\mathcal{H}$ (like splines) and β are the basis coefficients.

Example 2.3.3. ***Partition functions:***

$$\mathcal{H} = \left\{ h\colon \mathcal{X} \to \mathbb{R}\colon x \mapsto \sum_{k=1}^{K} \beta_k \mathbb{1}(x \in \mathcal{X}_k),\ \beta \in \mathcal{B} \subset \mathbb{R}^K, \mathcal{X} = \bigcup_{k=1}^{K} \mathcal{X}_k \right\}.$$

[1] Technically, even a family of maps (one for each n), but that's not an important detail.

The functions produced by (boosted/bagged) trees can be written that way.

Example 2.3.4. ***Neural networks:***

$$\mathcal{H} = \left\{ h \colon \mathcal{X} \to \mathbb{R} \colon x \mapsto W_1 \sigma(W_2 \sigma(\cdots \sigma(W_M x))),\ W_k \in \mathcal{W}_k \subset \mathbb{R}^{p_k \times p_{k+1}} \right\},$$

where the m-th row of W_k are the bias and weights for the m-th neuron of the k-th layer.

2.4. Empirical risk minimization (ERM)

Optimally, we would construct an algorithm that minimizes the risk $R(\widehat{h})$. But the risk depends on the true measure P which we don't know. To construct algorithms, we can simply replace it with the empirical measure $P_n = n^{-1} \sum_{i=1}^n \delta_{Z_i}$. By the law of large numbers, the empirical risk

$$R_n(h) = \frac{1}{n} \sum_{i=1}^n L(Z_i, h(Z_i))$$

approximates $R(h)$ asymptotically. To construct algorithms, we may therefore minimize the empirical risk R_n.

Definition 2.4.1. *The* ***empirical risk minimizer*** *is defined as*

$$\mathcal{A}(\mathcal{D}_n) = \widehat{h}_{ERM} = \arg\min_{h \in \mathcal{H}} R_n(h). \tag{2.1}$$

Many modern ML algorithms can be framed that way, at least idealistically. This makes the ERM a useful algorithm to study.

The ERM is a theoretical construct, however. In all examples above, the minimum in (2.1) is over sets $\mathcal{H}$ containing uncountably many hypotheses. We can't simply try them all and pick the best. Instead, we have to run a numerical optimization algorithm. Quite often, such algorithms find a hypothesis close to the true ERM. For deep neural networks, the global minimizer is neither unique nor easy to find, and different optimizers find noticeably different hypotheses. Other methods, like gradient boosting or random forests, find hypotheses in different ways. So although the ERM is a good concept to have in mind, the following developments will allow for more general algorithms.

Even in an idealistic view, one must be careful with ERM. If the class $\mathcal{H}$ is too large, the ERM is likely to *overfit* a given data set $\mathcal{D}_n$. Overfitting means that the algorithm learns patterns that can be attributed purely to random noise. Consider for example a regression task with square loss. If $\mathcal{H} = \mathbb{R}^{\mathcal{Z}}$ contains all functions from $\mathcal{Z}$ to $\mathbb{R}$, there are infinitely many $h \in \mathcal{H}$ with $R_n(h) = 0$. However,

the test loss $R(h)$ of such functions can be arbitrarily large. To avoid overfitting, we must either restrict the hypothesis set $\mathcal{H}$, or use an algorithm that picks a 'good' $h \in \mathcal{H}$.

2.5. Probably approximately correct (PAC) learning

Let's now discuss which results we are after. We like hypotheses $h \in \mathcal{H}$ that have small risk $R(h)$. Here, small means that it's not much larger than the possible best risk $R_0 = \min_{h \in \mathcal{U}^{\mathcal{Z}}} R(h)$. We may call a hypothesis h with $R(h) - R_0 \leq \epsilon$ *approximately correct*.

To generate such hypotheses, we run an algorithm $\mathcal{A}$ on a data set $\mathcal{D}_n$. But the data set $\mathcal{D}_n$ is random and so is the hypothesis $\widehat{h} = \mathcal{A}(\mathcal{D}_n)$ an algorithm spits out. Consequently, also the risk $R(\widehat{f})$ is a random variable. It would be too much to ask that an algorithm *always* returns hypotheses with small risk. But it should at least do so *with high probability* (over the randomness in $\mathcal{D}_n$). Let's formalize this.

Definition 2.5.1. *An algorithm $\mathcal{A}$ is called **probably approximately correct (PAC)**, if for every $\epsilon, \delta > 0$ and all probability measures, there exists a training set size $n = n(\epsilon, \delta)$ such that*

$$\mathbb{P}\{R(\widehat{h}) - R_0 > \epsilon\} \leq \delta.$$

An algorithm is PAC, if achieves risk arbitrarily close to the Bayes risk R_0, with arbitrarily high probability, if we run it on data sets of sufficient size. This sample size, $n(\epsilon, \delta)$, is called the *sample complexity* of the algorithm. We want it to be small of course. Another way to read the inequality above is: with probability at least $1 - \delta$, we have

$$R(\widehat{h}) - R_0 \leq \epsilon.$$

We can also frame PAC-learning by reversing the roles of n and ϵ: For every $\delta > 0$, there is a sequence $\epsilon_n(\delta)$ such that for every $n \in \mathbb{N}$ and probability at least $1 - \delta$.

$$R(\widehat{h}) - R_0 \leq \epsilon_n(\delta).$$

This may be a more natural way of thinking. The sequence ϵ_n is called *convergence rate* and quantifies how fast the algorithm achieves small risk as n grows. This formulation is slightly stronger because it must hold for any n, not just large ones. We shall call any of the inequalities about a *PAC bound*.

Definition 2.5.1 compares $R(\widehat{h})$ with the Bayes risk, the absolute best we can aim for. This aim may be too ambitious in general. In the following section, we'll prove that no algorithm can be universally PAC for all tasks. If we assume $\widehat{h} \in \mathcal{H}$, it can also make sense to compare to $R(\widehat{h})$ to the best achievable over $\mathcal{H}$, i.e. $R(h^*) = \min_{h \in \mathcal{H}} R(h)$.

Definition 2.5.2. *A hypothesis class $\mathcal{H}$ is called* ***PAC-learnable*** *if there exists an algorithm $\mathcal{A}$ such that for all probability measures and $\epsilon, \delta > 0$, there exists a sample size $n = n(\epsilon, \delta)$ such that $\widehat{h} = \mathcal{A}(\mathcal{D}_n)$, $n \geq N$, satisfies*

$$\mathbb{P}\{R(\widehat{h}) - R(h^*) > \epsilon\} \leq \delta.$$

The easiest example of a PAC-learnable class is a set $\mathcal{H} = \{h\}$ with only one element. In that case, $R(\widehat{h}) - R(h^*) = 0$ with probability 1. We'll see much richer PAC-learnable classes later on.

2.6. There's no free lunch

We should understand the choice of $\mathcal{H}$ as an *inductive bias*: a rough idea of how the function h_0 looks like. Is it linear? Is it smooth? Can it be composed of simple functions? Such inductive biases are important, even necessary in some sense. This is formalized by *no-free-lunch theorems*, which can be summarized as follows:

> *For every algorithm $\mathcal{A}$, there is a task on which it fails.*

Here, 'task' essentially means the unknown distribution P, including the 'true hypothesis' $h_0 = \arg\min_h R(h)$.

The no-free-lunch principle does not mean that there are tasks for which no algorithm succeeds. But if we have an algorithm that does well on some tasks, we can necessarily find tasks for which it does poorly. An important consequence is that there is no universal ranking of algorithms. Which algorithms perform well is intricately linked to the set of tasks we test them on. Let's see a simple variant of such theorems.

Theorem 2.6.1. *Let $R(h) = \mathbb{P}\{Y \neq \operatorname{sign} h(X)\}$ be the misclassification risk and $\mathcal{A}\colon (\{-1, 1\} \times \mathcal{X})^n \to \mathbb{R}^{\mathcal{X}}$ be an arbitrary algorithm. Then for every n, there exists a distribution P and another algorithm $\mathcal{B}$, such that*

$$\mathbb{E}[R(\mathcal{A}(\mathcal{D}_n))] \geq \frac{1}{4} \quad \text{and} \quad \mathbb{E}[R(\mathcal{B}(\mathcal{D}_n))] = 0,$$

where the expectations are over the sample $\mathcal{D}_n \overset{iid}{\sim} P$.

Remark 2.6.2. *The number $1/4$ is somewhat arbitrary and can be made arbitrarily close to $1/2$ by adapting the proof.*

The theorem states that for any sample size and algorithm $\mathcal{A}$, we can find a 'bad' task. On average, the algorithm misclassifies at least a quarter of fresh samples. This isn't because the task is hard per se, however. There does exist an algorithm $\mathcal{B}$ achieving zero misclassification risk (i.e., the task is realizable).

An important subtlety is that we can choose the task depending on the sample size, and the algorithm $\mathcal{B}$ depending on the task. Indeed, we do so in the proof ahead. Its core idea is as follows. For any finite data set $\mathcal{D}_n$, the algorithm $\mathcal{A}$ only has information for those feature values x present in the data set. Whatever the algorithm does to extrapolate on unseen feature values, there is a task that calls for the opposite. Let's make this formal.

Proof of Theorem 2.6.1. Let $\mathcal{X}' \subset \mathcal{X}$ be a set with $|\mathcal{X}'| = 2n$ and P_X be the uniform distribution over $\mathcal{X}'$. We consider tasks where there's a deterministic relationship between label Y and feature X, i.e., $Y = f(X)$ for some function $f\colon \mathcal{X}' \to \{-1, 1\}$. Denote the corresponding joint distribution for (Y, X) as P_f. A data set $\mathcal{D}_n \overset{iid}{\sim} P_f$ can then be written as $\mathcal{D}_n = (f(S), S)$ for some $S \in \mathcal{X}'^n$. Let's also assume that $h\colon \mathcal{X} \to \{-1, 1\}$ for simplicity.

Fixing S, any task f has a complementary task

$$f_S^c(x) = \begin{cases} f(x), & \text{for } x \in S \\ -f(x), & \text{for } x \notin S. \end{cases}$$

On $x \notin S$, any hypothesis h must now misclassify one of the tasks:

$$\mathbb{1}\{f(x) \neq \operatorname{sign} h(x)\} + \mathbb{1}\{f_S^c(x) \neq \operatorname{sign} h(x)\} \geq 1.$$

Then using that P_X is uniform over $\mathcal{X}'$ (see exercises),

$$R_f(h) + R_{f_S^c}(h) \geq \frac{1}{2}.$$

We have shown the following: for any data set, we can find two tasks that may have generated it, but any hypothesis must fail on one of them. It remains to take the expectation over data sets.

Let $S_1, \ldots, S_M \in \mathcal{X}'^n$ be all the $M = \binom{2n}{n}$ possible size n-tuples from $\mathcal{X}'$ and write $h_{f,m} = \mathcal{A}(f(S_m), S_m)$. Because each of the S_m is equally likely,

$$\mathbb{E}_{\mathcal{D}_n \overset{iid}{\sim} P_f}[R_f(\mathcal{A}(\mathcal{D}_n))] = \frac{1}{M} \sum_{m=1}^{M} R_f(h_{f,m}).$$

Also define $\mathcal{F} = -1, 1^{\mathcal{X}'}$ as the set of all possible tasks and note that this set is finite ($|\mathcal{F}| = 2^{2n}$). Because the maximum is larger than the average, we get

$$\begin{aligned} \max_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}_n \overset{iid}{\sim} P_f}[R_f(\mathcal{A}(\mathcal{D}_n))] &\geq \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}_n \overset{iid}{\sim} P_f}[R_f(\mathcal{A}(\mathcal{D}_n))] \\ &= \frac{1}{M} \sum_{m=1}^{M} \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} R_f(h_{f,m}) \\ &= \frac{1}{M} \sum_{m=1}^{M} \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \frac{R_f(h_{f,m}) + R_{f_{S_m}^c}(h_{f,m})}{2} \geq \frac{1}{4}. \end{aligned}$$

The second to last equality holds because for any set S, $\mathcal{F} = \{f_S^c \colon f \in \mathcal{F}\}$. We have shown that there is a task $f \in \mathcal{F}$; for which $\mathbb{E}[R_f(\mathcal{A}(\mathcal{D}_n))] \geq 1/4$ as claimed. Finally, define $\mathcal{B}$ as the algorithm always returning this f. Because $Y = f(X)$ with probability 1, it holds $R_f(\mathcal{B}(\mathcal{D}_n)) = R_f(f) = 0$ irrespective of $\mathcal{D}_n$. □

As a corollary, we obtain that not all classes $\mathcal{H}$ are PAC-learnable. This reifies the importance of inductive biases.

Corollary 2.6.3. *The set $\mathcal{H} = \{-1, 1\}^{\mathcal{X}}$ of all measurable functions $\mathcal{X} \to \{-1, 1\}$ is not PAC-learnable.*

Proof. Recall the setting of Theorem 2.6.1. Because $Y = h(X)$ is deterministic, the best achievable risk is $R^* = 0$. By definition of the 0-1-loss, the risk $R(h)$ of any hypothesis lies in the interval $[0, 1]$. Now for any random variable $R \in [0, 1]$, it holds

$$\begin{aligned}
\mathbb{E}[R] &= \mathbb{E}[R\mathbb{1}\{R > 1/8\}] + \mathbb{E}[R\mathbb{1}\{R \leq 1/8\}] \\
&\leq \mathbb{E}[\mathbb{1}\{R > 1/8\}] + \frac{1}{8}\mathbb{E}[\mathbb{1}\{R \leq 1/8\}] \\
&= \mathbb{P}\{R > 1/8\} + \frac{1}{8}(1 - \mathbb{P}\{R > 1/8\}) \\
&= \frac{7}{8}\mathbb{P}\{R > 1/8\} + \frac{1}{8}
\end{aligned}$$

If further $E[R] \geq 1/4$ as in Theorem 2.6.1, we get

$$\frac{1}{4} \leq \frac{7}{8}\mathbb{P}\{R > 1/8\} + \frac{1}{8} \quad \Leftrightarrow \quad \mathbb{P}\{R > 1/8\} \geq \frac{1}{7}.$$

Hence, for any algorithm, there is a task $h \in \mathcal{H}$, for which the risk is larger than $1/8$ with probability at least $1/7$. Hence, no algorithm is PAC for $\mathcal{H}$. □

3. Preliminary bounds on the risk

3.1. Risk decomposition

A central question in this course is: when do algorithms have small (excess) risk? To answer this, we want to establish PAC bounds such as

$$R(\widehat{h}) - R_0 \leq \epsilon_n(\delta) \quad \text{w.p. at least } 1-\delta.$$

A key ingredient to arriving at and understanding such results is a decomposition of the risk. Let $\mathcal{H}$ be the set of all hypotheses the algorithm can produce and define the best hypothesis in class as $h^* = \arg\min_{h\in\mathcal{H}} R(h)$. We have:

$$R(\widehat{h}) - R_0 = \underbrace{R(\widehat{h}) - R(h^*)}_{\text{estimation error}} + \underbrace{R(h^*) - R(h_0)}_{\text{approximation error}}. \tag{3.1}$$

The excess risk is split into two parts — one random, one deterministic (illustrated in Fig. 3.1). The estimation error is the random part. The randomness is driven by the data set $\mathcal{D}_n$ that leads to $\widehat{f} = \mathcal{A}(\mathcal{D}_n)$. The error is small if the algorithm likely picks hypotheses close to h^*, the best in class. This is easy when there are only a few hypotheses to choose from:

> *The smaller $\mathcal{H}$, the smaller the estimation error.*

The approximation error is deterministic. It mainly depends on the richness of $\mathcal{H}$. If it is rich enough to contain a hypothesis h^* close to h_0, the error is small:

> *The larger $\mathcal{H}$, the smaller the approximation error.*

Remark 3.1.1. *This is a good time to emphasize that 'size' is not to be taken literally. All interesting hypothesis classes $\mathcal{H}$ have infinitely many elements, often uncountably many. What we really mean is how diverse the functions contained in $\mathcal{H}$ are. More accurate terms for $\mathcal{H}$'s 'size' are* ***complexity, capacity, or richness of the class****.*

The complexity of $\mathcal{H}$ creates tension between approximation and estimation error. For successful ML applications, the two forces have to be balanced carefully. This is what hyperparameter tuning is all about. How many levels should my regression tree have? How many layers the DNN? How much should I regularize the empirical risk in my algorithm? All of this calibrates the complexity of $\mathcal{H}$. The optimal tuning parameters are those that strike the best balance.

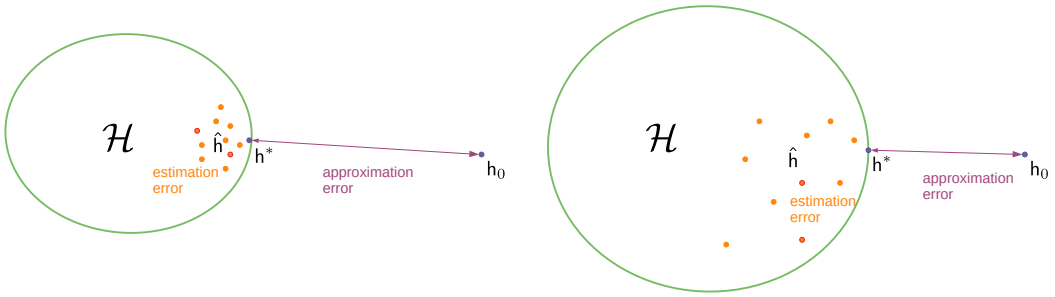

Figure 3.1.: The effect of the size of the hypothesis set on estimation and approximation error.

The estimation error normally decreases in the sample size n. More data means more information and much of the randomness can be averaged out. To maintain the right balance, we should thus increase the complexity of $\mathcal{H}$ with n. We rarely make this explicit, because we are after bounds that are valid for any n. It is nevertheless worth keeping in mind.

The estimation error will be our main focus because it is more interesting from a statistical perspective. The approximation error is deterministic and a core subject of the mathematical field of approximation theory. We'll see a few results in that direction later in this course. The key takeaway is simple: the larger $\mathcal{H}$, the smaller the approximation error.

3.2. Risk bounds

The following result will be useful for bounding the estimation error $R(\widehat{h}) - R(h^*)$.

Proposition 3.2.1. *Suppose the algorithm produces hypotheses with small empirical risk in the sense that $R_n(\widehat{h}) - R_n(h^*) \leq 0$. Then*

$$R(\widehat{h}) - R(h^*) \leq \sup_{h \in \mathcal{H}} \Big[R(h) - R_n(h) \Big] - [R(h^*) - R_n(h^*)].$$

Proof. By adding and subtracting terms, we get

$$\begin{aligned} R(\widehat{h}) - R(h^*) &= R(\widehat{h}) - R_n(\widehat{h}) + R_n(\widehat{h}) - R_n(h^*) + R_n(h^*) - R(h^*) \\ &\leq R(\widehat{h}) - R_n(\widehat{h}) + R_n(h^*) - R(h^*) \\ &\leq \sup_{h \in \mathcal{H}} \Big[R(h) - R_n(h) \Big] - [R(h^*) - R_n(h^*)], \end{aligned}$$

because $R_n(\widehat{h}) - R_n(h^*) \leq 0$ (1st inequality) and $\widehat{h} \in \mathcal{H}$ (2nd inequality). □

Remark 3.2.2. *The proposition imposes an assumption on the algorithm: $R_n(\widehat{h}) - R_n(h^*) \leq 0$. Intuitively speaking, it means that the algorithm adapts to the data at least to some degree. Its training error R_n should be smaller than that of*

the fixed hypothesis h^*, which does not adapt to the training data at all. The condition is trivially satisfied for the ERM $\widehat{h}_{ERM} = \arg\min_{h\in\mathcal{H}} R_n(f)$. But it's equally plausible for most other methods that use the empirical risk as a training criterion in some way.

The second term $R(h^*) - R_n(h^*)$ only involves a fixed hypothesis h^*. It is of order $O(1/\sqrt{n})$ by the law of large numbers (or a variance calculation). The term $\sup_{h\in\mathcal{H}}[R(h) - R_n(h)]$ is the maximal difference between empirical risk R_n and population risk R over all elements of $\mathcal{H}$. As apparent from the proof, it comes from bounding another interesting quantity.

The **generalization gap** is the difference between test and training error:

$$R(\widehat{h}) - R_n(\widehat{h}) \leq \sup_{h\in\mathcal{H}}\Big[R(h) - R_n(h)\Big].$$

This generalization bound has a more practical flavor. The empirical risk $R_n(\widehat{h})$ is something we observe. If we can make the upper bound small, say $\sup_{h\in\mathcal{H}}[R(h) - R_n(h)] \leq \epsilon$, we can extrapolate from the training error to the test error. Rearranging the inequality above yields

$$R(\widehat{h}) \leq R_n(\widehat{h}) + \epsilon.$$

So the actual risk of $\widehat{h}$ is at most ϵ larger than the empirical risk.

The uniform difference $\sup_{h\in\mathcal{H}}[R(h) - R_n(h)]$ bounds both the estimation error and the generalization gap. It is thus key to a theory of statistical learning. In much of the following, we derive and discuss bounds on this quantity. These bounds then immediately translate to bounds of the estimation error and generalization gap.

3.3. The role of uniform convergence

The term $\sup_{h\in\mathcal{H}}[R(h) - R_n(h)]$ is a random quantity. Recall that

$$R_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(Z_i, h(Z_i)), \quad R(h) = \mathbb{E}_{Z\sim P}[L(Z, h(Z))].$$

The randomness comes from the empirical risk R_n, which depends on the random data set $\mathcal{D}_n = (Z_i)_{i=1}^n$. Because $R(h) - R_n(h)$ is the difference between average and expectation, the law of large numbers implies

$$R(h) - R_n(h) \to_P 0 \quad \text{as } n \to \infty,$$

or, by definition of convergence in probability,

$$R(h) - R_n(h) \leq \epsilon \quad \text{w.p. at least } 1 - \delta, \tag{3.2}$$

for all $\epsilon, \delta > 0$ and n large enough. The law of large numbers can be seen as one of the fundamental laws of statistics. It motivates and justifies most statistical methods. Without it, learning from data would be impossible.

Equation (3.2) holds for every h, but that's not enough. For $\sup_{h\in\mathcal{H}}[R(h) - R_n(h)]$ to be small, we need $R_n(h)$ to converge to $R(h)$, *uniformly* over $h \in \mathcal{H}$:

We say that $R_n - R$ converges uniformly over $\mathcal{H}$, if

$$\sup_{h\in\mathcal{H}}\Big[R(h) - R_n(h)\Big] \to_P 0,$$

Uniform convergence of $R(h) - R_n(h)$ is the core problem of *empirical process theory* (Van der Vaart and Wellner, 1996). For any function f, denote the sample average and expectation as

$$P_n(f) = \frac{1}{n}\sum_{i=1}^{n} f(Z_i), \quad P(f) = \mathbb{E}_{Z\sim P}[f(Z)].$$

In this notation, the law of large numbers is $P_n(f) - P(f) \to_P 0$.

Definition 3.3.1. *A collection of differences* $\{P(f) - P_n(f) \colon f \in \mathcal{F}\}$ *is called* ***empirical process*** *indexed by* $\mathcal{F}$.

Defining $(L \circ h)(z) = L(z, h(z))$, we see that

$$R(h) - R_n(h) = P(L \circ h) - P_n(L \circ h),$$

and, thus,

$$\sup_{h\in\mathcal{H}}\Big[R(h) - R_n(h)\Big] = \sup_{\ell\in\mathcal{L}}\Big[P(\ell) - P_n(\ell)\Big],$$

where

$$\mathcal{L} = L \circ \mathcal{H} = \{L(\cdot, h(\cdot)) \colon h \in \mathcal{H}\}$$

is the **loss class**. Thus, $R(h) - R_n(h)$ converges uniformly, when a *uniform law of large numbers* holds:

$$\sup_{\ell\in\mathcal{L}}\Big[P(\ell) - P_n(\ell)\Big] \to_P 0.$$

We'll shoot higher than this though. The law of large numbers only establishes convergence, but it doesn't say anything about sample complexity or convergence rates. For deeper insights, we'll use *concentration inequalities*, which are the subject of the following chapter.

Remark 3.3.2. *The reframing in terms of the empirical process* $\{P_n(\ell) - P(\ell) \colon \ell \in \mathcal{L}\}$ *shows that the complexity of* $\mathcal{L}$ *is what really matters. But since* L *is fixed,* $\mathcal{L}$*'s*

and $\mathcal{H}$'s complexity are almost equivalent. We'll see some theoretical justification later on.

4. Bounds for finite hypothesis classes

4.1. Main result

Recall our preliminary bound for the generalization error:

$$R(\widehat{h}) \leq R_n(\widehat{h}) + \sup_{h \in \mathcal{H}} \Big[R(h) - R_n(h) \Big].$$

To bound the risk on the left, our goal is to provide high probability bounds on the supremum on the right. In particular, we shall prove the following result.

Theorem 4.1.1. *Suppose that* $|L(z, h(z))| \leq B < \infty$ *for all* $z \in \mathcal{Z}$ *and* $|\mathcal{H}| < \infty$. *Then*

$$\sup_{h \in \mathcal{H}} \Big[R(h) - R_n(h) \Big] \leq B \sqrt{\frac{2 \ln(|\mathcal{H}|) + 2 \ln(1/\delta)}{n}} \quad \text{w.p. at least } 1 - \delta.$$

Some observations:

- As $n \to \infty$, the right hand side decreases as $O(1/\sqrt{n})$. This rate is the same as in the central limit theorem. The above result is non-asymptotic, though. It is valid for any n.
- The probability threshold δ appears logarithmically in the bound. Because $\delta \mapsto \sqrt{\ln(1/\delta)}$ is a decreasing function, the more certain we want to be (small δ), the larger the bound.
- The upper bound of the loss function, B, shows as a constant factor. For classification, $B = 1$. For regression with square loss, B is only bounded if the data are.
- The cost of taking the sup over $|\mathcal{H}|$ functions is the $\ln(|\mathcal{H}|)$ term. The more functions are contained in $\mathcal{H}$, the looser is our bound. The bound increases only very slowly in $|\mathcal{H}|$.

Let's discuss the assumptions of the theorem. If the loss function L is unbounded, we require the data Z_i to be bounded. This is a strong assumption, many interesting random variables are unbounded. Most results we develop in this course can be generalized to unbounded random variables — at the cost of

additional assumptions on their moments and unnecessary complications in the proofs.

In this course, we mostly assume that random variables are bounded. It allows us to avoid distractions without losing anything in terms of insight.

The second assumption is that $\mathcal{H}$ contains only finitely many hypotheses. Although the bound increases only logarithmically in $|\mathcal{H}|$, it is vacuous for most interesting examples (including all in Section 2.3). But don't worry, we'll get to infinite classes.

Proving Theorem 4.1.1 serves a didactical purpose. So far, we have set a theoretical framework. It's time to develop a mathematical toolkit. We essentially need two fundamental laws of probability: the union bound and concentration of measure. The first controls a maximum of random variables, and the second the isolated deviations $[R(h) - R_n(h)]$.

4.2. The union bound

The union bound is a basic inequality for the probability of union of events. Depending on the field, the union bound is also known as *Boole's inequality* or *Bonferroni inequality*.

Theorem 4.2.1 (Union bound). *For any countable set of events $E_1, E_2, \ldots,$*

$$\mathbb{P}\left(\bigcup_{k=1}^{\infty} E_k\right) \leq \sum_{k=1}^{\infty} \mathbb{P}(E_k).$$

Remark 4.2.2. *The bound also applies to finitely many events $E_1, \ldots, E_K$, by setting $E_k = \emptyset$ for all $k > K$.*

The union bound is useful because it gives us control of maxima of random variables.

Corollary 4.2.3. *It holds,*

$$\mathbb{P}\left(\max_{1\leq k\leq K} X_k > \epsilon\right) \leq K \max_{1\leq k\leq K} \mathbb{P}(X_k > \epsilon).$$

Proof.

$$\mathbb{P}\Big(\max_{1\leq k\leq K} X_k > \epsilon\Big) = \mathbb{P}\left(\bigcup_{k=1}^{K} \{X_k > \epsilon\}\right) \leq \sum_{k=1}^{K} \mathbb{P}(X_k > \epsilon) = n \max_{1\leq k\leq K} \mathbb{P}(X_k > \epsilon).$$

□

Let's apply this to our statistical learning problem. Enumerating the hypotheses $h_1, \ldots, h_K$ with $K = |\mathcal{H}|$ and substituting $X_k = R_n(h_k) - R(h_k)$ gives

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}}\Big[R(h)-R_n(h)\Big]>\epsilon\right)\leq K\max_{1\leq k\leq K}\mathbb{P}\Big(R_n(h_k)-R(h_k)>\epsilon\Big). \tag{4.1}$$

We want the right-hand side to vanish as $n\to\infty$. By the law of large numbers,

$$\max_{1\leq k\leq K}\mathbb{P}(R_n(h_k)-R(h_k)>\epsilon)\to 0 \quad \text{as } n\to\infty.$$

Theorem 4.1.1 is much more precise however. It is non-asymptotic and quantifies the effects of sample size an n and the number of hypotheses $K=|\mathcal{H}|$. To get such deep insights, we need better tools.

4.3. Concentration of measure

4.3.1. Motivation

The law of large numbers (LLN) is a fundamental principle in statistics. It states: as the sample size increases the sample average of independent random variables, a random variable, converges in a probabilistic sense to the expectation, a deterministic quantity. We can also say that the average *concentrates* around the expectation for large n. The LLN is the most basic description of a more general phenomenon: *concentration of measure*.

Denote the empirical measure by P_n and the true probability measure by P. The sample average and expectation can then be written as the Lebesgue integrals[1]

$$\overline{X}_n=\frac{1}{n}\sum_{i=1}^n X_i=\int x\,dP_n(x) \quad \text{and} \quad \mathbb{E}[X]=\int x\,dP(x).$$

The law of large numbers $\overline{X}_n\to_P\mathbb{E}[X]$ is a consequence of the empirical measure P_n concentrating around P in an appropriate sense. Key to this is the independence of the random variables.[2]

The LLN only establishes that concentration takes place and says nothing about its tightness or speed. This is where concentration inequalities come in. They bound the probability of deviations between sample average and expectation:

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^n X_i-\mathbb{E}[X]\right|>t\right\}\leq\delta_n.$$

[1] In that sense, the sample average is nothing else than an expectation with respect to the empirical measure.

[2] The law of large numbers also holds if dependence is sufficiently weak. Roughly speaking, we require that X_i and X_j become almost independent when the gap $|i-j|$ increases. This is beyond the scope of this course, however.

In many interesting settings, these probabilities decay exponentially fast in n.[3] Because the decay is so quick, we will be able to aggregate such bounds across huge numbers of hypotheses. This eventually allows us to get uniform control over the empirical process, but let's not get ahead of ourselves.

4.3.2. Basic tail bounds

We first discuss some basic probability bounds, some of which you should already know.

Theorem 4.3.1 (Markov's inequality). *For any random variable $X \geq 0$ and $t > 0$,*

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

Proof. We calculate

$$\mathbb{E}[X] = \int_0^\infty x dP(x) \geq \int_t^\infty x dP(x) \geq t \int_t^\infty dP(x) = t\mathbb{P}(X \geq t). \qquad \square$$

The theorem is more powerful than it may look at first sight. Other than being non-negative the random variable X is arbitrary. Let's discuss some useful implications.

Corollary 4.3.2 (Moment inequalities). *For any random variable X, $k \in \mathbb{N}$, and $t > 0$,*

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

Proof. Because $\phi(t) = |t|$ is strictly increasing on $[0, \infty)$,

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq t) = \mathbb{P}\Big(|X - \mathbb{E}[X]|^k \geq t^k\Big).$$

Now apply Markov's inequality (Theorem 4.3.1). $\square$

The special case $k = 2$ is known as *Chebyshev's inequality*. This corollary relates the tail of the distribution of X to its (centralized) moments $\mathbb{E}[|X - \mathbb{E}[X]|^k]$. The more of these moments exist, the faster the decay of the tail probabilities $\mathbb{P}(|X - \mathbb{E}[X]| \geq t)$ for $t \to \infty$. However, the decay is polynomial in t.

Another useful version arises with $\phi(t) = \exp(\lambda t)$ for some $\lambda > 0$.

Corollary 4.3.3 (Chernoff bounds). *For any random variable X, and $\lambda, t > 0$,*

$$\mathbb{P}(X - \mathbb{E}[X] \geq t) \leq e^{-\lambda t}\mathbb{E}\Big[e^{\lambda(X - \mathbb{E}[X])}\Big].$$

[3] In that case, concentration inequalities are sometimes referred to as *exponential inequalities*.

Proof. The function $\phi(t) = \exp(\lambda t)$ is increasing and $e^{\lambda(X-\mathbb{E}[X])}$ is a positive random variable. Thus, Markov's inequality yields

$$\mathbb{P}(X - \mathbb{E}[X] \geq t) = \mathbb{P}\Big(e^{\lambda(X-\mathbb{E}[X])} \geq e^t\Big) \leq e^{-\lambda t}\mathbb{E}\Big[e^{\lambda(X-\mathbb{E}[X])}\Big]. \qquad \square$$

The term $M_X(\lambda) = \mathbb{E}[\exp\{\lambda(X-\mathbb{E}[X])\}]$ is also known as the *moment generating function*. If $M_X(\lambda)$ exists for some $\lambda > 0$, the Chernoff bound implies exponential tail decay. This will be key for sharp concentration inequalities. An important special case are bounded random variables.

Lemma 4.3.4 (Hoeffding's lemma)**.** *For any random variable X with $|X| \leq B$ almost surely and $\lambda > 0$,*

$$\mathbb{E}\Big[e^{\lambda(X-\mathbb{E}[X])}\Big] \leq \exp\left(\frac{\lambda^2 B^2}{2}\right).$$

Now we have the tools we need to study concentration of measure.

4.3.3. Hoeffding's inequality

Suppose $X_1, \ldots, X_n \in [-B, B]$ are *iid* random variables. We are looking for tail bounds for the difference between sample average $\overline{X}_n$ and its expectation $\mathbb{E}[\overline{X}_n] = \mathbb{E}[X]$. The Chernoff bound (Corollary 4.3.3) gives us a tail bound with exponential decay in t. To turn this into exponential decay in n, we have to exploit the observations' independence.

Theorem 4.3.5 (Hoeffding's inequality)**.** *For* iid *random variables $X_1, \ldots, X_n \in [-B, B]$ and all $t > 0$,*

$$\mathbb{P}\Big(\mathbb{E}[X] - \overline{X}_n \geq t\Big) \leq \exp\left(-\frac{nt^2}{2B^2}\right).$$

Proof. For any $\lambda > 0$, the Chernoff bound gives

$$\begin{aligned}
\mathbb{P}\Big(\overline{X}_n - \mathbb{E}[X] \geq t\Big) &\leq e^{-\lambda t}\mathbb{E}\Big[e^{\lambda(\overline{X}_n - \mathbb{E}[X])}\Big] \\
&= e^{-\lambda t}\mathbb{E}\Big[e^{\sum_{i=1}^n \lambda(X_i - \mathbb{E}[X])/n}\Big] \\
&= e^{-\lambda t}\mathbb{E}\left[\prod_{i=1}^n e^{\lambda(X_i - \mathbb{E}[X])/n}\right] \\
&= e^{-\lambda t}\prod_{i=1}^n \mathbb{E}\Big[e^{\lambda(X_i - \mathbb{E}[X])/n}\Big] \\
&= e^{-\lambda t}\mathbb{E}\Big[e^{\lambda(X - \mathbb{E}[X])/n}\Big]^n,
\end{aligned}$$

where we used independence and identical distribution of $X_1, \ldots, X_n$ in the last two steps. The random variable $Z = (X - \mathbb{E}[X])/n$ is bounded by B/n. Using

Hoeffding's lemma (Lemma 4.3.4), we obtain

$$\mathbb{E}\Big[e^{\lambda(X_i-\mathbb{E}[X])/n}\Big]^n \leq \exp\left(\frac{\lambda^2(B/n)^2}{2}\right)^n = \exp\left(\frac{\lambda^2 B^2}{2n}\right).$$

Substituting in the previous display gives

$$\mathbb{P}(\overline{X}_n - \mathbb{E}[X] \geq t) \leq \exp\left(-\lambda t + \lambda^2 \frac{B^2}{2n}\right).$$

Since λ was arbitrary, we can choose the one that minimizes the right-hand side. The exponential function is increasing, so we can simply minimize its exponent. Setting its derivative to zero gives

$$-t + \frac{\lambda B^2}{n} = 0 \qquad \Leftrightarrow \qquad \lambda = \frac{nt}{B^2}.$$

Substituting this value yields

$$\begin{aligned}\mathbb{P}(\overline{X}_n - \mathbb{E}[X] \geq t) &\leq \exp\left(-\frac{nt}{B^2}t + \left[\frac{nt}{B^2}\right]^2 \frac{B^2}{2n}\right) \\ &= \exp\left(-\frac{nt^2}{B^2} + \frac{nt^2}{2B^2}\right) \\ &= \exp\left(-\frac{nt^2}{2B^2}\right).\end{aligned}$$

This also implies

$$\mathbb{P}\Big(\mathbb{E}[X] - \overline{X}_n \geq t\Big) = \mathbb{P}\Big(\overline{-X_n} - \mathbb{E}[-X] \geq t\Big) \leq \exp\left(-\frac{nt^2}{2B^2}\right). \qquad \square$$

Let's see how Hoeffding's inequality helps us understand statistical learning. Combining it with the union bound (4.1) yields

$$\begin{aligned}\mathbb{P}\left\{\sup_{h\in\mathcal{H}} R(h) - R_n(h) > t\right\} &= \mathbb{P}\left\{\sup_{h\in\mathcal{H}} [P_n(L\circ h) - P(L\circ h)] > t\right\} \\ &\leq |\mathcal{H}| \max_{h\in\mathcal{H}} \mathbb{P}\left\{P_n(L\circ h_i) - P(L\circ h_i) > t\right\} \\ &\leq |\mathcal{H}| \exp\left(-\frac{nt^2}{2B^2}\right).\end{aligned}$$

Setting the right hand side equal to δ and solving gives $t = B\sqrt{2\ln(|\mathcal{H}|/\delta)/n}$. Thus:

$$\sup_{h\in\mathcal{H}} |R(h) - R_n(h)| \leq B\sqrt{\frac{2\ln(|\mathcal{H}|) + 2\ln(1/\delta)}{n}} \quad \text{w.p. at least } 1-\delta.$$

We have proven Theorem 4.1.1. The tools acquired along the way prepare us for the next step: infinite classes.

4.3.4. Sub-Gaussian random variables*

The only place where we use boundedness of X in the proof is Hoeffding's lemma. Instead, we could directly assume that

$$\mathbb{E}\left[e^{\lambda(X-\mathbb{E}[X])}\right] \leq \exp\left(\frac{\lambda^2\sigma^2}{2}\right). \tag{4.2}$$

Random variables satisfying this inequality are called **Sub-Gaussian** with parameter σ^2. For $X \in \mathcal{N}(\mu, \sigma^2)$, (4.2) holds with equality. Other X satisfying (4.2) then have similar or lighter tails than $\mathcal{N}(\mu, \sigma^2)$.

Hoeffding's lemma shows that random variables with $|X| \leq B$ are Sub-Gaussian with parameter $\sigma^2 = B^2$. For general Sub-Gaussian variables, Hoeffding's inequality would read

$$\mathbb{P}\left(\overline{X}_n - \mathbb{E}[X] \geq t\right) \leq \exp\left(-\frac{nt^2}{2\sigma^2}\right).$$

In principle, we could replace almost all our assumptions on boundedness with Sub-Gaussianity. This still excludes random variables with heavier tails, though. We'll maybe discuss how to deal with those a bit later, but continue with boundedness for simplicity.

5. Bounds for infinite hypothesis classes

The bounds we derived in the previous chapter were educational but unrealistic. There are hardly any interesting ML algorithms where the hypothesis class $\mathcal{H}$ is finite. We nevertheless learned that concentration of measure enables generalization and know how to bound deviation probabilities. We'll also need this when there are infinitely many hypotheses. Our key issue is that the size $|\mathcal{H}|$ is no longer a sensible measure for $\mathcal{H}$'s capacity.

Key to our treatment of the finite case was the concentration of measure phenomenon. Hoeffding's inequality gives a tight bound on the concentration of the sample average $\overline{X}_n$ around its expectation $\mathbb{E}[X]$. The concentration of measure phenomenon is more general, however. In particular, with probability at least $1-\delta$:

$$\sup_{h\in\mathcal{H}}\Big[R(h)-R_n(h)\Big] \le \mathbb{E}\left[\sup_{h\in H}\Big[R(h)-R_n(h)\Big]\right] + B\sqrt{\frac{2\ln(1/\delta)}{n}}. \tag{5.1}$$

The supremum on the left concentrates around its expectation with radius $O(1/\sqrt{n})$. The expectation is a deterministic quantity. We'll develop several meaningful complexity measures $\mathcal{C}(\mathcal{H})$ by finding upper bounds to the expectation. In particular, we'll show

$$\mathbb{E}\left[\sup_{h\in H}\Big[R(h)-R_n(h)\Big]\right] \lesssim \frac{\mathcal{C}(\mathcal{H})}{\sqrt{n}}.$$

Each measure $\mathcal{C}(\mathcal{H})$ provides its own view of the complexity of a function class. In particular, we'll develop three complexity measures that give successively cruder bounds: Rademacher complexity, covering entropy, and VC-dimension.

5.1. McDiarmid's inequality

To prove (5.1), we need a generalization of Hoeffding's inequality. Concentration of measure also appears for other functions $g(X_1,\dots,X_n)$ of independent random variables.

Definition 5.1.1. *A function $g\colon \mathbb{R}^n \to \mathbb{R}$ satisfies the* ***bounded difference condition*** *on a set S, if for all $x_1,\dots,x_n,x_1',\dots,x_n' \in S$, there exists a constant*

$M < \infty$ such that

$$|g(x_1, \dots, x_n) - g(x_1, \dots, x_{i-1}, x_i', x_{i+1}, \dots, x_n)| \leq M.$$

The condition only allows functions that aren't too sensitive to individual inputs. For example, this excludes functions that explode on S.

The following result shows that $g(X_1, \dots, X_n)$ concentrates around its expectation.

Theorem 5.1.2 (McDiarmid's inequality)**.** *Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a function satisfying the bounded difference condition from Definition 5.1.1 with constant M. For* iid *random variables $X_1, \dots, X_n \in S$ and all $t > 0$, it holds*

$$\mathbb{P}\left(\mathbb{E}[g(X_1, \dots, X_n)] - g(X_1, \dots, X_n) \geq t\right) \leq \exp\left(-\frac{2t^2}{nM^2}\right).$$

Proof sketch, optional. The proof is similar to Hoeffding's inequality. We start with

$$\mathbb{P}(g(X_1, \dots, X_n) - \mathbb{E}[g(X_1, \dots, X_n)] \geq t) \leq e^{-\lambda t}\mathbb{E}\left[e^{\lambda(g(X_1,\dots,X_n) - \mathbb{E}[g(X_1,\dots,X_n)])}\right].$$

Defining

$$V_i = \mathbb{E}[g(X_1, \dots, X_n) \mid X_1, \dots, X_i] - \mathbb{E}[g(X_1, \dots, X_n) \mid X_1, \dots, X_{i-1}],$$

we can write our quantity of interest as a telescoping sum:

$$g(X_1, \dots, X_n) - \mathbb{E}[g(X_1, \dots, X_n)] = \sum_{i=1}^{n} V_i.$$

Thus,

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda(g(X_1,\dots,X_n) - \mathbb{E}[g(X_1,\dots,X_n)])}\right] &= \mathbb{E}\left[e^{\lambda \sum_{i=1} V_i}\right] \\
&= \mathbb{E}\left[e^{\lambda V_n} e^{\lambda \sum_{i=1}^{n-1} V_i}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[e^{\lambda V_n} \mid X_1, \dots, X_{n-1}\right] e^{\lambda \sum_{i=1}^{n-1} V_i}\right].
\end{aligned}$$

The last equality follows from the law of iterated expecations and the fact that $V_{n-1}, \dots, V_1$ do not depend on X_n. Note that $\mathbb{E}[V_i] = 0$ and $|V_i| \leq M$ because g has bounded differences. Hoeffding's lemma gives

$$\mathbb{E}\left[e^{\lambda V_n} \mid X_1, \dots, X_{n-1}\right] \leq e^{-\lambda^2 M^2/8}.$$

Applying the conditioning trick iteratively gives

$$E\left[e^{\lambda \sum_{i=1}^{n} V_i}\right] \leq e^{-\lambda^2 M^2 n/8}.$$

Now proceed as in the proof of Hoeffding's inequality. □

It remains to show that $g(Z_1, \dots, Z_n) = \sup_{h \in \mathcal{H}} |R(h) - R_n(h)|$ satisfies the bounded difference property (Definition 5.1.1). This is left as an exercise.

Theorem 5.1.3. *Define $B = \sup_{z\in\mathcal{Z},\ell\in\mathcal{L}} |\ell(z)| < \infty$. For all $\delta > 0$, it holds with probability at least $1-\delta$,*

$$\sup_{h\in\mathcal{H}}[R(h) - R_n(h)] - \mathbb{E}\left[\sup_{h\in\mathcal{H}}[R(h) - R_n(h)]\right] \leq B\sqrt{\frac{2\ln(1/\delta)}{n}}.$$

Theorem 5.1.3 shows that, with high probability,

$$\sup_{h\in\mathcal{H}} |R(h) - R_n(h)| \leq \mathbb{E}\left[\sup_{h\in\mathcal{H}}[R(h) - R_n(h)]\right] + O(n^{-1/2}).$$

The expectation on the right is a deterministic quantity, which makes our lives a bit easier. But it is not a very useful measure for the complexity of $\mathcal{H}$. We don't know much about it theoretically and it's impossible to estimate without knowing the true probability measure P. In what follows we derive and discuss complexity measures $\mathcal{C}(\mathcal{H})$ that bound this expectation.

5.2. Rademacher complexity

5.2.1. Definition and derivation

The expectation in Theorem 5.1.3 is intimately linked to a quantity called *Rademacher complexity*.

Definition 5.2.1. *The **Rademacher complexity** of a function class $\mathcal{F}$ is*

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(Z_i)\ \right],$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are iid Rademacher variables *with $\mathbb{P}(\varepsilon_i = 1) = \mathbb{P}(\varepsilon_i = -1) = 1/2$ independent of $(Z_i)_{i=1}^n$.*

Remark 5.2.2. *The Rademacher complexity is sometimes defined with an absolute value inside the supremum. The two versions are practically equivalent. Indeed,*

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(Z_i)\right] \leq \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(Z_i)\right|\right] = \mathbb{E}\left[\sup_{f\in\mathcal{F}\cup-\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(Z_i)\right].$$

Pretty much all of the following results can be formulated for both versions of the complexity.

The Rademacher complexity arises from mathematical trickery, called *symmetrization argument*. We introduce a hypothetical "ghost sample" $\mathcal{D}'_n = (Z'_i)_{i=1}^n$. The

$Z_1', \dots, Z_n'$ are independent of $Z_1, \dots, Z_n$, but have the same distribution. Then

$$\mathbb{E}[f(Z)] = \mathbb{E}_{Z'}\left[\frac{1}{n}\sum_{i=1}^{n} f(Z_i')\right],$$

where $\mathbb{E}_{Z'}$ denotes expectation with respect to $\mathcal{D}_n'$ only. Now we can write

$$\mathbb{E}[f(Z)] - \frac{1}{n}\sum_{i=1} f(Z_i) = \mathbb{E}_{Z'}\left[\frac{1}{n}\sum_{i=1}\Big(f(Z_i') - f(Z_i)\Big)\right].$$

Together this gives

$$\begin{aligned}\mathbb{E}\left[\sup_{f\in\mathcal{F}} \mathbb{E}[f(Z)] - \frac{1}{n}\sum_{i=1} f(Z_i)\right] &= \mathbb{E}_Z\left[\sup_{f\in\mathcal{F}} \mathbb{E}_{Z'}\left[\frac{1}{n}\sum_{i=1}(f(Z_i) - f(Z_i'))\right]\right] \\ &\le \mathbb{E}_Z\mathbb{E}_{Z'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}(f(Z_i) - f(Z_i'))\right],\end{aligned}$$

by Jensen's inequality. The term $\Big(f(Z_i') - f(Z_i)\Big)$ has the same distribution as $\varepsilon_i\Big(f(Z_i') - f(Z_i)\Big)$, because of independence of Z_i and Z_i' and symmetry. Because additionally $Z_i \overset{d}{=} Z_i'$, the double expectation in the last display equals

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}\varepsilon_i(f(Z_i) - f(Z_i'))\right] \le 2\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(Z_i)\right] = 2\mathcal{R}_n(\mathcal{F}).$$

We have shown that twice the Rademacher complexity is an upper bound to $\mathbb{E}[\sup_{f\in\mathcal{F}} P(f) - P_n(f)]$. In fact, the Rademacher complexity also shows up in a lower bound.

Theorem 5.2.3. *For any function class $\mathcal{F}$ with $\sup_{f\in\mathcal{F}, z\in\mathcal{Z}} |f(z)| \le B$,*

$$\frac{1}{2}\mathcal{R}_n(\mathcal{F}) - \frac{B}{\sqrt{n}} \le \mathbb{E}\left[\sup_{f\in\mathcal{F}}[P(f) - P_n(f)]\right] \le 2\mathcal{R}_n(\mathcal{F}).$$

Proof (optional). It remains to prove the lower bound. Adding and subtracting $\mathbb{E}[f(Z)]$ and the triangle inequality give

$$\begin{aligned}\mathcal{R}_n(\mathcal{F}) &= \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(Z_i)\Big] \\ &= \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\big(f(Z_i) - \mathbb{E}[f(Z)]\big) + \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbb{E}[f(Z)]\Big] \\ &\le \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\big(f(Z_i) - \mathbb{E}[f(Z)]\big)\Big] + \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbb{E}[f(Z)]\Big].\end{aligned}$$

Using the same arguments as in the proof of the upper bound, we can show that

$$\mathbb{E}\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\big(f(Z_i)-\mathbb{E}[f(Z)]\big)\Big]\leq 2\mathbb{E}\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}f(Z_i)-\mathbb{E}[f(Z)]\Big].$$

Furthermore,

$$\begin{aligned}\mathbb{E}\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbb{E}[f(Z)]\Big]^2 &\leq B^2\mathbb{E}\Big[\Big|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\Big|\Big]^2\\ &\leq \frac{B^2}{n^2}\mathbb{E}\Big[\Big(\sum_{i=1}^{n}\varepsilon_i\Big)^2\Big] && \text{(Jensen's inequality)}\\ &= \frac{B^2}{n^2}\mathbb{E}\Big[\sum_{i=1}^{n}\varepsilon_i^2\Big]+\frac{B}{n}\mathbb{E}\Big[\sum_{i\neq j}\varepsilon_i\varepsilon_j\Big]\\ &= \frac{B^2}{n},\end{aligned}$$

because $\mathbb{E}[\varepsilon_i^2]=1$ and the ε_i are pairwise independent. Altogether, we have

$$\mathcal{R}_n(\mathcal{F})\leq 2\mathbb{E}\Big[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}f(Z_i)-\mathbb{E}[f(Z)]\Big]+\frac{B}{\sqrt{n}}.$$

Rearranging terms proves our claim. □

Let's apply this to the statistical learning context. Recall the definition of the loss class

$$\mathcal{L}=L\circ\mathcal{H}=\{L(\cdot,h(\cdot))\colon h\in\mathcal{H}\}$$

and

$$\sup_{h\in\mathcal{H}}[R(h)-R_n(h)]=\sup_{\ell\in\mathcal{L}}[P(\ell)-P_n(\ell)].$$

Theorem 5.2.3 together with Theorem 5.1.3 gives:

With high probability,

$$\frac{1}{2}\mathcal{R}_n(\mathcal{L})-O(n^{-1/2})\leq\sup_{h\in\mathcal{H}}[R(h)-R_n(h)]\leq 2\mathcal{R}_n(\mathcal{L})+O(n^{-1/2}).$$

The upper bounds shows that the Rademacher complexity is indeed a useful complexity measure for statistical learning. If the complexity is small, also the generalization gap must be small. The link is even stronger than that: the generalization gap is also bounded from below by the Rademacher complexity. As a consequence,

$$\sup_{h\in\mathcal{H}}\Big[R(h)-R_n(h)\Big]\to_p 0\qquad\Leftrightarrow\qquad\mathcal{R}_n(\mathcal{L})\to 0.$$

This shows that the Rademacher is a fundamental complexity measure. Let's formally state the upper bound for the generalization gap for future reference.

Theorem 5.2.4 (Rademacher generalization bound). *Define* $B = \sup_{z\in\mathcal{Z},\ell\in\mathcal{L}} |\ell(z)| < \infty$. *For all* $\delta > 0$, *it holds*

$$\sup_{h\in\mathcal{H}}[R(h) - R_n(h)] \leq 2\mathcal{R}_n(\mathcal{L}) + B\sqrt{\frac{2\ln(1/\delta)}{n}} \qquad \text{w.p. at least } 1-\delta.$$

Proof. Follows immediately from Theorem 5.1.3 and Theorem 5.2.3. □

5.2.2. Interpretation and properties

The Rademacher complexity is a probabilistic measure of the complexity of the loss class $\mathcal{L} = L \circ \mathcal{H}$. There is randomness from the unknown measure P and additional randomness from the Rademacher variables $\varepsilon_1, \ldots, \varepsilon_n$. It's a bit annoying that we're forced to think about the loss class $\mathcal{L}$ instead of the hypothesis class $\mathcal{H}$ however. Let's change that.

Lemma 5.2.5 (Talagrand's contraction lemma). *For any hypothesis set* $\mathcal{H}$ *and loss function* L *with*

$$|L(z, h(z)) - L(z, h'(z))| \leq c|h(z) - h'(z)| \quad \text{for all } h, h' \in \mathcal{H}, z \in \mathcal{Z}, \tag{5.2}$$

it holds

$$\mathcal{R}_n(\mathcal{L}) = \mathcal{R}_n(L \circ \mathcal{H}) \leq c\mathcal{R}_n(\mathcal{H}).$$

Proof (optional). Fix a sample $(z_1, \ldots, z_n)$ and define the set

$$A = \{a \in \mathbb{R}^n : a_i = h(z_i) \text{ for some } \ell \in \mathcal{L}'\}.$$

It holds

$$\begin{aligned}
\mathcal{R}_n(\mathcal{L}) &= \mathbb{E}\Big[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i L(z_i, h(z_i))\Big] \\
&= \mathbb{E}\Big[\sup_{a\in A} \frac{1}{n}\sum_{i=1}^n \varepsilon_i L(z_i, a_i)\Big] \\
&= \frac{1}{2}\mathbb{E}\Big[\sup_{a,a'\in A} \frac{1}{n}\sum_{i=1}^n \varepsilon_i L(z_i, a_i) - \frac{1}{n}\sum_{i=1}^n \varepsilon_i L(z_i, a_i')\Big] && (\varepsilon_i \overset{d}{=} -\varepsilon_i) \\
&\leq \frac{1}{2}\mathbb{E}\Big[\sup_{a,a'\in A} |L(z_i, a_1) - L(z_i, a_1')| + \frac{1}{n}\sum_{i=2}^n \varepsilon_i\big(L(z_i, a_i) - L(z_i, a_i')\big)\Big] \\
&\leq \frac{1}{2}\mathbb{E}\Big[\sup_{a,a'\in A} c|a_1 - a_1'| + \frac{1}{n}\sum_{i=2}^n \varepsilon_i\big(L(z_i, a_1) - L(z_i, a_1')\big)\Big] && (L \text{ Lipschitz}) \\
&\leq \frac{1}{2}\mathbb{E}\Big[\sup_{a,a'\in A} c(a_1 - a_1') + \frac{1}{n}\sum_{i=2}^n \varepsilon_i\big(L(z_i, a_1) - L(z_i, a_1')\big)\Big] \\
&= \mathbb{E}\Big[c\sup_{a\in A} \varepsilon_1 a_1 + \frac{1}{n}\sum_{i=2}^n \varepsilon_i L(z_i, a_i)\Big].
\end{aligned}$$

Repeating the argument for $i = 2, \dots, n$ proves the claim. □

In most cases, Talagrand's lemma allows us to replace the loss complexity $\mathcal{L}$ with the hypothesis complexity $\mathcal{R}_n(\mathcal{H})$. For example, one can verify that (5.2) holds with

- $c = 1/2$ for the 0-1-loss,
- $c = 1$ for the hinge and logistic loss,
- $c = 2\sqrt{B}$ for the square loss,
- $c = 1$ for the pinball loss,
- $c = 1/\inf_{z \in \mathcal{Z}, h \in \mathcal{H}} h(z)$ for the log-loss.

So let's focus on the Rademacher complexity of the hypothesis class:

$$\mathcal{R}_n(\mathcal{H}) = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i h(Z_i)\right].$$

Intuitively, it measures how well the hypothesis class $\mathcal{H}$ can 'match' random noise — or, more pessimistically: how likely we are fooled by random noise. If $\mathcal{H}$ is small, this is very unlikely.

Example 5.2.6 (Single hypothesis)**.** *If $|\mathcal{H}| = 1$, we have $\mathcal{R}_n(\mathcal{H}) = 0$ (exercise).*

$$\mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i h(Z_i)\right] = 0.$$

$$\mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i h(Z_i)\right] = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i h(Z_i)\right] = \mathbb{E}[\varepsilon_i]\mathbb{E}[h(Z_i)] = 0,$$

using that $(\varepsilon_i)_{i=1}^n$ is independent from $(Z_i)_{i=1}^n$ and $\mathbb{E}[\varepsilon_i] = 0$.

If $\mathcal{H}$ is too large, we can fit almost any pattern and are prone to be fooled by noise. This interpretation is most obvious for classification.

Example 5.2.7 (Binary classification)**.** *For $h \colon \mathcal{Z} \to \{-1, 1\}$, we have*

$$\begin{aligned}
\mathcal{R}_n(\mathcal{H}) = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i h(Z_i)\right] &= 1 + \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1} [\varepsilon_i h(Z_i) - 1]\right] \\
&= 1 + \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1} -2\mathbb{1}\{\varepsilon_i \neq h(Z_i)\}\right] \\
&= 1 - 2\mathbb{E}\left[\min_{h \in \mathcal{H}} R_n^{\varepsilon}(h)\right],
\end{aligned}$$

where R_n^ε is the empirical 0-1-risk on a data set with completely random labels ε_i. If $\mathcal{H}$ is so large that it can fit any random label pattern, then $\mathbb{E}[\min_{h\in\mathcal{H}} R_n^\varepsilon(h)] = 0$. Then $\mathcal{R}_n(\mathcal{H}) = 1$ and we cannot expect to generalize well. If $\mathcal{H}$ is very small, however, the expected empirical risk $\min_{h\in\mathcal{H}} R_n^\varepsilon(h)$ will be close to $1/2$ and $\mathcal{R}_n(\mathcal{H}) \approx 0$, which gives a tight upper bound to the generalization gap.

The Rademacher complexity does not only depend on the size of $\mathcal{H}$, but also on the probability measure P. If P is nice, we may afford larger hypothesis classes.

Example 5.2.8. *Consider the extreme case, where P is a point mass at $z_0 \in \mathcal{Z}$, i.e., $P\{Z = z_0\} = 1$. It holds*

$$\mathcal{R}_n(\mathcal{H}) = \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i h(Z_i)\right] = \mathbb{E}\left[\sup_{h\in\mathcal{H}} h(z_0)\frac{1}{n}\sum_{i=1}^n \varepsilon_i\right] \leq \sup_{h\in\mathcal{H}} |h(z_0)| \times \mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i\right|\right],$$

and

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i\right|\right] \leq \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \varepsilon_i\right)^2\right]^{1/2} = \frac{1}{\sqrt{n}},$$

where we used that ε_i iid with $\mathbb{E}[\varepsilon_i] = 0, \mathbb{V}\mathrm{ar}[\varepsilon_i] = 1$. As you can see, $\mathcal{R}_n(\mathcal{H})$ does not depend on the number of hypotheses in $\mathcal{H}$ at all.

Before we see how to compute the Rademacher complexity for more interesting examples, let's mention a few other intuitive properties of $\mathcal{R}_n$ that follow directly from the definition.

Proposition 5.2.9.

1. *$\mathcal{R}_n(\mathcal{F}) \leq \mathcal{R}_n(\mathcal{F}')$ if $\mathcal{F} \subset \mathcal{F}'$.*
2. *$\mathcal{R}_n(\mathcal{F} + \mathcal{F}') = \mathcal{R}_n(\mathcal{F}) + \mathcal{R}_n(\mathcal{F}')$.*
3. *$\mathcal{R}_n(f + c\mathcal{F}) = |c|\mathcal{R}_n(\mathcal{F})$ for any $c \in \mathbb{R}$ and function f.*
4. *$\mathcal{R}_n(\mathrm{conv}\mathcal{F}) = \mathcal{R}_n(\mathcal{F})$ with the* convex hull *of $\mathcal{F}$ defined as*

$$\mathrm{conv}\mathcal{F} = \left\{f = \sum_k \lambda_k f_k \colon f_k \in \mathcal{F}, \sum_k |\lambda_k| = 1\right\}.$$

Lastly, we state a lemma for Rademacher complexities of finite sets. It will prove essential for computing bounds for infinite sets as well.

Lemma 5.2.10 (Massart's lemma). *For any $A \subset \mathbb{R}^n$ with $\sup_{a\in A} \|a\|_2 \leq r\sqrt{n}$, it holds*

$$\mathbb{E}\left[\max_{a\in A} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i a_i\right] \leq r\sqrt{\frac{2\ln(|A|)}{n}}.$$

Proof (optional). For any $\lambda > 0$, we have

$$\begin{aligned}
\mathbb{E}\left[\max_{a\in A'} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i a_i\right] &= \frac{1}{n\lambda}\mathbb{E}\left[\max_{a\in A} \lambda\sum_{i=1}^{n} \varepsilon_i a_i\right] \\
&\leq \frac{1}{n\lambda}\ln \mathbb{E}\left[\max_{a\in A'} e^{\lambda\sum_{i=1}^{n}\varepsilon_i a_i}\right] && \text{(Jensen)} \\
&\leq \frac{1}{n\lambda}\ln \mathbb{E}\left[\sum_{a\in A} e^{\lambda\sum_{i=1}^{n}\varepsilon_i a_i}\right] && (\max_a \leq \textstyle\sum_a) \\
&= \frac{1}{n\lambda}\ln \mathbb{E}\left[\sum_{a\in A}\prod_{i=1}^{n} e^{\lambda\varepsilon_i a_i}\right] \\
&= \frac{1}{n\lambda}\ln \sum_{a\in A}\prod_{i=1}^{n} \mathbb{E}\left[e^{\lambda\varepsilon_i a_i}\right] && (\varepsilon_i \text{ independent}) \\
&= \frac{1}{n\lambda}\ln \sum_{a\in A}\prod_{i=1}^{n} \frac{e^{\lambda a_i} + e^{-\lambda a_i}}{2} && (\mathbb{P}(\varepsilon_i = 1) = \mathbb{P}(\varepsilon_i = -1) = 1/2) \\
&\leq \frac{1}{n\lambda}\ln \sum_{a\in A}\prod_{i=1}^{n} e^{\lambda^2 a_i^2/2} && (e^x + e^{-x} \leq 2e^{x^2/2}) \\
&\leq \frac{1}{n\lambda}\ln \sum_{a\in A} e^{\lambda^2\|a\|^2/2} && (\|a\|^2 = \textstyle\sum_i a_i^2) \\
&\leq \frac{1}{n\lambda}\ln\left(|A|\max_{a\in A} e^{\lambda^2\|a\|^2/2}\right) && (\textstyle\sum_a \leq |A|\max_a) \\
&= \frac{1}{n\lambda}\left(\ln|A| + \max_{a\in A}\lambda^2\|a\|^2/2\right) \\
&= \frac{\ln|A|}{n\lambda} + \frac{\lambda r^2}{2}. && (\|a\| \leq r\sqrt{n})
\end{aligned}$$

This is minimized for $\lambda = \sqrt{2\ln|A|/r^2 n}$. Substituting this value in the last display proves our claim. □

As a corollary, we get that for finite $\mathcal{H}$,

$$\mathcal{R}_n(\mathcal{H}) \leq \sup_{h,z}|h(z)|\sqrt{\frac{2\ln|\mathcal{H}|}{n}},$$

which leads to similar generalization bounds as in Section 4.3.3.

5.2.3. Empirical Rademacher complexity

The Rademacher complexity depends on the distribution $P \sim Z$. For a given choice of P, we can compute the complexity via Monte-Carlo simulation. For the learning problems we encounter in practice, we don't know P however. Interestingly, we can still estimate the Rademacher complexity from the training data.

Definition 5.2.11. *Given a training set $\mathcal{D}_n = (Z_i)_{i=1}^n$ **empirical Rademacher complexity** of a function class $\mathcal{F}$ is*

$$\widehat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)\right].$$

The empirical Rademacher complexity takes expectation with respect to $\varepsilon_1, \dots, \varepsilon_n$ only. It is still a random variable, because the result depends on $\mathcal{D}_n$. Its expectation is the usual Rademacher complexity, since

$$\mathbb{E}[\widehat{\mathcal{R}}_n(\mathcal{F})] = \mathbb{E}_Z\mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)\right] = \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)\right] = \mathcal{R}_n(\mathcal{F}).$$

The empirical complexity concentrates around its expectation. This follows from a straightforward application of McDiarmid's inequality.

Theorem 5.2.12. *If $\sup_{z\in\mathcal{Z}, f\in\mathcal{F}} |f(z)| \le B$, it holds*

$$\mathcal{R}_n(\mathcal{F}) - \widehat{\mathcal{R}}_n(\mathcal{F}) \le B\sqrt{\frac{2\ln(1/\delta)}{n}} \qquad \text{w.p. at least } 1-\delta.$$

We can substitute this into generalization bounds. Aggregating terms and probabilities from Theorem 5.2.4 and Theorem 5.2.12 leads to

$$\sup_{h\in\mathcal{H}} R(h) - R_n(h) \le \widehat{\mathcal{R}}_n(\mathcal{L}) + 3B\sqrt{\frac{2\ln(1/\delta)}{n}}.$$

Now the right-hand side does not depend on P and can, in principle, be estimated from the training data:

Algorithm

1. For $k = 1, \dots, K$:
   (i) simulate Rademacher variables $\varepsilon_1, \dots, \varepsilon_n$,
   (ii) compute $\mathcal{R}_k = \max_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)$,
2. Compute $\widehat{\mathcal{R}}_n(\mathcal{F}) \approx K^{-1}\sum_{k=1}^K \mathcal{R}_k$.

In practice, cross-validation gives much better estimates of the generalization gap (because it does not involve bounds). The empirical complexity can sometimes be useful for studying Rademacher complexities itself, however.

5.3. Applications

5.3.1. Penalized linear models

To illustrate how the Rademacher complexity helps us understand statistical learning, we start with a simple model class. Let $\mathcal{Z} = \mathbb{R}^d$ and consider the class of norm-constrained linear functions

$$\mathcal{H}_{q,M} = \{z \mapsto \beta^\top z \colon \|\beta\|_q \leq M\}.$$

Such classes appear natural in penalized regression problems. Using Lagrange duality, there is $\lambda \geq 0$ such that

$$\beta^* = \arg\min_{\|\beta\|_q \leq M} R_n(\beta) \quad \Leftrightarrow \quad \beta^* = \arg\min_{\beta \in \mathbb{R}^d} R_n(\beta) + \lambda \|\beta\|_q^q.$$

In particular, for the square-loss and $q = 2$ this corresponds to *ridge regression*, for $q = 1$ to the *lasso*. Finding the exact λ that corresponds to a given M is difficult, but that's not important qualitatively. As the penalty λ increases, the bound M decreases and vice versa. We get the following result.

Theorem 5.3.1. *It holds*

$$\mathcal{R}_n(\mathcal{H}_{1,M_1}) \leq M_1 \sup_{z \in \mathcal{Z}} \|z\|_\infty \sqrt{\frac{2\ln(2d)}{n}} \quad \text{and} \quad \mathcal{R}_n(\mathcal{H}_{2,M_2}) \leq M_2 \sqrt{\frac{\mathbb{E}[\|Z\|_2^2]}{n}}$$

Proof. We start with a preliminary bound that applies to both $q = 1$ and $q = 2$. Compute

$$\begin{aligned}
\mathcal{R}_n(\mathcal{H}_{q,M}) &= \mathbb{E}\left[\sup_{\|\beta\|_q \leq M} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \beta^\top Z_i\right] \\
&= \frac{1}{n} \mathbb{E}\left[\sup_{\|\beta\|_q \leq M} \beta^\top \sum_{i=1}^n \varepsilon_i Z_i\right] \\
&\leq \frac{1}{n} \mathbb{E}\left[\sup_{\|\beta\|_q \leq M} \|\beta^\top\|_q \left\|\sum_{i=1}^n \varepsilon_i Z_i\right\|_{q/(q-1)}\right] \qquad \text{(Hölder's inequality)} \\
&\leq \frac{M}{n} \mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i Z_i\right\|_{q/(q-1)}\right],
\end{aligned}$$

where $\|x\|_\infty = \max_{1 \leq j \leq d} |x^{(j)}|$. The expectation on the right does not depend on $\mathcal{H}_{q,M}$, only on q and the distribution of Z_i. For $q = 2$, we have $q/(q-1) = 2$, and a computation similar to Example 5.2.8 gives

$$\mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i Z_i\right\|_2\right]^2 \leq \mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i Z_i\right\|_2^2\right] = \sum_{j=1}^d \mathbb{E}\left[\left|\sum_{i=1}^n \varepsilon_i Z_i^{(j)}\right|^2\right] \leq n \sum_{j=1}^d \mathbb{E}[|Z_i^{(j)}|^2] = n\mathbb{E}[\|Z_i\|_2^2].$$

This proves the second inequality. For $q = 1$ see the exercises. For $q = 1$, we have $q/(q-1) = \infty$ and write

$$\mathbb{E}\left[\left\|\sum_{i=1}^{n} \varepsilon_i Z_i\right\|_\infty\right] = \mathbb{E}\left[\max_{1\le j\le d}\left|\sum_{i=1}^{n} \varepsilon_i Z_i^{(j)}\right|\right].$$

Condition on $\mathcal{D}_n = (Z_i)_{i=1}^n$, define $A = \bigcup_{j=1}^d \{(Z_i^{(j)})_{i=1}^n\}$ and note that $|A| = d$ and $\max_{a\in A} \|a\|_2 \le \sqrt{n} \max_{a\in A} \|a\|_\infty$. Apply Massart's lemma (Lemma 5.2.10) to get

$$\begin{aligned}\mathbb{E}_\varepsilon\left[\max_{1\le j\le d}\left|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i Z_i^{(j)}\right|\right] = \mathbb{E}_\varepsilon\left[\max_{a\in A}\left|\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i a_i\right|\right] &= \mathbb{E}_\varepsilon\left[\max_{a\in A\cup -A}\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i a_i\right] \\ &\le \max_{a\in A}\|a\|_\infty \sqrt{\frac{2\ln|2d|}{n}}.\end{aligned}$$

Now the first claim follows from

$$\max_{a\in A}\|a\|_\infty = \max_{1\le j\le d}\max_{1\le i\le n}|Z_i^{(j)}| \le \sup_{z\in\mathcal{Z}}\|z\|_\infty. \qquad \square$$

With everything else fixed, both bounds decrease as $O(1/\sqrt{n})$ and increase linearly in M_q. Their main difference is the norms of β and Z — in particular, how they scale with the dimension d. The term $B = \sup_{z\in\mathcal{Z}}\|z\|_\infty$ does not depend on the dimension at all, but M_1, M_2, and $\mathbb{E}[\|Z\|_2^2]$ do.

A case for ridge regression

Assuming that all entries of a vector $x \in \mathbb{R}^d$ have a similar scale, we have $\|x\|_q = O(d^{1/q})$. In particular, we should expect $M_1 = O(d), M_2 = O(\sqrt{d})$, and $\sqrt{\mathbb{E}[\|Z\|_2^2]} = O(\sqrt{d})$. Plugging this into the bounds above gives

$$\mathcal{R}_n(\mathcal{H}_{1,M_1}) = O\left(d\sqrt{\frac{\log(2d)}{n}}\right) \quad \text{and} \quad \mathcal{R}_n(\mathcal{H}_{2,M_2}) = O\left(d\sqrt{\frac{1}{n}}\right).$$

The lasso complexity $\mathcal{R}_n(\mathcal{H}_{1,M_1})$ scales slightly worse then the ridge complexity $\mathcal{R}_n(\mathcal{H}_{2,M_2})$. If d is large compared to n, lasso may generalize slightly worse.

A case for the lasso

The situation changes if we assume β to be s-sparse (only having s nonzero entries). Define[1]

$$\mathcal{H}_{q,M,s} = \{z \mapsto \beta^\top z \colon \|\beta\|_q \le M, \|\beta\|_0 \le s\}.$$

[1]The '0-norm' $\|x\|_0$ is simply counting the non-zero elements of x.

Now $\|\beta\|_q = O(s^{1/q})$, and substituting gives $M_1 = O(s), M_2 = O(\sqrt{s})$, and

$$\mathcal{R}_n(\mathcal{H}_{1,M_1,s}) = O\left(s\sqrt{\frac{\log(2d)}{n}}\right) \quad \text{and} \quad \mathcal{R}_n(\mathcal{H}_{2,M_2,s}) = O\left(\sqrt{\frac{sd}{n}}\right).$$

If the sparsity is s is small compared to d, the lasso-complexity scales much better. So if we believe to find sparse patterns, we should prefer the lasso over ridge penalty.

The approximation-generalization trade-off

The Rademacher complexity bounds the statistical part of the risk, the estimation error. For the overall error, we also have to account for approximation error. Theorem 5.2.4 and Eq. (3.1) give

$$R(\widehat{h}) \leq \inf_{h\in\mathcal{H}} R(h) + 2\mathcal{R}_n(\mathcal{L}) + B\sqrt{\frac{2\ln(1/\delta)}{n}}. \tag{5.3}$$

The first term on the right is the best possible risk attainable by $\mathcal{H}$. It generally decreases with the capacity of $\mathcal{H}$. For example, suppose L is the square loss and $\widehat{\beta} = \arg\min_{\beta\in\mathcal{H}_{2,M_2}} R_n(\beta)$ the ridge-regression solution.

Proposition 5.3.2. *Let $\beta_0 = \arg\min_{\beta\in\mathbb{R}^n} R(\beta)$ be the optimal solution and $C^2 = \mathbb{E}[\|Z\|_2^2]$. Then with probability at least $1-\delta$,*

$$R(\widehat{\beta}) \leq \underbrace{R(\beta_0)}_{\text{optimal risk}} + \underbrace{C^2(\|\beta_0\|_2 - M_2)_+^2}_{\text{approximation error}} + \underbrace{\frac{4\sqrt{B}M_2C + B\sqrt{2\ln(1/\delta)}}{\sqrt{n}}}_{\text{generalization error}},$$

where $A_+ = \max(A, 0)$ denotes the positive part.

The optimal risk is out of our control. As long as $M_2 < \|\beta_0\|$, the class $\mathcal{H}_{2,M_2}$ does not include the optimal solution. In that case, the is approximation error is nonzero. To make it small, we want to make M_2 large. But this might hurt us in terms of the generalization error, which is of order $O(M_2/\sqrt{n})$.

Proof. By the contraction lemma (Lemma 5.2.5) and Theorem 5.3.1,

$$\mathcal{R}_n(\mathcal{L}) = \mathcal{R}_n(L \circ \mathcal{H}_{2,M_2}) \leq 2\sqrt{B}\mathcal{R}_n(\mathcal{H}_{2,M_2}) \leq \frac{2\sqrt{B}M_2C}{\sqrt{n}}.$$

Then

$$R(\widehat{\beta}) \leq \inf_{\|\beta\|\leq M_q} R(\beta) + \frac{4\sqrt{B}M_2C}{\sqrt{n}} + B\sqrt{\frac{2\ln(1/\delta)}{n}}.$$

Furthermore,

$$
\begin{aligned}
&\inf_{\|\beta\| \le M_q} R(\beta) - R(\beta_0) \\
&= \inf_{\|\beta\| \le M_q} \mathbb{E}[(Y - \beta^\top Z)^2 - (Y - \beta_0^\top Z)^2] \\
&= \inf_{\|\beta\| \le M_q} \mathbb{E}[-2Y(\beta^\top Z - \beta_0^\top Z) + (\beta^\top Z)^2 - (\beta_0^\top Z)^2] \\
&= \inf_{\|\beta\| \le M_q} \mathbb{E}[-2(\beta_0^\top Z + \epsilon)(\beta^\top Z - \beta_0^\top Z) + (\beta^\top Z)^2 - (\beta_0^\top Z)^2] \\
&= \inf_{\|\beta\| \le M_q} \mathbb{E}[-2(\beta_0^\top Z)(\beta^\top Z - \beta_0^\top Z) + (\beta^\top Z)^2 - (\beta_0^\top Z)^2] \qquad (\mathbb{E}[\epsilon \mid Z] = 0) \\
&= \inf_{\|\beta\| \le M_q} \mathbb{E}[(\beta^\top Z - \beta_0^\top Z)^2] \\
&\le C^2 \inf_{\|\beta\| \le M_q} \|\beta - \beta_0\|_2^2, \qquad \text{(Cauchy-Schwarz)} \\
&= C^2 (\|\beta_0\|_2 - M_2)_+^2. \qquad \square
\end{aligned}
$$

5.3.2. Interpreting bounds and learning from proofs

Now is a good time to emphasize that we are dealing with bounds, not equalities. Bounds can be loose and one more than another. A comparison of two bounds can lead to wrong conclusions. To make a stupid example, the following version of the first inequality of Theorem 5.3.1 is also true:

$$\mathcal{R}_n(\mathcal{H}_{1,M_1}) \le M_1 \sup_{z \in \mathcal{Z}} \|z\|_\infty d^{100}.$$

Proof: $\sqrt{\log(2d)} \le d^{100}$ and $1/n \le 1$. Now the bound scales terribly with d and does not decrease in n at all.

So how can we gain confidence in our bounds? The gold standard is to prove matching lower bounds, at least up to constants. In the learning context, this is typically much harder than proving upper bounds. There are some amazing results along that line and will probably touch on that topic a bit later.

A more manageable alternative is to check the steps of our proof. Proves of upper bounds consist of a chain of arguments, successively bounding terms in little steps. In each of the steps, we can check: how loose is this bound? How likely are edge cases where equality is attained? How much smaller is the term under likely/normal conditions?

In Theorem 5.3.1, we used quite similar arguments for the two bounds. The inequalities we used (Jensen, Hölder, norm bounds, Massart's lemma) are known to be sharp — at least there are edge cases that attain them. That should give us some confidence that the two bounds are comparable, at least in how they scale with d and n. That's why understanding mathematical proofs is so valuable. We learn about the forces that act and in which situations they are good or bad. It can also give us ideas on how to improve methods or invent new ones. In fact, most methods you know and use have arisen from a mathematical understanding

of the problem and forces that act.

There's no need to ponder hours about all the proofs in these notes and look for lower bounds at every step. Essentially all bounds can be improved for special cases or under additional assumptions. Statistical learning theory is not about finding the sharpest of all bounds, but about understanding the fundamentals of the problem.

5.3.3. Ensembles

The Rademacher complexity also gives interesting insights into ensemble methods. Let $\mathcal{W}$ be a simple hypothesis class. An ensemble algorithm learns composite hypotheses

$$\mathcal{H}_{K,\mathcal{W}} = \left\{ h = \sum_{k=1}^{K} \lambda_k h_k \colon h_1, \dots, h_k \in \mathcal{W}, \sum_{k=1}^{K} |\lambda_k| = 1 \right\}.$$

Example 5.3.3 (Random forest)**.** *A random forest uses decision trees as base class,*

$$\mathcal{W} = \left\{ h \colon \mathcal{X} \to \mathbb{R} \colon x \mapsto \sum_{j=1}^{J} \beta_j \mathbb{1}(x \in \mathcal{X}_k),\ \beta \in \mathcal{B} \subset \mathbb{R}^J, \mathcal{X} = \bigcup_{j=1}^{J} \mathcal{X}_j \right\},$$

where the $\mathcal{X}_j$ are axis-aligned rectangles. The ensemble members $\widehat{h}_k \in \mathcal{W}$ are trained independently on random subsets of the samples and features. The algorithm then outputs the final hypothesis

$$\widehat{h} = \frac{1}{K} \sum_{k=1}^{K} \widehat{h}_k \in \mathcal{H}_{K,\mathcal{W}}.$$

Example 5.3.4 (Boosting)**.** *Decision trees are often used in boosting methods, where $\mathcal{W}$ is called the class of weak learners. Now the weak learners $\widehat{h}_k$ are constructed in sequence, such that $\widehat{h}_k$ corrects errors made by $\widehat{h}_1, \dots, \widehat{h}_{k-1}$ and $\widehat{\lambda}_k$ are data-driven weights. After K iterations, the algorithm outputs the final hypothesis*

$$\widehat{h} = \sum_{k=1}^{K} \widehat{\lambda}_k \widehat{h}_k \in r\mathcal{H}_{K,\mathcal{W}},$$

where $r = \sum_{k=1}^{K} |\widehat{\lambda}_k|$.

By definition, $\mathcal{H}_{K,\mathcal{W}}$ lies in the convex hull of $\mathcal{W}$. Proposition 5.2.9 then implies

$$\mathcal{R}_n(\mathcal{H}_{K,\mathcal{W}}) = \mathcal{R}_n(\mathcal{W}).$$

No matter how many hypotheses we average, our generalization error bound

does not change. For boosting, the weights $|\widehat{\lambda}_k|$ don't usually sum to one. Often the sum grows only very slowly in K however. When that's the case, running a boosting algorithm for many rounds bears little statistical cost. (There is computational cost of course). This is in line with empirical findings. At least for boosting methods, running additional rounds decreases the approximation error

$$\inf_{h \in \mathcal{H}_{K,\mathcal{W}}} [R(h) - R(h_0)]$$

to some degree. For most popular choices of $\mathcal{W}$, one can often find simple and interpretable bounds on the Rademacher complexity $\mathcal{R}_n(\mathcal{W})$. This is the subject of the next sections.

5.3.4. Algorithms using basis approximation

Our analysis of linear models is easy to extend to a much more general setting. Suppose

$$Y_i = f(X_i) + \epsilon_i, \qquad \text{with} \qquad \mathbb{E}[\epsilon_i \mid X_i] = 0, \mathbb{V}\mathrm{ar}[\epsilon_i] = \sigma^2.$$

Let $\phi = (\phi_1, \ldots, \phi_K)$ be a vector of *basis functions* $\phi_k \colon \mathcal{Z} \to \mathbb{R}$. To approximate the function f, we use

$$h_\beta(x) = \sum_{k=1}^{K} \beta_j \phi_k(x) = \beta^\top \phi(x),$$

which is essentially a linear model. This formulation has many important special cases, including splines, wavelets, and SVMs. Often, one constructs n or more basis functions and relies on penalties/norm constraints for generalization.

To measure the risk of the algorithm, we switch the loss function. The scaling of the square loss generally makes the preceding bounds suboptimal. With refined arguments, one can typically achieve a generalization error scaling as $O(1/n)$ instead of $O(1/\sqrt{n})$. We'll get back to that later. The absolute error $L(y, h(x)) = |-h_0(x)|$ has the right scaling. Define the constrained least absolute deviation solution

$$\widehat{\beta} = \arg \min_{\|\beta\|_2 \le M_2} \frac{1}{n} \sum_{i=1}^{n} |Y - h_\beta(X_i)|,$$

which corresponds to median regression. Following similar arguments, we get:

Proposition 5.3.5. *Let $C^2 = \mathbb{E}[\|\phi(X)\|_2^2]$. Then with high probability,*

$$R(h_{\widehat{\beta}}) \le \sigma + \inf_{\|\beta\|_2 \le M_2} \mathbb{E}_X[|\beta^\top \phi(X) - f(X)|] + O\left(\frac{M_2 C}{\sqrt{n}}\right).$$

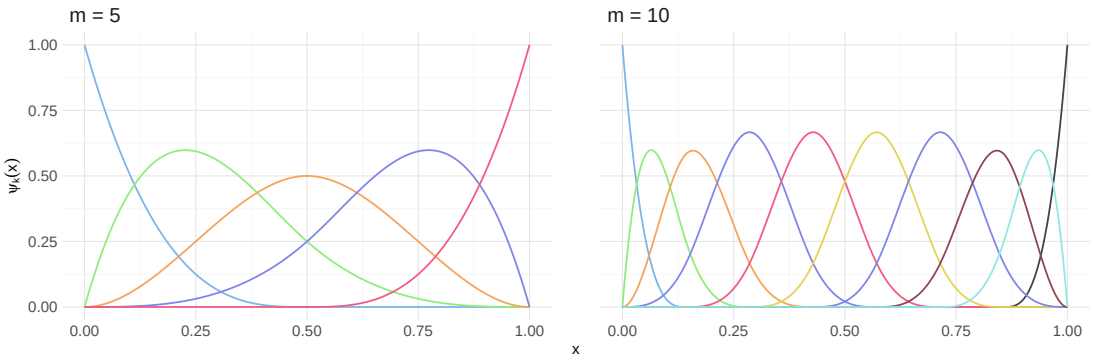

Figure 5.1.: Cubic spline bases with m basis functions.

Proof idea. Start again with (5.3).

$$R(\widehat{h}) \leq \inf_{h \in \mathcal{H}} R(h) + 2\mathcal{R}_n(\mathcal{L}) + B\sqrt{\frac{2\ln(1/\delta)}{n}}.$$

Using similar computations as in Theorem 5.3.1, we can show that the last two terms are of order $O(CM_2/\sqrt{n})$. For the first term,

$$\inf_{h \in \mathcal{H}} R(h) = R(f) + \inf_{h \in \mathcal{H}} [R(h) - R(f)].$$

It holds

$$R(f) = \mathbb{E}[|Y - f(X)|] \leq \sqrt{\mathbb{E}[|Y - f(X)|^2]} = \sqrt{\mathbb{E}[\epsilon^2]} = \sigma$$

and

$$|R(h_\beta) - R(f)| \leq \mathbb{E}[|h_\beta(X) - f(X)|] = \mathbb{E}[|\beta^\top \phi(X) - f(X)|]. \qquad \square$$

The risk is bounded by three terms. The first is the unavoidable noise variance. The second is the approximation error. It depends on both the expressivity of the basis ϕ and the norm constraint M_2. The third is the generalization error. It decreases with n, and increases in M_2 and C. The term $C^2 = \sum_{k=1}^{K} \mathbb{E}[\phi_k(Z)^2]$ measures the probablisitic size of the K-dimensional vector of basis functions $\phi(Z)$. How this term scales with K, depends on the basis system. For splines, the component norms $\mathbb{E}[\phi_k(Z)^2]$ decrease with their number K. For wavelet and Fourier bases, $\mathbb{E}[\phi_k(Z)^2]$ typically decreases with its index k.

Tensor product splines

Let's make this concrete for cubic product splines with m basis functions for each dimension. We start with a univariate B-spline basis $\psi_1, \ldots, \psi_m$. Such bases are illustrated in Fig. 5.1 for $m = 5, 10$. To approximate a univariate function

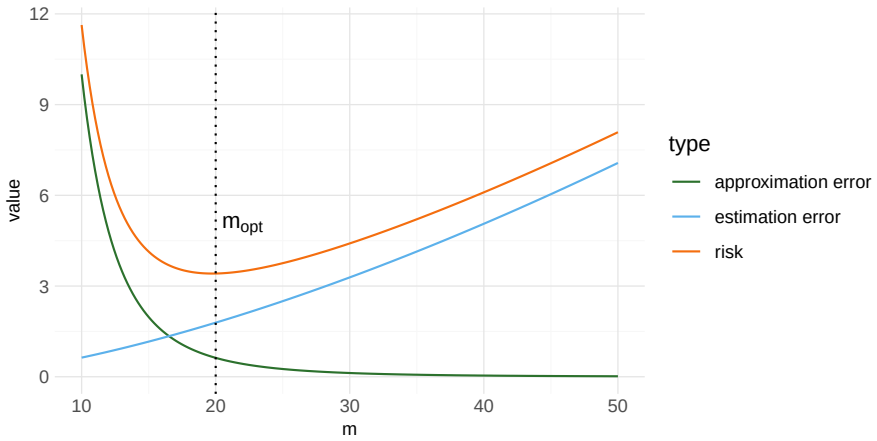

Figure 5.2.: The bias-variance tradeoff for tensor product splines with $d = 3$.

$f\colon \mathbb{R} \to \mathbb{R}$, we compute a weighted sum of the basis functions:

$$f(x) \approx \sum_{k=1}^{m} \beta_k \psi_k(x).$$

The exact formula for the basis functions ψ_k isn't too important, some visual intuition is enough. Each basis function concentrates on a subset of $[0, 1]$ of length at most $5/m$. The more basis functions we have, the more the individual functions concentrate. As m grows, this gives us increasingly fine local control of the approximation. One can show that as $m \to \infty$, we can approximate any four times continuously differentiable function g to arbitrary accuracy. More precisely,

$$\inf_{\beta} \sup_{x \in \mathcal{X}} |\beta^\top \psi(x) - f(x)| = O(m^{-4}).$$

We'll see later where this comes from.

To approximate multivariate functions $f\colon \mathbb{R}^d \to \mathbb{R}$, we can combine univariate bases:

$$f(x) \approx \sum_{k_1=1}^{m} \cdots \sum_{k_d=1}^{m} \beta_{k_1,\ldots,k_d} \phi_{k_1}(x_1) \cdots \phi_{k_d}(x_d).$$

This construction is called a *tensor product spline*. We can rewrite this as a linear model $\beta^\top \phi(x)$ by collecting all coefficients $\beta_{k_1,\ldots,k_d}$ in a long vector $\beta \in \mathbb{R}^{m^d}$ and defining

$$\phi(x) = \begin{pmatrix} \psi_1(x_1) \cdots \psi_1(x_d) \\ \psi_2(x_1)\psi_1(x_2) \cdots \psi_1(x_d) \\ \vdots \\ \psi_m(x_1) \cdots \psi_m(x_d). \end{pmatrix}$$

If M_2 is large enough and f four times continuously differentiable, one can

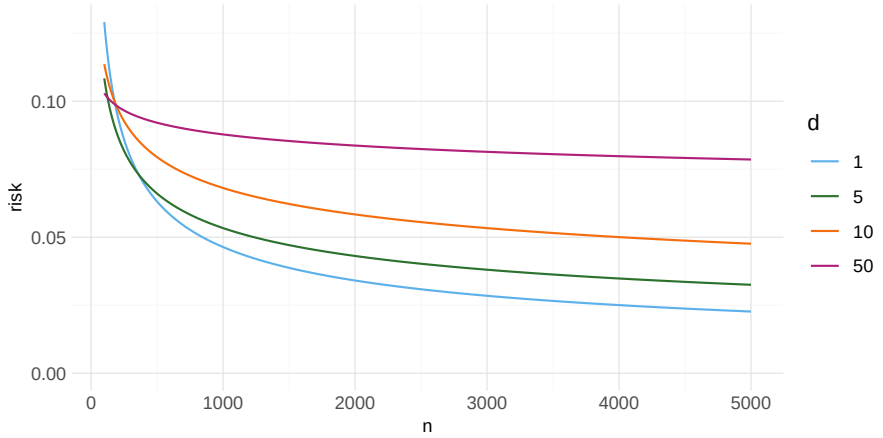

Figure 5.3.: The optimal convergence rate for different dimensions. Convergence is much slower when the dimension is large.

show

$$\inf_{\|\beta\|_2 \leq M_2} \sup_{x \in \mathcal{X}} |\beta^\top \phi(x) - f(x)| = O(m^{-4}),$$
$$M_2 = O(m^{d/2}),$$
$$C^2 = \sum_{k=1}^{K} O(m^{-d}) = O(1).$$

Together this gives

$$R(h_{\widehat{\beta}}) \leq \sigma + O(m^{-4}) + O(m^{d/2} n^{-1/2}).$$

The approximation error $O(m^{-4})$ decreases with m, the generalization error $O(m^{d/2} n^{-1/2})$ increases with m. The optimal choice of m balances the two terms. This is illustrated in Fig. 5.2 for $d = 3$. The approximation error decreases as m^{-4}, the estimation error increases as $m^{3/2}$, and the risk is the sum of the two terms plus σ. The perfect balance is attained at $m_{opt} = 20$ in this case.

More generally, optimizing the risk $O(m^{-4} + m^{d/2} n^{-1/2})$ for m gives $m_{opt} \propto n^{1/(8+d)}$, and

$$R(h_{\widehat{\beta}}) \leq \sigma + O(n^{-4/(8+d)}).$$

We determined the optimal value of m only up to unknown constants. This is sensible since constants appearing in bounds are rarely optimal. The analysis still has practical consequences. Since we know m scales as $An^{1/(8+d)}$, we can tune the hyperparameter A with a small subset of the training data, and then scale m up to the actual training set size.

The curse of dimensionality

The best possible convergence rate is another insight of our analysis. The convergence rate $O(n^{-4/(8+d)})$ is known to be optimal for estimating four times differentiable functions. No algorithm $\widehat{h}$ can achieve a faster rate on all of them. The convergence rate scales pretty badly with d. No matter the value of d, the rate is slower than $O(n^{-1/2})$. This is typical in *nonparametric* problems, where an infinite-dimensional object (the function f) is learned. While for $d = 1$ the rate $O(n^{-4/9})$ is pretty close to $O(n^{-1/2})$, already for $d = 10$ we get $O(n^{-2/9})$ which is much slower. In practice, that means we need waaaay more training data to learn the function f to a given accuracy. This phenomenon is known as *curse of dimensionality*. Fig. 5.3 illustrates the different convergence rates. While convergence is relatively fast for $d = 1$, there is hardly any convergence when $d = 50$. To put this into perspective: to achieve the same accuracy as for $d = 1$ and $n = 10^3$, we need around $n \approx 10^6$ when $d = 10$, and $n \approx 10^9$ when $d = 50$.

The curse can only be mitigated by imposing additional assumptions. For example, assuming a higher degree of smoothness and using higher-order splines improves the convergence rates, but not their scaling with d. To improve this scaling, one needs structural assumptions that constrain the effective dimension of the function f.

Additive splines

A common structural assumption to avoid the curse of dimensionality is additivity: We assume that f can be decomposed into a sum of d univariate functions,

$$f(x) = \sum_{j=1}^{J} f_j(x_j),$$

and approximate it with a basis vector $\phi\colon \mathbb{R} \to \mathbb{R}^m$ and

$$h_\beta(x) = \sum_{j=1}^{d} \beta_j^\top \phi(x_j), \quad \beta = (\beta_1, \dots, \beta_J) \in \mathbb{R}^{m \times d}.$$

Now there are only md (instead of m^d) basis coefficients to train. Now we have

$$\begin{aligned}
&\inf_{\|\beta\|_2 \le M_2} \sup_{x \in \mathcal{X}} |\beta^\top \phi(x) - f(x)| = O(dm^{-4}),\\
&M_2 = O(\sqrt{dm}),\\
&C^2 = \sum_{k=1}^{dm} O(m^{-1}) = O(d),
\end{aligned}$$

which gives

$$R(h_{\widehat{\beta}}) \le \sigma + O(m^{-4}) + O(d^{3/2} m^{1/2} n^{-1/2}).$$

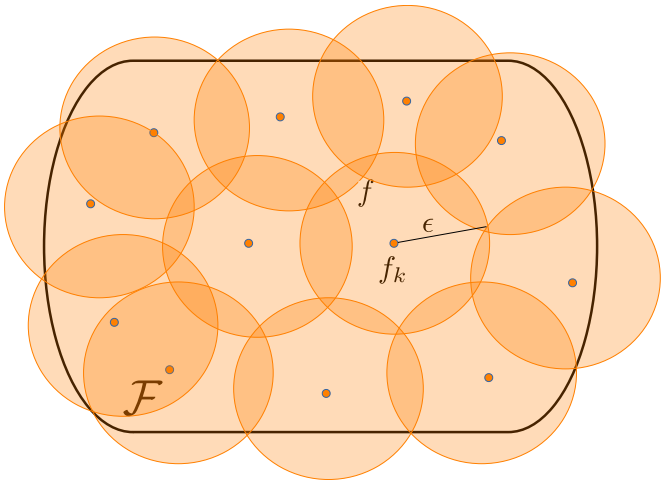

Figure 5.4.: A covering of the function class $\mathcal{F}$ with balls of size ϵ. The minimal number of balls required is the covering number.

The optimal choice is $m_{opt} \propto d^{-1/3} n^{-1/9}$, yielding

$$R(h_{\widehat{\beta}}) \leq \sigma + O(d^{4/3} n^{-4/9}).$$

Now the error scales much better with d. The dimension only shows up polynomially but does not affect how the error scales with n. For any finite d, the convergence rate is $O(n^{-4/9})$ equivalent to a one-dimensional problem.

5.4. Covering numbers and entropy

The Rademacher complexity measures the capacity of the hypothesis class and allows to bound the generalization error. In general, it depends on both $\mathcal{H}$ and the unknown measure P. There are other capacity measures that upper bound the Rademacher complexity and do not depend on P. This has two advantages. The generalization bounds become not only simpler but also stronger: they are valid uniformly over all possible P. If P is irrelevant, such complexity measures focus purely on the geometry of the hypothesis class $\mathcal{H}$. *Covering numbers* give a geometric measure for capacity through discretization.

5.4.1. Definition

A collection of balls $B(f_k, \epsilon) = \{f \in \mathcal{F} \colon \|f - f_k\| \leq \epsilon\}$ is called a cover of $\mathcal{F}$ if $\mathcal{F} \subset \bigcup_{k=1}^{N} B(f_k, \epsilon)$. An ϵ-cover is illustrated in Fig. 5.4. The minimal number of balls required to cover $\mathcal{F}$ is the covering number. The following definition is equivalent.

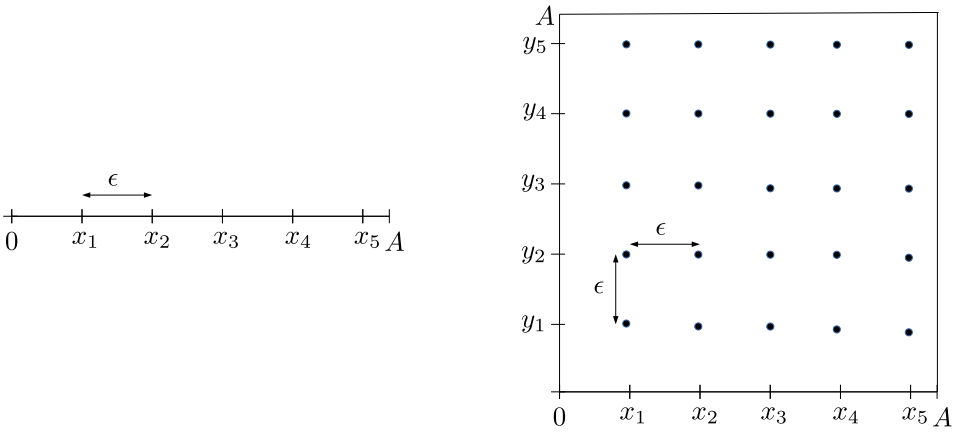

Figure 5.5.: Coverings of $[0, 1]$ (left) and $[0, 1]^2$ (right) in the ℓ_∞ norm.

Definition 5.4.1. *For any (semi)-norm $\|\cdot\|$ on $\mathcal{F}$ and $\epsilon > 0$, the **covering number** $N = N(\epsilon, \mathcal{F}, \|\cdot\|)$ is the minimal number of elements $f_1, \dots, f_N$ required such that for all $f \in \mathcal{F}$,*

$$\min_{1 \le k \le N} \|f - f_k\| \le \epsilon.$$

*The **covering entropy** is its logarithm, $\ln N(\epsilon, \mathcal{F}, \|\cdot\|)$.*

The covering number quantifies how easy it is to discretize $\mathcal{F}$ to a given accuracy, in a given norm $\|\cdot\|$. Instead of all functions in $\mathcal{F}$, we only consider a finite number of the centers $f_1, \dots, f_N$ of an ϵ-covering. Then for any function $f \in \mathcal{F}$, we can find a center f_k with distance at most ϵ. The larger the covering number N is for a given ϵ, the more capacity $\mathcal{F}$ has at that scale.

To build intuition, it helps to think of covering numbers of Euclidean sets $\mathcal{F} \subset \mathbb{R}^d$. As the simplest example, take $\mathcal{F} = [0, A]$ and $\|\cdot\| = |\cdot|$. We want to pick numbers $x_1, \dots, x_N$ such that for any x, there is k with $|x - x_k| \le \epsilon$. This is easily achieved by $x_1 = \epsilon, x_2 = 2\epsilon, \dots, x_N = 1$, see Fig. 5.5. In this construction, we need $N \le 1/\epsilon$ numbers, so $N(\epsilon, [0, A], |\cdot|) \le \lceil A/\epsilon \rceil$. The larger the interval is, the more points we need in the cover for any given ϵ.

We can extend this argument to $[0, A]^2 \subset \mathbb{R}^2$ by adding a sequence $y_k = k\epsilon$ and taking all combinations $(x_k, y_j), k, j = 1 \dots, \lceil A/\epsilon \rceil$. This gives us $N \le \lceil A/\epsilon \rceil^2 = O(1/\epsilon^2)$ and for any $(x, y) \in [0, A]^2$, there is (k, j) such that

$$\|(x, y) - (x_k, y_j)\|_\infty = \max\{|x - x_k|, |y - y_k|\} \le \epsilon.$$

The choice of norm makes a difference. For example,

$$\|(x, y) - (x_k, y_j)\|_1 = |x - x_k| + |y - y_k| \ge \|(x, y) - (x_k, y_j)\|_\infty.$$

So we need more balls to cover $[0, A]^2$ in the 1-norm. For Euclidean sets, this difference is negligible, however. In general, we have the following result.

Lemma 5.4.2. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$ and $S = \{x \in \mathbb{R}^d \colon \|x\| \leq r\}$. Then*

$$\left(\frac{r}{\epsilon}\right)^d \leq N\big(\epsilon, S, \|\cdot\|\big) \leq \left(\frac{3r}{\epsilon}\right)^d.$$

The covering number scales polynomially in $1/\varepsilon$. How fast it grows is determined by the intrinsic dimension d of the space.

We conclude that the covering number of a class depends on the norm, the diameter of a set in this norm, and its inherent dimension.

5.4.2. Covering bound on the Rademacher complexity

The Rademacher complexity involves a supremum over infinitely many $f \in \mathcal{F}$. If we are satisfied with an error of ϵ, we can replace the infinite supremum with a maximum over finitely many $f_1, \ldots, f_N$. Together with Massart's lemma, this gives an easy bound on the Rademacher complexity.

Remark 5.4.3. *In the following, we assume for simplicity that all $f \in \mathcal{F}$ are uniformly bounded by 1, i.e., $\sup_{f\in\mathcal{F}, z\in\mathcal{Z}} |f(z)| \leq 1$. If f was uniformly bounded by B instead, we could appeal to Proposition 5.2.9 (c) which shows*

$$\mathcal{R}_n(\mathcal{F}) = B\mathcal{R}_n(\mathcal{F}/B).$$

Any $f \in \mathcal{F}/B$ is now uniformly bounded by 1.

Proposition 5.4.4. *Suppose $\sup_{f\in\mathcal{F}, z\in\mathcal{Z}} |f(z)| \leq 1$. For all $\epsilon > 0$,*

$$\mathcal{R}_n(\mathcal{F}) \leq \epsilon + \sqrt{\frac{2\sup_Q \ln N\big(\epsilon, \mathcal{F}, L_2(Q)\big)}{n}},$$

where the supremum is taken over all probability measures.

Proof. Let's fix the data $Z_1, \ldots, Z_n$ for the moment. Define the $L_2(P_n)$ norm with respect to the empirical measure P_n by

$$\|f\|_{L_2(P_n)} = \sqrt{\int f(z)^2 dP_n(z)} = \sqrt{\frac{1}{n}\sum_{i=1}^n f(z_i)^2}.$$

Let $f_1, \ldots, f_N$ denote the centers of an ϵ-covering in this norm. Then

$$\begin{aligned}
&\mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(Z_i)\right] \\
&\leq \mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i [f(Z_i) - f_k(Z_i)] + \frac{1}{n}\sum_{i=1}^n \varepsilon_i f_k(Z_i)\right] \qquad \text{(take } f_k \text{ closest to } f\text{)}
\end{aligned}$$

$$\leq \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} |f(Z_i) - f_k(Z_i)| + \mathbb{E}\left[\max_{1\leq k\leq N} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f_k(Z_i)\right]. \quad \text{(triangle inequality)}$$

For any $f \in \mathcal{F}$, Jensen's inequality gives

$$\frac{1}{n}\sum_{i=1}^{n} |f(Z_i) - f_k(Z_i)| \leq \sqrt{\frac{1}{n}\sum_{i=1}^{n} |f(Z_i) - f_k(Z_i)|^2} = \|f - f_k\|_{L_2(P_n)} \leq \epsilon,$$

where we used that $f_1, \ldots, f_N$ are centers of an ϵ-covering. Further, Massart's lemma (Lemma 5.2.10) yields

$$\mathbb{E}_\varepsilon\left[\max_{1\leq k\leq N} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f_k(Z_i)\right] \leq \sqrt{\frac{2\ln N\big(\epsilon, \mathcal{F}, L_2(P_n)\big)}{n}} \leq \sqrt{\frac{2\sup_Q \ln N\big(\epsilon, \mathcal{F}, L_2(Q)\big)}{n}}.$$

The claim follows from combining the last three displays. □

Let's discuss the bound.

- Its first term increases with the discretization error ϵ.
- Its second decreases with the training set size n as $O(1/\sqrt{n})$, just as in the examples discussed in the previous section.
- It increases with the **uniform covering entropy** $\sup_Q \ln N\big(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(Q)}\big)$.

Clearly, $N(\epsilon, \mathcal{F}, \|\cdot\|)$ is a decreasing function of ϵ: the smaller the discretization error ϵ, the more balls we need to cover $\mathcal{F}$. This creates tension between the two terms in Proposition 5.4.4. Since ϵ is arbitrary, we can choose the one that minimizes the bound.

5.4.3. Euclidean function classes

Proposition 5.4.4 is most useful for *Euclidean classes*.

Definition 5.4.5 (Euclidean class). *A uniformly bounded function class $\mathcal{F}$ is called Euclidean if there are constants A, V such that for all $\epsilon \in (0, 1]$,*

$$\sup_Q N\big(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(Q)}\big) \leq A\epsilon^{-2V}.$$

The scaling of the covering number in ϵ is similar to the one of norm balls in $\mathbb{R}^d$ (Lemma 5.4.2), hence the name. Now, the number V plays the role of the intrinsic dimension of the space. The larger it is, the more complex is the class.

For $\epsilon \to 0$, the uniform covering *entropy* of a Euclidean class is

$$\sup_Q \ln N\big(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(Q)}\big) \approx V \ln(1/\epsilon). \tag{5.4}$$

This term grows very slowly as $\epsilon \to 0$. For example, taking $\epsilon = 1/n^2$ in Proposition 5.4.4 gives

$$\mathcal{R}_n(\mathcal{F}) \lesssim \frac{1}{n^2} + \sqrt{\frac{V \ln n}{n}}.$$

The first term is negligible compared to the second, which decays as $O(\sqrt{\ln n/n})$ — only slightly slower than $O(1/\sqrt{n})$. The $\ln n$ factor can be removed with a finer bound developed later.

Euclidean classes are essentially finite dimensional. We can learn them as well as Euclidean parameters $\theta \in \mathbb{R}^p$. Unsurprisingly, important examples of Euclidean function classes are those that are parametrized by a Euclidean vector.

Example 5.4.6. *Suppose $\mathcal{F}$ is contained in a bounded subset of p-dimensional vector space of functions. That is, there is $M < \infty$ such that we can write $f \in \mathcal{F}$ as $f(z) = \sum_{j=1}^p \theta_j \phi_j(z)$ for some basis functions $\phi_1, \dots, \phi_p$ and coefficient $\theta \in \mathbb{R}^p$ with $\|\theta\| \le M$. Then $\mathcal{F}$ is Euclidean with dimension $V = p + 1$.*

Example 5.4.7. *Let Θ be a bounded subset of $\mathbb{R}^p$. Define $\mathcal{F} = \{f_\theta \colon \theta \in \Theta\}$ as a collection of functions that are Lipschitz in θ, i.e.,*

$$|f_\theta(z) - f_{\theta'}(z)| \le \Phi(z)\|\theta - \theta'\|,$$

with $\|\Phi\|_{L_2(P)} < \infty$. Then $\mathcal{F}$ is Euclidean with dimension $V = p$.

Example 5.4.8. *Let $\psi \colon \mathbb{R} \to \mathbb{R}$ be any function of bounded variation. Then*

$$\{\psi(\|Az - b\|) \colon A \in \mathbb{R}^{m \times k}, b \in \mathbb{R}^m\}$$

is Euclidean with $V = m(k + 1)$ and

$$\{\psi(\beta^\top z + a) \colon \beta \in \mathbb{R}^p, a \in \mathbb{R}\}$$

is Euclidean with $V = p + 1$.

Remark 5.4.9. *We didn't address the constant A from Definition 5.4.5. Its magnitude is often an artifact of the proof and nobody really cares to make them small. We're already far down a chain of several bounds. Their purpose is not numerical accuracy, but to help us understand which forces act on generalization.*

The linear models

$$\mathcal{H}_{2,M} = \{z \mapsto \beta^\top z \colon \beta \in \mathbb{R}^d, \|\beta\|_2 \le M\}$$

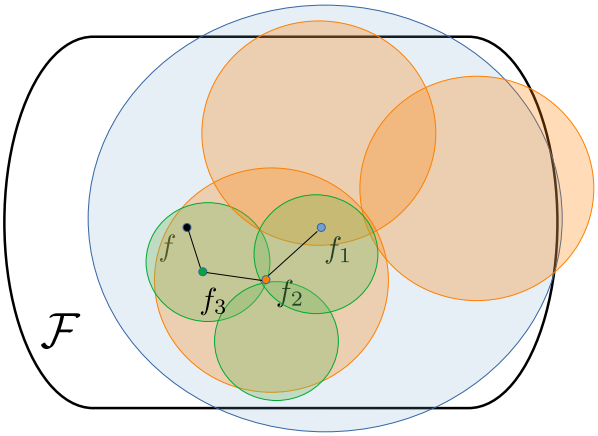

Figure 5.6.: A chain of successively finer coverings.

are a special case of all examples above with envelope $F(z) = M\|z\|_2$. Letting $B = \sup_{z\in\mathcal{Z}, f\in\mathcal{F}} |f(z)|$ and $\sup_{z\in\mathcal{Z}} \|z\|_2 \leq 1$, we have $B \leq M$. Recalling Remark 5.4.3, we get the Rademacher bound

$$\mathcal{R}_n(\mathcal{H}_{2,M}) \lesssim M\sqrt{\frac{d\ln n}{n}}.$$

This is much worse than our direct bound from Theorem 5.3.1:

$$\mathcal{R}_n(\mathcal{H}_{2,M}) \lesssim M\sqrt{\frac{1}{n}}.$$

The extra $\sqrt{d}$ factor is clearly unnecessary, so the covering bounds are rather crude. A similar factor for the number of basis coefficients would also appear for the basis expansions in Section 5.3.4, which is a special case of Example 5.4.6. This highlights once again that it's important to be careful when interpreting bounds. Depending on the focus of our analysis, we may or may not get away with cruder bounds.

Neural networks are also a special case of Example 5.4.7. The corresponding Rademacher bound is $\mathcal{R}(\mathcal{H}) \lesssim \sqrt{p/n}$, where p is the number of parameters of the network. Since in modern applications, $p \gg n$, such simple bounds cannot explain generalization in the overparametrized regime. We'll need more attention to detail.

5.4.4. Chaining and the entropy integral

The Rademacher bound in Proposition 5.4.4 is far from optimal. It is sufficient for Euclidean classes, but useless for more complex classes. Classes of general smooth functions are an example.

Example 5.4.10. *Let C_M^k be the set of all k-times continuously differentiable functions $\mathcal{Z} \to \mathbb{R}$ with all derivatives bounded uniformly by M and $K = \sup_{z\in\mathcal{Z}} \|z\|_\infty$. Then*

$$\sup_Q \ln N(\epsilon, C_M^k, L_2(Q)) \lesssim MK\left(\frac{1}{\epsilon}\right)^{d/k}.$$

This is a bound on the covering entropy, not the covering number. Compared to the bound for Euclidean classes, we have a logarithm on the left-hand side. The covering *number* bound would scale as $O(e^{\epsilon^{-d/k}})$, which is exponentially worse than for Euclidean classes.

Even for such large classes, we can derive a meaningful Rademacher bound. It arises from a clever technique called *chaining*. The idea is to construct a sequence of coverings at successively finer scales. Then follow the path from f (the finest scale with $\epsilon = 0$) to an approximation on coarse scale. This is illustrated in Fig. 5.6. In each step along this chain, we can bound the complexity separately, and aggregate the results.

Theorem 5.4.11 (Dudley's Theorem)**.** *Suppose $\sup_{f\in\mathcal{F},z\in\mathcal{Z}} |f(z)| \leq 1$. It holds:*

$$\mathcal{R}_n(\mathcal{F}) \leq \frac{12}{\sqrt{n}} \sup_Q \int_0^1 \sqrt{\ln N\Big(\epsilon, \mathcal{F}, L_2(Q)\Big)}\, d\epsilon.$$

More precisely: for any $\alpha \geq 0$,

$$\mathcal{R}_n(\mathcal{F}) \leq 4\alpha + \frac{12}{\sqrt{n}} \sup_Q \int_\alpha^1 \sqrt{\ln N\Big(\epsilon, \mathcal{F}, L_2(Q)\Big)}\, d\epsilon.$$

Proof. Let $\delta_j = (1/2)^j$ and $\mathcal{F}^{(j)} = \{f_1^{(j)}, \ldots, f_{N_j}^{(j)}\}$ be the centers of δ_j-coverings in the $L_2(P_n)$-norm. Note that because $\sup_{f\in\mathcal{F},z\in\mathcal{Z}} |f(z)| \leq 1$, $\mathcal{F}$ is contained in a single $L_2(P_n)$-ball of size $\delta_0 = 1$. Denote the center closest to f by $f^{(j)} \in \mathcal{F}^{(j)}$. By adding and subtracting term, expand

$$f = f - f^{(J)} + f^{(J)} - f^{(J-1)} + f^{(J-1)} + \cdots - f^{(0)} + f^{(0)}.$$

Taking $J \to \infty$, we have $f - f^{(J)} \to 0$ and thus

$$f = f^{(0)} + \sum_{j=1}^{\infty} (f^{(j)} - f^{(j-1)}).$$

Substitute this expression in the empirical Rademacher complexity:

$$\mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i f(Z_i)\right]$$

$$= \mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f^{(0)} + \sum_{j=1}^\infty \frac{1}{n}\sum_{i=1}^n \varepsilon_i (f^{(j)}(Z_i) - f^{(j-1)}(Z_i))\right]$$
$$\leq \mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f^{(0)}(Z_i)\right] + \sum_{j=1}^\infty \mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i (f^{(j)}(Z_i) - f^{(j-1)}(Z_i))\right].$$

[The term in the third line is larger, because there we can take a worst-case f for every j, while in the second we have to settle on one f for all j.] Because $\mathcal{F}^{(0)} = \{f^{(0)}\}$ consists of a single function, we have

$$E_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f^{(0)}(Z_i)\right] = E_\varepsilon\left[\frac{1}{n}\sum_{i=1}^n \varepsilon_i f^{(0)}(Z_i)\right] = 0.$$

To bound the other terms, we want to apply Massart's lemma. We first find a bound on the radius. By the triangle inequality,

$$\|f^{(j)} - f^{(j-1)}\|_{L_2(P_n)} \leq \|f^{(j)} - f\| + \|f - f^{(j-1)}\|_{L_2(P_n)} \leq \delta_j + \delta_{j-1} = 3\delta_j.$$

Furthermore, there are at most $N_j N_{j-1}$ functions in the set

$$\mathcal{F}^{(j)} - \mathcal{F}^{(j-1)} = \left\{f^{(j)} - f^{(j-1)} \colon f^{(j)} \in \mathcal{F}^{(j)}, f^{(j-1)} \in \mathcal{F}^{(j-1)}\right\}.$$

Denote for simplicity $N(\delta) = N\big(\delta, \mathcal{F}, \|\cdot\|_{L_2(P_n)}\big)$. Now,

$$\begin{aligned}
&\sum_{j=1}^\infty \mathbb{E}_\varepsilon\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i (f^{(j)} - f^{(j-1)})\right] \\
&\leq 3\sum_{j=1}^\infty \delta_j \sqrt{\frac{2\ln[N(\delta_j)N(\delta_{j-1})]}{n}} && \text{(Massart's lemma)} \\
&\leq \frac{6}{\sqrt{n}} \sum_{j=1}^\infty \delta_j \sqrt{\ln N(\delta_j)} && (N(\delta_j) \geq N(\delta_{j-1})) \\
&= \frac{12}{\sqrt{n}} \sum_{j=1}^\infty (\delta_j - \delta_{j+1}) \sqrt{\ln N(\delta_j)} && (\delta_j = 2(\delta_j - \delta_{j+1})) \\
&\leq \frac{12}{\sqrt{n}} \int_0^{\delta_0} \sqrt{\ln N(\delta)}\, d\delta. && \text{(Riemann series)}
\end{aligned}$$

The last inequality holds because $\sqrt{\ln N(\delta)}$ is a decreasing function in δ, so $N(\delta_j) \leq N(\delta)$ for all $\delta \in [\delta_{j+1}, \delta_j]$. The term $(\delta_j - \delta_{j+1})\sqrt{\ln N(\delta_j)}$ is then the area of a box *underneath* the graph of $\sqrt{N(\delta)}$. Now take supremum over probability measures. The second bound involving α can be derived by stopping the chain early. □

5.4.5. Applications

Dudley's theorem allows us to get rid of the unnecessary $\ln n$ factor in generalization bounds for Euclidean classes (5.4).

Example 5.4.12 (Euclidean classes)**.** *If $\mathcal{F}$ is Euclidean, so $N(\epsilon) \lesssim \epsilon^{-2V}$, we have*

$$\sup_Q \int_0^1 \sqrt{\ln N(\epsilon, \mathcal{F}, L_2(Q))} d\epsilon \lesssim \sqrt{2V} \int_0^1 \sqrt{\log(1/\epsilon)} d\epsilon \lesssim \sqrt{V}.$$

Dudley's theorem implies

$$\mathcal{R}_n(\mathcal{F}) \lesssim \sqrt{\frac{V}{n}},$$

The unnecessary parameter dimension in the generalization bounds for linear models or neural networks does not immediately disappear, however. The theorem's real power lies elsewhere. Dudley's theorem let's us treat classes $\mathcal{F}$ whose covering number scales much worse than ϵ^{-V}.

For classes with $\ln N(\epsilon) \leq A/\epsilon^2$ (notice the log!),[a]

$$\mathcal{R}_n(\mathcal{F}) \lesssim \sqrt{\frac{A \ln n}{n}}.$$

[a] Apply Theorem 5.4.11 with $\alpha = n^{-2}$.

Compare this to the bound implied by Proposition 5.4.4:

$$\mathcal{R}_n(\mathcal{F}) \lesssim \epsilon + \sqrt{\frac{A/\epsilon^2}{n}}.$$

The optimal choice is $\epsilon \propto n^{-1/4}$, for which $\mathcal{R}_n(\mathcal{F}) \lesssim A^{1/4} n^{-1/4}$ which vanishes much slower. So especially for large function classes, Dudley's theorem gives us much sharper control over the generalization gap.

Example 5.4.13 (Smooth functions)**.** *The class of k-smooth functions from satisfies $\ln N(\epsilon) \leq A/\epsilon^2$ if $k \geq d/2$ and, thus,*

$$\mathcal{R}_n(\mathcal{F}) \lesssim \sqrt{\frac{A \ln n}{n}}.$$

Warning: In the following, we hide logarithmic factors in the $\lesssim$-sign and O-symbol.

If $k < d/2$, we need more care to find the best α and will get rates slower than $O(1/\sqrt{n})$. This is left as an exercise. The set of all k-smooth functions is incredibly rich. It's a truly nonparametric, infinite-dimensional class of functions. This also provides a new viewpoint on parametric classes.

Theorem 5.4.14. *Let $q \geq 1$, $\sup_{x\in\mathcal{X}} \|x\|_{(q-1)/q} \leq B$, and*

$$\mathcal{F}_{q,M} = \{x \mapsto \beta^\top x \colon \beta \in \mathbb{R}^d, \|\beta\|_q \leq M\}.$$

Then

$$\mathcal{R}(\mathcal{F}_{q,M}) \lesssim \frac{M(B+1)}{\sqrt{n}}.$$

Proof. For any function $f(x) = \beta^\top x$, Hölder's inequality gives

$$|f(x)| \leq \|\beta\|_q \|x\|_{(q-1)/q} \leq MB.$$

Further

$$\|\nabla f(x)\|_\infty = \|\beta\|_\infty \leq \|\beta\|_q \leq M,$$

and $\|\nabla^k f(x)\|_\infty = 0$ for all $k \geq 2$. Because $\max(MB, M) \leq MB + M$, we have shown that $\mathcal{F}_q \subset C^k_{M(B+1)}$ for any $k \geq 0$. □

This new bound is similar to the Rademacher bounds from Theorem 5.3.1. It does not have the redundant $\sqrt{d}$-factor that appeared when we treated $\mathcal{F}_{q,M}$ as a Euclidean class in Section 5.4.3.

A similar analysis also applies to neural networks.

Theorem 5.4.15. *For*

$$\mathcal{F} = \left\{ f(x) \mapsto W_1\sigma(W_2\sigma(\cdots\sigma(W_L x))) \colon \sup_{x\in\mathcal{X}} \max_{k\geq 1} \|\nabla^k f(x)\|_\infty \leq M \right\},$$

it holds

$$\mathcal{R}(\mathcal{F}) \lesssim \frac{M}{\sqrt{n}}.$$

The bound does not depend on the number of neurons or layers, only on the smoothness of the networks in the class. So if we can ensure that our optimizer returns $\widehat{h} \in \mathcal{F}$ with high probability, we can get relatively sharp generalization bounds. That's a big 'if', but we gained a useful conceptual insight. We'll get back to this later when we discuss overparametrization in more detail.

5.5. Vapnik-Chervonenkis dimension

5.5.1. Some context

Statistical learning theory was long dominated by another complexity measure. It has a surprisingly long history and starts way before modern ML was even thinkable. In the 1960s and 1970s *Vladimir Vapnik*, along with *Alexey Chervonenkis*, developed a formal theory of empirical risk minimization. This work culminated in two monographs (Vapnik, 1999, 1998) summarizing and synthesizing the main results and findings. One major achievement was the invention of Support Vector Machines (SVMs), heavily inspired by the theoretical insights on the driving forces behind generalization.

The foundation of this development was a complexity measure called *Vapnik-Chervonenkis dimension* or just *VC dimension*.[2] Most of the early developments focused on binary classification problems, and the VC dimension is tailored to those. An extension for regression problems, the *fat-shattering dimension*, never really caught on. Because it often leads to suboptimal bounds, the VC dimension is in a way superseded by the other measures we discussed. But the VC dimension is still useful and widely known. Not least because of its historical importance, a course on statistical learning theory simply can't do without it. But we'll keep our account much shorter than what's been traditionally found in such courses.

5.5.2. Derivation

Consider a binary classification problem with the 0-1-loss $L(y, h(x)) = \mathbb{1}\{y \neq h(x)\}$. We'll take a modern viewpoint and start with the empirical Rademacher complexity of the loss class $\mathcal{L} = L \circ \mathcal{H}$:

$$\widehat{\mathcal{R}}_n(\mathcal{L}) = \mathbb{E}_\varepsilon\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1} \varepsilon_i \mathbb{1}\{Y_i \neq h(X_i)\}\right].$$

The expectation is over ε only, so (Y_i, X_i) are considered fixed for now. Defining

$$A = \left\{a \in \{0,1\}^n \colon a_i = \mathbb{1}\{Y_i \neq h(X_i)\}, h \in \mathcal{H}\right\},$$

we can rewrite this as

$$\widehat{\mathcal{R}}_n(\mathcal{L}) = \mathbb{E}_\varepsilon\left[\sup_{a\in A} \frac{1}{n}\sum_{i=1} \varepsilon_i a_i\right].$$

[2] Their seminal paper, Vapnik and Chervonenkis (1971), is an English translation of a paper published a few years earlier in Russian.

If A was finite, we could now apply Massart's lemma to find an upper bound:

$$\mathbb{E}\left[\max_{a\in A}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i a_i\right] \leq \sqrt{\frac{2\ln|A|}{n}}.$$

But A is finite! There are at most 2^n possibilities for the vector $a \in \{0,1\}^n$. Plugging this into the bound gives $\widehat{\mathcal{R}}_n(\mathcal{L}) \leq \sqrt{2\ln 2}$. Unfortunately, the bound is useless because it doesn't decrease in n. The size of A is additionally limited by the capacity of $\mathcal{H}$. For most ML algorithms, the set A is much smaller than 2^n, so we may get meaningful bounds. An additional complication is that the size of A depends on the realizations of $(Y_i, X_i)_{i=1}^n$. This motivates the following definition.

Definition 5.5.1. *The* ***growth function*** $\Gamma_n(\mathcal{H})$ *is defined as*

$$\Gamma_n(\mathcal{H}) = \sup_{x_1,\dots,x_n\in\mathcal{X}}\left|\{\big(h(x_1),\dots,h(x_n)\big)\colon h\in\mathcal{H}\}\right|.$$

Now Massart's lemma immediately gives the bound

$$\mathcal{R}_n(\mathcal{L}) \leq \sqrt{\frac{2\ln\Gamma_n(\mathcal{H})}{n}}.$$

The growth function does not depend on the probability measure P. It is a purely combinatorial measure of $\mathcal{H}$'s capacity. It counts how many possible configurations a classifier $h \in \mathcal{H}$ can take on an arbitrary data set of size n. If there is a data set, where any prediction pattern is possible, we have $\Gamma_n(\mathcal{H}) = 2^n$. We say that $\mathcal{H}$ *shatters* a set of n points. To make the bound above useful, we need $\Gamma_n(\mathcal{H}) < 2^n$. There is more hope for this when n is large because h needs to generate more patterns. The largest n where all patterns can be generated is the VC dimension.

Definition 5.5.2.

1. *The* ***VC dimension*** *is defined as*

$$\mathrm{VC}(\mathcal{H}) = \sup\{n\in\mathbb{N}\colon \Gamma_n(\mathcal{H}) = 2^n\}.$$

2. $\mathcal{H}$ *is called a* ***VC class*** *if* $\mathrm{VC}(\mathcal{H}) < \infty$.

5.5.3. Examples and Implications

Example 5.5.3. *Suppose* $\mathcal{X} = \mathbb{R}^2$ *and define the class of linear classification rules*

$$\mathcal{H} = \Big\{h(x) = \mathrm{sign}(\beta_0 + \beta_1 x_1 + \beta_2 x_2)\colon \beta_1,\beta_2\in\mathbb{R}\Big\}.$$

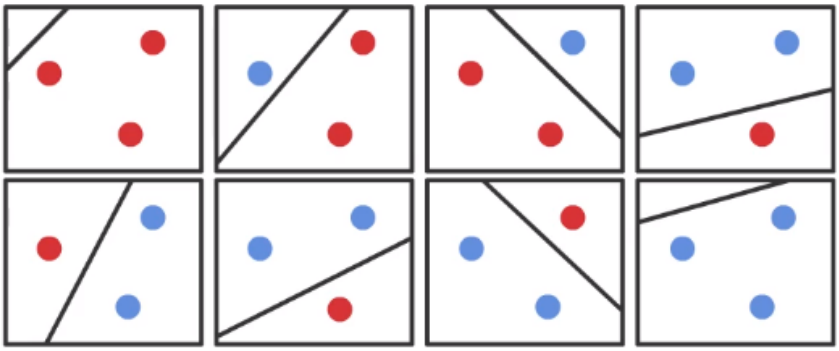

Figure 5.7.: Three points in $\mathbb{R}^2$ can be shattered by linear hypotheses.

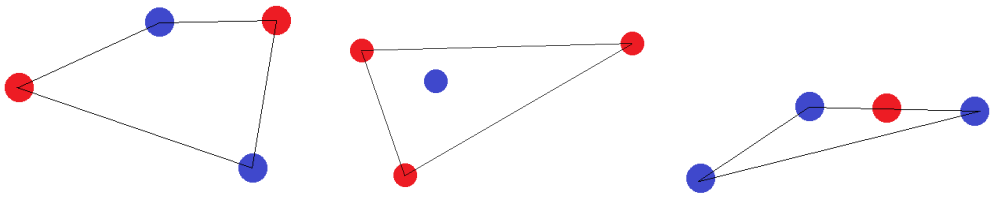

Figure 5.8.: No four points in $\mathbb{R}^2$ can be shattered by linear hypotheses.

Fig. 5.7 shows an arrangement of three points that is shattered by $\mathcal{H}$. The hypothesis class $\mathcal{H}$ can generate all possible classification patterns on these points and, thus, $\Gamma_n(\mathcal{H}) = 2^n$. If we add another point, the points can no longer be shattered (Fig. 5.8). Assuming that the points form a trapezoid, it is not possible to label opposite corners differently. If they don't form a trapezoid, then either three points lie on a line segment or one point is in the interior of a triangle. In both cases, it is impossible to generate all combinations. We have shown that $\mathrm{VC}(\mathcal{H}) = 3$.

Example 5.5.4. *In general, the VC-dimension is unrelated to the number of parameters in a class. A famous example is the class*

$$\mathcal{H} = \Big\{h\colon \mathbb{R} \to \mathbb{R}, h(x) = \operatorname{sign}\Big(\sin(ax)\Big) : a \in \mathbb{R}\Big\}.$$

It has only one parameter, but $\mathrm{VC}(\mathcal{H}) = \infty$*: on n given points, any pattern can be generated by choosing the frequency a large enough.*

When n hits the VC-dimension, a surprising phase shift occurs. The growth function becomes polynomial.

Lemma 5.5.5 (Sauer's lemma)**.** *It always holds*

$$\Gamma_n(\mathcal{H}) \quad \begin{cases} = 2^n, & n \leq \mathrm{VC}(\mathcal{H}), \\ \leq \left(\dfrac{3n}{\mathrm{VC}(\mathcal{H})}\right)^{\mathrm{VC}(\mathcal{H})}, & n > \mathrm{VC}(\mathcal{H}). \end{cases}$$

This implies

$$\mathcal{R}_n(\mathcal{L}) \leq \sqrt{\frac{2\,\mathrm{VC}(\mathcal{H})\ln(3n/\,\mathrm{VC}(\mathcal{H}))}{n}}.$$

The $\ln n$ term can again be removed. The easiest way to do this is to invoke Dudley's entropy integral. This is possible because the VC-dimension allows to upper bound the covering number. The full proof of the following result is quite tedious, see Van der Vaart and Wellner (1996, Theorem 2.6.7).

Theorem 5.5.6. *There exist universal constants $C, K < \infty$ such that for all $\epsilon \in [0, 1)$,*

$$\sup_Q N\Big(\epsilon, \mathcal{H}, \|\cdot\|_{L_2(Q)}\Big) \leq CK^{\mathrm{VC}(\mathcal{H})}\,\mathrm{VC}(\mathcal{H})\left(\frac{1}{\epsilon}\right)^{2\,\mathrm{VC}(\mathcal{H})}.$$

Proof (optional). We can prove a slightly looser bound with less effort. The idea is actually quite nice.

- First we relate the $L_2(Q)$-norm to a probability:

$$\|\ell_1 - \ell_2\|^2_{L_2(Q)} = \mathbb{E}[|\ell_1(Z) - \ell_2(Z)|^2] = \mathbb{E}[|\ell_1(Z) - \ell_2(Z)|] = \mathbb{P}_Q\{h_1(X) \neq h_2(X)\}.$$

- Then we construct an ϵ-*packing* of $\mathcal{H}$, a collection ϵ-balls that have centers inside $\mathcal{H}$ but do not touch. That is, there are $h_1, \ldots, h_M$ such that

$$\|h_i - h_j\|_{L_2(Q)} > \epsilon.$$

  The maximal number $M = M(\epsilon, \mathcal{H}, \|\cdot\|_{L_2(Q)})$ we can pack into $\mathcal{H}$ is related to the covering number via

$$N(2\epsilon, \mathcal{H}, \|\cdot\|_{L_2(Q)}) \leq M(\epsilon, \mathcal{H}, \|\cdot\|_{L_2(Q)}) \leq N(\epsilon, \mathcal{H}, \|\cdot\|_{L_2(Q)}).$$

- Now we relate the packing number to the growth function. We will see that for $n = \ln M^2/\epsilon^2$, there is a configuration $x_1, \ldots, x_n$ such that $|\{(h(x_1), \ldots, h(x_n))\colon h \in \mathcal{H}\}| \geq M$. It thus holds

$$N(2\epsilon) \leq M(\epsilon) \leq |\{(h(x_1), \ldots, h(x_n))\colon h \in \mathcal{H}\}| \leq \Gamma_n(\mathcal{H}).$$

  Using Sauer's lemma Lemma 5.5.5, this gives

$$N(2\epsilon) \leq \left(\frac{3n}{\mathrm{VC}(\mathcal{H})}\right)^{\mathrm{VC}(\mathcal{H})} \leq \left(\frac{3\ln N(2\epsilon)^2}{\mathrm{VC}(\mathcal{H})}\right)^{\mathrm{VC}(\mathcal{H})} \left(\frac{1}{\epsilon}\right)^{2\,\mathrm{VC}(\mathcal{H})}.$$

  The bound isn't perfect, because $N(2\epsilon)$ shows also up on the right. Because it only comes in algorithmically, this factor is almost negligible compared to the ϵ^{-2}-term. This is as good as it gets with our "simple' method.

- To show that such $x_1, \dots, x_n$ exists, we use the *probabilistic method*: We show that the existence of such $x_1, \dots, x_n$ has non-zero probability, so it must exist. For $X_1, \dots, X_n \overset{iid}{\sim} Q$, we have

$$\begin{aligned}
\mathbb{P}\Big(|\{(h(X_1), \dots, h(X_n))\colon h \in \mathcal{H}\}| \geq M\Big) &\geq \mathbb{P}\Big(\forall i, j\colon (h_i(X_1), \dots, h_i(X_n)) \neq (h_j(X_1), \dots, h_j(X_n))\Big) \\
&= 1 - \mathbb{P}\Big(\exists i, j\colon (h_i(X_1), \dots, h_i(X_n)) = (h_j(X_1), \dots, h_j(X_n))\Big) \\
&\geq 1 - \sum_{i<j} \mathbb{P}\Big((h_i(X_1), \dots, h_i(X_n)) = (h_j(X_1), \dots, h_j(X_n))\Big) \\
&= 1 - \sum_{i<j} \mathbb{P}\big(h_i(X_1) = h_j(X_1)\big)^n
\end{aligned}$$

  Because

$$\mathbb{P}\big(h_i(X_1) \neq h_j(X_1)\big) = \|\ell_1 - \ell_2\|^2_{L_2(Q)} > \epsilon^2,$$

  we have

$$\mathbb{P}\big(h_i(X_1) = h_j(X_1)\big)^n < (1-\epsilon^2)^n \geq \exp(-\epsilon^2 n) \overset{n=\ln M^2/\epsilon^2}{=} \frac{1}{M^2}.$$

  So overall,

$$\mathbb{P}\Big(|\{(h(X_1), \dots, h(X_n))\colon h \in \mathcal{H}\}| \geq M\Big) \geq 1 - \binom{N}{2}\frac{1}{M^2} \geq \frac{1}{2}. \qquad \square$$

This bound is similar to the definition of Euclidean classes (Definition 5.4.5). In fact, any VC class is also Euclidean. The Euclidean formulation is more general though because it also applies to non-binary functions.

There are bounds on the VC dimension on virtually any popular ML algorithm. Several of the bounds on covering numbers were initially derived that way. Because we have better tools, we won't get into that. We'll end this digression with a funny result due to Goldberg and Jerrum (1993).

Proposition 5.5.7. *Let $h_\theta\colon \mathbb{R}^d \to \{-1, 1\}$ denote an algorithm that computes its result with at most K of the following operations: (i) basic real arithmetic $(+, -, \cdot, /)$, (ii) if-branches based on (in)equality comparison. Then the class $\mathcal{H} = \{h_\theta\colon \theta \in \mathbb{R}^p\}$ has*

$$\mathrm{VC}(\mathcal{H}) \leq 2p(2K + 88).$$

6. Further topics

6.1. Fast rates

When discussing splines in Section 5.3.4, I mentioned that the square loss has suboptimal scaling. Risk bounds can be improved from $O(1/\sqrt{n})$ to $O(1/n)$. This can also happen for classification. The rate $O(1/\sqrt{n})$ is sometimes called the *slow rate* and everything faster than that a *fast rate*. So how do we improve the rate?

6.1.1. Intuition

Let's outline the central idea. Recall the basic bound from Proposition 3.2.1 and let $h^* = \arg\min_{h\in\mathcal{H}} R(h)$ be the best hypothesis in the class. For an algorithm $\widehat{h}$ with $R_n(\widehat{h}) - R_n(h^*) \leq 0$, we have

$$R(\widehat{h}) - R(h^*) \leq \sup_{h\in\mathcal{H}}\Big[R(h) - R_n(h)\Big] - \Big[R(h^*) - R_n(h^*)\Big].$$

Assuming that the loss function L is Lipschitz (as in Lemma 5.2.5), this leads to the bound

$$R(\widehat{h}) - R(h^*) \lesssim \mathcal{R}_n(\mathcal{H}) \quad \text{w.h.p.}, \tag{6.1}$$

which is normally of order $O(1/\sqrt{n})$. The bound is quite pessimistic because we take the supremum over all of $\mathcal{H}$. What if $\widehat{h}$ concentrates on a rather small subset $\mathcal{H}'$ of $\mathcal{H}$? Then we could restrict the supremum to this subset and potentially get the tighter bound

$$R(\widehat{h}) - R(h^*) \lesssim \mathcal{R}_n(\mathcal{H}') \quad \text{w.h.p.} \tag{6.2}$$

We can take this to the extreme. If $\widehat{h}$ would converge to h^*, then the set $\mathcal{H}'$ would become smaller and smaller as n grows. For fixed $\mathcal{H}$, the Rademacher complexity normally decays like $\mathcal{R}_n(\mathcal{H}) = O(1/\sqrt{n})$. By factoring in the decrease in the size of $\mathcal{H}'$, we may be able to prove fast rates.

So how do we make sure that $\widehat{h}$ concentrates? Our bound (6.1) already restricts where $\widehat{h}$ lives ins some sense: $\widehat{h}$ lies with high probability in the set

$$\mathcal{H}' = \{h \in \mathcal{H} : R(h) - R(h^*) \lesssim \mathcal{R}_n(\mathcal{H})\}.$$

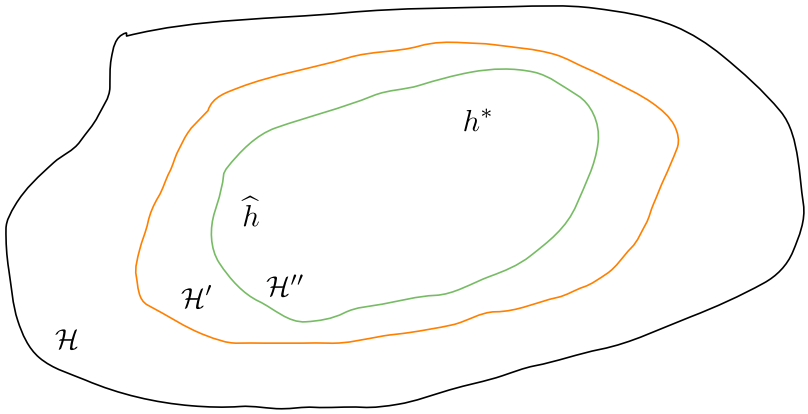

Figure 6.1.: Fast rates may arise from iteratively constraining the set where $\widehat{h}$ lies (with high probability).

This is illustrated in Fig. 6.1. Now we can iterate the same argument and show

$$R(\widehat{h}) - R(h^*) \lesssim \mathcal{R}_n(\mathcal{H}'') \quad \text{w.h.p..}$$

for

$$\mathcal{H}'' = \{h \in \mathcal{H} : R(h) - R(h^*) \lesssim \mathcal{R}_n(\mathcal{H}')\},$$

and so forth. We get fast rates, if the complexity of

$$\mathcal{H}_r = \{h \in \mathcal{H} : R(h) - R(h^*) \leq r\}$$

decreases quick enough with its "radius" r.

6.1.2. Formal result

We will now make our intuition formal. Our argument above has a subtle issue. The first bound (6.1) is still OK, the second (6.2) is not. We implicitly conditioned on an event $E = \{R(\widehat{h}) - R(h^*) \leq r\}$. This event is *not* independent of the random variables $R_n(h)$. In particular, the samples $(Z_i)_{i=1}^n$ aren't independent conditionally on E, which invalidates our concentration inequalities.

There's a smart way to work around this. Instead of conditioning, we slice the hypothesis space into *shells*

$$\mathcal{S}_k = \left\{h \in \mathcal{H} \colon 2^k a_n < R(h) - R(h^*) \leq 2^{k+1} a_n\right\},$$

where a_n is the convergence rate we aim for, e.g., $a_n = 1/n$. This is illustrated in Fig. 6.2. You can now think of the hypothesis space $\mathcal{H}$ as an onion. Its *core* $\{h \in \mathcal{H} \colon R(h) - R(h^*) \leq 2^M a_n\}$ is surrounded by a sequence of small shells $\mathcal{S}_M, \mathcal{S}_{M+1}, \ldots$. We want to show that $\widehat{h}$ most likely falls inside the core. We can

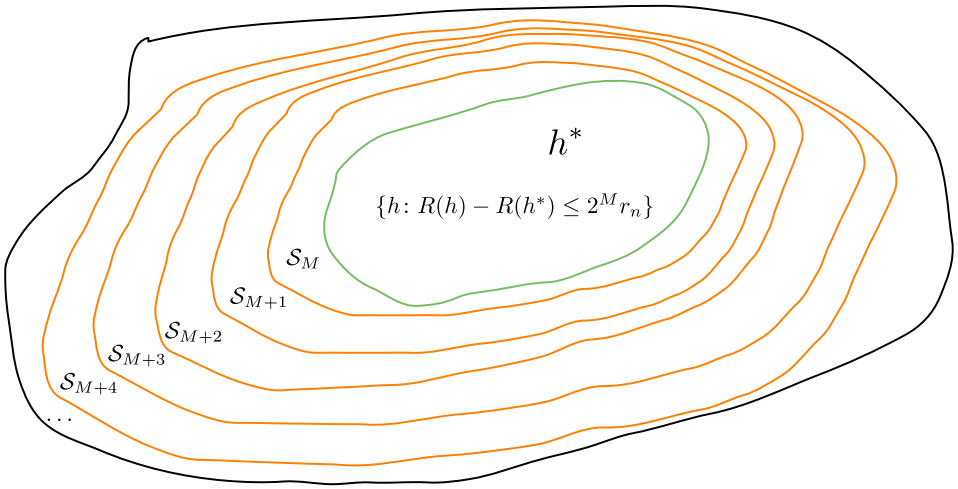

Figure 6.2.: The formal proof slices the hypothesis space into infinitely many shells $\mathcal{S}_k$ and bound the excess risk probability by peeling them off.

peel off the shells $\mathcal{S}_k$ around the core one by one and bound the probabilities $\mathbb{P}(\widehat{h} \in \mathcal{S}_k)$ individually. In particular, we have

$$\mathbb{P}\Big\{R(\widehat{h}) - R(h^*) > 2^M a_n\Big\} = \sum_{k=M}^{\infty} \mathbb{P}\Big\{\widehat{h} \in \mathcal{S}_k\Big\}.$$

In each shell, the excess risk $R(\widehat{h}) - R(h^*)$ is upper and lower bounded. This allows us to restrict the supremum of $R(h) - R_n(h)$ to a smaller set without conditioning. The full proof will be given at the end of this section.

Theorem 6.1.1. *Assume the loss function is Lipschitz and there are $C < \infty$ and $\alpha \in [0, 1]$ such that*

$$\mathcal{R}_n(\mathcal{H}) \le C\sqrt{\frac{1}{n}} \quad \text{and} \quad \mathcal{R}_n(\mathcal{H}_r) \le C\sqrt{\frac{r^\alpha}{n}}. \tag{6.3}$$

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

$$R(\widehat{h}) - R(h^*) \lesssim \left(\frac{(C/\delta)^2}{n}\right)^{\frac{1}{2-\alpha}}.$$

Let's compare this to a standard bound implied by Theorem 5.2.4 and (6.3):

$$R(\widehat{h}) - R(h^*) \lesssim C\left(\frac{\ln(1/\delta)}{n}\right)^{\frac{1}{2}}. \tag{6.4}$$

We distinguish three cases:

- **Case $\alpha = 0$:** Theorem 6.1.1 gives

$$R(\widehat{h}) - R(h^*) \lesssim C\left(\frac{1/\delta^2}{n}\right)^{\frac{1}{2}}.$$

  The scaling in C and n is the same, but the scaling in δ is worse than our old bound (6.4). It scales like $O(1/\delta^2)$ instead of $O(\ln(1/\delta))$. This is an artifact of the proof. The scaling can often be improved by more tedious arguments involving *Talagrand's inequality* (see, e.g., Bousquet et al., 2004, Section 6.4), a generalization of McDiarmid's inequality that also takes the variance into account.

- **Case $\alpha = 1$:** Theorem 6.1.1 gives

$$R(\widehat{h}) - R(h^*) \lesssim \frac{C^2/\delta^2}{n}.$$

  Apart from the scaling in δ, we have essentially squared the old bound (6.4). In particular, we have the fast rate $O(1/n)$ instead of $O(1/\sqrt{n})$. This is traded off by a worse dependence in C. This trade-off is unavoidable.

- **Case $\alpha \in (0, 1)$:** This case interpolates between the two extremes and gives fast rates of the order $O(1/n^{2-\alpha})$.

We can achieve a fast rate when condition (6.3) is satisfied with $\alpha > 0$. The constant C is normally related to the complexity of the full class $\mathcal{H}$. The relative influence of C and n is the same, irrespective of α. That's why we're often satisfied with studying cruder bounds with slow rates. Apart from the exact rate of convergence, they provide the same insight but are much easier to derive.

Proof of Theorem 6.1.1 (optional). We start by rewriting the probabilities. Define $a_n = n^{-1/(2-\alpha)}$ and the shells

$$\mathcal{S}_k = \left\{h \in \mathcal{H} \colon 2^k a_n < R(h) - R(h^*) \leq 2^{k+1} a_n\right\}.$$

Then $\widehat{h} \in \mathcal{S}_k$ implies

$$\mathbb{1}_{\mathcal{S}_k}(\widehat{h})[R(\widehat{h}) - R(h^*)] > 2^k a_n.$$

Since $R_n(\widehat{h}) \leq R_n(h^*)$, we have

$$\begin{aligned} R(\widehat{h}) - R(h^*) = (R - R_n)(\widehat{h}) - (R - R_n)(h^*) + R_n(\widehat{h}) - R_n(h^*) &\leq (R - R_n)(\widehat{h}) - (R - R_n)(h^*) \\ &= (P - P_n)(\ell_{\widehat{h}} - \ell_{h^*}), \end{aligned}$$

where $\ell_h = L \circ h$. Note that $(P - P_n)(\ell_{\widehat{h}} - \ell_{h^*}) \geq 0$ because $R(\widehat{h}) \geq R(h^*)$ by definition of h^*. Then also $\sup_{h \in \mathcal{S}}(P - P_n)(\ell_h - \ell_{h^*})$ is positive and Markov's inequality gives

$$\mathbb{P}\left\{\widehat{h} \in \mathcal{S}_k\right\} \leq \mathbb{P}\left\{\mathbb{1}_{\mathcal{S}_k}(\widehat{h})[R(\widehat{h}) - R(h^*)] > 2^k a_n\right\} \leq \mathbb{P}\left\{\sup_{h \in \mathcal{S}_k}(P - P_n)(\ell_h - \ell_{h^*}) > 2^k a_n\right\}$$

$$\leq \frac{\mathbb{E}\big[\sup_{h\in\mathcal{S}_k}(P-P_n)(\ell_h-\ell_{h^*})\big]}{2^k a_n}.$$

Defining $\mathcal{L}_k = \{\ell_h : h \in \mathcal{S}_k\}$, we have

$$\mathbb{E}\Big[\sup_{h\in\mathcal{S}_k}(P-P_n)(\ell_h-\ell_{h^*})\Big] \lesssim \mathcal{R}_n(\mathcal{L}_k-\ell_{h^*}) = \mathcal{R}_n(\mathcal{L}_k) \lesssim \mathcal{R}_n(\mathcal{S}_k),$$

where we used shift invariance of $\mathcal{R}_n$ (Proposition 5.2.9 iii) in the second and the contraction lemma (Lemma 5.2.5) in the third step.

Since $\mathcal{S}_k \subset \{h : R(h) - R(h^*) \leq 2^{k+1}a_n\}$, assumption Eq. (6.3) gives

$$\mathcal{R}_n(\mathcal{S}_k) \lesssim C\sqrt{\frac{2^{k\alpha}a_n^\alpha}{n}}.$$

Altogether, we have shown

$$\mathbb{P}\big\{\widehat{h} \in \mathcal{S}_k\big\} \lesssim \frac{2^{k\alpha/2}C\sqrt{a_n^\alpha/n}}{2^k a_n} = 2^{-k(1-\alpha/2)}C.$$

Summing up the probabilities,

$$\mathbb{P}\big\{R(\widehat{h}) - R(h^*) \geq 2^M/n\big\} = \sum_{k=M}^{\infty}\mathbb{P}\big\{\widehat{h}\in\mathcal{S}_k\big\} \lesssim C\sum_{k=M}^{\infty}2^{-k(1-\alpha/2)} \lesssim C2^{-M(1-\alpha/2)},$$

by the geometric series. Choosing $2^{-M} = (\delta/C)^{2/(2-\alpha)}$, we have shown that with probability at least $1-\delta$,

$$R(\widehat{h}) - R(h^*) \lesssim \left(\frac{C}{\delta}\right)^{2/(2-\alpha)} a_n. \qquad \square$$

6.1.3. When fast rates are possible

So how do we establish a condition like (6.3)? A typical strategy consists of two ingredients. The first relates the excess risk to the $L_2(P)$-norm on $\mathcal{H}$. Suppose the loss function L is such that

$$R(h) - R(h^*) \gtrsim \|h - h^*\|_{L_2(P)}^\kappa, \tag{6.5}$$

for some $\kappa > 0$. The estimation error $R(\widehat{h}) - R(h^*)$ upper bounds the $L_2(P)$-distance between $\widehat{h}$ and h^*. So if the estimation error is small, $\widehat{h}$ concentrates on a small $L_2(P)-$neighborhood of h^*. The radius of this neighborhood shows up in a version of Dudley's covering entropy bound (see, Van der Vaart and Wellner, 1996, Lemma 3.4.2, for a version with bracketing entropy).

Lemma 6.1.2. *For $\mathcal{H}_r = \{h \in \mathcal{H} : \|h-h^*\|_{L_2(P)}^\kappa \leq r\}$ and $\sup_{h\in\mathcal{H}}\|h\|_\infty < \infty$, it holds*

$$\mathcal{R}_n(\mathcal{H}_r) \lesssim \frac{J(r^{1/\kappa},\mathcal{H})}{\sqrt{n}} + \frac{J(r^{1/\kappa},\mathcal{H})^2}{r^{2/\kappa}n},$$

where $J(\delta,\mathcal{H}) = \int_0^\delta \sup_Q \sqrt{\ln N(\epsilon,\mathcal{H},L_2(Q))}d\epsilon$.

Example 6.1.3. *As an example, Euclidean classes (Definition 5.4.5) satisfy (up to logarithmic factors)*

$$J(r, \mathcal{H}) \lesssim r\sqrt{V}.$$

Substituting in Lemma 6.1.2 gives

$$\frac{J(r^{1/2}, \mathcal{H})}{\sqrt{n}} + \frac{J(r^{1/2}, \mathcal{H})^2}{rn} \lesssim \sqrt{\frac{r^{2/\kappa} V}{n}} + \frac{V}{n} \lesssim \sqrt{\frac{r^{2/\kappa} V}{n}}, \quad \text{(assuming } r^{2/\kappa} \geq V/n\text{)}$$

so condition (6.3) holds with $\alpha = 2/\kappa$. Under condition (6.5), the risk of Euclidean classes of algorithms converges at the fast rate $O(n^{-1/(2-2/\kappa)})$.

The question remains when a condition like (6.5) holds. Mean regression and classification are two important examples.

Mean regression

Consider the square loss $L(y, h(x)) = (y - h(x))^2$. Assume for simplicity that

$$Y = h^*(X) + \epsilon, \qquad \mathbb{E}[\epsilon \mid X] = 0,$$

for some function $h^* \in \mathcal{H}$. Because the noise ϵ is unavoidable, there is a minimum risk that all $h \in \mathcal{H}$ share. It disappears when we look at the difference between two risks. The difference is a pure distance measure between h and h^*, irrespective of the noise.

Lemma 6.1.4. *The square risk satisfies*

$$R(h) - R(h^*) = \|h - h^*\|^2_{L_2(P)}.$$

Proof. Exercise. □

The scaling of the square-loss translates into a squared scaling for the error $\widehat{h} - h^*$. That's why fast rates are possible. This shouldn't be surprising. For example, take the linear regression model $h_\beta(x) = \beta^\top x$. The OLS estimator satisfies $\widehat{\beta} - \beta = O(1/\sqrt{n})$. Then Lemma 6.1.4 implies

$$R(h_{\widehat{\beta}}) - R(h_{\beta^*}) = \|h_{\widehat{\beta}} - h_{\beta^*}\|_{L_2(P)} = O(\|\widehat{\beta} - \beta\|^2) = O(1/n).$$

We could've derived this already in an early course on statistical inference. Our results here are much more general. We now know that empirical risk minimizers over Euclidean hypothesis classes achieve a fast rate. In particular, Theorem 6.1.1 also applies to basis expansions and tree ensembles. Because the rate is now sharp, the bounds can be used to determine optimal hyperparameters as in Section 5.3.4 and exact convergence rates.

Binary classification

Now consider binary classification with the 0-1 loss $L(y, h(x)) = \mathbb{1}\{y \neq h(x)\}$, where $h\colon \mathcal{X} \to \{-1, 1\}$. Denote $\eta(x) = P(Y = 1 \mid X = x)$ and the Bayes' classifier by

$$h^*(x) = \operatorname{sign}[\eta(x) - 1/2].$$

The corresponding excess risk is

$$R(h) - R(h_0) = P\{Y \neq \operatorname{sign} h(X)\} - P\{Y \neq \operatorname{sign} h_0(X)\}.$$

Lemma 6.1.5. *If*

$$c := \inf_{x \in \mathcal{X}} |\eta(x) - 1/2| > 0, \tag{6.6}$$

the 0-1-risk satisfies

$$R(h) - R(h^*) \geq \frac{c}{4} \|h - h^*\|^2_{L_2(P)}.$$

Proof (optional). First note that

$$\begin{aligned} R(h) = \mathbb{E}_{Y,X}[\mathbb{1}\{Y \neq h(X)\}] &= \mathbb{E}_X\left[\eta(X)\mathbb{1}\{1 \neq h(X)\} + [1 - \eta(X)]\mathbb{1}\{-1 \neq h(X)\}\right] \\ &= \mathbb{E}_X\left[\eta(X)\mathbb{1}\{1 \neq h(X)\} + [1 - \eta(X)][1 - \mathbb{1}\{1 \neq h(X)\}]\right] \\ &= \mathbb{E}_X\left[[2\eta(X) - 1]\mathbb{1}\{1 \neq h(X)\}\right] + 1 - \mathbb{E}_X[\eta(X)]. \end{aligned}$$

Therefore,

$$R(h) - R(h^*) = \mathbb{E}_X\left[[2\eta(X) - 1][\mathbb{1}\{1 \neq h(X)\} - \mathbb{1}\{1 \neq h^*(X)\}]\right] = \mathbb{E}_X\left[|2\eta(X) - 1|\mathbb{1}\{h(X) \neq h^*(X)\}\right],$$

where the last equality comes from the Bayes classifier h^* predicting 1 if and only if $2\eta(X) - 1 \geq 0$. Massart's condition (6.6) now gives

$$R(h) - R(h^*) \geq c\mathbb{P}\{h(X) \neq h^*(X)\}\big] = \frac{c}{4}\mathbb{E}[|h(X) - h^*(X)|^2] = \frac{c}{4}\|h - h^*\|^2_{L_2(P)}. \qquad \square$$

Equation (6.6) is known as *Massart's noise condition* and has an intuitive interpretation. It forces the true class probabilities $\eta(x)$ and $1 - \eta(x)$ away from $1/2$. Samples with $\eta(x) = 1/2$ are considered hard because it's impossible to do better than random guessing. When $\eta(x)$ is bounded away from $1/2$, the classification task is much easier. Why? An algorithm $\widehat{h}$ (implicitly or explicitly) estimates the probability $\eta(x)$ by $\widehat{\eta}(x) = 1/2 + \widehat{h}(x)$ and returns $\operatorname{sign}\widehat{h}(x) = \operatorname{sign}[\widehat{\eta}(x) - 1/2]$. If $c > 0$, estimation errors $|\widehat{\eta}(x) - \eta(x)|$ smaller than c become completely irrelevant. Hence, we can get away with relatively large estimation errors and still classify optimally.

An important case where Massart's condition applies is when the problem is *realizable*, i.e., $\mathbb{P}(Y = 1 \mid X) \in \{0, 1\}$. Here the Bayes classifier makes zero errors, so realizable classification problems are maximally easy. In other situations, the condition is rather unrealistic. For example, it excludes cases where η is a

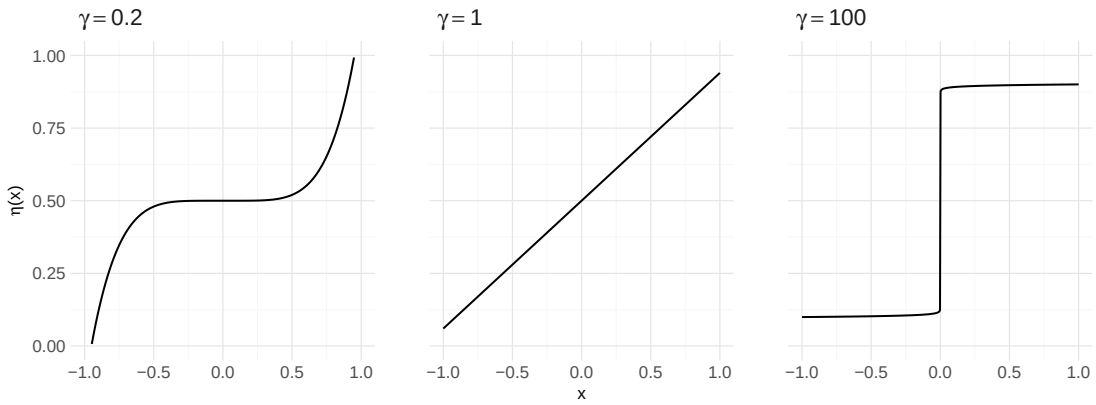

Figure 6.3.: Visualization of the "easiness parameter" γ in Tsybakov's noise condition. The graphs show the true classification probability $\eta(x) = \mathbb{P}(Y = 1 \mid X = x)$ under three noise scenarios.

continuous function ranging from -1 to 1. To get fast rates, it is enough that the region of $\mathcal{X}$ where $\eta(x) \approx 1/2$ is small.

Lemma 6.1.6. *If for every $t > 0$ and some $\gamma > 0$, $C < \infty$,*

$$\mathbb{P}\{|\eta(X) - 1/2| \leq t\} \leq Ct^\gamma, \tag{6.7}$$

the 0-1-risk satisfies

$$R(h) - R(h^*) \geq 8(8C)^{-1/\gamma} \|h - h^*\|_{L_2(P)}^{2+2/\gamma}.$$

Proof (optional). The proof is similar to Lemma 6.1.5, but using the new noise condition in the last step:

$$\begin{aligned}
R(h) - R(h^*) &= \mathbb{E}_X\left[|2\eta(X) - 1| \mathbb{1}\{h(X) \neq h^*(X)\}\right] \\
&\geq t\mathbb{E}_X\left[\mathbb{1}\{h(X) \neq h^*(X)\}\mathbb{1}\{|2\eta(X) - 1| \geq t\}\right] \\
&\geq t\mathbb{E}_X\left[\mathbb{1}\{h(X) \neq h^*(X)\}\right] - t\mathbb{E}_X[\mathbb{1}\{|2\eta(X) - 1| \leq t\}] \\
&\geq t\mathbb{E}_X\left[\mathbb{1}\{h(X) \neq h^*(X)\}\right] - Ct^{\gamma+1} \\
&\geq t\frac{1}{4}\|h - h^*\|_{L_2(P)}^2 - Ct^{\gamma+1}.
\end{aligned}$$

Now choose $t = (8C)^{-1/\gamma}\|h - h^*\|_{L_2(P)}^{2/\gamma}$. □

Equation Eq. (6.7) is *Tsybakov's noise condition*. Lemma 6.1.5 can be seen as a special case with $\gamma = \infty$. Condition (6.7) ensures that areas with $\eta(X) \approx 1/2$ have low probability. The condition is illustrated in Fig. 6.3 for $X \sim \text{Uniform}[-1, 1]$ and three functions $\eta(x)$ with $\gamma = 0.2, 1, 100$. When γ is large, the classification problem is easy. Most samples x quite clearly belong to either class -1 or 1. On the other hand, when γ is small, there is a fairly large region around $x = 0$, where it's hard to do better than random guessing.

Example 6.1.7. *An application of Theorem 6.1.1 and Lemma 6.1.2 to Euclidean classes now gives the fast rate $O(n^{-(1+\gamma)/(2+\gamma)})$. The "easiness" parameter γ*

interpolates between $O(1/n)$ under Massart noise ($\gamma = \infty$) and the slow rate $O(1/\sqrt{n})$ for $\gamma = 0$.

6.1.4. Application

Let us revisit the ridge regression example from Section 5.3.1. Recall $\mathcal{Z} = \mathbb{R}^d$ and consider the class of norm-constrained linear functions

$$\mathcal{H}_{2,M,r} = \{z \mapsto \beta^\top z \colon \|\beta\|_2 \le M, \|\beta - \beta^*\|_2^2 \le r\}.$$

Proposition 6.1.8. *It holds*

$$\mathcal{R}_n(\mathcal{H}_{2,M,r}) \le \sqrt{\frac{r\mathbb{E}[\|Z\|_2^2]}{n}},$$

and $\widehat{\beta} = \arg\min_{\|\beta\|_2 \le M} R_n(\beta)$ satisfies

$$R(\widehat{\beta}) - R(\beta^*) \lesssim \frac{M^2\mathbb{E}[\|Z\|_2^2]}{n} \quad \text{w.h.p.}$$

Proof. Here's a proof outline; you'll the gaps in the exercises.

1. Observe (why?)

$$\mathbb{E}\left[\sup_{\|\beta\|_2 \le M} \sum_{i=1}^n \varepsilon_i \beta^\top Z_i\right] = \mathbb{E}\left[\sup_{\|\beta\|_2 \le M} \sum_{i=1}^n \varepsilon_i (\beta - \beta^*)^\top Z_i\right].$$

2. Adjust the first steps in the proof of Theorem 5.3.1 to show

$$\mathcal{R}_n(\mathcal{H}_{2,M,r}) \le \frac{\sqrt{r}}{n}\mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i Z_i\right\|_2\right].$$

3. Together with

$$\mathbb{E}\left[\left\|\sum_{i=1}^n \varepsilon_i Z_i\right\|_2\right] \le \sqrt{n}\mathbb{E}[\|Z\|_2^2]. \qquad \square$$

(already shown in Theorem 5.3.1) we can now invoke Theorem 6.1.1 with $\alpha = 1$ and $C = M\sqrt{\mathbb{E}[\|Z\|_2^2]}$.

This is essentially the squared versions of the bound obtained in Section 5.3.1. So a qualititive interpretation of the bounds would lead to the same conclusions.

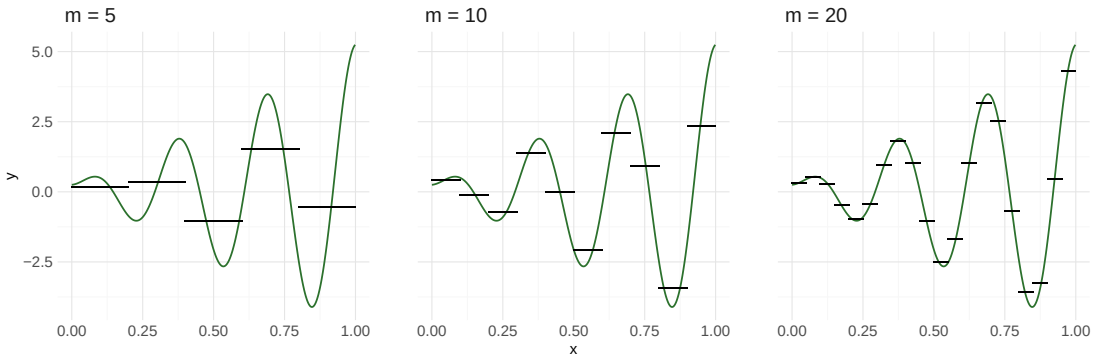

Figure 6.4.: Approximation of a function with piece-wise constant functions on m intervals.

6.2. Approximation error

Recall from (3.1) that

$$R(\widehat{h}) - R_0 = \underbrace{R(\widehat{h}) - R(h^*)}_{\text{estimation error}} + \underbrace{R(h^*) - R(h_0)}_{\text{approximation error}},$$

where $h^* = \arg\min_{h \in \mathcal{H}} R(h)$ is the best hypothesis in class and $h_0 = \arg\min R(h_0)$ the best hypothesis possible. We have gone a long way bounding the estimation error. Except for a small digression in Section 5.3.4. We haven't spoken much about the approximation error. This error is deterministic, so the laws of probability don't help much in its analysis. Bounding the approximation error for given $\mathcal{H}$ is subject of the mathematical field of approximation theory. An in-depth analysis requires new concepts and tools, which go beyond the scope of this course. Another issue is that each approximation method requires a tailor-made analysis. We'll be satisfied with a discussion of the key ideas. Since almost always

$$R(h^*) - R(h_0) \lesssim \sup_{x \in \mathcal{X}} |h^*(x) - h_0(x)|^a =: \|h^* - h_0\|_\infty^a, a > 0,$$

for some $a > 0$, we bound the term on the right to avoid getting distracted by different loss functions.

6.2.1. Pice-wise constant functions

Let's start with a simple function class. Let $\mathcal{X} = [0, 1]$ and

$$\mathcal{H}_m = \left\{ h_\beta(x) = \sum_{k=1}^{m} \beta_k \mathbb{1}_{[\xi_{k-1}, \xi_k)}(x) \colon \beta \in \mathbb{R}^m \right\},$$

be the set of functions that are piece-wise constant on the intervals $[\xi_{k-1}, \xi_k)$, where we take $\xi_0 = 0$ and $\xi_m = 1.1$ by convention. Zero-degree splines and partition trees are of that form. If h_0 is smooth, we can find successively closer

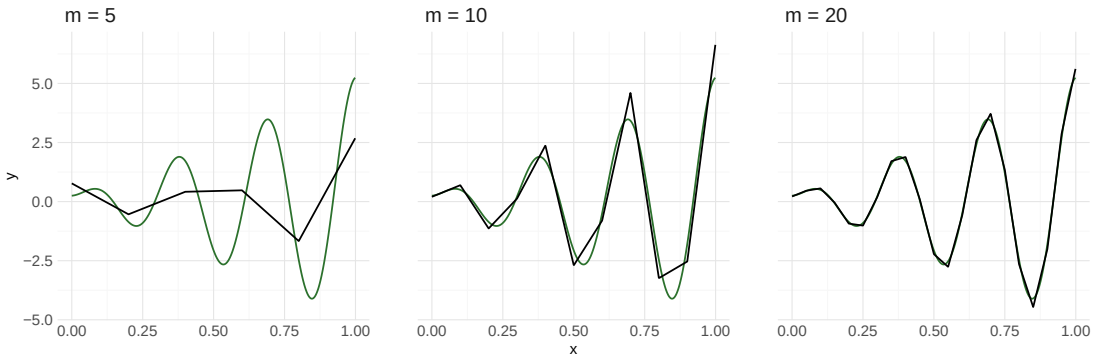

Figure 6.5.: Approximation of a function with piece-wise linear functions on m intervals.

approximations h_β by taking $m \to \infty$. This is illustrated in Fig. 6.4 and formalized in the following.

Theorem 6.2.1. *Let $h_0 \colon [0,1] \to \mathbb{R}$ be a Lipschitz continuous function, i.e,*

$$|h_0(x) - h_0(x')| \leq K|x - x'| \quad \text{for all } x, x' \in [0,1].$$

Then for any $\epsilon > 0$, there are $(\xi_k)_{k=0}^m$ and $\beta \in \mathbb{R}^m$ with $m = O(\epsilon^{-1})$ such that

$$\|h_0(x) - h_\beta(x)\|_\infty \leq \epsilon.$$

Proof. Let x be arbitrary, $\xi_j = j/m$, and k be the index with $x \in [k/m, (k+1)/m)$. Setting $\beta_k = h_0(k/m)$ and noting $|x - k/m| \leq 1/m$, we have

$$\begin{aligned} |h(x) - h_\beta(x)| = |h_0(x) - \beta_k| &= |h_0(x) - h(k/m)| \\ &\leq K|x - k/m| \qquad (h \text{ is } K\text{-Lipschitz}) \\ &\leq K/m. \end{aligned}$$

□

The theorem shows that we can approximate h_0 to arbitrary accuracy, provided we choose m large enough. The more accurate we want to be, the more intervals we need. In the context of risk minimization, the function with $f_\beta^* = \arg\min_{\beta^*} R(h_\beta)$ has risk at most as high as the approximating function found in Theorem 6.2.1. If the loss function L is Lipschitz with respect to h, then $R(h_{\beta^*}) - R(h_{\beta_0}) \lesssim 1/m$. For the square loss, we would get $R(h_{\beta^*}) - R(h_{\beta_0}) \lesssim 1/m^2$.

6.2.2. Exploiting higher-order smoothness

An important assumption of the theorem is that h_0 is K-Lipschitz. The smoothness of the target function plays a big role in approximation theory. There are several definitions of smoothness, typically related to the existence and boundedness of derivatives. Smoother functions are easier to approximate. Knowing that a function is smooth calls for an *inductive bias* in our hypothesis class. By

exploiting smoothness in the way we construct approximations, we can typically achieve higher accuracy with fewer parameters. For example, the piece-wise linear approximations in Fig. 6.5 appear to converge much faster to the true function than the piece-wise constant approximations in Fig. 6.5.

Let's see why that is the case. Denote by $C_K^s([0,1])$ the set of all q-times continuously differentiable functions $[0,1] \to \mathbb{R}$ with all derivatives bounded uniformly by K. Assume for example that $h_0 \in C_K^2([0,1])$. For any fixed $x' \in [0,1]$, a Taylor expansion gives

$$h_0(x) = h_0(x') + h_0^{(1)}(x')(x - x') + R_n,$$

with $|R_n| \leq K|x - x'|^2$. So on any interval $[k/m, (k+1)/m)$, there is a linear function $x \mapsto \beta_{k,1} + \beta_{k,2}x$, such that

$$|h_0(x) - \beta_{k,1} + \beta_{k,2}x| \leq K|x - x'|^2 \leq K/m^2.$$

This indeed has better dependence on m than a piece-wise constant approximation. The following result can be derived from higher-order Taylor expansions.

Theorem 6.2.2. *Let $h_0 \in C_K^s([0,1])$. Then there are $(\xi_k)_{k=0}^m$ and a piece-wise polynomial h_β of degree $s-1$ with $m = O(\epsilon^{-1/s})$ such that*

$$\sup_{x \in [0,1]} |h_0(x) - h_\beta(x)| \leq \epsilon.$$

B-splines form a basis for piece-wise polynomials that satisfy additional smoothness constraints. With quite some additional effort, a result like Theorem 6.2.2 can also be derived for this restricted class (with the same scaling in m).

6.2.3. The curse of dimensionality

So far we have only talked about univariate functions. For multi-dimensional x, a new effect kicks in. Multivariate functions are more complex and harder to approximate. This is known as the curse of dimensionality.

The cause of the curse is illustrated in Fig. 6.6. For univariate x, we partitioned the interval $[0,1]$ into m segments of length $1/m$. This led to an error of at most K/m. If we partition each coordinate in $[0,1]^d$ similarly, we get boxes with length $1/m$ in each coordinate. On each box, we can achieve an error of K/m with a piece-wise constant if h_0 is K-Lipschitz. To cover $[0,1]^d$, we need m^d such boxes.

We now need $m^d = O(\epsilon^{-d})$ parameters to achieve an approximation error of ϵ with piece-wise constants.

The same idea applies to piece-wise polynomial approximations of smoother functions, where we need $O(\epsilon^{-d/s})$ coefficients. Increasing the number of parameters typically increases the complexity of a function class and this may hurt the generalization error. The curse of dimensionality is unavoidable in general. Smoothness

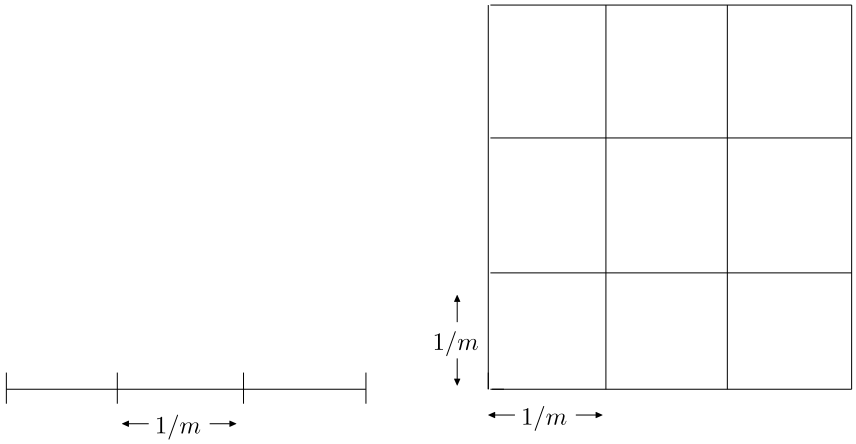

Figure 6.6.: Covering the unit cube with intervals/boxes with side length 1/3. We need exponentially more boxes when d is large.

ameliorates the effect of dimension to some degree, but the exponential scaling in d remains. To overcome the curse, we have to assume more structure.

6.2.4. Exploiting sparsity

The term 'sparsity' has a variety of meanings depending on the context. In our setting, it roughly means that some parts of the problem are irrelevant or particularly simple. Sparsity often allows approximating functions with fewer parameters than usual.

Example 6.2.3 (Irrelevant features)**.** *Suppose that $h_0 \colon [0,1]^d \to \mathbb{R}$ is a function of the first d' variables only. Then we can cover $[0,1]^{d'}$ with $O(\epsilon^{-d'})$ boxes B_k. Because the remaining $d - d'$ variables do not affect the function, a piece-wise constant on the sets $B_k \times [0,1]^{d-d'}$ achieves an error of ϵ with only $O(\epsilon^{-d'})$ parameters. See Fig. 6.7 (left) for an illustration.*

Example 6.2.4 (Manifold structure)**.** *Now suppose that $\mathcal{X} \subset \mathbb{R}^d$ is a d'-dimensional manifold. That is, $\mathcal{X}$ can be covered with $O(\epsilon^{-d'})$ boxes with side-length ϵ. If $h_0 \colon \mathcal{X} \to \mathbb{R}$ is K-Lipschitz, a piece-wise constant function achieves an error of ϵ with only $O(\epsilon^{-d'})$ parameters. See Fig. 6.7 (right) for an illustration.*

Any function $h \colon [0,1]^d \to \mathbb{R}$ can be decomposed as

$$h(x) = \sum_{S \subseteq \{1,\dots,d\}} h_S(x_S),$$

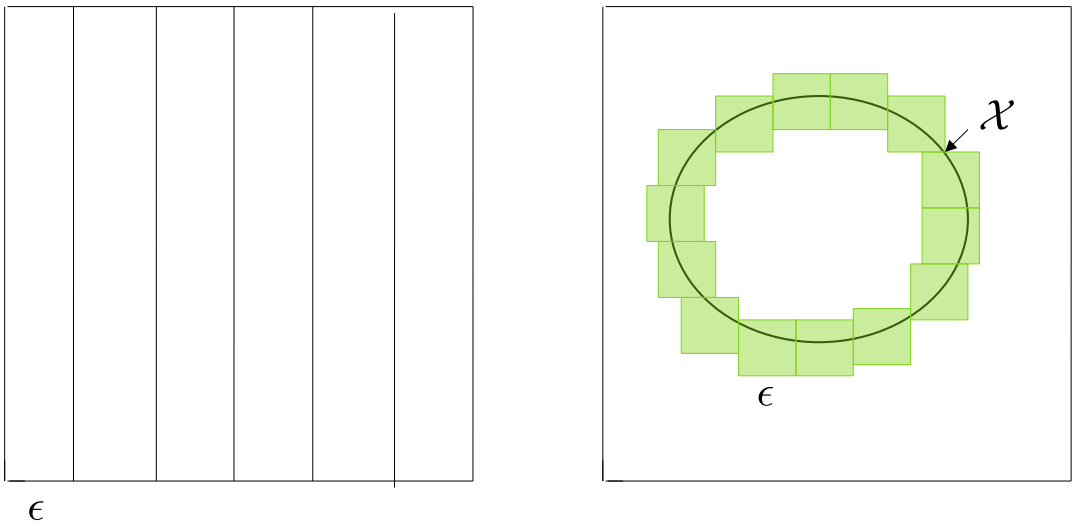

Figure 6.7.: Exploiting sparsity: an $O(\epsilon)$-approximating partition when the second component of x is irrelevant in $h_0(x)$ (left); an ϵ-covering of a 1-dimensional manifold $\mathcal{X}$ only requires $O(\epsilon^{-1})$ boxes.

where $h_{0,S}\colon [0,1]^{|S|} \to \mathbb{R}$ and $x_S = (x_j)_{j\in S}$. To see this, take $h_{\{1,\dots,d\}}(x) = h(x)$ and $h_S(x) = 0$ for $S \neq \{1,\dots,d\}$. But other decompositions are possible. For example, the *so-called functional ANOVA* decomposition has

$$\begin{aligned} h_\emptyset &= \int h(x)dx \\ h_{\{j\}}(x_j) &= \int h(x) \prod_{k\neq j} dx_k - h_\emptyset, \\ h_{\{i,j\}}(x_i,x_j) &= \int h(x) \prod_{k\neq i,j} dx_k - h_{\{i\}}(x_i) - h_{\{j\}}(x_j) + h_\emptyset \\ &\dots \end{aligned}$$

The functions h_S can be understood as 'interactions' between the variables in the set S.

Example 6.2.5 (Limited interaction). *A function h_0 has limited interaction if we can write it as*

$$h_0(x) = \sum_{S \subseteq \{1,\dots,d\}, |S| \leq r} h_{0,S}(x_S), \qquad \text{for some } r < d.$$

In the special case $r = 1$, we call h_0 additive, because

$$h_0(x) = h_{0,\emptyset} + h_{0,1}(x_1) + \cdots + h_{0,d}(x_d).$$

To exploit this type of sparsity, we can approximate each interaction function $h_{0,S}$ separately with a piece-wise constant using at most $O(\varepsilon^{-r})$ parameters by Example 6.2.3. Summing the approximation functions gives overall approximation error ϵ with a total of $O(\varepsilon^{-r})$ parameters.

6.2.5. The effectiveness of neural networks

Recall that an L-layer feed-forward neural network can be written recursively as

$$h_\beta(x) = W^{(L)} x^{(L-1)} + b^{(L)},$$

where

$$x^{(\ell)} = \sigma(W^{(\ell)} x^{(\ell-1)} + b^{(\ell)}), \ell = 1, \dots, L-1,$$

and $x^{(0)} = x$. The parameter

$$\beta = (W^{(1)}, b^{(1)}, \dots, W^{(L)}, b^{(L)}),$$

collects all *weight matrices* $W^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and *biases* $b^\ell \in \mathbb{R}^{N_\ell}$. The *activation function* σ is applied componentwise to vectors.

Neural networks are particularly powerful approximators. The study of their expressivity has a long history. We shall only take a glimpse into the vast literature; a recent review can be found in Gühring et al. (2020). In 1989, several independent works showed that neural networks are *universal function approximators*.

Theorem 6.2.6 (Universal approximation theorem). *Let $h_0 \colon [0,1]^d \to \mathbb{R}$ be a continuous function and the activation function σ be non-polynomial. Then there is a two-layer neural network h such that $\|h_0 - h\|_\infty \leq \epsilon$.*

To be more precise, one can show that one needs $O(\epsilon^{-d/s})$ neurons in the hidden layer to approximate a $C^s_K([0,1]^d)$ function. This is the same scaling as for piece-wise polynomials. In practice, two-layer neural networks are rarely used. Deep networks have additional benefits. One can show that networks with fixed width but unlimited depth are also universal function approximators. By

trading off width and depth, there are often more efficient (= fewer parameters) approximations than when fixing width or depth. To prove such results, one first shows that a shallow neural network can approximate the basis functions of an appropriate system (for example, polynomials, splines, or wavelets). Then such small networks are stacked together and combined in the output layer to give a global approximation.

Deep networks can also efficiently adapt to the types of sparsity discussed in the previous section. An important case where deep networks shine is when h_0 has a compositional structure. For example, assume $x \in [0,1]^d$ and

$$h_0(x) = h_1\Big(h_{2,1}(x_1, x_2), h_{2,2}(x_3, x_4)\Big).$$

To approximate this function, construct three separate networks approximating $h_1, h_{2,1}$, and $h_{2,2}$. Then stack the two latter networks and feed their outputs into the network approximating h_1. If all functions are Lipschitz, we require only $O(\epsilon^{-2})$ parameters. With shallow networks, this compositional construction isn't possible and we need $O(\epsilon^{-d}) = O(\epsilon^{-4})$ parameters. We'd also need so many coefficients in spline, wavelet, and other linear basis expansions.

In summary, neural networks can efficiently approximate various function classes and adapt to a variety of smoothness and sparsity patterns.

6.3. Lower bounds and minimax risk

6.3.1. Motivation

A combination of bounds on the approximation and estimation error gives us upper bounds for the excess risk $R(\widehat{h}) - R(h_0)$. Recall again that

$$R(\widehat{h}) - R_0 = \underbrace{R(\widehat{h}) - R(h^*)}_{\text{estimation error}} + \underbrace{R(h^*) - R(h_0)}_{\text{approximation error}},$$

A combination of bounds on the estimation and approximation error gives us upper bounds for the excess risk $R(\widehat{h}) - R(h_0)$. The estimation error grows in the complexity $\mathcal{C}(\mathcal{H})$ of the hypothesis class; the approximation error decreases. Combining bounds on both normally gives something like

$$R(\widehat{h}) - R_0 \lesssim \underbrace{\left(\frac{\mathcal{C}(\mathcal{H})}{n}\right)^a}_{\text{estimation error}} + \underbrace{\left(\frac{1}{\mathcal{C}(\mathcal{H})}\right)^b}_{\text{approximation error}}.$$

Depending on the exponents a and b, verify that the optimal complexity and rate of convergence are

$$\mathcal{C}(\mathcal{H})^* \propto n^{a/(a+b)}, \quad R(\widehat{h}) - R_0 \lesssim n^{-ab/(a+b)}.$$

For any given algorithm, we can use estimation/approximation error bounds to find a and b. But how do we know whether the bound/algorithm is good? Simply choosing $\widehat{h} = h_0$ gives zero excess risk. So is any algorithm that doesn't have zero excess risk bad? Of course not. Choosing $\widehat{h} = h_0$ isn't practical because h_0 depends on the unknown probability measure P. In reality, we construct $\widehat{h}$ by running an algorithm $\mathcal{A}$ on observed data $\mathcal{D}_n = (Z_i)_{i=1}^n$. We want this algorithm to work for many different distributions $P \in \mathcal{P}$ simultaneously. In that case, there are limits on what's possible.

We've already seen an extreme example in the no-free-lunch theorem (Theorem 2.6.1). But even when learning is possible for a class of probabilities $\mathcal{P}$, there are lower bounds on how fast we can learn. The smallest risk that is achievable uniformly over $\mathcal{P}$ is called *minimax risk*. It minimizes (over algorithms) the maximum (over probability measures) risk. In the following, we formalize this notion, discuss some known lower bounds, and how they can be derived.

6.3.2. Definition

To simplify matters, let's consider the expected risk in what follows. Let $\mathcal{P}$ be a set of probability measures on $\mathcal{Z}$, $\mathcal{D}_n$ a data set generated *iid* from $P \in \mathcal{P}$, and $\widehat{h} = \mathcal{A}(\mathcal{D}_n)$ the output of the algorithm $\mathcal{A}$. Denote the function/parameter of interest as $h_0 = h_P$. We are interested in bounds on the worst-case excess risk

$$\sup_{P \in \mathcal{P}} \mathbb{E}_P\Big[R_P(\mathcal{A}(\mathcal{D}_n)) - R_P(h_P)\Big].$$

The expectation is over the random output $\widehat{h}$ of the algorithm $\mathcal{A}(\mathcal{D}_n)$. Here, 'algorithm' simply means 'function of $\mathcal{D}_n$'. Also the risk functionals R_P depend on the probability measure P, because $R_P(h) = \mathbb{E}_{Z \sim P}[L(Z, h(Z))]$. This dependence is somewhat annoying, so it is customary to write this in a slightly different way.

Definition 6.3.1. *Let $d(h, h')$ be some measure of distance between h, h'. The* ***minimax risk*** *with respect to d is defined as*

$$R_n^*(\mathcal{P}) = \inf_{\widehat{h}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\widehat{h}, h_P)],$$

where the infimum is taken over all algorithms $\mathcal{A}$ producing $\widehat{h}$.

The risk is called minimax because we minimize over the algorithm $\mathcal{A}$ and maximize over the probability measure $P \in \mathcal{P}$.

It's hard to compute the minimax risk exactly. Instead, we aim for matching lower and upper bounds (up to constants).

Definition 6.3.2. *If there constants c, C with $0 < c \leq C < \infty$ and*

$$ca_n \leq \inf_{\widehat{h}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[d(\widehat{h}, h_P)] \leq Ca_n,$$

*we call a_n the **minimax rate of convergence**. Any algorithm $\widehat{h}$ attaining the upper bound is called **minimax optimal**.*

For the upper bound, we just have to find an algorithm that attains the rate a_n. For the lower bounds, we need new tools.

6.3.3. Lower bounds for the minimax risk

Let's start with some intuition. Suppose that there are two distributions P_1, P_2 that are very similar, but the true hypotheses h_1, h_2 implied by P_1, P_2 are far apart. Then any algorithm will have a hard time distinguishing between a data set from P_1 and a data set from P_2. It's difficult to decide whether it should put $\widehat{h}$ closer to h_1 or h_2. Because the hypotheses are far apart, the algorithm cannot produce a hypothesis that is close to both. So the maximal risk over $\{P_1, P_2\}$ has to be large.

More formally, suppose that $d(h_1, h_2) \geq \Delta > 0$. The triangle inequality implies

$$\Delta \leq d(h_1, h_2) \leq d(\widehat{h}, h_1) + d(\widehat{h}, h_2) \leq 2 \max\{d(\widehat{h}, h_1), d(\widehat{h}, h_2)\},$$

so any $\widehat{h}$ must satisfy $\max\{d(\widehat{h}, h_1), d(\widehat{h}, h_2)\} \geq \Delta/2$. A convenient measure of similarity between two probability measures is the *Kullback-Leibler divergence* or just *KL-divergence*:

$$\mathrm{KL}(P_1 \,||\, P_2) = \int \log\left(\frac{dP_1(z)}{dP_2(z)}\right) dP_1(z).$$

You can think of dP as the density or probability mass function, depending on whether Z is continuous or discrete. We normally prove lower bounds with simple distributions.

Example 6.3.3 (KL divergence of Gaussians)**.** *If $P_1 = \mathcal{N}(\mu_1, \sigma^2)$ and $P_2 = \mathcal{N}(\mu_2, \sigma^2)$, then*

$$\mathrm{KL}(P_1 \,||\, P_2) = (\mu_1 - \mu_2)^2/(2\sigma^2).$$

Example 6.3.4 (KL divergence of Bernoulli)**.** *If $P_1 = \mathrm{Bernoulli}(p_1)$ and $P_2 = \mathrm{Bernoulli}(p_2)$, then*

$$\mathrm{KL}(P_1 \,||\, P_2) = p_1 \log\left(\frac{p_1}{p_2}\right) + (1 - p_1) \log\left(\frac{1 - p_1}{1 - p_2}\right)$$

Now we can formalize our intuition above: when P_1, P_2 are similar but h_{P_1}, h_{P_2} are not, the minimax risk is lower bounded.

Theorem 6.3.5 (Le Cam's method). *For any $P_1, P_2 \in \mathcal{P}$, it holds*

$$\inf_{\widehat{h}} \sup_{P\in\mathcal{P}} \mathbb{E}_P[d(\widehat{h}, h_P)] \geq \frac{d(h_1, h_2)}{8} e^{-nKL(P_1||P_2)}.$$

The lower bound on the right increases in the distance $d(\widehat{h}, h_P)$ between h_1 and h_2 and decreases in the distance $KL(P_1 \,||\, P_2)$ between the corresponding probability measures. In concrete learning problems, the difficulty is now to find a similar pair P_1, P_2 for which h_1, h_2 are sufficiently separated. As a simple example, suppose we want to learn the mean $\mu_P = \mathbb{E}_P[Z]$.

Proposition 6.3.6. *Let $\mathcal{P}$ be the set of all probability measures with variance bounded by $C < \infty$. Then there is a constant $c > 0$ such that*

$$\frac{c}{\sqrt{n}} \leq \inf_{\widehat{\mu}} \sup_{P\in\mathcal{P}} \mathbb{E}_P[|\widehat{\mu} - \mu_P|] \leq \frac{C}{\sqrt{n}}.$$

In particular, $n^{-1/2}$ is the minimax rate of convergence and the sample mean is minimax optimal.

Proof. Define $P_1 = \mathcal{N}(\mu_1, 1)$ and $P_2 = \mathcal{N}(\mu_2, 1)$ and $d(\mu_1, \mu_2) = |\mu_1 - \mu_2|$. Taking $\mu_1 = \mu_2 + 1/\sqrt{n}$, Theorem 6.3.5 and Example 6.3.3 imply

$$\inf_{\widehat{\mu}} \sup_{P\in\mathcal{P}} \mathbb{E}_P[|\widehat{\mu} - \mu_P|] \geq \frac{1/\sqrt{n}}{8} e^{-n\times(n^{-1}/2)} = \frac{c}{\sqrt{n}}$$

with $c = e^{-1/2}/8$. We have found a lower bound for the minimax risk. Finding an upper bound is easy. The sample mean $\widehat{\mu} = n^{-1}\sum_{i=1}^n Z_i$ satisfies

$$\mathbb{E}_P[|\widehat{\mu} - \mu_P|] \leq \mathbb{E}_P[(\widehat{\mu} - \mu_P)^2]^{1/2} = \mathbb{V}\mathrm{ar}_P[\widehat{\mu}]^{1/2} \leq \frac{C}{\sqrt{n}},$$

for any $P \in \mathcal{P}$. □

Le Cam's method is useful when there is a single number to learn. For multi-dimensional or infinite-dimensional (= learning functions) problems, we need more than two probability measures to compare. The following result comes in handy

Theorem 6.3.7 (Fano's method). *For any $P_1, \ldots, P_N \in \mathcal{P}$, it holds*

$$\inf_{\widehat{h}} \sup_{P\in\mathcal{P}} \mathbb{E}_P[d(\widehat{h}, h_P)] \geq \frac{\min_{j\neq k} d(h_j, h_k)}{2} \left(1 - \frac{n \max_{j\neq k} \mathrm{KL}(P_j \,||\, P_k)}{\log N}\right).$$

Note that finding *any* lower bound does not suffice. We need to find the largest possible lower bound. In many interesting applications, finding appropriate measures P_j is difficult and the number N of measures needs to grow with n. We won't get into that, but you should get the idea.

6.3.4. Some examples

Learning smooth functions

We close with a few examples of known minimax rates. Take $\mathcal{P}$ as the set of measures P with

$$Y = h_P(X) + \varepsilon, \quad \mathbb{E}[\varepsilon \mid X] = 0, \mathbb{V}\mathrm{ar}[Y] < \infty,$$

where $h_P \in C_K^s$ (see Example 5.4.10). We get the following minimax rates:

Distance	Sparsity	minimax rate
$\lvert h_1(x) - h_2(x)\rvert$		$n^{-s/(2s+d)}$
$\sup_x \lvert h_1(x) - h_2(x)\rvert$		$(n/\ln n)^{-s/(2s+d)}$
$\int \lvert h_1(x) - h_2(x)\rvert dx$		$n^{-s/(2s+d)}$
$\int \lvert h_1(x) - h_2(x)\rvert^2 dx$		$n^{-2s/(2s+d)}$
	d' relevant features	$n^{-2s/(2s+d')}$
	d'-dimensional manifold	$n^{-2s/(2s+d')}$
	additive	$n^{-2s/(2s+1)}$
	r-limited interaction	$n^{-2s/(2s+r)}$.

Table 6.1.: Minimax rates for regression problems.

Essentially the same rates hold for most other problems where smooth functions are the target, like quantile regression and density estimation. We can see the curse of dimensionality acting in these rates, and how sparsity can help.

Classification

Classification is different. The excess risk for the zero-one-loss is

$$R_P(h) - R_P(h_P) = P\{Y \neq h(X)\} - P\{Y \neq h_0(X)\}.$$

We don't really learn a smooth function h_P, but its binarized version $\operatorname{sign} h_P$. This is the same as learning the set S_P on which $\operatorname{sign} h_P(x) = 1$. In place of smoothness of the function h_P, we care about the complexity of the decision sets S_P. Mammen and Tsybakov (1999) formalized this by the sets' covering number for a particular distance satisfying $d(S, S_P) = R_P(h) - R_P(h_P)$. Their condition reads

$$\ln N(\epsilon, S_P, d) \lesssim \epsilon^{-\rho} \quad \text{for some } \rho > 0.$$

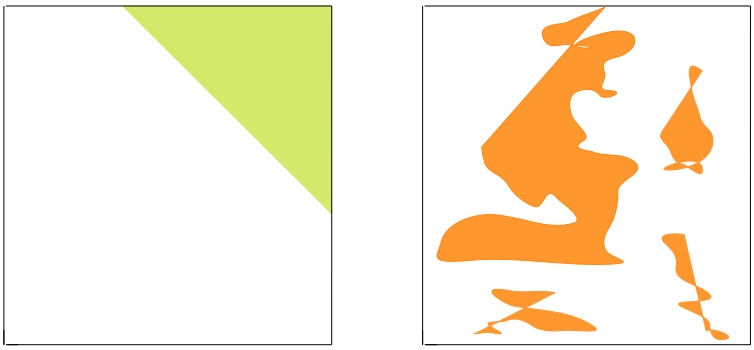

Figure 6.8.: Examples of easy (left) and difficult (right) decision boundaries.

For large ρ we need more balls to cover the decision sets. The collection of sets is more complex, so this corresponds to harder classification problems. For example, classes of decision boundaries similar to the one in the left panel of Fig. 6.8 are easy and can be covered with few balls. Classes of decision boundaries similar to the right panel are hard.

Recall Tsybakov's noise condition (6.7),

$$\mathbb{P}_X\{|2h_P(X) - 1/2| \leq t\} \leq Ct^\gamma,$$

and that $\gamma > 0$ controls the easiness of the problem (large γ is easier). Then the minimax rate is

$$a_n = \begin{cases} n^{-\frac{1+\gamma}{2+\gamma(1+\rho)}}, & \text{if } \rho < 1, \\ n^{-\frac{1}{2}} \ln n, & \text{if } \rho = 1, \\ n^{-\frac{1}{1+\rho}}, & \text{if } \rho > 1. \end{cases}$$

We see several interesting effects. The minimax rate is split into three cases depending on ρ. When ρ is large, the decision boundary of the sets is complicated. If $\rho \geq 1$, we can't do better than $a_n = 1/\sqrt{n}$. Tsybakov's noise condition becomes irrelevant to the easiness of the problem. If $\rho < 1$, how good we can do depends on both the value of ρ and Tsybakov's exponent γ. If we take $\gamma \to \infty$, the rate approaches $a_n = (1/n)^{1/(1+\rho)}$, which is always better than $1/\sqrt{n}$. And if ρ is close to zero, we have $a_n \approx 1/n$. This is the maximally easy case: Samples close to the decision boundary are unlikely (γ large) and the decision boundary itself is simple (small ρ).

7. Closing remarks

The 'Further topics' chapter could go on for a long time: PAC-Bayes, margin bounds, online learning, implicit regularization, transfer learning, and so on. Each has their own concepts and ideas. Some of them I might cover in future iterations. This course focused on core concepts and techniques in learning theory and provides an entry to current research close to this core.

The current hot topic is understanding the success of deep learning. Most standard results can't explain it because bounds are too loose. Here are some starting points:

- Zhang et al. "Understanding deep learning requires rethinking generalization" https://arxiv.org/abs/1611.03530
- Hastie et al. "Surprises in High-Dimensional Ridgeless Least Squares Interpolation" https://arxiv.org/abs/1903.08560
- Belkin et al. "Reconciling modern machine learning practice and the bias-variance trade-off" https://arxiv.org/abs/1812.11118
- Belkin et al. "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation" https://arxiv.org/abs/2105.14368
- Bubeck and Selke "A Universal Law of Robustness via Isoperimetry" https://arxiv.org/abs/2003.00307
- Lotfi et al. "PAC-Bayes Compression Bounds So Tight That They Can Explain Generalization" https://arxiv.org/abs/2211.13609

Bibliography

Bousquet, O., Boucheron, S., and Lugosi, G. (2004). Introduction to statistical learning theory. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pages 169–207.

Goldberg, P. and Jerrum, M. (1993). Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. In *Proceedings of the 6th annual conference on Computational Learning Theory*, pages 361–369.

Gühring, I., Raslan, M., and Kutyniok, G. (2020). Expressivity of deep neural networks. *arXiv preprint arXiv:2007.04759*.

Mammen, E. and Tsybakov, A. B. (1999). Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829.

Van der Vaart, A. W. and Wellner, J. A. (1996). *Weak convergence*. Springer.

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280.

A. Common notation

symbol	meaning
$\mathcal{A}$	ML algorithm
$\mathcal{D}_n = (Z_i)_{i=1}^n$	training data set
L	loss function
$\mathcal{H}$	hypothesis class
$R(h) = \mathbb{E}_Z[L(Z, h(Z))]$	risk/test error of hypothesis h,
$R_n(h) = \frac{1}{n}\sum_{i=1}^n L(Z_i, h(Z_i))]$	empirical risk/training error of hypothesis h,
h	arbitrary hypothesis
$h_0 = \arg\min_h R(h)$	best possible hypothesis among all functions
$h^* = \arg\min_{h\in\mathcal{H}} R(h)$	best possible hypothesis in $\mathcal{H}$
$\widehat{h} = \mathcal{A}(\mathcal{D}_n)$	hypothesis generated by ML algorithm
$R(\widehat{h}) - R_n(\widehat{h})$	generalization gap
$\mathcal{R}_n$	Rademacher complexity
$\widehat{\mathcal{R}}_n$	empirical Rademacher complexity
$N(\varepsilon, \mathcal{F}, \|\cdot\|)$	covering number
$\ln nN(\varepsilon, \mathcal{F}, \|\cdot\|)$	covering entropy

B. Mathematical preliminaries

B.1. Basic probability

Lemma B.1.1 (Law of iterated expectations/tower rule). *For any random variables X, Y,*

$$\mathbb{E}[\mathbb{E}[Y \mid X]] = \mathbb{E}[Y].$$

B.2. O-notation

Definition B.2.1. *We write*

(i) $a_n = O(b_n)$ *("a_n is big-O of b_n") if* $\limsup_{n\to\infty} |a_n/b_n| < \infty$.

(ii) $a_n = o(b_n)$ *("a_n is little-O of b_n") if* $\lim_{n\to\infty} |a_n/b_n| = 0$.

Interpretation:

- Case $b_n \to 0$:
  - $a_n = o(b_n)$ means that a_n goes faster to zero than b_n,
  - $a_n = O(b_n)$ means that a_n goes at least as fast to zero as b_n.
- Case $b_n \to \infty$:
  - $a_n = o(b_n)$ means that a_n goes slower to ∞ than b_n,
  - $a_n = O(b_n)$ means that a_n goes at most as fast to ∞ as b_n.
- Case $b_n = const.$:
  - $a_n = o(1)$ means that $a_n \to 0$,
  - $a_n = O(1)$ means that a_n is bounded.

B.3. Norms

Definition B.3.1 (Vector q-norm). *For $x \in \mathbb{R}^d$ and $q \in [1, \infty]$, the q-norm is defined as*

$$\|x\|_q = \left(\sum_{j=1}^{d} |x_j|^q \right)^{1/q}.$$

The special case $q = \infty$ is understood as

$$\|x\|_\infty = \max_{1 \le j \le d} |x_j|.$$

Definition B.3.2 ($L_q(P)$-norm)**.** *Let $q \in [1, \infty]$ and P be a measure. The $L_q(P)$-norm of a function $f \colon \mathcal{X} \to \mathbb{R}$ is defined as*

$$\|f\|_{L_2(P)} = \left(\mathbb{E}_P[|f(X)|^q]\right)^{1/q} = \left(\int |f(x)|^q dP(x)\right)^{1/q}.$$

Vector q-norms are a special case for P the counting measure.

B.4. Elementary inequalities

Lemma B.4.1 (Jensen's inequality)**.** *For any convex function ϕ and random variable X,*

$$\phi(\mathbb{E}[X]) \le \mathbb{E}[\phi(X)].$$

For example, $\mathbb{E}[X]^2 \le \mathbb{E}[X^2]$.

Lemma B.4.2 (Cauchy-Schwarz inequality)**.** *For vector space V with inner product $\langle \cdot, \cdot \rangle$ and norm $\|x\| = \sqrt{\langle x, x \rangle}$, it holds*

$$|\langle x, y \rangle| \le \|x\| \|y\| \quad \text{for all } x, y \in V.$$

Important special cases are:

- $|x^\top y| \le \|x\|_2 \|y\|_2$,
- $|\mathbb{E}[XY]| \le \sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}$.

Hölder's inequality is a generalization that applies to specific inner product spaces.

Lemma B.4.3 (Hölder's inequality)**.** *Let $q \in [1, \infty]$ and $p = q/(q-1)$. Then for any two random variables $X, Y \in \mathbb{R}$,*

$$\mathbb{E}[|XY|] \le \left(\mathbb{E}[|X|^p]\right)^{1/p} \left(\mathbb{E}[|Y|^q]\right)^{1/q},$$

and any vectors $x, y \in \mathbb{R}^d$,

$$|x^\top y| \le \|x\|_q \|y\|_p.$$