

Learning theory from first principles

Lecture 1: Introduction to supervised learning

Francis Bach

September 18, 2020

Class summary

- Information on the course
- Decision theory (loss, risk, optimal predictors)
- Decomposition of excess risk into approximation and estimation errors
- No free lunch theorems
- Basic notions of concentration inequalities (MacDiarmid, Hoeffding, Bernstein)

1 Information on the course

- The class will be organized in 9 three-hour sessions, each with a precise topic except the last one dedicated to recent learning theory results.
- Validation: one written in-class exam, and (very) simple coding assignments (to illustrate convergence results).
- Register online: <https://forms.gle/f4nXh5u6VR98dGJAA>
- Ask questions! (chat or directly)
- References: [1, 2, 3, 4, 5].
- Prerequisites: We will prove results in class so a good knowledge of undergraduate mathematics is important, as well as basic notions in probability. Having followed an introductory class on machine learning is beneficial. Good references for introduction to machine learning are [6, 7].



Each student is expected to read them before the class.



1.1 Goals of the course

- The goal of this class is to present old and recent results in learning theory, for the most widely-used learning architectures. This class is geared towards theory-oriented students as well as students who want to acquire a basic mathematical understanding of algorithms used throughout the masters program.
- A particular effort will be made to prove **many results from first principles**, while keeping the exposition as simple as possible. This will naturally lead to a choice of key results that show-case in simple but relevant instances the important concepts in learning theory. Some general results will also be presented without proofs.
- Arbitrary (and personal) choice of topics. Many forgotten ones (e.g., bandits, reinforcement learning, unsupervised learning, etc.). Suggestions of extra themes are welcome!

1.2 Syllabus

1. September 18: **Learning with infinite data (population setting)**
 - Decision theory (loss, risk, optimal predictors)
 - Decomposition of excess risk into approximation and estimation errors
 - No free lunch theorems
 - Basic notions of concentration inequalities (MacDiarmid, Hoeffding, Bernstein)
2. September 25: **Linear least-squares regression**
 - Guarantees in the fixed design settings (simple in closed form)
 - Guarantees in the random design settings
 - Ridge regression: dimension independent bounds
3. October 2: **Classical risk decomposition**
 - Approximation error
 - Convex surrogates
 - Estimation error through covering numbers (basic example of ellipsoids)
 - Modern tools (no proof): Rademacher complexity, Gaussian complexity (+ Slepian/Lipschitz)
 - Minimax rates (at least one proof)

⚠ No class on October 9

4. October 16: **Optimization for machine learning**
 - Gradient descent
 - Stochastic gradient descent
 - Generalization bounds through stochastic gradient descent
5. October 23: **Local averaging techniques**
 - Kernel density estimation
 - Nadaraya-Watson estimators (simplest proof to be found with apparent curse of dimensionality)
 - K-nearest-neighbors
 - Decision trees and associated methods

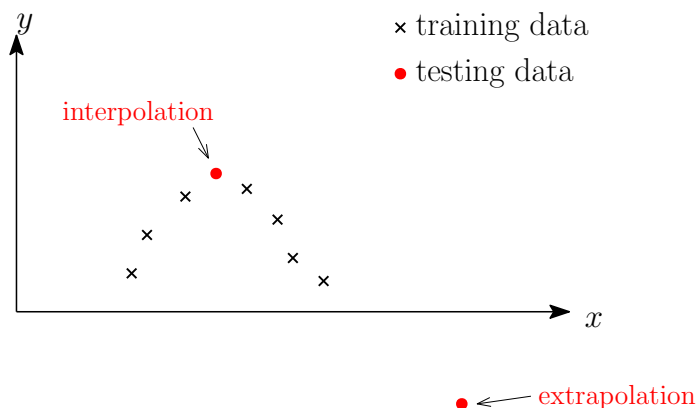
6. October 30: **Kernel methods**
 - Modern analysis of non-parametric techniques (simplest proof with results depending on s and d)
7. November 6: **Model selection**
 - L0 penalty with AIC
 - L1 penalty
 - High-dimensional estimation
8. November 13: **Neural networks**
 - Approximation properties (simplest approximation result)
 - Two layers
 - Deep networks
9. November 20: **Special topics**
 - Generalization/optimization properties of infinitely wide neural networks
 - Double descent

2 Supervised machine learning: introduction

- Main goal: given some observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, of inputs/outputs, features/labels, covariates/responses (training data), given a new $x \in \mathcal{X}$, predict $y \in \mathcal{Y}$ (testing data).
- Many examples where \mathcal{X} and \mathcal{Y} can be very diverse, from many areas of science and engineering:
 - \mathcal{X} : images, sounds, videos, text, proteins, web pages, social networks, sensors from industry, etc.
 - \mathcal{Y} : binary labels $\mathcal{Y} = \{0, 1\}$, real response $\mathcal{Y} = \mathbb{R}$, multiclass $\mathcal{Y} = \{1, \dots, k\}$, more generally structured outputs (e.g., graph prediction, visual scene analysis, source separation).

Why is it difficult?

- The label y is not a deterministic function of x : given $x \in \mathcal{X}$, the outputs are noisy, as $y = f(x) + \varepsilon$, e.g., with noise due to diverging views between labellers, dependence on external unobserved quantities (that is $y = f(x, z)$, z random).
- The prediction function f may be quite complex (highly non-linear).
- Only a few x 's are observed: need interpolation and potentially extrapolation (see below), and overfitting is always a possibility.



- Weak link between training and testing distributions.
- What is the criterion for performance?

Main formalization

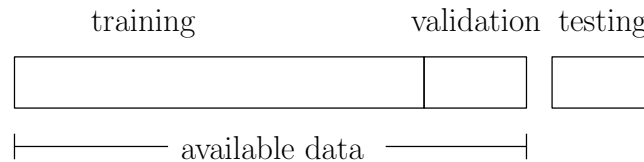
- See (x_i, y_i) as a realization of random variables (X_i, Y_i) , and the criterion is to minimize the expectation of some “performance” measure with respect to the distribution of the test data.
- Classical (never met in practice...) assumptions: the random variables (X_i, Y_i) are independent and identically distributed with the same distribution as the testing distribution. In this course, we will ignore the potential mismatch between train and test distributions (although this is an important research topic).

- A machine learning algorithm \mathcal{A} is then a function that goes from a dataset, i.e., an element of $(\mathcal{X} \times \mathcal{Y})^n$, to a function from \mathcal{X} to \mathcal{Y} . In other words, the output of a machine learning algorithm is an algorithm itself!

Practical performance evaluation

In practice, we not have access to the test distribution, but samples from it.

- In most cases, given the data given to the machine learning, it is split into three parts:
 - the training set, on which learning models will be estimated,
 - the validation set, to estimate hyperparameters (all learning techniques have some),
 - the testing set, to evaluate the performance of the final model (formally, the test set can only be used once!)



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




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

3 Decision theory


Main question: what is the optimal performance, regardless of the finiteness of the training data? In other words, if we have a perfect knowledge of the underlying probability distribution, what should be done?

- We consider a fixed (testing) distribution $dp_{X,Y}$ on $\mathcal{X} \times \mathcal{Y}$, with marginal distribution dp_X on \mathcal{X} .

 We ignore on purpose measurability issues.

 We will almost always use the overloaded notation dp , to denote $dp_{X,Y}$ and dp_X , where the context can always make the definition unambiguous. For example, when $f : \mathcal{X} \rightarrow \mathbb{R}$ and $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, we have $\mathbb{E}f(X) = \int_{\mathcal{X}} f(x)dp(x)$ and $\mathbb{E}g(X, Y) = \int_{\mathcal{X} \times \mathcal{Y}} g(x, y)dp(x, y)$.

Loss functions

- We consider a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ (often \mathbb{R}_+); $\ell(y, z)$ is the loss of predicting z while the true label is y .  Some authors swap y and z in the definition above.
- **Binary classification:** $\mathcal{Y} = \{0, 1\}$ (or often $\mathcal{Y} = \{-1, 1\}$, or, less often, when seen as a subcase of the loss below, $\mathcal{Y} = \{1, 2\}$), and $\ell(y, z) = 1_{y \neq z}$ (“0-1” loss), that is, 0 if y is equal to z (no mistake), and 1 otherwise (mistake).




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

- **Multi-category classification:** $\mathcal{Y} = \{1, \dots, k\}$, and $\ell(y, z) = 1_{y \neq z}$ (“0-1” loss).
- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y, z) = (y - z)^2$ (square loss). The absolute loss $\ell(y, z) = |y - z|$ is often used for robust estimation.
- **Many other potential losses!** E.g., Hamming loss for the multi-label problem $\mathcal{Y} = \{0, 1\}^k$ equal to $\ell(y, z) = \sum_{j=1}^k 1_{y_j \neq z_j}$, loss for ranking, etc. See, e.g., examples in [8] and references therein.
- In principle, the loss function is imposed by the final user, as this is the way models will be evaluated.

Risk

- The *risk*, *generalization performance*, or *testing error* of a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as:

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

 Be careful with the randomness or lack thereof of f : when performing learning from data, f will depend on the random training data and not on the testing data, and thus $\mathcal{R}(f)$ is typically

random because of the dependence on the training data. However, as a function on functions, \mathcal{R} is deterministic.

It depends on the distribution $dp = dp_{X,Y}$ on (X, Y) . We sometimes use the notation $\mathcal{R}_{dp}(f)$ to make it explicit.

Note that sometimes, we consider random predictions, that is for any x , we output a distribution on y , and then the risk is taken as the expectation over the randomness of the outputs.

- Averaging the loss on the training data defines the *empirical risk*, or *training error*:

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

$\widehat{\mathcal{R}}$ is a random function on functions.

- **Expression of the risk for classical losses:**

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

Main question: what is the best prediction function f ?

- Using conditional expectation and its associated tower law, we have

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

which we can rewrite

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

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

$$r(z|x) = \mathbb{E}(\ell(Y, z)|X = x),$$

which leads to

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

A minimizer of $\mathcal{R}(f)$ can be obtained by considering for any $x \in \mathcal{X}$, the function value $f(x)$ to be equal to a minimizer $z \in \mathcal{Y}$ of $r(z|x) = \mathbb{E}(\ell(Y, z)|X = x)$. We can therefore **consider all x as being treated independently**. This leads to the following propositions.

Bayes risk and Bayes predictor

- **Proposition 1 (Bayes predictor)** *The risk is minimized at a Bayes predictor $f^* : \mathcal{X} \rightarrow \mathcal{Y}$ satisfying for all $x \in \mathcal{X}$, $f^*(x) \in \arg \min_{z \in \mathcal{Y}} \mathbb{E}(\ell(Y, z)|X = x) = \arg \min_{z \in \mathcal{Y}} r(z|x)$. The Bayes risk \mathcal{R}^* is the risk of all Bayes predictors and is equal to*

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

Note that (a) the Bayes predictor is not always unique, but that all lead to the same Bayes risk (for example in binary classification when $\mathbb{P}(Y = 1|X = x) = 1/2$), and (b) that the Bayes risk is usually non zero (unless the dependence between x and y is deterministic).

Definition 1 (Excess risk) *The excess risk of a function from $f : \mathcal{X} \rightarrow \mathcal{Y}$ is equal to $\mathcal{R}(f) - \mathcal{R}^*$ (it is always non-negative).*

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

Main examples


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

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

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

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

This extends directly to multiple categories $\mathcal{Y} = \{1, \dots, k\}$, for $k \geq 2$, where we have $f^*(x) \in \arg \max_{i \in \{1, \dots, k\}} \mathbb{P}(Y = i|X = x)$. Exercise: write down the Bayes risk.

 These Bayes predictors and risks are only valid for the 0-1 loss. Less symmetric losses are very common in applications (e.g., for spam detection).

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

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

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

- **Exercise:** What is the Bayes predictor for regression with the absolute loss $\ell(y, z) = |y - z|$? Solution: $f^*(x)$ is a median of the distribution of Y given $X = x$.
- **Exercise:** We consider a random prediction rule where we predict from the probability distribution of Y given $X = x$. When is this achieving the Bayes risk? Solution: when the Bayes risk is zero.

4 Learning from data

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

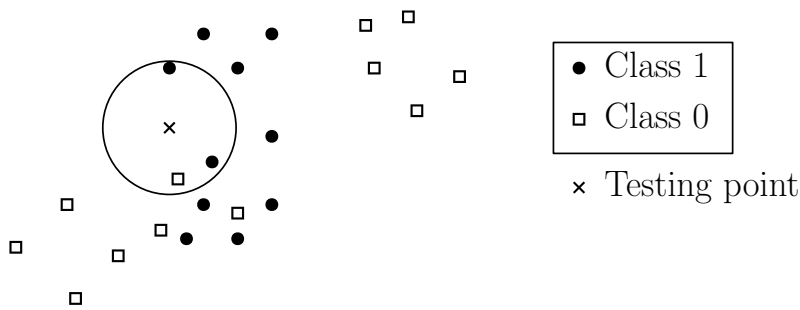
- (1) Local averaging (Lecture 5).
- (2) Empirical risk minimization (Lectures 2, 3, 4, 6, 7, 8).

4.1 Local averaging

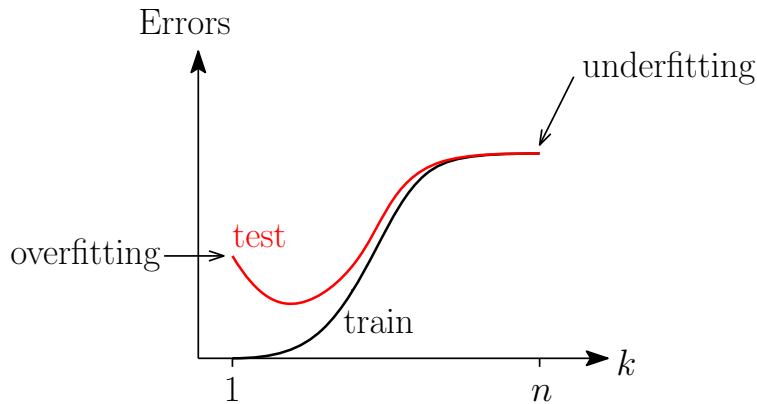
The goal here is to try to approximate / emulate the Bayes predictor, e.g., $f^*(x) = \mathbb{E}(Y|X = x)$ for least-squares regression, from empirical data. This is done often by explicit/implicit estimation of the conditional distribution by *local averaging* (k -nearest neighbors, which is used as the main example for this lecture, Nadaraya Watson, decision trees). See Lecture 5 for details.

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

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



- Pros: (a) no optimization or training, (b) often easy to implement, (c) can get very good performance in low dimensions (in particular for non-linear dependences between X and Y).
- Cons: (a) slow at query time: must pass through all training data at each testing point (there are algorithmic tools to reduce complexity, see Lecture 3), (b) bad for high-dimensional data (curse of dimensionality, more on this in Lecture 5), (c) the choice of local distance function is crucial, (d) the choice of “width” parameters (or k) has to be performed.
- Plot of training error and testing errors as a function of k for a typical problem. When k is too large, there is *underfitting* (the learned function is too close to a constant, which is too simple), while for k too small, there is *overfitting* (there is a strong discrepancy between the testing and training errors).



- **Exercise:** How would the curve move when n increases? Solution: Minimum test error goes down, optimal k is shifted to the right.

4.2 Empirical risk minimization

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

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

This defines the estimator $\hat{\theta} \in \arg \min_{\theta \in \Theta} \hat{\mathcal{R}}(f_\theta)$.

The most classical example is linear least-squares regression (studied at length in Lecture 2), where we minimize

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

where f is linear in some feature vector $\varphi(x) \in \mathbb{R}^d$ (no need for \mathcal{X} to be a vector space). The vector $\varphi(x)$ can be quite large (or even implicit, like in kernel methods, see Lecture 6). Other examples include neural networks (Lecture 8).

- Pros: (a) can be relatively easy to optimize (e.g., least-squares with simple derivation and numerical algebra, see Lecture 2), many algorithms available (mostly based on gradient descent, see Lecture 4), (b) can be applied in any dimension (if a reasonable feature vector is available).
- Cons: (a) can be relatively hard to optimize (e.g., neural networks), (b) need a good feature vector for linear methods, (c) dependence on parameters can be complex (e.g., neural networks), (d) need some capacity control to avoid overfitting, (e) how to parameterize functions with values in $\{0, 1\}$ (see Lecture 3 for the use of convex surrogates)?

Risk decomposition

The material in this section will be studied further in more details in Lecture 3

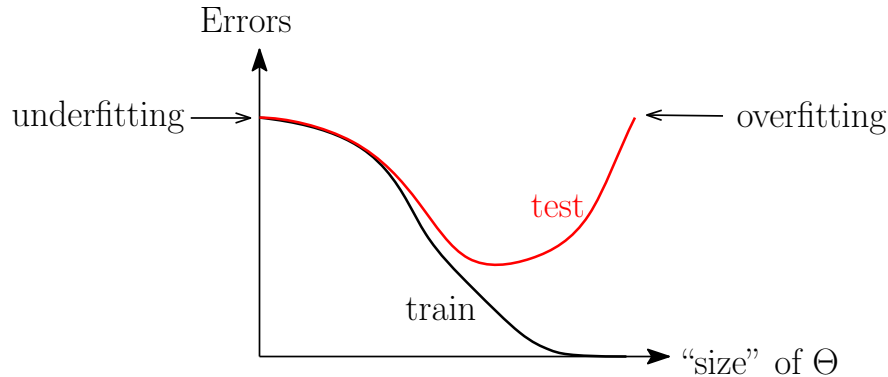
- Risk decomposition in estimation error + approximation error : given any $\hat{\theta} \in \Theta$,

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

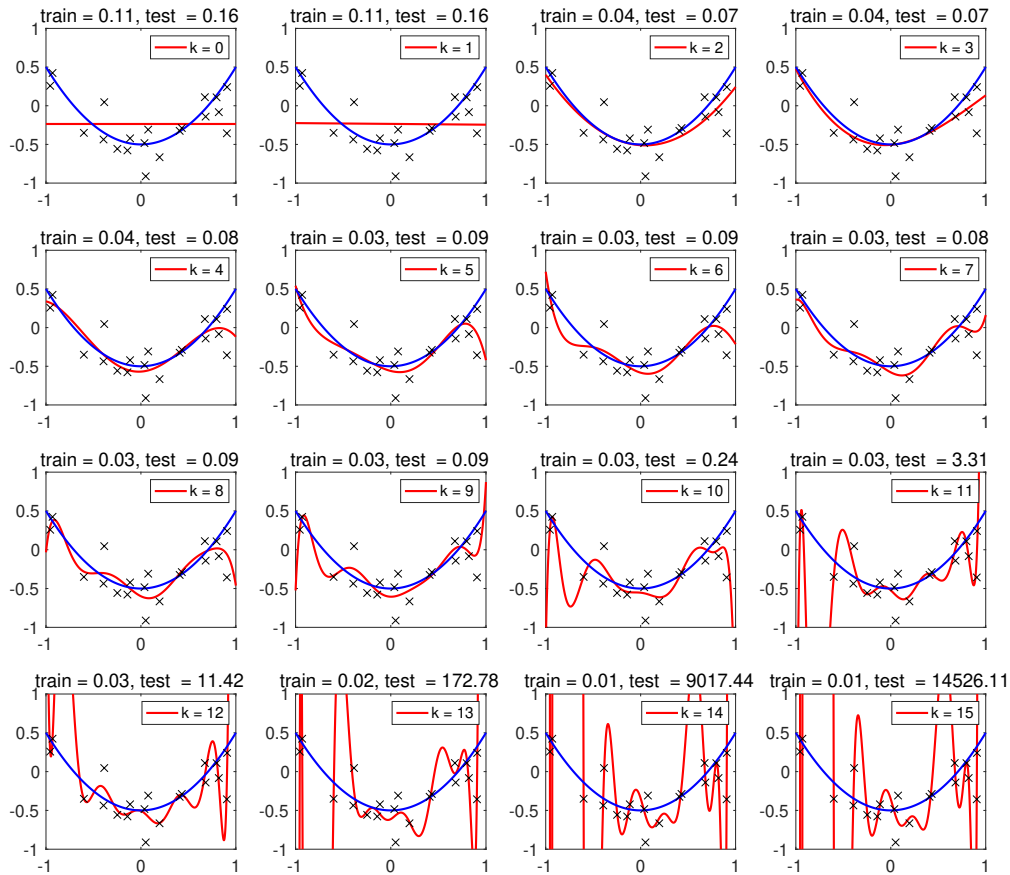
The approximation error does not depend on the chosen $f_{\hat{\theta}}$ and depends only on the class of functions parameterized by $\theta \in \Theta$. It is thus always deterministic function, that characterizes the modelling assumptions made by the models. Then Θ grows, the approximation error goes down, to zero if arbitrary functions can be approximated arbitrary well by the functions f_{θ} . It is also independent of n .

The estimation error is typically random, because $f_{\hat{\theta}}$ is random. It is typically decreasing in n , and usually goes up when Θ grows.

The typical error curves look like this:



- **Examples of approximations by polynomials in one-dimensional regression:** we consider $(X, Y) \in \mathbb{R} \times \mathbb{R}$, with prediction functions which are polynomials of order k , from $k = 0$ (constant functions) to $k = 15$. For each k , the model has $k + 1$ parameters. The training error (using square loss) is minimized with $n = 20$ observations. The data were generated as a quadratic function plus some independent additive noise. The training error is monotonically decreasing in k , while the testing error goes down and then up.



- Typically, we will see in later lectures that the estimation error is often decomposed as

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

The uniform deviation grows with the “size” of Θ , and usually decays with n . See more details in Lecture 3.

Capacity control

- “Capacity control” (making sure that the set of allowed functions is not too large) by typically reducing number of parameters, or by restricting the norm of predictors: this leads to constrained optimization.

- Capacity control can also be done by regularization: minimize

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

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

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

This is often easier for optimization, but harder to analyze (see Lectures 3 and 4)



Difference between parameters (e.g., θ) learned on the training data and hyperparameters (e.g., λ) learned on the validation data.

4.3 Statistical learning theory

- The goal is to provide some guarantees of performance on unseen data. A common assumption is that the data $\mathcal{D}_n(dp) = \{(x_1, y_1), \dots, (x_n, y_n)\}$ is obtained as independent and identically distributed (i.i.d.) observations, from some unknown distribution dp .
- An algorithm \mathcal{A} is a mapping from $\mathcal{D}_n(dp)$ (for any n) to a function from \mathcal{X} to \mathcal{Y} . The risk depends on the probability distribution $dp \in \mathcal{P}$, as $\mathcal{R}_{dp}(f)$. The goal is to find \mathcal{A} such that the risk

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

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

Measures of performance

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

- *Expected error*: we measure performance as

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

where the expectation is with respect to the training data.

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


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

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

The crux is to find ε which is as small as possible (typically as a function of δ). The notion of PAC consistency corresponds, for any $\varepsilon > 0$ to have such an inequality for each n , and a sequence δ_n that tends to zero.

Notions of consistency

- An algorithm is said *universally consistent* (in expectation) if for all distributions $dp = dp_{X,Y}$ on (X, Y) the algorithm \mathcal{A} is consistent in expectation for the distribution dp .

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

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

We thus aim at finding an algorithm \mathcal{A} such that

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

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

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

This is typically a function of the sample size n and of properties of \mathcal{X} , \mathcal{Y} and the allowed set of problems \mathcal{P} (e.g., dimension of \mathcal{X} , number of parameters).

- One given algorithm with a convergence proof provides an upper-bound.
- Lower-bounding the optimal performance: in some set-ups, it is possible to show that the infimum over all algorithms is greater than a certain quantity. Machine learners are happy when upper-bounds and lower-bounds match (up to constant factors).
- The analysis can be “non-asymptotic”, with an upper-bound with explicit dependence on all quantities; the bound is then valid for all n , even if sometimes vacuous (e.g., a bound greater than 1 for a loss uniformly bounded by 1).

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



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

5 No free lunch theorems (◆)

“Learning is not possible without assumptions.” See [2, Chapter 7] for details.

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

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

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

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

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

Denoting $\hat{f}_{\mathcal{D}_n} = \mathcal{A}(\mathcal{D}_n(dp))$ the classifier, and $S(r) = \mathbb{E} \left[\mathcal{R}_{dp}(\hat{f}_{\mathcal{D}_n}) \right]$ the expected risk, we want to maximize $S(r)$ with respect to $r \in \{0, 1\}^k$; the maximum is greater than the expectation of $S(r)$ for any distribution dq on r , in particular the uniform distribution (each r_j an independent unbiased Bernoulli variable). Then

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

because X is almost surely in $\{1, \dots, k\}$ and $Y = r_X$ almost surely. Note that we take expectations with respect to X_1, \dots, X_n, X , and r (all being independent from each other).

Then, we get:

$$\begin{aligned} \mathbb{E}_{r \sim dq(r)} S(r) &= \mathbb{E} \left[\mathbb{P}(\hat{f}_{\mathcal{D}_n}(X) \neq r_X | X_1, \dots, X_n, r_{X_1}, \dots, r_{X_n}) \right] \text{ by the law of total expectation,} \\ &\geq \mathbb{E} \left[\mathbb{P}(\hat{f}_{\mathcal{D}_n}(X) \neq r_X \ \& \ X \notin \{X_1, \dots, X_n\} | X_1, \dots, X_n, r_{X_1}, \dots, r_{X_n}) \right] \\ &\hspace{15em} \text{by monotonicity of probabilities,} \\ &= \mathbb{E} \left[\frac{1}{2} \mathbb{P}(X \notin \{X_1, \dots, X_n\} | X_1, \dots, X_n, r_{X_1}, \dots, r_{X_n}) \right], \end{aligned}$$

because $\mathbb{P}(\hat{f}_{\mathcal{D}_n}(X) \neq r_X | X \notin \{X_1, \dots, X_n\}, X_1, \dots, X_n, r_{X_1}, \dots, r_{X_n}) = 1/2$ (the label $Y = r_X$ has the same probability of being 0 or 1, given that it was not observed). Thus,

$$\mathbb{E}_{r \sim dq(r)} S(r) \geq \frac{1}{2} \mathbb{P}(X \notin \{X_1, \dots, X_n\}) = \frac{1}{2} \mathbb{E} \left[\prod_{i=1}^n \mathbb{P}(X_i \neq X | X) \right] = \frac{1}{2} (1 - 1/k)^n.$$

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

A caveat is that the hard distribution may depend on n (and, from the proof, it takes k values, with k tending to infinity). The following theorem is given without proof and is much “stronger” [2, Theorem 7.2], as it more convincingly shows that learning can be arbitrarily slow without assumption (note that the earlier one is not a corollary of the later one).

Theorem 2 (no free lunch - sequence of errors) Consider the binary classification with 0 – 1 loss, with \mathcal{X} infinite. Let \mathcal{P} denote the set of all probability distributions on $\mathcal{X} \times \{0, 1\}$. For any decreasing sequence a_n tending to zero and such that $a_1 \leq 1/16$, for any learning algorithm \mathcal{A} , there exists $dp \in \mathcal{P}$, such that for all $n \geq 1$:

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

6 Concentration inequalities

- Union bound: given events indexed by $f \in \mathcal{F}$ (which can have an infinite number of elements), we have:

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

- Supremum of random variables (proof by direct application of the union bound):

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

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

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




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

- The equality above can be interpreted as

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

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

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

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



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

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

6.1 Hoeffding's inequality

- **Hoeffding's inequality:** if Z_1, \dots, Z_n are independent random variables such that $Z_i \in [0, 1]$ almost surely, then, for any $t \geq 0$,

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

– Corollary (by just applying to Z_i 's and $1 - Z_i$'s and using the union bound):

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

Note the difference with the central limit theorem, which is more precise (as it involves the variance of Z_i 's, but is asymptotic). Bernstein inequalities (see below) are in between.

- Proof of Hoeffding inequality:

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

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

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

We have that $\varphi'(0) = 0$ and $\varphi''(s)$ is the variance of some random variable $\tilde{Z} \in [0, 1]$, with distribution proportional to $e^{s(z - \mathbb{E}[Z])} d\mu(z)$ where $d\mu(z)$ is the distribution of Z . We can thus bound the variance of \tilde{Z} as

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

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

(b) By Markov inequality, for any $t \geq 0$, and denoting $\bar{Z} = \frac{1}{n} \sum_{i=1}^n Z_i$:

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

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

- **Exercise:** Write down Hoeffding's inequality for $Z_i \in [a, b]$ almost surely.

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

- Such an inequality is often used “in the other direction”: For any $\delta \in (0, 1)$, with probability greater than $1 - \delta$, we have:

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

Note the dependence in n as $1/\sqrt{n}$ and the logarithmic dependence in δ .

- When $Z_i \in [a_i, b_i]$ almost surely, with potentially different a_i 's and b_i 's, the probability upper-bound can be replaced by $2 \exp(-2nt^2/c^2)$, where $c^2 = \frac{1}{n} \sum_{i=1}^n (b_i - a_i)^2$.
- Note: the result extends to martingales with essentially the same proof, leading to Azuma's inequality. See https://en.wikipedia.org/wiki/Azuma's_inequality.

6.2 Expectation of the maximum

- Overall message: Taking the maximum of n bounded random variable leads to an extra factor of $\sqrt{\log n}$. Note here that we do not impose independence.

⚠ This logarithmic factor appears many times in this class.

- Expectation of the maximum: if Z_1, \dots, Z_n are (potentially dependent) random variables such that $Z_i \in [0, 1]$ almost surely, then

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

Proof:

$$\begin{aligned} \mathbb{E}[\max\{Z_1 - \mathbb{E}[Z_1], \dots, Z_n - \mathbb{E}[Z_n]\}] &\leq \frac{1}{t} \log \mathbb{E}[e^{t \max\{Z_1 - \mathbb{E}[Z_1], \dots, Z_n - \mathbb{E}[Z_n]\}}] \text{ by Jensen's inequality,} \\ &= \frac{1}{t} \log \mathbb{E}[\max\{e^{tZ_1 - \mathbb{E}[Z_1]}, \dots, e^{tZ_n - \mathbb{E}[Z_n]}\}] \\ &\leq \frac{1}{t} \log \mathbb{E}[e^{tZ_1 - \mathbb{E}[Z_1]} + \dots + e^{tZ_n - \mathbb{E}[Z_n]}] \\ &\leq \frac{1}{t} \log(ne^{t^2/8}) = \frac{\log n}{t} + \frac{t}{8} = \frac{\sqrt{2 \log n}}{2} \text{ with } t = 2\sqrt{2 \log n}, \end{aligned}$$

using the step (a) in Hoeffding's inequality proof.

6.3 MacDiarmid's inequality

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

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

Then

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

- Proof (which generalizes Hoeffding's inequality, which corresponds to $f(z) = \frac{1}{n} \sum_{i=1}^n z_i$) of the one-sided inequality, $\mathbb{P}\left(f(Z_1, \dots, Z_n) - \mathbb{E}f(Z_1, \dots, Z_n) \geq t\right) \leq \exp(-2t^2/(nc^2))$, which is sufficient to get the two-sided one: we simply introduce the random variables

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

We have $\mathbb{E}(V_i | Z_1, \dots, Z_{i-1}) = 0$, $|V_i| \leq c$ almost surely, and $f(Z_1, \dots, Z_n) - \mathbb{E}f(Z_1, \dots, Z_n) = \sum_{i=1}^n V_i$. Using the same argument as in part (a) of Hoeffding's inequality proof, we can show $\mathbb{E}(e^{sV_i} | Z_1, \dots, Z_{i-1}) \leq e^{s^2 c^2 / 8}$, and we can obtain a proof with the same steps as part (b) of Hoeffding's inequality by being careful with conditioning. See [9] for details.

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

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

Hint: take $f(z_1, \dots, z_n) = \left\| \frac{1}{n} \sum_{i=1}^n z_i \right\|_2$.

6.4 Bernstein's inequality (♦)

- We consider n independent random variables Z_1, \dots, Z_n such that $|X_i| \leq c$ almost surely and $\mathbb{E}(Z_i) = \mu$. Then

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

where $\sigma^2 = \frac{1}{n} \sum_{i=1}^n \text{var}(Z_i)$. A useful corollary in the "reverse direction": with probability greater than $1 - \delta$, we have:

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

Note here that we get the same dependence than for the central limit theorem for small deviations t (and a strict improvement on Hoeffding because the variance is essentially bounded by the squared diameter of the support), while for t large, the dependence in t is worse than Hoeffding's inequality.

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

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

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

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

(b) Using Markov's inequality, with $\sigma_i^2 = \text{var}(Z_i)$, we have:

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

By choosing $s = \frac{1}{c} \log(1 + tc/\sigma^2)$, we get a bound equal to $\exp\left(-\frac{n\sigma^2}{c^2} h(ct/\sigma^2)\right)$, with $h(\alpha) = (1 + \alpha) \log(1 + \alpha) - \alpha \geq \frac{\alpha^2}{2+2\alpha/3}$. See [9] for details.

Acknowledgements

These class notes have been adapted from the notes of many colleagues I have the pleasure to work with, in particular Lénaïc Chizat, Pierre Gaillard, Alessandro Rudi and Simon Lacoste-Julien.

References

- [1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.
- [2] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic Theory of Pattern Recognition*, volume 31. Springer Science & Business Media, 1996.
- [3] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
- [4] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.
- [5] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.
- [6] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2020.
- [7] Chloé-Agathe Azencott. *Introduction au Machine Learning*. Dunod, 2019.
- [8] Alex Nowak, Francis Bach, and Alessandro Rudi. Sharp analysis of learning with discrete losses. In *International Conference on Artificial Intelligence and Statistics*, pages 1920–1929, 2019.
- [9] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.