Algorithmic Foundations of Learning **Lecture 1**

Introduction

Lecturer: Patrick Rebeschini *Version: February 9, 2022*

1.1 Introduction

This course is about the algorithmic paradigms that lay the foundations of machine learning and the theory that is needed for their design and analysis, with particular emphasis on the non-asymptotic methods for the study of random structures in high-dimensional probability, statistics, and optimization.

This set of lecture notes is meant to offer a streamlined presentation of the material. Students are expected to find and read the relevant chapters from the reading materials in [7, 6, 5, 3, 4, 10], which are all available online, and to independently fill in the gaps intentionally left in these notes, particularly when it comes to background knowledge.

As the statistics department in Oxford offers many courses in machine learning with an applied focus, this course is intentionally only covering theoretical aspects. In particular, no simulations will be discussed.

This course investigates the following three main learning paradigms.

1. Offline Statistical Learning: Prediction.
2. Offline Statistical Learning: Estimation.
3. Online Statistical Learning.

Today we describe the first learning framework in depth, only quickly mentioning the other two.

Remark 1.1 (Notation) *Throughout, we use UPPERCASE letters to denote random variables and lowercase letters to denote deterministic variables. We use cursive letters to denote sets. For a given set $\mathcal{T}$, we denote by $|\mathcal{T}|$ its cardinality. For a positive integer n, we use the notation $[n] = \{1, \dots, n\}$. For a vector $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, $p \geq 1$ we let $\|x\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$ denote the ℓ_p norm and $\|x\|_\infty = \max_{i \in [d]} |x_i|$ denote the ℓ_∞ norm. We denote by $x^\top$ the transpose of x. When dealing with linear algebra computations (i.e. matrix-vector multiplications), we typically interpret $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ as a* column *vector, instead of writing $x = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$. This will be clear from the context.*

1.2 Offline Statistical Learning: Prediction

The standard statistical learning framework for prediction is defined as follows [8].

Algorithm 1: Statistical Learning: Prediction

1. Observe a sample of *training* examples $S = \{Z_1, \dots, Z_n\} \in \mathcal{Z}^n$, assumed to be i.i.d. from an *unknown* probability distribution supported on a space $\mathcal{Z}$.
2. As a function of the random sample S (and possibly some external source of randomness), make decision (or choose action) $A \in \mathcal{A} \subseteq \mathcal{B}$, where $\mathcal{A}$ is a chosen set of admissible actions, subset of a larger set of actions $\mathcal{B}$.
3. Let $\ell : \mathcal{B} \times \mathcal{Z} \to \mathbb{R}_+$ be a given *prediction* loss function. Let $r : \mathcal{B} \to \mathbb{R}_+$ be the *expected or population risk*, defined as the average loss function

$$\boxed{r(a) := \mathbf{E}\, \ell(a, Z)}$$

where $Z \in \mathcal{Z}$ is a new, independent, *test* data point from the same unknown data distribution. Define the *excess risk* as follows

$$\boxed{\underbrace{r(A) - \inf_{a \in \mathcal{B}} r(a)}_{\text{excess risk}} = \underbrace{r(A) - \inf_{a \in \mathcal{A}} r(a)}_{\text{estimation error}} + \underbrace{\inf_{a \in \mathcal{A}} r(a) - \inf_{a \in \mathcal{B}} r(a)}_{\text{approximation error}}}$$

Goal: Minimize and control the estimation error as a function of the sample size n and notions of "complexity" of the action set $\mathcal{A}$ of the loss function ℓ.

Remark 1.2 (On the definition of excess risk) *The definition of expected risk holds for any* deterministic *action a:*

$$r(a) := \mathbf{E}\, \ell(a, Z).$$

If we consider a random *action A, the notation $r(A)$ refers to the random variable that is obtained by evaluating the deterministic function r at the random action A. Hence, we have*

$$r(A) \neq \mathbf{E}\, \ell(A, Z),$$

as the left-hand side is random, while the right-hand side is deterministic (being the expected value of random quantities). The correct expression are:

$$\begin{aligned} r(A) &= \mathbf{E}[\ell(A, Z)|A], \\ \mathbf{E}\, r(A) &= \mathbf{E}\, \ell(A, Z), \end{aligned}$$

where conditioning on A means conditioning on the randomness on which A depends, i.e. the training dataset S and possibly other "external" sources of randomness in the case of randomized algorithms.

Throughout the course we will assume that the minima are attained, and we denote them by

$$\begin{aligned} a^\star &\in \operatorname*{argmin}_{a \in \mathcal{A}} r(a), \\ a^{\star\star} &\in \operatorname*{argmin}_{a \in \mathcal{B}} r(a), \end{aligned}$$

(note that there could be multiple minimizers). The previous error decomposition can be written as

$$\underbrace{r(A) - r(a^{\star\star})}_{\text{excess risk}} = \underbrace{r(A) - r(a^{\star})}_{\text{estimation error}} + \underbrace{r(a^{\star}) - r(a^{\star\star})}_{\text{approximation error}} .$$

The quantity $r(a^{\star\star})$ is typically called the *irreducible* risk, or irreducible error.

The estimation error measures how much extra (average) loss the decision maker suffers by choosing action A compared to an optimal decision in the admissible action set $\mathcal{A}$. The estimation error is controlled by the number of training examples n, and by notions of "complexity" of the set of actions $\mathcal{A}$ and of the loss function ℓ. The approximation error measures how closely actions in $\mathcal{A}$ can approximate actions in the larger set $\mathcal{B}$. Larger sets of admissible actions lead to smaller approximation error but higher estimation error. This gives rise to the *approximation-estimation tradeoff*, also known as *bias-complexity tradeoff*.

Remark 1.3 (On randomness) *As the data distribution is assumed to be unknown in this framework, the risk function r can not be computed, and so also the excess risk and estimation error are uncomputable. Nevertheless, the estimation error is used as a way to assess how well the chosen action performs, and the goal of statistical learning is to control the estimation error by establishing upper bounds to it. The estimation error is a random variable, as the learning rule A depends on the random sample S (and possibly on other sources of "external" randomness). Controlling the estimation error means bounding its expected value, or showing that the estimation error is small with a certain probability (possibly high probability), i.e., with probability at least $1 - \delta$ for a certain value of $\delta \in [0, 1]$. As we will see, upper bounds on the estimation error typically depend on notions of complexity of the set $\mathcal{A}$ and the loss function ℓ, and they can be data-dependent (i.e., depend on data distribution) or data-independent. On the other hand, the approximation error is deterministic, as it simply deals with quantifying the error introduced by considering the family $\mathcal{A}$ instead of the family $\mathcal{B}$. As a result, statistical learning theory typically focuses on controlling the estimation error, which is where probability plays a role. Controlling the approximation error and determining a suitable class $\mathcal{A}$ is purely a problem of functional and numerical analysis.*

Remark 1.4 (On regularization and No Free Lunch theorems) *The restriction of the space of admissible actions (a.k.a. functions or hypotheses) to a subset $\mathcal{A}$ of the original set $\mathcal{B}$ has to do with using prior information on the learning system (as encoded by the choice of the subset $\mathcal{A}$) over the original class $\mathcal{B}$ (that represents lack of prior knowledge). This restriction (or other type of restrictions) is needed for the following two main reasons:*

- *The so-called No Free Lunch theorem, which investigates what can happen to a* given *learning algorithm when* multiple *distributions are considered. There are many versions available of the main principle. One common version says that for every learning algorithm, or learner, (i.e. any prescription to choose an action $A \in \mathcal{B}$ as a function of the data S), there exists a task (i.e. a data distribution $\mathbf{P}$) on which it fails, even though that task can be successfully learned by another learner. Using prior knowledge allows us to restrict to $\mathcal{A} \subseteq \mathcal{B}$ so to avoid distributions that will cause our learning task to fail. This is related to the concept of PAC learnability in binary classification, which is a property that holds* uniformly *over the choice of the data distribution. We refer to Chapter 5 of the book [6] for a discussion on these lines.*
- *The general need to impose some form of regularization to avoid overfitting and solve the stochastic problem of interest starting from a dataset S that only contains a* finite *amount of samples. Throughout this course, in various specific examples, we will discuss various ways of imposing regularization and why it is needed.*

In this course we will be interested in supervised learning, where the observed examples correspond to pairs of points, i.e., $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$. The point X_i is called *feature* or *covariate*, and the point Y_i is

the corresponding *label*. The set of admissible decisions is a subset of the set functions from $\mathcal{X}$ to $\mathcal{Y}$, i.e., $\mathcal{A} \subseteq \mathcal{B} := \{a : \mathcal{X} \to \mathcal{Y}\}$, and the loss function is of the form $\ell(a,(x,y)) = \phi(a(x),y)$, for a function $\phi : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. In this setting, the goal is to use the training sample S to choose a *predictor* or *hypothesis* $a \in \mathcal{A}$ that can be used to predict the label Y of a new test example (X,Y). In this case, the optimal decision in $\mathcal{B}$ given by $a^{\star\star} = \operatorname{argmin}_{a\in\mathcal{B}} \mathbf{E}\,\phi(a(X),Y)$ is called the *Bayes decision rule*, and its corresponding value of risk $r(a^{\star\star}) = \mathbf{E}\,\phi(a^{\star\star}(X),Y)$ is called the *Bayes risk*. As the following lemma shows, the Bayes decision rule for any loss function ϕ is the solution of a deterministic minimization problem, which depends on the conditional distribution $\mathbf{P}(Y \in \cdot\,|X)$ (via integrals with respect to this distribution, i.e., via the conditional expectation). Recall that this distribution is not computable in our setting. In this setting, statistical learning aims at finding actions that are as close as possible to the uncomputable Bayes decision rule. It is instructive to keep in mind what the Bayes decision rules look like for the applications that we will consider.

Lemma 1.5 (Bayes decision rule) *We have*

$$a^{\star\star}(x) = \operatorname*{argmin}_{\hat{y}\in\mathcal{Y}} \mathbf{E}[\phi(\hat{y},Y)|X=x] = \operatorname*{argmin}_{\hat{y}\in\mathcal{Y}} \int \phi(\hat{y},y)\,\mathbf{P}(Y \in \mathrm{d}y|X=x).$$

Proof: Let $a^{\star\star}$ given as in the statement of the lemma. We will prove that $r(a^{\star\star}) \le r(a)$ for any $a \in \mathcal{B}$. By construction, for any $\hat{y} \in \mathcal{Y}$ we have

$$\mathbf{E}[\phi(a^{\star\star}(x),Y)|X=x] \le \mathbf{E}[\phi(\hat{y},Y)|X=x]$$

so that, for any $a \in \mathcal{B}$,

$$\mathbf{E}[\phi(a^{\star\star}(x),Y)|X=x] \le \mathbf{E}[\phi(a(x),Y)|X=x].$$

By the tower property of conditional expectations, we have

$$r(a^{\star\star}) = \mathbf{E}\,\phi(a^{\star\star}(X),Y) = \mathbf{E}\,\mathbf{E}[\phi(a^{\star\star}(X),Y)|X] \le \mathbf{E}\,\mathbf{E}[\phi(a(X),Y)|X] = \mathbf{E}\,\phi(a(X),Y) = r(a).$$

■

Example 1.6 (Regression) *The regression setting is represented by the choice $\mathcal{X} = \mathbb{R}^d$ for a given dimension d, $\mathcal{Y} = \mathbb{R}$, and $\mathcal{B}$ is the set of functions from $\mathcal{X}$ to $\mathcal{Y}$.*

Typical choices of loss functions ϕ are as follows.

- *ℓ_2* ***loss function.*** *$\phi(\hat{y},y) = (\hat{y}-y)^2$. The Bayes decision rule is the conditional mean, $a^{\star\star}(x) = \mathbf{E}[Y|X=x]$. See* **Problem 1.2** *in the Problem Sheets.*
- *ℓ_1* ***loss function.*** *$\phi(\hat{y},y) = |\hat{y}-y|$. The Bayes decision rule is the conditional median, $a^{\star\star}(x) =$* ***Median****$[Y|X=x]$. (note that the conditional median need not be unique)*

Typical choices of admissible set $\mathcal{A}$ yield the following examples.

- ***Linear predictors.*** *The set $\mathcal{A}$ of admissible actions is made by affine functions on $\mathbb{R}^d$, i.e., $a(x) = w^\top x + b$ for parameters $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$, possibly with some restrictions on the values the parameters can take, such as $\|w\|_2 \le c$ or $\|w\|_1 \le c$ for a positive constant $c > 0$.*
- ***Neural networks.*** *(to be defined later on in the course)*
- ***Kernel methods.*** *(to be defined later on in the course)*

Remark 1.7 (Regression with the square loss) *The regression setting with the square loss is a popular framework for a variety of reasons, primarily for its simplicity as this setting often yields close form solutions and expression that are easy to interpret. In this case, the approximation-estimation decomposition takes a specific form, which is:*

$$\underbrace{r(A)-r(a^{\star\star})}_{\text{excess risk}} = \underbrace{r(A)-r(a^{\star})}_{\text{estimation error}} + \underbrace{r(a^{\star})-r(a^{\star\star})}_{\text{approximation error}} = \underbrace{\mathbf{E}[(A(X)-a^{\star}(X))^2|A]}_{\text{estimation error}} + \underbrace{\mathbf{E}[(a^{\star}(X)-a^{\star\star}(X))^2])}_{\text{approximation error}}.$$

In this setting, another popular decomposition for the expected *excess risk holds, the bias-variance decomposition:*

$$\underbrace{\mathbf{E}\, r(A)-r(a^{\star\star})}_{\text{expected excess risk}} = \underbrace{\mathbf{E}\Big[\Big(\mathbf{E}[A(X)|X]-a^{\star\star}(X)\Big)^2\Big]}_{\text{expected squared bias}} + \underbrace{\mathbf{E}\,\mathbf{Var}[A(X)|X]}_{\text{expected variance}}.$$

See **Problem 1.4** *in the Problem Sheets. This decomposition gives rise to another tradeoff: to achieve small expected excess risk, we need to find a predictor $A \in \mathcal{A}$ that simultaneously minimizes the expected square bias and the expected variance. Note that in this decomposition, the randomness that is considered in the conditional expectation and conditional variance (conditioned on a certain test feature X) is the randomness in A, i.e. the randomness in the training data and, possibly, in other "external" sources. Given a test data point X, the squared bias $(\mathbf{E}[A(X)|X]-a^{\star\star}(X))^2$ is a measure of the error on X that is due by considering model $A \in \mathcal{A}$ instead of the best model $a^{\star\star} \in \mathcal{B}$. On the other hand, the variance $\mathbf{Var}[A(X)|X]$ is a measure of the amount by which $A(X)$ changes if we estimate A using different training data. In general, the "larger" the class $\mathcal{A}$ is (i.e. the more "flexible" the methods we consider are), the higher the variance is and the smaller the bias is.*
If, additionally, one considers the case of linear predictors, the expressions derive above admit easy close forms in terms of linear algebra quantities.

Example 1.8 (Classification) *The classification setting is a particular case of regression, and it is represented by the choice $\mathcal{X} = \mathbb{R}^d$ for a given dimension d, $\mathcal{Y} = \{y_1, \dots, y_k\}$ for a given k, and $\mathcal{B}$ is the set of functions from $\mathcal{X}$ to $\mathcal{Y}$. In this setting, the typical loss function is given by the zero-one loss.*

- ***Zero-one loss function (a.k.a. true loss).*** *$\phi(\hat{y}, y) = \mathbf{1}_{\hat{y} \neq y}$, namely,*

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

  By Lemma 1.5, the Bayes decision rule reads $a^{\star\star}(x) = \operatorname{argmax}_{\hat{y} \in \mathcal{Y}} \mathbf{P}(Y = \hat{y}|X = x)$, also called maximum a posteriori (MAP) estimate of Y given X. See **Problem 1.2** *in the Problem Sheets.*

With the zero-one loss function we have $r(a) = \mathbf{E}\,\phi(a(X), Y) = \mathbf{P}(a(X) \neq Y)$. In this course we will be interested in binary classification where $k = 2$. The binary case is simpler, and encompasses most of the key ideas that are needed to tackle the general case $k > 2$. For concreteness, we will consider the setting $\mathcal{Y} = \{-1, 1\}$. In this case, the admissible action set $\mathcal{A}$ is typically taken to be the sign of the predictors used in regression, such as $a(x) = \operatorname{sign}(w^\top x + b)$. It is also common to consider convex relaxation of the zero-one loss. If $\mathcal{Y} = \{-1, 1\}$, the loss functions are chosen of the form $\phi(\hat{y}, y) = \varphi(\hat{y}y)$ for a function $\varphi : \mathbb{R} \to \mathbb{R}_+$. The zero-one loss function can be written as $\mathbf{1}_{\hat{y}y \leq 0}$, so $\varphi(u) = \mathbf{1}_{u \leq 0}$. Convex losses that uniformly bound from above the zero-one loss, namely, $\mathbf{1}_{u \leq 0} \leq \varphi(u)$ for all $u \in \mathbb{R}$, are given below.

- ***Exponential loss.*** *$\varphi(u) = e^{-u}$.*
- ***Hinge loss.*** *$\varphi(u) = \max\{1 - u, 0\}$.*
- ***Logistic loss.*** *$\varphi(u) = \log_2(1 + e^{-u})$.*

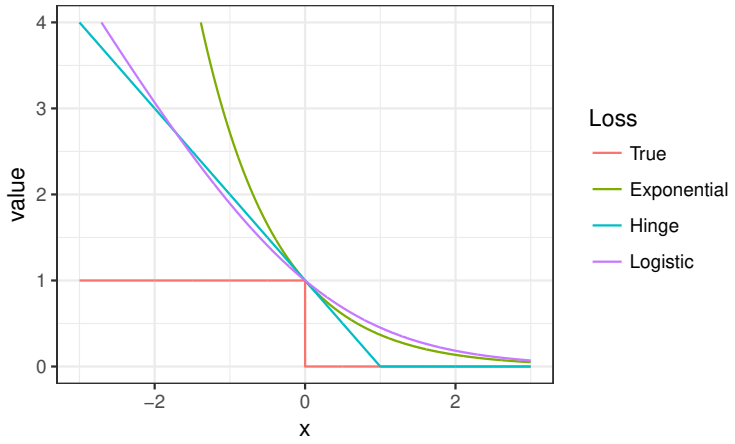

Figure 1.1: Convex loss surrogates.

1.2.1 Statistics

In this course we will discuss some of the main algorithmic paradigms that are used to construct an action $A \in \mathcal{A}$ as a function the training sample $S = \{Z_1, \dots, Z_n\}$ with the goal to minimize the estimation error $r(A) - r(a^\star)$. An important class of algorithms that we will consider is based on the empirical risk minimization (ERM) framework. ERM uses the empirical risk function $R : \mathcal{B} \to \mathbb{R}_+$ defined for any $a \in \mathcal{B}$ as

$$\boxed{R(a) := \frac{1}{n} \sum_{i=1}^{n} \ell(a, Z_i)}$$

as a computable proxy for the uncomputable expected/population risk function r. Notice that R is a random function, as it depends on the training sample S. On the other hand, r is a deterministic function, as it is defined as an expectation. By the Law of Large Numbers we know that for any *fixed* $a \in \mathcal{B}$, as a function of the number of data points n, the sequence of random variables $(R(a))_{n \geq 0}$ converges almost surely to $r(a)$. Hence, even for any finite n it seems reasonable to consider the ERM problem $\inf_{a \in \mathcal{A}} R(a)$ as a proxy for the optimization problem $\inf_{a \in \mathcal{A}} r(a)$. Let us use the notation

$$A^\star \in \operatorname*{argmin}_{a \in \mathcal{A}} R(a)$$

to denote any of the minimizers of R in $\mathcal{A}$ (again, we assume that the minimum is attained, but there could be more than one minimum).

A first question that we set to investigate in this course is the development of upper bounds for the estimation error that is obtained when the action $A \in \mathcal{A}$ chosen by the decision maker is given by the ERM rule $A^\star$, namely, $r(A^\star) - r(a^\star)$. We are interested in deriving two types of bounds.

Bounds in expectation: Find `Expectation`, a positive quantity (depending on n and $\mathcal{A}$), such that

$$\mathbf{E}\, r(A^\star) - r(a^\star) \leq \boxed{\texttt{Expectation}}$$

Bounds in probability: Find `UpperTail`, a strictly decreasing function of ε, such that for any $\varepsilon \geq 0$

$$\mathbf{P}\Big(r(A^\star) - r(a^\star) \geq \varepsilon\Big) \leq \boxed{\texttt{UpperTail}(\varepsilon)}$$

or, equivalently (setting $\texttt{UpperTail}(\varepsilon) = \delta$), for any $\delta \in [0,1]$

$$\mathbf{P}\Big(r(A^\star) - r(a^\star) < \boxed{\texttt{UpperTail}^{-1}(\delta)}\Big) \geq 1 - \delta.$$

To get an understanding of the main ideas that we need to develop, let us consider the following decomposition of the estimation error:

$$\boxed{r(A^\star) - r(a^\star) = r(A^\star) - R(A^\star) + \underbrace{R(A^\star) - R(a^\star)}_{\leq 0} + R(a^\star) - r(a^\star)}$$

where the inequality below the curly brackets follows from the fact that by definition of $A^\star$, $R(A^\star) \leq R(a)$ for any $a \in \mathcal{A}$. It follows that

$$r(A^\star) - r(a^\star) \leq r(A^\star) - R(A^\star) + R(a^\star) - r(a^\star).$$

Ideally, we would like to be able to derive bounds for the quantities on the right hand side of the previous inequality. However, this is not an easy task as the action $A^\star$ is possibly a very involved function of the random sample S. *Uniform learning* is a learning paradigm that circumvents this problem by taking the supremum in the above inequality over all possible actions in $\mathcal{A}$, namely,

$$\boxed{r(A^\star) - r(a^\star) \leq \underbrace{\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} + \sup_{a\in\mathcal{A}}\{R(a) - r(a)\}}_{\texttt{Statistics}}}$$

We are then left with deriving bounds for the `Statistics` term, i.e., for the quantity $\sup_{a\in\mathcal{A}}\{r(a) - R(a)\}$ and its symmetric version. As we will see, this is now a more amenable task as for any *deterministic* $a \in \mathcal{A}$, the function $R(a)$ is a simple function of the random sample S (recall that $R(a)$ is a sum of independent random variables).

Bounds in expectation: To establish bounds in expectation, it is enough to find `ExpectationStats`, a positive quantity, such that

$$\begin{aligned} \mathbf{E}\sup_{a\in\mathcal{A}}\{R(a) - r(a)\} &\leq \frac{1}{2}\boxed{\texttt{ExpectationStats}} \\ \mathbf{E}\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} &\leq \frac{1}{2}\boxed{\texttt{ExpectationStats}} \end{aligned} \tag{1.1}$$

Clearly,

$$\mathbf{E}\, r(A^\star) - r(a^\star) \leq \boxed{\texttt{ExpectationStats}}$$

Establishing bounds of the type (1.1) falls within the scope of bounding the expected value of the supremum of empirical processes, or, equivalently, establishing non-asymptotic results for the *uniform* law of large numbers.

Bounds in probability: To establish bounds in probability, it is enough to find `UpperTailStats`, a strictly decreasing function of ε such that for any $\varepsilon \geq 0$

$$\begin{aligned} \mathbf{P}\Big(\sup_{a\in\mathcal{A}}\{R(a) - r(a)\} \geq \mathbf{E}\sup_{a\in\mathcal{A}}\{R(a) - r(a)\} + \varepsilon\Big) &\leq \frac{1}{2}\boxed{\texttt{UpperTailStats}(\varepsilon)} \\ \mathbf{P}\Big(\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} \geq \mathbf{E}\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} + \varepsilon\Big) &\leq \frac{1}{2}\boxed{\texttt{UpperTailStats}(\varepsilon)} \end{aligned} \tag{1.2}$$

or, equivalently, for any $\delta \in [0,1]$

$$\mathbf{P}\Big(\sup_{a\in\mathcal{A}}\{R(a) - r(a)\} < \mathbf{E}\sup_{a\in\mathcal{A}}\{R(a) - r(a)\} + \boxed{\texttt{UpperTailStats}^{-1}(2\delta)} \Big) \geq 1-\delta$$
$$\mathbf{P}\Big(\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} < \mathbf{E}\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} + \boxed{\texttt{UpperTailStats}^{-1}(2\delta)} \Big) \geq 1-\delta$$

In fact, with probability at least $1-2\delta$ we have (see **Problem 1.3** in the Problem Sheets)

$$\begin{aligned} r(A^\star) - r(a^\star) &\leq \sup_{a\in\mathcal{A}}\{r(a) - R(a)\} + \sup_{a\in\mathcal{A}}\{R(a) - r(a)\} \\ &< \mathbf{E}\sup_{a\in\mathcal{A}}\{R(a) - r(a)\} + \mathbf{E}\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} + 2\boxed{\texttt{UpperTailStats}^{-1}(2\delta)} \\ &\leq \boxed{\texttt{ExpectationStats}} + 2\boxed{\texttt{UpperTailStats}^{-1}(2\delta)} \end{aligned}$$

so that

$$\mathbf{P}\Big(r(A^\star) - r(a^\star) < \boxed{\texttt{ExpectationStats}} + 2\boxed{\texttt{UpperTailStats}^{-1}(\delta)} \Big) \geq 1-\delta.$$

Establishing bounds of the type (1.2) falls within the scope of establishing *concentration inequalities* for a deterministic function f of random variables $Z_1, \ldots, Z_n$, namely, find $\texttt{UpperTail}_f$, a strictly decreasing function of ε such that for any $\varepsilon \geq 0$

$$\mathbf{P}\Big(f(Z_1,\ldots,Z_n) - \mathbf{E}\, f(Z_1,\ldots,Z_n) \geq \varepsilon \Big) \leq \boxed{\texttt{UpperTail}_f(\varepsilon)}$$

or, equivalently, for any $\delta \in [0,1]$,

$$\mathbf{P}\Big(f(Z_1,\ldots,Z_n) - \mathbf{E}\, f(Z_1,\ldots,Z_n) < \boxed{\texttt{UpperTail}_f^{-1}(\delta)} \Big) \geq 1-\delta.$$

In the case we are interested in, $f(Z_1,\ldots,Z_n) = \sup_{a\in\mathcal{A}}\{R(a) - r(a)\}$. Concentration inequalities deal with establishing how close a function of many random variables is to its expected value. These inequalities will play a crucial role in this course.

1.2.2 Optimization

In the previous section we implicitly assumed that the decision maker who wants to adopt the ERM paradigm can in fact compute $A^\star$, which amounts to solving the ERM optimization problem exactly. Hence, the focus in the above was on investigating the *statistical* properties of $A^\star$. In practice, however, computing $A^\star$ is intractable (NP hard) for most problems of interest, so we need a more flexible strategy to solve our problem. Furthermore, even when the ERM problem is tractable, the decision maker may still decide to run a more computationally efficient algorithm to get an *approximation* A to the exact ERM minimizer $A^\star$. This is the case, for instance, when computing $A^\star$ involves inverting a $n \times n$ matrix which, in general, can be done in time $O(n^3)$ via procedures based on Gaussian elimination. On the other hand, often (e.g. for positive semidefinite matrices) an approximate solution to the inverse can be obtained in time $O(n^2)$ via methods based on gradient descent (fast solvers) [9].

Henceforth, let us denote by A an approximation to the ERM minimizer (e.g. think of A as the output of a gradient descent algorithm run to minimize the function R over elements in $\mathcal{A}$). The approximation A is also a random quantity, as it is a function of the data and possibly of other sources of randomness. Let us consider the following new decomposition for the estimation error:

$$r(A) - r(a^\star) = r(A) - R(A) + R(A) - R(A^\star) + \underbrace{R(A^\star) - R(a^\star)}_{\leq 0} + R(a^\star) - r(a^\star),$$

which yields

$$\boxed{r(A) - r(a^\star) \leq \underbrace{R(A) - R(A^\star)}_{\texttt{Optimization}} + \underbrace{\sup_{a\in\mathcal{A}}\{r(a) - R(a)\} + \sup_{a\in\mathcal{A}}\{R(a) - r(a)\}}_{\texttt{Statistics}}}$$

The new term $R(A) - R(A^\star)$ represents the `Optimization` term, and in this new setting computing error bounds for the estimation error $r(A) - r(a^\star)$ entails also computing error bounds for the optimization term.

Bounds in expectation: Find `ExpectationOpt`, a positive constant, such that

$$\mathbf{E}[R(A) - R(A^\star)] \leq \boxed{\texttt{ExpectationOpt}}$$

Clearly,

$$\mathbf{E}\, r(A) - r(a^\star) \leq \boxed{\texttt{ExpectationOpt}} + \boxed{\texttt{ExpectationStats}}$$

A comparison between the bound for the statistics term and for the optimization term can be used to tune the optimization routine to find an approximate solution up to the precision given by the statistical accuracy. For instance, following the guidelines in [1], we only need to solve the optimization problem up to the precision needed so that the following holds:

$$\boxed{\texttt{ExpectationOpt}} \lesssim \boxed{\texttt{ExpectationStats}}$$

Bounds in probability: To establish bounds in probability, it is enough to find `UpperTailOpt`, a strictly decreasing function of ε such that for any $\varepsilon \geq 0$

$$\mathbf{P}\Big(R(A) - R(A^\star) \geq \mathbf{E}[R(A) - R(A^\star)] + \varepsilon\Big) \leq \boxed{\texttt{UpperTailOpt}(\varepsilon)}$$

or, equivalently, for any $\delta \in [0, 1]$

$$\mathbf{P}\Big(R(A) - R(A^\star) < \mathbf{E}[R(A) - R(A^\star)] + \boxed{\texttt{UpperTailOpt}^{-1}(\delta)}\Big) \geq 1 - \delta.$$

Proceeding as above (see **Problem 1.3** in the Problem Sheets), we find

$$\mathbf{P}\Big(r(A) - r(a^\star) < \boxed{\texttt{ExpectationStats}} + \boxed{\texttt{ExpectationOpt}} + 2\boxed{\texttt{UpperTailStats}^{-1}(2\delta/3)} + \boxed{\texttt{UpperTailOpt}^{-1}(\delta/3)}\Big) \geq 1 - \delta.$$

1.3 Offline Statistical Learning: Estimation

The standard statistical learning framework for estimation is defined as follows.

Algorithm 2: Statistical Learning: Estimation

1. Observe a sample of training examples $S = \{Z_1, \ldots, Z_n\} \in \mathcal{Z}^n$, assumed to be i.i.d. from a probability distribution supported on $\mathcal{Z}$ that is parametrized by a parameter $a^\star \in \mathcal{A}$.
2. As a function of the random sample S (and possibly some external source of randomness), choose a parameter $A \in \mathcal{A}$.
3. Suffer a loss $\ell(A, a^\star)$ where $\ell : \mathcal{A} \times \mathcal{A} \to \mathbb{R}_+$ is a given *estimation* loss function.

Goal: Minimize and control the estimation error $\ell(A, a^\star)$ as a function of the sample size n and notions of "complexity" of the action set $\mathcal{A}$ of the loss function ℓ.

In some settings, the prediction and estimation problems are closely connected. See **Problem 1.4** in the Problem Sheets.

1.4 Online Statistical Learning

The standard online statistical learning framework is defined as follows [2].

Algorithm 3: Online Statistical Learning

At every time step $t = 1, 2, \ldots, n$:

1. Choose an action $A_t \in \mathcal{A}$ (possibly using some external source of randomness), where $\mathcal{A}$ is a set of admissible actions.
2. A data point $Z_t \in \mathcal{Z}$ is sampled from an *unknown* distribution. The setting where Z_t is revealed to the player is called the *full information* setting. The setting where Z_t is not revealed to the player is called the *limited information* setting, or *bandit* setting.
3. Suffer a loss $\ell(A_t, Z_t)$ where $\ell : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}_+$ is a given loss function. Let $r : \mathcal{A} \to \mathbb{R}_+$ be the expected/population risk, defined as the average loss function

$$\boxed{r(a) := \mathbf{E}\, \ell(a, Z)}$$

Define the *normalized pseudo-regret* as follows

$$\boxed{\frac{1}{n} \sum_{t=1}^{n} r(A_t) - \inf_{a \in \mathcal{A}} r(a)}$$

Goal: Minimize and control the normalized pseudo-regret as a function of the sample size n and notions of "complexity" of the action set $\mathcal{A}$ of the loss function ℓ.

The cumulative pseudo-regret is the difference between the cumulative average loss of the player and the cumulative average loss of the best action in hindsight (i.e., the having access to the data points $Z_1, \ldots, Z_n$).

Remark 1.9 (Offline and online statistical learning) *The main difference between the offline statistical learning framework for prediction defined in Algorithm 1 and the online statistical learning framework is that in the former case the action $A \in \mathcal{A}$ can be chosen to be a function of the full training data $Z_1, \ldots, Z_n$, while in the latter case each function A_t can only be a function of the information available at time t, namely, $\{A_1, \ldots, A_{t-1}\}$ and $\{Z_1, \ldots, Z_{t-1}\}$ in the full information setting, or $\{A_1, \ldots, A_{t-1}\}$ and $\{\ell(A_1, Z_1), \ldots, \ell(A_{t-1}, Z_{t-1})\}$ in the bandit setting.*

References

[1] Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

[2] Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.

[3] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[4] Tor Lattimore and Csaba Szepesvári. Bandit algorithms, 2018.

[5] Philippe Rigollet. *Mathematics of Machine Learning*. MIT Course, 2015.

[6] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[7] Ramon van Handel. Probability in high dimension. Technical report, PRINCETON UNIV NJ, 2014.

[8] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1995.

[9] Nisheeth K Vishnoi et al. Lx= b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013.

[10] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.