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 [6, 5, 3, 4], 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.

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 \( T \), we denote by \( |T| \) its cardinality. For a positive integer \( n \), we use the notation \( [n] = \{1, \ldots, n\} \). For a vector \( x = (x_1, \ldots, x_d) \in \mathbb{R}^d \), \( p \geq 1 \) we let \( \|x\|_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p} \) denote the \( \ell_p \) norm and \( \|x\|_{\infty} = \max_{i \in [d]} |x_i| \) denote the \( \ell_{\infty} \) norm. We denote by \( x^\top \) the transpose of \( x \).
1.2 Offline Statistical Learning: Prediction

The standard statistical learning framework for prediction is defined as follows [7].

<table>
<thead>
<tr>
<th>Algorithm 1: Statistical Learning: Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Observe a sample of training examples $S = {Z_1, \ldots, Z_n} \in \mathcal{Z}^n$, assumed to be i.i.d. from an unknown probability distribution supported on a space $\mathcal{Z}$.</td>
</tr>
<tr>
<td>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 $\mathcal{B}$.</td>
</tr>
<tr>
<td>3. Let $\ell : \mathcal{B} \times \mathcal{Z} \to \mathbb{R}<em>+$ be a given prediction loss function. Let $r : \mathcal{B} \to \mathbb{R}</em>+$ be the expected or population risk, defined as the average loss function $r(a) := E \ell(a, Z)$ where $Z \in \mathcal{Z}$ is a new, independent, test data point from the same unknown data distribution.</td>
</tr>
</tbody>
</table>

Define the excess risk as follows

\[
\begin{align*}
    r(A) - \inf_{a \in \mathcal{B}} r(a) = r(A) - \inf_{a \in \mathcal{A}} r(a) + \inf_{a \in \mathcal{A}} r(a) - \inf_{a \in \mathcal{B}} r(a) \\
    \text{excess risk} & \quad \text{estimation error} & \quad \text{approximation error}
\end{align*}
\]

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 $\ell$.

In the following we assume that the minima are attained, and we denote them by

\[
    a^* \in \arg\min_{a \in \mathcal{A}} r(a),
    \quad a^{**} \in \arg\min_{a \in \mathcal{B}} r(a),
\]

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

\[
    r(A) - r(a^{**}) = r(A) - r(a^*) + r(a^*) - r(a^{**}).
\]

The estimation error measures how much extra (average) loss the decision maker suffers by choosing action $A$ compared to the 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 $\ell$. 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 1.2 (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 is performing, 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 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 control on other sources of randomness. ...on 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 $A$ and the loss function $\ell$, and they can be data-dependent (i.e., depend on data distribution) or data-independent.

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 $X \times Y$. 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 $X$ to $Y$, i.e., $A \subseteq B := \{a: X \to Y\}$, and the loss function is of the form $\ell(a,(x,y)) = \phi(a(x),y)$, for a function $\phi: Y \times X \rightarrow \mathbb{R}_+$. In this setting, the goal is to use the training sample $S$ to choose a predictor or hypothesis $a \in A$ that can be used to predict the label $Y$ of a new test example $(X,Y)$. In this case, the optimal decision in $B$ given by $a^{**} = \arg\min_{a \in A} \mathbb{E}\phi(a(X),Y)$ is called the Bayes decision rule, and its corresponding value of risk $r(a^{**}) = \mathbb{E}\phi(a^{**}(X),Y)$ is called the Bayes risk. As the following lemma shows, the Bayes decision rule for any loss function $\phi$ is the solution of a deterministic minimization problem, which depends on the conditional distribution $P(Y \in \cdot | X)$ (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 sense, 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 in the applications that we will consider.

**Lemma 1.3 (Bayes decision rule)** We have

$$a^{**}(x) = \arg\min_{\hat{y} \in Y} \mathbb{E}\phi(\hat{y},Y)|X = x] = \arg\min_{\hat{y} \in Y} \int \phi(\hat{y}, y) P(Y \in dy | X = x).$$

**Proof:** By construction, for any $\hat{y} \in Y$ we have

$$\mathbb{E}\phi(a^{**}(x),Y)|X = x] \leq \mathbb{E}\phi(\hat{y},Y)|X = x]$$

so that, for any $a \in A$,

$$\mathbb{E}\phi(a^{**}(x),Y)|X = x] \leq \mathbb{E}\phi(a(x),Y)|X = x].$$

By the tower property of conditional expectations, we have

$$\mathbb{E}\phi(a^{**}(X),Y) = \mathbb{E}\mathbb{E}\phi(a^{**}(X),Y)|X] \leq \mathbb{E}\mathbb{E}\phi(a(X),Y)|X] = \mathbb{E}\phi(a(X),Y).$$

**Example 1.4 (Regression)** The regression setting is represented by the choice $X = \mathbb{R}^d$ for a given dimension $d$, $Y = \mathbb{R}$, and $B$ is the set of functions from $X$ to $Y$.

Typical choices of loss functions $\phi$ are as follows.

- **$\ell_2$ loss function.** $\phi(\hat{y}, y) = (\hat{y} - y)^2$. The Bayes decision rule is the conditional mean, $a^{**}(x) = \mathbb{E}[Y|X = x]$. See Problem 1.2 in the Problem Sheets.

- **$\ell_1$ loss function.** $\phi(\hat{y}, y) = |\hat{y} - y|$. The Bayes decision rule is the conditional median, $a^{**}(x) = \text{Median}[Y|X = x]$. (note that the conditional median need not be unique)

Typical choices of admissible set $A$ 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 restriction on the values of the parameters, such as $\|w\|_2 \leq R$ or $\|w\|_1 \leq R$.

• Neural networks. (to be defined later on in the course)

• Kernel methods. (to be defined later on in the course)

Example 1.5 (Classification) The classification setting is a particular case of regression, and it is represented by the choice $X = \mathbb{R}^d$ for a given dimension $d$, $Y = \{y_1, \ldots, y_k\}$ for a given $k$, and $B$ is the set of functions from $X$ to $Y$. In this setting, the typical loss function is given by the zero-one loss.

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

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

By Lemma 1.3, the Bayes decision rule reads $a^*(x) = \arg\max_{\hat{y} \in Y} P(Y = \hat{y} | X = x)$, 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 $r(a) = \mathbb{E} \phi(a(X), Y) = P(a(X) \neq Y)$. 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 $Y = \{-1, 1\}$. In this case, the admissible action set $\mathcal{A}$ is typically taken to be the sign of the predictors used in regression, such as $a(x) = \text{sign}(w^\top x + b)$. It is also common to consider convex relaxation of the zero-one loss. If $Y = \{-1, 1\}$, 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 $1_{\hat{y}y \leq 0}$, so $\varphi(u) = 1_{u \leq 0}$. Convex losses that uniformly bound from above the zero-one loss, namely, $1_{u \leq 0} \leq \varphi(u)$ for all $u \in \mathbb{R}$, are given below.

• Hinge loss (Support Vector Machines). $\varphi(u) = \max\{1 - u, 0\}$.

• Exponential loss. $\varphi(u) = e^{-u}$.

• Logistic loss. $\varphi(u) = \log_2(1 + e^{-u})$.

1.2.1 Statistics

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

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

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 $a \in B$, 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 \( \inf_{a \in A} R(a) \) as a proxy for the optimization problem \( \inf_{a \in A} r(a) \). Let us use the notation
\[
A^* \in \arg\min_{a \in A} R(a)
\]
to denote any of the minimizers of \( R \) in \( A \) (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 \( A^* \in A \) chosen by the decision maker is given by the ERM rule \( A^* \), namely, \( r(A^*) - r(a^*) \). We are interested in deriving two types of bounds.

**Bounds in expectation:** Find \( \text{Expectation} \), a positive constant (depending on \( n \) and \( A \)), such that
\[
\mathbb{E} [r(A^*) - r(a^*)] \leq \text{Expectation}
\]

**Bounds in probability:** Find \( \text{UpperTail} \), a strictly decreasing function of \( \varepsilon \), such that for any \( \varepsilon \geq 0 \)
\[
\mathbb{P} \left( r(A^*) - r(a^*) \geq \varepsilon \right) \leq \text{UpperTail}(\varepsilon)
\]
or, equivalently (setting \( \text{UpperTail}(\varepsilon) = \delta \)), for any \( \delta \in (0, 1) \)
\[
\mathbb{P} \left( r(A^*) - r(a^*) < \text{UpperTail}^{-1}(\delta) \right) \geq 1 - \delta.
\]

To get an understanding of the main ideas that we need to develop, let us consider the following decomposition of the estimation error:
\[
r(A^*) - r(a^*) = r(A^*) - R(A^*) + R(A^*) - R(a^*) + R(a^*) - r(a^*) \leq 0
\]
where the inequality comes from the fact that by definition of \( A^* \), \( R(A^*) \leq R(a) \) for any \( a \in A \). That is,
\[
r(A^*) - r(a^*) \leq r(A^*) - R(A^*) + R(a^*) - r(a^*).
\]

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 \( A^* \) is a possibly-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 \( A \), namely,
\[
r(A^*) - r(a^*) \leq \sup_{a \in A} \{r(a) - R(a)\} + \sup_{a \in A} \{R(a) - r(a)\}
\]

We are then left with deriving bounds for the \( \text{Statistics} \) term, i.e., for the quantity \( \sup_{a \in A} \{r(a) - R(a)\} \) and its symmetric version. As we will see, this is now a more amenable task as for any deterministic \( a \in A \), 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 \( \text{Expectation}_{\text{Stats}} \), a positive constant, such that
\[
\mathbb{E} \sup_{a \in A} \{r(a) - R(a)\} \leq \frac{1}{2} \text{Expectation}_{\text{Stats}}
\]
\[
\mathbb{E} \sup_{a \in A} \{r(a) - R(a)\} \leq \frac{1}{2} \text{Expectation}_{\text{Stats}}
\] (1.1)
Clearly, \[ E r(A^*) - r(a^*) \leq \text{ExpectationStats} \]

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 \( \text{UpperTailStats} \), a strictly decreasing function of \( \varepsilon \) such that for any \( \varepsilon \geq 0 \)

\[
P\left( \sup_{a \in A} \{ R(a) - r(a) \} \geq E \sup_{a \in A} \{ R(a) - r(a) \} + \varepsilon \right) \leq \frac{1}{2} \text{UpperTailStats}(\varepsilon) \]

\[
P\left( \sup_{a \in A} \{ r(a) - R(a) \} \geq E \sup_{a \in A} \{ r(a) - R(a) \} + \varepsilon \right) \leq \frac{1}{2} \text{UpperTailStats}(\varepsilon) \quad (1.2) \]

or, equivalently, for any \( \delta \in (0, 1) \)

\[
P\left( \sup_{a \in A} \{ R(a) - r(a) \} < E \sup_{a \in A} \{ R(a) - r(a) \} + \text{UpperTailStats}^{-1}(2\delta) \right) \geq 1 - \delta \]

\[
P\left( \sup_{a \in A} \{ r(a) - R(a) \} < E \sup_{a \in A} \{ r(a) - R(a) \} + \text{UpperTailStats}^{-1}(2\delta) \right) \geq 1 - \delta \]

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

\[
r(A^*) - r(a^*) \leq \sup_{a \in A} \{ r(a) - R(a) \} + \sup_{a \in A} \{ R(a) - r(a) \} \]

\[
< E \sup_{a \in A} \{ R(a) - r(a) \} + E \sup_{a \in A} \{ r(a) - R(a) \} + 2 \text{UpperTailStats}^{-1}(2\delta) \]

\[
\leq \text{ExpectationStats} + 2 \text{UpperTailStats}^{-1}(2\delta) \]

so that

\[
P\left( r(A^*) - r(a^*) < \text{ExpectationStats} + 2 \text{UpperTailStats}^{-1}(\delta) \right) \geq 1 - \delta. \]

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 \( \text{UpperTail}_f \), a strictly decreasing function of \( \varepsilon \) such that for any \( \varepsilon \geq 0 \)

\[
P\left( f(Z_1, \ldots, Z_n) - E f(Z_1, \ldots, Z_n) \geq \varepsilon \right) \leq \text{UpperTail}_f(\varepsilon) \]

or, equivalently, for any \( \delta \in (0, 1) \),

\[
P\left( f(Z_1, \ldots, Z_n) - E f(Z_1, \ldots, Z_n) < \text{UpperTail}_f^{-1}(\delta) \right) \geq 1 - \delta. \]

In the case we are interested in, \( f(Z_1, \ldots, Z_n) = \sup_{a \in A} \{ R(a) - r(a) \} \).

### 1.2.2 Optimization

In the previous section we implicitly assumed that the decision maker who wants to adopt the ERM algorithm can compute \( A^* \). Hence, the focus was on the statistical properties of \( A^* \). In practice, however, computing \( A^* \) is intractable for most problems of interest, and even when it is tractable the decision maker may still decide
to run a more computationally-convenient algorithm to get an approximation $A$ to the ERM minimizer $A^\star$. In this case, let us consider the following new decomposition for the estimation error:

$$r(A) - r(a^\star) = r(A) - R(A) + R(A) - R(A^\star) + R(A^\star) - R(a^\star) + R(a^\star) - r(a^\star),$$

which yields

$$r(A) - r(a^\star) \leq R(A) - R(A^\star) + \sup_{a \in \mathcal{A}} \{r(a) - R(a)\} + \sup_{a \in \mathcal{A}} \{R(a) - r(a)\}.$$

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 $\text{ExpectationOpt}$, a positive constant, such that

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

Clearly,

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

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 [1]:

$$\text{ExpectationOpt} \lesssim \text{ExpectationStats}$$

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

$$\mathbf{P} \left( R(A) - R(A^\star) \geq \mathbf{E}[R(A) - R(A^\star)] + \varepsilon \right) \leq \text{UpperTailOpt}(\varepsilon)$$

or, equivalently, for any $\delta \in (0, 1)$

$$\mathbf{P} \left( R(A) - R(A^\star) < \mathbf{E}[R(A) - R(A^\star)] + \text{UpperTailOpt}^{-1}(\delta) \right) \geq 1 - \delta.$$

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

$$\mathbf{P} \left( r(A) - r(a^\star) < \text{ExpectationStats} + \text{ExpectationOpt} \right.$$}

$$+ 2 \text{UpperTailStats}^{-1}(2\delta/3) + \text{UpperTailOpt}^{-1}(\delta/3) \right) \geq 1 - \delta.$$

### 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 $a^* \in \mathcal{A}$.

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 $\ell(A, a^*)$ where $\ell : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}_+$ is a given estimation loss function.

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

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 $\ell : \mathcal{A} \times \mathcal{Z} \rightarrow \mathbb{R}_+$ is a given loss function. Let $r : \mathcal{A} \rightarrow \mathbb{R}_+$ be the expected/population risk, defined as the average loss function

$$r(a) := \mathbb{E}_a \ell(a, Z)$$

Define the normalized pseudo-regret as follows

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

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 $\ell$.

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 1.6 (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


