Contents

1 Lecture 1

We define **machine learning** as a computational method to convert experience into expertise. We say that *experience* is past data, and *expertise* is the prediction of future outcomes.

Some standard learning tasks are: classification, regression, clustering, dimensionality reduction/manifold learning (find lower dimensional representation of data while preserving its properties).

In this course:

1. Theretical foundations and statistical learning theory. PAC learning framework, Rademacher complexity, VC-dimension.
2. Analysis of ML methods & algorithms (with applications of part 1).
   (a) SVM & kernel methods
   (b) Boosting
   (c) Logistic regression
   (d) SGD
   (e) Neural networks (deep learning)

There may be a part 2 of this course.

1.1 Basic ML Setup

We have an **input space** X for example $X = \mathbb{R}^n, [0,1]^2, \ldots$. We have an **output space** Y, which cna be for example $\{0,1\}, \mathbb{R}, \ldots$. We also have **training data**, which are tuples with the data and a label: $(x_i, y_i),\ i = 1, \ldots, m$ for labeled data, or unlabeled data given by simply a list $x_i, i = 1, \ldots, m$. We also have a **hypothesis class** $\mathcal{H}$, which is a set of functions $h : X \to Y$.

Now, the question is: what class of functions should we learn? Why do we need to choose $\mathcal{H}$? We use the example of a papaya classifcation, and describe the natural way to iterate from overfit data to a simple rectangular classifier, on the inputs of softness and color as two axes (therefore the input space is $X = [0,1]^2$). This is described in some detail in the live notes, but does not warrant TeXed notes.

Now, there are several different types of learning that are sketched.

1. Supervised.
2. Unsupervised.
3. Semisupervised.
4. Online learning.
5. RL.
6. Active learning.

1.2 PAC Learning Framework

We now describe the **PAC learning framework**. Consider a supervised learning scenario for binary classification. We have X our input space, Y our output, and training data

$$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times Y)^m. \tag{1}$$

We make some assumptions.

1. $x_i,\ i = 1, \ldots, m$ are drawn iid, according to some *unknown* probability distribution $\mathcal{D}$ on X.
2. Labels are given $y_i = f(x_i),\ i = 1, \ldots, m$ for some map $f : X \to Y$.

Our goal is to find f (at least approximately). More generally (later), we will assume that (x_i, y_i) are drawn iid from $\mathcal{D}$ on $X \times Y$.

The learner's output will be a prediction rule

$$h : X \to Y, \tag{2}$$

and ideally $h = f$. The learner will select a hypothesis from

$$\mathcal{H} \subset \{g : X \to Y\} \tag{3}$$

and f may or may not be contained in $\mathcal{H}$.

Assumptions on measurable spaces: several assumptions are made, simply to remove pathological mathematical cases. For measurable spaces $X, Y, Z, \ldots$:

1. If the space is finite/countably infinite, it is equipped with the σ-algebra consisting of all subsets of the space.
2. Otherwise, assume it is a metric space, complete and separable, equipped with the corresponding Borel σ-algebra.
3. For products $X \times Y$, it carries the product σ-algebra of X and Y.

We can also define the **generalization error**.

Definition 1.2.1: Generalization Error/Risk

*For $h : X \to Y = \{0, 1\}$, the **generalization error** or **risk** of h is:*

$$R(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}(h(x) \neq f(x)) = \mathbb{E}[\mathbb{1}_{\{x | h(x) \neq f(x)\}}]. \tag{4}$$

To see that this makes sense, note that

$$\mathbb{P}(A) = \mathbb{E}[\mathbb{1}_A], \tag{5}$$

since

$$\mathbb{P}(A) = \int_A 1 \, dP(\omega) \tag{6}$$

$$\mathbb{E}[\mathbb{1}] = \int_\Omega \mathbb{1}_A \, dP(\omega). \tag{7}$$

Also note that above, we assumed a fixed labeling function $f : X \to Y = \{0, 1\}$ and probability distribution $\mathcal{D}$ on X.

2 Lecture 2

Definition 2.0.1: Empirical Risk

*Let $h : X \to Y = \{0, 1\}$, the (true) labeling function is $f : X \to Y$, and sames $S = (x_1, \ldots, x_m)$ (with $x_i \in X$). Then the **empirical risk** is*

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\{h(x_i) \neq f(x_i)\}} = \frac{1}{m} \# \{i \in [m] : h(x_i) \neq f(x_i)\} \tag{8}$$

where $[m] := \{1, \ldots, m\}$; for a finite set A, we have that $\# A = |A|$.

Drawing $x_i, \ldots, x_m$ from $\mathcal{D}$ means that we draw $S = (x_1, \ldots, x_m) \sim \mathcal{D}$. Consequently, we obtain a random variable which we also denote by $\hat{R}(h)$.

Lemma 2.0.2: Expectation of Empirical Risk equals Risk

If $x_1, \ldots, x_m$ are drawn i.i.d. according to $\mathcal{D}$, then for any (measurable) $h : X \to \{0, 1\}$,

$$\mathbb{E}[\hat{R}] = R(h). \tag{9}$$

Definition 2.0.3: Empirical Risk Minimization

Let $f : X \to Y = \{0,1\}$ be the true labeling function and let $\mathcal{H} \subset \{h : X \to Y\}$ be a hypothesis set. Given a sample $S = (x_1, \ldots, x_m) \in X^m$ with corresponding labels $y_i = f(x_i)$, for $i \in \{1, \ldots, m\}$, the ***empirical risk minimization*** *consists of selecting a minimizer $h \in \mathcal{H}$ of $\hat{R}$, i.e. selecting an h realizing*

$$\min_{h \in \mathcal{H}} \hat{R}(h) = \min_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{h(x_i \neq f(x_i)}. \tag{10}$$

An example of overfitting is provided: we can easily overfit with empirical risk minimization, see live notes.

Definition 2.0.4: PAC-learning, Consistent Case

A hypothesis class $\mathcal{H} \subset \{h : X \to Y = \{0,1\}\}$ is ***PAC-learnable*** *if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ with the following property:*

For every $\varepsilon, \delta \in (0,1)$, for all probability distributions $\mathcal{D}$ over X, for every labeling function $f \in \mathcal{H}$, the following holds:

If $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ and $S = (x_1, \ldots, x_m)$ is an i.i.d. sample, $S \sim D^m$, then given the data $(x_i, y_i) = (x_i, f(x_i))$, $i = 1, \ldots, m$, the algorithm $\mathcal{A}$ returns a hypothesis

$$h_S \in \mathcal{H} \tag{11}$$

such that with probability of at least $1 - \delta$ (over $S \sim D^m$),

$$R(h_S) \leq \varepsilon. \tag{12}$$

Remarks.

1. There are analogous definitions of empirical risk, empirical risk minimization, PAC-learnable for any Y consisting of two elements.

2. The **sample complexity** $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ determines the number of required training data in order to learn $\mathcal{H}$. It will depend on the accuracy ε and the confidence δ and on properties of $\mathcal{H}$. Ideally, $m_{\mathcal{H}}$ should be bounded by a polynomial in $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$. (More precisely, the sample complexity should be defined as the minimal $m_{\mathcal{H}}$ satisfying the conditions in the definition.

3. The definition of PAC-learnable does not require that the algorithm $\mathcal{A}$ be efficient; only the existence of a possibly slow (intractable) algorithm is assumed. If the runtime of the algorithm is polynomial (in $\frac{1}{\varepsilon}, \frac{1}{\delta}$, and

the "computational representation of $f \in \mathcal{H}$"), then we call $\mathcal{H}$ **efficiently PAC-learnable**.

4. Empirical risk minimization is a possible "algorithm", but it may not always be the optimal one. Depending on $\mathcal{H}$, it may be efficient or not.

5. In some cases, PAC-learnability may be shown directly (e.g., learning axis-aligned rectangles, see Exercise Sheet 1).

Theorem 2.0.5: Finite $\mathcal{H}$, consistent case

Let $\mathcal{H}$ = a finite set of functions $h : X \to y = \{0, 1\}$. Assume that the labeling function f belongs to $\mathcal{H}$ and let $\mathcal{A}$ be an algorithm that for each i.i.d. sample $S = (x_1, \ldots, x_m)$ and labeled training data $(x_i, y_i) = (x_i, f(x_i))$, $i = 1, \ldots, m$ returns a consistent hypothesis $h_S \in \mathcal{H}$; i.e., $\hat{R}(h_S) = 0$. Then for $\varepsilon, \delta \in (0, 1)$, if

$$m \geq \frac{1}{\varepsilon}(\log |\mathcal{H}| + \log\left(\frac{1}{\delta}\right) \tag{13}$$

the inequality

$$\mathbb{P}[R(h_S) \leq \varepsilon] \geq 1 - \delta \tag{14}$$

holds.

In other words, with probability at least $1 - \delta$,

$$R(h_S) \leq \frac{1}{m}(\log |\mathcal{H}| + \log\left(\frac{1}{\delta}\right). \tag{15}$$

Proof. Proof in live notes. Please examine it carefully. □

The theorem shows in its assumptions that $\mathcal{H}$ is PAC-learnable with

$$m_{\mathcal{H}}(\varepsilon, \delta) = \frac{1}{\varepsilon}\lceil \log |\mathcal{H}| + \log\left(\frac{1}{\delta}\right)\rceil. \tag{16}$$

Conclusion.

1. For finite hypothesis set $\mathcal{H}$, a consistent learning algorithm $\mathcal{A}$ is a PAC learning algorithm with sample complexity polynomial (even linear) in $\frac{1}{\varepsilon}$, and logarithmic in $\frac{1}{\delta}$ and $|\mathcal{H}|$.

2. $\log |\mathcal{H}|$ may be interpreted as the number of bits to represent $\mathcal{H}$ (up to a constant factor).

3. Note that $f \in \mathcal{H}$ guarantees that empirical risk minimization always returns an h_S with $\hat{R}(h_S)$.

4. Note: "**consistent** h" means that $\hat{R}(h) = 0$. The "**consistent case**" is where $f \in \mathcal{H}$.

Index