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

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

We make some assumptions.

1. $x_i,\ i = 1, \dots, m$ are drawn iid, according to some *unknown* probability distribution $\mathcal{D}$ on X.

2. Labels are given $y_i = f(x_i),\ i = 1, \dots, m$ 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 $X, Y, Z, \dots$:

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

*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.

Index