LECTURE #2

LECTURER: ELAD HAZAN SCRIBE: ARIEL SCHVARTZMAN

Last time we introduced the definition of the PAC learning model; today, we will discuss it in more detail. In particular, we will discuss its pros and cons, show the impossibility of learning unrestricted hypothesis classes (see Theorem 2.1 for formal statement), and demonstrate that the Empirical Risk Minimization (ERM) algorithm we saw last time can be used to learn some "rich" hypothesis classes.

1 PAC Learnability.

Recall that for our learning problems, our data points come from some domain set $\mathcal{X}$, have labels belonging to some set $\mathcal{Y}$, and can be sampled i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. We are interested in finding some hypothesis $h \in \mathcal{H}$ that explains an underlying concept $f : \mathcal{X} \to \mathcal{Y}$. We will measure the effectiveness of our hypothesis through a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. In many of our applications, this loss function will be 0 if we classify correctly and 1 otherwise, and we will say that the error of our hypothesis h is $\text{err}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$. For the time being, we will make the *realizability assumption*, i.e. that there exists some hypothesis $h^* \in \mathcal{H}$ such that $\text{err}(h^*) = 0$. In words, this assumption means that there is a hypothesis $h \in \mathcal{H}$ that perfectly explains the data.

In our first lecture, we established that in order to have a credible definition of learning, our error function must behave "nicely" as the number of samples goes to infinity. This is formalized as follows.

Definition 1.1. A learning problem $(\mathcal{X}, \mathcal{Y}, \ell)$ is PAC (Probably Approximately Correct) learnable with respect to the hypothesis class $\mathcal{H}$ under the realizability assumption (concept $f \in \mathcal{H}$), if there exists a function $m : [0,1]^2 \to \mathbb{R}$ and an algorithm with the following property: For every $\varepsilon, \delta > 0$, for every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, running the algorithm on $m(\varepsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis $h \in \mathcal{H}$ such that, with probability at least $1 - \delta$, $\text{err}(h) \leq \varepsilon$.

It is relevant to remind ourselves that there are two sources of randomness in this definition. The first is the distribution $\mathcal{D}$, from which the samples are drawn. The second source is the algorithm itself, which is allowed to use randomization.

This model has advantages and disadvantages: it is robust, it is oblivious to the underlying distribution, it can be efficiently used on finite hypothesis classes (which encompasses a wide variety of problems!), and there exist simple algorithms that perform well on it. However, it relies on the realizability assumption, it is offline, it can not handle infinite hypothesis classes, and it also relies on the sampling process being oblivious (as opposed to adversarial).

2 No Free Lunch Theorem.

The follow theorem shows that PAC-learning is impossible without restricting the hypothesis class $\mathcal{H}$.

Theorem 2.1 (No Free Lunch Theorem)**.** *Consider any $m \in \mathbb{N}$, any domain $\mathcal{X}$ of size $|\mathcal{X}| = 2m$, and any algorithm A which outputs a hypothesis $h \in \mathcal{H}$ given a sample S. Then there exists a concept $f : \mathcal{X} \to \{0,1\}$ and a distribution $\mathcal{D}$ such that:*

- *The error $err(f) = 0$*
- *With probability at least $\frac{1}{10}$, $err(A(S)) \geq \frac{1}{10}$.*

Proof. The proof uses the probabilistic method. We will show that for any learner, there is some learning task (i.e. "hard" concept) that it will not learn well. Formally, take $\mathcal{D}$ to be the uniform distribution over $\{(x, f(x))\}$. Our proof strategy will be to show the following inequality

$$Q \stackrel{def}{=} \mathbb{E}_{f:X\to\{0,1\}}[\mathbb{E}_{S\sim\mathcal{D}^m}[\text{err}(A(S))]] \geq \frac{1}{4}$$

as an intermediate step, and then use Markov's Inequality to conclude.

We proceed by invoking Fubini's theorem (to swap the order of expectations) and then conditioning on the event that $x \in S$.

$$\begin{aligned} Q = \mathbb{E}_S[\mathbb{E}_f[\mathbb{E}_{x\in\mathcal{X}}[A(S)(x) \neq f(x)]]] &= \mathbb{E}_{S,x}[\mathbb{E}_f[A(S)(x) \neq f(x)|x \in S]]\mathbb{P}(x \in S) \\ &+ \mathbb{E}_{S,x}[\mathbb{E}_f[A(S)(x) \neq f(x)|x \notin S]]\mathbb{P}(x \notin S) \end{aligned}$$

The first term is, in the worst case, at least 0. Also note that $\mathbb{P}(x \notin S) \geq \frac{1}{2}$. Finally, observe that $\mathbb{P}(A(S)(x) \neq f(x)) = \frac{1}{2}$ for all $x \notin S$ since we are given that the "true" concept is chosen uniformly at random. Hence, we get that:

$$Q \geq 0 + \frac{1}{2}\cdot\frac{1}{2} = \frac{1}{4},$$

which is the intermediate step we wanted to show. We conclude the proof by a simple application of the reverse Markov Inequality:

$$\mathbb{P}(Q \geq 1/10) \geq \frac{1/4 - 1/10}{1 - 1/10} \geq 1/10$$

□

3 What can we learn?

An important question to ask is what hypothesis classes can we learn? First, we show that all finite hypothesis classes are learnable. In fact, we will show that even the simple Empirical Risk Minimization (ERM) algorithm can learn them.

Theorem 3.1. *All finite hypothesis classes $\mathcal{H}$ are PAC-learnable.*

Proof. We take a sample of size m, and let $S \sim D^m$. Our ERM algorithm returns an $h_{\text{erm}} : \mathcal{X} \to \mathcal{Y} = \text{argmin}_{h\in\mathcal{H}}\{\text{err}_S(h)\} = \text{argmin}_{h\in\mathcal{H}}\,\mathbb{E}_{(x,y)\sim S}[\ell(h(x), y)]$.

Suppose that we have a hypothesis $\hat{h}$ such that $\text{err}(\hat{h}) \geq \varepsilon$. We will show that it is very unlikely our algorithm outputs such a hypothesis. Note that the probability that such a hypothesis is an empirical risk minimizer is small: $\mathbb{P}(\text{err}_S(\hat{h}) = 0) \leq (1-\varepsilon)^m$. Hence, the probability that ERM returns a bad hypothesis is

$$\mathbb{P}(\text{err}(h_{\text{ERM}}) > \varepsilon) \leq \mathbb{P}(\exists\hat{h}, \text{err}(\hat{h}) > \varepsilon, \text{err}_S(\hat{h}) = 0) \leq \sum_{\hat{h}:\text{err}(\hat{h})>\varepsilon} \mathbb{P}(\text{err}(\hat{h}) = 0) \leq (1-\varepsilon)^m|\mathcal{H}|,$$

where we have taken a union bound on the last step over all hypothesis in our class. In particular, if we want this probability to be smaller than some $\delta > 0$, it would suffice to let $m > \frac{\log(|H|/\delta)}{\varepsilon}$. □

The quantity m is often referred to as the *sample complexity* of the learning problem. It may seem that finite hypothesis classes are uninteresting, but they can still model a large variety of problems. Moreover, as we will see now and show in future lectures, we can also work with some infinite hypothesis classes.

Example 3.2. Consider the problem of learning a cut-off value. Formally, let $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \{0, 1\}$, and consider the hypothesis class $\mathcal{H} = \{h_a|a \in \mathbb{R}\}$, where $h_a(x) = 1$ if $x \geq a$ and 0 otherwise.

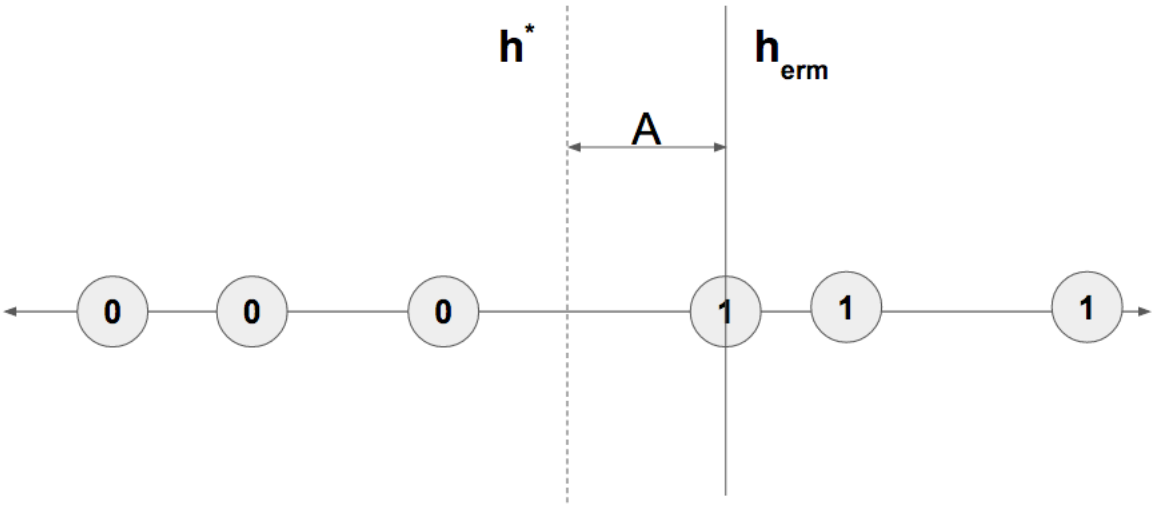

Figure 1: Learning a cut-off value

Lemma 3.3. *This infinite hypothesis class is PAC-learnable.*

Proof. We will use the ERM algorithm again. Given the realizability assumption, Figure 1 illustrates what our sample will look like.

Hence, the true h^* must lie somewhere between the last 0 and the first 1. Our algorithm will certainly return a value in this range, but it could be the wrong one. Suppose $h_{\text{erm}} \neq h^*$. Let A be the random variable that measures the probability mass between h^*, h_{erm}. Then $\mathbb{P}(A \geq \varepsilon) \leq (1-\varepsilon)^m$. In order to force this to be less than some given $\delta > 0$, we may simply set $m > \frac{\log(1/\delta)}{\varepsilon}$.

□