S&DS 669: Statistical Learning Theory

Instructor: Omar Montasser
Scribe: Anish Lakkapragada

September 2, 2025

Contents

1 PAC Learning and VC Theory **1**
1.1 Course Logistics (Lecture 1) 1
1.2 Introducing the Statistical Learning Theory Framework (Lecture 1) 1
1.3 Consistent Learning Rule Bound for Finite Hypothesis Class (Lecture 1) 2

Chapter 1

PAC Learning and VC Theory

1.1 Course Logistics (Lecture 1)

We start the class by introducing our names & majors before getting into objectives of this course. Oman also covers the syllabus (recommended prerequisites, grading, and AI policy) before going over a roadmap of the things we will cover. Okay, let's start!

1.2 Introducing the Statistical Learning Theory Framework (Lecture 1)

We now introduce the statistical learning theory framework where we have the following objects:

- Domain X (e.g. $X = \mathbb{R}^d$) where each $x \in X$ is called an "instance"
- Label Space Y (e.g. $Y = \{\pm 1\}$ or $Y = \mathbb{R}$)
- Unknown source distribution D over $X \times Y$. This is an assumption on the data generating process (formed by "nature" or "reality").
- Goal: find a predictor $h : X \to Y$ achieving small *expected error* $L_D(h) := \mathbb{P}_{(x,y)\sim D}\{h(x) \neq y\}$.
- Access to an oracle: We have an i.i.d training sample $S = \{(x_i, y_i)\}_{i=1}^m$ drawn from D (notated by $S \sim D^m$)

Restated, our goal is to create some learner $A : (X \times Y)^\star \to Y^X$, where the $\star$ denotes a variable-length sequence of $X \times Y$ (i.e. our dataset) and Y^X is the set of all functions mapping from X to Y.

Oman notes that we will first start by assuming that any instance $x \in X$ has a "ground-truth" label, as opposed to a case where D allows for 50% probability mass on $(x, +1)$ and $(x, -1)$ (such a case could happen to to reflect uncertainty in the label of x). More generally, we will start with these strong assumptions in the bulleted list above and relax them later.

It's worth emphasizing **two main assumptions about our data** within this framework:

- We observe i.i.d training samples from (unknown) distribution D.
- Future (*unseen*) examples are drawn from the same distribution D.

The second point is easier to forget.

Expected vs. Empirical Error. Let's look a bit more closely at our objective: minimizing our *expected error*

$$L_D(h) := \mathbb{P}_{(x,y)\sim D}\{h(x) \neq y\} \tag{1.1}$$

Why not minimize this directly? Answer: we don't assume access to the data distribution D. Hence, given some sample S we use the *empirical error*:

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{h(x_i) \neq y_i\} \tag{1.2}$$

as our proxy to D. While this is a typical setup in machine learning, it leads to the following questions:

- How should we use the empirical error?
- Is it a good estimate for the expected error? And how good?

Specifically, we are interested in their difference:

$$|L_D(h) - L_S(h)| = |\mathbb{P}_{(x,y)\sim D}\{h(x) \neq y\} - \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{h(x_i) \neq y_i\}| \tag{1.3}$$

Remark 1.1. Please recognize that the empirical error $L_S(h)$ is a random variable as it is a function of the randomly drawn dataset $S \sim D^m$ whereas $L_D(h)$ is just a population statistic. The relationship between the two should be more clear from the below quick exercise:

$$\forall\ h : X \to \{\pm 1\} \text{ with } D \text{ over } X \times \{\pm 1\},\ \text{show that } \mathbb{E}_{S\sim D^m}[L_S(h)] = L_D(h) \tag{1.4}$$

But this is not a useful fact as its an asymptotic, and we are more interested in the difference in the case of a finite dataset size m. Thus, we often use tools like *concentration inequalities* (e.g. Hoeffding's) to create bounds like the below for some fixed h and m:

$$\mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| > \epsilon] \leq 2\exp(-2\epsilon^2 m) \tag{1.5}$$

Or restated equivalently (i.e. define $\delta := 2\exp(-2\epsilon^2 m)$ and "invert" the probabilities),

$$\mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \leq \sqrt{\frac{\ln(2/\delta)}{2m}}] \geq 1 - \delta \tag{1.6}$$

From this expression it should be clear that as $m \to \infty$, our expected difference between expected and empirical error goes to zero.

1.3 Consistent Learning Rule Bound for Finite Hypothesis Class (Lecture 1)

Before actually creating another bound ourselves, we structure our problem even more with some prior knowledge/decisions we make:

- We restrict ourselves to a subset of functions from X to Y called our *hypothesis class* $H \subseteq Y^X$. Examples of H are given below:
  - Linear Predictors
  - Support Vector Machines (SVMs)
  - Neural Networks

H will represent our "prior knowledge" or "expert knowledge". For example, if our domain X is a set of images we would likely consider using a convolutional neural network (CNN) as our H as CNNs perform well on this kind of data.

- Assume some true function $y = f^\star(x)$ where $f^* \in H$. Our learner A will know H but not $f^\star$ (it will have to learn this function!).
- As an implication of the above assumption, we will say a sequence $((x_i, y_i))_{i=1}^m$ is *realizable* by H if the true function $f^\star \in H$ gives matching ground truth predictions[1]

Having established these assumptions, we are now ready to put them to use by creating our own bound!

Warm-up: Finite Classes. Consider the following assumptions:

- H is finite
- D is realizable by H (i.e. $\exists\ f^\star \in H$ s.t. $L_D(f^\star) = \mathbb{P}_{(x,y)\sim D}[f^\star(x) \neq y] = 0$)

Note that, as stated before, we cannot minimize $\min_{h\in H} L_D(h)$ directly and instead must work on our sample $S \sim D^m$. We present the following definition:

Definition 1.2. We have a *consistent learning rule (CLR)* when for any input $S = \{(x_i, y_i)\}_{i=1}^m$, we can output *any* $h \in H$ s.t. $\forall_{1\leq i\leq m}, h(x_i) = y_i$.

Then, we have the following question. If $\hat{h} := \text{CLR}_H(S)$ for some consistent learning rule CLR_H on hypothesis class H, what can we say about $L_S(\hat{h})$? It should be zero, but does this imply that $L_D(\hat{h}) = 0$?

Here's a closely-related example: consider some h where $L_D(h) = \frac{1}{2}$. This is a bad function that is correctly 50% of the time in truth. But $\mathbb{P}_{S\sim D^m}[L_S(h) = 0] > 0 \neq 0$, meaning $\exists\ S$ s.t. $L_S(h) = 0$ (i.e. we can be fooled to think h is good on some sample S.) Thus, we now **create a bound for a finite hypothesis class to control $L_D(h)$ on some CLR-learned h.**

Derivation of CLR bound for finite hypothesis class. Fix any function $h \in H$. We define ϵ s.t. $L_D(h) > \epsilon$. We proceed with the following steps:

- We first can find the probability of the bad event ($L_S(h) = 0$) below: [2]:
  $\mathbb{P}_{S\sim D^m}[L_S(h) = 0] = \prod_{i=1}^m \mathbb{P}_{S\sim D^m}\{h(x_i) = y_i\} = \prod_{i=1}^m (1 - L_D(h)) \leq (1-\epsilon)^m \leq \exp(-\epsilon m)$
- But this is just one bad function $\in H$! We can a *group* of bad functions with $B_\epsilon := \{h \in H : L_D(h) > \epsilon\} \subset H$. Then to get the probability that any CLR-learned function is "bad" we can use a *union bound*:

$$\mathbb{P}_{S\sim D^m}\big[\text{CLR}_H(S) \in B_\epsilon\big] \leq \mathbb{P}_{S\sim D^m}\big[\exists\, h \in B_\epsilon : L_S(h) = 0\big] \tag{1.7}$$

$$\leq \sum_{h\in B_\epsilon} \mathbb{P}_{S\sim D^m}\big[L_S(h) = 0\big] \leq |B_\epsilon|\, e^{-m\epsilon} \leq |H|\, e^{-m\epsilon}. \tag{1.8}$$

- We can then set $\delta := |H| \exp(-m\epsilon)$ and invert the expression. So to ensure $\text{CLR}_H(S) \notin B_\epsilon \iff \text{CLR}_H(S) < \epsilon$ with probability $\geq 1 - \delta$ for some predecided $\delta \in (0, 1)$, we will want need $m(\epsilon, \delta) = \frac{\ln|H| + \ln(1/\delta)}{\epsilon}$ many samples[3].

[1] Mathematically speaking, this means $\forall x_i, f^\star(x_i) = y_i \implies L_S(f^\star) = 0$. This realizability assumption is non-trivial and we will discuss it further in the course.

[2] The last argument here is done using Bernoulli's Inequality.

[3] To get this expression, solve for m in terms of δ.

Pat yourself on the back! We resummarize this bound in the following theorem:

Theorem 1.3. *For any finite class H, any (realizable) distribution D, any $(\epsilon, \delta) \in (0,1)^2$, with $m = \frac{\ln|H| + \ln(1/\delta)}{\epsilon}$, we have:*

$$\mathbb{P}_{S\sim D^m}[CLR_H(S) \leq \epsilon] \geq 1 - \delta \tag{1.9}$$

Choosing to take the perspective that our number of samples m is fixed and so we are interested in the lowest possible error we can achieve w.h.p, we can use the following theorem:

Theorem 1.4. *For any finite class H, any (realizable) distribution D, any $\delta \in (0,1), m \in \mathbb{N}$:*

$$\mathbb{P}_{S\sim D^m}[CLR_H(S) \leq \frac{\ln|H| + \ln(1/\delta)}{m}] \geq 1 - \delta \tag{1.10}$$

This constitutes the first learning guarantee that we have derived. Note that in our derivation we did not pay much attention to the implementation or procedure of the CLR, which will depend on H. We also used the realizability assumption, which has some implications:

- What if there is *no* predictor $h \in H$ s.t. $L_D(h) = 0$?
- In such a case, can we use with $\min_{h\in H} L_D(h)$?

So the next learning rule we will study is **empirical risk minimization** where:

$$\text{ERM}_H(S) = \arg\min_{h\in H} \frac{1}{|S|} \sum_{(x,y)\in S} \mathbf{1}\{h(x_i) \neq y_i\} \tag{1.11}$$

You might be wondering if we can use our previous Hoeffding bound:

$$\mathbb{P}_{S\sim D^m}[|L_S(h) - L_D(h)| \leq \sqrt{\frac{\ln(2/\delta)}{2m}}] \geq 1 - \delta \tag{1.12}$$

The answer is no. This is because that bound operates on a fixed h seen *a priori* before our sampled data, whereas $\hat{h}$ is a function of the data (e.g. $\hat{h}$ is a random variable) and so the inequality does not apply. Furthermore, while the set of cases where $|L_S(h) - L_D(H)| > \sqrt{\frac{\ln(2/\delta)}{2m}}$ may only have δ probability w.r.t. $S \sim D^m$, it is in these sets of cases where $L_S(h)$ is much lower than $L_D(h)$. This is the problem of *overfitting*.