CIS5200: Machine Learning Spring 2023

Lecture 14 and 15: Learning Theory

Date: March 2 and 14, 2023 *Author: Surbhi Goel*

Acknowledgements. These notes are heavily inspired by notes by Rob Schapire (Princeton), Michael Kearns (UPenn), and Hamed Hassani (UPenn).

Disclaimer. These notes have not been subjected to the usual scrutiny reserved for formal publications. If you notice any typos or errors, please reach out to the author.

1 Generalization

So far, we have mostly focused on solving the empirical risk minimization problem for various models, that is, the problem of finding a minimizer of the loss over the training set amongst a function/hypothesis class. However, doing well on the training data may not suffice for doing well on unseen data that we may encounter when deploying our models. For example, consider a rather silly predictor that memorizes the entire training set, that is, outputs the correct value, but predicts 0 everywhere else. This silly predictor gets 0 training error or empirical risk, however, we do not expect this to do well in practice. Therefore, what we want from our learned model is that it *generalize* well, that is, it has small true risk over the underlying distribution that the training data is drawn from.

More formally, for a fixed hypothesis/function class $\mathcal{F}$, let $\hat{f}$ be the empirical risk minimizer with respect to loss ℓ over training data $\mathcal{S} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, that is,

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \underbrace{\frac{1}{m} \sum_{i=1}^{m} \ell(f(x_i), y_i)}_{\text{Empirical Risk } \hat{R}(f)}.$$

Then the risk of the predictor $\hat{f}$ (generalization error) on unseen examples from the underlying distribution $\mathcal{D}$ is as follows,

$$\underbrace{R(f)}_{\text{True Risk}} = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell(f(x), y)\right].$$

Using this we have,

$$R(\hat{f}) = \underbrace{(R(\hat{f}) - \hat{R}(\hat{f}))}_{\text{generalization gap}} + \hat{R}(\hat{f}).$$

ERM only guarantees that the latter quantity is small, and we need to ensure that the former (generalization gap) is also small in order to get small generalization error.

In this and the next lecture, we will quantify this gap in terms of the size of the training dataset and the complexity of the underlying function class $\mathcal{F}$.

2 PAC Learning

So our goal is to minimize the generalization error of the learned predictor from any given training dataset drawn from the underlying distribution. To find exactly the optimal classifier by observing only a finite training dataset is too strong of an expectation. Furthermore, finding such a predictor for every possible training dataset is also too strong of an expectation. [1]

In practice, we would be happy with an *approximately* good solution that can be found with *high probability* from a large enough finite training set. This idea was captured beautifully in the PAC (Probably Approximately Correct) model introduced by Leslie Valiant in 1984. We will focus on the 0/1 loss for the rest of the lecture.

Definition 1. *A function class $\mathcal{F}$ is PAC learnable if there exists an algorithm $\mathcal{A}$ and a function $m_{\mathcal{F}} : (0,1)^2 \to \mathbb{N}$ with the following property: such for every labelling function $f \in \mathcal{F}$, for every distribution $\mathcal{D}$ on feature space $\mathcal{X}$[2], and for all $\epsilon, \delta \in (0,1)$, if $\mathcal{A}$ is given access to a training dataset S of size $m \geq m_{\mathcal{F}}(\epsilon, \delta)$ where the features are drawn from $\mathcal{D}$ and labels are according to f, then with probability $1-\delta$ (over the choice of the training dataset) algorithm $\mathcal{A}$ outputs a predictor $\hat{f}$ such that $\mathcal{R}(\hat{f}) \leq \epsilon$, that is,*

$$\Pr_{x \sim \mathcal{D}} \left[\hat{f}(x) \neq f(x) \right] \leq \epsilon.$$

Here ϵ is the *error* parameter and δ is the *confidence* parameter. Note that the predictor $\hat{f}$ is approximately correct with high probability and hence the name Probably Approximately Correct learning.

Sample Complexity The function $m_{\mathcal{F}} : (0,1)^2 \to \mathbb{N}$ determines the sample complexity of learning, that is, how large should your training dataset be in order to get generalization error at most ϵ with probability at least $1-\delta$. This quantity depends on the *complexity/size* of the underlying function class $\mathcal{F}$. Intuitively, the larger or more complex the function class, the more data you would need to learn it.[3]

3 Complexity of Function Class

We will now formalize what the complexity of a function class is. We will first look at finite function classes and then consider the more general class of infinite function classes which includes the models we have been looking at such as linear classifiers.

3.1 Finite Function Classes

Consider a finite function class $\mathcal{F} := \{f_1, \ldots, f_{|\mathcal{F}|}\}$, then a natural notion of complexity is the size of this function class $|\mathcal{F}|$. A natural question to ask is, a*re finite function classes PAC learnable?*

[1] Imagine a bad draw of the dataset that has very limited information about the true predictor.

[2] Note that here we defined $\mathcal{D}$ to be a distribution over $\mathcal{X}$ and not $\mathcal{X} \times Y$. This is because we are making a *realizability* assumption, where for any input x, the label y is determined deterministically as $f(x)$ for some $f \in \mathcal{F}$. Thus, there is no distribution over y. This model can be extended to handle non-realizable data.

[3] This intuition is not always correct!

And if so, then how does the sample complexity depend on $|\mathcal{F}|$?

Theorem 2. *Every finite function class $\mathcal{F}$ is PAC learnable with sample complexity*

$$m_{\mathcal{F}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{F}|/\delta)}{\epsilon} \right\rceil.$$

Proof. We will show that $\mathcal{A}$ being the ERM algorithm suffices. Let $\hat{f}_S$ be the hypothesis output by $\mathcal{A}$ on dataset S. Since we are assuming that there is a perfect classifier f_*, we have $\hat{f}_S(x_i) = f_*(x_i)$ for all $i \in [m]$. To show that $\mathcal{F}$ is PAC learnable up to error ϵ with probability $1 - \delta$, we need to upper bound the probability that $\hat{f}_S$ is bad ($R(\hat{f}_S) > \epsilon$) by δ, that is, $\Pr\left[\hat{f}_S \text{ is } \epsilon\text{-bad}\right] \leq \delta$.

Let $\mathcal{B} = \{f \in \mathcal{F} : f \text{ is } \epsilon\text{-bad}\}$. Note that $\mathcal{B}$ is not dependent on the training set we draw. Now we have

$$\begin{aligned}
\Pr_S\left[\hat{f}_S \text{ is } \epsilon\text{-bad}\right] &= \Pr_S\left[\hat{f}_S \text{ is } \epsilon\text{-bad} \wedge \hat{R}(\hat{f}_S) = 0\right] && \hat{f}_S \text{ is an ERM} \\
&\leq \Pr_S\left[\exists f \in \mathcal{F} : f \text{ is } \epsilon\text{-bad} \wedge \hat{R}(f) = 0\right] && \text{for events } A, B \text{ if } A \implies B \text{ then } \Pr[A] \leq \Pr[B] \\
&= \Pr_S\left[\exists f \in \mathcal{B} : \hat{R}(f) = 0\right] && \text{by definition of } \mathcal{B} \\
&\leq \sum_{f \in \mathcal{B}} \Pr_S\left[\hat{R}(f) = 0\right] && \text{by union bound} \\
&= \sum_{f \in \mathcal{B}} \Pr_S\left[\forall i \in [m] : f(x_i) = f_*(x_i)\right] \\
&= \sum_{f \in \mathcal{B}} \prod_{i=1}^{m} \Pr_S\left[f(x_i) = f_*(x_i)\right] && \text{samples are i.i.d.} \\
&= \sum_{f \in \mathcal{B}} \prod_{i=1}^{m} \left(1 - \Pr_S\left[f(x_i) \neq f_*(x_i)\right]\right) \\
&= \sum_{f \in \mathcal{B}} \prod_{i=1}^{m} (1 - R(f)) && \Pr\left[f(x) \neq f_*(x)\right] = R(f) \\
&\leq \sum_{f \in \mathcal{B}} (1 - \epsilon)^m && f \text{ is } \epsilon\text{-bad} \\
&= |\mathcal{B}|\,(1 - \epsilon)^m \\
&\leq |\mathcal{F}|\,(1 - \epsilon)^m && \mathcal{B} \subseteq \mathcal{F} \\
&\leq |\mathcal{F}| \exp(-m\epsilon) && 1 - a \leq \exp(-a)
\end{aligned}$$

Union bound: For events A, B, $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Since we want the above quantity to be $\leq \delta$ in order to get our PAC guarantee, we get,

$$|\mathcal{F}| \exp(-m\epsilon) \leq \delta \implies m \geq \frac{\log(|\mathcal{F}|/\delta)}{\epsilon}.$$

This concludes our proof. □

Another way to read this is,

$$R(\hat{f}_S) \leq \frac{\log(|\mathcal{F}|/\delta)}{m}$$

Things to note here:

- As the function class becomes larger, the upper bound becomes larger. The more the number of possible rules, the more challenging it is to generalize.
- The more data we have in the training dataset, the smaller our upper bound gets.
- For higher probability of success, we need more data to get the same error guarantee.

3.2 Infinite Function Classes

It is not immediately clear how to characterize the complexity of an infinite function class. However, one important consequence of PAC-learnability is that it is sufficient to just work with finite datasets and observe the behavior of the function class only on the training dataset.

We can define the behavior of a function class on a training set S as $\Pi_{\mathcal{F}}(S) = \{(f(x_1), \ldots, f(x_m)) : f \in \mathcal{F}\}$, that is, all possible labellings that any function in the class can generate. Define $\Pi_{\mathcal{F}}(m) = \max_{S;|S|=m} |\Pi_{\mathcal{F}}(S)|$ to be the maximum possible labellings over all possible training sets. this is also known as the growth function for function class $\mathcal{F}$. Note that $\Pi_{\mathcal{F}}(m) \leq 2^m$ since the total number of possible labellings of m examples is 2^m.

We can use $\Pi_{\mathcal{F}}(m)$ as a proxy of the size of the function class to get an analogous result,

Theorem 3. *With probability $1 - \delta$, any predictor $f \in \mathcal{F}$ with $\hat{\mathcal{R}}(f) = 0$ for training set size m satisfies*

$$\mathcal{R}(f) \leq \left\lceil \frac{\log(|\Pi_{\mathcal{F}}(2m)|/\delta)}{m} \right\rceil.$$

Proof. We will not cover the proof in class. The proof uses some cool symmetrization ideas. If you are interested, the full proof can be found in the following notes: the proof starts in Section 3 and is continued in Section 1. □

Example 1 (thresholds): Let us consider the function class of one-dimensional thresholds (see Figure 1) $\mathcal{F} = \{f_a : a \in \mathbb{R}\}$ where

$$f_a(x) = \begin{cases} 1 & \text{if } x \geq a \\ -1 & \text{otherwise.} \end{cases}$$

Consider a dataset of 3 points $x_1 < x_2 < x_3$. There are 4 possible labellings :

- $(-1, -1, -1)$ with f_a such that $a > x_3$
- $(-1, -1, 1)$ with f_a such that $x_2 < a \leq x_3$
- $(-1, 1, 1)$ with f_a such that $x_1 < a \leq x_2$

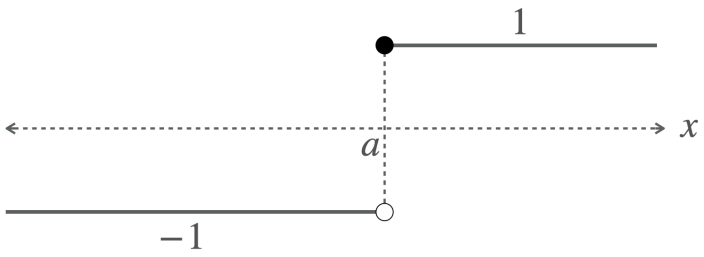

Figure 1: One-dimensional threshold $f_a(x) = \begin{cases} 1 & \text{if } x \geq a \\ -1 & \text{otherwise.} \end{cases}$

- $(1, 1, 1)$ with f_a such that $a \leq x_1$

The other 4 labellings, $(-1, 1, -1), (1, -1, 1)$, $(1, 1, -1), (1, -1, -1)$, are not attainable, since the functions in $\mathcal{F}$ can have only one point of change from -1 to 1 as we go from smaller to larger values of x. In general, for m datapoints, the same logic gives us, $\Pi_{\mathcal{F}}(m) = m + 1$ which is exponentially smaller than 2^m.

Example 2 (intervals): Let us consider the function class of intervals (ee Figure 2) $\mathcal{F} = \{f_{a,b} : a \leq b \in \mathbb{R}\}$ where

$$f_{a,b}(x) = \begin{cases} 1 & \text{if } a \leq x \leq b \\ -1 & \text{otherwise.} \end{cases}$$

Consider again the dataset of 3 points $x_1 < x_2 < x_3$. There are now 7 possible labellings :

- $(-1, -1, -1)$ with $f_{a,b}$ such that $a, b > x_3$
- $(-1, -1, 1)$ with $f_{a,b}$ such that $x_2 < a \leq x_3$ and $x_3 \leq b$
- $(-1, 1, 1)$ with $f_{a,b}$ such that $x_1 < a \leq x_2$ and $x_3 \leq b$
- $(1, 1, -1)$ with $f_{a,b}$ such that $a \leq x_1$ and $x_2 \leq b < x_3$
- $(1, -1, -1)$ with $f_{a,b}$ such that $a \leq x_1$ and $x_1 \leq b < x_2$
- $(1, 1, 1)$ with $f_{a,b}$ such that $a \leq x_1$ and $x_3 \leq b$
- $(-1, 1, -1)$ with $f_{a,b}$ such that $x_1 < a \leq x_2$ and $x_2 \leq b < x_3$

The labelling $(1, -1, 1)$ is not attainable, since the functions in $\mathcal{F}$ has 1s in a contiguous block. In general, for m datapoints, $\Pi_{\mathcal{F}}(m) = \frac{m(m+1)}{2} + 1$. *Try to prove this, think about how many options do you have for a, b.* Note that this is $O(m^2)$ compared to $O(m)$ in the threshold case.

4 VC Dimension

In order to define VC dimension, we will first define the notion of shattering.

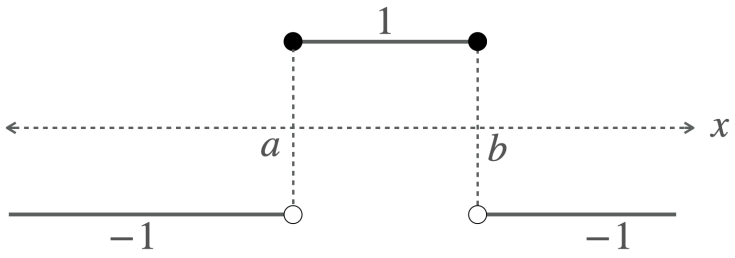

Figure 2: Intervals $f_{a,b}(x) = \begin{cases} 1 & \text{if } a \leq x \leq b \\ -1 & \text{otherwise.} \end{cases}$

Definition 4 (shattered set)**.** *A set S of inputs is said to be shattered by function class $\mathcal{F}$ if $|\Pi_{\mathcal{F}}(S)| = 2^{|S|}$, that is, $\mathcal{F}$ can realize all possible labellings for the set of points in S.*

Definition 5 (VC dimension)**.** *VC dimension of a function class $\mathcal{F}$ is the size of the largest set S that can be shattered by $\mathcal{F}$.*

In order to show a VC dimension bound of d for a function class $\mathcal{F}$, we need to do the following 2 steps:

- Show that $VC(\mathcal{F}) \geq d$ by giving an explicit set of d points that are shattered by $\mathcal{F}$.
- Show that $VC(\mathcal{F}) \leq d$ by proving that no set of $d+1$ points can by shattered by $\mathcal{F}$.

Together these imply that $VC(\mathcal{F}) = d$. Let us look at few examples:

Example 1 (thresholds): Consider the class of 1-dimensional thresholds. Any set of size 1 can be easily shattered. However, no set of size 2 can be shattered. Consider any set of size 2 $\{x_1, x_2\}$. Assume WLOG that $x_1 \leq x_2$. Then $(1, -1)$ is not attainable since any function in the class is non-decreasing with respect to the input. Therefore, the VC dimension is 1.

Example 2 (intervals): Consider the class of intervals. We previously showed that for any 3 points $x_1 \leq x_2 \leq x_3$, the labelling $(1, -1, 1)$ is not possible since the positives have to be contiguous. However, any $x_1 < x_2$ can be shattered. *You can check this by finding a, b that give all possible labellings.* Thus, the VC dimension of intervals is 2.

Example 3 (axis-aligned rectangles): Consider the class of axis-aligned rectangles in $\mathbb{R}^2$, where everything inside the rectangle is labelled 1 and everything outside is -1. Figure 3 shows the existence of 4 points that can be shattered by this class. However, no 5 points can be shattered by this function class. In order to prove this, consider any set of 5 points and let x_L, x_R, x_T, x_B be the leftmost, rightmost, topmost, and bottom most points respectively from the set. Now observe that the the remaining 5th point must lie within the rectangle formed by these 4 points. Therefore, the labelling of x_L, x_R, x_T, x_B being 1 and the remaining 5th point being -1 is not possible.

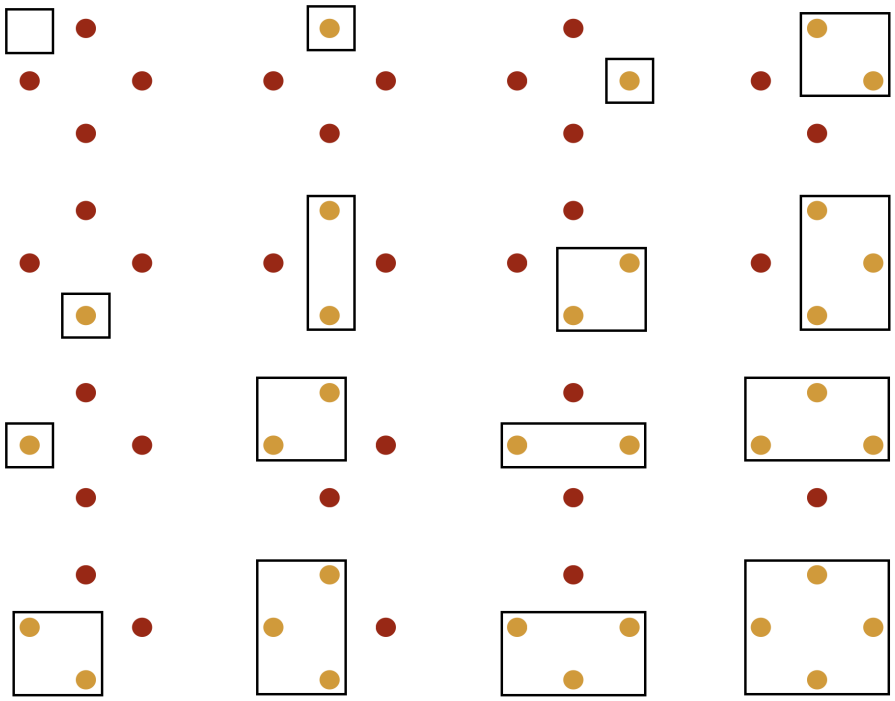

Figure 3: A set of 4 points that are shattered by the class of axis-aligned rectangles in $\mathbb{R}^2$. Here yellow indicates label 1 and red indicates label -1. Figure idea borrowed from Rob Schapire's notes.

Example 4 (halfspaces): The VC dimension of linear classifiers (halfspaces) $\mathsf{sign}(w^\top x + b)$ is $d + 1$. The VC dimesnion of linear classifiers with margin γ and inputs bounded by norm 1 is $O(1/\gamma^2)$.

An amazing theorem by Sauer relates VC dimension to $\Pi_{\mathcal{F}}(m)$.

Theorem 6 (Sauer's Theorem)**.** *Let $d = VC(\mathcal{F})$. Then for any $m > d$,*

$$\Pi_{\mathcal{F}}(m) = O(m^d).$$

Using this in Theorem 3, we get that for $m > d$, for any ERM f, $R(f) \leq \left\lceil \frac{\log(|\Pi_{\mathcal{F}}(2m)|/\delta)}{m} \right\rceil \lesssim \frac{d \log(m/\delta)}{m}$.[4] This can be improved to get the following fundamental result.

Theorem 7 (Fundamental Theorem of Learning)**.** *With probability $1-\delta$, any predictor $f \in \mathcal{F}$ with $\hat{\mathcal{R}}(f) = 0$ for training set size m satisfies*

$$\mathcal{R}(f) \lesssim \frac{d + \log(1/\delta)}{m}.$$

More generally, when there is no perfect classifier, we can still get an (agnostic) PAC learning guarantee.

[4] $a \lesssim b$ means that there exists some constant $c > 0$ such that $a \leq cb$.

Theorem 8 (Agnostic PAC Learning)**.** *With probability* $1-\delta$*, for any ERM* $\hat{f} \in \arg\min_f \hat{R}(f)$ *over training set size* m *satisfies*

$$\mathcal{R}(\hat{f}) - \min_f R(f) \lesssim \sqrt{\frac{d + \log(1/\delta)}{m}}.$$

ERM solution gets true risk close to the best possible true risk attainable by any function in the function class even when data is not perfectly separable by the function class.