CIS5200: Machine Learning Spring 2023

Lecture 16: Learning Theory - 3

Date: March 16, 2023 *Author: Surbhi Goel*

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

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 VC Dimension

Last class we studied the definition of VC dimension.

Definition 1 (VC dimension)**.** *VC dimension of a function class $\mathcal{F}$ is the size of the largest set S that can be shattered by $\mathcal{F}$. Here, we say that a set S is shattered by $\mathcal{F}$ if $\Pi_{\mathcal{F}}(S) = 2^{|S|}$.*

We saw that the Sauer's Lemma relates VC dimension to $\Pi_{\mathcal{F}}(m)$ and we get that for $m > d$, for any ERM f,

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

This can be improved to:

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

1.1 VC Classes

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 $VC(\mathcal{F}) = d$.

In the last class we looked at the examples of thresholds, intervals, and rectangles. Let us look at the class of linear classifiers.

Example 4 (linear classifiers/halfspaces): The VC dimension of linear classifiers (halfspaces) $\mathcal{F} = \{x \mapsto \mathsf{sign}(w^\top x) : w \in \mathbb{R}^d\}$ is d. If we allow for the bias term, then the VC dimension is $d+1$.

In order to prove this, let us first describe a set S of d points that is shattered by the class. Consider the set of points $x_i = e_i$ for $i \in [d]$ where e_i is the ith standard basis that has 1 at coordinate i and 0 everywhere else. In order to show that these points can be shattered, for all labeling $y_1, \ldots, y_d \in \{-1, 1\}$ we need to show the existence of $f \in \mathcal{F}$ that realizes it. Consider labeling $y_1, \ldots, y_d \in \{-1, 1\}$, then choosing w as below suffices.

$$w = \sum_{i=1}^{m} y_i e_i$$

Then we have, that for all $i \in [m]$, $\mathsf{sign}(w^\top x_i) = \mathsf{sign}(y_i) = y_i$. Thus it generates the labeling $y_1, \ldots, y_m$. Since we can do this for any labeling, these points can be shattered.

Now we need to show that no $d+1$ points can be shattered. In order to show this, let us consider any set of $d+1$ points $x_1, \ldots, x_{d+1}$. We know that no set of $d+1$ d-dimensional vectors can be linearly independent, thus there exists some $j \in [d+1]$ such that

$$x_j = \sum_{i \neq j} \alpha_i x_i,$$

such that at least one $\alpha_i \neq 0$. Suppose we consider the labeling where $y_j = -1$ and for all $i \neq j$, $y_i = \text{sign}(\alpha_i)$ if $\alpha_i \neq 0$ else $y_i = 1$. We will show that no w can achieve this labeling. Suppose there is a w that achieves this labeling then we have for all $i \neq j$, if $\alpha_i \neq 0$ then $\alpha_i(w^\top x_i) > 0$ since $\text{sign}(w^\top x_i) = y_i = \text{sign}(\alpha_i)$. This gives us,

$$w^\top x_j = \sum_{i \neq j} \alpha_i w^\top x_i > 0.$$

Thus w would label x_j incorrectly as positive when $y_j = -1$. This proves that no $d+1$ points can be shattered.

2 Uniform Convergence

The VC dimension actually characterizes a stronger property of generalization known as uniform convergence. This property becomes important when we move away from realizability.

Definition 2 (Uniform Convergence)**.** *A function class $\mathcal{F}$ has the uniform convergence property if for any distribution $\mathcal{D}$ over the input space and for any $\epsilon, \delta > 0$, there exists a sample size m such that for any training set of m samples S drawn i.i.d. from $\mathcal{D}$, with probability at least $1 - \delta$, the following holds for all $f \in \mathcal{F}$:*

$$\left|\mathcal{R}(f) - \hat{\mathcal{R}}(f)\right| \leq \epsilon.$$

For classes with bounded VC dimension, we have

Theorem 3 (Uniform convergence for VC classes)**.** *Let $\mathcal{F}$ be a function class with VC dimension d, then with probability $1 - \delta$ over the draw of a training set of size m, for all $f \in \mathcal{F}$*

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

We will not prove the above, but we will prove a similar statement for finite classes.

Theorem 4 (Uniform convergence for finite classes)**.** *Let $\mathcal{F}$ be a finite function class, then with probability $1-\delta$ over the draw of a training set of size m, for all $f \in \mathcal{F}$*

$$\left|\mathcal{R}(f) - \hat{\mathcal{R}}(f)\right| \lesssim \sqrt{\frac{\log|\mathcal{F}| + \log(1/\delta)}{m}}.$$

Proof. We will prove this theorem using the union bound and Hoeffding's inequality[1].

Consider a $f \in \mathcal{F}$, and let Z_i be the loss of f on the i-th example. We have:

$$\hat{\mathcal{R}}(f) = \frac{1}{m}\sum_{i=1}^{m} Z_i.$$

By definition, the true risk $\mathcal{R}(f)$ is the expected value of the loss of for any example i:

$$\mathcal{R}(f) = \mathbb{E}[Z_i].$$

Since the Z_i are i.i.d., we can apply Hoeffding's inequality to bound the probability of the difference between the empirical risk and the true risk being large:

$$\Pr\left[|\hat{\mathcal{R}}(f) - \mathcal{R}(f)| \geq \epsilon\right] \leq 2\exp\left(-2m\epsilon^2\right).$$

This was for a single f. Now, we want it to hold for all $f \in \mathcal{F}$. To do this, we apply the union bound:

$$\Pr\left[\bigcup_{f\in\mathcal{F}}\left\{|\hat{\mathcal{R}}(f) - \mathcal{R}(f)| \geq \epsilon\right\}\right] \leq \sum_{f\in\mathcal{F}}\Pr\left[|\hat{\mathcal{R}}(f) - \mathcal{R}(f)| \geq \epsilon\right] \leq 2|\mathcal{F}|\exp\left(-2m\epsilon^2\right).$$

We want this probability to be at most δ, so we set:

$$2|\mathcal{F}|\exp\left(-2m\epsilon^2\right) \leq \delta.$$

Solving for ϵ, gives us the result. □

3 Agnostic Learning

Now using uniform convergence, we can show a PAC guarantee when there is no perfect classifier.

Theorem 5 (Agnostic PAC Learning)**.** *With probability $1-\delta$, for any ERM $\hat{f} \in \arg\min_{f\in\mathcal{F}}\hat{\mathcal{R}}(f)$ over training set size m, we have,*

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

[1]The form of Hoeffding's inequality we use can be stated using coin tosses as: Consider a coin with bias p flipped m times. Let X be the number of times the coin showed up as heads, then $\Pr\left[\left|\frac{X}{m} - p\right| > \epsilon\right] \leq 2\exp(-2m\epsilon^2)$.We will not cover the proof of the Hoeffding's inequality but use it as a tool. You can read more about it here

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.

Proof. First, let $f^* \in \arg\min_{f \in \mathcal{F}} \mathcal{R}(f)$ be the function with the smallest true risk in the function class $\mathcal{F}$. Also, consider the ERM hypothesis $\hat{f} \in \arg\min_{f \in \mathcal{F}} \hat{R}(f)$. By definition of ERM, we have that

$$\hat{\mathcal{R}}(\hat{f}) \leq \hat{\mathcal{R}}(f^*).$$

From the uniform convergence theorem (Theorem 3) for VC classes, we have that with probability at least $1 - \delta$, for all $f \in \mathcal{F}$:

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

This implies that with probability $1 - \delta$,

$$\mathcal{R}(\hat{f}) - \hat{\mathcal{R}}(\hat{f}) \lesssim \sqrt{\frac{d + \log(2/\delta)}{m}}$$

and

$$\hat{\mathcal{R}}(f^*) - \mathcal{R}(f^*) \lesssim \sqrt{\frac{d + \log(2/\delta)}{m}}.$$

By adding the above and using $\hat{\mathcal{R}}(\hat{f}) \leq \hat{\mathcal{R}}(f^*)$, we get with probability $1 - \delta$,

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f^*) = \mathcal{R}(\hat{f}) - \min_{f \in \mathcal{F}} \mathcal{R}(f) \lesssim \sqrt{\frac{d + \log(1/\delta)}{m}}.$$

□

4 Bias/Variance

We can look at the error sources in the previous bound as *bias* and *variance*.

$$\mathcal{R}(\hat{f}) \leq \underbrace{\min_{f \in \mathcal{F}} \mathcal{R}(f)}_{\text{bias}} + \underbrace{O\left(\sqrt{\frac{d + \log(1/\delta)}{m}}\right)}_{\text{variance}}.$$

The bias is the error that is due to the model itself, that is, the choice of the function class even if given infinite training data. The variance is the error that is due to the finite amount of data used, that is, the error induced by different choices of the finite training data. See Figure 1 for a more intuitive visualization that captures the notion of bias and variance.

Keeping the training dataset size fixed, as we increase the model/function class complexity, the bias reduces as we are able to approximate the label better whereas the variance increases since the model has more capacity to fit to the noise in the training dataset. See Figure 2 for a graphical sketch of this. This suggests that there is a sweet spot of model complexity which trades off the two errors to get the best possible total error.

Furthermore, in the regime of high bias and low variance, we are *underfitting*, that is, our function class is not good enough to fit the labels. On the other hand, in the regime of low bias and high

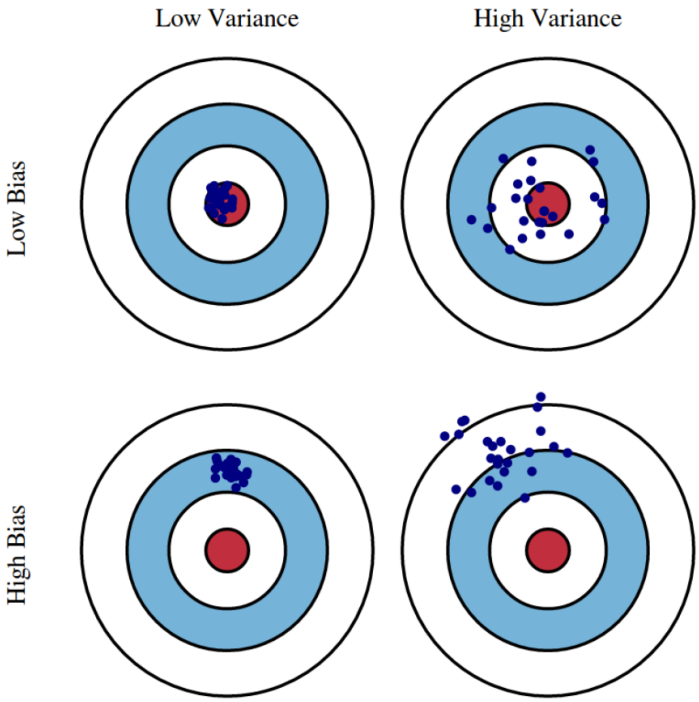

Figure 1: Bias and variance visualization. Here the center of the target achieves the lowest possible error with the error increasing as we go outside. Each point represents a new training data and its corresponding error If the points are off the center, then the model has high bias. If they are more spread out, then it has high variance. Src: http://scott.fortmann-roe.com/docs/BiasVariance.html

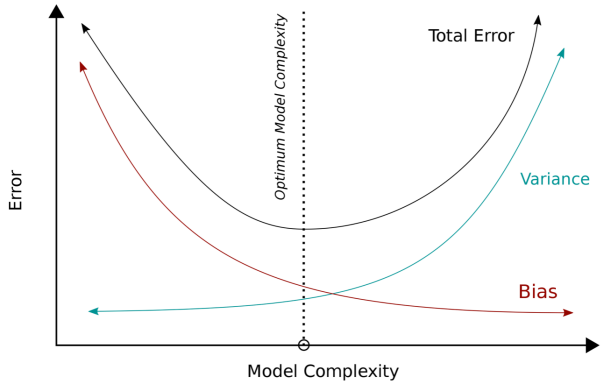

Figure 2: Bias-variance tradeoffs for fixed training set size m. Src: https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

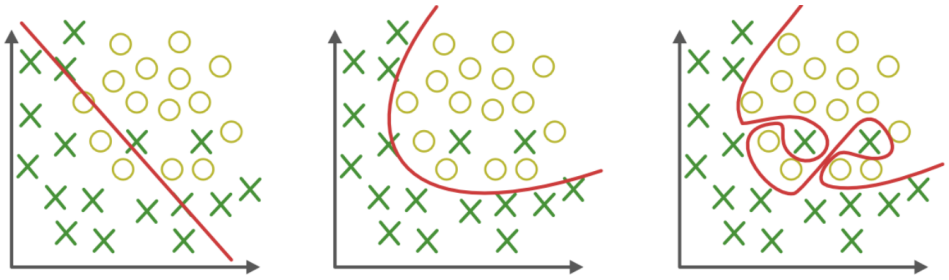

Figure 3: Underfitting and overfitting models. Src: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

variance, we are *overfitting*, that is, our function class is good enough for our task but the model is learning a more training set dependent classifier. See Figure **??** for an example of the two regimes.

Now we can decide what to do based on which regime we are in:

If we have **high variance**, that is training error is close to 0 but test loss is high, then we can remedy this by increasing the training set size, or reducing the complexity of the function class. We can also use a procedure called bagging to reduce variance that we will discuss in a future lecture.

If we have **high bias**, that is, the training error is high, then we can remedy this by increasing the complexity of our dataset, or making the feature space richer. We can also use a procedure called boosting, which we will talk about in a future lecture.