CS269: Machine Learning Theory
Lecture 5: Infinite Function Classes, Part 2

October 11, 2010

Lecturer: Jennifer Wortman Vaughan
Scribe: Ertan Dogrultan and Paul Wais

1 VC Dimension Upper Bound on Error and Using Sauer's Lemma

In the the last lecture, we defined the Behavior Set and Growth Function of $\mathcal{H}$, which are two useful metrics for problem complexity in the case that $\mathcal{H}$ is infinite. Consider an arbitrary vector of input points $S = \langle x_1, \ldots, x_m \rangle$ of size m.

Definition 1. *Behavior Set*

$$\Pi_{\mathcal{H}}(S) = \{\langle h(x_1), \ldots, h(x_m) \rangle : h \in \mathcal{H}\}$$

Definition 2. *Growth Function*

$$\Pi_{\mathcal{H}}(m) = \max_{S:|S|=m} |\Pi_{\mathcal{H}}(S)|$$

Since there are 2^m distinct ways to assign binary labels to m points, $\Pi_{\mathcal{H}}(m) \leq 2^m$. This bound appears to imply that some problems may be *unlearnable* since $\Pi_{\mathcal{H}}(m)$ is so large. However, as we discussed in the last lecture, there are cases where $\Pi_{\mathcal{H}}(m)$ is not exponential. In particular, Sauer's Lemma shows that if the VC Dimension of $\mathcal{H}$ is finite, then $\Pi_{\mathcal{H}}(m)$ will be polynomial in m. Let us recall the definition of VC Dimension and Sauer's Lemma.

Definition 3. *The VC Dimension of $\mathcal{H}$ is the size of the largest set S that $\mathcal{H}$ shatters.*

Phrased differently: The VC Dimension of $\mathcal{H}$ is the largest set of points such that for any labeling of these points, $\exists h \in \mathcal{H}$ where h achieves the labeling. [1]

Lemma 1. *Sauer's Lemma.*[2] *$\forall \mathcal{H}$ with finite VC Dimension d,*

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} = O(m^d)$$

Together, Sauer's Lemma and the following theorem imply that $\mathcal{H}$ is *learnable* if $\mathcal{H}$ has finite VC Dimension (and learnable with a polynomial number of examples if the VC Dimension is polynomial).

[1] See also Section 3.2 of "An Introduction to Computational Learning Theory" by Kearns and Vazirani.

[2] For a proof, see Section 3.4 of Kearns and Vazirani or notes from Lecture #7 of Rob Schapire's class available online at: `http://www.cs.princeton.edu/courses/archive/spring08/cos511/scribe_notes/0225.pdf`

Theorem 1. *Given any concept class $\mathcal{C}$ and hypothesis class $\mathcal{H}$, and given any m examples drawn i.i.d. from any distribution D and labeled by C, consider an algorithm A that returns a hypothesis $h \in \mathcal{H}$ consistent with the labeled examples. Then, for any $\delta \in (0,1)$, with probability of at least $1-\delta$,*

$$\mathrm{err}(h) \leq 2\frac{\log \Pi_{\mathcal{H}}(2m) + \log(\frac{2}{\delta})}{m}$$

We may apply Sauer's Lemma to Theorem 1 to achieve tighter error bounds in the case that the VC Dimension of $\mathcal{H}$ is finite.

1.1 Applying Sauer's Lemma to Theorem 1

Using Sauer's Lemma and Theorem 1, we obtain that under the same conditions in which Theorem 1 holds, with probability of at least $1-\delta$,

$$\mathrm{err}(h) \leq 2\frac{d\log(\frac{2em}{d}) + \log\frac{2}{\delta}}{m}$$

Thus, for a function class with finite VC Dimension, error will scale about linearly with dimension. Observe that one can achieve $\mathrm{err}(h) \leq \epsilon$ with number of examples

$$m = O\left(\frac{1}{\epsilon}\log\frac{1}{\delta} + \frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$$

Thus the number of examples needed to achieve $\mathrm{err}(h) \leq \epsilon$ scales linearly with the VC dimension d of the function class. Intuitively, this result indicates that if one were to add more features to a model (and thus increase d), the number of training examples m needed to achieve less than ϵ error scales linearly with d; each dimension of the model requires at most a constant number of examples to learn. Let's digest the significance of this conclusion in a couple of concrete examples.

Case: 1-dimensional threshold functions

Recall from the last lecture that the VC Dimension of 1-dimensional threshold functions is $d = 1$. [3] Using Sauer's Lemma, we can show that for this class $\mathcal{C}$, with probability at least $1-\delta$,

$$\begin{aligned}\mathrm{err}(h) &\leq 2\frac{d\log(\frac{2em}{d}) + \log\frac{2}{\delta}}{m} \\ &\leq 2\frac{\log(2em) + \log\frac{2}{\delta}}{m}\end{aligned}$$

Recall that in Lecture 2, we used a class-specific argument using the structure of $\mathcal{C}$ to prove that

$$\mathrm{err}(h) \leq \tfrac{1}{m}\ln(\tfrac{2}{\delta})$$

Observe that the bound we achieve using VC Dimension theory is not as tight but still has about the same dependence on m and δ as the the bound achieved using the class-specific argument. This simple example serves as a sanity check of our application of Sauer's Lemma and ensures that VC Dimension theory yields a useful bound in this simple case.

[3] If we require that a 1-dimensional threshold function with threshold t assign all x to label 1 if $x > t$, then we cannot shatter two points if for some $x_1 < x_2$, since there is no c for which $c(x_1) = 1$ but $c(x_2) = 0$.

Case: n-dimensional linear separators
Recall that in the last lecture we showed that for $n \geq 2$ dimensions, the VC Dimension of linear separators with arbitrary thresholds is $d = n + 1$. Using Sauer's Lemma, we can use a similar argument to the one above to show that

$$\mathrm{err}(h) \leq 2\frac{(n+1)\log(\frac{2em}{n+1}) + \log\frac{2}{\delta}}{m}$$

Therefore, in this case $\mathrm{err}(h)$ again grows linearly with the number of features n of our model.

1.2 The Unrealizable Setting

So far, we have focused on learning in the realizable setting. We will mention briefly that the results we have observed may be extended to the unrealizable setting.

Recall in Lecture 2 that we derived the General Learning Bound[4] in the case of a *finite* concept class $\mathcal{C}$ in the unrealizable setting. This bound shows that if $\hat{h}$ is the hypothesis function that minimizes the empirical error, then with probability at least $1 - \delta$,

$$\mathrm{err}(\hat{h}) \leq \min_{h \in \mathcal{H}} \mathrm{err}(h) + O\left(\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{m}}\right)$$

So far in this lecture, we have discussed the case of learning an *infinite* function class $\mathcal{H}$ in the *realizable* setting and have shown that learning is feasible when $\mathcal{H}$ has finite VC Dimension. It is possible to use an argument similar to the one given in Lecture 2 to derive uniform convergence bounds on $|\mathrm{err}(h) - \widehat{\mathrm{err}}(h)|$ (where $\widehat{\mathrm{err}}(h)$ is the empirical error) in terms of the VC Dimension d of an infinite hypothesis class $\mathcal{H}$. We will not provide this derivation here, but will state the interesting result.[5] Using Sauer's Lemma, it is possible to show, with probability at least $1 - \delta$,

$$\mathrm{err}(\hat{h}) \leq \min_{h \in \mathcal{H}} \mathrm{err}(h) + O\Big(\sqrt{\tfrac{d}{m}\log(\tfrac{m}{d}) + \tfrac{1}{m}\log(\tfrac{1}{\delta})}\Big)$$

Notice that this bound exhibits an Occam's razor tradeoff in the VC Dimension of the hypothesis class d. On one hand, increasing the dimension of our hypothesis class may decrease the minimum error $\min_{h\in\mathcal{H}} \mathrm{err}(h)$; on the other hand, the second term in the bound on $\mathrm{err}(\hat{h})$ will grow in d.

2 VC Dimension Lower Bound on Error

So far, we've proved an upper bound on the error rate based upon the VC Dimension of the target function class. In this section, we prove a *lower* bound on error rate. We will prove the following theorem:

Theorem 2. *Fix a concept class $\mathcal{C}$ to have VC Dimension d. For any learning algorithm A, $\exists c \in \mathcal{C}$ and $\exists$ distribution D such that if A receives $m \leq \frac{d}{2}$ examples sampled i.i.d. from D labeled by c and A computes hypothesis h, then*

$$Pr\big[\mathrm{err}(h) > \tfrac{1}{8}\big] > \tfrac{1}{8}$$

[4] See section 2 of Lecture 2 scribe notes

[5] For the especially curious, consult the proof of Theorem 4.2 (as well as most of Chapter 4) in "Neural Network Learning: Theoretical Foundations" by Anthony and Bartlett to see how this is derived.

Note that the lower bound in Theorem 2 holds for *any* algorithm A, whether the function h that A outputs is consistent or not. Thus Theorem 2 shows that no algorithm can achieve arbitrarily small error with arbitrarily high probability unless the algorithm is given a number of examples that is at least linear in the VC Dimension of $\mathcal{C}$.

Furthermore, note that this result also shows that it is impossible to PAC-learn $\mathcal{C}$ without a number of examples m that is at least linear in d. Recall that in order to PAC-learn a concept class, we must (by definition) be able to achieve the error bound $\Pr[\mathrm{err}(h) < \epsilon] \geq 1 - \delta$ for *any* $c \in \mathcal{C}$ and *any* distribution D. Since Theorem 2 shows that there exists $c \in \mathcal{C}$ and some distribution D such that arbitrarily low error is not possible, we cannot PAC-learn $\mathcal{C}$ with so few examples.

Preliminaries. In this section, we will define *marginalization* using the following notation:

$$\begin{aligned} \Pr_{x,y}[z(x,y)] &= \sum_y \Pr(y)\Pr_x(z(x,y) \mid y) \\ &= \mathbb{E}_y[\Pr_x(z(x,y) \mid y)] \end{aligned}$$

We now present the proof.

Proof of Theorem 2: First of all, we need to show that for any $\mathcal{C}$ and A, we can find a distribution D and target $c \in \mathcal{C}$ such that algorithm A will have bad performance. However, one particular c cannot be bad for all algorithms. For instance, for any c we can define an algorithm that always outputs $h(\vec{x}) = c(\vec{x})$. Therefore, we will pick c uniformly at random from a set $\mathcal{C}'$ of all the possible distinct labelings of the points.

By the definition of VC Dimension, $\exists$ a set of points $\vec{x_1}, \ldots, \vec{x_d}$ that $\mathcal{C}$ shatters. Let $\mathcal{C}' \subseteq \mathcal{C}$ be a set of 2^d functions that shatter these points (i.e. the functions in $\mathcal{C}'$ achieve every possible labeling of the sample points). Suppose we picked the samples from a "bad" distribution D that is uniform over $\vec{x_1}, \ldots, \vec{x_d}$. We want to show that we can find some c such that $\mathrm{err}(h) = \Pr_{\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ is large and independent of ϵ and δ.

We wish to answer the following question: if we choose a target c uniformly from $\mathcal{C}'$ and run A on a sample from the "bad" distribution D, and output a function h, what is the probability that h makes an error on a new point? Let us construct Scenario 1 that corresponds to the procedure proposed in this question. Unfortunately, we find that Scenario 1 is difficult to analyze. Thus we present a similar Scenario 2 that is (as we will see) easier to analyze yet is equivalent in the sense that $\Pr_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ is the same for Scenarios 1 and 2.

Scenario 1

- Choose c at random from $\mathcal{C}'$
- Choose a sample S, $|S| = m$ i.i.d from D
- Compute h by running A on S labeled by c

Scenario 2

- Choose a sample S, $|S| = m$ i.i.d. from D
- Assign random labels $c(x_i)$ to each $x_i \in S$

- Randomly pick labels $c(x_i)$ for all $x_i \notin S$ (Note that we will not send these labels to A, but we still need to define what they are so that the probability $\text{Pr}_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ will be well-defined.)
- Compute h by running A on sample S with the randomly generated labels

Consider again the probability $\text{Pr}_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ in each scenario. Since the functions in $\mathcal{C}'$ achieve every possibly labeling, there is no difference between choosing a labeling function uniformly at random from $\mathcal{C}'$ and assigning random labels to each example x_i. In both cases the random sample S is selected independently of the random target c, and labels assigned by c are drawn uniformly at random from $\mathcal{C}'$ (i.e. the set of all possible distinct labelings). Therefore, the quantity $\text{Pr}_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ is the same for both scenarios. If we bound this quantity in Scenario 2, the bound will hold for Scenario 1 too.

Let us consider the quantity $\text{Pr}_{c,s,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ in Scenario 2.

$$\begin{aligned}
\text{Pr}_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})] &\geq \text{Pr}_{c,S,\vec{x}}[\vec{x} \notin S \wedge h(\vec{x}) \neq c(\vec{x})] \\
&= \text{Pr}[x \notin S]\text{Pr}[h(\vec{x}) \neq c(\vec{x}) \mid x \notin S] \quad \text{(definition of conditional probability)} \\
&\geq \frac{1}{2}\text{Pr}[h(\vec{x}) \neq c(\vec{x}) \mid x \notin S] \quad \text{(sample has } m \leq \tfrac{d}{2} \text{ points; at most half of } S \text{ is in the sample)} \\
&\geq \frac{1}{2}\left(\frac{1}{2}\right) \quad (c(\vec{x}) \text{ is chosen uniformly at random, so Pr}[h(\vec{x}) \neq c(\vec{x}) = \tfrac{1}{2}) \\
&\geq \frac{1}{4}
\end{aligned}$$

As we argued above, this bound holds for Scenario 1, which is the scenario we really care about, too. Now, if we marginalize $\text{Pr}_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$ over c and use the result from above, we find that

$$\begin{aligned}
\text{Pr}_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})] &= \mathbb{E}_c[\text{Pr}_{S,\vec{x}}[h(\vec{x}) \neq c(\vec{x}) \mid c]] \\
&\geq \frac{1}{4}
\end{aligned}$$

Using the observation that $\mathbb{E}[x] \geq k$ implies that $\exists$ some value of $x \geq k$, we find that $\exists c$ where $\text{Pr}_{S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})] \geq \frac{1}{4}$. Now, marginalize over S and work towards obtaining a bound on the probability that $h(\vec{x}) \neq c(\vec{x})$:

$$\begin{aligned}
\text{Pr}_{S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})] &= \mathbb{E}_S[\text{Pr}_{\vec{x}}[h(\vec{x}) \neq c(\vec{x})]] \\
&= \mathbb{E}_S[\text{err}(h)]] \quad \text{(by definition of err}(h)) \\
&= \text{Pr}(\text{err}(h) > \tfrac{1}{8})\underbrace{\mathbb{E}_S[\text{err}(h) \mid \text{err}(h) > \tfrac{1}{8}]}_{\leq 1, \text{ because err}(h) \leq 1} + \\
&\qquad \underbrace{\text{Pr}(\text{err}(h) \leq \tfrac{1}{8})}_{\leq 1, \text{ because probabilities are} \leq 1} \quad \underbrace{\mathbb{E}_S[\text{err}(h) \mid \text{err}(h) \leq \tfrac{1}{8}]}_{\leq \frac{1}{8}, \text{ because err}(h) \leq \frac{1}{8}} \quad \text{(expand expectation)} \\
\text{Pr}_{S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})] &\leq \text{Pr}(\text{err}(h) > \tfrac{1}{8}) + \tfrac{1}{8} \\
\frac{1}{4} &\leq \text{Pr}(\text{err}(h) > \tfrac{1}{8}) + \tfrac{1}{8} \quad \text{(using the bound proved above)} \\
\frac{1}{8} &\leq \text{Pr}(\text{err}(h) > \tfrac{1}{8})
\end{aligned}$$

□

Remarks

During the second half of the class, we discussed Homework 1 as well as the Final Project Guidelines.