CS260: Machine Learning Theory
Lecture 6: VC Dimension Lower Bound
October 12, 2011

Lecturer: Jennifer Wortman Vaughan

1 A Lower Bound

In the last lecture, we discussed a result that shows that in the realizable (perfect target function) setting, if our algorithm is given m i.i.d. labeled examples and outputs a consistent function $h \in \mathcal{H}$, then with probability at least $1-\delta$, $\text{err}(h) \leq \epsilon$ for

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

where d is the VC dimension of the hypothesis class $\mathcal{H}$. This gives us an *upper bound* on sample complexity that is linear in the VC dimension. But is this linear dependence tight?

In this class, we will see that it is tight. In particular, we will prove a *lower bound* showing that at least $d/2$ examples are needed to guarantee error less than ϵ with probability $1-\delta$ for any $\epsilon, \delta < 1/8$. This shows that the linear dependence on VC dimension in the sample complexity is necessary.

(This lower bound can actually be modified to show that $\Omega(d/\epsilon)$ examples are needed, proving that the linear dependence on $1/\epsilon$ is necessary too; see Chapter 3.6 of Kearns and Vazirani for a rough sketch, but be careful – there is a bug in their use of Markov's inequality, so the argument doesn't quite work as presented!)

In this theorem, we consider the case in which $\mathcal{C} = \mathcal{H}$; that is, we are learning $\mathcal{C}$ by $\mathcal{C}$.

Theorem 1. *Fix an arbitrary concept class $\mathcal{C}$ with VC Dimension d. For any learning algorithm $\mathcal{A}$, $\exists c \in \mathcal{C}$ and $\exists \mathcal{D}$ such that if $\mathcal{A}$ receives $m \leq d/2$ examples sampled i.i.d. from $\mathcal{D}$ labeled by c and outputs a hypothesis h, then*

$$Pr\left(\text{err}(h) > \frac{1}{8}\right) > \frac{1}{8}.$$

Note that the lower bound in Theorem 2 holds for *any* algorithm $\mathcal{A}$, not only algorithms that output consistent functions. 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 $\mathcal{C}$. This in turn tells us that any class with infinite VC dimension is not PAC learnable; a class can only be PAC learnable (by our modified definition) if its VC dimension is at most polynomial in the size of the input.

We now present the proof.

Proof of Theorem 1: We need to show that for any $\mathcal{C}$ and $\mathcal{A}$, there exists a distribution $\mathcal{D}$ and target $c \in \mathcal{C}$ for which $\mathcal{A}$ has bad performance. We start by constructing $\mathcal{D}$.

All CS260 lecture notes build on the scribes' notes written by UCLA students in the Fall 2010 offering of this course. Although they have been carefully reviewed, it is entirely possible that some of them contain errors. If you spot an error, please email Jenn.

By the definition of VC Dimension, there must exist a set of points $\{\vec{x}_1, \ldots, \vec{x}_d\}$ that $\mathcal{C}$ shatters. Let $\mathcal{D}$ assign weight $1/d$ to each of these points, and weight 0 to all other points.

Now, let $\mathcal{C}' \subseteq \mathcal{C}$ be a set of 2^d functions that shatter these points. For every possible labeling of the d points, there is exactly one function in $\mathcal{C}'$ that achieves the labeling. We will eventually want to show that there exists a particular $c \in \mathcal{C}'$ that is "bad" for $\mathcal{A}$, but for now, let's suppose that we choose the target function c uniformly at random from these 2^d functions. This is equivalent to flipping a fair coin to determine the label of each of the d points.

Consider the following question: if we choose a target c uniformly from $\mathcal{C}'$, run $\mathcal{A}$ on a sample of m points S drawn i.i.d. from the "bad" distribution $\mathcal{D}$, and output a function h, what is the probability that h makes an error on a new point? There are three sources of randomness here: the choice of c, the choice of the sample S, and the choice of a new point $\vec{x}$.[1] We can write this probability as $\Pr_{c,S,\vec{x}}[h(\vec{x}) \neq c(\vec{x})]$. We have

$$\begin{aligned}
\Pr_{c,S,\vec{x}}(h(\vec{x}) \neq c(\vec{x})) &\geq \Pr_{c,S,\vec{x}}(\vec{x} \notin S \wedge h(\vec{x}) \neq c(\vec{x})) \\
&= \Pr(x \notin S)\Pr(h(\vec{x}) \neq c(\vec{x}) \mid x \notin S) \\
&\geq \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} .
\end{aligned}$$

We can marginalize to get

$$\Pr_{c,S,\vec{x}}(h(\vec{x}) \neq c(\vec{x})) = \sum_{c \in \mathcal{C}'} \Pr(c)\Pr_{S,\vec{x}}(h(\vec{x}) \neq c(\vec{x})|c) = \mathrm{E}_c \left[\Pr_{S,\vec{x}}(h(\vec{x}) \neq c(\vec{x})|c)\right] .$$

Combining this with the previous result yields

$$\mathrm{E}_c \left[\Pr_{S,\vec{x}} \left(h(\vec{x}) \neq c(\vec{x})|c)\right)\right] \geq \frac{1}{4},$$

which implies that there must exist at least one function $c \in \mathcal{C}'$ such that

$$\Pr_{S,\vec{x}} \left(h(\vec{x}) \neq c(\vec{x})\right) \geq \frac{1}{4} .$$

This is the function we will choose as the "bad" target for $\mathcal{A}$.

Finally, we marginalize over S and work towards obtaining a bound on the probability that $\mathrm{err}(h)$ is high.

$$\begin{aligned}
\Pr_{S,\vec{x}} \left(h(\vec{x}) \neq c(\vec{x})\right) &= \mathrm{E}_S[\Pr_{\vec{x}} \left(h(\vec{x}) \neq c(\vec{x})\right)] \\
&= \mathrm{E}_S[\mathrm{err}(h)] \\
&= \Pr\left(\mathrm{err}(h) > \frac{1}{8}\right) \mathrm{E}_S \left[\mathrm{err}(h) \mid \mathrm{err}(h) > \frac{1}{8}\right] \\
&\quad + \Pr\left(\mathrm{err}(h) \leq \frac{1}{8}\right) \mathrm{E}_S \left[\mathrm{err}(h) \mid \mathrm{err}(h) \leq \frac{1}{8}\right] \\
&\leq \Pr\left(\mathrm{err}(h) > \frac{1}{8}\right) + \frac{1}{8} .
\end{aligned}$$

[1] We are implicitly assuming $\mathcal{A}$ is deterministic here, but it is an easy exercise to modify this proof so it holds for randomized algorithms too.

Using the bound above, this gives us that for the particular choice of $\mathcal{D}$ and c defined above,

$$\frac{1}{4} \leq \Pr\left(\mathrm{err}(h) > \frac{1}{8}\right) + \frac{1}{8}$$

and so

$$\Pr\left(\mathrm{err}(h) > \frac{1}{8}\right) \geq \frac{1}{8}\,.$$

□

2 And the rest...

We spent the remainder of class talking about the solutions to problem set 1 and guidelines for the course projects. We will continue talking about problem set 1 on Monday.