CS260: Machine Learning Theory
Lecture 5: Infinite Function Classes and VC Dimension

October 10, 2011

Lecturer: Jennifer Wortman Vaughan

1 Infinite Function Classes

The general learning bounds that we proved in the last few lectures hold for any finite class $\mathcal{H}$. In this lecture, we will consider the case in which $\mathcal{H}$ is infinite. We have already seen a couple examples of an infinite function classes: the class of one-dimensional threshold functions we analyzed in class, and the class of axis-aligned rectangles that you analyzed in the first homework assignment.

To handle infinite function classes, we need to replace the $\ln |\mathcal{H}|$ term in our bounds with some other notion of complexity. To gain some intuition, we begin with a "bad" argument of how this might be done. Suppose we have an $\mathcal{H}$ in which each function is parameterized by k real numbers. For example, an n-dimensional axis-aligned rectangle has $k = 2n$ parameters. An n-dimensional linear threshold function has $k = n + 1$ parameters. Suppose we would like to store a representation of such a function on a computer with finite memory. If we use b bits to represent each real valued parameter, it takes a total of kb bits to represent a function. Our hypothesis class would then effectively have 2^{kb} different hypotheses.

If we substitute this value into our sample complexity bound, we observe that the number of training examples we need to guarantee low error with high probability is $O\left(\frac{1}{\epsilon}(k + \ln(1/\delta))\right)$. In this case, the number of training samples needed is *linear* in the number of parameters.

While this argument provides a decent heuristic (complexity of a class is roughly equal to the number of parameters), the argument we have used here is not completely satisfying. In this lecture, we will discuss more accurate ways to measure the complexity of a concept class.

In everything that follows, we will assume that we are working in a model of computation in which we can store and manipulate real numbers in constant space and time. This is crucial if we want to say anything about efficient algorithms in this setting.

1.1 The Growth Function

Let S be a vector of m (arbitrary) examples $x_1, \ldots, x_m$. Given $h \in \mathcal{H}$ we define $h(S) = \langle h(x_1), \ldots, h(x_m) \rangle$ to be the behavior of h on the examples. There might be other $h' \in \mathcal{H}$ with identical behavior on these points, that is, other h' such that $h(S) = h'(S)$. The *behavior set* of $\mathcal{H}$ on S, denoted $\Pi_{\mathcal{H}}(S)$, is the set of all possible behaviors of functions $h \in \mathcal{H}$ on the set S, that is

$$\Pi_{\mathcal{H}}(S) = \{h(S) | h \in \mathcal{H}\}.$$

For binary classification, we have $|\Pi_{\mathcal{H}}(S)| \leq 2^m$ for all sets S of size m.

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.

The *growth function* of $\mathcal{H}$ is defined as

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

For any $\mathcal{H}$, it measures the maximum number of different ways that functions in $\mathcal{H}$ can behave on any set of points of a particular size. Note that $|\Pi_{\mathcal{H}}(m)| \leq 2^m$.

Let's look at some examples to get intuition.

Example 1. *Let $\mathcal{H}$ be the class of one-dimensional threshold functions. Given a single point in $(0,1)$, there are two ways that functions in $\mathcal{H}$ can label the point, so $\Pi_{\mathcal{H}}(1) = 2$. Given two points, there are three ways that functions in $\mathcal{H}$ can label them:*

$$\begin{array}{cc} - & - \\ - & + \\ + & + \end{array}$$

so $\Pi_{\mathcal{H}}(2) = 3$. Given 3 distinct points in $(0,1)$, there are 4 different possibilities for $h(S)$:

$$\begin{array}{ccc} - & - & - \\ - & - & + \\ - & + & + \\ + & + & + \end{array}$$

so $\Pi_{\mathcal{H}}(3) = 4$. In general, if we have m points, $\Pi_{\mathcal{H}}(m) = m + 1$. Notice that this is significantly smaller than the general bound 2^m.

Example 2. *Let $\mathcal{H}$ be the class of one-dimensional interval functions. Each function in the class is parameterized by two threshold values, a lower threshold (call this l) and an upper threshold (call this u). A point x is labeled positive if $x \in [l, u]$ and labeled negative otherwise.*

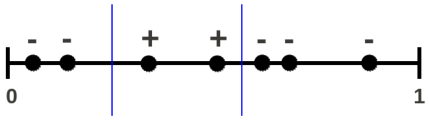

Figure 1: Intervals

If we are given m distinct points in $(0,1)$. How many different behaviors can we observe? To answer this question, it doesn't matter where exactly the interval boundaries lie, what matters is which pairs of points they lie between. Our m points define $m+1$ regions in $(0,1)$. Then the number of different behaviors equals the $\binom{m+1}{2}$ different ways we can choose these regions plus 1 (which is the case obtained if the two boundaries are in the same region and so all the points are labeled negative), which is $O(m^2)$. This is again far less than the pessimistic exponential bound.

It turns out that the growth function can be used to generate an error bound for infinite function classes. Note that we are back to considering the realizable setting in which there is a perfect target function. We will not discuss the proof of this result in class, but it is given in Chapter 3 of Kearns and Vazirani.

Theorem 1. *Consider any concept classes $\mathcal{C}$ and $\mathcal{H}$ for input space $\mathcal{X}$. Suppose we have an algorithm $\mathcal{A}$ that for any target function $c \in \mathcal{C}$, given a sample $x_1, \ldots, x_m \in \mathcal{X}$ labeled by c, will return a function $h \in \mathcal{H}$ consistent with this data. Then for any distribution $\mathcal{D}$ on $\mathcal{X}$, for any $c \in C$, for any $\delta \in (0, 1/2)$, if $\mathcal{A}$ is run on a sample of m points drawn i.i.d. from $\mathcal{D}$ and labeled by c, then with probability at least $1 - \delta$,*

$$\mathrm{err}(h) \leq k \left(\frac{\ln(\Pi_{\mathcal{H}}(2m)) + \ln(2/\delta)}{m} \right)$$

for a small constant k.

This bound is meaningless if the growth function is 2^m. However, as we have seen, it is often much smaller. The problem with this bound is that the growth function is a tricky quantity to calculate in general. For this reason, it is useful to introduce another notion of complexity, the VC dimension.

2 VC Dimension

We say that a set $S = \{x_1, \ldots, x_m\}$ of size m is *shattered* by the class $\mathcal{H}$ if all possible labelings of S are achievable by some $h \in \mathcal{H}$. Formally, $\mathcal{H}$ shatters S if $|\Pi_{\mathcal{H}}(S)| = 2^m$. The *VC dimension* of $\mathcal{H}$ is the cardinality of the largest set S that can be shattered by $\mathcal{H}$.

The VC dimension is convenient because it can be calculated for many of classes of interest. To prove that the VC dimension of a class $\mathcal{H}$ is d, it is necessary to 1) give an example of a set of d points that can be shattered, and 2) prove that no set of $d + 1$ points can be shattered.

Example 3. *Once again, let $\mathcal{H}$ be the class of one-dimensional thresholds. It is easy to see that 1 point can be shattered, but 2 cannot. Thus, $VC(\mathcal{H}) = 1$.*

Example 4. *Let $\mathcal{H}$ be the class of one-dimensional intervals. We observe that 2 points can be shattered, but 3 cannot. (See Figure 2.) Thus, $VC(\mathcal{H}) = 2$.*

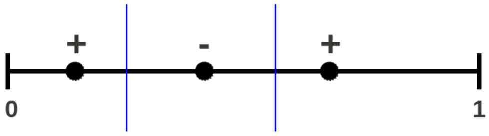

Figure 2: This configuration is not possible

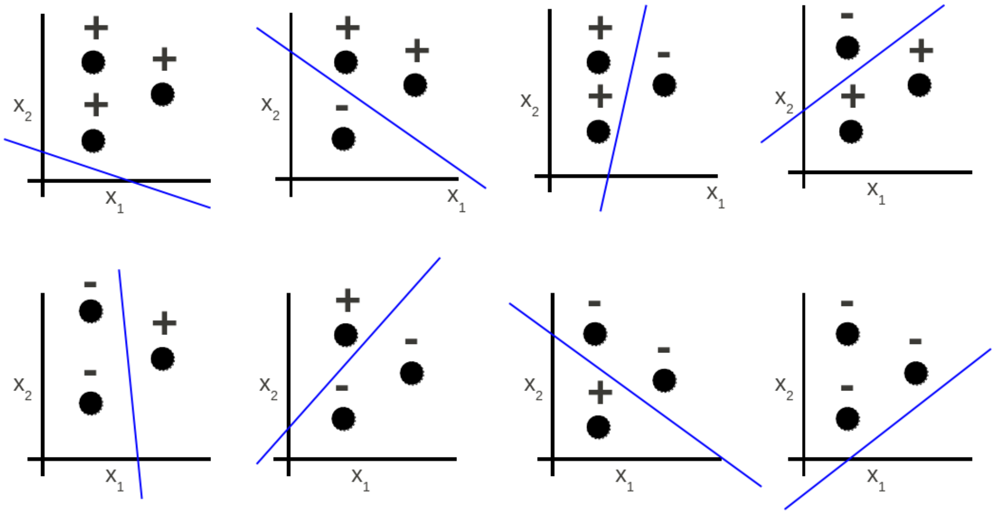

Figure 3: All the configurations possible

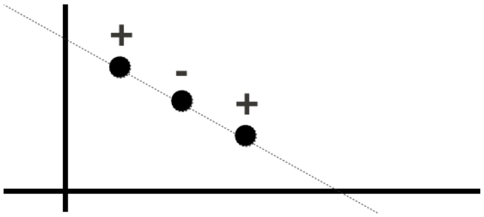

Figure 4: This configuration is not possible

Example 5. *Let $\mathcal{H}$ be the class of linear threshold functions in two dimensions. Any 3 points that do not lie in a line can be shattered.*

Note that a set of 3 points that lie on the same line cannot be shattered. However, this has no influence on the VC dimension because we only need there to exist one set of three points that can be shattered.

To prove that the VC dimension is 3, we must show that no set of four points can be shattered. We can break this down into two cases: First, if one of the 4 points lies in the convex hull of the other three, that point cannot get a different label than the rest, so the points cannot be shattered. If no point lies in the convex hull of the other three, then we can draw a square with the points as the four corners. Pick one pair of points that are diagonally across from each other. It is impossible to label these two points + and the other two -. Therefore, the points cannot be shattered in this case either.

This result extends beyond two dimensions. In fact, it is the case that the VC dimension of linear thresholds in n-dimensional space is $n + 1$.

VC dimension is useful because of its relationship with the growth function.

Lemma 1. *(Sauer's Lemma) For any $\mathcal{H}$ with finite VC dimension d,*

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

This lemma is very powerful. It tells us that all hypothesis classes fall into one of two categories: If d is infinite, then $\Pi_{\mathcal{H}}(m) = 2^m$. The bound in Theorem 1 is then meaningless. On the other hand, if d is finite, then $\Pi_{\mathcal{H}}(m) = O(m^d)$. In this case, the theorem gives us something very nice since $\ln |\Pi_{\mathcal{H}}(m)| = O(d \ln m)$. In this case, the error bound is linear in d and decreases to 0 as m goes to infinity.

Using Sauer's Lemma, it is possible to prove that under the same conditions in which Theorem 1 holds, if $\mathcal{H}$ has VC dimension d, then with probability of at least $1 - \delta$,

$$\mathrm{err}(h) = O\left(\frac{d \ln m + \ln(1/\delta)}{m}\right).$$

It is also possible to rearrange this bound to show that one can achieve $\mathrm{err}(h) \leq \epsilon$ with probability at least $1 - \delta$ with a 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

We have established that the VC dimension of one-dimensional threshold functions is 1. The bound above tells us that in order to achieve an error less than ϵ with high probability, we need

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

Recall that in Lecture 2, we used the structure of this class to prove that we can achieve the same guarantee with

$$m = \frac{1}{\epsilon}\ln\left(\frac{2}{\delta}\right) .$$

Observe that the bound we achieve using VC Dimension theory is not as tight, but not too bad.

Case: n-dimensional linear separators

We said that the VC Dimension of n-dimensional linear separators is $n + 1$. The bound above tells us that we can achieve error less than ϵ with high probability with

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

This follows the intuition in the bad argument we gave earlier, that the number of samples we need is roughly linear in the number of parameters.

3 The Unrealizable Setting

The bounds presented here can be extended to the unrealizable setting. Again, the sample complexity will scale with $1/\epsilon^2$ instead of $1/\epsilon$. Anyone especially curious about this extension can see Chapter 4 of "Neural Network Learning: Theoretical Foundations" by Anthony and Bartlett for a full derivation of results.