CHAPTER

7 VC Dimension

Up to this point, we have talked generally about the complexity or "flexibility" of our learner, or the set of functions it is able to approximate, without being too precise. We have seen several examples in which we increase the flexibility of our learner, for example moving from a linear classifier on our given features, to one on an extended feature set (such as quadratic features), allowing our function to fit the training data better but increasing our risk of overfitting.

One way to try to quantify the complexity of a particular learner is the Vapnik–Chervonenkis dimension, or VC dimension for short. This quantity can then be used to understand our risk of overfitting with a particular learner, at least to some extent, and even make guarantees about test time performance based only on training error rates.

Definitions

Before defining the VC dimension, let us state a few definitions.

DEFINITION 7.1. *We say that a learner $f(x;\theta)$ **separates** a particular data set $D = \{(x^{(i)}, y^{(i)}), i \in 1 \dots h\}$ if there exists some setting of the parameter θ such that, for all i, $f(x^{(i)};\theta) = y^{(i)}$.*

Separation is a property of both the learner f and the particular data set D; some data sets may be easy to predict for a particular learner, and some not.

Next, we use separation to define the concept of shattering:

DEFINITION 7.2. *We say that a learner $f(x;\theta)$ **shatters** a particular set of h points $X = \{x^{(i)} : i \in 1 \dots h\}$ if, for* every *set of labels $Y = \{y^{(i)} : i \in 1 \dots h\}$, with $y^{(i)} \in \{-1, +1\}$, the learner f can separate $D = \{(x^{(i)}, y^{(i)})\}$. In other words,*

$$\forall Y = \{y^{(i)}\},\ \exists \theta_Y \text{ such that } f(x^{(i)};\theta_Y) = y^{(i)}\ \forall i$$

Shattering generalizes the notion of separation to depend only on the feature vectors X, and measures whether it is easy for f to learn arbitrary binary functions over that set of points X.

Finally, we are ready to define the VC dimension:

DEFINITION 7.3. *The **VC dimension** H of the learner $f(x;\theta)$ is the largest value of h such that there exists an $X = \{x^{(i)} : i \in 1 \dots h\}$ that can be shattered by f:*

$$H = \max_h \text{ such that } \exists X = \{x^{(1)}, \dots, x^{(h)}\}\ \forall Y = \{y^{(1)}, \dots, y^{(h)}\}\ \exists \theta_Y\ :\ f(x^{(i)};\theta_Y) = y^{(i)}\ \forall i$$

In general, it is easier to prove a lower bound on H (that it is at least some value h) than an upper bound (that it can be no larger than h). If the feature vectors $x^{(i)}$ are real-valued, proving that no such set X exists can be difficult (involving geometric arguments, for example), while proving that one does exist can be done by simply constructing an example. We shall see this in several examples in the sequel.

It is sometimes useful to think of checking whether $H \geq h$ for some h as a two-player game. The definition of the learner f and feature space (dimensionality and possible values of $x^{(i)}$) form the "rules" of the game. Then first, Player 1 selects h feature locations, forming X; given these locations, Player 2 tries to select a labeling of the points, Y, that will be difficult for f to represent. Finally, Player 1 selects a value of θ; if $f(x;\theta)$ can represent the mapping of X and Y, Player 1 wins. If the VC dimension is at least h, then Player 1 will always win (assuming optimal play); if not, Player 2 can always force a loss.

Examples

It is helpful to see a few examples of VC dimension to understand how it works. For each example, we will carefully specify the form of our classifier $f(x;\theta)$, where x are the input features, and θ are the parameters that will be fit to training data.

Example 7-1 : Circle

For our first example, define

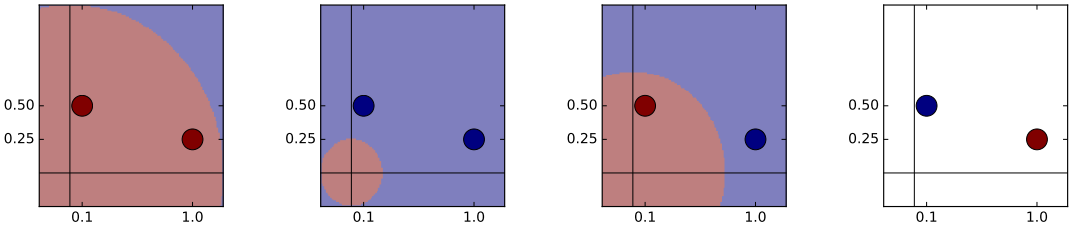
$$f(x;r) = \begin{cases} +1 & x \odot x^T \leq r^2 \\ -1 & \text{otherwise} \end{cases} \tag{7.1}$$

which is a learner that predicts $+1$ within a circle of radius r, and -1 outside it. Let us determine the VC dimension of f. We will assume for visualization that x is two-dimensional (two features), although the arguements hold more generally.

It is easy to verify that $H > 1$ constructively, by placing $x^{(1)}$ anywhere except the origin. Then, if $y^{(i)} = +1$, we simply select the parameter $r > \|x^{(1)}\|$, and if $y^{(1)} = -1$ we select $r < \|x^{(1)}\|$.

With a bit more work, we can prove that $H = 1$, by proving that H must be less than 2. Let us start by choosing a specific location for the two points $X = \{x^{(1)}, x^{(2)}\}$, and show that f cannot shatter those points; then we will use the intuition we obtain to argue that no set of shatterable points X exists.

Suppose that we select the locations $x^{(1)} = [0.1,\ 0.5]$ and $x^{(2)} = [1.0,\ 0.25]$. Then, three possible labelings of Y can be separated by some value of r, but the fourth, $y^{(1)} = -1, y^{(2)} = +1$, cannot:



since any setting of r that predicts $+1$ for $x^{(2)}$ must also predict $+1$ for $x^{(1)}$ (since its length is smaller). Moreover, this makes it easy to see why no choice of X with $h = 2$ can be shattered, since if one of the two points has smaller length than the other, the same pattern (in which the smaller-length data point is labelled negative) will not be separable; or, if the two points have the same length, then any pattern with $y^{(1)} \neq y^{(2)}$ cannot be separated.

As another example, consider a linear classifier using two features:

Example 7-2 : 2D Linear

Let our decision function be defined by,

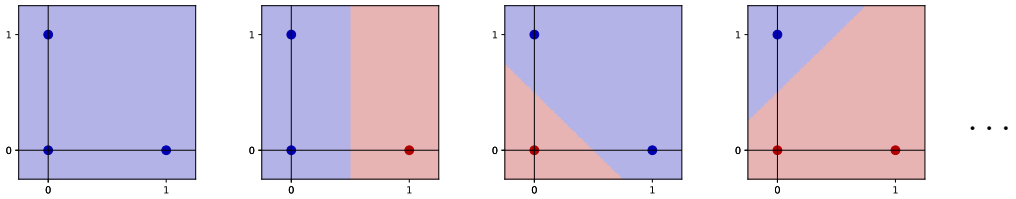
$$f(x; \theta = [a, b, c]) = \begin{cases} +1 & a + bx_1 + cx_2 \geq 0 \\ -1 & \text{otherwise.} \end{cases} \tag{7.2}$$

Informally, we know that any value of (a, b, c) will divide the space into positive and negative halves, with the decision boundary being a line, and that conversely we can find values of (a, b, c) to produce any linear decision boundary.

Now, again, we can easily prove that $H \geq 3$ by construction, placing three data points at (say)

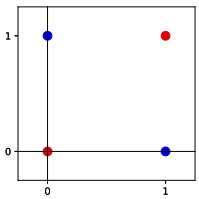
$$x^{(1)} = [0, 0] \qquad x^{(2)} = [1, 0] \qquad x^{(3)} = [0, 1].$$

Then, we can simply enumerate each of the 2^3 possible target labellings of these points, and show that there exists a value of (a, b, c) that correctly predicts them:



$r(x) =$ $\quad -1 \quad$ $2x_1 - 1$ $\quad -2x_1 - 2x_2 + 1 \quad$ $2x_1 - 2x_2 + 1$ $\quad \ldots$

To prove that $H = 3$ exactly, we show that $H < 4$. Again, if we explicitly place four points $x^{(i)}$, we can see what patterns are difficult to predict; in this case, we can place our points in a square, in which case we find (as we saw in Example 4-1) that the "XOR-like" pattern of label values cannot be separated by a linear boundary. With a bit more work, we can show that this is general: *any* positioning of four points in two dimensions must admit a similar pattern that cannot be linearly separated.



It is possible to derive general VC dimension results for some popular types of classifiers. For example, for a linear classifier (perceptron) with n features, $\text{sign}(\theta_0 + \sum_j \theta_j x_j)$, one can show that the VC dimension is $n + 1$ **?**. **add cite, make theorem?**

In many cases, the VC dimension of a learner will match its number of parameters (as was the case in the previous two examples). This is perhaps because often, more parameters are used to make the functional form of f more flexible, leading to a larger set of reproducible patterns; indeed, the number of parameters of a model is often a simple stand-in for measuring its complexity and ability to overfit. However, a learner's VC dimension tries to more precisely quantify its ability to overfit, and in some cases can be either greater or less than the number of parameters.

To see how the VC dimension can be fewer than the number of parameters, we can simply add parameters that have redundant or useless effects:

Example 7-3 : Circle Revisited

Let our decision function have parameters r, s and be defined by,

$$f(x; r, s) = \begin{cases} +1 & x \odot x^T \leq r^2 + s^2 \\ -1 & \text{otherwise} \end{cases}.$$

Compared to the learner in (7.1), this learner has two parameters, r and s, but both together determine the radius of the circle, resulting in exactly the same set of prediction functions as before, and hence the same VC dimension. In contrast, if we were to use the additional parameter s more effectively, say:

$$f(x; r, s) = \begin{cases} \text{sign}(s) & x \odot x^T \leq r^2 \\ -\text{sign}(s) & \text{otherwise} \end{cases}$$

we can easily prove that the new VC dimension has increased, from $H = 1$ to $H = 2$.

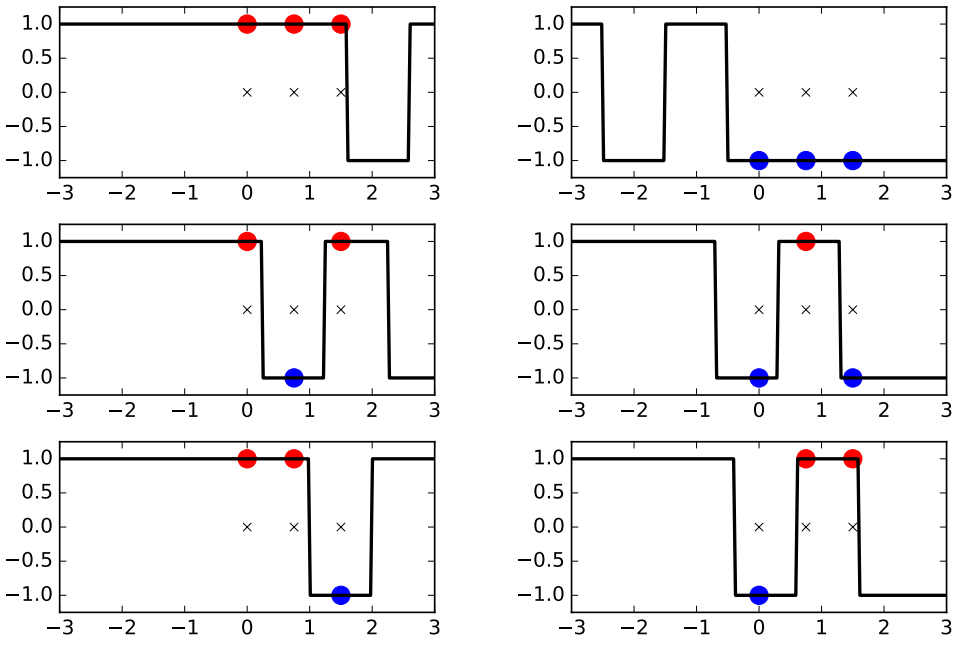
We can also construct examples in which a small number, or even a single parameter can be used to produce a large number of patterns, giving a VC dimension larger than the number of parameters. Take the following example:

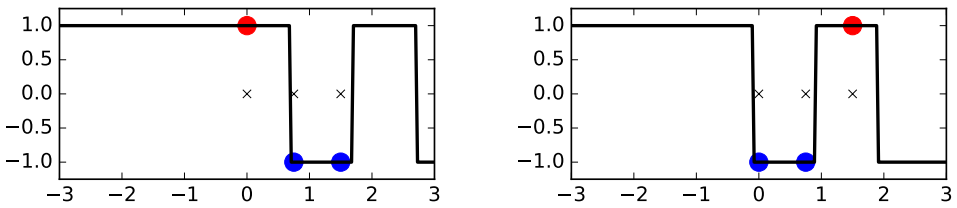
Example 7-4 : Double Pulse

Let our decision function be defined by the parameter t, with

$$f(x; t) = \begin{cases} +1 & x \in [-\inf, t] \cup [t + 1, t + 2] \\ -1 & \text{otherwise} \end{cases}$$

Now, for a carefully chosen set of points, for example $x^{(1)} = 0.0$, $x^{(2)} = 0.75$, $x^{(3)} = 1.5$, we can check by enumerating all the possibilities that X can be shattered:





However, it is easy to see that this $f(x)$ cannot shatter four points. Order the points by increasing value, then set the labels y to the pattern -1, +1, -1, +1. To correctly predict the first, we need $t < x^{(1)} < t+1 < x^{(2)} < t+2 < x^{(3)}$, but then $f(x^{(4)})$ cannot be +1.

It is even possible to design a learner that has a single parameter, yet infinite VC dimension:

Example 7-5 : Infinite VC dimension

Let our target values be $y \in \{0, 1\}$, select the points $X = \{x^{(i)} = 2^i, i = 1 \ldots\}$, and define our learner and its parameter value by

$$f(x\,;\,\omega) = \mathbb{1}[\sin(-\omega\pi x) \geq 0] \qquad\qquad \omega = \sum_i y^{(i)} \cdot 2^{-i}$$

Then, at $x^{(1)} = 2$ we have $f(x^{(1)}) = \sin(-\pi y^{(1)} - \pi\epsilon)$ for some $0 \leq \epsilon < 1$, which will be nonnegative if and only if $y^{(1)} = 1$. At $x^{(2)} = 4$ we have $f(x^{(2)}) = \sin(-2\pi y^{(1)} - \pi y^{(2)} - \pi\epsilon)$ for some $0 \leq \epsilon < 1$, which will now be nonnegative if and only if $y^{(2)} = 1$, and so on. Since we can do this process with any number of data points h, the VC dimension must be infinite.

This example highlights one of the drawbacks of VC dimension – since it considers how a particular learner might be able to fit arbitrary patterns on *some* set of locations X, it can overestimate what we might consider the "intuitive" complexity of a function. Most of the time, we are not concerned with arbitrary, possibly pathological placements of data points, but random draws from a distribution. However, the worst-case analysis of VC dimension allows us to obtain strong claims about the potential of a learner to overfit.

VC Bounds on Error

The VC dimension of f is a powerful characterization of its ability to learn arbitrary patterns, and hence to overfit to a given data set. For example, Vapnik **?** proved that the test error rate can be bounded with high probability in terms of the training error rate and a formula that involves the VC dimension. Specifically, suppose that the training data set D is drawn i.i.d. from $p(x, y)$. Then, after training a learner f with VC dimension H on $D = \{x^{(i)}, y^{(i)}\}$, we have:

With probability at least $1 - \eta$,

$$\mathbb{E}[\,\mathbb{1}[y \neq f(x)]] \;\leq\; \frac{1}{m}\sum_i \mathbb{1}[y^{(i)} \neq f(x^{(i)})] \;+\; \sqrt{\frac{1}{m}\left(H \log\left(\frac{2m}{H}\right) + H - \log\frac{\eta}{4}\right)}$$

$$\text{(test error)} \quad \leq \quad \text{(training error)} \quad + \quad \text{(bound)}$$

The probabilistic nature of the bound (i.e., that it holds with probability $1 - \eta$) is due to viewing the training data set as a random draw from $p(x, y)$ – intuitively, there is some probability that we are unlucky in our data collection process, and our training set is not really representative of the distribution. For example, we could have the bad fortune to collect *only* data with $y^{(i)} = -1$; then our model would obtain zero training error, but could not possibly learn the correct relationship between x and y. However, the probability of such an unrepresentative draw of training data rapidly decreases as we obtain more training data (m increases).

Sample complexity rates

include?

Structural risk minimization

Model selection using VC dimension loss bound. Empirical loss plus a complexity term.