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 VapnikChervonenkis 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 $D = \{(x^{(i)}, y^{(i)}), i \in 1 \ldots h\}$ 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 $X = \{x^{(i)} : i \in 1 \ldots h\}$ if, for* every *set of labels $Y = \{y^{(i)} : i \in 1 \ldots h\}$, with $y^{(i)} \in \{-1, +1\}$, the learner f can separate $D = \{(x^{(i)}, y^{(i)})\}$. In other words,*

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

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 $X = \{x^{(i)} : i \in 1 \ldots h\}$ that can be shattered by f:*

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

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

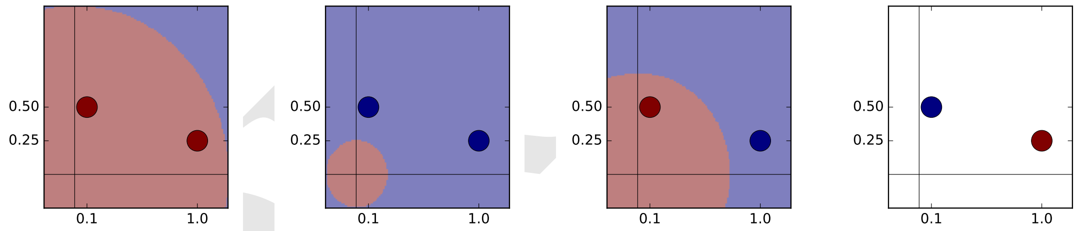
Fix Define

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

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.

It is easy to verify that $H > 1$ 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 $r < x^{(1)}$.

We can similarly prove that $H = 1$ by proving that H must be less than 2. Suppose that we use the points $x^{(1)} = [0.1,\ 0.5]$ 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, no choice of X with $h = 2$ can be shattered, since either one of the two points will have smaller length than the other (leading to the preceding example), or they will have the same length (in which case any pattern with $y^{(1)} \neq y^{(2)}$ cannot be separated).

Linear classifier (perceptron): has VC dimension $n+1$, where n is the number of features.

In many cases, the VC dimension of a learner will match its number of parameters (as 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. Take the following trivial example:

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

Compared to the learner in (**??**), 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:

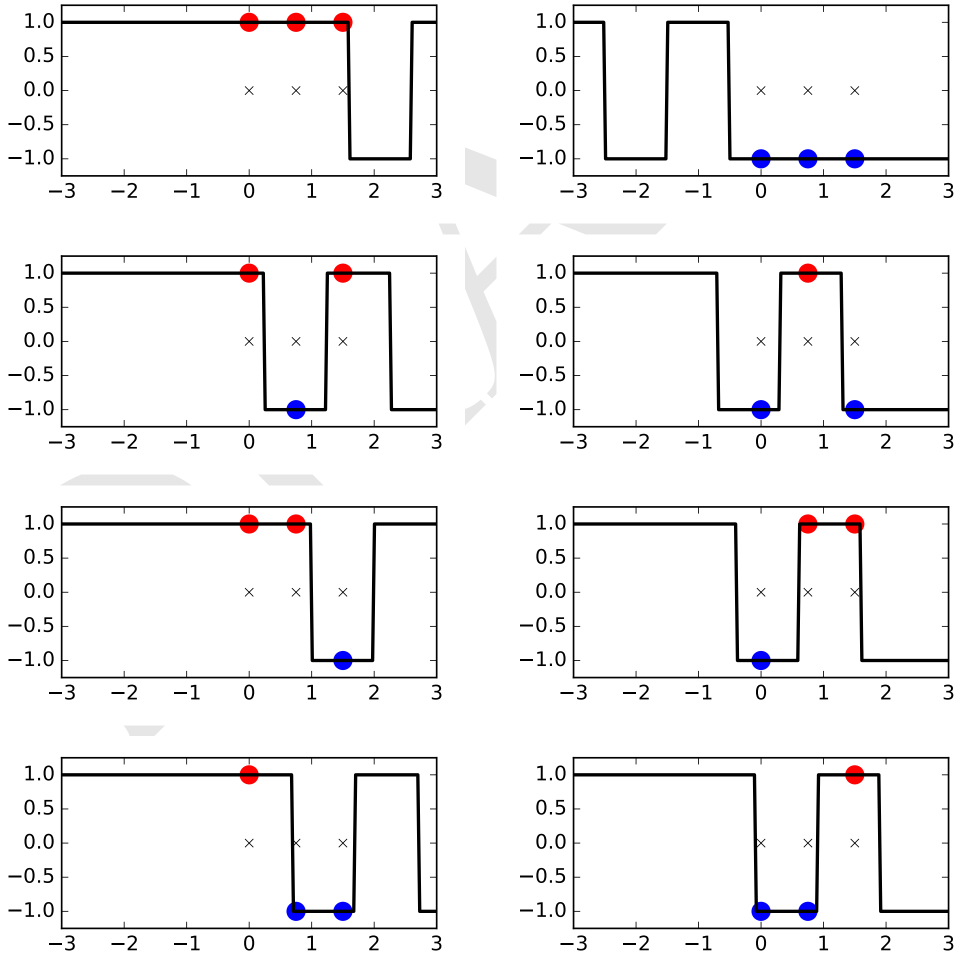
$$f(x; r, s) = \begin{cases} \text{sign}(s) & x \cdot x^T \leq r^2 \\ -\text{sign}(s) & \text{otherwise} \end{cases}$$

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:

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

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.

Example of infinite VC dimension: let $y \in \{0, 1\}$, $f(x) = \mathbb{1}[\sin(-\omega \pi x) \geq 0]$; take $x^{(i)} = 2^i$ for $i = 1 \ldots h$, for any finite h, and $\omega = \sum y^{(i)} \cdot 2^{-i}$. Then at $x^{(1)} = 2$ we have $\sin(-\pi y^{(1)} - \pi\epsilon)$ for some $0 \leq \epsilon < 1$, which will be nonnegative if and only if $y^{(1)} = 1$. At $x = 4$ we have $\sin(-2\pi y^{(1)} - \pi y^{(2)} - \pi\epsilon)$ for some $0 \leq \epsilon < 1$, which will now be nonnegative if and only if $y^{(2)} = 1$, and so on.

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 $1 - \eta$,

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

(test error) (training error) + (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 – there is some probability (rapidly decreasing as the number of training examples M increases) that we are unlucky, and our training set is not really representative of the distribution $p(x, y)$.