Learnability Analysis

Paul Laliberte' | CSCI-5622

1. Finding a bounds for triangles on a graph.

We must first identify which bound to use. Consider our hypothesis class H. We have that $H = C$ since our hypothesis class H must also be concepts defined by triangles which have vertices of form $(i, j) \in [0, 99]$. Then H is finite (only considers values on the interval $[0, 99]$) and our training error, $\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{I}_{h(x_i)\neq c(x_i)} = 0$, for an algorithm $\mathcal{A}$ that learns H. What we mean by, "the training error is zero," is that since $H = C$ there cannot be any occurrences of $h(x) \neq c(x)$, $h \in H$, $c \in C$, $x \in S$ (where S is our labeled set of training examples). Thus, we are in the case of a finite consistent hypothesis class. The bound for this case is

$$m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right),$$

where m is the size of S.

Now consider our problem at hand. We first need to calculate what ϵ and δ are. We define ϵ as the error rate, which is given as $\epsilon = .15$. Similarly, we define δ as the error in our confidence rate. We have a goal of a .95 confidence rate, and thus the error of this is $\delta = 1 - .95 = .05$. Now we need to find $|H|$, or the cardinality of H. The degenerate cases are accepted and both x and y can take on 100 possible integer values. Pair that with the property that any given coordinate is distinct, and we can use the combinatorial identity, $\binom{100+100}{k} = \binom{200}{k}$, to count the number of possible triangles. Hence, for three vertices we have $k = 3$ such that

$$|H| = \binom{200}{3} = 1313400.$$

Thus, our bound is as follows

$$\begin{aligned} m &\geq \frac{1}{\epsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right) \\ &\geq \frac{1}{.15}\left(\ln(1313400) + \ln\left(\frac{1}{.05}\right)\right) \\ &\geq 113.892. && \text{(Rounded value)} \end{aligned}$$

Therefore, we need approximately 114 training examples.

2. VC Dimension.

Proposition. The VCdim$(H) = 2$, where H is the class of hypotheses defined by circles centered at the origin and where a hypothesis $h \in H$ can either classify points as positive if they lie on the boundary or interior of the circle, or can classify points as positive if they lie on the boundary or exterior of the circle.

Proof. Consider the lower bound on VCdim(H) first. Obviously, we can bound any vector $\mathbf{y}$ consisting of only one point $y_1 \in \mathbb{R}$ by adjusting the radius r of some circle hypothesis $h \in H$ that is centered at the origin. For a vector of two points $\mathbf{y} = (y_1 \ y_2) \in \mathbb{R}$, we can shatter some $\mathbf{y}$, but not all of them (take for example when $y_1 < 0$ and $y_2 = |y_1|$). However, by definition of VC dimension, we only need to satisfy one set of points for which we can shatter all possible classifications ($2^2 = 4$ for two points), not all possible sets of real numbers. This can be done by considering the set of points where $y_1 \leq y_2$. If both are classified as positive (+1) then we can have a hypothesis h, with radius r, shatter the points by $r \leq y_1 \leq y_2$ or $y_1 \leq y_2 \leq r$. If both points are classified as negative then we can shatter the points by $r < y_1 \leq y_2$ or $y_1 \leq y_2 < r$. If y_1 and y_2 have different classifications (i.e. one is positive and the other negative) then we can shatter them by $y_1 \leq r < y_2$ and $y_1 < r \leq y_2$. Thus, a lower bound is VCdim$(H) \geq 2$.

Now consider the upper bound on VCdim(H). Suppose we can also shatter a vector of three points $\mathbf{y} = (y_1 \ y_2 \ y_3) \in \mathbb{R}$. Without loss of generality, consider the set of points where $y_1 \leq y_2 \leq y_3$, and let y_1 and y_3 have positive classification (+1), whereas let y_2 have negative classification (-1). Also, let some circle hypothesis $h \in H$ with radius r, and centered at the origin, shatter $\mathbf{y}$. Since y_1 and y_3 are both positively labeled (+1) they must both lie on the boundary of r, or in the boundary of r, or outside the boundary of r. The first option is not feasible since if $|y_1| = |y_3| = r$ then $|y_1| = |y_2| = |y_3| = r$ and the radius boundary is strictly classified as positive (y_2 would be misclassified). Note that we use absolute values to emphasize that both a negatively and positively valued point, not the classification labels, can lie on the r boundary (i.e. if $y_1 = (-4, 0)$, $y_2 = (0, 4)$, $y_3 = (4, 0)$, and $r = 4$ then all are located on the r boundary). For the last two options we have that $r < y_1 \leq y_2 \leq y_3$ (outside boundary) or $y_1 \leq y_2 \leq y_3 < r$ (inside boundary), which is not feasible since either the entire interior is classified as positive or the entire exterior is classified as positive (y_2 misclassified in both cases). Hence, we have a contradiction and the upper bound is VCdim$(H) < 3$.

Therefore, we have that VCdim$(H) \geq 2$ and VCdim$(H) < 3$, which results in VCdim$(H) = 2$. □

3. Sine Curve Classification.

We have the following solution technique as described in section 2.3 of, "A Tutorial on Support Vector Machines for Pattern Recognition," by Christopher Burges. We consider the family of one one-paramter functions, $f(x, \alpha) \equiv \theta\big(\sin(\alpha x)\big)$, with $x, \alpha \in \mathbb{R}$. We assign arbitrary binary labels, i.e. $y_i \in \{-1, +1\}$, $i = 1, \ldots, l$, where l is the number of training points present. Burges claims that for some l we can shatter these l points in $x_i = 10^{-i}$ ways , $i = 1, \ldots, l$, producing an infinite VC dimension. Furthermore, $f(\alpha)$ produces the correct labeling if we define α to be

$$\alpha = \pi \left(1 + \sum_{i=1}^{l} \frac{(1 - y_i)10^i}{2} \right).$$

We proved this, by construction, in vc_sin.py. Note that in the vc_sin.py script we choose our shattered points in $x_i = 2^{-i}$ ways, and thus in α we swap the 10 for a 2 (i.e. 2^i instead of 10^i).

As mentioned in Burges, even though we can shatter an arbitrarily large amount of points, it is also possible to create 4 points in which we cannot shatter. We provide an example of this in vc_sim.py (run with parameter '--cs y'). To describe the example in more detail, Burges states we need 4 equally spaced points with labeling $(+1, +1, -1, +1)$. In our example we provide a slightly different labeling (though of the same 3-to-1 structure) with the same points:

$$\text{array of points that cannot be shattered} = \big[(1, \text{False}), (2, \text{False}), (3, \text{True}), (4, \text{False})\big].$$

The point (2, False) is mislabeled and we cannot shatter this set. This can also be shown with any example that fulfills the requirements laid out by Burges. For example, another set of 4 points that cannot be shattered would be 2, 4, 6, 8 with the same 3-to-1 labeling.