6 Learning and VC-dimension

6.1 Learning

Learning algorithms are general purpose tools that solve problems often without detailed domain-specific knowledge. They have proved to be very effective in a large number of contexts. We start with an example. Suppose one wants an algorithm to recognize whether a picture is that of a car. One could develop an extensive set of rules (one possible rule is that it should have at least 4 wheels) and then have the algorithm check the rules. Instead, in learning, the rules themselves are learnt or developed by the algorithm. One first has a human judge or ``teacher'' label many examples as either ``car'' or ``not car''. The teacher provides no other information. The labeled examples are fed to a learning algorithm or ``learner'' whose task is to output a ``classifier'' consistent with the labeled examples. Note that the learner is not responsible for classifying all examples correctly, only for the given examples, called the ``training set'', since that is all it is given.

Intuitively, if our classifier is trained on sufficiently many training examples, then it seems likely that it would work well on the space of all examples. The theory of Vapnik-Chervonenkis dimension (VC-dimension), we will see later, indeed confirms this intuition. The question we consider now is: what classifiers can be learnt in polynomial time (or more efficiently)? Efficient algorithms for this question will depend on the classifier; in general, optimization techniques such as linear and convex programming, play an important role.

Each object, a picture of a car in the above example, is represented in the computer by a list of ``features''. A feature may be the intensity of a particular pixel in a picture, a physical dimension, or a Boolean variable indicating whether the object has some property. The choice of features is domain specific and we do not go into this here. In the abstract, one could think of each object as a point in d-dimensional space, one dimension standing for the value of each feature. This is similar to the vector-space model in Chapter 2 for representing documents. The teacher labels each example as +1 for a car or -1 for not a car.

The simplest rule is a half-space: does a weighted sum of feature-values exceed a threshold? Such a rule may be thought of as being implemented by a threshold gate that takes the feature values as inputs, computes their weighted sum and outputs yes or no depending on whether or not the sum is greater than the threshold. One could also look at a network of interconnected threshold gates called a neural net. Threshold gates are sometime called perceptrons since one model of human perception is that it is done by a neural net in the brain.

6.2 Learning Linear Separators, Perceptron Algorithm and Margins

The problem of learning a half-space (or a linear separator) consists of *n* labeled examples $a_1, a_2, \ldots a_n$ in *d*-dimensional space. The task is to find a *d*-vector *w* (if one exists) and a threshold *b* such that

$$\begin{aligned} w \cdot a_i &> b \quad \text{for each } a_i \text{ labelled +1} \\ w \cdot a_i &< b \quad \text{for each } a_i \text{ labelled -1.} \end{aligned} \tag{1}$$

A vector-threshold pair, (w,b), satisfying the inequalities is called a ``linear separator''.

The above formulation is a linear program (LP) in the unknowns w_j and *b* and can be solved by a general purpose LP algorithm. Linear programming is solvable in polynomial time but a simpler algorithm called the perceptron learning algorithm can be much faster when there is a feasible solution *w* with a lot of ``wiggle room'' (or margin), though, it is not polynomial time bounded in general.

First, a technical step: add an extra coordinate to each a_i and *w* to write: $\hat{a}_i = (a_i, 1)$ and $\hat{w} = (w, -b)$. Suppose l_i is the ± 1 label on a_i. Then, (1) can now be rewritten as

$$(\hat{w} \cdot \hat{a}_i) l_i > 0 \quad 1 \le i \le n$$

Since the right hand side is 0, scale $\hat{a}_i$ so that $|\hat{a}_i| \le 1$. Adding the extra coordinate increased the dimension by one but now the separator contains the origin. For simplicity of notation, in the rest of this section, we drop the hats, and let a_i and *w* stand for the corresponding $\hat{a}_i$ and $\hat{w}$.

The Perceptron Learning Algorithm

The perceptron learning algorithm is simple and elegant. We wish to find a solution *w* to :

$$(w \cdot a_i) l_i > 0 \qquad 1 \le i \le n, \quad \text{where } |a_i| \le 1. \tag{2}$$

Starting with $w = l_1 a_1$, pick any example a_i with $(w \cdot a_i) l_i \le 0$, and replace *w* by $w + l_i a_i$. Repeat until $(w \cdot a_i) l_i > 0$ for all *i*.

The intuition is that correcting *w* by adding $a_i l_i$ causes the new $(w \cdot a_i) l_i$ to be higher by $a_i \cdot a_i l_i^2 = |a_i|^2$. This is good for this *i*. But this change may be bad for other a_j. However, the proof below shows that this very simple process yields a solution *w* fast provided there exists some solution with a good margin.

Definition: For a solution *w* to (2), where $|a_i| \le 1$ for all examples, the margin is defined to be the minimum distance of the hyperplane $\{x : w \cdot x = 0\}$ to any a_i, namely, the margin is $\frac{\text{Min}_i (w \cdot a_i) l_i}{|w|}$. If we do not require in (2) that all $|a_i| \le 1$, then we could artificially

increase the margin by scaling up all a_i. If we did not divide by |w| in the definition of margin, then again, we can artificially increase the margin by scaling w up.

The interesting thing is that the number of steps of the algorithm depends only upon the best margin any solution can achieve, not upon *n* and *d*. In practice, the perceptron learning algorithm works well.

Theorem 6.1: Suppose there is a solution w^* to (2) with margin $\delta > 0$. Then, the Perceptron Learning Algorithm finds some solution w with $(w \cdot a_i) l_i > 0$ in at most $\frac{1}{\delta^2} - 1$ iterations.

Proof: Without loss of generality assume $|w^*| = 1$ (by scaling w^*). Consider the cosine of the angle between the current vector w and w^*, that is, $\frac{w \cdot w^*}{|w|}$. In each step of the algorithm, the numerator of this fraction increases by at least δ because

$$(w + a_i l_i) \cdot w^* = w \cdot w^* + l_i a_i \cdot w^* \geq w \cdot w^* + \delta .$$

On the other hand, the square of the denominator increases by at most 1 because

$$|w + a_i l_i|^2 = (w + a_i l_i) \cdot (w + a_i l_i) = |w|^2 + 2(w \cdot a_i) l_i + |a_i|^2 l_i^2 \leq |w|^2 + 1$$

(since $w \cdot a_i l_i \leq 0$ implies that the cross term is non-positive). Therefore, after t iterations, $w \cdot w^* \geq (t+1)\delta$ (since at the start, $w \cdot w^* = l_1 (a_1 \cdot w^*) \geq \delta$) and $|w|^2 \leq t+1$, i.e., $|w| \leq \sqrt{t+1}$ (since at the start, $|w| = |a_1| \leq 1$). Thus, the cosine of the angle between w and w^* is at least $\frac{(t+1)\delta}{\sqrt{t+1}}$ and the cosine cannot exceed 1. Thus, the algorithm must stop before $\frac{1}{\delta^2} - 1$ iterations and at termination, $(w \cdot a_i) l_i > 0$ for all *i*. This yields the Theorem.

■

How strong is the assumption that there is a separator with margin at least δ? Suppose for the moment, the a_i are picked from the uniform density on the surface of the unit hypersphere. We saw in Chapter 2 that for any fixed hyperplane passing through the origin, most of the mass is within distance $O(1/\sqrt{d})$ of the hyperplane. So, the probability of one fixed hyperplane having a margin of more than $c/\sqrt{d}$ is low. But this does not mean that no hyperplane can have a larger margin: if we just have the union bound, we can only assert that the probability of some hyperplane having a large margin is at most the probability of any one having a large margin times the number of hyperplanes (which is infinite!). Later we will see using VC-dimension arguments that indeed, the probability of some hyperplane having a large margin is low if the examples are uniformly random from the hypersphere. So, this assumption of large marign separators existing may not be valid for the simplest random models. But intuitively, if what is to be leaned, like whether something is a car, is not very hard, then, with enough features in the model, there will not be many ``near cars'' that could be confused with

cars nor many ``near non-cars''. So uniform density is not a valid assumption. In this case, there would indeed be a large margin separator and the theorem will work.

The question arises as to how small margins can be. Suppose the examples $a_1, a_2, \cdots, a_n$ were vectors with *d* coordinates, each coordinate a 0 or 1 and the decision rule for labeling the examples was the following.

If the least i such that a_i is 1 is odd, label the example +1

If the least i such that a_i is 1 is even, label the example -1

This rule can be represented by the decision rule

$$\left(a_{i,1}, a_{i,2}, \cdots, a_{i,n}\right)\left(1, -\tfrac{1}{2}, \tfrac{1}{4}, -\tfrac{1}{8}, \cdots\right)^T = a_{i,1} - \frac{1}{2}a_{i,2} + \frac{1}{4}a_{i3} - \frac{1}{8}a_{i4} + \ldots > 0$$

(see Exercise ???). But the margin for this can be exponentially small. Indeed, if for an example *a*, the first *d*/10 coordinates are all zero, then the margin is $O(2^{-d/10})$ as an easy calculation shows.

Maximizing the Margin

In this section, we present an algorithm to find the maximum margin separator. The margin of a solution w to (2) as defined is Min $\frac{l_i(w \cdot a_i)}{|w|}$. This is not a concave function of w. So, it is difficult to deal with computationally. Recall that the broadest class of functions, which we know how to maximize or minimize (over a convex set), are concave functions. Both these problems go under the name of ``Convex Optimization''.

A slight rewrite of (2) makes the job easier. If the margin of w is δ, we have

$$l_i\left(\frac{wa_i}{|w|\delta}\right) > 1.$$

Now, if $v = \frac{w}{\delta |w|}$, maximizing δ is equivalent to minimizing $|v| = \frac{1}{\delta}$. So we have the restated problem:

$$\text{Min } |v| \text{ subject to } l_i(v \cdot a_i) > 1, \forall i.$$

As we see from the exercise, $|v|^2$ is a better function than $|v|$ to minimize (since it is differentiable), so we use that and reformulate the problem as:

Maximum Margin Problem: Min $|v|^2$ subject to $l_i(v \cdot a_i) > 1$.

This Convex Optimization problem has been much studied and algorithms use the special structure of this problem to solve it more efficiently than general convex optimization problems. We do not discuss these improvements here.

Linear Separators that classify most examples correctly

It may happen that there are linear separators for which almost all but a small fraction of examples are on the correct side. Going back to (2), we could ask if there is a w for which at least $(1-\epsilon)n$ of the n inequalities in (2) are satisfied. Unfortunately, such problems are NP-hard and there are no good algorithms to solve them. A way to think about this is : we suffer a ``loss'' of 1 for each misclassified point and would like to minimize the loss. But this loss function is terribly discontinuous, it goes from 0 to 1 abruptly. However, with nicer loss functions, it is possible to solve the problem. One possibility is to introduce slack variables $y_i, i = 1, 2, \ldots n$, where y_i measures how badly classified example a_i is. We then include the slack variables in the objective function to be minimized:

$$\begin{array}{ll} \text{Min} & |v|^2 + c\sum_{i=1}^{n} y_i \\ \text{subject to} & \left.\begin{array}{l} (v \cdot a_i) l_i \geq 1 - y_i \\ y_i \geq 0. \end{array}\right\} i = 1, 2, \ldots n \end{array}$$

Note that if for some i, $l_i(v \cdot a_i) \geq 1$, then we would set y_i to its lowest value, namely 0, since each y_i has a positive coeffiecnt. If however, $l_i(v \cdot a_i) < 1$, then we would set $y_i = 1 - l_i(v \cdot a_i)$, so y_i is just the amount of violation of this inequality. Thus, the objective function is trying to minimize a combination of the total violation as well as 1/margin. It is easy to see that this is the same as minimizing

$$|v|^2 + c\sum_i \left(1 - l_i(v \cdot a_i)\right)^+,$$

where the second term is the loss function equal to sum of the violations.

6.3 Non-Linear Separators, Support Vector Machines and Kernals

There are problems where no linear separator exists but where there are non-linear separators. For example, there may be a polynomial $p(\cdot)$ such that $p(a_i) > 1$ for all +1 labeled examples and $p(a_i) < 1$ for all -1 labeled examples. A simple instance of this is the unit square partitioned into four pieces where the top right and the bottom left pieces are the +1 region and the bottom right and the top left are the -1 region. For this, $x_1 x_2 > 0$ for all +1 examples and $x_1 x_2 < 0$ for all -1 examples. So, the polynomial $p(\cdot) = x_1 x_2$ separates the regions.

A more complicated instance is the checker-board pattern in Figure 6.2 below with alternate +1 and -1 squares.

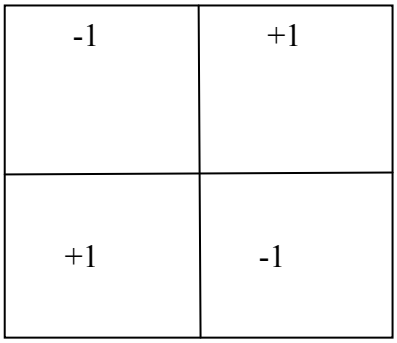


Figure 6.1

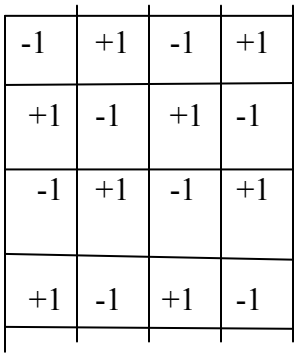


Figure 6.2

If we know that there is a polynomial p of degree[1] at most D such that an example a has label +1 if and only if $p(a) > 0$, then the question arises as to how to find such a polynomial. Note that each *d*-tuple of integers $(i_1, i_2, \ldots, i_d)$ with $i_1 + i_2 + \ldots + i_d \le D$ can lead to a distinct monomial : $x_1^{i_1} x_2^{i_2} \ldots x_d^{i_d}$. So, the number of monomials in the polynomial *p* is at most the number of ways of inserting d-1 dividers into a sequence of D+d-1 positions which is $\binom{D+d-1}{d-1} \le (D+d-1)^{d-1}$. Let $m = (D+d-1)^{d-1}$ be the upper bound on the number of monomials.

We can let the coefficients of the monomials be our unknowns and then it is possible to see that we can formulate a linear program in m variables whose solution gives us the required polynomial. Indeed, suppose the polynomial *p* is

$$p(x_1, x_2, \ldots, x_d) = \sum_{\substack{i_1, i_2, \ldots, i_d \\ i_1 + i_2 + \ldots i_d \le D}} w_{i_1, i_2, \ldots i_d} x_1^{i_1} x_2^{i_2} \ldots x_d^{i_d}.$$

Then the statement $p(a_i) > 0$ (recall a_i is a *d*-vector) is just a linear inequality in the $w_{i_1, i_2, \ldots i_d}$. So, one may try to solve such a linear program. But the exponential number of variables for even moderate values of D makes this approach infeasible. However, the theoretical approach we used for this will indeed be useful as we will see. First, we clarify the discussion above with an example. Suppose $d = 2$ and $D = 2$ (as in Figure 1). Then the possible (i_1, i_2) form the set $\{(1,0),(0,1),(1,1),(2,0),(0,2)\}$. We ought to

[1] The degree is the ``total degree''. The degree of a monomial is the sum of the powers of each variable in the monomial and the degree of the polynomial is the maximum degree of its monomials. In the example of **?????**Figure 2, the degree is 6.

include the pair $(0,0)$ also; but we will find it convenient to have a separate constant term which we will call b again. So we can write

$$p(x_1, x_2) = b + w_{1,0}x_1 + w_{0,1}x_2 + w_{1,1}x_1x_2 + w_{20}x_1^2 + w_{02}x_2^2$$

Each example a_i is a 2-vector which we denote (a_{i1}, a_{i2}). Then the Linear Program is:

$$b + w_{1,0}a_{i1} + w_{0,1}a_{i2} + w_{1,1}a_{i1}a_{i2} + w_{20}a_{i1}^2 + w_{02}a_{i2}^2 > 0 \quad \text{for } i : l_i = +1$$
$$b + w_{1,0}a_{i1} + w_{0,1}a_{i2} + w_{1,1}a_{i1}a_{i2} + w_{20}a_{i1}^2 + w_{02}a_{i2}^2 < 0 \quad \text{for } i : l_i = -1$$

Note that we ``pre-compute'' $a_{i1}a_{i2}$, so this does not cause a non-linearity. The point is that we have linear inequalities in the unknowns which are the w's and b.

The approach above can be thought of as ``embedding'' the examples a_i (which are in d space) into a M dimensional space where there is one coordinate for each $i_1, i_2, \ldots i_d$ summing to at most D (except $(0,0,0\ldots0)$) and if $a_i = (x_1, x_2, \cdots, x_d)$, this coordinate is $x_1^{i_1}x_2^{i_2}\ldots x_d^{i_d}$. Call this embedding $\phi(x)$. When $d = D = 2$ as in the above example, $\phi(x) = (x_1, x_2, x_1x_2)$. If $d = 3$ and $D = 2$, $\phi(x) = (x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3)$, and so on.

We then try to find a m-dimensional vector w such that the dot product of w and $\phi(a_i)$ is positive if the label is +1, negative otherwise. Note that this w is not necessarily the ϕ of some vector in d space.

Instead of finding any w, we will again want to find the w maximizing the margin. As earlier, we write this program as

$$\min |w|^2 \text{ subject to } (w \cdot \phi(a_i))l_i \geq 1 \text{ for all } i.$$

The major question is whether we can avoid having to explicitly compute/write down the embedding ϕ (and also w)? Indeed, an advantage of Support Vectors Machines (SVM's) is that we will only need to have ϕ and w implicitly. This is based on the simple, but crucial observation that any optimal solution w to the convex program above is a linear combination of the $\phi(a_i)$. **EXPAND**

Lemma 6.1: Any optimal solution w to the convex program above is a linear combination of the $\phi(a_i)$.

Proof: If w has a component perpendicular to all the $\phi(a_i)$, simply zero out that component. This preserves all the inequalities since the $w \cdot \phi(a_i)$ do not change, but, decreases $|w|^2$ contradicting the assumption that w is an optimal solution.

■

Thus, we may now assume that w is a linear combination of the $\phi(a_i)$. Say $w = \sum_i y_i \phi(a_i)$, where the y_i are real variables. Note that then

$$|w|^2 = \left(\sum_i y_i \phi(a_i)\right) \cdot \left(\sum_j y_j \phi(a_j)\right) = \sum_{i,j} y_i y_j (\phi(a_i) \cdot \phi(a_j)).$$

Reformulate the convex program as

$$\text{minimize} \quad \sum_{i,j} y_i y_j (\phi(a_i) \cdot \phi(a_j))$$

$$\text{subject to} \quad l_i \left(\sum_j y_j (\phi(a_j) \cdot \phi(a_i))\right) \geq 1 \quad \forall i.$$

The important thing to notice now is that we do not need ϕ itself but only the dot products of $\phi(a_i)$ and $\phi(a_j)$ for all i and j including $i = j$. The *kernel matrix* K defined as

$$k_{i,j} = \phi(a_i) \cdot \phi(a_j),$$

suffices since we can rewrite the convex program as

$$\min \sum_{ij} y_i y_j k_{ij} \quad \text{subject to} \quad l_i \sum_j k_{i,j} y_j \geq 1.$$

The advantage is that K has only d^2 entries instead of the $O(d^D)$ entries in each $\phi(a_i)$. So, instead of specifying $\phi(a_i)$, we just need to write down how we get K from the a_i. This is usually described in closed form. For example, the often used ``Gaussian kernel'' is given by:

$$k_{ij} = \phi(a_i) \cdot \phi(a_j) = e^{-c|a_i - a_j|^2}.$$

WHAT IS $\phi(a_i)$ AND WHERE DID MINUS SIGN COME FROM?

An important question arises. Given a matrix K, such as the one above for the Gaussian kernel, without the ϕ, how do we know that it arises from an embedding ϕ as the pairwise dot products of $\phi(a_i)$? This is answered in the following lemma.

Lemma 6.2: A matrix K is a kernel matrix (i.e., there is some embedding ϕ such that $k_{i,j} = \phi(a_i) \cdot \phi(a_j)$) if and only if K is positive semi-definite.

Proof: If K is positive semi-definite then it can be expressed as $K = BB^T$. Define $\phi(a_i)$ to be the i[th] row of B. Then $k_{i,j} = \phi(a_i) \cdot \phi(a_j)$. Conversely, if there is an embedding ϕ such that $k_{i,j} = \phi(a_i) \cdot \phi(a_j)$, then putting the $\phi(a_i)$ as the rows of a matrix B, we have that $K = BB^T$ and so K is positive semi-definite.

■

Recall that a function of the form $\sum_{ij} y_i y_j k_{ij} = y^T K y$ is convex if and only if K is positive semi-definite. So the support vector machine problem is a convex program. Thus, we may use any positive semi-definite matrix as our kernel matrix. We will give a few important examples of kernel matrices which are used. The first example is $k_{i,j} = (a_i \cdot a_j)^p$, where p is a positive integer. We prove that this matrix is positive semi-definite. Suppose u is any n-vector. We must show that $u^T K u = \sum_{i,j} k_{i,j} u_i u_j \geq 0$.

$$\sum_{i,j} k_{ij} u_i u_j = \sum_{ij} u_i u_j (a_i \cdot a_j)^p = \sum_{ij} u_i u_j \left(\sum_k a_{ik} a_{jk} \right)^p$$

$$= \sum_{ij} u_i u_j \left(\sum_{k_1, k_2, \ldots k_p} a_{ik_1} a_{ik_2} \ldots a_{ik_p} a_{jk_1} \ldots a_{jk_p} \right) \text{ by expansion.}$$

Note that $k_1, k_2, \ldots k_p$ need not be distinct. Now we exchange the summations and do some simplification to get :

$$\sum_{ij} u_i u_j \left(\sum_{k_1, k_2, \ldots k_p} a_{ik_1} a_{ik_2} \ldots a_{ik_p} a_{jk_1} \ldots a_{jk_p} \right) = \sum_{k_1, k_2, \ldots k_p} \sum_{ij} u_i u_j a_{ik_1} a_{ik_2} \ldots a_{ik_p} a_{jk_1} \ldots a_{jk_p}$$

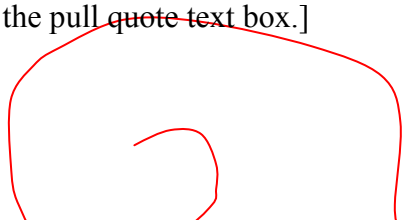
$$= \sum_{k_1, k_2, \ldots k_p} \left(\sum_i u_i a_{ik_1} a_{ik_2} \ldots a_{ik_p} \right)^2 .$$

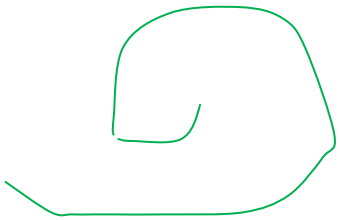
The last term is a sum of squares and thus non-negative proving that K is positive semi-definite.

Example 6.1: Use of the Gaussian Kernel:

Consider a situation where examples are points in the plane on two juxtaposed curves as shown in the diagram (the red curve and the green curve), where points on the first curve are labeled +1 and points on the second are labeled -1. Suppose examples are spaced δ apart on each curve and the minimum distance between the two curves is $\Delta >> \delta$. Clearly, there is no half-space in the plane which classifies the examples correctly. Since the curves intertwine a lot, intuitively, any polynomial which classifies them correctly must be of high complexity. But consider the Gaussian kernel $e^{-|a_i - a_j|^2/\delta^2}$. For this kernel, the K has $k_{ij} \approx 1$ for adjacent points on the same curve and $k_{ij} \approx 0$ for all other pairs of points. Reorder the examples so that we first list in order all examples on the blue curve, then on the red curve. Now, let y_1 and y_2 be the vectors of y for points on the two curves.

[Type a quote from the document or the summary of an interesting point. You can position the text box anywhere in the document. Use the Text Box Tools tab to change the formatting of the pull quote text box.]





K has the block form: $K = \begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix}$, where K_1 and K_2 are both roughly the same size and are both block matrices with 1's on the diagonal and slightly smaller constants on the diagonals one off from the main diagonal and then exponentially falling off with distance from the diagonal.

The SVM is easily seen to be essentially of the form: Min $y_1^T K_1 y_1 + y_2^T K_2 y_2$ subject to $K_1 y_1 \geq 1; K_2 y_2 \leq -1$. This separates into two programs, one for y_1 and the other for y_2 and from the fact that $K_1 = K_2$, the solution will have $y_2 = -y_1$. Further by the structure (essentially the same everywhere except at the ends of the curves), y_1 and y_2 will each have the same entries; so essentially, y_1 will be 1 everywhere and y_2 will be 0 everywhere. The y_i values then provide a nice simple classifier: $l_i y_i > 1$.

6.4 Strong and Weak Learning - Boosting

A strong learner is an algorithm which takes n labeled examples and produces a classifier which correctly labels each of the given examples. Since the learner is given the n examples (and recall that it is responsible for only the training examples given) with their labels, it seems a trivial task – just encode into a table the examples and labels and each time we are asked the label of one of the examples, do a table look-up. But here, we require the Occam's razor principle – the classifier produced by the learner must be (considerably) more concise than a table of the given examples. The time taken by the learner, and the length/complexity of the classifier output are both parameters by which we measure the learner. But now we focus on a different aspect. The word strong refers to the fact that the output classifier must label all the given examples correctly; no errors are allowed.

A weak learner is allowed to make mistakes. It is only required to get a strict majority, namely, $(\frac{1}{2} + \gamma)$ fraction of the examples correct where γ is a positive real number. This seems very weak. But with a slight generalization using a technique called Boosting, we can do strong learning with a weak learner!

Definition: Suppose $U = \{a_1, a_2, \ldots a_n\}$ are n labeled examples. A weak learner is an algorithm which given as input the examples, their labels, and a non-negative real weight w_i on each example a_i, produces a classifier which correctly labels a subset of examples with total weight at least $(\frac{1}{2} + \gamma) \sum_{i=1}^{n} w_i$.

A strong learner can be built by making $O(\log n)$ calls to a weak learner (WL) by a method called Boosting. Boosting makes use of the intuitive notion that if an example was misclassified one needs to pay more attention to it.

Boosting algorithm

Make the first call to the WL with all $w_i = 1$.

At time $t+1$ multiply the weight of each example which was misclassified the previous time by $1+\epsilon$. Leave the other weights as they are.

After T steps**,** stop and output the following classifier:

Label each of the examples $\{a_1, a_2, \ldots, a_n\}$ by the label given to it by a majority of calls to the WL. Assume T is odd, so there is no tie for the majority.

Suppose m is the number of examples the final classifier gets wrong. Each of these m examples was misclassified at least T/2 times so each has weight at least $(1+\epsilon)^{T/2}$. This says the total weight is at least $m(1+\epsilon)^{T/2}$. On the other hand, at time $t+1$, we only increased the weight of examples misclassified the last time. By the property of weak learning, the total weight of misclassified examples is at most $f = (\frac{1}{2} - \gamma)$ of the total weight at time t. So, we have

Total weight at time $t+1$ $\leq$ $\big((1+\epsilon) f + (1-f)\big)$ times total weight at time t

$\leq (1 + \frac{\epsilon}{2} - \gamma\epsilon)$ times total weight at time t.

Thus

$$m(1+\epsilon)^{T/2} \leq \text{Total weight at end} \leq n(1 + \frac{\epsilon}{2} - \gamma\epsilon)^T$$

Taking logs

$$\ln m + \frac{T}{2}\ln(1+\epsilon) \leq \ln n + T\ln(1 + \frac{\epsilon}{2} - \gamma\epsilon).$$

To a first order approximation, $\ln(1+\delta) \approx \delta$ for δ small. So $\ln m \leq \ln n - T\gamma\epsilon$. Make ϵ a small constant (say $\epsilon = 0.01$) and $T = (2 + \ln n)/\gamma\epsilon$, then $m \leq \frac{1}{2}$. Thus the number of misclassified items, m, must be zero.

OTHER EXAMPLES OF BOOSTING AS PROBLEMS ??????????????

6.5 Number of examples needed for prediction: VC-dimension

Training and Prediction

Up to this point, we dealt only with training examples and focused on building a classifier which works correctly on them. Of course, the ultimate purpose is prediction of labels on future examples. In the car verses non-car example, we want our classifier to classify future pictures as car or non-car without human input**.** Clearly, we cannot expect the classifier to predict every example correctly. So we must attach a probability distribution on the space of examples, so that we can measure how good the classifier is based on the probability of misclassification. A fair criterion is to assume the same probability distribution on the space of training examples as on the space of test examples. The strongest results would build a learner without actually knowing this probability distribution.

A second question is how many training examples suffice so that we can assert that as long as a classifier gets all the training examples correct (strong learner), the probability that it makes a prediction error (measured with the same probability distribution) of more than ϵ is less than δ? Ideally, we would like this number to be sufficient whatever the unknown probability distribution is. The theory of VC-dimension will provide an answer to this as we see later.

A Sampling Motivation

The concept of VC-dimension is fundamental and is the backbone of Learning Theory. It is also useful in many other contexts. Our first motivation will be from a database example. Consider a database consisting of the salary and age of each employee in a company and a set of queries of the form: how many individuals between ages 35 and 45 have a salary between 60,000 and 70,000? Each employee is represented by a point in the plane where the coordinates are age and salary. The query asks how many data points fall within an axis-parallel rectangle. One might want to select a fixed sample of the data (before queries arrive) and estimate the number of points in a query rectangle by the number of sample points in the rectangle. At first, such an estimate would not seem to work. Applying a union bound, the probability that it fails to work for some rectangle is at most the product of the probability that it fails for one particular rectangle times the number of possible rectangles. But, there are an infinite number of possible rectangles. So such a simple union bound argument does not work.

Define two axis parallel rectangles to be equivalent if they contain the same data points. If there are n data points, only $O(n^4)$ of the 2^n subsets can correspond to the set of points in a rectangle. To see this, consider any rectangle R. If one of its sides does not pass through one of the n points, then move the side parallel to itself until for the first time, it passes through one of the n points. Clearly, the set of points in R and the new rectangle are the same since we did not ``cross'' any point. By a similar process, modify all four

sides, so that there is at least one point on each of them. Now, the number of rectangles with at least one point on each side is at most $O(n^4)$. (Exercise). The exponent four plays an important role; it will turn out to be the VC-dimension of axis-parallel rectangles.

Let U be any set of n points in the plane. Each point would correspond to one employee's age and salary. Let $\epsilon > 0$ be a given error parameter. Pick a sample S of size s from U uniformly at random with replacement. When a query rectangle R arrives, we estimate $|R \cap U|$ (in the example, this is the number of employees of age between 35 and 45 and salary between 60 and 70K) by the quantity $\frac{n}{s}|R \cap S|$. This is the number of employees in the sample within the ranges scaled up by $\frac{n}{s}$, since we picked a sample of size s out of n. We wish to assert that the fractional error is at most ϵ for every rectangle R, i.e., that

$$\left| |R \cap U| - \frac{n}{s}|R \cap S| \right| \leq \epsilon n \text{ for every } R.$$

Of course, the assertion is not absolute, there is a small probability that the sample is atypical, for example picking no points from a rectangle R which has a lot of points. So we can only assert the above with high probability or that its negation holds with very low probability:

$$\text{Prob}\left(\left| |R \cap U| - \frac{n}{s}|R \cap S| \right| > \epsilon n \text{ for some R} \right) \leq \delta, \tag{6.5.1}$$

where δ >0 is another error parameter. Note that it is very important that our sample S be good for every possible query, since we do not know beforehand which queries will arise.

How many samples are necessary to ensure that (6.5.1) holds? There is a technical point as to whether sampling is with or without replacement. This does not make much difference since s will be much smaller than n. However, we will assume here that the sampling is with replacement. So, we make s independent and identically distributed trials, in each of which we pick one sample uniformly at random from the n. Now, for one fixed R, the number of samples in R is a random variable which is the sum of s independent 0-1 random variables, each with probability of having value 1 equal to

$$\frac{|R \cap U|}{n}.$$

Let $q = \frac{|R \cap U|}{n}$. Now, $|R \cap S|$ has distribution Binomial(s, q). Also,

$$\left| |R \cap U| - \frac{n}{s}|R \cap S| \right| > \epsilon n \quad \equiv \quad \left| |R \cap S| - sq \right| > \epsilon s.$$

So, from **??? Concentration inequality from the Emerging Graph Section or 2.5 of Janson et al.????????),** we have for $0 \leq \epsilon \leq 1$,

$$\text{Prob}\left(\left|\,|R\cap U| - \frac{n}{s}|R\cap S|\,\right| > \epsilon n\right) \le 2e^{-\epsilon^2 s/(3q)} \le 2e^{-\epsilon^2 s/3}$$

Using the union bound and noting that there are only $O(n^4)$ possible sets $R\cap U$ yields

$$\text{Prob}\left(\left|\,|R\cap U| - \frac{n}{s}|R\cap S|\,\right| > \epsilon n \text{ for some R}\right) \le cn^4 e^{-\epsilon^2 s/3},$$

and so setting $s \ge \Omega(\ln n/\epsilon^2)$, we can ensure (6.5.1). In fact, we will see later that even the logarithmic dependence on n can be avoided. Indeed, we will see using VC-dimension that as long as s is at least a certain number depending upon the error ϵ and the VC-dimension of the set of shapes, (6.5.1) will hold.

In another situation, suppose we have an unknown probability distribution P over the plane and ask what is the probability mass $P(R)$ of a query rectangle R? We might estimate the probability mass by first drawing a sample S of size s in s independent and identically distributed trials, in each of which we draw one sample according to P and wish to know how far the sample estimate $|S\cap R|/s$ is from the probability mass $P(R)$. Again, we would like to have the estimate be good for every rectangle. This is a more general problem than the first problem of estimating $|R\cap U|$. To see this, let U consist of n points in the plane and let the probability distribution, P, have value $\frac{1}{n}$ at each of n points. Then $\frac{1}{n}|R\cap U| = P(R)$.

The reason this problem is more general is that there is no simple argument bounding the number of rectangles to $O(n^4)$ - in fact moving the sides of the rectangle is no longer valid, since it could change the probability mass enclosed. Further, P could be a continuous distribution, when the analog of n would be infinite. So an argument as above using the union bound would not solve the problem. The VC-dimension argument which is cleverer will yield the desired result for the more general situation as well.

The question is of interest for shapes other than rectangles as well. Indeed, half-spaces in d dimensions is an important class of ``shapes'', since they correspond to learning threshold gates.

A class of regions such as rectangles has a parameter called VC-dimension and we can bound the probability of the discrepancy between the sample estimate and the probability mass in terms of the VC-dimension of the shapes allowed. That is,

$$|\text{prob mass - estimate}| < \varepsilon$$

with probability $1-\delta$ where δ depends on ε and the VC-dimension.

In summary, we would like to create a sample of the data base without knowing which query we will face, knowing only the family of possible queries (like rectangles). We would like our sample to work well for every possible query from the class.

With this motivation, we introduce VC-dimension and later, we will relate it to learning.

Vapnik-Chervonenkis or VC-dimension

A *set system* $(U, \mathbb{S})$ consists of a set U along with a collection $\mathbb{S}$ of subsets of U. The set U may be finite or infinite. An example of a set system is the set $U = R^2$ of points in the plane, with $\mathbb{S}$ being the collection of all axis parallel rectangles.

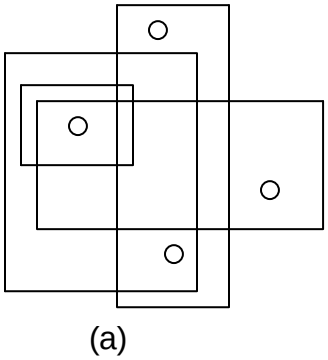
Let $(U, \mathbb{S})$ be a set system. A subset $A \subseteq U$ is *shattered* by $\mathbb{S}$ if each subset of A can be expressed as the intersection of an element of $\mathbb{S}$ with A. The VC-dimension of $(U, \mathbb{S})$ is the maximum size of any subset of U shattered by $\mathbb{S}$.

Examples of Set Systems and Their VC-Dimension

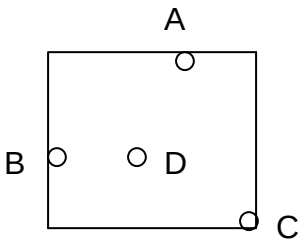
John: As we discussed, the notion of VC dimension is something students can really relate to since shattering is combinatorial. So, so far, we have all the examples that you had in your notes. Some points : Should we change this list by adding or subtracting ? I have not completely edited the examples (though I have made one pass thro them..)

Rectangles with horizontal and vertical edges

There exist sets of four points that can be shattered by rectangles with horizontal and vertical edges. For example, four points at the vertices of a diamond. However, rectangles cannot shatter any set of five points. To see this, find the minimum enclosing rectangle for the five points. For each edge there is at least one point that has stopped its movement. Identify one such point for each edge. The same point maybe identified as stopping two edges if it is at a corner of the minimum enclosing rectangle. If two or more points have stopped an edge designate only one as having stopped the edge. Now, at most four points have been designated. Any rectangle enclosing the designated points must include the undesignated points. Thus the subset of designated points cannot be expressed as the intersection of a rectangle with the five points. Therefore, the VC-dimension of axis parallel rectangles is 4.



(a)



(b)

Figure 6.XXX: (a) shows a set of four points along with some of the rectangles that shatter it. Not every set of four points can be shattered as seen in (b). Any rectangle containing points A, B, and C must contain D. No set of five points can be shattered by rectangles with horizontal and vertical edges. No set of 3 colinear points can be shattered, since any rectangle that contains the two end points must also contain the middle point. More generally, since rectangles are convex, a set with one point inside the convex hull of the others cannot be shattered.

Intervals of the reals

Intervals on the real line can shatter any set of two points but no set of three points since the subset of the first and last points cannot be isolated. Thus, the VC-dimension of intervals is two.

Pairs of intervals of the reals

There exists a set of size four that can be shattered but no set of size five since the subset of first, third and last point cannot be isolated. Thus, the VC-dimension of pairs of intervals is four.

Convex Polygons

Consider the system of all convex polygons in the plane. For any positive integer m, place *m* points on the unit circle. Any subset of the points are the vertices of a convex polygon. Clearly that polygon will not contain any of the points not in the subset. This shows that we can shatter arbitrarily large sets, so the VC-dimension is infinite.

Half spaces in d dimensions

Let $convex(A)$ denote the convex hull of the set of points in A. Define a half space to be the set of all points on one side of a hyper plane, i.e., a set of the form $\{x : a \cdot x \geq a_0\}$. The VC-dimension of half-spaces is $d+1$. We will use the following result from geometry to prove this.

Theorem 6.2 (Radon): Any set $S \subseteq R^d$ with $|S| \geq d+2$, can be partitioned into 2 disjoint subsets A and B such that $convex(A) \cap convex(B) \neq \phi$.

Proof: First consider four points in 2-dimensions. If any three of the points lie on a straight line, then the result is obviously true. Thus, assume that the no three of the points lie on a straight line. Select three of the points. The three points must form a triangle. Extend the edges of the triangle to infinity. The three lines divide plane into seven regions, one finite and six infinite. Place the fourth point in the plane. If the point is placed in the triangle, then it and the convex hull of the triangle intersect. If the fourth point lies in a two sided infinite region, the convex hull of the point plus the two opposite points of the triangle contains the third vertex of the triangle. If the fourth point is in a

three sided region, the convex hull of the point plus the opposite vertex of the triangle intersects the convex hull of the other two points of the triangle.

We prove the general case ``algebraically'' (rather than geometrically). Without loss of generality, assume $|S| = d+2$. Form a $d \times (d+2)$ matrix with one column for each point of S. Add an extra row of all 1's to construct a $(d+1)\times(d+2)$ matrix B. Clearly, since the rank of this matrix is at most $d+1$, the columns are linearly dependent. Say $x = (x_1, x_2, \cdots, x_{d+2})$ is a non-zero vector with $Bx = 0$. Reorder the columns so that $x_1, x_2, \ldots x_s \geq 0$ and $x_{s=1}, x_{s+2}, \ldots x_{d+2} < 0$. Let B_i (respectively A_i) be the i^{th} column of B (respectively A). Then, we have

$\sum_{i=1}^{s} |x_i| B_i = \sum_{i=s+1}^{d+2} |x_i| B_i$ from which we get

$\sum_{i=1}^{s} |x_i| A_i = \sum_{i=s+1}^{d+2} |x_i| A_i$ and $\sum_{i=1}^{s} |x_i| = \sum_{i=s+1}^{d+2} |x_i|$.

Let $\sum_{i=1}^{s} |x_i| = a$. Then, $\sum_{i=1}^{s} \frac{|x_i|}{a} A_i = \sum_{i=s+1}^{d+2} \frac{|x_i|}{a} A_i$. Each side of this equation is a convex combination of columns of A which proves the theorem.

■

Radon's theorem immediately implies that half-spaces in d dimensions do not shatter any set of $d+2$ points. Divide the set of *d+2* points into sets A and B as in Theorem 6.2 where $convex(A) \cap convex(B) \neq \phi$. Suppose that some half space separates *A* from *B*. Then the half space contains *A* and the complement of the half space contains *B*. This implies that the half space contains the convex hull of *A* and the complement of the half space contains the convex hull of *B*. Thus, $convex(A) \cap convex(B) = \phi$ contradicting Radon's Theorem. Therefore, no set of *d+2* points can be shattered by half planes in *d* dimensions.

There exists a set of size *d+1* that can be shattered by half spaces. Select the d unit coordinate vectors plus the origin to be the *d+1* points. Suppose A is any subset of these $d+1$ points. Without loss of generality assume that the origin is in A. Take a 0-1 vector a which has 1's precisely in the coordinates corresponding to vectors not in *A*. Then clearly *A* lies in the half-space $a \cdot x \leq 0$ and the complement of *A* lies in the complementary half-space.

■

Hyperspheres in d-dimensions

A *hypersphere* or ball in d space is a set of points of the form $\{x : |x - x_0| \leq r\}$. The VC-dimension of balls is *d+1*, namely it is the same as that of half spaces. First, we prove that no set of $d+2$ points can be shattered by balls. Suppose some set S with $d+2$ points can be shattered. Then for any partition A_1, A_2 of S, there are balls B_1 and B_2 such that $B_1 \cap S = A_1$ and $B_2 \cap S = A_2$. Now B_1 and B_2 may intersect, but there is no point of S in

their intersection. It is easy to see then that there is a hyper plane with all of A_1 on one side and all of A_2 on the other and this implies that half spaces shatter S, a contradiction. Therefore no $d+2$ points can be shattered by balls.

It is also not difficult to see that the set of $d+1$ points consisting of the unit vectors and the origin can be shattered by balls. Suppose A is a subset of the $d+1$ points. The center a_0 of our ball will be the sum of the vectors in A.

For every unit vector in A, its distance to this center will be $\sqrt{|A|-1}$ and for every unit vector outside A, its distance to this center will be $\sqrt{|A|+1}$. The distance of the origin to the center is $\sqrt{|A|}$. Thus it is easy to see that we can choose the radius so that precisely the points in A are in the ball.

Finite sets

The system of finite sets of real numbers can shatter any finite set of points on the reals and thus the VC dimension of finite sets is infinite.

Intuitively the VC-dimension of a collection of sets is often closely related to the number of free parameters.

Shape	VC-dimension	Comments
Interval	2	
2 intervals	4	
Rectangle with horizontal and vertical edges	4	
Rectangle rotated	7	
Square h&v	3	
Square rotated	5	
Triangle		
Right triangle		
Circle	3	
Convex polygon	∞	
Corner		

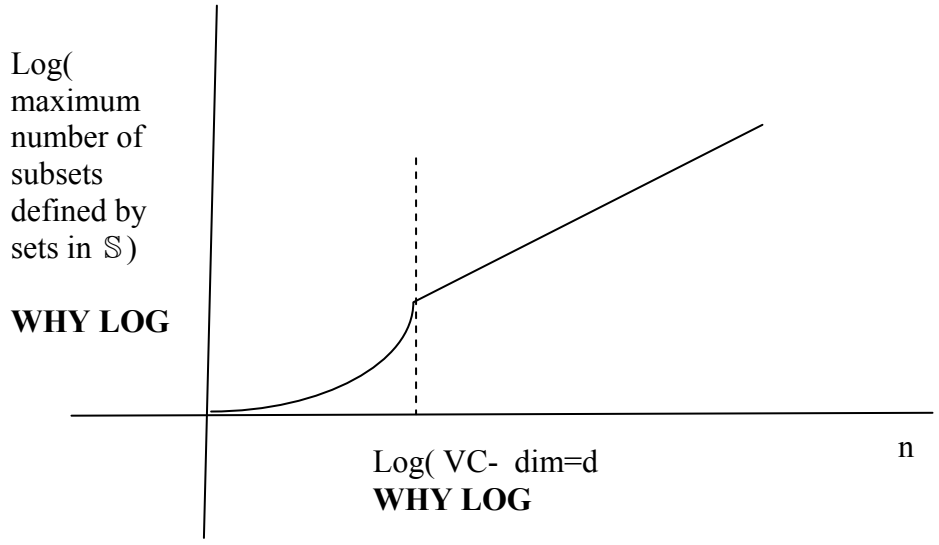
The shatter function

Consider a set system $(U,\mathbb{S})$ of finite VC dimension *d*. For $n \le d$ there exists a subset $A \subseteq U$, $|A| = n$, such that A can be shattered into 2^n pieces. This raises the question for $|A| = n$, $n > d$ as to what is the maximum number of subsets of A that can be expressed by $S \bigcap A$ for $S \in \mathbb{S}$. We shall see that this maximum number is at most a polynomial in *n* with degree d.

The *shatter function* $\pi_{\mathbb{S}}(n)$ of a set system $(U,\mathbb{S})$ is the maximum number of subsets that can be defined by the intersection of sets in $\mathbb{S}$ with some n element subset A of U. Thus

$$\pi_{\mathbb{S}}(n) = \max_{\substack{A \subseteq U \\ |A|=n}} |\{A \bigcap S \mid S \in \mathbb{S}\}|$$

For small values of n, $\pi_{\mathbb{S}}(n)$ will grow as 2^n. Once *n* equals the VC-dimension of $\mathbb{S}$, it grows more slowly. The definition of VC-dimension can clearly be reformulated as $\dim(\mathbb{S}) = \max\{n \mid \pi_{\mathbb{S}}(n) = 2^n\}$. Curiously, as we shall see, the growth of $\pi_{\mathbb{S}}(n)$ must be either polynomial or exponential in *n*. If the growth is exponential, then the VC-dimension of S is infinite.



Examples of set systems and their shatter function.

Example 6.1: Half spaces and circles in the plane have VC-dimension 3. So, their primal shatter function is 2^n for n=1, 2 and 3. For n>3, their primal shatter function grows as a polynomial in n. Axes parallel rectangles have VC-dimension 4 and thus their primal shatter function is 2^n for n=1,2, 3, and 4. For n>4, their primal shatter function grows as a polynomial in n.

■

We already saw that for axes-parallel rectangles in the plane, there are at most $O(n^4)$ possible subsets of an n element set that arise as intersections with rectangles. The argument was that we can move the sides of the rectangle until each side is ``blocked'' by one point. We also saw that the VC-dimension of axes-parallel rectangles is 4. We will see here that the two fours, one in the exponent of n and the other the VC-dimension, being equal is no accident. There is another four related to rectangles, that is, it takes four parameters to specify an axis parallel rectangle. This latter four is a coincidence.

Shatter Function for Set Systems of Bounded VC-Dimension

We shall prove that for any set system $(U,\mathbb{S})$ of VC-dimension *d* that the quantity

$$\sum_{i=0}^{d}\binom{n}{i}=\binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{d}\le n^d$$

bounds the shatter function $\pi_{\mathbb{S}}(n)$. That is, the formula $\sum_{i=0}^{d}\binom{n}{i}$ bounds the number of subsets of any *n* point subset of *U* that can be expressed as the intersection with a set of $\mathbb{S}$. Thus, the shatter function $\pi_{\mathbb{S}}(n)$ is either 2^n if d is infinite or it is bounded by a polynomial of degree *d*.

Lemma 6.3: For any set system $(U,\mathbb{S})$ of VC-dimension at most d, $\pi_{\mathbb{S}}(n)\le\sum_{i=0}^{d}\binom{n}{i}$ for all *n*.

Proof: The proof is by induction on both d and n. The base case will handle all pairs (d,n) with either $n\le d$ or $d=0$. The general case (d,n) will use the inductive assumption on the cases $(d-1,n-1)$ and $(d,n-1)$.

For $n\le d$, $\sum_{i=0}^{d}\binom{n}{i}=2^n$ and $\pi_{\mathbb{S}}(n)\le 2^n$. For $d=0$, note that a set system $(U,\mathbb{S})$ can have at most one set in $\mathbb{S}$ otherwise there would exist a set *A* of cardinality one that could be shattered. If $\mathbb{S}$ contains only one set, then $\pi_{\mathbb{S}}(n)=1$ for all n.

Consider the case for general *d* and *n*. Fix a subset A of U with $|A| = n$. We could just assume that $U = A$; replace each set S in $\mathbb{S}$ by $S \cap A$ for this purpose and remove duplicates – i.e., if for $S_1, S_2 \in \mathbb{S}$ have $S_1 \cap A = S_2 \cap A$, only keep one of them. Note that this does not increase the VC-dimension of $\mathbb{S}$. Now $|U| = n$ and we need to get an upper bound on just $|\mathbb{S}|$.

Remove any element $u \in U$ from the set U and from each set in $\mathbb{S}$. Consider the set system $\mathbb{S}_1 = \left(U - \{u\}, \{S \setminus \{u\} : S \in \mathbb{S}\}\right)$. For $S \subseteq A \setminus \{u\}$, if exactly one of S and $S \cup \{u\}$ is in $\mathbb{S}$, then that set contributes one set to the set system $\mathbb{S}_1$ whereas, if both S and $S \cup \{u\}$ are in $\mathbb{S}$, then both sets contribute to the set system $\mathbb{S}_1$; we eliminate duplicates and keep only one copy in this case. But there were two sets, $S \cap A$ and $S \cup \{u\} \cap A$. To account for this, define another set system

$$\mathbb{S}_2 = \left(U - \{u\}, \left\{S \mid \text{both } S \text{ and } S \bigcup \{u\} \text{ are in } \mathbb{S}\right\}\right).$$

Thus, we have

$$|\mathbb{S}| \quad \leq \quad |\mathbb{S}_1| + |\mathbb{S}_2| = \pi_{\mathbb{S}_1}(n-1) + \pi_{\mathbb{S}_2}(n-1).$$

WHAT IS $|\mathbb{S}|$?

We make use of two facts about VC dimension. If the set system $(U, \mathbb{S})$ with $|U| = n$ has VC dimension *d* then

(1) $\mathbb{S}_1$ has dimension at most *d*, and

(2) $\mathbb{S}_2$ has dimension at most $d-1$.

(1) follows because if $\mathbb{S}_1$ shatters a set of cardinality $d+1$, then $\mathbb{S}$ also would shatter that set producing a contradiction. (2) follows because if $\mathbb{S}_2$ shattered a set $B \subseteq U \setminus \{u\}$, then $B \cup \{u\}$ would be shattered by $\mathbb{S}$, again producing a contradiction if $|B| \geq d$.

By the induction hypothesis applied to $\mathbb{S}_1$, we have $|\mathbb{S}_1| \leq \pi_{\mathbb{S}_1}(n-1) \leq \sum_{i=0}^{d} \binom{n-1}{i}$. By the induction hypotheses applied to $\mathbb{S}_2$, (with $d-1, n-1$), we have $|\mathbb{S}_2| \leq \sum_{i=0}^{d-1} \binom{n-1}{i}$.

Since $\binom{n-1}{d-1} + \binom{n-1}{d} = \binom{n}{d}$ and $\binom{n-1}{0} = \binom{n}{0}$

$$\pi_{\mathbb{S}}(n) \le \binom{n-1}{0}+\binom{n-1}{1}+\cdots+\binom{n-1}{d}+\binom{n-1}{0}+\binom{n-1}{1}+\cdots+\binom{n-1}{d-1}$$
$$\le \binom{n-1}{0}+\left[\binom{n-1}{1}+\binom{n-1}{0}\right]+\cdots+\left[\binom{n-1}{d}+\binom{n-1}{d-1}\right]$$
$$\le \binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{d}$$

■

Intersection Systems

Let $(U,\mathbb{S}_1)$ and $(U,\mathbb{S}_2)$ be two set systems on the same underlying set U. Define another set system (called the intersection system) $(U,\mathbb{S}_1 \cap \mathbb{S}_2)$, where $\mathbb{S}_1 \cap \mathbb{S}_2 = \{A \cap B : A \in \mathbb{S}_1 ; B \in \mathbb{S}_2\}$. In words, we take the intersections of every set in $\mathbb{S}_1$ with every set in $\mathbb{S}_2$. A simple example is $U = R^d$ and $\mathbb{S}_1 = \mathbb{S}_2 =$ the set of half spaces. Then $\mathbb{S}_1 \cap \mathbb{S}_2$ consists of all sets defined by the intersection of two half spaces. This corresponds to taking the Boolean AND of the output of two threshold gates and is the most basic neural net besides a single gate. We can repeat this process and take the intersection of k half spaces. The following simple lemma helps us bound the growth of the primal shatter function as we do this.

Lemma 6.4: Suppose $(U,\mathbb{S}_1)$ and $(U,\mathbb{S}_2)$ are two set systems on the same set U. Then $\pi_{\mathbb{S}_1 \cap \mathbb{S}_2}(n) \le \pi_{\mathbb{S}_1}(n)\pi_{\mathbb{S}_2}(n)$.

Proof: The proof follows from the fact that for any $A \subseteq U$, the number of sets of the form $A \cap (S_1, S_2)$ with $S_1 \in \mathbb{S}_1$ and $S_2 \in \mathbb{S}_2$ is at most the number of sets of the form $A \cap S_1$ times the number of sets of the form $A \cap S_2$.

6.6 The VC- theorem

Suppose we have a set system $(U,\mathbb{S})$ with an unknown probability distribution $p(x)$ over the elements of U. Suppose $u_1, u_2, \ldots u_n$ are n independent and identically-distributed samples, each drawn according to the probability distribution p. How large a number of examples is needed to ``represent'' each set $S \in \mathbb{S}$ correctly, in the sense that the proportion of samples in S is a good estimate of the probability mass of S. If the number of samples necessary to represent each set correctly is not too large, then the set U can be replaced by the set of samples in computations. Recall the example of the database of employees with rectangle queries. There $U = R^2$ and $\mathbb{S}$ consisted of all rectangles. Recall also that in Eq. 6.3.1, we had to bound the probability that the sample works for all rectangles.

The VC- theorem answers the question as to how large n needs to be by proving a bound on n in terms of the VC-dimension of the set system. We first prove a simpler version of the theorem, which illustrates the essential points. In the simpler version, we want to assert that with high probability, every set S in the set system with probability $p(S) \geq \epsilon$ gets represented at least once in the sample. The general VC theorem asserts that every set S gets represented by a number of samples proportional to the probability mass of the set. There is one technical point. We will use sampling with replacement. So, the ``set'' of samples we pick, $\{u_1, u_2, \ldots u_n\}$, is really a multi-set. However, to make the proof easier to understand we use the term set rather then multiset.

Theorem 6.3: (Simple version of Vapnik-Chervonenkis): Suppose $(U, \mathbb{S})$ is a set system with VC dimension d. Let p be any probability distribution on U. Let ϵ be between 0 and 1. Suppose $m \in \Omega\left(\frac{d}{\epsilon}\log\frac{d}{\epsilon}\right)$. The probability that a set T of m samples picked from U according to p does not intersect some $S \in \mathbb{S}$ with $p(S) \geq \epsilon$ is at most $2^{-\epsilon m/4}$.

Proof: If one is concerned with only one set S with probability $p(S) \geq \varepsilon$ and picked m samples, then with high probability one of the samples will lie in S. However, $\mathbb{S}$ may have infinitely may S and thus even a small probability of all the samples missing a specific S translates into the samples almost surely missing some S in $\mathbb{S}$. To resolve this problem, select the set of points T and let S_0 be a set missed by all points in T. Select a second set of points T'. With high probability some point in the second set T' will lie in S_0. This follows, since with the second set of points, we are concerned with only one set, S_0. With high probability, $\frac{1}{m}|T' \cap S_0|$ is a good estimate of $P(S_0)$ and by a Chernoff bound $|S_0 \cap T'| \geq \frac{\epsilon}{2}m$ with probability at least ½.

WOULD T_1 AND T_2 BE BETTER OTATION? IF SO WHAT ABOUT E_1 AND E_2?

Let E be the event that there exists an S with $p(S) \geq \varepsilon$ and all points in T miss S. Let E' be the event that in addition to event E, T' intersects S in at least $\frac{\varepsilon m}{2}$ points. That is,

$$E' : \exists S \in \mathbb{S} \text{ with } \mathrm{p}(S) \geq \epsilon, |S \cap T| = \varnothing, \text{and } |S \cap T'| \geq \frac{\epsilon}{2}m.$$

Since $|S \cap T'| \geq \frac{\epsilon}{2}m$ with probability at least ½, $\mathrm{Prob}(E' \mid E) \geq \frac{1}{2}$. Thus

$$\mathrm{Prob}(E') = \mathrm{Prob}(E' \mid E)\mathrm{Prob}(E) + \mathrm{prob}(E' \mid \neg E)\mathrm{Prob}(\neg E)$$

$$\geq \mathrm{Prob}(E' \mid E)\mathrm{Prob}(E) \geq \frac{1}{2}\mathrm{Prob}(E).$$

This equation allows one to upper bound Prob(E) by upper bounding Prob(E').

The $\text{Prob}(E')$ is bounded by a double sampling technique. Instead of picking T and then T', pick a set W of $2m$ samples. Then pick a subset of size m out of W without replacement to be T and let $T' = W \setminus T$. It is easy to see that the distribution of T and T' obtained this way is the same as picking T and T' directly.

The double sampling technique bounds the probability that m samples will miss S completely when another m samples will surely hit S. Now if E' occurs, then for some $S \in \mathbb{S}$, with $p(S) \geq \epsilon$, we have both $|S \cap T| = 0$ and $|S \cap T'| \geq \frac{\epsilon}{2} m$. Since $|S \cap T'| \geq \frac{\epsilon}{2} m$ and $T' \subseteq W$, it follows that $|S \cap W| \geq \frac{\epsilon}{2} m$. But if $|S \cap W| \geq \frac{\epsilon}{2} m$ and T is a random subset of cardinality m out of W, the probability that $|S \cap T| = 0$ is at most

$$\frac{\binom{2m-(\epsilon/2)m}{m}}{\binom{2m}{m}} \leq \frac{m(m-1)\ldots(m-\frac{\epsilon}{2}m+1)}{(2m)(2m-1)\ldots(2m-\frac{\epsilon}{2}m+1)} \leq 2^{-\frac{\epsilon m}{2}}.$$

This is the failure probability for just one S. We would like to use the union bound for all $S \in \mathbb{S}$, but there may be a large number or even infinitely many of them. But note that we need only consider $W \cap S, S \in \mathbb{S}$. It is important to understand that W is fixed and then we make the random choices to select T out of W. The number of possible $W \cap S$ is at most $\pi_{\mathbb{S}}(2m)$ which from Lemma 6.3 is at most $(2m)^d$. So the present theorem follows since with $m \in \Omega\left(\frac{d}{\epsilon}\log\frac{d}{\epsilon}\right)$, with a suitable constant independent of d and ϵ, we have $\epsilon m > 4d \log_2(2m)$. (Checking this is left as an exercise.)

■

Next we prove the general VC-theorem. Whereas the special theorem proved in the last section asserted that for every S with $p(S) \geq \epsilon$, our sample set T has at least one element of S, the general theorem will assert that we can estimate $p(S)$ for any set $S \in \mathbb{S}$ by $|T \cap S| / |T|$ making an error of less than ϵ. Recall that we will be doing sampling with replacement, so T is really a multi-set.

In the proof of the following theorem it will be useful to have a verbal description for certain events. For a set T of samples and an $S \in \mathbb{S}$,

if $\left|\frac{|S \cap T|}{|T|} - p(S)\right| > \epsilon$, we say that ``$T$ estimates $p(S)$ badly'', and

if $\left|\frac{|S\cap T|}{|T|}-p(S)\right|\le\frac{\epsilon}{2}$, we say that ``$T$ estimates S very well''.

Theorem 6.4: (General version of the VC-Theorem) Let $(U,\mathbb{S})$ be a set system with VC-dimension d and let p be any probability distribution on U. For any $\epsilon\in[0,1]$, if $n=\Omega\left(\frac{d}{\epsilon^2}\log(d/\epsilon)\right)$, and T is a set of n independent samples drawn from U according to p, then

$$\text{Prob}\left(\exists S\in\mathbb{S}:\left|\frac{|S_0\cap T|}{n}-P(S_0)\right|>\epsilon\right)\le 2e^{-\epsilon^2 n/6}.$$

WHY SUBSCRIPT 0?

Proof: Pick an auxiliary sample T' of size $m=4n/\epsilon$. Let E be the event that there exists $S\in\mathbb{S}$ such that $\left|\frac{|S\cap T|}{|T|}-P(S)\right|>\epsilon$.

Define another event E' that says there is a set $S\in\mathbb{S}$ for which T estimates $p(S)$ badly, but T' estimates $p(S)$ very well. We can again argue that $\text{Prob}(E')\ge(1/2)\text{Prob}(E)$. The idea of introducing E' is that it is the AND of two contradictory events, so its probability will be relatively easy to upper bound.

Again use double sampling. Pick a set W of cardinality $n+m$, then pick a random subset T from W of cardinality m. Assume now E' happens and for an $S_0\in\mathbb{S}$, T estimates $p(S_0)$ badly, but T' estimates $p(S_0)$ very well. We denote this event happening for S_0 by $E'(S_0)$. It follows from the fact that T' estimates $p(S_0)$ very well that $W=T\cup T'$ estimates $p(S_0)$ ``moderately well'' since T' is larger than T and so T 's corruption does not cost us much. More precisely, we have

$$\begin{aligned}|W\cap S_0| &\ge|T'\cap S_0|\ge mp(S_0)-\frac{\epsilon m}{2}\Rightarrow\\ \frac{|W\cap S_0|}{m+n} &\ge\frac{m}{m+n}p(S_0)-\frac{\epsilon}{2}\ge p(S_0)-\frac{n}{m+n}-\frac{\epsilon}{2}\ge p(S_0)-\frac{3\epsilon}{4}.\end{aligned}$$

and we also have

$$\begin{aligned}|W\cap S_0| &\le|T'\cap S_0|+|T|\le mp(S_0)+\frac{\epsilon m}{2}+n\\ \frac{|W\cap S_0|}{m+n} &\le\frac{m}{m+n}p(S_0)+\frac{\epsilon}{2}+\frac{n}{m+n}\le p(S_0)+\frac{3\epsilon}{4}.\end{aligned}$$

Thus, together, these yield

$$\left|\frac{|W \cap S_0|}{m+n} - P(S_0)\right| \leq \frac{3\epsilon}{4} \tag{22}$$

We know that T estimates $p(S_0)$ badly. By the double sampling argument, T is just a uniform random subset of W of cardinality n and W estimates $p(S_0)$ moderately well. We will use these facts to show that the probability of E' cannot be big. We use the double sampling technique. For the moment, assume that T is picked out of W with replacement. The probability that each trial picks an element of S_0 is $\frac{|W \cap S_0|}{m+n}$. Thus, the distribution of $|T \cap S_0|$ is Binomial$(n, \frac{|W \cap S_0|}{m+n})$. Since $\epsilon \in [0,1]$ from the concentration inequality from the Emerging Graph Section or 2.5 of Janson et al we have

$$\text{Prob}\left(\left||T \cap S_0| - n\frac{|W \cap S_0|}{m+n}\right| > \frac{\epsilon n}{4}\right) \leq 2e^{-\epsilon^2 n/3}.$$

Using (22) and the assumption that T estimates $p(S_0)$ badly, we get that under the event $E'(S_0)$, $\left||T \cap S_0| - n\frac{|W \cap S_0|}{m+n}\right| > \frac{\epsilon n}{4}$ and so by the above, we have the desired upper bound that $\text{Prob}(E'(S_0)) \leq 2e^{-\epsilon^2 n/3}$. The bound is just for one S_0. We need to apply the union bound over all possible S_0; but clearly, the only candidates for S_0 that need to be consider are sets of the form $S \cap W, S \in \mathbb{S}$ of which there are at most $\pi_{\mathbb{S}}(n+m)$ which is at most $(n+m)^d$. Since $m = \frac{4}{\epsilon}n$, we have that $(n+m)^d \leq (8n)^d / \epsilon^d$. Now for $n \in \Omega\left(\frac{d}{\epsilon^2}\ln(d/\epsilon)\right)$, with a suitable constant, it is easy to see by a calculation that $\frac{\epsilon^2 n}{6} \geq d\ln(8n/\epsilon)$, whence it follows that

$$\text{Prob}(E'(S_0)) \text{ (for one } S_0\text{) times (number of possible } S_0\text{)} \leq 2e^{-\epsilon^2 n/6}$$

proving the theorem.

6.7 Priors and Bayesian Learning

Section to be written

SHOULD WE INCLUDE A SECTION ON OVERFITTING?

Exercises

Exercise 6.1: (Boolean OR has a linear separator.) Take as examples all the 2^d elements of $\{0,1\}^d$. Label the example by +1 if there is at least one coordinate with a +1 and label it by -1 if all its coordinates are 0. This is like taking the Boolean OR, except we look upon the coordinates as real numbers. Show that there is a linear separator for these labeled examples. Show that we can achieve a margin of $\Omega(1/d)$ for this problem.

Exercise 6.2: Similar to previous exercise : deal with the AND function.

Exercise 6.3: Similar to previous for majority and minority functions.

Exercise 6.4: Show that the parity function, the Boolean function that is 1 if and only if an odd number of inputs is 1, cannot be represented as a threshold function.

Exercise 6.5: Suppose we were lucky and the starting w made an angle of 45° with a w^* whose margin is δ. Would you be able to conclude that the number of iterations satisfies a smaller upper bound than $\frac{1}{\delta^2}-1$ either directly or with a small modification ?

Exercise 6.6: The proof of Theorem 6.1 shows that for every w^*, with $l_i(w^*\cdot a_i)\geq\delta$ for $i=1,2,\ldots n$, the cosine of the angle between w and w^* is at least $\sqrt{t+1}\delta$ after t iterations. What happens if there are multiple w^*, all satisfying $l_i(w^*\cdot a_i)\geq\delta$ for $i=1,2,\ldots n$? Then, how can our one *w* make a small angle with all of these w^* ?

Exercise 6.7: Suppose examples are points with 0,1 coordinates in *d*-space and the label is +1 if and only if the least i for which $x_i=1$ is odd. Otherwise the label on the example is -1. Show that the rule can be represented by the linear threshold function

$$(x_1,x_2,\cdots,x_n)\left(1,-\tfrac{1}{2},\tfrac{1}{4},-\tfrac{1}{8},\cdots\right)^T = x_1-\frac{1}{2}x_2+\frac{1}{4}x_3-\frac{1}{8}x_4+\ldots\geq 0$$

Exercise 6.8: (Hard) Can the above problem be represented by any linear threshold function with margin at least $1/f(d)$ where $f(d)$ is bounded above by a polynomial function of d ? Prove your answer.

Exercise 6.9: Modify the Perceptron Learning as follows: Starting with $w=(1,0,0,\ldots,0)$, repeat until $(w\cdot a_i)l_i>0$ for all i : Add to w the average of all $a_i l_i$ with $(w\cdot a_i)l_i\leq 0$.

Show that this is a ``noise-tolerant'' version ????????????? MORE DETAIL ??????????

Exercise 6.10: Let w be the weight vector of a linear separator with margin δ and write $v = \frac{w}{\delta |w|}$. Show that $|v|$ and $|v|^2$ are convex functions of the coordinates of w. Find the gradient of $|v|^2$ and show that it exists everywhere. How about the gradient of |v| ?

Exercise 6.11: Show that the above can be reformulated as the unconstrained minimization of

$$|v|^2 + c\sum_i \left(1 - l_i (v \cdot a_i)\right)^+ .$$

Show that x^+ is a convex function. But the function x^+ does not have a derivative at 0. The function $(x^+)^2$ is smoother (its first derivative at 0 exists) and it is better to minimize

$$|v|^2 + c\sum_i \left(\left(1 - l_i (v \cdot a_i)\right)^+\right)^2 .$$

Exercise 6.12: Assume that the center of Figure 2 is (0,0) and the side of each small square is of length 1. Show that a point has label +1 if and only if

$$(x_1 + 1)x_1(x_1 - 1)(x_2 + 1)x_2(x_2 - 1) \geq 0;$$

consider only examples which are interior to a small square.

Exercise 6.13: Prove the the number of of monomials in the polynomial *p* is at most

$$\sum_{D'=0}^{D} \binom{d + D' - 1}{d - 1}.$$

Then prove that $\sum_{D'=0}^{D} \binom{d + D' - 1}{d - 1} \leq D(d + D)^{\min(d-1, D)}$.

Exercise 6.14: Prove that any matrix K where K_{ij} has a power series expansion $K_{ij} = \sum_{p=0}^{\infty} c_p (a_i \cdot a_j)^p$ with all $c_p \geq 0$ and the series is uniformly convergent is positive semi-definite.

Exercise 6.15: Show that the **Gaussian kernel** $K_{ij} = e^{-|a_i - a_j|^2/(2\sigma^2)}$ is positive semi-definite for any value of σ.

Exercise 6.16: In the case when $n = 2$ and $d = 1$, produce a polynomial $p(x, y)$ (whose arguments x,y are just real numbers) and a pair of reals a_1, a_2 so that the matrix $K_{ij} = p(a_i, a_j)$ is not positive semi-definite.

Exercise 6.17: Make the above argument rigorous by using inequalities instead of first order approximation – prove that $T = \frac{3 + \ln n}{\gamma \epsilon} + \frac{1}{\epsilon^2}$ will do for $\epsilon < 1/8$ …CHECK????

Exercise 6.18: Experts picking stocks: Suppose there are n experts who are predicting whether one particular stock is going up or down at each of t time periods. [There are only two outcomes – up/down at each time.] Without knowing in advance their predictions or any other information, can we pick as well as the best expert ? Or nearly as well as the best expert ? Indeed, we will show that boosting can help us predict essentially within a factor of 2 of the best expert ! The idea is as follows : First start with a weight of 1 on each expert. Assume each expert predicts +1 for up or -1 for down. Our prediction will be +1 if the weighted sum of the experts' prediction is non-negative and -1 otherwise. After making the prediction, we find out the true outcome (of the market that day) – up or down. Then we modify the weights as follows : each expert who predicted correctly has his/her weight multiplied by $1+\epsilon$. Show that the number of mistakes we make through time t is at most $\frac{\log t}{\epsilon} + c$ (number of mistakes made by the best expert) by using an argument similar to the one above : Argue an upper bound on the total weight at time t of all experts based on the number of mistakes we make. [Further Hint: If we make m mistakes, then show that the total weight at the end is at most $n(1+\epsilon)^{T-m}(1+\frac{\epsilon}{2})^m$. Also argue a lower bound on the weight of each expert at time t based on the number of mistakes he/she makes. Compare these two.

Exercise 6.19: What happens if instead of requiring

$$\text{Prob}\left(\left|\,|R\cap U| - \frac{n}{s}|R\cap S|\,\right| \le \epsilon n \text{ for every R}\right) \ge 1-\delta,$$

one requires only:

$$\text{Prob}\left(\left|\,|R\cap U| - \frac{n}{s}|R\cap S|\,\right| \le \epsilon n\right) \ge 1-\delta, \quad \text{for every R ?}$$

Exercise 6.20: Suppose we have n points in the plane and C is a circle containing at least three points. Show that there is a circle C' so that (i) there are 3 points lying on C' or two points lying on a diameter of C' and (ii) the set of points in C is the same as the set of points in C'.

■

Exercise 6.21: Given n points in the plane define two circles as equivalent if they enclose the same set of points. Prove that there are only $O(n^3)$ equivalence classes of points defined by circles and thus only n^3 subsets out of the 2^n subsets can be enclosed by circles.

Exercise 6.22: Consider 3-dimensiional space.
(a) What is the VC-dimension of rectangular boxes with axes parallel sides?
(b) What is the VC-dimension of spheres?

Exercise 23: Show that seven points of a regular septagon are separatable by rotated rectangles.

Exercise 24: Prove that no set of eight points can be separated by rotated rectangles.

Exercise 25: Show that the VC dimension of arbitrary right triangles is seven.

Exercise 26: Show that the VC dimension of arbitrary triangles is seven.

Exercise 27: Show that the VC dimension of axes-aligned right triangles with the right angle in the lower left corner is four.

Exercise 28: Prove that the VC dimension of 45°, 45°, 90° triangles with right angle in the lower left is four.

Exercise29: Prove that no set of six points can be shattered by squares in arbitrary position.

Exercise 30: Prove that the VC dimension of convex polygons is infinite.

Exercise 31: Create list of simple shapes for which we can calculate the VC-dimension.

Exercise 32: If a class contains only convex sets prove that no set in which some point is in the convex hull of other points can be shattered.

Exercise 33: (Open) What is the relationship if any between the number of parameters defining a shape and the VC-dimension of the shape.

Exercise 34: (Squares) Show that there is a set of 3 points which can be shattered by axis-parallel squares. Show that the system of axis-parallel squares cannot shatter any set of 4 points.

Exercise 35: Square in general position: Show that the VC-dimension of (not necessarily axes-parallel) squares in the plane is 4.

Exercise 36: Show that the VC-dimension of (not necessarily axes-parallel) rectangles is 7.

Exercise 37: What is the VC-dimension of triangles? Right triangles?

Exercise 38: What is the VC-dimension for a corner? I.e. all points (x,y) such that either

(1) $(x - x_0, y - y_0) \geq (0,0)$,

(2) $(x_0 - x, y - y_0) \geq (0,0)$,

(3) $(x_0 - x, y_0 - y) \geq (0,0)$, or

(4) $(x - x_0, y_0 - y) \geq (0,0)$

for some (x_0, y_0).

Exercise 39: For large *n*, how should you place n points on the plane so that the maximum number of subsets of the n points are defined by rectangles? Can you achieve 4n subsets of size 2? Can you do better? What about size 3? What about size 10?

Exercise 40: Intuitively define the most general form of a set system of VC dimension one. Given an example of such a set system that can generate n subsets of an n element set.

Exercise 41: (Hard) We proved that if the VC dimension is small, then the shatter function is small as well. Prove a sort of converse to this ??????????????

Exercise 42: If $(U, \mathbb{S}_1), (U, \mathbb{S}_2), \ldots (U, \mathbb{S}_k)$ are k set systems on the same ground set U show that $\pi_{\mathbb{S}_1 \cap \mathbb{S}_2 \cap \ldots \mathbb{S}_k}(n) \leq \pi_{\mathbb{S}_1}(n)\pi_{\mathbb{S}_2}(n)\ldots\pi_{\mathbb{S}_k}(n)$.

Exercise43: What does it mean to shatter the empty set? How many subsets does one get?

Exercise 44: In the proof of the simple version of Vapnik-Chervonenkis theorm we claimed that if $P(S_0) \geq \epsilon$ and we selected m elements of U for T that $|S_0 \cap T'| \geq \frac{\epsilon}{2} m$ was at leat ½. Write out the details of the proof of this statement.

Exercise 45: Show that in the ``double sampling'' procedure, the probability of picking a pair of multi-sets T and T', each of cardinality m, by first picking T and then T' is the same as picking a W of cardinality 2m and then picking uniformly at random a subset T out of W of cardinality m and leting T' be $W \setminus T$. For this exercise, assume that P, the underlying probability distribution is discrete.