VC Dimension

November 8, 2005

We introduce here the concepts of risk and VC dimension.

1 Risk

For this discussion remember that we are interested, based on a set of training points (binary classification) to find that function, from among a class of functions which best "behaves" on this set of training points.

More precisely, given the training set $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ where y_i are labels $\{+1, -1\}$, we are interested in functions of the form $h_a(x)$ such that $h_a(x_i) = y_i$ or in any case those functions which minimize the quantity

$$R_{emp} = \frac{1}{2m} \sum_{i=1}^{i=m} |y_i - h_a(x)| \tag{1}$$

Here R_{emp} means the *empirical risk*, that is, the risk, or error based on the training set, rather than the (expected) risk, defined as

$$R(a) = \int \frac{1}{2} |y - h_a(x)| dP(x, y) \tag{2}$$

where P(x,y) denotes an (unknown) probability distribution from which the data are drawn. This means that it is assumed that the data (training and test) are all iid, that is, all drawn from the same probability distribution.

In R_{em} the quantity $\frac{1}{2}|y_i - h_a(x)|$ is called the *loss* and when $y_i = \pm$ it can take only the values 0 or 1.

If we now take $0 \leq e \leq 1$ then with probability $1 - e$ the following holds:

$$R(a) \leq R_{emp}(a) + \sqrt{\frac{h(\log \frac{2m}{h} + 1) - \log(\frac{e)}{4}}{m}} \tag{3}$$

where h is a non-negative integer called the Vapnik-Cervonenkis (VC) dimension which reflects the ability to learn any training set without error (this ability is called the *capacity* of the learning machine).

Remember that the idea in learning is to achieve a kind of balance between the capacity and the accuracy of a particular training set.

The right hand side of 3 is usually called *risk bound*, and in this the second term (the square root) is called *VC confidence*.

Remark 1 *We make the following remarks:*

1. *The bound in the above equation is* **independent** *of P.*
2. *The left hand side cannot usually be calculated.*
3. *To calculate the right hand side we need to know h.*

The above suggests then a way of selecting the function $h_a(x)$n namely, to select a value e and then select $h_a(x)$ which minimizes the risk bound. Such an approach is called *structural risk minimization.*

2 VC Dimension

Informally the VC dimension will convey the size of the largest training set that can be learned (separated) by a family of functions.

Shattering: Let us consider again the collection of points $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ with all possible labeling, that is, there are 2^m possible labeling.

We say that the family of functions $h_a(x)$ shatters S if for each possible labeling, there exists a member in this family, that is, an a_0, such that $h_{a_0}(x_i)$ outputs the correct label for x_i, $i = 1, ..., m$.

VC dimension: The VC dimension for a set of functions $\{h_a(x)\}$ is defined as the maximum number of training points that can be shattered by $\{h_a(x)\}$.

Remark 2 *If the VC dimension of* $\{h_a(x)\}$ *is h it means that* **there exists at least one set of** h **points** *that can be shattered by* $\{h_a(x)\}$ *but it* **will not be true that all such sets** *will be shattered.*
However, for the set which is shattered, **all** *possible labeling will be accounted for by* $\{h_a(x)\}$.

2.1 Shattering points in $\mathcal{R}^n$

It is clear that $\mathcal{R}$, that is, when $n = 1$ any set of two points can be shattered:
Let $\{x_i\}_{i=1,2},\ y_i = \mp$ the corresponding labels and let

$$h(x) = (-1)^i y_i \frac{2x_i - x_1 - x_2}{x_2 - x_1}$$

shatters any such set of two points.
Let us now consider $n = 2$, and the family of functions to be lines (oriented, in the sense that one side we assign the label 1 o the other we assign the label -1). Obviously any set of two points can be shattered. (Find the function).
If we consider a set of three points. It is easy to see that any labeling of this set can be shattered by an oriented line.

However, for four points this cannot be done. Therefore the VC dimension of $\mathcal{R}^2$ is 3. In fact, in general we have the following theorem:

Theorem 1 *Consider a set of m points in* $\mathcal{R}^n$. *Choose any of these points as origin. Then the m points can be shattered by oriented hyper planes (linear functions) if and only if the remaining* $m - 1$ *points are linearly independent.*

(We omit the proof)

Corollary 1 *The VC dimension of* $\mathcal{R}^n$ *is* $n + 1$.

Indeed, if we start with $n + 1$ points and select one of them as the origin, the remaining n can be linearly independent, and in fact for some such set the remaining will be linearly independent.

However, if we start with $n + 2$ points and select one of them as the origin, the remaining $n + 1$ will never be linearly independent (no $n + 1$ vectors can be independent in $\mathcal{R}^n$!)

Recall that we use interchangeably the terms points and vectors. You should recall that a point $P \in \mathcal{R}^n$ is the same as the vector with n components, originating in the origin of the system of coordinates and ending in P.

2.2 VC dimension and the number of parameters of a function

It may seem that if the family of function has many parameters then its VC dimension is very large. In general it is not true that the VC dimension depends on the number of parameters.

example: Let $s(t) = 1$ if $t > 0$ and -1 otherwise.

Let $h_a(x) = s(\sin(ax))$ for $x, a \in \mathcal{R}$.

Show that for a given m be a positive integer there exists a such that $h_a(x)$ shatters m points.

Proof: let $x_i = 10^{-i}$, with $i = 1, ..., m$.

Let $y_i = \mp 1$ an arbitrary labeling of the set $\{x_1, ..., x_m\}$.

Then if

$$a = \pi(1 + \sum_{i=1}^{m} \frac{(1 - y_i)10^i}{2})$$

$h_a(x)$ shatters the set $\{x_1, ..., x_m\}$.

Remark 3 *It should be noted that if the VC dimension for a family of functions, $h_a(x)$, is h it does not mean that $h_a(x)$ shatters* **all** *sets with less than h elements.*

To see this consider the above example, but now select the following points $\{x_i;\ i = 1, ..., 4\}$, such that for some a we have $ax_i = 2n\pi + i\delta$ for some δ.

Then $h_a(x_i) = \sin(i\delta)$. It is easy to see that there are labeling for these four pints that cannot be output by this function: for example, suppose that $y_1 = y_2 = y_4 = +1$ and $y_3 = -1$.

Then, from the conditions on the first three points we derive that

$$\frac{\pi}{3} < \delta < \frac{\pi}{2}$$

which then implies for $h_a(x_4) = -1$ which contradicts the desired labeling of this point.

2.3 Minimizing the risk bound by minimizing the VC dimension

Let us denote by $f_m(h)$ the quantity of the right hand side of equation (3) which depends on h and m, when $e = 0.05$, that is:

$$f_m(h) = \sqrt{\frac{h(\log \frac{2m}{h} + 1) - \log(\frac{e)}{4}}{m}} \tag{4}$$

As a function of the ratio $0 < \frac{h}{m} \leq 1$ this function is increasing (in a $\surd$ fashion) from 0 to approximately 1.3 (when $m = 10,000$). This is true for any value of the m.
The value 1 attained for $\frac{h}{m} = 0.37$. This means that for higher values the (of $\frac{h}{m}$) the bound is guaranteed no to be tight.

Thus, given several learning machines with $R_{emp} = 0$, we want to select from these that which has minimal VC dimension, as this will give a better upper bound on the actual error.
In general, if $R_{emp} \neq 0$ we want to minimize $R_{emp} + f_m$.

This strategy makes use only of equation (3) which gives the bound on the error with some probability. But even if this probability os very high it is still possible to have other set of functions, which have the same empirical risk and higher VC dimension to perform better.
At an extreme it can happen that the set of functions has good performance even though it has an infinite dimension ($k-$nearest neighbor classifier, with $k = 1$ has infinite VC dimension, 0 emprirical risk and still performs well). Thus **infinite VC-dimension DOES NOT MEAN poor performance!!!**.