# 1. Performance Bounds

Machine learning theorists have developed clever ways to set bounds on the performance capability of data mining procedures.

Suppose one has a training sample $\{(y_i, \boldsymbol{x}_i)\}$ and wants to predict a future $Y$ value from measured $\boldsymbol{X}$ values. To do this, one chooses a family of models $\mathcal{F} = \{f(\boldsymbol{x}, \boldsymbol{\theta})\}$ and uses the training sample to find a value of $\boldsymbol{\theta}$ that gives "good" performance.

This structure holds in both regression or classification. Here $f(\cdot, \boldsymbol{\theta}) : \mathbb{R}^p \to \mathbb{R}$, and the family of functions (the model $\mathcal{F}$) is indexed by $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

In multiple linear regression, $\mathcal{F}$ consists of all $p$-dimensional hyperflats and $\boldsymbol{\theta}$ are the regression coefficients. In two-class linear discriminant analysis, $\mathcal{F}$ consists of all $p$-dimensional hyperflats and the $\boldsymbol{\theta}$ are the values in the Mahalanobis-distance rule.

The performance of any model in $\mathcal{F}$ is assessed through a loss function $L[f(\boldsymbol{x}, \boldsymbol{\theta}), y]$. This measures the deviation between the true value $y$ and a predicted value $f(\boldsymbol{x}, \boldsymbol{\theta})$.

For regression, standard loss functions include:

- absolute loss: $L[f(\boldsymbol{x}, \boldsymbol{\theta}), y] = |f(\boldsymbol{x}, \boldsymbol{\theta}) - y|$
- squared error loss: $L[f(\boldsymbol{x}, \boldsymbol{\theta}), y] = [f(\boldsymbol{x}, \boldsymbol{\theta}) - y]^2$.

For binary classification, labeled so that $y \in \{-1, 1\}$, a standard loss function is:

- $L[f(\boldsymbol{x}, \boldsymbol{\theta}], y) = I[-yf(\boldsymbol{x}, \boldsymbol{\theta}) > 0]$.

This last is an indicator function that is 1 iff the sign of $y$ is different from the sign of $f(\boldsymbol{x}, \boldsymbol{\theta})$. Recall that the usual rule in classification is to predict 1 or -1 according to the sign of the classification rule $f(\boldsymbol{x}, \boldsymbol{\theta})$.

Assume that the joint distribution of future values of $(Y, \boldsymbol{X})$ is $P(y, \boldsymbol{x})$. Then the risk, or expected loss in a future prediction, is

$$R(\boldsymbol{\theta}) = \int_{\mathbb{R}^p \times \mathbb{R}} L[f(\boldsymbol{x}, \boldsymbol{\theta}), y] \, dP(y, \boldsymbol{x}).$$

But this cannot be calculated without knowing $P(y, \boldsymbol{x})$.

Consequently, people use the training sample to calculate the empirical risk:

$$R_e(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L[f(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i].$$

We want to find a bound on the risk such that with high probability,

$$R(\boldsymbol{\theta}) \leq R_e(\boldsymbol{\theta}) + B.$$

That will tell us how bad our empirical risk is as an estimate of the true risk.

In the context of two-class classification, Vapnik (*Statistical Learning Theory*, 1995) showed that with probability $q$,

$$R(\boldsymbol{\theta}) \leq R_e(\boldsymbol{\theta}) + \sqrt{\frac{1}{n}[v + \ln(v/h) - \ln(q/4)]}$$

where $v$ is a non-negative integer called the Vapnik-Červonenkis (VC) dimension.

Note that the bound:

- does not depend on the $P(y, \boldsymbol{x})$;
- assumes that the training sample is a random sample from $P(y, \boldsymbol{x})$;
- is simple to compute, if $v$ is known;
- can exceed 1, in which case it is useless.

The VC dimension depends upon the class $\mathcal{F}$. To start, we consider only the two-class discrimination problem, so $f(\boldsymbol{x}, \boldsymbol{\theta}) \in \{-1, 1\}$.

A given set of $n$ points can be labeled in $2^n$ possible ways. If for any such labeling, there is a member of $\mathcal{F}$ that can correctly assign those labels, then we say that the set of points is shattered by $\mathcal{F}$.

The VC dimension for $\mathcal{F}$ is defined as the maximum number of points (i.e., training data) that can be shattered by the elements of $\mathcal{F}$.

Note: If the VC dimension is $v$, then there exists at least one set of $v$ points that can be shattered. In general, it is not true that *every* set of $v$ points can be shattered.

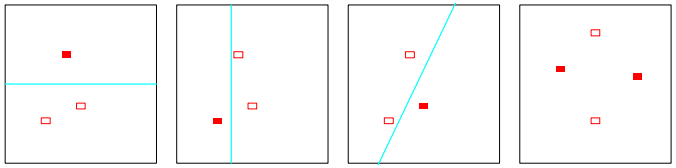

Figure 7.6: *The first three panels show that the class of lines in the plane can shatter three points. The last panel shows that this class cannot shatter four points, as no line will put the hollow points on one side and the solid points on the other. Hence the VC dimension of the class of straight lines in the plane is three. Note that a class of nonlinear curves could shatter four points, and hence has VC dimension greater than three.*

For the two-class linear discrimination problem in $\mathbb{R}^p$, one can prove that the VC dimension is $v = p + 1$. Thus the bound on the risk gets large as $p$ increases.

One expects that as the elements of $\mathcal{F}$ get more flexible, then the VC dimension should increase. But the situation is complex. Consider $\mathcal{F} = \{f(x, \theta)\}$ where

$$f(x, \theta) = \begin{cases} 1 & \textbf{if } \sin(\theta x) > 0 \\ -1 & \textbf{if } \sin(\theta x) \leq 0. \end{cases}$$

Select the points $\{x_i = 10^{-i}\}$ for $i = 1, \ldots, n$. Let $y_i$ be the label of $x_i$. The one can show that for any choice of labels,

$$\theta = \pi \left[ 1 + \sum_{i=1}^{n} \frac{1}{2}(1 - y_i)10^i \right]$$

gives the correct classification (example due to Levin and Denker).

Thus a one-parameter family can have infinite VC dimension.

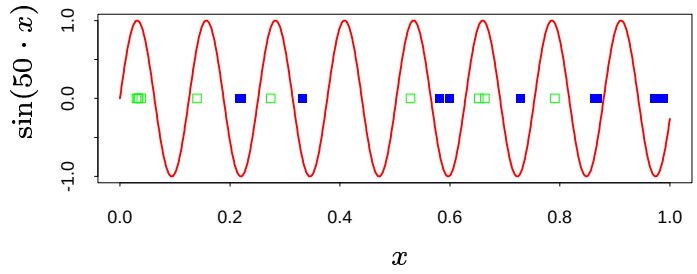


Figure 7.5: *The solid curve is the function* $\sin(50x)$ *for* $x \in [0,1]$. *The blue (solid) and green (hollow) points illustrate how the associated indicator function* $I(\sin(\alpha x) > 0)$ *can shatter (separate) an arbitrarily large number of points by choosing an appropriately high frequency* $\alpha$.

The motivation behind this approach is the old bias-variance tradeoff (or overfitting, or capacity control). One gets the best predictive performance if the family (or machine) $\mathcal{F}$ strikes the right balance between the capacity of the family and the accuracy of the performance on the training set.

Capacity refers to the ability of the machine to perfectly fit the training sample. If the family $\mathcal{F}$ is so flexible that it can perform without error on the training sample, then it is fitting the noise as well as the signal (i.e., overfitting).

A family with too much capacity is like a biologist with perfect memory, who when shown a dog, cannot identify its species because it has a different number of hairs than any dog he previously studied. A family with too little capacity is like a lazy biologist who classifies everything with four legs as a dog.

Consider the 1-nearest neighbor classifier. This $\mathcal{F}$ has infinite $v$, since any number of arbitrarily labeled points can be shattered. And the empirical risk is zero (unless two observations with opposite labels coincide). So in this case the VC bound gives no useful information.

Burges (1998; *Data Mining and Knowledge Discovery*, **2**, 121-167) considers a "notebook" classifier for a 50/50 mix of population types A and B. The notebook has $m$ pages; one writes down the labels of the first $m$ training observations, for $m < n$. With all subsequent data, predict type A.

The empirical risk (under standard classification loss) for the first $m$ values is 0; the empirical risk for the next $n - m$ is .5. The true risk on future samples is .5. And the VC dimension is $v = m$.

Similarly to classification, one can find a VC-bound on the risk in regression:

$$R(\boldsymbol{\theta}) \leq \frac{R_e(\boldsymbol{\theta})}{(1 - c\sqrt{\delta})_+}$$

where

$$\delta = \frac{a}{n}\left[v + v \ln \frac{bn}{v} - \ln \frac{q}{4}\right].$$

As before, the probability that this bound holds is $1 - q$.

The bound was developed by Cherkassky and Mulier (1998; *Learning From Data*, 108-11). One can tune the constants $a, b, c$ to the application. Cherkassky and Mulier recommend taking $a = b = c = 1$ for regression applications.

The bound tends to be loose.

To use the VC dimension $v$ for regression problems, one needs to extend its definition from classes of dyadic (i.e., +1/-1) functions $\mathcal{F}$ to classes of real-valued functions.

The strategy is to take any class of real-valued functions $\{f(\boldsymbol{x}, \boldsymbol{\theta})\}$ and create a set of dyadic functions

$$f_y(\boldsymbol{x}, \boldsymbol{\theta}) \begin{cases} 1 & \textbf{if } f(\boldsymbol{x}, \boldsymbol{\theta}) - y > 0 \\ -1 & \textbf{if } f(\boldsymbol{x}, \boldsymbol{\theta}) - y \leq 0 \end{cases}$$

where $y$ is in the range $R$ of $f$. The VC-dimension $v_y$ for $\{f_y(\boldsymbol{x}, \boldsymbol{\theta})\}$ is now defined in the same way as before.

The VC dimension for the regression class $\mathcal{F}$ is $v = \sup_R\{v_y\}$.

Vapnik advocates Structural Risk Minimization to exploit the VC bound.

Consider a nested sequence of function classes:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots \subset \mathcal{F}_k = \mathcal{F}$$

where $k$ may go to infinity. Let class $\mathcal{F}_i$ have VC dimension $v_i$, and this sequence is strictly increasing: $v_1 < v_2 < \ldots < v_k$.

For example, in classification $\mathcal{F}_1$ might be linear discrimainators, $\mathcal{F}_2$ might be quadratic discriminant functions, and so forth. It is known that the VC dimension for a polynomial discriminator of degree $d$ in $\mathbb{R}^2$ is $(d^2 + 3d + 2)/2$ (Vapnik, 1995).

The SRM strategy is to start at $\mathcal{F}_1$ and increase. The empirical risk for the best function in $\mathcal{F}_i$ decreases monotonically in $i$, and the second term in the risk bound increases monotonically in $i$. One stops increasing the class at the $i^*$ for which the risk bound is minimized.

People have worked hard to find the VC-dimension for:

- support vector machines
- neural networks
- tree-based classifiers

The situation for ensemble methods is still largely unresolved.

Alternatives to the VC-dimension approach include

- cross validation
- the AIC
- other kinds of PAC bounds

PAC stands for **Probably Approximately Correct**. In the context of classification of two groups, it defines a **concept class** as:

A family $\mathcal{H}$ of Boolean functions on a domain $\boldsymbol{X}$, with each function $h :\rightarrow \{0, 1\}$.

It also defines a **target concept** as a function $c(\boldsymbol{x})$ such that:

The probability of a +1 label for a case with given value $\boldsymbol{x} \in \boldsymbol{X}$, or $c(\boldsymbol{x}) = \mathbb{P}[y = 1|\boldsymbol{x}]$, where $(\boldsymbol{x}, y)$ has joint distribution $f(\boldsymbol{x}, y)$.

An **oracle** can draw $\boldsymbol{x} \in \boldsymbol{X}$ according to the marginal distribution $f(\boldsymbol{x})$ and then assigns label $y = 1$ with probability $c(\boldsymbol{x})$ and $y = 0$ with probability $1 - c(\boldsymbol{x})$. Having an oracle is like having an infinite training sample that draws randomly from the joint distribution. (Note: we have abandoned our previous convention of labeling the classes as 1 or -1, just to make the following definition easier to write.)

Definition: A concept $h \in \mathcal{H}$ (i.e., a classification rule) is $(\epsilon, \gamma)$-good if:

$$\mathbb{E}_{\boldsymbol{x}}[I(|h(\boldsymbol{x}) - c(\boldsymbol{x})| > \gamma)] \leq \epsilon.$$

A target concept $c(\boldsymbol{x})$ is **universally-learnable** if there exists an algorithm $\mathcal{A}$ such that for all target concepts $c(\boldsymbol{x})$ (i.e., for all distributions $f(\boldsymbol{x}, y)$) and for all positive $\gamma$ and $\epsilon$, the algorithm can look at a sequence of oracle outputs and eventually terminate, returning a concept (classification rule) $h$ that is $(\epsilon, \gamma)$-good.

VC-theory, based on empirical risk minimization, leads to algorithms that produce universally-learnable classification rules for a broad class of problems.

Often one can find algorithms that learn in polynomial time.