On the Effective VC Dimension

Léon Bottou,
Neuristique, 28 rue des Petites Ecuries, 75010 Paris

Corinna Cortes & Vladimir Vapnik,
AT&T Bell Laboratories, Holmdel NJ 07733, USA

February 4, 1994

Abstract

The very idea of an "Effective Vapnik Chervonenkis (VC) dimension" (Vapnik, Levin and Le Cun, 1993) relies on the hypothesis that the relation between the generalization error and the number of training examples can be expressed by a formula algebraically similar to the VC bound. This hypothesis calls for a serious discussion since the traditional VC bound widely overestimates the generalization error.

In this paper we describe an algorithm and data dependent measure of capacity. We derive a confidence interval on the difference between the training error and the generalization error. This confidence interval is much tighter than the traditional VC bound.

A simple change of the formulation of the problem yields this extra accuracy: our confidence interval bounds the error difference between a training set and a test set, rather than the error difference between a training set and some hypothetical grand truth. This "transductive" approach allows for deriving a data and algorithm dependent confidence interval.

1 Introduction.

1.1 Context.

The notion of an "Effective Vapnik Chervonenkis (VC) dimension" dates back to attempts to measure experimentally the VC dimension of a neural network (Levin, Le Cun and Vapnik, 1992), (Vapnik, Levin and Le Cun, 1993). These experiments exhibited puzzling results.

According to the definition of the VC dimension, the most obvious approach consists in searching the largest subset of examples that successfully can be split in all possible ways by the network.

This approach must be discarded for practical reasons: It is neither computationally feasible to test all possible subsets, nor reliable to test a few random subsets only, for fear to miss the largest subset. Finally, the network training algorithm sometimes misses the existing network solutions.

Another idea consists in evaluating the VC dimension from cross-validation experiments. The procedure described below relies on an average (instead of a supremum) over all subsets of the set of examples. We can thus obtain stable estimates with a few random subsets.

This procedure (Vapnik, Levin and Le Cun, 1993) operates on a set of random examples belonging to two classes:

The class labels of a random half of the set of examples are switched and the neural network is trained on both the correct examples and the mislabeled examples. The error difference between both halves is then averaged on several runs. This measure is then compared with a theoretical expression derived from the Vapnik Chervonenkis bounds (Vapnik, 1982).

This procedure is certainly not rigorous:

- There is no simple evidence that switching random labels ensures that the learning algorithm finds the set of weights which maximize the error difference.
- This procedure assumes that the Vapnik Chervonenkis bounds are tight enough to allow a successful interpretation.

On the other hand, this approach allows for studying data dependent and algorithm dependent effects by tampering with the learning procedure or by using specific examples (e.g. using character images instead of random patterns). This procedure was in fact designed as an exploratory method rather than an exact measure.

Experiments were first carried out on a linear network with the following results:

- By adding a scaling factor to the theoretical expressions, there is a good fit between the experimental points and the curve.
- When the random examples are replaced by real examples, we must *reduce the value of the VC dimension parameter* below its theoretical value. We obtain then a perfect fit with *the same scaling factor*.
- When a regularization term (e.g. a weight decay) is added to the learning procedure, there is a fit with again *the same scaling factor* and different *decreasing values for the VC dimension parameter* with increasing values of the regularization parameter.

Similar results have been obtained on multilayer networks (Cortes, personal communication) although the lack of stability of the scaling parameter and the training procedure strongly reduces their practical significance.

These results led us to conjecture that the relation between the generalization error and the number of examples can be expressed by a formula algebraically similar to the VC bound. The observed value of the VC dimension parameter has then been named "Effective VC dimension".

This empirically defined "Effective VC dimension" depends on both the problem and the algorithm. It effectively displays the effects of pruning a neural network, preprocessing the data or regularizing the training algorithm (Guyon et al., 1992).

This conjecture calls for a serious discussion since the traditional VC bound widely overestimates the generalization error.

1.2 Summary.

In this paper we describe an algorithm and data dependent measure of capacity. We derive a confidence interval on the difference between the training error and the generalization error. This confidence interval is much tighter than the traditional VC bound.

A simple change of the formulation of the problem yields this extra accuracy: our confidence interval bounds the error difference between a training set and a test set, rather than the error deviation between a training set and some hypothetical grand truth. This approach allows for deriving a data and algorithm dependent confidence interval.

Learning from examples refers to the setting where all the available information about a specific mapping is a pool of labeled patterns. Common learning theories assume that the labels are the result of applying some noisy unknown function, the *grand truth*, to the patterns.

Most learning procedures built on this assumption perform two inference steps: *Induction* consists in identifying the grand truth. *Deduction* consists in applying this result to new data. Learning theories have concentrated on the induction step. They attempt to predict how well the result of a learning algorithm estimates the grand truth.

Unlike these theories, we use an approach related to the bootstrap methods (Efron, 1979). Instead of assuming a grand truth distribution we consider all possible partitions of a finite pool of example into a training set and a test set, and study how well the performance on the training set estimates the performance on the test set.

This theoretical framework is close to actual experimental settings. We call it *transduction* because it bypasses the grand truth. An advantage of this approach is that it drastically reduces the technical problems involved in the derivation of a bound.

We first present the basic hypothesis of the transductive approach. We then derive a data dependent and learning algorithm dependent notion of capacity based on the VC dimension framework. We finally discuss how this theoretical result makes a stronger basis for the empirical notion of an "effective VC dimension".

2 Settings.

From the discrete, transductive point of view, the basic elements of a learning problem are:

- An arbitrary set S of $m = l + n$ labeled examples $z_1, \cdots, z_{l+n}$. There are C_m^l ways to split the data set S into a training set S_1 of size l and a test set S_2 of size n.
- A deterministic learning algorithm $\mathcal{A}$ which produces a working device w^* using the split set of examples. Given a set S and a training set size l, there is a finite set $\Omega_l(S)$ of possible devices returned by algorithm $\mathcal{A}$.
- A loss function $Q(z_i, w)$ which measures the performance of device w on example z_i. In this paper, we only consider the case of binary loss functions. Such a loss function returns 1 in the case of a misclassification and returns 0 otherwise:

$$Q(z_i, w) = \begin{cases} 0 & \text{if device } w \text{ classifies example } z_i \text{ correctly,} \\ 1 & \text{otherwise.} \end{cases}$$

These definitions can be illustrated from a multilayer network. Each example z_i is a pair (x_i, y_i) composed of an input vector x_i and an output vector y_i. The loss function $Q(z_i, w)$ indicates the performance of the network on pattern (x_i, y_i).

For each choice of a training set S_1 and a test set S_2, and for each device w in set $\Omega_l(S)$, we can define the training error ν_1, the test error ν_2 and the total error ν as:

$$\begin{aligned} \nu_1(w) &= \frac{1}{l} \sum_{z_i \in S_1} Q(z_i, w) \\ \nu_2(w) &= \frac{1}{n} \sum_{z_i \in S_2} Q(z_i, w) \\ \nu(w) &= \frac{1}{m} \sum_{z_i \in S} Q(z_i, w) \end{aligned}$$

This paper presents a result involving the maximal deviation between the quantities $\nu_1(w)$ and $\nu_2(w)$ simultaneously valid for all devices $w \in \Omega_l(S)$ reachable by the training algorithm:

$$\underset{w \in \Omega_l(S)}{\mathrm{Max}} \mid \nu_2(w) - \nu_1(w) \mid \tag{1}$$

In the rest of this paper, we refer to this deviation as the *uniform error deviation.*

The uniform error deviation depends on the training set S_1 and the test set S_2, on which we measure respectively the training error $\nu_1(w)$ and the test error $\nu_2(w)$. We study the distribution of this deviation for all possible splits of the total set S into a training set of size l and a test set of size n:

$$Pr\left\{ \underset{w \in \Omega_l(S)}{\mathrm{Max}} \mid \nu_2(w) - \nu_1(w) \mid > \epsilon \right\}$$

or equivalently:

$$Pr\left\{ \exists w \in \Omega_l(S), \quad \mid \nu_2(w) - \nu_1(w) \mid > \epsilon \right\} \tag{2}$$

A few facts are worth noticing about this distribution:

- The uniform error deviation distribution is both *data dependent* and *algorithm dependent*: set $\Omega_l(S)$ is defined as the set of the possible outcomes of applying algorithm $\mathcal{A}$ to all training sets of size l extracted from the data set S.

- The probability distribution is discrete: among the C_m^l possible choices of a training set, $Pr(\mathcal{H}) \times C_m^l$ choices only fulfill the condition $\mathcal{H}$.

  We use the notation $Pr(\cdots)$ instead of $P(\cdots)$ to recall that we are only using a discrete probability defined by all the possible splits of an example set into a training set and a test set.

- The deviation $\nu(w) - \nu_1(w)$ takes on discrete values because both the total error and the training error take on discrete values. Formula (2) thus describes the discrete *cumulative histogram* of the uniform error deviation.

3 Uniform error deviation.

We now derive a bound for the distribution of uniform error deviation (1). We then argue that this bound models closely of actually measured distributions, and thus leads to accurate confidence intervals in this regime.

3.1 Preliminary result.

First we recall the definition of the hypergeometrical distribution and derive a preliminary result.

A bag contains m balls, p of which are red. We extract l balls simultaneously from this bag. The probability that k of these l balls are red follows the hypergeometrical law:

$$h(m,l,p,k) \;=\; \frac{C_p^k \; C_{m-p}^{l-k}}{C_m^l} \qquad 0 \le k \le \mathrm{Max}\{l,p\}$$

Consider an arbitrary but fixed corner q of the unit hypercube with m dimensions. Let $\mu(q)$ be the average of its m coordinates $(q_1, \cdots, q_m) \in \{0,1\}^m$.

$$\mu(q) = \frac{1}{m}\sum_{i=1}^{m} q_i$$

There are C_m^l possible splits of this set of coordinates into a set S_1 of size l and a set S_2 of size n. Let us denote as $\mu_1(q)$ (respectively $\mu_2(q)$) the average of the coordinates of the first set (respectively the second set):

$$\begin{aligned} \mu_1(q) &= \frac{1}{l}\sum_{i \in S_1} q_i \\ \mu_2(q) &= \frac{1}{n}\sum_{i \in S_2} q_i \end{aligned}$$

We obtain the cumulative distribution of the deviation between these two averages by counting the proportion of possible choices of l coordinates which fulfill the condition $\mid \mu_2(q) - \mu_1(q) \mid > \epsilon$.

$$Pr\,\{\mid \mu_2(q) - \mu_1(q) \mid > \epsilon\} \;=\; \sum_{\mid (m\mu(q)-k)/n \;-\; k/l \mid > \epsilon} h(m,l,m\mu(q),k)$$

where the running variable k denotes the number of non zero coordinates in set S_1. It is then practical to rewrite this inequality as:

$$Pr\,\{\mid \mu_2(q) - \mu_1(q) \mid > \epsilon(\mu(q),l,m,\eta)\} \;=\; \eta \qquad (3)$$

Alas, there are no simple analytical expression of $\epsilon(\mu,l,m,\eta)$. There are however numerous bounds or approximations. In particular, the following results hold when both sets have the same size (i.e. $m = 2l = 2n$):

- A first result is derived from equation (A15) page 176 of (Vapnik, 1982). This result gives an *absolute* upper bound. This bound is rather tight when $\mu \approx 0.5$ and l is large enough:

$$\epsilon(\mu, l, 2l, \eta) \;\leq\; \sqrt{\frac{\log(2/\eta)}{l-1}} \tag{4}$$

- A second result is derived from equation (A22) page 180 of (Vapnik, 1982). This second result gives a *relative* upper bound. This bound is tighter when μ is small:

$$\epsilon(\mu, l, 2l, \eta) \;\leq\; 2\sqrt{\mu\frac{\log(2/\eta)}{l}} \tag{5}$$

There is much literature about such bounds. There are also numerical methods for computing this quantity (Press et al., tion) with a good accuracy. Therefore, we consider in this paper that $\epsilon(\mu, l, m, \eta)$ is known and tabulated.

3.2 Uniform bound.

Given a device w^*, the loss function $Q(z, w^*)$ maps the total set S on a corner of the unit hypercube in m dimensions:

$$\forall z_i \in S, \quad q_i = Q(z_i, w^*)$$

Two different choices of a training set can of course produce two different devices which eventually map the examples on the same corner of the hypercube. We can thus define equivalence classes on the choices of the training sets. Each equivalence class gathers the choices that drive the total set of examples on the same corner of the hypercube.

We define the quotient set $\Delta_l(S)$ as the subset of the corners of the hypercube reached by applying all the devices $w^* \in \Omega_l(S)$, obtained by running the training algorithm on all possible training sets S_1 extracted from set S.

$$\Delta_l(S) \;=\; \{q = (Q(z_1, w), \cdots, Q(z_m, w))\,, \;\; \forall w \in \Omega_l(S)\}$$

Applying result (3) within each equivalence class produces then a bound for the uniform error deviation (2):

$$
\begin{aligned}
&Pr\{\ \exists w \in \Omega_l(S),\ \mid \nu_2(w) - \nu_1(w) \mid > \epsilon(\nu(w), l, m, \eta)\ \} \\
&\quad = Pr\{\ \exists q \in \Delta_l(S),\ \mid \mu_2(q) - \mu_1(q) \mid > \epsilon(\mu(q), l, m, \eta)\ \} \\
&\quad = Pr\left(\bigcup_{q \in \Delta_l(S)} \{\ \mid \mu_2(q) - \mu_1(q) \mid > \epsilon(\mu(q), l, m, \eta)\ \} \right) \\
&\quad \leq \sum_{q \in \Delta_l(S)} Pr\{\ \mid \mu_2(q) - \mu_1(q) \mid > \epsilon(\mu(q), l, m, \eta)\ \} \\
&\quad = \eta\ \mathrm{Card}(\Delta_l(S))
\end{aligned}
\tag{6}
$$

As usual, the notation $Pr(\cdots)$ denotes a discrete probability. This bound addresses the proportion of choices of a training set (within the total set of examples S) that lead to an error deviation larger than $\epsilon(\mu(q), l, m, \eta)$.

3.3 The uniform bound and the VC dimension.

Following (Vapnik, 1982), we next seek a bound on $\mathrm{Card}(\Delta_l(S))$. Each corner of the hypercube embodies a dichotomy of the total set of examples S. By definition, the set $\Delta_l(S)$ gathers the dichotomies implemented by one of the devices produced by algorithm $\mathcal{A}$. The cardinal of $\Delta_l(S)$ is thus equal to the number of dichotomies implemented by the family of functions *reachable by the training algorithm*:

$$\{z \rightarrow Q(z, w),\quad \forall w \in \Omega_l(S)\}$$

According to (Vapnik, 1982), this number is either 2^m or bounded by a polynomial quantity of degree h, where h is a positive integer named the *VC dimension* of the family of functions.

$$\mathrm{Card}(\Delta_l(S)) \ \leq\ 1.5\frac{m^h}{h!} \ \leq\ \left(\frac{me}{h}\right)^h \tag{7}$$

We can thus write the following uniform bound, which is obviously related to the usual uniform convergence results (Vapnik, 1982):

$$
\begin{aligned}
&Pr\{\ \exists w \in W,\ \nu_2(w) - \nu_1(w) > \epsilon(\nu(w), l, m, \eta)\ \} \\
&\qquad \leq\ \eta\ \mathrm{Card}(\Delta_l(S))\ \leq\ \eta\left(\frac{me}{h}\right)^h
\end{aligned}
\tag{8}
$$

There are however several important differences between the usual uniform convergence results and bound (8):

- The set of devices does not define a fixed family of functions with a fixed VC dimension. The effective VC dimension depends on both the data set S and the algorithm $\mathcal{A}$. Our result is therefore both *data dependent* and *algorithm dependent.*
- This result bounds the error deviation between two sets extracted from a finite example set of size m. The standard Vapnik Chervonenkis results bound the difference between the training error and the asymptotical generalization error.

The importance of these differences is easily displayed if we attempt to derive a bound on the uniform error deviation when we randomly draw the training set and the test set from a grand truth distribution P.

We consider the following equivalent procedure:

i) We first draw a random set S of m independent examples from the grand truth distribution.

ii) We randomly select a training subset of size l. The remaining examples are then the testing set.

A confidence interval is obtained by taking the expectation of result (8) with respect to the selection of the total set S of random examples.

$$P\{\ \exists w \in \Delta_l(S),\ \mid \nu_2(w) - \nu_1(w) \mid > \epsilon(\nu(w), l, m, \eta)\ \} \\ \leq \eta\ E(\mathrm{Card}(\Delta_l(S))) \tag{9}$$

where the notations $P(\cdots)$ and $E(\cdots)$ denote the probability and the expectation with respect to the selection of the training set and the test set from the grand truth distribution.

In the case $m = 2l = 2n$, we can eliminate the quantity $\nu(w)$ in this equation by replacing the quantity $\epsilon(\nu(w), l, m, \eta)$ by the approximations (4) or (5). We obtain then an absolute bound and a relative bound, which are algebraically similar to the usual bounds (Vapnik, 1982):

$$P\{\ \exists w \in \Delta_l(S),\ \mid \nu_2(w) - \nu_1(w) \mid > \sqrt{\frac{\log(2/\eta)}{l-1}}\ \} \leq \eta\ E(\mathrm{Card}(\Delta_l(S)))$$

$$P\{\ \exists w \in \Delta_l(S),\ \frac{\mid \nu_2(w) - \nu_1(w) \mid}{\sqrt{\nu(w)}} > 2\sqrt{\frac{\log(2/\eta)}{l}}\ \} \leq \eta\ E(\mathrm{Card}(\Delta_l(S)))$$

Again, there are two essential differences between these bounds and bounds derived from the traditional Vapnik Chervonenkis results:

- The right hand side of the bounds derived from the usual results is typically proportional to the largest number of dichotomies that the family of functions can achieve in a set of examples of size $m = l + n$.
- We have here the average number of dichotomies reachable by our algorithm on each data set of size $m = l + n$.

3.4 Quality of the uniform bound.

The derivation of bound (6) contains only one bounding operation. Precisely, the probability of a union of event is bounded by the sum of the probabilities of the events. This error is smaller than the probability of overlaps between these events.

This probability of overlap is related to the ultrametric properties of the set of corners $\Delta_l(S)$ in a rather complex way. The more identical the corners q in set $\Delta_l(S)$, the larger the overlap.

When m increases, we consider in fact the overlaps of a polynomial number of exponentially improbable events:

- Both results (4) and (5) indeed show that the probability of these events decreases exponentially when m increases.
- Result (7) shows that the number of events grows only polynomially with the number of examples m.

Unless the corners q in set $\Delta_l(S)$ have a very odd repartition, the probability of the intersection of two events of probability η will be $\mathcal{O}(\eta^2)$. Since there are about $\mathrm{Card}(\Delta_l(S))^2/2$ possible overlaps, the total probability of the overlaps is $\mathcal{O}(\mathrm{Card}(\Delta_l(S))^2\eta^2)$.

Equation (6) can then be rewritten as

$$Pr\{\ \exists w \in \Omega_l(S),\ \mid \nu_2(w) - \nu_1(w) \mid > \epsilon(\nu(w), l, m, \frac{\eta'}{\mathrm{Card}(\Delta_l(S))})\ \} \ \approx\ \eta' + \mathcal{O}(\eta'^2)$$

This *intuitive argument* explain why bound (6) is likely to closely model the tail of the distribution of the uniform error deviation. A rigorous proof of this fact should however formalize the fact that the corners do not have a very odd repartition. We have no such proof so far ...

4 Discussion.

Let us recall and comment the VC dimension measurement procedure briefly described above.

This procedures consists in dividing the set of examples in two halves. The labels of the first set are switched. When a network is trained on both the correct and the mislabeled examples, the network learns:

- to achieve a low error rate on the correct examples,
- to achieve a high error rate on the mislabeled examples, since it actually learns to associate the example with the wrong class.

We average then this largest error deviation found by the training algorithm between randomly selected halves of the total set of examples.

The procedure measure the expectation of the uniform error deviation (1) for a given data set and a given[1] training procedure.

The Effective VC Dimension experiments become understandable if we believe, as argued before, that result (9) models the tail of the uniform error deviation distribution with a sufficient accuracy.

When the VC dimension is finite indeed, the quantity $E(\mathrm{Card}(\Delta_l(S)))$, bounded by a polynomial expression, grows at most polynomially with the number of examples m. Since the distribution of the uniform error deviation only depends on the logarithm of this quantity (see formulas 4 and 5), we mostly care about the polynomial degree of this growth. This degree embodies most of the dependence of the confidence intervals with the number of examples.

This degree is both data and algorithm dependent. It has the algebraic properties of a VC dimension. It is thus a good candidate for being the *Effective VC dimension.*

5 Conclusion.

This work on the measure of the VC dimension has uncovered certain aspects of the generalization problem:

- We have derived a tight bound for the uniform error deviation. This bound unifies the various absolute and relative Vapnik Chervonenkis bounds with a single call to the hypergeometrical distribution.

[1] In our framework, a training procedure computes a device using the split set of examples. Regular training procedures attempt to learn the training set. Our framework however can handle a paradoxical procedure which learns the first set and "mislearn" the second set.

- We have set theoretical grounds for the notion of an *Effective VC Dimension.* The algebraical expression of the Vapnik Chervonenkis bounds has indeed a much larger validity than initially expected. These formula can be used for performance improvement as explained in (Guyon et al., 1992).

Acknowledgments.

We would like to acknowledge discussions with Isabelle Guyon and Sara A. Solla from the AT&T Bell Laboratories group, Holmdel, NJ.

References

Efron, B. (1979). Bootstrap methords: Another look at the jackknife. *Annals of Statistics*, 7:1–26.

Guyon, I., Vapnik, V. N., Boser, B. E., Bottou, L., and Solla, S. A. (1992). Structural risk minimization for character recognition. In *Advances in Neural Information Processing Systems*, volume 4, Denver. Morgan Kaufman.

Levin, E., Le Cun, Y., and Vapnik, V. N. (1992). Measuring the capacity of a learning machine (ii). Technical Report TM 11359-920728-20TM, AT&T Bell Laboratories.

Press, W. H., Flannery, B. P., A., T. S., and T., V. W. (1992 (2nd edition)). *Numerical Recipes.* Cambridge University Press, Cambridge.

Vapnik, V. N. (1982). *Estimation of dependences based on empirical data.* Springer Verlag.

Vapnik, V. N., Levin, E., and Le Cun, Y. (1993). Measuring the VC-Dimension of a learning machine. *Neural Computation.* to appear.