Predicting Learning Curves without the Ground Truth Hypothesis

Léon Bottou
AT&T Labs - Research
Red Bank NJ07701
leonb@research.att.com

Yann LeCun
AT&T Labs - Research
Red Bank NJ07701
yann@research.att.com

Vladimir Vapnik
AT&T Labs - Research
Red Bank NJ07701
vlad@research.att.com

Abstract

Upper bounds for the deviation between test error and training error of a learning machine are derived in the case where no probability distribution that generates the examples is assumed to exist. The bounds are data-dependent and algorithm dependent. The result justifies the concept of data-dependent and algorithm dependent VC-dimension.

1 Introduction

The main purpose of learning theory is to study how a learning machine trained on a finite number of samples will perform on new, unseen samples. More specifically, we are interested in predicting the difference between the error rate measured on the new samples, and the error rate obtained on the training samples. In most theories a link is established between the training samples and the test samples by assuming that they are all drawn independently from an unknown probability distribution. We call this distribution the "ground truth", because to know it is to solve the learning problem (though solving the learning problem does not require to know it perfectly).

Although epistemology theories have long played with the idea that the world is ruled by simple universal truths waiting to be uncovered, it can be argued that in many real learning situations, there may be no such thing as an unattainable underlying process that generates the data. The only thing that is available to us for sure is the finite set of samples we have been given. In many cases, we cannot hope to ever obtain more data than what we have been given. the concept of underlying distribution has little relevance in that context.

Establishing a relationship between the training samples and the test samples can be done by posing the so-called *exchangeability hypothesis*: the dataset is given to us once and for all, but any partition of this dataset into a training set and a test set is seen as fortuitous. Therefore, we will seek results that take into account all the possible partitions of the data set into training set and test set.

We will derive bounds on the distribution of the deviation between the test error and the training error with no additional assumption but the exchangeability of the samples. The bounds will be not only data dependent, but also algorithm dependent.

There are several conceptual, technical, and practical reasons for replacing the "ground truth hypothesis" by the "exchangeability hypothesis". First of all, the ground truth distribution cannot conceivably be determined empirically from a finite dataset, unless strong assumptions are made about its nature. Second, there is no statistical test to determine whether a dataset has indeed been drawn independently from a distribution, although it is possible to devise tests to determine that they have not. Third, many empirical studies of learning do rely on splitting a pre-existing dataset into training sets and test sets. A consistent performance over a large number of arbitrary splits is considered a convincing estimate of the generalization performance of the proposed algorithm.

In the following, we apply the standard mathematical techniques developed by Vapnik and Chervonenkis to derive bounds on the distribution of the deviation between the test error and training error over all possible splits of a given dataset. Surprisingly the elimination of the ground truth hypothesis makes these bounds more accurate than the comparable Vapnik Chervonenkis bounds. It also simplifies the mathematics, and provides simple ways to produce data dependent and algorithm dependent predictions of the learning curves.

2 Definitions and Notations

We are given a finite set S of $n = n_1 + n_2$ labeled samples $z_1, \cdots, z_n$. The dataset is split into a training set S_1 with n_1 samples, and a test set S_2 with n_2 samples. There are $C_n^{n_1} = n!/(n_1!\,n_2!)$ different ways to chose that split. A loss function $Q(z, w)$ measures the correctness on sample z of the answer produced by a learning machine parameterized by w. In this paper we only consider the case of binary loss functions that take the value 1 if the answer is wrong and 0 if it is correct. Finally, a deterministic learning algorithm $\mathcal{A}$ produces the parameter w^{S_1} when given the training set S_1.

In the case of a multi-layer perceptron for instance, the parameter w is the weight vector, each example z_i is a pair (x_i, y_i) composed of an input vector x_i and an output class y_i. The loss function $Q(z_i, w)$ indicates the performance of the network w on each example (x_i, y_i).

For each choice of a training set S_1 and a test set S_2, and for each system w, we can define the training error ν_1, the test error ν_2 and the total error ν as:

$$\nu_1(w) = \frac{1}{n_1} \sum_{z_i \in S_1} Q(z_i, w) \ , \qquad \nu_2(w) = \frac{1}{n_2} \sum_{z_i \in S_2} Q(z_i, w)$$

$$\nu(w) = \frac{1}{n} \sum_{z_i \in S} Q(z_i, w)$$

The results presented in this paper concern the distribution over all possible splits of the dataset of the deviation between the training error $\nu_1(w^{S_1})$ and the testing error $\nu_2(w^{S_1})$, where w^{S_1} represents the system produced by running the learning algorithm $\mathcal{A}$ on the training set S_1 :

$$Pr\left\{ \ \left| \ \nu_2(w^{S_1}) - \nu_1(w^{S_1}) \ \right| > \epsilon \ \right\} \tag{1}$$

The notation $Pr(\mathcal{H})$ denotes the ratio of the number of training set/test set splits for which condition $\mathcal{H}$ is true, over the total number $C_n^{n_1}$ of possible splits.

3 Misclassification Vectors

For each system w, the loss function $Q(z,w)$ maps the full set of examples S onto a binary vector $q(w) = (Q(z_1,w),\ldots,Q(z_n,w))$ of length n called the *misclassification vector*. Each component is 0 if the system's answer for the corresponding sample is correct, and 1 if it is incorrect.

The total error $\nu(w)$, the training error $\nu_1(w)$ and the testing error $\nu_2(w)$ depend solely on the system w by means of the corresponding misclassification vector $q(w)$. Therefore we can simplify the notations and write these errors as $\nu(q)$, $\nu_1(q)$ and $\nu_2(q)$ when appropriate. With this convention, we can write:

$$Pr\left\{ \ \left| \ \nu_2(w^{S_1}) - \nu_1(w^{S_1}) \ \right| > \epsilon \ \right\} \ = \ Pr\left\{ \ \left| \ \nu_2(q^{S_1}) - \nu_1(q^{S_1}) \ \right| > \epsilon \ \right\} \tag{2}$$

where q^{S_1} is the misclassification vector produced by the parameter w^{S_1}.

For the sake of clarity, we are now going to assume that given n, n_1, a misclassification vector q, and a positive real number η, we can compute $\epsilon(\nu(q),n,n_1,\eta)$ such that:

$$Pr\{ \ | \ \nu_2(q) - \nu_1(q) \ | > \epsilon(\nu(q),n,n_1,\eta) \ \} = \eta \tag{3}$$

Function $\epsilon(\nu,n,n_1,\eta)$ will be derived in section 5. We can now rewrite equation (2) by replacing ϵ by $\epsilon(\nu(q^{S_1}),n,n_1,\eta)$:

$$\begin{aligned} & Pr\left\{ \ \left| \ \nu_2(w^{S_1}) - \nu_1(w^{S_1}) \ \right| > \epsilon(\nu(w^{S_1}),n,n_1,\eta) \ \right\} \\ = \ & Pr\left\{ \ \left| \ \nu_2(q^{S_1}) - \nu_1(q^{S_1}) \ \right| > \epsilon(\nu(q^{S_1}),n,n_1,\eta) \ \right\} \end{aligned} \tag{4}$$

4 Uniform Bound on the Error Deviation

Because q^{S_1} depends on the particular split of the dataset, there is no simple way to characterize the distribution (4) without introducing precise knowledge about the nature of the learning algorithm $\mathcal{A}$. Following [1] we will remove this dependency by seeking a uniform bound, i.e. a bound which is simultaneously valid for all the misclassification vectors produced by the learning algorithm $\mathcal{A}$ over all possible splits of the dataset. This set of misclassification vectors is defined by:

$$\mathcal{Q}_{\mathcal{A}}(S,n_1) \ = \ \{q = (Q(z_1,w),\cdots,Q(z_m,w)) \, , \ \ \forall w \in \mathcal{W}_{\mathcal{A}}(S,n_1)\} \tag{5}$$

where the symbol $\mathcal{W}_{\mathcal{A}}(S,n_1)$ denotes the set of all systems w obtained by running algorithm $\mathcal{A}$ on all possible training sets of size n_1 extracted from dataset S.

Set $\mathcal{W}_{\mathcal{A}}(S,n_1)$ is the smallest family of parameters that contains w^{S_1} for all possible choices of S_1. Therefore set $\mathcal{Q}_{\mathcal{A}}(S,n_1)$ is the smallest family of misclassification vectors that contains q^{S_1} for all possible choices of S_1. Since set $\mathcal{Q}_{\mathcal{A}}(S,n_1)$ always contain q^{S_1}, we can write:

$$\begin{aligned} & Pr\left\{ \ \left| \ \nu_2(q^{S_1}) - \nu_1(q^{S_1}) \ \right| > \epsilon(\nu(q^{S_1}),n,n_1,\eta) \ \right\} \\ \leq \ & Pr\{ \ \exists q \in \mathcal{Q}_{\mathcal{A}}(S,n_1), \ | \ \nu_2(q) - \nu_1(q) \ | > \epsilon(\nu(q),n,n_1,\eta) \ \} \end{aligned} \tag{6}$$

This uniform bound is much tighter than the usual Vapnik Chervonenkis uniform bounds [1] where a (large) family of systems is assumed to pre-exist prior to drawing the data sets. The narrow family $\mathcal{W}_{\mathcal{A}}(S,n_1)$ contains a considerable amount of information about the learning algorithm $\mathcal{A}$, and about its sensitivity to the choice of the training set. Furthermore, running the algorithm on two different training sets of size n_1 in general produces different parameters from $\mathcal{W}_{\mathcal{A}}(S,n_1)$, but may occasionally map to identical misclassification vectors in $\mathcal{Q}_{\mathcal{A}}(S,n_1)$. Several considerations of a different nature can also be used to show that the above bound is a near-equality, but a full discussion is beyond the scope of this paper.

We can now use the well-known inequality $Pr\{A \text{ OR } B\} \leq Pr\{A\} + Pr\{B\}$ to bound the right hand side of equation 6 as:

$$\begin{aligned} & Pr\{\ \exists q \in \mathcal{Q}_{\mathcal{A}}(S, n_1),\ \mid \nu_2(q) - \nu_1(q) \mid > \epsilon(\nu(q), n, n_1, \eta)\} \\ \leq & \sum_{q \in \mathcal{Q}_{\mathcal{A}}(S,n_1)} Pr\{\ \mid \nu_2(q) - \nu_1(q) \mid > \epsilon(\nu(q), n, n_1, \eta)\ \} \end{aligned} \tag{7}$$

Since the terms in the sum are all equal to η (c.f. equation (3)), the right hand side is equal to $\eta\,\mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$, where $\mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$ denotes the number of elements in set $\mathcal{Q}_{\mathcal{A}}(S, n_1)$, i.e. the number of different misclassification vectors obtained by running algorithm $\mathcal{A}$ on all the possible splits of S. The final result is then obtained by putting together the successive results (4), (6), (7):

$$Pr\{\ \left| \nu_2(w^{S_1}) - \nu_1(w^{S_1}) \right| > \epsilon(\nu(q), n, n_1, \eta)\ \}\ \leq\ \eta\ \mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1)) \tag{8}$$

This bound concerns the distribution of the error deviation when we consider all possible choices of training set and testing set. The result was obtained with no assumption regarding the origin or the independence of the $z_1, \ldots, z_n$. It does not require the ground truth hypothesis.

5 Distribution of the Error Deviation for a Single Misclassification Vector

We now proceed with the derivation of function $\epsilon(\nu, n, n_1, \eta)$ as defined in (3). Although there is no simple analytical expression for this function, we give various approximations and suggest numerical estimation procedures.

We need to compute the distribution $|\nu_2(q) - \nu_1(q)|$ for one particular misclassification vector q composed of $p = n\nu(q)$ ones and $n - p$ zeroes.

Let us gather all these ones and zeroes into a jar. We pick a training set by drawing a set of n_1 digits from the jar. Out of the $C_n^{n_1}$ possible drawings of these n_1 digits, there are exactly $C_p^k C_{n-p}^{n_1-k}$ ways to draw exactly k ones. The probability of drawing k ones therefore follows the *hypergeometric distribution*:

$$Pr\{\mathit{Drawing}\ k\ \mathit{ones\ in}\ n_1\ \mathit{digits}\} = \frac{C_p^k\ C_{n-p}^{n_1-k}}{C_n^{n_1}} \qquad \text{with } 0 \leq k \leq p$$

Since there are $n\nu(q)$ ones in the jar, when we have drawn exactly k ones in the training set, the difference between the number of ones in the training set and the test set will be

$$\mid \nu_2(q) - \nu_1(q) \mid\ =\ \left| \frac{n\,\nu(q) - k}{n_2} - \frac{k}{n_1} \right| \tag{9}$$

The cumulative distribution of the deviation between $\nu_1(q)$ and $\nu_2(q)$ over all possible splits is easily written by summing up the above counts for all the values of k that belong to the following set

$$K(\epsilon)\ =\ \left\{\ k \in \{0, \ldots, p\},\ \left| \frac{m\,\nu(q) - k}{n} - \frac{k}{l} \right|\ >\ \epsilon\ \right\} \tag{10}$$

We can therefore write:

$$Pr\{\ \mid \nu_2(q) - \nu_1(q) \mid > \epsilon\}\ =\ \sum_{k \in K(\epsilon)} \frac{C_{n\,\nu(q)}^k\ C_{n-n\,\nu(q)}^{n_1-k}}{C_n^{n_1}} \tag{11}$$

The function $\epsilon(\nu(q), n, n_1, \eta)$ that we set out to compute produces quantiles of the above distribution. Although there is no simple analytical expression for it, we can numerically tabulate $\epsilon(\nu(q), n, n_1, \eta)$. This task is somewhat facilitated by efficient numerical methods for computing the value of the cumulative distribution function of the hypergeometric distribution [2].

For more approximate, but more palatable estimates, we can draw from two results previously obtained by [1] when $n_1 = n_2$. A first result is derived from section A5 page 173 of [1]. This result gives an *absolute* upper bound. This bound is rather tight when $\nu \approx 0.5$ and n_1 is large enough:

$$\epsilon(\nu, n_1, 2n_1, \eta) \;\leq\; \sqrt{\frac{\log(2/\eta)}{n_1 - 1}} \tag{12}$$

A second result is derived from section A6 page 180 of [1]. This second result gives a *relative* upper bound. This bound is tighter when ν is small:

$$\epsilon(\nu, n_1, 2n_1, \eta) \;\leq\; 2\sqrt{\nu\frac{\log(2/\eta)}{n_1}} \tag{13}$$

Replacing these two results into our main result (8) gives the following inequalities:

$$Pr\left\{ \; \left| \; \nu_2(w^{S_1}) - \nu_1(w^{S_1}) \; \right| > \sqrt{\frac{\log(2/\eta)}{n_1 - 1}} \; \right\} \;\leq\; \eta \;\; \mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$$

and:

$$Pr\left\{ \; \left| \; \frac{\nu_2(w^{S_1}) - \nu_1(w^{S_1})}{\sqrt{\nu(w^{S_1})}} \; \right| > \sqrt{\frac{\log(2/\eta)}{n_1}} \; \right\} \;\leq\; \eta \;\; \mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$$

These two results should be seen as simple approximations of the more accurate result (8). The left hand sides of these two inequalities can be compared with the absolute and relative Vapnik Chervonenkis bounds.

Unlike the above results however, the Vapnik Chervonenkis bounds assume a predefined family of function, an underlying ground truth distribution, an independence hypothesis, and do not account for the properties of specific algorithms or specific datasets.

6 Data and Algorithm Dependent Bound

This result is incomplete without a discussion of $\mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$, i.e. the number of misclassification vectors reachable by algorithm $\mathcal{A}$ on the set of examples S. Each misclassification vector can be viewed as a dichotomy on the set of points $\{z_1, \ldots, z_n\}$ implemented by the loss function $Q(z, w)$ for some value of w.

According to the VC Dimension theory [1], the maximum number of dichotomies achievable, on any set of n points, by any function in a predefined family, is either equal to 2^n or bounded by $1.5n^h/h!$. The positive integer h is named *VC Dimension* of the family of functions. This result provides an obvious upper bound:

$$\mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1)) \;\leq\; 1.5\frac{n^h}{h!} \;\leq\; \left(\frac{ne}{h}\right)^h \tag{14}$$

The typical value of $\mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$ is much smaller than this bound. Instead of considering all possible sets of n examples, it only accounts for the set of the actual

n examples. Furthermore, instead of considering a large predefined family of functions, $\mathrm{Card}(\mathcal{Q}_{\mathcal{A}}(S, n_1))$ only considers the few functions reachable by a particular learning algorithm running on various training sets of size n_1 extracted from the actual n examples. Therefore, unlike the classical Vapnik Chervonenkis bounds, bound (8) is *data dependent* and *algorithm dependent*.

This observation clarifies results obtained in [3]. Empirical measurements of the uniform error deviation were shown to fit the algebraic expression of certain Vapnik Chervonenkis bounds, with a value for h smaller than the VC Dimension. These results led to conjecture the existence of a data dependent "effective VC Dimension".

Bound (14) also gives an insight on the tightness of inequality (7) which bounds the probability of a union of events by the sum of the individual probabilities. We are considering a union of a polynomial number of events (cf. equation (14)) whose probabilities are exponentially small (cf. equation (12)). Chances are that the overlap between these events shrinks quickly as n increases.

7 Comparison with Traditional VC Bounds

Although results (8) looks very similar to the traditional Vapnik Chervonenkis bounds, there are several important differences in both meaning and accuracy.

a. — No assumptions are made about the origin or the independence of the examples. Result (8) simply counts how many splits of the data set result in large deviations between the training error and the test error. This follows very precisely the testing procedures commonly used in the literature.

b. — Experiments could be designed to directly measure all the quantities involved in inequality (8). The value of these quantities depends only on the set of examples S and the algorithm $\mathcal{A}$. In fact such measurements have been attempted [3] with some success.

c. — The uniform error deviation only takes into account the systems that can be obtained by running a particular learning algorithm $\mathcal{A}$ on training sets extracted from a particular data set S. As a consequence, unlike the traditional VC bounds, bound (8) does not involve a *growth function* defined as a supremum on all the potential set of examples. Bound (8 depends only on the examples we have. This data dependent result can be likened to probability dependent bounds based on the *annealed VC entropy* [4].

d. — The traditional VC bounds must rely on additional bounding techniques such as Chernoff bounds or Hoeffding bounds [5]. Our discrete framework instead uses an *exact expression* (11) derived from the hypergeometric distribution.

e. — The traditional derivation of the VC proofs consists of two steps. The first step is essentially similar to the present work and provides a bound for the maximal error deviation between two disjoint sets of examples. The second step, which is rather intricate, transforms this result into the well known Vapnik Chervonenkis theorems using an additional inequality (c.f. [1] page 168). The present work avoids this second step, but as a consequence maintains an artificial symmetry between the training set and the test set. This symmetry shows up in (14) for instance, because the right hand side depends on $n = n_1 + n_2$ and therefore grows even when the training set size is fixed and the testing set size increases. This suggest that better results could be obtained by better accounting for the fact that the learning algorithm only gets to see the training set.

8 Conclusion

Result (8) has been so far presented as describing how a learning algorithm may withstand the usual testing procedure. We must then believe that this procedure is actually convincing enough. Considering all possible splits of our data set S into a training set S_1 and a test set S_2 is convincing enough for many problems such as isolated character recognition. This is clearly not an acceptable procedure for time series.

The splitting procedure can be re-interpreted as follows. We first reorder the n examples in set S using an arbitrary permutation. Then we use the first n_1 examples as a training set S_1 and the remaining n_2 examples as a testing set. Our testing procedure is convincing if we believe that all permutation are equally likely. In other words, we assume that all our examples can be *exchangeable* at will.

Exchangeability hypotheses have been studied in the theoretical Bayesian framework. Both objective and subjective Bayesians rely on the famous *exchangeability theorem* [6] to justify particular forms of the belief probability distribution. Disturbing results [7] however demonstrate that the Bayesian framework lacks a general discussion of the consistency issues. The classical framework now addresses such consistency issues using the Vapnik Chervonenkis theory (see [8] for instance). The present work demonstrates that exchangeability hypothesis is sufficient to derive results similar to the Vapnik Chervonenkis theory. This remark may lead to a new way to address consistency in the Bayesian framework.

The bounds presented here show that the concept of effective VC-dimension introduced in [3], i.e. the concept of a data-dependent and algorithm-dependent VC-dimension, can be justified from first principles with a minimal set of hypotheses. The present results can also be used to devise practical procedures to measure the parameters of learning curves in real situations.

References

[1] V. N. Vapnik. *Estimation of dependences based on empirical data.* Springer Series in Statistics. Springer Verlag, Berlin, New York, 1982.

[2] Trong Wu. An accurate computation of the hypergeometric distribution function. *ACM Transactions on Mathematical Software*, 19(1):33–43, March 1993.

[3] V. N. Vapnik, E. Levin, and Y. LeCun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

[4] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[5] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of American Statist. Ass.*, 58:13–30, 1963.

[6] B. De Finetti. La prévision, ses lois logiques, ses sources subjectives. *Ann. Inst. H. Poincaré*, 7:1–68, 1937.

[7] P. Diaconis and D. Freedman. On the consistency of Bayes estimates. *Ann. Statistics*, 14:1–67, 1986.

[8] M. Vidyasagar. *A Theory of Learning and Generalization with Application to Neural Networks and Control Systems.* Springer Verlag, 1997.