Error Estimation and Model Selection

Tobias Scheffer
Otto-von-Guericke-Universität
FIN/IWS, Universitätsplatz 2, 39106 Magdeburg, Germany
scheffer@iws.cs.uni-magdeburg.de

July 14, 1999

Abstract

In order to select a good hypothesis language (or *model class*) from a collection of possible model classes, we have to assess the generalization performance of the hypothesis which is returned by a learner that is bound to use a particular model class. This abstract of my doctoral dissertation deals with an analysis of the expected error rate of classifiers that leads to a new and very efficient way of assessing this error rate. Similar analyses can be applied to quantify the generalization performance of a holdout testing based model selection algorithm, and to quantify the optimistic bias of the error estimate which is imposed by running several learners on the same data set and selecting the one with the lowest holdout error rate. The analysis provides a model selection algorithm which can solve model selection (*e.g.,* feature subset selection) problems with as many as 10,000 attributes and 12,000 examples.

1 Introduction

In the setting of *classification learning*, the task of a *learner* is to approximate a joint distribution on *instances* and *class labels* as well as possible. For example, an *instance* could be an image of a hand-written character and a *class label* could be the character (perhaps represented by the corresponding ASCII code) which the writer intended to notate. In this example of the classification problem, we would like our learner to find a function (a *hypothesis*) that maps images of characters accurately to ASCII codes. "Accurately" means that we want the probability of the hypothesis returning an erroneous ASCII code to be small. At this point, the principle difficulty of classification learning arises: in order to measure the *true* (or *generalization*) *error rate* of our hypothesis (the probability of an erroneous response), we would need to "try it out" on all possible hand-written characters in the world. Unfortunately, we have only a small sample of labeled instances available from which we can measure the *empirical error rate* which is only an estimate of the true error rate. We know, however, how good an estimate the empirical error rate is: when the true error rate is ϵ, then the empirical error rate e is governed by the *binomial distribution* with mean value ϵ and variance $m\epsilon(1-\epsilon)$, where m is the sample size. All that a *learner* can do is to minimize this empirical error rate. Usually, a learner is constrained to a specific *model class* which is a set of potentially available hypotheses. For instance, a learner might be restricted to use a neural network with five hidden units, or a decision tree with seven leaf nodes. Let us assume that, within that model class, the learner selects the hypothesis with least empirical error rate. The empirical error rate of some of the hypotheses will be an optimistic estimate of the corresponding true error rate while the empirical error of others will be a pessimistic estimate. When selecting the hypothesis with the lowest empirical error, the learner is almost certain to choose a hypothesis with an optimistically biased empirical error rate. This bias will grow stronger when the number of distinct hypotheses in the model class grows. Therefore, the empirical error rate of the apparently best hypothesis does not give us any information on that hypothesis' true error rate. We can often fix this problem by using an *independent* sample to assess the hypothesis which is returned by the learner (referred to as *holdout testing or cross validation*). We can then try out various model classess, assess the resulting hypothesis returned by the learner and then select the model class that gaves us the lowest cross validation error. This process of selecting the model class that leads to optimal generalization is usually referred to as *model selection*. From the engineering point of view, the main drawback of cross validation is that it is not a good solution for large-scale learning problems because the learner has to be run at least once for each model class. From a scientific point of view, despite its ability to provide us with an estimate of the learning curve, cross validation does not help us to *understand* what is going on.

2 Expected Error Analysis

We have the laws of mechanics that provide us with a mathematical model of the behavior of physical items; these laws can be used to *predict the behavior* of physical objects. However, we do not have a good model that characterizes the behavior of *learning algorithms* and that can be applied to predict, for instance, the error rate of a specific learner for a given problem. Since we do not have to accelerate or move objects in order to find out what would happen if we applied a force to them, from an analogous theory of learning algorithms we would expect to become able to predict learning curves without actually running the learner. What we have by now are *worst-case* bounds which basically say that the greatest difference between true and empirical error of any hypothesis in some model class can be no more than a certain value. Such *PAC*- or *VC-style* [2, 7] bounds define a (fairly wide) interval within which all learning curves of all possible learners for any learning problem must lie with high probability. But *exactly where* in this interval an actual learning curve lies depends on the given problem, and typically the learning curves are far away from the boundaries.

Perhaps surprisingly, the *expected* behavior (expected over all samples of the given size) of a learning algorithm (with respect to an *actual*, given learning problem) can be characterized relatively easily, although empirical error rate and complexity of the model class alone do not suffice to predict the exact error rate [1]. The key observation is that the error rate of the empirical error minimizing hypothesis is determined by a certain joint property of the learning problem and the model class which the learner uses: the distribution of error rates of hypotheses in the model class. For every error ϵ, this distribution sais how many hypotheses in the model class incur that particular error with respect to the given problem. This (one-dimensional, typically bell-shaped) distribution provides us with the information necessary to get from *worst-case error bounds* to the *actual* error.

Theorem [Scheffer & Joachims, 1999] *Let L be a learner that minimizes the empirical error rate within a finite model class H_i (breaking ties at random). The distribution that governs the true error rate of h_L (the hypothesis returned by L) – and thereby the expected error rate of h_L – is uniquely determined by (and can be computed from) three quantities: (a) the sample size, (b) the size of the model class, and (c) the distribution of error rates of the hypotheses in the model class.*

The theorem which characterizes the behavior of learning algorithm unfortunately depends on a quantity which is not always known in advance (the distribution of error rates in the model class), just like the law of acceleration depends on a quantity which is not known either (mass). However, both quantities (the distribution of error rates in the model class for machine learning, and mass for mechanics) can be *estimated* and, in most cases, estimation is quite easy. There is an empirical counterpart of this distribution: namely, the distribution of empirical error rates of hypotheses in the model class which counts how often each possible empirical error value occurs in the model class. This discrete, one-dimensional (often bell-shaped) distribution can usually be estimated by simply recording the empirical error rates of $O(\log m)$ randomly drawn hypotheses. Hence, the key feature of the theorem is that it provides us with an estimate of the true error rate of the hypothesis returned by an empirical error minimizing rate *without* us having to run the learner at all. Experimental results [6, 4] show that the error estimate is often at least as accurate as an estimate obtained by 10-fold cross validation. Since the estimate is obtained very efficiently we now have a means of conducting model selection that can be applied in cases in which n-fold cross validation cannot. In fact, we can solve model selection problems with as many as 12,000 examples and 10,000 attributes. [6, 4].

3 Holdout Testing Based Model Selection

In the previous section, we have seen that there is a mathematical model that characterizes the actual *behavior* of exhaustive, empirical error minimizing learners quite accurately. We can use it, for instance, to predict which model class will lead to optimal results for a given problem. So how about more sophisticated learners; can we, perhaps, find a model of the behavior of holdout testing based learning algorithms and use that model to construct an optimal such learner for a given problem?

A holdout testing based learner splits the available sample into a training and a holdout part. For every model class in a given collection of model classes (this collection may, for instance, contain neural networks with one through k hidden units, the ith model class containing networks with exactly i hidden units) a hypothesis is generated using the training part of the data. The resulting hypotheses (one for each model class) are compared using the holdout part of the data. The model class with the lowest holdout error rate is then chosen and the hypothesis with least empirical error in that model class (this time using the complete sample) is finally returned.

Theorem [Scheffer 1999] *Let L be a learner that first uses holdout testing (with training set size m' and holdout set size m'') to determine an optimal model class, and then minimizes the empirical error rate (using $m' + m''$) within the chosen model class. The distribution that governs the true error rate of h_L – and thereby the expected error rate of h_L – is uniquely determined by (and can be computed from) these quantities: (a) the sample sizes m' and m'', (b) the size of the model classes, and (c) for each model class in the considered collection, the distributions of error rates of the hypotheses in that model class.*

Again, sample size and the size of the model classes are known whereas the distributions of error rates in the model classes have to be estimated for the theorem to be practically applicable. Experiments have shown [4] that the theorem predicts, for instance, the optimal training/test split ratio reliably. Together, the two theorems predict whether a simple empirical error minimization based learner will, for a given learning, do better or worse than a holdout testing based learner. Given several possible collections of models, the theorem can be applied to estimate which collection of models will lead to the lowest generalization error rate. Like in the previous section, we obtain the error estimate without having to invoke the learner.

4 Quantifying Parameter Adaptation Bias

Many "practical" learning algorithms possess complexity regularization parameters, such as pruning thresholds or weight decay terms that have to be adjusted for each new learning problem. Typically, this is achieved by holdout testing or n-fold cross validation. Several distictly parameterized versions of the learner are started and the resulting hypotheses are compared using the same holdout set, or the same cross validation data. What happens now is very similar to overfitting that occurs during the process of empirical error minimization: the holdout (or cross validation) error rates of some parametrizations are optimistic estimates of the true error rate while the holdout error rates of others are pessimistic. When we select the parametrization that gave us the lowest holdout error and publish the observed holdout error rate, we are likely to select a hypothesis (and publish a result) that is subject to an optimistic bias. It would be interesting to known just how strong this bias is. We can easily quantify the bias using Chernoff bounds; unfortunately, the resulting bounds are too crude to be of much practical relevance (the resulting equation reads like "a sample size which is enormous plus linear in the number of parameters is required to make sure that the bias is reasonably small"). The following consideration, however, leads to an accurate quantification of the *parameter adaptation bias* [5]: Suppose that the entropy of the holdout data is zero (*i.e.,* all class labels are equal). Even if our hypothesis is really arbitrarily inaccurate, we can "guess" all holdout labels if we try out just two distinct hypotheses. If the entropy of the holdout data is one, then we need to try out two hypotheses to "guess" one holdout class label. We can turn these simple considerations into a theorem that quantifies the parameter adaptation bias for both holdout testing and n-fold cross validation, based on some measurable properties of the data set, such as entropy. It turns out that the bias is particularly strong when the difference between the lowest observed error rate and the error rate of the default classifier is small, when the number of parameters is large, and, of course, when the sample size is small. For some of the *UCI repository* data sets (in particular, for problems studied in *Inductive Logic Programming*) many published results are likely to be strongly optimistically biased. In such cases, an almost unbiased estimate of the true error rate can be obtained by nesting two separate loops of n-fold cross validation (resulting in an n^2-*fold cross validation* setting). In the outer loop, only one single hypothesis is assessed on the hold out data at each fold which avoids an optimistic bias at the cost of computational expenses of $O(n^2)$ instead than $O(n)$.

5 Conclusion

The expected error analysis characterizes the behavior of learning algorithms for a given learning problem – as opposed to PAC analysis which is *worst-case* with respect to the learning problem and the learner. A similar characterization of the *actual-case* behavior of a Naive Bayesian learner has recently been given by Langley and Sage [3]. But their analysis can only be applied when the target function is known whereas expected error analysis is based on the distribution of error rates in the model class which can be estimated from the sample. The negative result of [1] which bounds the performance of all complexity penalization based model selection algorithms does not hold for expected error analysis based model selection (because we base our estimate of the learning curve on more information). Since we do not run the learner to obtain an error estimate of the resulting hypothesis we can solve model selection problems which cannot be solved by cross validation due to a prohibitive amount of necessary computation.

Short CV. Tobias Scheffer studied computer science at the Technische Universität Berlin, where he received his doctoral degree in 1999. He became an Ernst von Siemens fellow in 1996 and has since then worked at the Technische Universität Berlin, Siemens Corporate Research, Princeton, and the University of New South Wales, Sydney. Since 1999, he is a lecturer at the Otto-von-Guericke-Universität, Magdeburg.

References

[1] M. Kearns, Y. Mansour, A. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning Journal*, 27:7–50, 1997.

[2] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[3] P. Langley and S. Sage. Tractable average case analysis of naive bayes classifiers. In *ICML-99*, pages 220–228, 1999.

[4] T. Scheffer. Error estimation and model selection, 1999. Doctoral Dissertation, Technische Universitaet Berlin, School of Computer Science. To appear as a book (hopefully) soon; for ordering information, please contact the author.

[5] T. Scheffer and R. Herbrich. Unbiased assessment of learning algorithms. In *IJCAI-97*, pages 798–803, 1997.

[6] T. Scheffer and T. Joachims. Expected error analysis for model selection. In *Proceedings of the International Conference on Machine Learning (ICML-99)*, 1999.

[7] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.