Evaluating Evaluation Measures

Amit Dhurandhar ASD@CISE.UFL.EDU
Alin Dobra ADOBRA@CISE.UFL.EDU
Computer and Information Science and Engineering,
University of Florida, Gainesville, FL 32611 USA

Abstract

Model selection measures such as hold-out set error, cross-validation error, leave-one-out error etc. are used to evaluate the performance of a classification algorithm on a given data set. To get an accurate estimate of the performance it is important that we choose the appropriate model selection measure or evaluation measure for the setting of interest. In this paper, we describe in brief a recently introduced methodology, which can be used to accurately and efficiently study the behavior of such evaluation measures in relevant settings. We also discuss the implications of having such a methodology as an exploratory tool and the potential challenges for the future.

1. Introduction

The problem of automated classification is omnipresent in todays world. Various domains ranging from health care to finance require efficient and accurate classification tools (i.e. classification algorithms) that perform inference on huge amounts of available data. To evaluate the accuracy of these classification algorithms a number of evaluation measures have been developed in literature (Plutowski, 1996; Devroye et al., 1996). Some of the most popular evaluation measures are hold-out set error, cross-validation error and leave-one-out error. It would be interesting and useful to study the behavior of these measures under different settings, such as their behavior for different classification algorithms with varying dataset size or with varying amounts correlation between the input and output attributes or by varying parameters that are specific to the particular evaluation measure (for example, test set size in hold-out set or number of folds in cross-validation). Such studies would most likely lead to more informed decisions when choosing an appropriate measure in a real life setting, resulting in better evaluation of the chosen classification algorithms and hence improved decision making in the practice of model selection. Considering the potential impact of such studies, we in this paper, discuss a recently proposed *semi-analytical* methodology which can be used to study the behavior of these measures accurately and efficiently for settings such as the ones just mentioned.

The rest of the paper is organized as follows: In the next section we motivate and briefly describe this methodology. In Section 3, we discuss applications of the methodology. We conclude in section 4, by looking at possible roadblocks and explore potential research opportunities for the future.

2. Methodology

We motivate the methodology by first explaining the underlying philosophy it is based on, following which we provide a brief overview of the methodology.

2.1. Underlying Philosophy

The two prevalent approaches to study learning algorithms are either based on theory or on empirical studies but usually not both[1]. While both methods are powerful in themselves, each suffers from at least a major deficiency.

The theoretical method depends on *nice* closed form formulae that restricts the types of results that can be obtained to asymptotic results (Shao, 1993) or statistical learning theory type of results (i.e. distribution free bounds) (Vapnik, 1998; J. Shawe-taylor & Anthony, 1998). These results are usually weak making them less applicable in practice.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1] unless empirical studies are used to validate the theory.

The empirical method is well suited for validating intuitions but is significantly less useful for finding novel, interesting things, since large number of experiments have to be conducted in order to reduce the error to a reasonable level. This is particularly true when small probabilities are involved, making the empirical evaluation impractical in such a case.

An ideal scenario, from the point of view of producing interesting results, would be to use theory to derive computationally efficient but potentially uninterpretable formulae and to use experiments to interpret these formulae. This would avoid the limitation of theory to use only *nice* formulae which leads to weak results and the limitation of empirical studies to perform large number of experiments. Thus, the role of the theory could be to significantly reduce the amount of computation required and the role of experiments (through visualization) to understand the potentially complicated theoretical formulae. This is the philosophy behind the methodology introduced in (Dhurandhar & Dobra, 2009); a new hybrid method to characterize and understand models and model selection measures. The work we discuss here is an initial forray into what might prove to be an useful tool for studying learning algorithms. We call this method semi-analytical, since not just the formulae, but visualization in conjunction with the formulae leads to interpretability. What makes such an endeavor possible is the fact that, mostly due to the linearity of expectation, moments of complicated random variables can be computed accurately and efficiently even though computing the exact distribution efficiently, is a daunting task.

2.2. Technical Overview

Consider the problem of estimating how a given classification algorithm performs on a given joint distribution over the input-output space ($X \times Y$). As opposed to the general setup in machine learning where the distribution is unknown and only independent and identically distributed (i.i.d.) samples are available, in this scenario, *in principle*, the behavior of classification algorithm can be accurately studied. Solving this problem efficiently, offers an alternative line of study for classification algorithms and potentially unique insights into the *non-asymptotic* behavior of other machine learning algorithms.

While the problem of estimating classification algorithm performance on a given distribution might look simple, solving it efficiently poses significant technical hurdles. The most natural way of studying a classification algorithm would be to sample N datapoints from the given distribution, train the algorithm to produce a classifier, test the classifier on a few sampled test sets and report the average error computed over these test sets. A shortcoming of the above approach is that based on just one single instance of the algorithm (since the algorithm was trained on a single data set of size N) we conclude about its general behavior. A straightforward extension of the above approach to make the results more relevant in studying the algorithm would be to sample multiple data sets of size N, train on each of them to produce different classifiers, compute the test error for each of the classifiers and calculate the average and variance of the obtained test errors. This procedure would be a better indicator of the behavior of the algorithm than the previous case since we study multiple instances of the algorithm than just an isolated instance. Ideally, we would want to study the behavior of the algorithm by training it on all possible data sets of size N producing a variety of classifiers and then evaluating the expected value and variance of the generalization error (GE) of each of these classifiers. The GE of a classifier ζ is given by, $GE(\zeta) = E\left[\lambda(\zeta(x), y)\right]$, where $\lambda(.,.)$ is a 0-1 loss function, x is an input and y is an output and the expectation is over the input-output space $X \times Y$. The expected value and variance of GE over all possible classifiers are denoted by, $E_{\mathcal{Z}(N)}\left[GE(\zeta)\right]$ and $Var(GE(\zeta))$ respectively. Here $\mathcal{Z}(N)$ represents the space of all possible classifiers produced by training the classification algorithm on all data sets of size N, drawn from the joint distribution. Thus, the moments provide a natural and informative avenue for studying classification algorithms.

In similar fashion, moments of the evaluation measures such as moments of hold-out error (HE) and moments of cross-validation error (CE) (leave-one-out is just a special of cross-validation, when the number of folds is N) computed over all datasets of size N and over all possible splits into training and testing, also provide important information regarding the non-asymptotic behavior of these measures. HE is formally defined as, $HE = \frac{1}{N_s}\sum_{(x,y)\in D_s} \lambda(\zeta(x), y)$ where D_s and N_s denote the test set and the size of the test set respectively. CE is formally defined as, $CE = \frac{1}{v}\sum_{i=1}^{v} HE_i$ where v is the number of folds and HE_i is the hold-out error on the i^{th} fold.

The important question now is; can we compute these moments efficiently and accurately? In our previous work (Dhurandhar & Dobra, 2009), we presented a general framework for computing these quantities for an arbitrary classification algorithm efficiently and accurately. By extensive use of the linearity of expectation and change of the order of sums (and integrals),

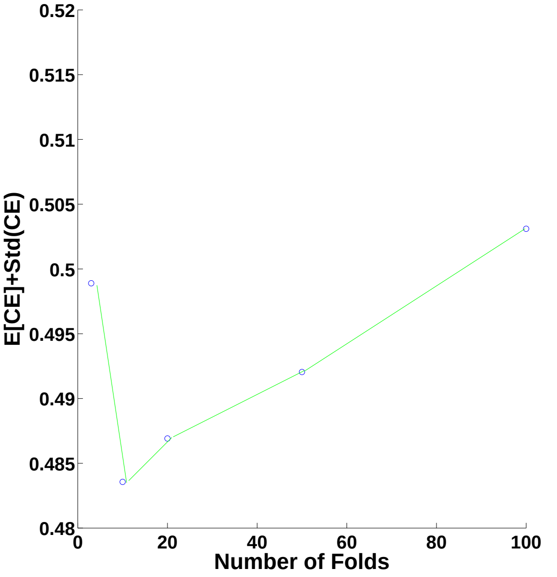

Figure 1. Behavior of CE for NBC when sample size is small and input-output correlation is low.

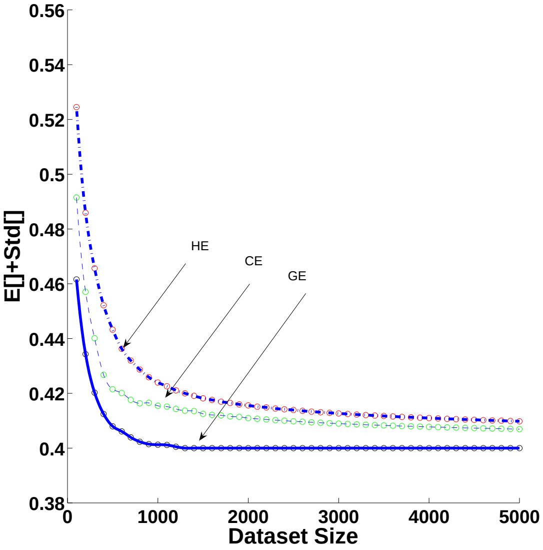

Figure 2. Convergence behavior of HE and CE towards GE for NBC.

the moments of GE can be expressed in terms of the behavior of the classification algorithm on specific inputs rather than on the whole space, thus reducing the complexity from an exponential in the size of the input space to linear for the computation of the first moment and quadratic for the second moment without loss of accuracy. In this work, we also drew relationships between the moments of GE and moments of HE and CE, thus enabling us to compute the moments of these evaluation measures efficiently and accurately as well. We then customized the generic expressions for the moments to particular classification models namely; Naive Bayes Classifier (NBC) (Dhurandhar & Dobra, 2009), Decision Trees (DT) (Dhurandhar & Dobra, 2008c) and k-Nearest Neighbor Classifier (KNN) (Dhurandhar & Dobra, 2008b), thus allowing us to study the behavior of the moments of GE, HE and CE for these algorithms in settings of interest. It was shown in the relevant prior works that estimating moments using these customized expressions is a more viable alternative compared to estimating them directly using Monte Carlo or to other evaluation methods such as distribution free bounds. The primary reason these expressions are more accurate than Monte Carlo, is that the parameter space of the individual terms in these customized expressions is significantly smaller than the entire space over which the moments are computed. The reason they are preferable to distribution free bounds such as SLT type bounds, is that the class of classifiers in our case which is induced by training the classification algorithm on i.i.d. samples of size N, is much smaller and more tightly coupled to the behavior of the algorithm on these samples than the class of classifiers considered in SLT type bounds (for example, bounds based on Vapnik-Chervonenkis dimension).

3. Applications

We now discuss the potential benefits of having such a methodology.

Gaining Insight: One of the main advantages of deploying such a methodology is that it can be used as an exploratory tool and as an analysis tool. We can accurately study when and why a particular evaluation measure or classification algorithm behaves in the manner it does.

For instance, in (Dhurandhar & Dobra, 2008a) we studied the behavior of cross-validation and provided interesting explanations for its behavior with respect to varying sample size, varying number of folds and varying amount of correlation between the input and output attributes. There, we were able to explain when and why we observe the ”V-shaped” behavior of cross-validation (i.e. when and why performance of cross-validation is best around intermediate (10-20) folds) shown in figure 1, by relating its behavior with the behavior of the covariances of CE between pairs of runs.

Finite Sample Convergence: Another benefit of the methodology is that it can be used to evaluate the performance of the evaluation measures in estimating GE under different conditions. For example, as shown in figure 2, we can study how well HE and CE estimate GE with increasing sample size. We can thus use CE below a certain sample size and HE beyond that sample size so as to estimate GE accurately and efficiently. The methodology can thus be used as a guidance tool.

Robustness: If an algorithm designer validates his/her algorithm by computing moments as mentioned earlier, it can instill greater confidence in the practitioner searching for an appropriate algorithm for his/her dataset. The reason for this being, if the practitioner has a dataset which has a similar structure or is from a similar source as the test dataset on which an empirical distribution was built and favorable results reported by the designer, then this would mean that the good results apply not only to that particular test dataset, but to other similar type of datasets and since the practitioner's dataset belongs to this similar collection, the results would also apply to his. Hence, the robustness of the algorithm can be evaluated using this methodology which can result in the algorithm having wider appeal.

Other Benefits: The methodology can be used to evaluate Probably Approximately Correct (PAC) Bayes bounds (McAllester, 2003) in certain settings. Roughly speaking, PAC Bayes bounds, bound the difference between the expected GE and expected empirical error where the expectation is over a distribution defined over the hypothesis space. In our case this distribution is induced by training a classification algorithm on all i.i.d. samples of size N. We can compute the moments of GE and the moments of the evaluation measures using our expressions for this case and compare them to verify the tightness of the corresponding PAC Bayes bounds.

The derived expressions can also be used to focus on specific portions of the data, since the individual probabilities in the expressions are only concerned with the behavior of the classification algorithm or evaluation measure on single or pairs of inputs.

4. Discussion

In the previous sections we argued that the methodology we discussed, can serve as a guidance tool, as an analysis tool and as an exploratory tool to accurately study classification algorithms in conjunction with evaluation measures. In the future it would be interesting to analyze and develop efficient characterizations for other classification algorithms and evaluation measures in this framework. This analysis will hopefully assist us in gaining new insights into the behavior of these techniques. A more ambitious goal is to extend this kind of analysis to study the more general class of learning algorithms.

A drawback of this methodology is that results are technique specific and scalable customized expressions can be tedious to obtain for arbitrary learning algorithms. We believe however, that studies such as these hold the key to delving deep into the non-asymptotic statistical behavior of learning algorithms.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0448264.

References

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition.* Springer.

Dhurandhar, A., & Dobra, A. (2008a). Insights into cross-validation. submitted.

Dhurandhar, A., & Dobra, A. (2008b). Probabilistic characterization of nearest neighbor classifiers. Tech. Report at www.cise.ufl.edu/∼asd/nnj.pdf.

Dhurandhar, A., & Dobra, A. (2008c). Probabilistic characterization of random decision trees. *Journal of Machine Learning Research*, *9*, 2321–2348.

Dhurandhar, A., & Dobra, A. (2009). Semi-analytical method for analyzing models and model selection measures based on moment analysis. *ACM Transactions of Knowledge Discovery and Data Mining*, *3*.

J. Shawe-taylor, P. Bartlett, R. W., & Anthony, M. (1998). Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, *44*, 1926–1940.

McAllester, D. (2003). Pac-bayesian stochastic model selection. *Mach. Learn.*, *51*.

Plutowski, M. (1996). Survey: Cross-validation in theory and in practice. www.emotivate.com/CvSurvey.doc.

Shao, J. (1993). Linear model selection by cross validation. *JASA*, *88*.

Vapnik, V. (1998). *Statistical learning theory.* Wiley & Sons.