

# Study of Classification Algorithms using Moment Analysis

Amit Dhurandhar  
University of Florida  
asd@cise.ufl.edu

Alin Dobra  
University of Florida  
adobra@cise.ufl.edu

## ABSTRACT

In this short paper we briefly discuss a moment based method that was recently introduced to study the behavior classification algorithms and model validation techniques for finite sample sizes. The method involves accurate and efficient computation of the moments of the generalization error which are over the space of all possible datasets of size  $N$  drawn from an underlying distribution. A classification algorithm trained on each of these datasets induces a space of classifiers (i.e. an empirical hypothesis space) and the moments can be equivalently computed over this space. In our previous work we also drew relationships between the moments of the generalization error and moments of hold out error, cross-validation error, leave one out error and hence these model validation techniques can also be studied accurately by our method. The primary goal of this paper is to familiarize machine learning researchers with this newly proposed methodology, so as to discuss its implications regarding important problems such as classification model selection.

## 1. INTRODUCTION

Let us consider the following question: how does a given classification algorithm behave with respect to a given joint distribution over the input-output space ( $X \times Y$ ) in the non-asymptotic regime? This is different from the general setup in machine learning where the distribution is unknown and only independent and identically distributed (i.i.d.) samples are available. If this problem is solved accurately and efficiently, it offers an alternative line of study for classification algorithms and potentially unique insights into the *non-asymptotic* behavior of learning algorithms.

A natural solution to solve the above problem is to sample multiple datasets of size  $N$ , train the chosen classification algorithm on each of them to produce potentially multiple classifiers, compute the empirical error for each of these classifiers and report the average and variance of these empirical errors as an indicator of the performance of the algorithm for that particular weighting of datasets or joint distribution. Ideally, we would want to train the algorithm on all possible datasets producing the corresponding classifiers, compute the generalization error ( $GE$ ) rather than empirical error for each classifier and report the expected value and variance of  $GE$  for all these classifiers. Accurate and efficient computation of these moments is what we review in this paper. However, before we concern ourselves with strategies to compute these moments accurately and efficiently, we discuss the benefits of computing them.

As mentioned before one of the benefits of computing these moments is that classification algorithms can be closely studied with respect to different distributions in the non-asymptotic regime. Another application is regarding studying robustness of algorithms and applicability of results. If an algorithm designer validates his/her algorithm by computing these moments, it can instill greater confidence in the practitioner searching for an appropriate algorithm for his/her dataset. This is because, if the practitioner has a dataset

which has a similar structure or is from a similar source as the dataset on which an empirical distribution was built and favorable results (i.e. low expected value and variance) reported by the designer, then this would mean that the results apply not only to that particular dataset, but to other similar types of datasets and since the practitioner's dataset belongs to this similar collection, the results would closely apply to his. Thus, the moments can be used as a tool to study and evaluate the behavior of classification algorithms in real life settings (i.e. over finite size samples).

In the rest of the paper we first compare our approach with other formal frameworks. We then discuss an important result and current successes of our approach. Finally, we conclude in section 4.

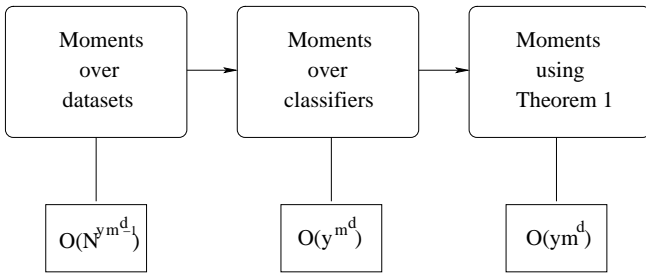
## 2. CLASS OF CLASSIFIERS

Vapnik-Chervonenkis theory (VC theory) [7] categorizes classification algorithms (rather learning algorithms) into different classes called Concept Classes. The concept class of a classification algorithm is determined by its VC dimension which is related to the shattering capability of the algorithm. The bounds on the  $GE$  of a classification algorithm derived by this theory are functions of the VC dimension, the sample size and training error. The strength of this technique is that by finding the VC dimension of an algorithm we can derive error bounds for the classifiers built using this algorithm, without ever referring to the underlying distribution. A fallout of this very general characterization is that the bounds are usually loose [4] which in turn result in making statements about any particular classifier weak. In an attempt to circumvent this problem the luckiness framework [6] was introduced. In this framework a function (called luckiness function) can be defined by the practitioner, which maps all relevant classifier-dataset pairs to a positive real value. Higher the value more likely that the particular classifier will perform well on the corresponding dataset. If the defined luckiness function is "correct", the bounds are valid and much tighter than those found by VC theory. This function (with constraints such as smoothness) however, may not be easy to find in practice since it is more or less equivalent to the problem of finding an appropriate prior in bayesian methods which we know is many times a challenge.

In our methodology we define a class of classifiers which is induced by a given classification algorithm trained on i.i.d. data of a given size. Any member of this class can be viewed as a sample classifier and the characterization of the class is strongly connected to the behavior of the classifier. This class of classifiers are much smaller than the classes considered in VC theory and hence the results are significantly tighter. The downside of our method is the fact that we lose the strength to make generalized statements to the extent that VC theory makes.

## 3. RESULTS

Let us now introduce some notation and define basic concepts such generalization error.  $X$  is a random vector mod-



**Figure 1: Number of terms for three methods that analytically compute the first moment are shown above. For the second moment the number of terms is just the square of the above complexity. It can be seen that our result is an efficient alternative.**

eling input whose domain is denoted by  $\mathcal{X}$ .  $Y$  is a random variable modeling output whose domain is denoted by  $\mathcal{Y}$  (set of class labels).  $Y(x)$  is a random variable modeling output for input  $x$ .  $\zeta$  represents a particular classifier with its GE denoted by  $GE(\zeta)$ .  $\mathcal{Z}(N)$  denotes the space of classifiers obtained by application of a classification algorithm to different samples of size  $N$ . Then  $GE(\zeta)$  is defined as,  $GE(\zeta) = E[\lambda(\zeta(x), y)]$  where  $\lambda(\cdot, \cdot)$  is a 0-1 loss function,  $x$  is an input and  $y$  is an output and the expectation is over the input-output space  $X \times Y$ . We now present the analytical formulas for computing the first two moments of  $GE$  efficiently. This result is under Theorem 1 in [2].

$$E_{\mathcal{Z}(N)} [GE(\zeta)] = \sum_{x \in \mathcal{X}} P[X=x] \sum_{y \in \mathcal{Y}} P_{\mathcal{Z}(N)} [\zeta(x)=y] P[Y(x) \neq y] \quad (1)$$

$$E_{\mathcal{Z}(N)} [GE(\zeta)^2] = \sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} P[X=x] P[X=x'] \cdot \sum_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} P_{\mathcal{Z}(N)} [\zeta(x)=y \wedge \zeta(x')=y'] P[Y(x) \neq y] P[Y(x') \neq y'] \quad (2)$$

Note that the probabilities in the above equations after summation over  $\mathcal{Y}$  are conditionals given a particular  $x$ . Also note that the formulas are applicable to the continuous domain by replacing summations over  $\mathcal{X}$  with the corresponding integrals. The probabilities of the form  $P_{\mathcal{Z}(N)}[\cdot]$  are the only terms in the above equations that depend both on the classification algorithm and the underlying distribution. The other terms depend only on the underlying distribution. Hence, if we have to find the moments for a particular classification algorithm we only need to characterize these probabilities which is essentially the behavior of a classification algorithm on individual inputs for the first moment and pairs of inputs for the second moment.

We say the above equations are efficient since the other alternatives for analytically computing the moments are significantly more expensive. For example the most obvious alternative is to compute the moments over all datasets of size  $N$ . The other alternative is to compute the moments over all classifiers as above but without the simplifications that have led to equations 1 and 2. If we assume that there are  $d$  attributes each having  $m$  distinct values, then the complexity of these two alternatives and our formulas are shown in Figure 1. There is practically an exponential gain in speed without compromising on accuracy with our method.

Another non-analytical alternative is to simply perform monte carlo simulations and compute the moments. From our studies on three algorithms namely; Naive bayes classifier [2], Random decision trees [3] and K-nearest neighbor [1] we have found that even when the probabilities in equations 1 and 2 are approximated using monte carlo the accuracy is still higher than computing the moments directly using monte carlo for the same amount of computation. The reason for this is that the parameter space of the individual probabilities is generally much smaller than the entire space over which the moments are computed. In fact for Random decision trees the moments computed using the above equations were considerably more accurate than Breimans bounds based on strength and correlation [5].

Using our closed form formulas for the moments of  $GE$  and the relationships between these moments and the moments of cross validation error ( $CE$ ) and hold out error, we were able to study the behavior of these errors. In these studies we observed finite sample convergence behavior of these errors to the generalization error. We were also able to reproduce the V-shaped behavior (i.e. error is least at around 10-20 folds and higher for lower and larger folds) of the cross validation error for small sample sizes and low input-output correlation with our formulas. We provided an explanation for this behavior which is closely linked to the behavior of the covariance between pairs of runs of cross validation. This shows that the moments can also be used as a tool to gain insights into some popular prevalent techniques.

## 4. CONCLUSION

In summary, we have briefly discussed a recent methodology that was introduced to study classification algorithms and model validation techniques for finite sample sizes. It remains to be seen how the analysis can be applied in an efficient manner to other learning algorithms. Though characterizing the probabilities in the moments that depend on the classification algorithm can be a tedious process, we believe that for accurate non-asymtotic studies of learning methods the approach has merit.

## Acknowledgements

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

## 5. REFERENCES

- [1] A. Dhurandhar and A. Dobra. Probabilistic characterization of nearest neighbor classifiers. Tech. Report at [www.cise.ufl.edu/~asd/nmj.pdf](http://www.cise.ufl.edu/~asd/nmj.pdf).
- [2] A. Dhurandhar and A. Dobra. Semi-analytical method for analyzing models and model selection measures based on moment analysis. To appear in ACM Transactions of Knowledge Discovery and Data Mining. [www.cise.ufl.edu/~asd/paper.pdf](http://www.cise.ufl.edu/~asd/paper.pdf).
- [3] A. Dhurandhar and A. Dobra. Probabilistic characterization of random decision trees. *Journal of Machine Learning Research*, 9:2321–2348, 2008.
- [4] S. Boucheron and O. Bousquet and G. Lugosi. Introduction to statistical learning theory. [www.kyb.mpg.de/publications/pdfs/pdf2819.pdf](http://www.kyb.mpg.de/publications/pdfs/pdf2819.pdf), 2005.
- [5] L. Breiman. Random forests. <http://oz.berkeley.edu/users/breiman/randomforest2001.pdf>, 2001.
- [6] R. Williamson and M. Anthony J. Shawe-taylor, P. Bartlett. Structural risk minimization over

data-dependent hierarchies. *IEEE transactions on Information Theory*, 44:1926–1940, 1998.

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