A Method for Discovering the Insignificance of One's Best Classifier and the Unlearnability of a Classification Task

George Forman
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2002-123 (R.2)
July 11th , 2002*

E-mail: gforman@hpl.hp.com

supervised machine learning, overfitting, 2001 KDD Cup thrombin classification competition

Consider the following common scenario: a data mining practitioner tries various specialized classification algorithms on a new dataset of unknown difficulty and selects the apparent best. Supposing its accuracy were 70% on a held-out test set, how can one know whether this is a significant result or not? It can be difficult to tell in the absence of standard benchmark results for the dataset. Surprisingly, it can also be difficult to tell even when the dataset has hundreds of benchmark results. This paper presents a method to address this question by comparing the chosen best classifier to the distribution of performance scores obtained by many simple classifiers that are randomly generated. This can also serve to discover when a classification problem appears nearly unlearnable. It is demonstrated for the results of the 2001 KDD Cup thrombin competition.

* Internal Accession Date Only Approved for External Publication
To be published in and presented at Data Mining Lessons Learned Workshop, the 19th International Conference on Machine Learning (ICML), 8-12 July 2002, Sydney, Australia

A Method for Discovering the Insignificance of One's Best Classifier and the Unlearnability of a Classification Task

George Forman GFORMAN@HPL.HP.COM
Hewlett-Packard Labs, 1501 Page Mill Rd. MS 1143, Palo Alto, CA 94304 USA

Abstract

Consider the following common scenario: a data-mining practitioner tries various specialized classification algorithms on a new dataset of unknown difficulty and selects the apparent best. Supposing its accuracy were 70% on a held-out test set, how can one know whether this is a significant result or not? It can be difficult to tell in the absence of standard benchmark results for the dataset. Surprisingly, it can also be difficult to tell even when the dataset has hundreds of benchmark results. This paper presents a method to address this question by comparing the chosen best classifier to the distribution of performance scores obtained by many simple classifiers that are randomly generated. This can also serve to discover when a classification problem appears nearly unlearnable. It is demonstrated for the results of the 2001 KDD Cup thrombin competition.

1. Introduction

A great deal of supervised machine learning research and industrial practice follows a pattern of trying a number of classification algorithms on a dataset, and then selecting and promoting the algorithm(s) that performed best according to cross-validation or a held-out test set. *How can one know whether the selected algorithm is a significant result?* To illustrate the problem, here are four scenarios where it may not be.

A. Specialized search space: Sometimes the researcher or practitioner conducting the experiments is a specialist in, say, neural networks or decision trees, and they experiment with variations in structure or learning parameters, without considering more diverse induction algorithms. Though their experiments may show that a particular improvement to an algorithm achieved the best results, it is possible that its performance is merely on par with trivial classification algorithms. For example, after a number of research papers published good results using sophisticated algorithms on the UCI classification datasets, Holte (1993) demonstrated competitive performance using a simple decision stump, which uses only one feature.

B. Overfitting one test set: If selecting the best algorithm via a single test set held-out during the experimentation phase, one runs the risk of overfitting the test set if there are many algorithms being evaluated and/or the test set is small (relative to the complexity of the true concept). The evaluation of data mining competitions in classification can be particularly prone to this pitfall, but unlike the first scenario, they do not suffer for method diversity.

C. Overfitting despite cross-validation: N-fold or leave-one-out cross-validation is regularly used to determine good estimates for the generalization performance of a classification method on future, unseen instances (Kohavi 1995). Nonetheless, if many methods are tested, cross-validation does not provide immunity from overfitting (e.g. Ng 1997).

D. Unlearnable tasks- concept drift, unpredictability: In situations where the training set may come from a very different distribution than the ultimate test set (e.g. if drawn from an earlier time period with substantial concept drift), or instead the training set features are not predictive of the class variable, then choosing the best method based on the training set will ultimately result in unpredictable testing performance. This may be viewed as a form of "overfitting" in that, if the chosen classifier matches the shape of the training set concept very precisely, then it will be sure not to match the deformed testing concept precisely. While most machine learning researchers avoid working with datasets of this nature, in real-world industrial settings, unlearnable classification tasks are regularly attempted and new techniques to identify these situations are called for. Such techniques could save others a great deal of wasted effort, much like NP-hard reductions do in theoretical computer science.

Overfitting is akin to the familiar risk in statistical testing when many hypotheses are tested: some may show statistical significance just by chance, without a repeatable, causal mechanism (e.g. Jensen & Cohen 2000). There are three central approaches for attempting to safeguard against this persistent issue:

1. **Data-oriented safeguards,** such as the bootstrap and cross-validation (Kohavi 1995).
2. **Representation-oriented safeguards,** penalizing complex models as in statistical learning theory and specifically support vector machines (Vapnik 1995).
3. **Process-oriented safeguards,** penalizing models the more one searches (Domingos 1999), or avoiding extensive search altogether.

None of these can prevent the possibility of overfitting, and each has its limitations. Ng (1997) and Domingos (1999) each lay out a case against methods 1 and 2. Domingos states in particular that representation-oriented penalties are "only appropriate when the simpler models are truly the more accurate ones, and there is mounting evidence that this is typically not the case." Finally, even with the process-oriented approach, there is certainly no guarantee that a greatly overfit model will not be hit upon early in the search process.

To these safeguards against overfitting, we contribute an analysis method that in some situations will reveal if overfitting could be a likely explanation for the apparent good performance of the best classifier obtained. The analysis can also expose the insignificance of one's best classifier in scenarios A and D above. These problems may be not be readily apparent, e.g. if benchmark results are not available for the classification task. Surprisingly, the problem can surface even when they are available, as we demonstrate for one of the three 2001 KDD Cup competition tasks (Cheng et al. 2001).

This work came about because of the unusual properties of the KDD Cup thrombin task. For the purpose of sharing lessons learned, we begin by presenting the clues that led us to the analysis, which is then demonstrated in Section 3 and described more abstractly in Section 4.

2. Clues from ROC Analysis

The 2001 KDD Cup thrombin task was a binary classification of a genomic dataset having 139,351 binary features. The training set was composed of 1909 cases, 42 being positive, yielding a class skew of 1:44. The contest clearly stated at the beginning that the test set would come from a somewhat different distribution—a set of 634 chemical compounds predicted by chemists to be active in binding (positive class) after they had analyzed the training set. One might suppose this would make the competition more difficult, but the effect was more dramatic than anyone had anticipated, as shown in the subsequent section.

The main clue we encountered that something was amiss with the test set was that our chosen classifier method exhibited a good ROC curve when cross-validating on the training set, but exhibited a nearly *flat* ROC curve on the official test set after the answer key was posted—i.e. as bad as random performance[1]. (See Figure 1.) Naturally, if a classifier has overfit the training set, one expects its ROC curve to degrade somewhat, but not to that extent. We did not attribute this to enormous overfitting because we had employed all three safeguard methods mentioned in the introduction: our search for a good classifier method (1) was guided by 20-fold cross-validation, (2) settled on using support vector machines, and (3) did not consider more than a dozen alternate methods. After this surprise, we tried a variety of other classifier methods and witnessed the same remarkable flattening of the ROC curve. This would suggest scenario D, a drifted task, or perhaps scenario A, since the winning entry was apparently able to learn the concept.

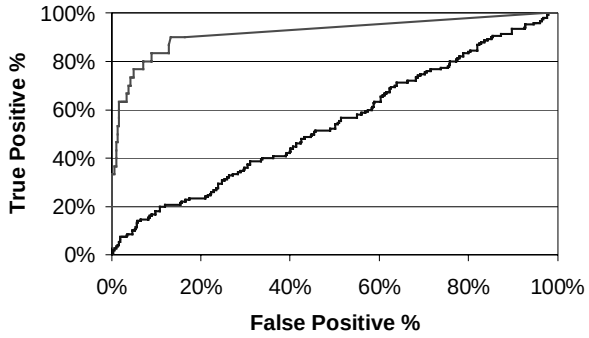

Figure 1. ROC curves for the SVM classifier.

3. A Randomized Distribution Analysis Applied to the Thrombin Task

There were 114 contest entrants for the thrombin task, which were to be judged by the average of their true positive rate and true negative rate. At the end of the competition, it had been a surprise to many that the best classifier for the task was a simple Bayes network involving only four binary features and arrived at without cross-validation. Further, it was nearly naïve Bayes—only one node had two parents. The final report showed a mediocre ROC curve, but it clearly had substantial lift over random. However, being chosen as the best of the contestants, it must show some degree of lift.

This observation opened a key question: how good would one expect the performance of the best classifier to be if the 114 entrants were all poor classifiers for the test set.

[1] Our first response was to manually validate that the software was working as intended, which is difficult for data mining on large sets—manually spot-checking format conversions, feature selection and other processes on the 139,000 binary features in the dataset.

To address this, we randomly generated thousands of naïve Bayes classifiers with four randomly selected features (for convenience, selecting from among 1200 predictive features we had pre-filtered to reduce the size of the dataset). We evaluated each of these according to the KDD Cup scoring metric using the labeled test data, and generated a cumulative distribution of the scores. We present the result in Figure 2, overlaid on top of the cumulative distribution of the actual contestant scores as provided in the final KDD Cup presentation (Haztis & Page 2001). The curves are a very close match, e.g. the median contestant performed about as well as the median random Bayes classifier. We then repeated this analysis, generating trivial classifiers that worked from a single, randomly chosen binary feature. This resulted in an S-curve with the same median score, but with a slightly steeper slope, as one might expect from the simpler decision function.

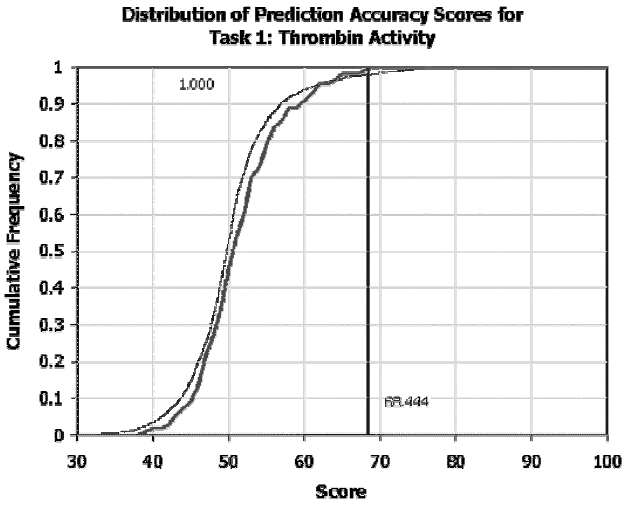

Figure 2. Cumulative distribution of contestant scores (thick curve) and random Bayes classifier scores (thin curve).

The vertical line marks the score of the winning entry, having a score of 68.444%. Of the 3500+ random pseudo-competitors we tested, 1.9% scored as well or better than the actual winning entry. So, given that the contest had 114 entrants, one would expect 2 entrants on average to score as well or better than the actual winner, if the classifiers had very little predictive power on the test set, e.g. concept drift. If this is the case, then sophisticated low-bias methods that carefully matched the training concept would actually be at a *dis*advantage to a variety of higher-bias methods that did not match the training concept precisely and therefore may stand a better chance of matching the drifted test concept.

(For contrast, we trained a SVM on half of the test set and tested on the other half, yielding a 78% performance score. Supposing for a moment this had been a contestant's entry, we compare it to the cumulative distribution of the randomized classifier scores and find this exceeds the performance of all of them. In this case we could have been reasonably confident in its validity.)

Considering the results of the analysis for the actual winner, we cannot safely reject the null hypothesis that it is merely the best of a set of poor classifiers, i.e. scenarios A or D. Given the diversity of methods used by the 114 competitors, it is reasonable to rule out scenario A. We infer that the thrombin test set was a nearly unlearnable task from the given training set using known classification algorithms, scenario D. The chemists, however, demonstrated the power of domain knowledge—although the training set only contained 42 positives (2.2%), the chemists were able to successfully identify 150 positives out of 634 attempts (24% positives, a factor of 10x more).

An alternate hypothesis for the consistently poor performance of the contest entries is that the high class skew defeats known induction algorithms. This can be ruled out by cross-validation on the training set alone. For example, the good quality ROC curve in Figure 1 was actually generated by training on a particularly difficult split of the training data: 25% of the positives and 75% of the negatives were used for training, leading to a much worse skew of ~1:120! Yet by the good ROC curve illustrated in the figure and other experiments we conducted, SVMs were clearly able to transfer the concept learned to this concocted testing set having a higher positive rate. Hence, the high degree of class skew would not account for the poor performance on the official test set.

4. General Randomized Distribution Analysis

Conventional wisdom about process-oriented safeguards suggests that one avoid testing many methods on the final testing set. However, once the best performing classifier(s) have been chosen, we propose comparing its performance against the distribution of performance scores that can be obtained on the test set by generating many simple classifiers. In contrast to Holte's (1993) work that generated a single trivial classifier for comparison, we generate many in order to consider their distribution. In fact, we generate so many that we fully expect some to overfit the testing set. These are not to be used in production for the ultimate classification task, but to provide a litmus test as to whether the best result obtained by traditional methods is in fact substantially better than simple methods (scenario A) and to check that the best result is not trivially explained by overfitting.

As with any data mining method, many variations can be constructed, but as an aid, we propose a specific method as a baseline: After the best classifier algorithm has been selected, generate a thousand simple naïve Bayes classifiers each with four randomly selected features. For each of these classifiers, measure its performance on the test set, using whatever scoring metric is appropriate for the project goal, e.g. accuracy, precision, recall, F-

measure, cost-sensitive evaluation, or area under the ROC curve. From these scores, generate a cumulative distribution curve, as in Figure 2, and compare to the distribution curve of the actual classifiers considered. (Alternately, traditional histograms may be more readable for some audiences.)

If the sophisticated algorithms are worthwhile for the dataset, their distribution curve will be shifted substantially to the right of the randomized classifier curve. On the other hand, if the curves coincide, they are either being defeated by an unlearnable task (scenario D) or are somehow inappropriate for the dataset (A).

Finally, consider also the number of different algorithms that were compared in selecting the best, or a rough estimate thereof. Multiply this by the percentage of randomized classifiers that exceeded the performance of the best algorithm. If this number is greater than one or two, consider the real possibility that the performance of the best classifier can be explained well by the null hypothesis. (Even if the number is less than one, this cannot rule out overfitting.)

Variations: Certainly one could use a different number of features as their baseline. For a text classification problem, 50 or 100 features may be appropriate. If the domain problem has only 5 features, it may be appropriate to use just one or two, and to consider another source of simple random variation in the classifiers (otherwise many of the 1000 classifiers would be identical). Another source of variation could be in discretization or the induction algorithm itself; however, it can be more difficult to generate 1000 variations.

5. Related Work

More generally, the randomized distribution analysis method forwarded here might be considered an instantiation of the *randomization method* discussed in Jensen & Cohen's (2000) paper on multiple comparisons. The generalized method constructs a collection of performance scores based on the null hypothesis (typically by randomly re-writing the labels of the training set), and rejects the null hypothesis only if the best performing system found is substantially better than the best of the random classifiers. Our method might then be viewed as an instantiation, where the null hypothesis is more elaborate: that the performance scores found in practice are no better than Naïve Bayes classifiers using four random features.

One disparity in this instantiation view is that the generalized method of Jensen & Cohen does not consider the number of different systems that have been considered in selecting the best. Suppose that the true model for a given dataset were actually Naïve Bayes using four of the features plus an additional fifty noise features. Their method would likely reject the correct classifier, whereas if the correct model were obtained in just one or a few trials due to the effectiveness of the induction algorithm, then the method in this paper would not consider it to be overfit. Additionally, their method only considers the performance of the best classifier, not the distribution curve of its competitors, which yields additional insight into their relative performance.

6. Discussion and Conclusion

Hindsight is 20-20—or at least 20-40 in data mining—and can lead to valuable lessons learned. The KDD Cup thrombin competition task is useful to sharpen us as a field to take greater heed of statistical significance when testing many alternatives, and it yielded the analysis method above for aiding us in knowing when our best result is actually mediocre. In short, it asks how reasonably can we reject the null hypothesis that the best obtained classifier's performance is what one would expect from a large sample of trivial classifiers?

Data mining competitions in classification are particularly prone to overfitting, since the winner is selected over many competitors based on a single test set—in some sense, the promoted result has the unfair advantage of learning on the test set, and likely does not have the best generalization error. We propose that future data mining competitions apply the analysis technique given in this paper in order to provide a baseline. With this, we can see what fraction of the contestants performed much better than trivial classifiers would. We also propose that some future data mining competition score the results based on ROC area analysis, partly to increase the awareness of this powerful technique, which yields much more information than a single accuracy score or a confusion matrix can. We hypothesize that the difficulty with the test set would have been more widely discovered if ROC analysis were more commonly used.

Furthermore, in this paper we addressed the situation where a classification task is very difficult or nearly unlearnable, and yet by trying many different methods, one can find a classifier that appears to have good performance. Real-world data mining efforts in industrial settings often have this property, and may go undiscovered. This calls for new research methods to detect and avoid spending effort on them, e.g. see Zhang et al. (2000) for a result on unlearnable regression tasks.

Finally, we must acknowledge the importance of leveraging domain knowledge. Much of the supervised machine learning research avoids using domain knowledge, because of the effort to obtain and integrate it and because one's research results are then open to the criticism that the discrimination power of the classifier is due largely to applying "just the right" domain knowledge. Nonetheless, significant advances in AI have come about through incorporating domain knowledge, especially in selecting the best problem representation.

Acknowledgements

We appreciate DuPont Pharmaceuticals Research Laboratories for making the thrombin dataset available and the organizers of the 2001 KDD Cup for arranging the competition, which regularly leads to valuable lessons learned in the data mining community. We would also like to thank Jaap Suermondt, Tom Fawcett, and Umesh Dayal, without whose encouragement this paper would not have been. Tom's critique helped make the paper much more understandable. Finally, we thank the anonymous reviewers for their helpful suggestions and the organizers of the Data Mining Lessons Learned workshop for providing an appropriate forum for the type of result presented in this paper.

References

Cheng, J., Hatzis, C., Hayashi, H., Krogel, M.-A., Morishita, S., Page, D. & Sese, J. (2001). KDD Cup 2001 Report. *ACM SIGKDD Explorations*, 3(2), pp.47-64.

Domingos, P. (1999). Process-Oriented Estimation of Generalization Error. *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pp.714-719, Stockholm, Sweden.

Haztis, C. & Page, D. (2001). KDD Cup 2001 Summary Presentation. August 26, 2001. Available at http://www.cs.wisc.edu/~dpage/kddcup2001

Holte, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning 11: 63-91.

Jensen, D. & Cohen, P. (2000). Multiple Comparisons in Induction Algorithms. Machine Learning 38(3):309-338

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1137-1145, Montréal, Canada.

Ng, A. Y. (1997). Preventing "overfitting" of cross-validation data. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pp.245-253, Nashville, TN.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Zhang, B., Elkan, C., Dayal, U., & Hsu, M. (2000). Model-Independent Measure of Regression Difficulty. Hewlett-Packard Technical Report HPL-2000-5.