Classifier Loss under Metric Uncertainty

David B. Skalak
Highgate Predictions, LLC
Ithaca, NY 14850 USA
skalak@cs.cornell.edu

Alexandru Niculescu-Mizil
Department of Computer Science
Cornell University
Ithaca, NY 14853 USA
alexn@cs.cornell.edu

Rich Caruana
Department of Computer Science
Cornell University
Ithaca, NY 14853 USA
caruana@cs.cornell.edu

Abstract

Classifiers that are deployed in the field can be used and evaluated in ways that were not anticipated when the model was trained. The ultimate evaluation metric may not have been known to the modeler at training time, additional performance criteria may have been added, the evaluation metric may have changed over time, or the real-world evaluation procedure may have been impossible to simulate. But unforeseen ways of measuring model utility can degrade performance. Our objective is to provide experimental support for modelers who face potential "cross-metric" performance deterioration. First, to identify model-selection metrics that lead to stronger cross-metric performance, we characterize the expected loss where the selection metric is held fixed and the evaluation metric is varied. Second, we show that the number of data points evaluated by a selection metric has a substantial effect on the optimal evaluation. In trying to address both these issues, we hypothesize that whether classifiers are calibrated to output probabilities may influence these issues. In our consideration of the role of calibration, we show that our experiments demonstrate that cross-entropy is the highest-performing selection metric where little data is available for selection. With these experiments, modelers may be in a better position to choose selection metrics that are robust where it is uncertain what evaluation metric will be applied.

Introduction

Most machine learning research on classification has assumed that it is best to train and select a classifier according to the metric upon which it ultimately will be evaluated. However, this characterization makes several assumptions that we question here. What if we don't know the metric upon which the classifier will be judged? What if the classification objective is not optimal performance, but simply robust performance across several metrics? Does it make any difference how much data is available on which to base model performance estimates? What if we want at least to avoid the worst-performing selection metrics?

In this paper we give experimental results to begin to answer questions like the ones we have just posed. The results show that the choice of selection metric depends to a large degree on how much data is available to measure performance and depends also on whether the underlying models produce accurate probabilities.

It is not so far-fetched that we may not have as much knowledge of — and access to — the ultimate evaluation metric as is usually assumed. In some situations a modeler may have the discretion to build models that optimize one of several metrics but not have access to an classification algorithm that directly optimizes the evaluation metric. For example, the modeler may decide between optimizing cross-entropy or root-mean-squared error through the choice of model class and training algorithm. But if these models are evaluated with respect to the F-score metric, it would be important to compare expected performance losses in going from cross-entropy to F-score and from root-mean-squared error to F-score. These considerations arise in natural language processing (NLP) tasks, such as noun phrase coreference resolution, where classification models may be built to maximize accuracy, but where F-score or average precision provides the ultimate measure of success (Munson, Cardie, & Caruana 2005). In fact, NLP tasks are often evaluated on multiple reporting metrics, compounding the cross-metric problem.

The substantial data preprocessing that is often required for NLP systems often places NLP classifiers in a pipeline where they are judged according to the performance they enable in downstream modules that receive the class predictions. Embedded classifiers may be subjected to evaluation(s) that cannot easily be tested and that may change according to evolving criteria of the entire system.

A marketing group in a large organization may request a model that maximizes response lift at 10% of the universe of customers. After the model has been built, the marketing budget for the campaign is cut, but the marketing group has the campaign ready to roll out and so not have the time to commission another model. In that case the database marketing group may decide to contact only 5% of the customers. The model that optimized response at the 10% level will now be judged in the field according to a different criterion: response from 5% of the customers. (Alternatively, the marketing group may not even specify its performance criterion, but may request a model that "simply" yields optimal profits, accuracy, and lift.) What model should be selected to be robust to changes such as these?

Thus real-world considerations may make evaluation

more complicated than might be generally assumed. Performance metrics may change over time, may not be known, may be difficult to simulate, or may be numerous. In this paper we examine uncertain metric evaluation by providing experimental answers to two questions:

1. What selection metrics yield the highest performance across commonly applied evaluation metrics?
2. What is the effect of the number of data points available for making model selection judgments where the ultimate evaluation metric may be unknown?

Related Research

As part of an extensive set of experiments, Huang and Ling defined a model selection ability measure called MSA to reflect the relative abilities of eight metrics to optimize one of three target metrics: accuracy, area under the ROC curve ("AUC") and lift (Huang & Ling 2006). Given one of these three "goal" metrics, MSA measures the probability that one of the eight metrics correctly identifies which member of all pairs of models will be better on the goal metric. While this is an attractive summary approach, our experiments hew more closely to how we see model selection done in practice. Our experiments measure how one metric's *best* performing models perform when measured by a second metric. Since practitioners tend to focus on superior models only, our methodology also reflects that bias. Our empirical study below also evaluates all our metrics as reporting methods rather than limiting the study to a subset of three goal metrics. The roles of probability calibration and data set size in reducing performance loss are also studied additionally here.

Several related efforts to develop algorithms to handle multiple performance criteria have also been made (Soares, Costa, & Brazdil 2000; Nakhaeizadeh & Schnabl 1997; Spiliopoulou *et al.* 1998). Additionally, Ting and Zheng (1998) have provided an approach to deal with changes in costs over time.

As part of a statistical study of AUC, Rosset (2004) showed empirically that, even where the goal is to maximize accuracy, optimizing AUC can be a superior strategy for Naive Bayes and k-nearest neighbor classifiers. Joachims (2005) has extended support vector methodology to optimize directly non-linear performance measures, measures that cannot be decomposed into measures over individual examples, and any measure that can be derived from a contingency table. Cortes and Mohri (2004) give a statistical analysis of accuracy and AUC and show that classifiers with the same accuracy can yield different AUC values when the accuracy is low.

Experimental Design

Performance Metrics

The performance metrics we study are accuracy (ACC), lift at the 25th percentile (LFT), F-score (FSC), area under the ROC curve (ROC), average precision (APR), precision-recall break-even point (BEP), root-mean squared error (RMS), and mean cross-entropy (MXE). We also synthesize a hybrid metric that is defined as the equally-weighted mean performance under RMS, ROC and ACC (called "ALL"). We follow the definitions of these performance metrics found in Caruana and Niculescu-Mizil (2004), since they are implemented in the PERF code that was made available by Caruana in connection with the KDD Cup 2004.

We have also adopted the same conventions as to the normalization of classifier performance with respect to various metrics. Unfortunately, normalization is necessary in order to compare directly metrics with different measurement scales. Metrics have been normalized to values in $[0, 1]$ where 0 represents the baseline performance of classifying all instances with the most frequent class in the data, and 1 corresponds to the best performance of any model developed in our lab on that data [1].

Problems

Eleven binary classification problems were used in these experiments. ADULT, COV_TYPE and LETTER are from the UCI Repository (Blake & Merz 1998). COV_TYPE has been converted to a binary problem by treating the largest class as the positive and the rest as negative. We converted LETTER to boolean in two ways. LETTER.p1 treats "O" as positive and the remaining 25 letters as negative, yielding a very unbalanced problem. LETTER.p2 uses letters A-M as positives and the rest as negatives, yielding a well balanced problem. HS is the IndianPine92 data set (Gualtieri *et al.* 1999) where the difficult class Soybean-mintill is the positive class. SLAC is a problem from the Stanford Linear Accelerator. MEDIS and MG are medical data sets. COD, BACT, and CALHOUS are three of the datasets used in (Perlich, Provost, & Simonoff 2003). ADULT, COD, and BACT contain nominal attributes. For ANNs, SVMs, KNNs, and LOGREG we transform nominal attributes to boolean (one boolean per value).

Model Types

The 10 model type that we used in this experiment were: back-propagation neural networks, bagging of decision trees, boosting of decision trees, k-nearest neighbor, logistic regression, Naive Bayes, random forests, decision trees, bagged decision stumps and support vector machines. We create a library of approximately 2,000 models trained on training sets of size 4,000. We train each of these models on each of the 11 problems to yield approximately 22,000 models. Each decision tree, bagged tree, boosted tree, boosted stump, random forest and Naives Bayes model is trained twice, once with transformed attributes and once with the original ones. The models are all as described in (Caruana & Niculescu-Mizil 2006).

The output of such learning methods as boosted decision trees, boosted decision stumps, SVMs and Naive Bayes cannot be interpreted as well-calibrated posterior probabilities (Niculescu-Mizil & Caruana 2005). This has a negative impact on the metrics that interpret predictions as probabilities: RMS, MXE and ALL (which invokes RMS). To ad-

[1] The performance upper bounds are available to interested researchers.

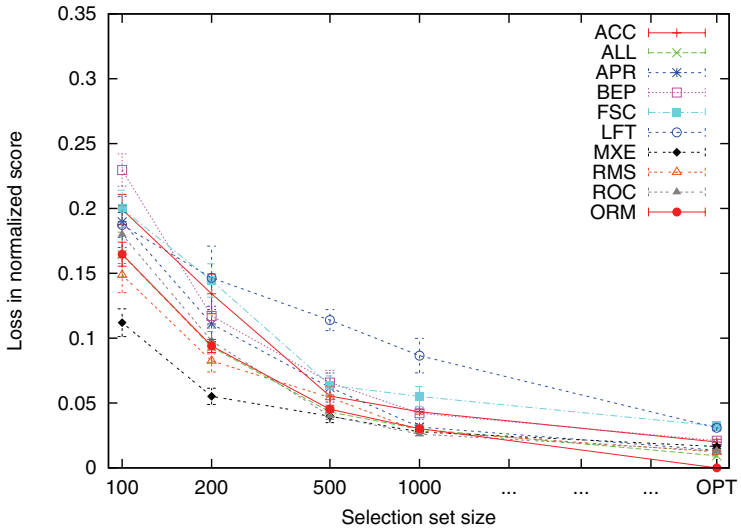

Figure 1: Average loss across all nine reporting metrics.

dress this problem, we use post-training calibration to transform the predictions of all the methods into well-calibrated probabilities. In this paper calibration is done via Platt scaling (Platt 1999). To fit the calibrated models we use a set of 1000 points that are reserved solely for calibration (i.e. they are not part of the training, validation or final test set). While in practice one would use the same set of points both for calibration and for model selection, here we choose to use separate sets in order to separate the effects of calibration from the effects of model selection on performance. The effect of calibration will be discussed later in the paper.

The Effect of Limited Sample Size on Selection Metric Choice

In this section we discuss the effect of small data sample size on the decision as to which selection metrics to use. Our primary objective in this section is to quantify the loss in selecting on one metric but reporting on another. To obtain the results in this section, we use the following methodology. For each problem, we train each of the approximately 2,000 models on a 4,000-point training set. All the trained models are then evaluated on a validation (selection) set, and the model with the best performance on the selection metric is found. Finally, we report the evaluation (reporting) metric performance of the best model on a final independent test set. To ensure that the results are not dependent on the particular train/validation/test set split, we repeat the experiment five times and report the average performance over the five randomized trials.

To investigate how the size of the selection set affects the performance of model selection for different selection metrics, we consider selection sets of 100, 200, 500 and 1000 points. For comparison we also show results for "optimal" selection, where the final test set is used as the selection set.

We use the following experimental procedure. We are given a problem, a selection metric, s, and a reporting metric, r. We choose from our library the classifier C_s that has the highest normalized score under the selection metric s. We then measure the score of that classifier C_s under the reporting metric r. Call that score $r(C_s)$.

Next we identify the classifier C^* that has the highest performance on the reporting metric. Call that score $r(C^*)$. The difference $r(C^*) - r(C_s)$ is the loss we report. The selection of C_s is done on a validation set and the reporting metric performance of both classifiers is computed on an independent test set.

Figure 1 shows the loss in performance due to model selection for nine selection metrics averaged across the nine reporting metrics. The tenth line, ORM (Optimize to the Right Metric), shows the loss of always selecting using the evaluation metric (i.e. select using ACC when the evaluation metric is ACC, ROC when the evaluation metric is ROC, etc.). On the X axis we vary the size of the selection set on a log scale. The right-most point on the graph, labeled OPT, shows the loss when selection is done "optimally" (by cheating) using the final test set. This represents the best achievable performance for any selection metric, and can be viewed as a bias, or mismatch, between the selection metric and the evaluation metric.[2]

The most striking result is the good performance of selecting on mean cross-entropy (MXE) for small sizes of the selection set. When the selection set has only 100 or 200 points, using cross-entropy as the selection metric incurs the lowest loss. In fact, at 100 and 200 points, selecting on MXE has the lowest loss for every individual reporting metric, not only on average! This may be a surprising result in that it undermines the common belief that it is always better to optimize to the metric on which the classifier will be evaluated.

We propose two hypotheses that would account for the superior performance of MXE for small data sets, but we do

[2]Of course, this bias/mismatch depends on the underlying set of classifiers available for selection.

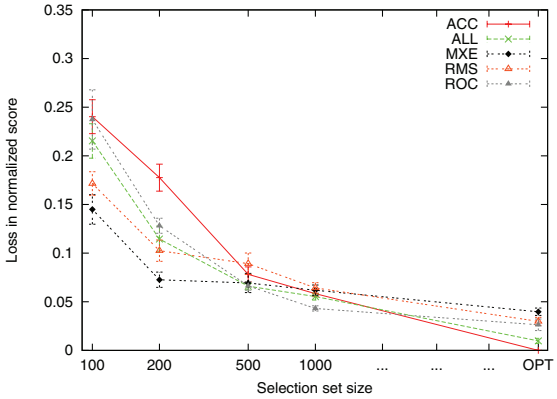

Figure 2: Loss when reporting on ACC.

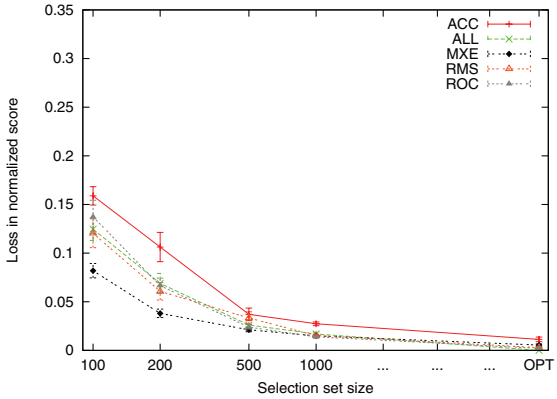

Figure 3: Loss when reporting on ALL.

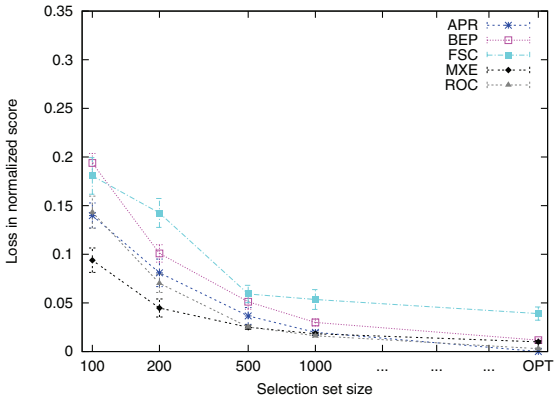

Figure 4: Loss when reporting on APR.

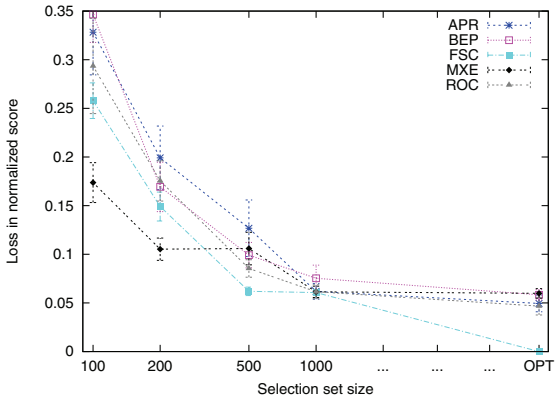

Figure 5: Loss when reporting on FSC.

not yet have support for these possible explanations. MXE provides the maximum likelihood probability estimation of the binary targets. Under this hypothesis, MXE reflects the "correct" prior for target values as a binomial distribution (Mitchell 1997). Priors are particularly important where data are scarce. The second hypothesis recognizes that (of the metrics we consider) MXE assesses the largest penalty for large errors, which may be desirable where not much data is available.

For larger selection sets, MXE continues to be competitive, but ROC and ALL catch up when the selection set has 500 points. At 1000 points all metrics except BEP, ACC, FSC, and LFT have similar performance (on average across reporting metrics). This result suggests that, when the evaluation metric is uncertain, cross-entropy should be used as a selection metric, especially when validation data is scarce. When the validation set is larger, ROC, RMS and ALL also are robust selection metrics. LFT and FSC seem to be the least robust metrics; BEP and ACC were slightly better than LFT and FSC.

Figure 2 shows the performance for a few selection metrics when ACC is the evaluation metric. The figure shows ROC is superior as a selection metric to ACC even when the evaluation metric is ACC. ROC-based selection yields lower loss across all selection set sizes (except of course for OPT where ACC has zero loss by definition). This confirms the observation made originally by Rosset 2004. Although at low selection set sizes MXE has the best performance (followed by RMS), looking at the OPT point, we see that MXE actually has the largest bias (followed by RMS). Of all metrics ALL has the smallest bias.

Figure 3 shows the case when the performance is evaluated using a combination of multiple metrics. For the figure, the reporting metric is ALL which is an equally weighed average of ACC, RMS and ROC. Selecting on the more robust RMS or ROC metrics performs as well as selecting on the evaluation metric ALL. This is not the case for ACC, which is a less robust metric. For small validation sets, cross-entropy is again the best selection metric.

In the Information Retrieval (IR) community, APR is often preferred to ROC as a ranking evaluation metric because it is more sensitive to the high end of the ranking and less sensitive to the low end. Figure 4 shows the loss in normalized score when the evaluation metric is APR. Besides APR and ROC, we also show the selection performance of MXE and two other IR metrics: BEP and FSC. The results suggest that selection based on ROC performs the same, or slightly better than selecting on APR directly. In fact ROC has a very low bias relative to APR, as shown by the OPT point in the graph. The other two IR metrics have lower performance,

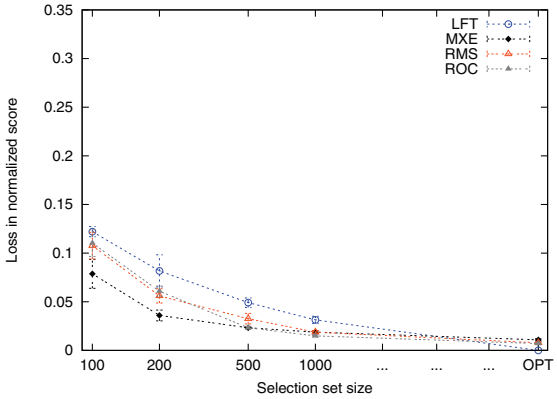

Figure 6: Loss when reporting on LFT.

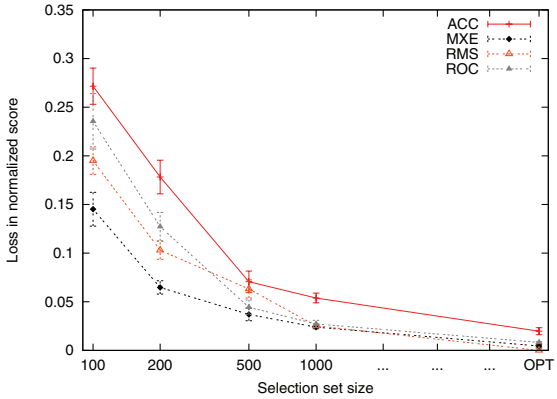

Figure 7: Loss when reporting on RMS.

with FSC incurring a significantly higher loss.

Figure 5 depicts the loss in normalized score when using FSC as an evaluation metric. This figure may also be of interest to IR practitioners, since FSC is often relied upon in that field. The figure shows that, except for small validation set sizes, if FSC is the metric of interest, then FSC should also be used as a selection metric. For small validation sets, cross-entropy again provides significantly lower loss. One other interesting observation is the large mismatch between FSC and the other metrics (the OPT point in the graph). This mismatch is one reason why, given enough validation data, FSC is the preferred selection metric when one is interested in optimizing FSC.

One other interesting case is shown in Figure 6 for lift as the evaluation metric. The figure shows that even if one is interested in lift, one should not select based on lift. Cross-entropy, squared error and ROC all lead to selecting better models.

Interestingly, even when squared error is the evaluation metric (Figure 7), it is still better to select using cross-entropy when the validation set is small. The result is somewhat surprising given that squared error and cross-entropy are similar metrics in that they both interpret the predictions as conditional probabilities. This might suggest that the good performance of cross-entropy when data is scarce is due to the high penalty it puts on the cases where the predicted probability of the true class is very low.

The Effect of Model Probability Calibration on Selection Metric Choice

In this section we investigate how cross-metric optimization performance is affected by the presence of poorly calibrated models such as boosted trees, boosted stumps, SVMs and Naive Bayes. To this end, we repeat the experiments in the previous section, but using the original uncalibrated models instead of the Platt-calibrated ones.

As expected, having a mix of well-calibrated and poorly calibrated models hurts cross-metric optimization. The effect of poorly calibrated models is two-fold. On one hand, when selecting on a metric such as ROC, APR or ACC that does not interpret predictions as probabilities, and evaluating on a metric such as RMS, MXE or ALL that is sensitive to probability calibration, it may happen that the selected model, while performing well on the "non-probability" measures, is poorly calibrated, thus incurring a high loss on the "probability" measures.

This effect can be seen clearly in Figure 8. The figure shows the loss in normalized score when the reporting metric is MXE, and the selection metric is MXE, ROC or ACC. For each selection metric, two lines are shown: one represents the performance when selecting from uncalibrated models, and the other shows the performance when selecting from Platt-calibrated models. Selecting from uncalibrated models, using either ROC or ACC as a selection metric (the top two lines) incurs a very large loss in performance (note the scale). In fact, quite often, the MXE performance of the selected models is worse than that of the baseline model (the model that predicts, for each instance, the ratio of the positive examples to all examples in the training set). Calibrated models eliminate this problem, driving down the loss.

On the other hand, when selecting on one of the "probability" measures (RMS, MXE or ALL), the poorly calibrated methods will not be selected because of their low performance on such metrics. Some of the poorly calibrated models, however, do perform very well on "non-probability" measures such as ROC, APR or ACC. This leads to increased loss when selecting on probability measures and evaluating on non-probability ones because, in a sense, selection is denied access to some of the best models.

Figure 9 shows the loss in normalized score when the reporting metric is APR, and the selection metric is MXE, ROC or ACC. Looking at MXE as a selection metric we see that, as expected, the loss from model selection is higher when using calibrated models than when using uncalibrated ones. Since calibration does not affect ROC or APR, selecting on ROC and evaluating on APR yields the same results regardless of whether the models were calibrated. This is not true when selecting on ACC because calibration can affect threshold metrics by effectively changing the threshold.

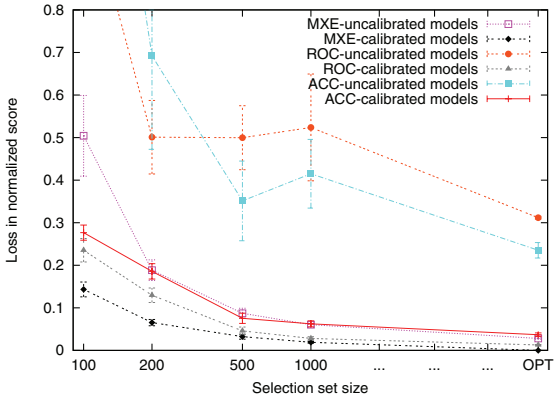

Figure 8: Loss when reporting on MXE.

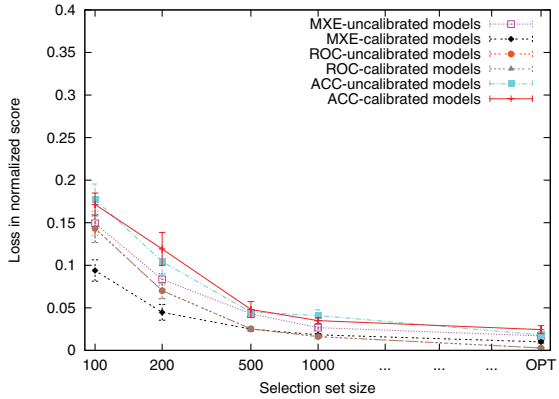

Figure 9: Loss when reporting on APR.

Conclusion

Our experiments have shown that when only a small amount of data is available, the cross-entropy selection metric yields the strongest cross-metric performance. The experiments have also shown that calibration can affect the performance of selection metrics in general, and of cross-entropy in particular. In general, MXE and ROC performed strongly as selection metrics and FSC and LFT performed poorly. The next step in our research is to go beyond the empirical results presented in this paper and try to create a formal decomposition of cross-metric loss.

Acknowledgments: This work was supported by NSF Award 0412930

References

Blake, C., and Merz, C. 1998. UCI repository of machine learning databases.

Caruana, R., and Niculescu-Mizil, A. 2004. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 69–78.

Caruana, R., and Niculescu-Mizil, A. 2006. An empirical comparison of supervised learning algorithms. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 161–168.

Cortes, C., and Mohri, M. 2004. Auc optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems 16*.

Gualtieri, A.; Chettri, S. R.; Cromp, R.; and Johnson, L. 1999. Support vector machine classifiers as applied to aviris data. In *Proc. Eighth JPL Airborne Geoscience Workshop*.

Huang, J., and Ling, C. X. 2006. Evaluating model selection abilities of performance measures. In *Evaluation Methods for Machine Learning, Papers from the AAAI workshop, Technical Report WS-06-06*, 12–17. AAAI.

Joachims, T. 2005. A support vector method for multivariate performance measures. In *Proceedings of the International Conference on Machine Learning (ICML), 2005*, 377–384.

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill.

Munson, A.; Cardie, C.; and Caruana, R. 2005. Optimizing to arbitrary NLP metrics using ensemble selection. In *Proc. of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 539–546.

Nakhaeizadeh, C., and Schnabl, A. 1997. Development of multi-criteria metrics for evaluation of data mining algorithms. In *Proceedings of the 3rd International Conference on Knowledge Discovery in Databases*.

Niculescu-Mizil, A., and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *Proc. 22nd International Conference on Machine Learning (ICML'05)*.

Perlich, C.; Provost, F.; and Simonoff, J. S. 2003. Tree induction vs. logistic regression: a learning-curve analysis. *J. Mach. Learn. Res.* 4:211–255.

Platt, J. C. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 61–74. MIT Press.

Rosset, S. 2004. Model selection via the auc. In *ICML '04: Proceedings of the Twenty-first International Conference on Machine Learning*.

Soares, C.; Costa, J.; and Brazdil, P. 2000. A simple and intuitive measure for multicriteria evaluation of classification algorithms. In *Proceedings of the Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*. Barcelona, Spain: ECML 2000.

Spiliopoulou, M.; Kalousis, A.; Faulstich, L. C.; and Theoharis, T. 1998. NOEMON: An intelligent assistant for classifier selection. In *FGML98*, number 11 in 98, 90–97. Dept. of Computer Science, TU Berlin.

Ting, K. M., and Zheng, Z. 1998. Boosting trees for cost-sensitive classifications. In *ECML*, 190–195.