Nicolas Lachiche
Cèsar Ferri
Sofus A. Macskassy (Eds.)

ROC Analysis in Machine Learning

3rd International Workshop, ROCML-2006
Pittsburgh, USA, June 29, 2006
Proceedings

Workshop associated to ICML-2006,
The 23rd International Conference on Machine Learning

Volume Editors

Nicolas Lachiche, University of Strasbourg, France.

Cèsar Ferri, Technical University of Valencia, Spain.

Sofus A. Macskassy, Fetch Technologies, Inc., USA.

Proceedings of the 3rd Intl. Workshop on ROC Analysis in Machine Learning, ROCML-2006.
Pittsburgh, USA, 29 June, 2006.

Preface

This volume contains the proceedings of the Third International Workshop on ROC Analysis in Machine Learning, ROCML-2006. The workshop was held as part of the 23rd International Conference on Machine Learning (ICML-2006) in Pittsburgh (USA) on June 29, 2006.

Receiver Operating Characteristic Analysis (ROC Analysis) is related in a direct and natural way to cost/benefit analysis of diagnostic decision making. Widely used in medicine for many decades, it has been introduced relatively recently in machine learning. In this context, ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. Furthermore, the Area Under the ROC Curve (AUC) has been shown to be a better evaluation measure than accuracy in contexts with variable misclassification costs and/or imbalanced datasets. AUC is also the standard measure when using classifiers to rank examples, and, hence, is used in applications where ranking is crucial, such as campaign design, model combination, collaboration strategies, and co-learning.

Nevertheless, there are many open questions and some limitations that hamper a broader use and applicability of ROC analysis. Its use in data mining and machine learning is still below its full potential. An important limitation of ROC analysis, despite some recent progress, is its possible but difficult extension for more than two classes.

This workshop follows up a first workshop (ROCAI-2004) held within ECAI-2004 and a second workshop (ROCML-2005) held within ICML-2005. This third workshop is intended to investigate on the hot topics identified during the two previous workshops (e.g. multiclass extension, statistical analysis, alternative approaches), on the one hand, and to encourage cross-fertilisation with ROC practitioners in medicine, on the other hand, thanks to an invited medical expert.

We would like to thank everyone who contributed to make this workshop possible. First of all, we thank all the authors who submitted papers to ROCML-2006. Each of these was reviewed by two or more members from the Program Committee, who finally accepted nine papers (eight research papers and one research note). In this regard, we are grateful to the Program Committee and the additional reviewers for their excellent job. We wish to express our gratitude to our invited speaker, Dr. Darrin C. Edwards from Department of Radiology, University of Chicago, who presented the state-of-the-art of ROC analysis in radiology. Moreover, his research group provided a three-class medical dataset to support exchanges between medical experts and participants. Finally, we have to express our gratitude to the ICML-2006 organization for the facilities provided.

N. Lachiche, C. Ferri, and S. A. Macskassy.

Program Committee

- Stephan Dreiseitl FHS Hagenberg, Austria.
- Richard M. Everson, University of Exeter, UK.
- Cèsar Ferri, Technical University of Valencia, Spain.
- Jonathan E. Fieldsend, University of Exeter, UK.
- Peter Flach, University of Bristol, UK.
- José Hernández-Orallo, Technical University of Valencia, Spain.
- Rob Holte, University of Alberta, Canada.
- Nicolas Lachiche, University of Strasbourg, France.
- Michele Sebag, LRI, CNRS-Université de Paris Sud , France.
- Sofus A. Macskassy, Fetch Technologies, Inc., USA.
- Alain Rakotomamonjy, Insa de Rouen, France.
- Francesco Tortorella, University of Cassino, Italy.

Organising Committee

- Cèsar Ferri, Technical University of Valencia, Spain.
- Nicolas Lachiche, University of Strasbourg, France.
- Sofus A. Macskassy, Fetch Technologies, Inc., USA.

Table of Contents

Resampling Methods for the Area Under the ROC Curve

Andriy I. Bandos ANB61@PITT.EDU
Howard E. Rockette HERBST@PITT.EDU
Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh, Pittsburgh, PA, U.S.A

David Gur GURD@UPMC.EDU
Department of Radiology, School of Medicine, University of Pittsburgh, Pittsburgh, PA, U.S.A

Abstract

Receiver Operating Characteristic (ROC) analysis is a common tool for assessing the performance of various classification tools including biological markers, diagnostic tests, technologies or practices and statistical models. ROC analysis gained popularity in many fields including diagnostic medicine, quality control, human perception studies and machine learning. The area under the ROC curve (AUC) is widely used for assessing the discriminative ability of a single classification method, for comparing performances of several procedures and as an objective quantity in the construction of classification systems. Resampling methods such as bootstrap, jackknife and permutations are often used for statistical inferences about AUC and related indices when the alternative approaches are questionable, difficult to implement or simply unavailable. Except for the simple versions of the jackknife, these methods are often implemented approximately, i.e. based on the random set of resamples, and, hence, result in an additional sampling error while often remaining computationally burdensome. As demonstrated in our recent publications, in the case of the nonparametric estimator of the AUC these difficulties can sometimes be circumvented by the availability of closed-form solutions for the ideal (exact) quantities. Using these exact solutions we discuss the relative merits of the jackknife, permutation test and bootstrap in application to a single AUC or difference between two correlated AUCs.

1. Introduction

Many different fields are faced with the practical problems of detection of a specific condition or classification of findings – the tasks that can be collectively described as classification of the subjects into categories. The system that defines the specific manner of a classification process is termed differently depending on the field and task at hand (e.g. diagnostic marker, diagnostic system, technology or practice, predictive model, etc.). In this manuscript we will use the terms classification system or tool to refer to such a system regardless of the field and the task.

Since the ultimate goal is an application of the classification system to subjects from the general "target" population the performance in the target population is one of the important characteristics of the classification system. Since in practice it is usually impossible to apply the classification system to the whole population it is applied to a sample of subjects from the target population. Based on such a sample the performance of the classification system in the target population can be assessed using statistical methods.

For classification problems, performance is typically assessed in terms of the multiple probabilities of the possible outputs conditional on the true status of subjects (for binary classification - sensitivity or true positive rate and specificity or false positive rate). Multiple probabilities are considered in order to avoid specification of the relative costs and conditioning on the true class is performed in order to eliminate a dependence on the class distribution within the sample.

Some classification systems can be supervised to produce different classification rules. Most commonly such classification systems produce a quantitative output (e.g. probability of belonging to a specific class) and a decision rule is determined by a specific threshold. Another example is an unlabelled classification tree where a decision rule is determined by a specific labeling of the terminal nodes (Ferri, Flach, & Hernandez-Orallo 2002). For such classification systems an operating mode (threshold, labeling etc.) is often chosen considering the class distribution in the target population and relative cost and benefits of the specific decisions. Because of that, when assessing the performance of the classification system using a sample from the population it is often

Key words: ROC, AUC, bootstrap, permutations, jackknife, exact variances

desirable to have a performance measure that is also independent from a specific operating mode.

For binary classification tasks (subjects are classified into the two classes), conventional ROC analysis provides a tool to assess the performance of a classification system simultaneously for all operating thresholds and independently of the class distribution in the sample and costs and benefits of various decisions. The conventional ROC analysis originated in signal detection theory and presently is a widely used tool for the evaluation of classification systems (Swets & Picket, 1982; Zhou, Obuchowski and McClish, 2002; Pepe, 2003). The keystone of ROC analysis is the ROC curve which is defined as a plot of sensitivity (true positive rate) versus 1-specificity (false positive rate) computed at different possible operating modes. It illustrates the tradeoff between the two classification rates and enables the assessment of the inherent ability of a classification system to discriminate between subjects from different classes (e.g. with and without a specific disease or abnormality). Another beneficial feature of the ROC curve is its invariance to monotone transformations of the data. For example, the ROC curve corresponding to a pair of normal distributions representing classification scores (binormal ROC) is the same as the ROC curve for any pair of distribution that is monotonically transformable to the original pair.

Because its construction requires the probabilities of various classifications conditional on the true class of the subjects, a conventional Receiver Operating Characteristic (ROC) analysis is only applicable in situations where the true class is known for all subjects. On the other hand this feature enables ROC analysis to be used for studies where a fixed number of subjects have been selected from each class separately as opposed to taking a sample from the total population. Selection of subjects from each class separately eliminates problem resulting from low frequency of a specific class (e.g. low prevalence of a specific disease) and permits more efficient study design in regard to statistical considerations.

Although the ROC curve is quite a comprehensive measure of performance, because it is a whole curve there is often a desire to obtain a simpler summary index. Thus, for summarizing the performance of a classification system, more simple indices such as the area under the ROC curve (AUC), or partial AUC are typically used. The area under the ROC curve (AUC) is a widespread measure of the overall diagnostic performance and has a practically relevant interpretation as the probability of a correct discrimination in a pair of randomly selected representatives of each class (Bamber, 1975; Hanley & McNeil, 1982). In the presence of a continuous classification score the AUC is the probability of stochastic dominance of an “abnormal’ class versus “normal” class, where “abnormal” class is expected to have greater scores on average.

The AUC is used for assessing the performance of a single classification system, comparing several systems and as an objective quantity for constructing a classifier (Verrelst et al 1998; Pepe & Tompson 2000; Ferri, Flach, & Hernandez-Orallo 2002; Yan et al 2003; Pepe, 2006).

An assessment of the performance of a single or a comparison of several classification systems is often initiated by computing the AUCs from the sample selected from the target population (“sample AUC”). Since the performances in the sample might differ from that in the target population, inferences about the population performance should incorporate assessment of the sample-related uncertainty. A common approach to evaluate the sample-related uncertainty is to estimate the variance of the AUC estimator. The variance estimator can than be used to place confidence intervals, test hypothesis or plan future studies.

When comparing two classification systems, an attempt is often made to control for variability by design. Namely, the data is collected under a paired design where the same set of subjects is evaluated under different classification systems, reducing the effect of heterogeneity of the samples of subjects. On the one hand the paired design leads to correlated estimators of the AUCs, requiring specific analytic methods, but on the other hand, similar to the paired t-test, because of the completely paired structure the variance for the difference of the correlated AUCs can be obtained from the variance of a single AUC by direct substitution.

Many nonparametric estimators of the variance of a single AUC and the difference between two correlated AUCs have been proposed. The methods proposed by Bamber in 1975 (based on formula from Noether 1967) and Wieand, Gail & Hanley (1983) provide unbiased estimators of the variance of a single AUC and the covariance of two correlated AUCs correspondingly. Hence, these estimators are useful for assessing the magnitude of the variability but may provide no advantages in hypothesis testing. The estimator proposed by Hanley & McNeil (1982) explicitly depends only upon the AUC and sample size and thus enables simple estimation of the sample size for a planned study. However, this estimator is known to underestimate or overestimate variance depending on the underlying parameters (Obuchowski 1994; Hanley & Hajian-Tilaki 1997) and thus is not optimal for either variance estimation or hypothesis testing (an improved estimator of the same kind was proposed by Obuchowski in 1994). Perhaps the most widely used estimator which offers both relatively accurate estimator of the variability and leads to acceptable hypothesis testing is the estimator proposed by DeLong, DeLong and Clarke-Pearson (1988). This estimator possesses an upward bias which on the one hand results in an improved (compared to the unbiased estimator) type I error of the statistical test for equality of the AUCs when AUCs are small, but on the other hand results in loss of statistical power when AUCs are large (Bandos 2005; Bandos, Rockette & Gur 2005).

Absence of a uniformly superior method, potentially poor small-sample properties of the asymptotic procedures; complexity or unavailability of the variance formulas for generalized indices (such as for AUC extensions for clustered, repeated and multi-class data) have lead many investigators to suggest using the resampling methods such as jackknife, bootstrap and permutations in applications to the AUC and its extensions (Dorfman, Berbaum & Metz, 1992; Mossman 1995; Song, 1997; Beiden, Wagner, & Campbell, 2000; Emir et al, 2000; Rutter, 2000; Hand & Till, 2001; Nakas & Yiannoutsos 2004; Bandos, Rockette, & Gur, 2005, 2006a,b).

Because of the variety of methods for assessing variability of a single AUC estimate or comparing several AUCs it is important to know their relative advantages and limitations. Previously we developed a permutation test for comparing AUCs with paired data, constructed a precise approximation based on the closed-form solution for the exact permutation variance and investigated its properties relative to the conventional approach (Bandos et al 2005). The closed-form solutions for the exact (ideal) resampling variances that we derived in that as well as in our other works permit a better understanding of the relationships and relative advantages of resampling procedures and other methods for the assessment of AUCs (Bandos 2005; Bandos et al. 2006b). In this paper we discuss the relative merits of the jackknife, bootstrap and permutation procedures applied to a single AUC or difference between two correlated AUCs.

2. Preliminaries

We assume that the true class ("normal" or "abnormal") is uniquely determined and known for each subject. Hence, according to the true status, every subject in the population can be classified as normal or abnormal. We term the ordinal output of the classification as the subject's classification score and denote $\boldsymbol{x}$ and $\boldsymbol{y}$ as scores for normal and abnormal subjects correspondingly. Furthermore, without loss of generality, we will assume that higher values of the scores are associated with higher probabilities of the presence of "abnormality".

The general layout of the data we consider consists of scores assigned to samples of N "normal" and M "abnormal" subjects by each of the classification systems. We enumerate subjects with subscripts *i*, *k* (for normal); *j*, *l* (for abnormal). Thus, x_i, y_j denote the classification scores assigned to the *i*th "normal" and *j*th "abnormal" subjects. When operating with more than one classification system we distinguish between them with the superscript m (e.g. x_i^m). However, when the discussion concerns primarily a single-system setting we omit the corresponding index for the sake of simplicity.

Using the conventions defined above, the nonparametric estimator of the AUC or "sample AUC" (equivalent to the Wilxocon-Mann-Whitney statistic) can be written as:

$$\hat{A} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\psi(x_i, y_j)}{NM} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\psi_{ij}}{NM} = \frac{\psi_{\bullet\bullet}}{NM} = \overline{\psi}_{\bullet\bullet} \qquad (1)$$

where the *order indicator,* $\boldsymbol{\psi}$, is defined as follows:

$$\psi_{ij} = \psi(x_i, y_j) = \begin{cases} 1 & x_i < y_j \\ ½ & x_i = y_j \\ 0 & x_i > y_j \end{cases} \qquad (2)$$

Also, the dot in the place of the index in the subscript of a quantity denotes summation over the corresponding index; and the bar over the quantity, placed in addition to the dot in the subscript, denotes the average over the doted index.

Under a paired design, the difference in AUCs can be written as:

$$\hat{A}^1 - \hat{A}^2 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\psi(x_i^1, y_j^1) - \psi(x_i^2, y_j^2)\right]}{NM} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} w_{ij}}{NM} = \overline{w}_{\bullet\bullet} \qquad (3)$$

where

$$w_{ij} = w(x_i, y_j) = \psi(x_i^1, y_j^1) - \psi(x_i^2, y_j^2) = \psi_{ij}^1 - \psi_{ij}^2 \qquad (4)$$

This representation illustrates that the difference in areas under a paired design has the same structure as the single AUC estimator (1) and allows one to modify expressions derived for a single AUC to those for the AUC difference simply by replacing ψ_{ij} with w_{ij}.

3. Resampling approaches

Resampling approaches such as jackknife, bootstrap, permutations and combination thereof are widely used whenever conventional solutions are questionable, difficult to derive or unavailable. Major advantages of these methods include offering reliable statistical inferences in small sample problems and circumventing the difficulties of deriving the statistical moments of complex summary statistics.

3.1 Jackknife

Jackknife is a simple resampling approach that is often attributed to Quenouille (1949) and Tukey (1958). Many different varieties of the jackknife can be implemented in practice. The performance of several of them in hypothesis testing about AUC was considered by Song (1997). Although often forgotten, the variance estimators used in the procedure proposed by the DeLong et al. (1989) is also a jackknife variance estimator for the two-sample U-statistics (Arvesen, 1969). This procedure, which we will often term as "two-sample jackknife", is perhaps the most commonly used nonparametric method for comparing several correlated AUCs. In a more complex multi-reader setting a conventional "one-sample" jackknife was employed by Dorfman, Berbaum & Metz (1992) within an ANOVA framework.

The general idea of the jackknife is to generate multiple samples from the single original one by eliminating a fixed number of observations. The jackknife samples are then used as a base for calculation of the pseudo-values of a summary statistic, that are later used for inferential purposes. Since the nonparametric estimator of the AUC is an unbiased statistic, the one-sample and two-sample jackknife estimator (averages of the pseudovalues) are equal to the original one. Thus, the difference in these jackknife approaches occurs in the variances. A one-sample jackknife computes the variability of the pseudovalues regardless of the class of the eliminated subject while the two-sample jackknife computes a stratified variance. Both variances can be expressed in a closed-form and thus permit an easy comparison of these (Bandos 2005). Namely, the two-sample jackknife variance for the AUC (DeLong et al) can be written as:

$$V_{J2}(A)=\frac{\sum_{i=1}^{N}\left(\overline{\psi}_{i\bullet}-\overline{\psi}_{\bullet\bullet}\right)^2}{N(N-1)}+\frac{\sum_{j=1}^{M}\left(\overline{\psi}_{\bullet j}-\overline{\psi}_{\bullet\bullet}\right)^2}{M(M-1)} \qquad (5)$$

A one-sample jackknife variance has the following form:

$$V_{J1}(A)=\left[\frac{\sum_{i=1}^{N}\left(\overline{\psi}_{i\bullet}-\overline{\psi}_{\bullet\bullet}\right)^2}{(N-1)^2}+\frac{\sum_{j=1}^{M}\left(\overline{\psi}_{\bullet j}-\overline{\psi}_{\bullet\bullet}\right)^2}{(M-1)^2}\right]\times\frac{N+M-1}{N+M} \qquad (6)$$

A straightforward comparison of formulas (5) and (6) reveals that a one-sample jackknife variance is always larger than the two-sample one. This fact limits the usefulness of a one-sample variance since the two-sample jackknife variance is already greater than the Bamber-Wieand unbiased estimator and thus has an upward bias (Bandos 2005).

Although the jackknife approach is straightforward to implement and possesses good asymptotic properties, it is generally considered to be inferior compared to more advanced resampling techniques such as bootstrap. In application to the difference between AUCs the bootstrap variance estimator was also found to have lower mean squared error than the jackknife (Bandos, 2005). However, under certain conditions the jackknife can be considered as a linear approximation to the bootstrap (Efron & Tibshirani, 1993) and for some problems the jackknife might result in a statistical procedure that is practically indifferent from the bootstrap-based one.

3.2 Bootstrap

A good summary of the general bootstrap methodology can be found in the book by Efron & Tibshirani (1993). In ROC analysis bootstrap is commonly used for estimation of variability or for construction of confidence intervals. In recent years it has gained increased popularity in connection with its ability to obtain insight into the components of the variability of the indices estimated in multi-reader data (Beiden, Wagner & Campbell, 2000). The bootstrap was also proposed to be used for estimation of the variance of the partial AUC (Dodd & Pepe, 2003b), variance of the AUC computed from patient-clustered (Rutter, 2000) and repeated measures data (Emir et al., 2000).

The concept of the bootstrap is to build a model for the population sample space from the resamples (with replacement) of the original data. The nonparametric bootstrap completes the formation of the bootstrap sample space by assigning equal probability to all bootstrap samples. Next, a value of the summary statistic (called its bootstrap value) is calculated from every bootstrap sample and the set of all bootstrap values determines a bootstrap distribution. Such a bootstrap distribution of the summary statistic is a nonparametric maximum likelihood estimator of the distribution of the statistic computed on a sample randomly selected from a target population and serves as the basis for the bootstrap estimators of distributional parameters.

Since, even for a moderately sized problem, it may not be computationally feasible to draw all possible bootstrap samples, the conventional approach is to approximate the bootstrap distribution by computing the bootstrap values corresponding to a random sample of the bootstrap samples. Such a procedure is often called Monte Carlo or approximate bootstrap and the quantities computed from an approximate bootstrap distribution are called Monte Carlo bootstrap estimators in contrast to the quantities of the exact bootstrap distribution which are called ideal bootstrap estimators. The Monte Carlo bootstrap might still be computationally burdensome and also leads to an additional sampling error in the resulting estimators.

Some summary statistics permit circumventing the drawbacks of the Monte Carlo approach by allowing computation of ideal (exact) bootstrap quantities directly from the data. Unfortunately, the exact bootstrap variance is rarely obtainable except for simple statistics such as the sample mean. Some other estimators for which the exact bootstrap moments have been derived include sample median (Maritz & Jarret, 1978) and L-estimators (Hutson & Ernst, 2000).

In our recent work (Bandos 2005; Bandos, Rockette & Gur, 2006b) we have shown that the nonparametric estimator of the AUC permits the derivation of the analytical expression for the ideal bootstrap variance for several commonly used data structures (the bootstrap expectation of the AUC is equal to the original estimate). These results not only eliminate the need of the Monte Carlo approximation to the bootstrap of the AUC in existing methods, but can also be extended to the bootstrap applications for the patient-clustered data, repeated measure data, partial areas and potentially to a multi-class AUC extension (Hand & Till, 2001; Nakas & Yiannoutsos, 2004). For the single AUC the exact bootstrap variance has the following form:

$$V_B(A) = \frac{\sum_{i=1}^{N}\left(\overline{\psi}_{i\bullet} - \overline{\psi}_{\bullet\bullet}\right)^2}{N^2} + \frac{\sum_{j=1}^{M}\left(\overline{\psi}_{\bullet j} - \overline{\psi}_{\bullet\bullet}\right)^2}{M^2} + \\ + \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\psi_{ij} - \overline{\psi}_{i\bullet} - \overline{\psi}_{\bullet j} + \overline{\psi}_{\bullet\bullet}\right)^2}{N^2 M^2} \qquad (7)$$

Unfortunately, there is no uniform relationship between the bootstrap variance and that of any of the considered jackknife variances. The Monte Carlo investigations indicate that the bootstrap variance has uniformly smaller mean squared error. It also has a smaller bias except for very large AUC. Thus, the bootstrap often provides a better estimator of the variability than the jackknife. However, the estimator of Bamber (1975) and Wieand et al. (1983), because of its unbiasedness, might be preferred by some investigators.

Although the nonparametric bootstrap is a powerful approach that produces nonparametric maximum likelihood estimators, it is not uniformly the best resampling technique. Davison & Hinkley (1997) indicate that for hierarchical data a combination of resampling with and without replacement may better reflect the correlation structure in the general population. Furthermore, although the bootstrap can be implemented for a broad range of problems, in situations where there is something to permute (e.g. single index hypothesis testing, comparison of several indices) the permutation approach may be preferable because of the exact nature of the inferences (Efron & Tibshirani, 1993).

3.3 Permutations

Permutation procedures are usually associated with the early works of Fisher (1935). In ROC analysis permutation tests have been employed for comparison of the diagnostic modalities (Venkatraman & Begg, 1996; Venkatraman 2000; Bandos, Rockette & Gur, 2005).

Permutation based procedures are resampling procedures that are specific to hypothesis testing. Similar to the bootstrap, a permutation procedure constructs a permutation sample space, which consists of the equally likely permutation samples. The permutation samples are created by interchanging the units of the data that are assumed to be “exchangeable” under the null hypothesis. However, unlike the bootstrap sample space, the permutation sample space is the exact probability space of the possible arrangements of the data under the null hypothesis given the original sample.

The same permutation scheme can be used with different summary statistics resulting in different statistical tests. The choice of the summary statistic determines the alternatives that are more likely to be detected, but may not affect the null hypothesis. In this respect, permutation tests are similar to the tests of trend which, still assuming overall equality under the null hypothesis, aim to detect specific alternatives in the complementary hypothesis, e.g. a specific trend (linear, quadratic).

For example, when two diagnostic systems are to be compared with paired data, the natural permutation scheme consists of exchanging the paired units. Several reasonable permutation tests are possible under such a permutation scheme. One of these was developed by Venkatraman & Begg (1996) for detecting any differences between two ROC curves. For this purpose the authors used a measure specifically designed to detect the differences at every operating point. In our recent work (Bandos, Rockette & Gur, 2005) on a test that is especially sensitive to the difference in overall diagnostic performance we used the differences in nonparametric AUCs as a summary measure. Both of these tests assume the same condition of exchangeability of the diagnostic results under the null hypothesis, but differ with respect to their sensitivity to specific alternatives and the availability of an asymptotic version. Namely our permutation test better detects different ROC curves if they differ with respect to the AUC, and it has an easy-to-implement and precise approximation which is unavailable for the test of Venkatraman & Begg.

The availability of the asymptotic approximation to the permutation test can be an important issue since the exact permutation tests are practically impossible to implement with even moderate sample sizes and the Monte Carlo approximation to the permutation test is associated with a sampling error. Fortunately, in some cases the asymptotic approximation can be constructed by appealing to the asymptotic normality of the summary statistic and using the estimator of its variance, if the latter is derivable. For the nonparametric estimator of the difference in the AUC we demonstrated (Bandos, Rockette & Gur, 2005) that the exact permutation variance can be calculated directly without actually permuting the data, i.e.:

$$V_\Omega\left(A^1 - A^2\right) = \frac{\sum_{i=1}^{N}\left(\overline{w}_{i\bullet}^{1\bullet}\right)^2}{N^2} + \frac{\sum_{j=1}^{M}\left(\overline{w}_{\bullet j}^{\bullet 1}\right)^2}{M^2} \qquad (8)$$

where

$$w_{ij}^{p,q} = \psi\left(x_i^p, y_j^q\right) - \psi\left(x_i^{3-p}, y_j^{3-q}\right)$$

denotes the difference in the order indicators computed over the scores combined over the two systems.

The availability of an analytical expression for the exact permutation variance not only permits constructing an easy-to-compute approximation, but also makes such an approximation very precise even with small samples. Because of the restriction to the null hypothesis, the permutation variance is not directly comparable to previously mention estimation methods which provide estimators of the variance regardless of the magnitude of the difference. However, the properties of the statistical tests can be compared directly with Monte Carlo and the availability of the closed-form solution for the permutation variance greatly alleviates the computational burden of this task. The comparison of the asymptotic permutation test with the widely used procedure of DeLong et al. indicate the advantages of the former for the range of parameters common in diagnostic imaging , i.e. AUC greater than 0.8 and correlation between scores greater than 0.4 (Bandos et al., 2005).

4. Discussion

In this paper we discussed the relative merits of basic resampling approaches and outline some recent developments in the resampling-based procedures focused on the area under the ROC curve. The major drawbacks of the advanced resampling procedures are computational burden and sampling error. Sampling error results from the application of the Monte Carlo approximation to the resampling process, and adds to the uncertainty of the obtained results. Although alleviated by the development of faster computers the computational burden can still be substantial especially in the case of iteratively obtained estimators such as m.l.e. of AUC (Dorfmann & Alf 1969; Metz, Herman & Shen 1998) or when assessing the uncertainty of the resampling-based estimators (e.g. jackknife- or bootstrap-after-bootstrap). In our previous works we showed that for the nonparametric estimator of the AUC presented here all of the considered resampling procedures permit derivation of the ideal variances directly avoiding implementation of the resampling process or its approximation. Such closed-form solutions greatly reduce computational burden, eliminate a sampling error associated with the Monte Carlo approximation to the resampling variances, permit construction of precise approximations to the exact methods and facilitate assessment and comparison of the properties of various statistical procedures based on resampling.

In general jackknife provides a somewhat simplistic method that, depending on the problem, may still offer valuable solutions. In application to estimation of the nonparametric AUC, the two-sample jackknife is preferable over the one-sample due to a smaller upward bias. Bootstrap is a more elaborate resampling procedure that provides nonparametric maximum likelihood estimators by offering an approximation to the population sample space. Bootstrap is usually preferred over the jackknife because of cleaner interpretation and sometimes better precision. Exploiting a formula for the exact bootstrap variance of the AUC we demonstrated that it provides an estimator of the variance that is more accurate in terms of the mean squared error than the two-sample jackknife variance and is often more efficient than the unbiased estimator. In the case of comparing two AUCs the asymptotic tests based on the bootstrap and jackknife variances have very similar characteristics. However, for more complex problems the bootstrap may perform better than the jackknife. The permutations explore the properties of the population sample space assuming the exchangeability satisfied under the null hypotheses. For the comparison of the performances under a paired design the permutation test can be considered as preferable over the bootstrap and jackknife due to the exact nature of the permutation inferences. The availability of the exact permutation variance permits construction of an easy-to–implement and precise approximation and facilitates investigation of the properties of the permutation test. Compared to the two-sample jackknife asymptotic test for comparing two correlated AUCs, the asymptotic permutation test was shown to have greater statistical power for the range of parameter common in diagnostic radiology.

Although this paper focuses on the most commonly used summary index, AUC, the availability of the analytical expression for the exact variances is not limited to this relatively simple case. Formulas for ideal variances may also appear derivable for other AUC related indices and for different types of data (multi-reader, clustered, repeated measures and multi-class data) as well as under other, more complex, resampling schemes or study designs.

Acknowledgments

This work is supported in part by Public Health Service grants EB002106 and EB001694 (to the University of Pittsburgh) from the National Institute for Biomedical Imaging and Bioengineering (NIBIB), National Health Institutes, Department of Health and Human Services.

References

Arvesen, J.N. (1969). Jackknifing U-statistics. *Annals of Mathematical Statistics* 40(6), 2076-2100.

Bamber D. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology* 12, 387-415.

Bandos, A. (2005). Nonparametric methods in comparing two ROC curves. Doctoral dissertation, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh. (http://etd.library.pitt.edu /ETD/available/etd-07292005-012632/)

Bandos, A.I., Rockette, H.E., Gur, D. (2005). A permutation test sensitive to differences in areas for comparing ROC curves from a paired design. *Statistics in Medicine* 24(18), 2873-2893.

Bandos, A.I., Rockette, H.E., Gur, D. (2006a). A permutation test for comparing ROC curves in multireader studies. *Academic Radiology* 13, 414-420.

Bandos, A.I., Rockette, H.E., Gur, D. (2006b). Components of the bootstrap variance of the areas under the ROC curve. *IBS ENAR* 2006, Tampa, FL.

Beiden, S.V., Wagner, R.F., Campbell, G. (2000). Components-of-variance models and multiple-bootstrap experiments: an alternative method for random-effects receiver operating characteristic analysis. *Academic Radiology* 7, 341-349.

Davison, A.C., Hinkley, D.V. (1997). *Bootstrap methods and their application*. Edinburgh: Cambridge University Press.

DeLong, E.R., DeLong, D.M., Clarke-Pearson, D.L. (1988). Comparing the area under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics* 44(3), 837-845.

Dodd, L.E., Pepe, M.S. (2003a). Semiparametric regression for the area under the receiver operating characteristic curve. *Journal of the American Statistical Association* 98, 409-417.

Dodd, L.E., Pepe, M.S. (2003b). Partial AUC estimation and regression. *Biometrics* 59, 614-623.

Dorfman, D.D., Alf JrE. (1969). Maximum likelihood estimation of parameters of signal detection theory and determination of confidence intervals – rating-method data. *Journal of Mathematical Psychology* 6, 487-496.

Dorfman, D.D., Berbaum, K.S., Metz, C.E. (1992). Receiver operating characteristic rating analysis: generalization to the population of readers and patients with the jackknife method. *Investigative Radiology* 27, 723-731.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Efron, B., Tibshirani, R.J. (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.

Emir, B., Wieand, S., Jung, S.H., Ying, Z. (2000). Comparison of diagnostic markers with repeated measurements: a non-parametric ROC curve approach. *Statistics in Medicine* 19, 511-523.

Ferri, C., Flach, P., Hernandez-Orallo, J. (2002). Learning decision trees using area under the ROC curve. *Proceedings of ICML-2002*.

Fisher, R.A. (1935). *Design of experiments*. Oliver and Boyd, Edinburgh

Hand, D.J., Till, R.J. (2001). A simple generalization of the area under the ROC curve for multiple class classification problems. *Machine Learning* 45, 171-186.

Hanley, J.A., McNeil, B.J. (1982). The meaning and use of the Area under Receiver Operating Characteristic (ROC) Curve. *Radiology* 143, 29-36.

Hanley, J.A., Hajian-Tilaki, K.O. (1997). Sampling variability of nonparametric estimates of the areas under receiver operating characteristic curves: an update. *Academic Radiology* 4, 49-58.

Hutson, A.D., Ernst, M.D. (2000). The exact bootstrap mean and variance of an L-estimator. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)* 62(1), 89-94.

Maritz, J.S., Jarrett, R.G. (1978). A note on estimating the variance of the sample median. *Journal of the American Statistical Association* 73(361), 194-196.

Metz, C.E., Herman, B.A., Shen, J. (1998). Maximum likelihood estimation of receiver operating characteristic (ROC) curves from continuously distributed data. *Statistics in Medicine* 17, 1033-1053.

Mossman, D. (1995). Resampling techniques in the analysis of non-binormal ROC data. Medical decision making 15, 358-366.

Nakas, C.T., Yiannoutsos, C.T. (2004). Ordered multiple-class ROC analysis with continuous measurements. *Statistics in Medicine* 23, 3437-3449.

Noether GE. Elements on Nonparametric Statistics. *Wiley & Sons Inc.:* New York 1967.

Obuchowski, N.A. (1994). Computing sample size for receiver operating characteristics studies. *Investigative Radiology* 29, 238-243.

Obuchowski, N.A. (1997). Nonparametric analysis of clustered ROC curve data. *Biometrics* 53, 567-578.

Pepe, M.S., Thompson, M.L. (2000). Combining diagnostic test results to increase accuracy. *Biostatistics* 1, 123-140.

Pepe, M.S. (2003). *The statistical evaluation of medical test for classification and prediction*. Oxford: Oxford University Press.

Pepe, M.S., Cai, T., Longton, G. (2006). Combining predictors for classification using the area under the receiver operating characteristic curve. *Biometrics* 62, 221-229.

Quenouille, M.H (1949). Approximate tests of correlation in time series. Journal of Royal Statistical Society, Series B 11, 18-84.

Rutter, C.M. (2000). Bootstrap estimation of diagnostic accuracy with patient-clustered data. *Academic Radiology* 7, 413-419.

Song, H.H. (1997). Analysis of correlated ROC areas in diagnostic testing. *Biometrics* 53(1), 370-382.

Swets, J.A., Picket, R.M. (1982). *Evaluation of diagnostic systems: methods from signal detection theory*. New York: Academic Press.

Tukey, J.W. (1958). Bias and confidence in not quite large samples (abstract). *Annals of Mathematical Statistics* 29, 614.

Venkatraman, E.S., Begg, C.B. (1996). A distribution-free procedure for comparing receiver operating characteristic curves from a paired experiment. *Biometrika* 83(4), 835-848.

Venkatraman, E.S. (2000) A permutation test to compare receiver operating characteristic curves. *Biometrics* 56, 1134-1136.

Verrelst, H., Moreau, Y., Vandewalle, J., Timmerman, D. (1998). Use a multi-layer perceptron to predict malignancy in ovarian tumors. *Advances in Neural Information Processing Systems*, 10.

Wieand, H.S., Gail, M.M., Hanley, J.A. (1983). A nonparametric procedure for comparing diagnostic tests with paired or unpaired data. *I.M.S. Bulletin* 12, 213-214.

Yan, L., Dodier, R., Mozer, M.C., Wolniewicz, R. (2003). Optimizing Classifier performance via an approximation to the Wilcoxon-Mann-Whitney Statistic. *Proceedings of ICML-2003*.

Zhou, X.H., Obuchowski, N.A., McClish D.K. (2002). *Statistical methods in diagnostic medicine*. New York: Wiley & Sons Inc.

Estimating the Class Probability Threshold without Training Data

Ricardo Blanco-Vega **RBLANCO@DSIC.UPV.ES**
César Ferri-Ramírez **CFERRI@DSIC.UPV.ES**
José Hernández-Orallo **JORALLO@DSIC.UPV.ES**
María José Ramírez-Quintana **MRAMIREZ@DSIC.UPV.ES**
Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, C. de Vera s/n, 46022 Valencia, Spain

Abstract

In this paper we analyse three different techniques to establish an optimal-cost class threshold when training data is not available. One technique is directly derived from the definition of cost, a second one is derived from a ranking of estimated probabilities and the third one is based on ROC analysis. We analyse the approaches theoretically and experimentally, applied to the adaptation of existing models. The results show that the techniques we present are better for reducing the overall cost than the classical approaches (e.g. oversampling) and show that cost contextualisation can be performed with good results when no data is available.

1. Introduction

The traditional solution to the problem of contextualising a classifier to a new cost is ROC analysis. In order to perform ROC analysis (as well as other techniques), we need a training or validation dataset, from which we draw the ROC curve in the ROC space. In some situations, however, we don't have any training or validation data analysis available.

This situation is frequent when we have to adapt an existing method which was elaborated by a human expert, or the model is so old that we do not have the old training data used for constructing the initial model available. This is a typical situation in many areas such as engineering, diagnosis, manufacturing, medicine, business, etc.

Therefore, the techniques from machine learning or data mining, although they are more and more useful and frequent in knowledge acquisition, cannot be applied if we have models that we want to adapt or to transform, but we do not have the original data.

An old technique that can work without training data is the recently called "cost-sensitive learning by example weighting" (Abe et. al., 2004). The methods which follow this philosophy modify the data distribution in order to train a new model which becomes cost-sensitive. The typical approach in this line is stratification (Breiman et. al., 1984; Chan and Stolfo, 1998) by oversampling or undersampling.

An alternative approach is the use of a threshold. A technique that could be adapted when data is not available can be derived from the classical formulas of cost-sensitive learning. It is straightforward to see (see e.g. Elkan, 2001) that the optimal prediction for an example *x* in class *i* is the one that minimises

$$L(x,i) = \sum_{j} P(j \mid x) C(i,j) \qquad \textbf{(1)}$$

where $P(j|x)$ is the estimated probability for each class *j* given the example *x*, and $C(i,j)$ is the cell in the cost matrix *C* which defines the cost of predicting class *i* when the true class is *j*. From the previous formula, as we will see, we can establish a direct threshold without having any extra data at hand. In fact, some existing works (Domingos, 1999) have used the previous formula to establish a threshold which generates a model which is cost sensitive.

One of the most adequate ways to establish a class threshold is based on ROC analysis. (Lachiche & Flach, 2003) extend the general technique and show that it is also useful when the cost has not changed. However, in these cases we need additional validation data, in order to draw the curves.

In order to tackle the problem that we have described at the beginning (adapting an existing model without data), it would be interesting, then, to analyse some techniques

Appearing in *Proceedings of the third Workshop on ROC Analysis in Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

which combine the direct threshold estimation based on formula 1 (which ignores any estimated probabilities) and methods which take them into account (either their ranking or their absolute value) in a similar way ROC analysis works, but without data.

In order to adapt the existing models, we use the mimetic technique (Domingos, 1997, 1998; Estruch, Ferri, Hernández & Ramírez, 2003; Blanco, Hernández & Ramírez, 2004) to generate a model which is similar to the initial model (oracle) but contextualised to the new cost. In order to do this, we propose at least six different ways to diminish the global cost of the mimetic model. Three criteria for adapting the classification threshold, as we have mentioned, and several different schemas for the mimetic technique are set out (without counting on the original data). We have centered our study on binary classification problems.

The mimetic method is a technique for converting an incomprehensible model into one simple and comprehensible representation. Basically, it considers the incomprehensible model as an oracle, which is used for labelling an invented dataset. Then, a comprehensible model (for instance, a decision tree) is trained with the invented dataset. The mimetic technique has usually been used for obtaining comprehensible models. However, there is no reason for ignoring it as a cost-sensitive adaptation technique since it is in fact a model transformation technique.

Note that the mimetic technique is a transformation technique which can use any learning technique, since the mimetic model is induced from (invented) data.

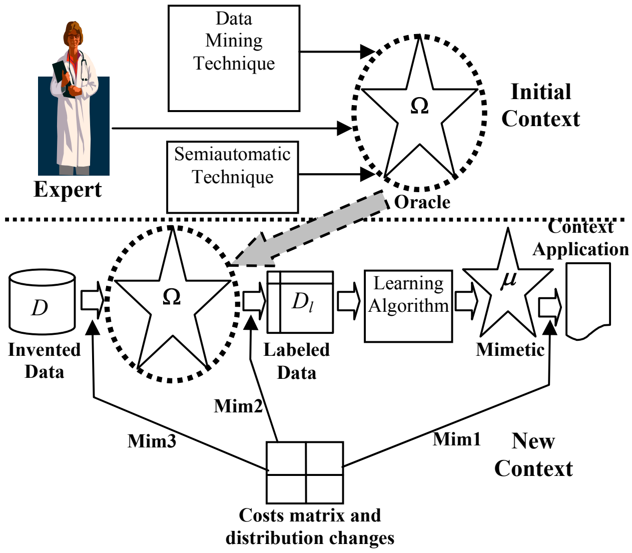


Figure 1. The mimetic context.

The mimetic context validation (see Figure 1) that we propose allows us to change the context of the initial model (oracle) so that it becomes sensitive to the new cost.

The main advantages of our proposal are that it does not require a retraining of the initial model with the old data and, hence, it is not necessary to know the original data. The only thing we need from the original data or for the formulation of the problem is to know the maximum and minimum values of its attributes.

From these maximum and minimum values and applying the uniform distribution we can obtain an invented dataset, which is labelled by using the oracle. We can use the cost information in different points: on the invented dataset, on the labelling of the data or on the thresholds. This settles three moments at which the cost information is used (see Figure 1, the points Mim1, Mim2 and Mim3). One of the points (Mim2) is especially interesting from the rest because it generates a "specific rule formulation" for the model, which might serve as any explanation of the adaptation to the new costs.

The paper is organised as follows. In section 2 we describe the three methods to determine the thresholds and analyse theoretically the relationships between them. In section 3 we describe the four styles for the generation of invented data and, the schemes used in this work for the learning of the mimetic models. In section 4 we describe the different configurations. We also include the experimental evaluation conducted and the general results, which demonstrate the appropriateness and benefits of our proposal to contextualise any model to a new cost context. Finally, section 5 presents the conclusions and future work.

2. Threshold Estimation

In this section, we present three different methods to estimate an optimal threshold following different philosophies. We also study some theoretical properties of the methods.

In contexts where there are different costs associated to the misclassification errors, or where the class distributions are not identical, a usual way of reducing costs (apart from oversampling) is to find an optimal decision threshold in order to classify new instances according to their associated cost. Traditionally, the way in which the threshold is determined is performed in a simple way (Elkan, 2001), only taking the context *skew* into account.

As we have said in the introduction, the methods based on ROC analysis (e.g. Lachiche & Flach, 2003) require a validation dataset, which is created at the expense of reducing data in the training dataset. Here, we are only interested in threshold estimation methods that don't require extra data, since we do not have any data available (either old or new training or test). Therefore, we will not study this method or others which are related which require a dataset. We will just present methods which can work without it.

In this section we consider two-class problems, with class names 0 and 1. Given a cost matrix C, we define the cost *skew* as:

$$skew = \frac{C(0,1) - C(1,1)}{C(1,0) - C(0,0)} \qquad \textbf{(2)}$$

2.1 Direct Threshold

The first method to obtain the threshold completely ignores the estimated probabilities of the models, i.e., to estimate the threshold it only considers the cost *skew*. According to (Elkan, 2001), the optimal prediction is class 1 if and only if the expected cost of this prediction is lower than or equal to the expected cost of predicting class 0:

$$P(0|x) \cdot C(1,0) + P(1|x) \cdot C(1,1) \le P(0|x) \cdot C(0,0) + P(1|x) \cdot C(0,1)$$

If $p = P(1|x)$ we have:

$$(1-p) \cdot C(1,0) + p \cdot C(1,1) \le (1-p) \cdot C(0,0) + p \cdot C(0,1)$$

Then, the threshold for making optimal decisions is a probability $p*$ such that:

$$(1-p^*) \cdot C(1,0) + p^* \cdot C(1,1) = (1-p^*) \cdot C(0,0) + p^* \cdot C(0,1)$$

Assuming that $C(1,0)>C(0,0)$ and $C(0,1)>C(1,1)$ (i.e. misclassifications are more expensive than right predictions), we have

$$p^* = \frac{C(1,0) - C(0,0)}{C(1,0) - C(0,0) + C(0,1) - C(1,1)}$$

$$p^* = \frac{1}{1 + skew}$$

Finally, we define the threshold as:

$$Threshold_{Dir} = 1 - p^* = \frac{skew}{1 + skew} \qquad \textbf{(3)}$$

2.2 Ranking or Sorting Threshold

The previous method for estimating the classification ignores the estimated probabilities in a proper way. This can be a problem for models that do not distribute the estimated probabilities. Imagine a model that only assigns probabilities within the range 0.6-0.7. In this situation, most of the *skews* will not vary the results of the model.

In order to partially avoid this limitation, we propose a new method to estimate the threshold. The idea is to employ the estimated probabilities directly to compute the threshold. For this purpose, if we have n examples, we rank these examples according to their estimated probabilities of being class 0. We select a point (Pos) between two points (a,b) in this rank such that there are (approximately) $n/(skew+1)$ examples on the left side and (n^* $skew/(skew+1)$) examples on the right side. In this division point we can find the desired threshold. We can illustrate this situation with Figure 2:

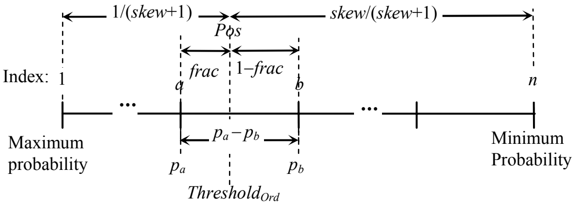


Figure 2. Position of the threshold in the sorting method

Following this figure, we have

$$Pos = \frac{n-1}{skew + 1} + 1, \; a = Lower(Pos), \;\; b = a + 1$$

where *Lower* computes the integer part of a real number. Then we estimate the threshold as:

$$Threshold_{Ord} = p_a - (p_a - p_b) \cdot frac \qquad \textbf{(4)}$$

where

$$frac = Pos - a$$

In the case we find more than one example with the same estimated probability, we distribute these examples in a similar way. A complete explanation of the procedure can be found in (Blanco, 2006).

2.3 ROC Threshold

Although the previous method considers the *skew* and the estimated probabilities to compute the threshold, it has an important problem because the value of the threshold is restricted to the range of the probabilities computed by the model. I.e, if a model always computes probability estimates between 0.4 and 0.5, the threshold will be within this range for any *skew*.

Motivated by this limitation, we have studied a new method to compute the threshold based on ROC analysis. Suppose that a model is well calibrated, this fact means that if a model gives a probability 0.8 of being class 0 to 100 examples, 80 should be of class 0, and 20 should be of class 1. In the ROC space, this will be a segment going from point (0,0) to the point (20,80) with a slope of 4.

In order to compute this new threshold we define a version of the ROC curve named NROC. This new curve is based on the idea that a probability represents a percentage of correctly classified instances (calibrated classifier).

If we have a set of n examples ranked by the estimated probability of being class 0, we define Sum^0 as the sum of these probabilities. We consider normalised probabilities, then $Sum^0 + Sum^1 = n$. The space NROC is a 2 dimension square limited by (0,0) and (1,1). In order to draw a NROC curve, we only take the estimated probabilities into account, and we proceed as follows. If the first example has an estimated probability p_1 of being class 0, we draw a segment from the point (0,0) to the point $((1-p_1)/Sum^1,\ p_1/Sum^0)$. The next instance ($p_2$) will correspond to the second segment will be from $((1-p_1)/Sum^1, p_1/Sum^0)$ to $(((1-p_1)+(1-p_2))/Sum^1, ((p_1+p_2)/Sum^0)$. Following this procedure, the last segment will be between the points $(Sum^1-(1-p_n))/Sum^1, (Sum^0-p_n)/Sum^0)$ and (1,1).

Once we have defined the NROC space, let us explain how we use it to determine the threshold. First, since we work on a normalised ROC space (1×1) and Sum^0 is not always equal to Sum^1, we need to normalise the *skew*.

$$skew' = skew \cdot \frac{Sum^0}{Sum^1}$$

If *skew'* is exactly parallel to a segment, then the threshold must be exactly the probability that corresponds to that segment, i.e if $skew' = p_i/(1-p_i)$ the threshold must be p_i. This means:

$$Threshold_{ROC} = \frac{skew'}{1+skew'}$$

Using the relationship between *skew*' and *skew*:

$$Threshold_{ROC} = \frac{skew \cdot \frac{Sum^0}{Sum^1}}{1 + skew \cdot \frac{Sum^0}{Sum^1}}$$

$$Threshold_{ROC} = \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^0}{Sum^1}} \qquad \textbf{(5)}$$

2.4 Theoretical analysis of the threshold methods

Now, we study some properties of the methods for obtaining the threshold which we have described in the previous subsections. First, we show that the threshold which is calculated by each of the three methods is well-defined, that is, it is a real value between 0 and 1, as expected. Secondly, we analyse which the relationship between the three thresholds is.

The maximum and minimum values of the $Threshold_{Dir}$ and $Threshold_{ROC}$ depend on the *skew* by definition (formulae 3 and 5). Trivially, $Threshold_{Ord}$ belongs to the interval [0..1] since it is defined as a value between two example probabilities.

Maximum: For the direct and the ROC methods, the maximum is obtained when *skew*=∞:

$$\lim_{skew\to\infty} Threshold_{Dir} = \lim_{skew\to\infty} \frac{skew}{1+skew} = 1$$

$$\lim_{skew\to\infty} Threshold_{ROC} = \lim_{skew\to\infty} \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^0}{Sum^1}} = 1$$

The upper limit of $Threshold_{Ord}$ is not necessarily 1, since it is given by the example with highest probability.

Minimum: For the direct and the ROC methods, the minimum is obtained when *skew*=0:

$$\lim_{skew\to 0} Threshold_{Dir} = \lim_{skew\to 0} \frac{skew}{1+skew} = 0$$

$$\lim_{skew\to 0} Threshold_{ROC} = \lim_{skew\to 0} \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^0}{Sum^1}} = 0$$

As in the previous case, the lower limit of $Threshold_{Ord}$ is not necessarily 0, since it is given by the example with lowest probability.

Regarding the relationship among the three threshold methods, it is clear that we can found cases for which $Threshold_{Dir} > Threshold_{Ord}$, and viceversa, because, as we have just said, the $Threshold_{Ord}$ value depends on the example probability of being of class 0. A similar relationship holds between $Threshold_{ROC}$ and $Threshold_{Ord}$.

However, the relationship between $Threshold_{ROC}$ and $Threshold_{Dir}$ depends on the relationship between Sum_1 and Sum_0, as the following proposition shows:

Proposition 2. *Given n examples, let Sum^0 be the sum of the n (normalised) example probabilities of being in class 0, and let Sum^1 be $1-Sum^0$. If $Sum^0/Sum^1 > 1$ then $Threshold_{ROC} > Threshold_{Dir}$, if $Sum^0/Sum^1 < 1$ then $Threshold_{ROC} < Threshold_{Dir}$, and if $Sum^0/Sum^1 = 1$ then $Threshold_{ROC} = Threshold_{Dir}$.*

The following theorem shows that the three thresholds coincide when the probabilities are uniformly distributed.

.**Proposition 3.** *Given a set of n examples whose probabilities are uniformly distributed. Let P^0 be the sequence of these probabilities ranked downwardly:*

$$P^0 = \{1, \frac{m-1}{m}, \ldots, \frac{2}{m}, \frac{1}{m}, 0\}$$

such that the probability of example i being in class 0 *and class* 1 *are given respectively by*

$$p_i^0 = \frac{m-i+1}{m} \quad y \quad p_i^1 = \frac{i-1}{m}$$

where $m=n-1$.

Then, $Threshold_{ROC}=Threshold_{Dir}=Threshold_{Ord}$.

3. Mimetic Context

In this section we present the mimetic models we will study experimentally in the next section along with the threshold estimation seen in Section 2. For this purpose, we first introduce several ways to generate the invented dataset, as well as different learning schemes. Then, each configuration to be considered will be obtained by inventing its training dataset in a certain way, by applying one of the learning schemes and by using one of the thresholds defined in the previous section.

3.1 Generation of the training dataset for the mimetic technique

As we said in the introduction, we are assuming that the original dataset used for training the oracle is not available. Hence, the mimetic model is training by using only an invented dataset (labelled by the oracle) which is generated using the uniform distribution. This is a very simple approach, because in very few cases data follow this distribution. If we could know the a priori distribution of the data or we could have a sample where we could estimate this distribution, the results would be probably better. Note that, in this way we only need to make use of the range value of each attribute (that is, its maximum and minimum values).

In general, the invented dataset *D* can be generated by applying one of the following methods:

- **Type a: A priori method**. In this method, *D* preserves the class distribution of the original training dataset. To do this, the original class proportion has to be known at the time of the data generation.
- **Type b: Balanced method**. The same number of examples of each class is generated by this method. So *D* is composed by a 50% of examples of class 1 and a 50% of examples of class 0.
- **Type c: Random method**. The invented dataset *D* is obtained by only using the uniform distribution as it is (that is, no conditions about the class frequency in *D* are imposed).
- **Type d: Oversampling method**. This method makes that the class frequencies in the invented dataset are defined in terms of the *skew*, such that *D* contains a proportion of 1/(*skew*+1) of instances of class 0 and a proportion of *skew*/(*skew*+1) of instances of class 1.

In order to obtain the four types, we generate random examples and then we label them using the oracle. This process is finished when we obtain the correct percentage according to the selected type.

3.2 Mimetic Learning Schemes

In order to use the mimetic approach for a context sensitive learning, different mimetic learning schemes can be defined depending on the step of the mimetic process the context information is used: at the time of generating the invented dataset (scheme 3), at the time of labelling the invented dataset (scheme 1) or at the time of application of the mimetic model (scheme 2). We also consider another scheme (scheme 0) which corresponds to the situation where the context information is not used (as a reference). More specifically, we define the following mimetic learning schemes:

- **Scheme 0 (Mim0 model)**: This is the basic mimetic scheme. The mimetic model is obtained by applying a decision tree learner to the labelled data, namely the J48 classifier with pruning (Figure 3). Then, Mim0 is applied as a non sensitive context model that classifies a new example of class 0 if the probability for this class is greater or equal to 0.5 (threshold=0.5).

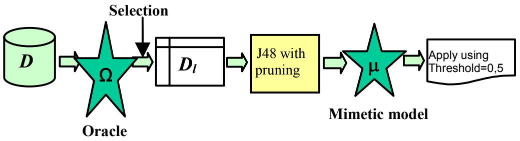


Figure 3. Scheme 0: The simple mimetic learning method.

- **Scheme 1 (Mim1 model)**: This is a posteriori scheme in that the context information is used when the mimetic model is applied. First, the mimetic model is obtained as usually (by using the J48 classifier without pruning). Then, the threshold is calculated from the mimetic model and the invented dataset. Finally, the Mim1 model uses these parameters to classify new examples. Figure 4 shows this learning scheme.

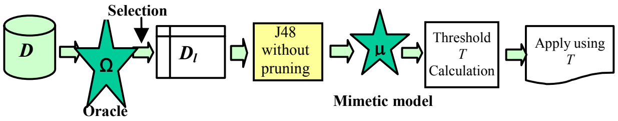


Figure 4. Scheme 1: The context information is used at the time of the mimetic model application.

- **Scheme 2 (Mim2 model):** This is a priori scheme in which the context information is used before the mimetic model is learned. Once the invented dataset has been labelled by the oracle, the threshold and the Ro index (if it is needed) are calculated from them. Then, the invented dataset is re-labelled using these parameters. The new dataset is used for training the mimetic model which is applied as in scheme 0. This learning scheme is very similar to the proposal of

(Domingos, 1999). Figure 5 illustrates this learning scheme.

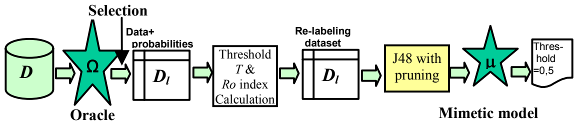


Figure 5. Scheme 2: The context information is used to re-label the invented dataset before the mimetic model is trained.

- **Scheme 3 (Mim3 model)**: This is a scheme in which the context information is used for generating the invented dataset using oversampling. Then, the mimetic model is generated and applied as in scheme 0 (Figure 6).

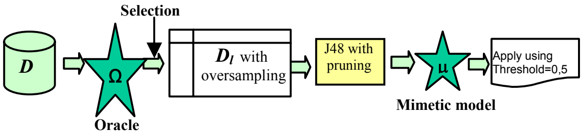


Figure 6. The context information is used at the time to generate the invented dataset by oversampling.

Note that the J48 learning method has been used with pruning in all the schemes except to scheme 1. This is due to the fact that, in this case, we need that the mimetic model provides good estimations of probabilities in order to calculate the threshold from them.

4. Experiments

In this section, we present the global results of the experimental evaluation of the mimetic technique as a model contextualization approach. A more exhaustive experimental evaluation can be found in (Blanco, 2006). The combinations we will analyse are obtained as follows. First, we combine the Mim1 and Mim2 models with the three thresholds defined in Section 2. This gives 6 different configurations. We also consider the Mim0 and Mim3 models. Finally, we combine all these models (except from Mim3) with the different ways of inventing the training dataset defined in Section 3. Summing up, the experimental configuration is composed by 22 mimetic models to be studied. In that follows, a mimetic model is denoted as Mim*nConfigType*, where *n* denotes a learning scheme (0≤*n*≤3), *Config* denotes the threshold used (Ord, Dir, ROC), and *Type* denotes the different types of invented dataset generation (a,b,c,d) described in section 3.1.

4.1 Experimental Setting

For the experiments, we have employed 20 datasets from the UCI repository (Black & Merz, 1998) (see Table 1**¡Error! No se encuentra el origen de la referencia.**).

Datasets from 1 to 10 have been used for the experiments in an (almost)-balanced data scenario, whereas the rest of them have been used for two unbalanced data situations: first, considering class 1 as the majority class and, secondly, as minority class. In all cases, we use cost matrices with *skew* values of 1, 2, 3, 5, and 10. The mimetic models have been built using the J48 algorithm implemented in Weka (Witten & Frank, 2005). Also, we have used two oracles: a Neural Network and a Naive Bayes algorithm (their implementations in Weka). This allows us to analyse our approach both when the oracle is calibrated (the case of the neural network which provides good calibration) and non-calibrated (the Naive Bayes classifier). The size of the invented dataset is 10,000 for all the experiments and we use Laplace correction for all the probabilities. For all the experiments, we use 10x10-fold cross-validation. Finally, when we show average results, we will use the arithmetic mean. We show the means because the number of variants is too large to include here the table with the paired t-tests. You can see these results in (Blanco, 2006).

Table 1. Information about the datasets used in the experiments.

No.	Dataset	Balanced	Attributes		Size	Size	
			Num.	Nom.		Class 0	Class 1
1	credit-a	Almost	6	9	690	307	383
2	heart-statlog	Almost	13	0	270	150	120
3	monks1	yes	0	6	556	278	278
4	monks3	Almost	0	6	554	266	288
5	monks2	Yes	0	6	412	206	206
6	tic-tac	Yes	0	8	664	332	332
7	breast-cancer	Yes	0	9	170	85	85
8	labor	Yes	8	8	40	20	20
9	vote	Yes	0	16	336	168	168
10	diabetes	Yes	8	0	536	268	268
11	haberman-breast	No	3	0	306	81	225
12	monks2	No	0	6	601	206	395
13	abalone-morethan	No	7	1	4177	1447	2730
14	tic-tac	No	0	8	958	332	626
15	breast-cancer	No	0	9	286	85	201
16	labor	No	8	8	57	20	37
17	vote	No	0	16	435	168	267
18	credit-g	No	7	13	1000	300	700
19	diabetes	No	8	0	768	268	500
20	liver	No	6	0	345	145	200

4.2 General Results

An overview of our approach is shown in Table 2, which presents the cost average of the mimetic models obtained in all experiments grouped by *skew*. As can be observed, for *skew*=1, the cost is quite similar in all mimetic models. However, as the *skew* value increases, the cost differences are more meaningful. Globally, and for *skew* values greater than 2, Mim2Ordb model presents the best behaviour, followed by Mim2Orda. For lower *skew* values, the best models are Mim2Dira and Mim2Dirb. Hence, from a cost point of view, it seems preferable to apply the cost information before the mimetic model is built (Mim2 configurations).

Table 2. Cost averages of all the mimetic models grouped by *skew*. In bold those with the lowest global cost.

Model	*skew*					Mean
	1	2	3	5	10	
Mim0a	**19.74**	29.58	39.68	59.53	109.81	51.67
Mim0b	20.24	30.30	40.27	60.91	114.45	53.24
Mim0c	20.56	31.35	41.95	63.39	116.87	54.82
Mim1Dira	20.00	29.41	38.03	51.34	75.88	42.93
Mim1Dirb	20.37	30.05	38.37	51.19	73.11	42.62
Mim1Dirc	20.72	30.82	39.29	51.74	73.16	43.15
Mim1Orda	25.20	32.00	34.32	36.86	39.07	33.49
Mim1Ordb	20.67	29.36	33.03	36.43	38.76	31.65
Mim1Ordc	26.17	34.85	39.49	43.84	47.22	38.32
Mim1ROCa	20.25	29.65	37.80	51.03	71.92	42.13
Mim1ROCb	20.37	29.84	37.66	50.60	72.49	42.19
Mim1ROCc	20.73	29.31	36.54	48.25	68.92	40.75
Mim2Dira	**19.74**	**27.00**	31.86	39.25	51.74	33.92
Mim2Dirb	20.24	27.58	32.24	39.67	51.91	34.33
Mim2Dirc	20.56	29.02	34.69	42.98	58.87	37.22
Mim2Orda	23.80	29.98	32.75	35.77	37.42	31.94
Mim2Ordb	20.61	28.51	**31.69**	**34.84**	**37.03**	**30.54**
Mim2Ordc	24.75	33.26	38.11	41.83	45.74	36.74
Mim2ROCa	20.43	27.77	32.65	39.10	50.22	34.03
Mim2ROCb	20.58	28.13	33.75	40.83	53.47	35.35
Mim2ROCc	20.83	29.30	34.92	44.46	60.12	37.93
Mim3	20.33	28.51	34.85	44.69	61.65	38.01

Table 3. Accuracies and cost averages of all the models according to the experiment type. Acc is Accuracy.

Model	Balanced		Majority		Minority	
	Acc.	Cost	Acc.	Cost	Acc.	Cost
Mim0a	77.40	23.32	73.62	52.76	**73.65**	78.93
Mim0b	**77.53**	23.46	72.09	64.49	72.32	71.75
Mim0c	76.24	29.53	72.92	63.73	73.01	71.22
Mim1Dira	76.22	20.30	73.36	41.73	70.84	66.76
Mim1Dirb	76.32	20.34	73.06	47.58	69.27	59.94
Mim1Dirc	75.62	22.98	72.98	47.77	70.34	58.69
Mim1Orda	65.38	17.39	69.53	33.69	50.70	49.39
Mim1Ordb	65.49	17.44	70.40	30.51	55.04	46.99
Mim1Ordc	63.88	20.87	68.03	41.35	54.28	52.72
Mim1Roca	76.31	20.28	73.42	46.78	68.27	59.33
Mim1Rocb	76.37	20.29	73.29	46.48	68.71	59.81
Mim1Rocc	75.81	19.52	73.36	45.30	67.74	57.43
Mim2Dira	75.05	**14.59**	74.11	35.63	67.92	51.53
Mim2Dirb	75.06	**14.59**	73.37	39.25	66.46	49.14
Mim2Dirc	73.75	20.96	73.33	40.68	66.77	50.03
Mim2Orda	67.58	16.53	71.32	32.59	54.77	46.71
Mim2Ordb	67.64	16.50	71.56	**30.18**	58.03	**44.94**
Mim2Ordc	65.98	20.39	69.82	39.31	57.17	50.51
Mim2ROCa	75.64	15.50	**73.86**	38.76	65.53	47.83
Mim2ROCb	75.57	15.65	73.21	40.51	66.42	49.90
Mim2ROCc	74.73	20.03	72.81	42.26	67.07	51.48
Mim3	76.08	19.44	73.38	39.98	68.02	54.60
Oracle	81.43	20.61	77.57	54.70	77.55	60.47

Let us see now the effect of working with balanced or non-balanced datasets on the accuracy and cost average(Table 3). Regarding the cost, we observe the same minima as in the overview. The greater increase w.r.t. the cost of the oracle is due to those datasets in which the *skew* acts positively over the majority class. The improvement of cost w.r.t. Mim3, which represents the approach by oversampling, is also meaningful. In the cases in which the *skew* acts positively over the minority class, the reduction of cost is also important for some methods (like Mim2Ordb) but not for all (for instance, Mim1Dira). Concerning accuracy, we do not observe a meaningful decrease. Note that the mimetic technique itself provides models whose accuracy is always lower than the accuracy of the oracle. Nevertheless, as expected, the success ratio in the case of minority class has been the most affected. Finally, the balanced situation shows an intermediate behaviour.

Table 4 shows the AUC of the models depending on the type of datasets. From these results, we can conclude that Mim1 obtains slightly better AUC than the rest of models. The differences are more important for the non-balanced datasets. Comparing and we can see that Mim2Roca is a good option if we look between a compromise between cost and AUC.

Table 4 AUC of all the models according to the experiment type.

Model	Balanced	Majority	Minority
Mim0a	0.811	0.722	0.722
Mim0b	0.812	0.727	0.727
Mim0c	0.804	0.726	0.726
Mim1Dira	0.813	0.731	0.732
Mim1Dirb	**0.814**	**0.733**	**0.733**
Mim1Dirc	0.811	0.731	0.731
Mim1Orda	0.813	0.731	0.732
Mim1Ordb	**0.814**	**0.733**	**0.733**
Mim1Ordc	0.811	0.731	0.731
Mim1Roca	0.813	0.731	0.732
Mim1Rocb	**0.814**	**0.733**	**0.733**
Mim1Rocc	0.811	0.731	0.731
Mim2Dira	0.796	0.707	0.721
Mim2Dirb	0.797	0.710	0.722
Mim2Dirc	0.786	0.706	0.717
Mim2Orda	0.758	0.688	0.693
Mim2Ordb	0.758	0.673	0.684
Mim2Ordc	0.738	0.668	0.684
Mim2ROCa	0.799	0.714	0.721
Mim2ROCb	0.799	0.712	0.722
Mim2ROCc	0.790	0.708	0.717
Mim3	0.807	0.716	0.723
Oracle	0.862	0.798	0.798

5. Conclusions

In this paper, we have presented several methods to derive a class threshold without training or validation data and we have analysed them theoretically and experimentally. As a result we can affirm that the introduced techniques are useful to reduce the costs of the model, being superior to the classical approach based on oversampling. So, not having data is not an obstacle if we want to adapt an existing model to a new cost context.

Theoretically, we have seen that the three approaches are similar if the probabilities are uniform. This is rarely the case. The approach based on ROC analysis is optimal, if the probabilities are well calibrated. However, this is not the case in many situations either. Consequently, the approach based on sorting the probabilities only assumes that the probabilities are reasonably well ordered and works well in this case. In general, this method seems to be better if no assumption is made on the quality of the probabilities.

From all the proposed configurations, Mim2 (a priori) is preferable and the reason can be found in the fact that the oracle is almost always better than its imitation (the mimetic model). So, the search of the threshold can be performed on the oracle more reliably. Secondly, from the three main types for the generation of the invented dataset, the results show that a) (a priori) and b) (balanced) are clearly better than c) (random). Hence, it is important to tune the proportion of classes which are labelled by the oracle. Although the differences between a) and b) are not high, they depend on the configuration and whether the dataset is balanced or not. Thirdly, regarding the method for determining the threshold, we can say that the direct method would work very well if the probabilities would be calibrated. Since this is not generally the case, we have to take the order of the probabilities into account as a more reliable thing and obtain the threshold according to this order (the sort method). This option seems to give the best results w.r.t. costs. Nonetheless, given that the threshold method is affected by the range in which the estimated probabilities can vary, we devised a method based on ROC analysis, and we proposed a threshold derivation based on the newly introduced NROC curves. Although they are worse on costs, they present a good compromise between cost, accuracy and AUC. The recommendation from the general results is that when the goal is to minimise the global cost the preferable configuration is to use the a priori method (i.e. Mim2), with the sort threshold and with the invented data in a balanced way (Mim2Ordb).

As future work it would be interesting to analyse the threshold derivation methods after performing a calibration. In this situation, we think that the method based on ROC analysis can be better than the other two. We have not tried this calibration for this paper since here we have considered a situation with almost no assumptions, in particular we do not have training or validation sets and, hence, we cannot calibrate the probabilities. An additional future work could be to find hybrid techniques between the Ord and ROC methods.

Acknowledgments

This work has been partially supported by the EU (FEDER) and the Spanish MEC under grant TIN 2004-7943-C04-02, ICT for EU-India Cross-Cultural Dissemination Project under grant ALA/95/23/2003/077-054, Generalitat Valenciana under grant GV06/301 and the Vic. de Innovación y Desarrollo (UPV) under project TAMAT ref 5771, SEIT-ANUIES scholarship and license commission of the Dirección General de Institutos Tecnológicos of México.

References

Abe, N., Zadrozny; B., Langford, J. (2004). An Iterative Method for Multi-class Cost-sensitive Learning. KDD’04, August 22–25, Seattle, Washington, USA.

Black C. L. ; Merz C. J. (1998). UCI repository of machine learning databases.

Blanco Vega, R. (2006). Extraction and constextualisation of comprehensible rules from black-box models. Ph.D. Dissertation. Universidad Politécnica de Valencia.

Blanco-Vega, R.; Hernández-Orallo, J.; Ramírez-Quintana M. J. (2004). Analysing the Trade-off between Comprehensibility and Accuracy in Mimetic Models. 7th International Conference on Discovery Science.

Bouckaert, R.; Frank, E. (2004). Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms, LNCS, Volume 3056, Page 3.

Breiman, L.; Friedman, J. H.; Olsen, R. A.; Stone, C. J. (1984) Classification and Regression Trees. Wadsworth International Group.

Chan, P.; Stolfo, S. (1998) Toward scalable learning with non-uniform class and cost distributions. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164–168.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost sensitive. In Procc of the 5th Int. Conf. on KDD and Data Mining, 155–164. ACM .

Domingos, P. (1997). Knowledge Acquisition from Examples Via Multiple Models. Proc. of the 14th Int. Conf. on Machine Learning, pp: 98-106.

Domingos, P. (1998). Knowledge Discovery Via Multiple Models. IDA, 2(1-4): 187-202.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. Proc. of the 17th Int. Joint Conf. on A.I.

Estruch, V.; Ferri, C.; Hernandez-Orallo, J.; Ramirez-Quintana. (2003). M.J. Simple Mimetic Classifiers, Proc. of the Third Int. Conf. on ML & DM, pp:156-171.

Lachiche, N.; Flach, P. A. (2003) Improving Accuracy and Cost of Two-class and Multi-class Probabilistic Classifiers Using ROC Curves. ICML 2003: 416-423.

Witten, I. H.; Frank, E. (2005). Data Mining: Practical ML tools with Java implementations. (Second Edition). Morgan Kaufmann.

New Algorithms for Optimizing Multi-Class Classifiers via ROC Surfaces

Kun Deng KDENG@CSE.UNL.EDU
Chris Bourke CBOURKE@CSE.UNL.EDU
Stephen Scott SSCOTT@CSE.UNL.EDU
N. V. Vinodchandran VINOD@CSE.UNL.EDU

Department of Computer Science & Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588-0115 USA

Abstract

We study the problem of optimizing a multi-class classifier based on its ROC hypersurface and a matrix describing the costs of each type of prediction error. For a binary classifier, it is straightforward to find an optimal operating point based on its ROC curve and the relative cost of true positive to false positive error. However, the corresponding multi-class problem (finding an optimal operating point based on a ROC hypersurface and cost matrix) is more challenging. We present several heuristics for this problem, including linear and nonlinear programming formulations, genetic algorithms, and a customized algorithm. Empirical results suggest that genetic algorithms fare the best overall, improving performance most often.

1. Introduction

We study the problem of re-weighting classifiers to optimize them for new cost models. For example, given a classifier optimized to minimize classification error on its training set, one may attempt to tune it to improve performance in light of a new cost model (say, a change in the ratio of true positive to false positive error). Equivalently, a change in the class distribution (the probability of seeing examples from each particular class) can be handled by modeling such a change as a change in cost model.

For two-class problems, the problem of finding the *optimal operating point* of a classifier given a ratio of true positive cost to false positive cost is straightforward via Receiver Operating Characteristic (ROC) analysis. ROC analysis takes a classifier h that outputs confidences in its predictions (i.e. a ranking classifier), and precisely describes the tradeoffs between true positive and false positive errors. By ranking all examples $x \in \mathcal{X}$ by their confidences $h(x)$ from largest to smallest (denoted $\mathcal{X} = \{x_1, \ldots, x_n\}$), one achieves a set of $n+1$ binary classifiers by setting thresholds $\{\theta_i = (h(x_i) + h(x_{i+1}))/2, 1 \leq i < n\} \cup \{h(x_1) - \epsilon, h(x_n) + \epsilon\}$ for some constant $\epsilon > 0$. Given a relative cost c of true positive error to false positive error and a validation set $\mathcal{X}$ of labeled examples, one can easily find the optimal threshold θ based on $\mathcal{X}$ and c (Lachiche & Flach, 2003). To do so, simply rank the examples in $\mathcal{X}$, try every threshold θ_i as described above, and select the θ_i minimizing the total cost of all errors on $\mathcal{X}$.

Though the binary case lends itself to straightforward optimization, working with multi-class problems makes things more difficult. A natural idea is to think of m-class ROC space having dimension $m(m-1)$. A point in this space corresponds to a classifier, with each coordinate representing the misclassification rate of one class into some other class[1]. According to Srinivasan (1999), the optimal classifier lies on the convex hull of these points. Given this ROC polytope, a validation[2] set $\mathcal{X}$, and an $m \times m$ cost matrix M with entries $c(C_j, C_k)$ (the cost associated with misclassifying a class C_j example as class C_k), Lachiche and Flach (2003) define the optimization problem as find-

[1] Assuming that cost is zero if the classification is correct, we need only $m(m-1)$ instead of m^2 dimensions.

[2] Lachiche and Flach ran their experiments with $\mathcal{X}$ as an independent validation set and with $\mathcal{X}$ as the original training set. They found little difference in their experimental results.

Appearing in *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, Pittsburgh, USA, 2006.

ing a weight vector $\vec{w} \geq \vec{0}$ to minimize[3]

$$\sum_{1 \leq j \leq m} \sum_{1 \leq k \leq m} p(C_j)\, r(C_j, C_k)\, c(C_j, C_k) \ , \qquad (1)$$

where $p(C_j)$ is the prior probability of class C_j, and $r(C_j, C_k)$ is the proportion of examples from $\mathcal{X}$ of actual class C_j that are predicted as class C_k. The predicted class of an example x is

$$\hat{y}_x = \underset{1 \leq i \leq m}{\operatorname{argmax}} \{w_i f(x, C_i)\} \ ,$$

where $f(x, C_i)$ is the classifier's confidence that example x is in class C_i.

No efficient algorithm is known to optimally solve (1), and Lachiche and Flach speculate that the problem is computationally hard. We present several new algorithms for this problem, including an integer linear programming relaxation, a sum-of-linear fractional functions (SOLFF) formulation, a direct optimization of (1) with a genetic algorithm and finally, a new custom algorithm based on partitioning $\mathcal{C}$ into *meta-classes*. In our experiments, our algorithms yielded several significant improvements both in minimizing classification error and minimizing cost.

The rest of this paper is as follows. In Section 2 we discuss related work and in Section 3 we discuss our approaches to this problem. We then experimentally evaluate our algorithms in Section 4 and conclude in Section 5.

2. Related Work

The success of binary ROC analysis gives hope that it may be possible to adapt similar ideas to multi-class scenarios. However, research efforts (Srinivasan, 1999; Hand & Till, 2001; Ferri et al., 2003; Lachiche & Flach, 2003; Fieldsend & Everson, 2005) have shown that extending current techniques to multi-class problems is not a trivial task. One key aspect to binary ROC analysis is that it is highly efficient to represent trade-offs of misclassifying one class into the other via binary ROC curves. In addition, the "area under the curve" (AUC) nicely characterizes the classifier's ability to produce correct rankings without committing to any particular operating point. Decisions can be postponed until a desired trade-off is required (e.g. finding the lowest expected cost).

Now consider the problem of classification in an m-class scenario. A natural extension from the binary case is to consider a multi-class ROC space as having dimension $m(m-1)$. A point in this space corresponds to a classifier with each coordinate representing the misclassification rate of one class into some other class. Following from Srinivasan (1999), the optimal classifier lies on the convex hull of these points.

Previous investigations have all shared this basic framework (Mossman, 1999; Srinivasan, 1999; Hand & Till, 2001; Ferri et al., 2003; Lachiche & Flach, 2003; Fieldsend & Everson, 2005; O'Brien & Gray, 2005). They differ, however, in the metrics they manipulate and in the approach they use to solve multi-class optimization problems. Mossman (1999) addressed the special case of three-class problems, focusing on the statistical properties of the volume under the ROC surface. This motivated the later work of Ferri et al. (2003), Lachiche and Flach (2003), and O'Brien and Gray (2005). Hand and Till (2001) extended the definition of two-class AUC by averaging pairwise comparisons. They used this new metric in simple, artificial data sets and achieved some success. Ferri et al. (2003) took a different approach in which they strictly followed the definition of two-class AUC by using "Volume Under Surface" (VUS). They were able to compute the bounds of this measure in a three-class problem by using Monte Carlo methods. However, it is not known how well this measure performs in more complex problems.

Fieldsend and Everson (2005), Lachiche and Flach (2003) and O'Brien and Gray (2005) developed algorithms to minimize the overall multi-class prediction accuracy and cost given some knowledge of a multi-class classifier. In particular, Fieldsend and Everson (2005) approximate the ROC Convex Hull (ROCCH) using the idea of "Pareto front." Consider the following formulation: let $R_{j,k}(\theta)$ be the misclassification rate of predicting examples from class j as class k. This is a function of some generalized parameter θ that depends on the particular classifiers. For example, θ may be a combination of a weight vector $\vec{w}$ and hypothetical cost matrix M. The goal is to find θ that minimizes $R_{j,k}(\theta)$ for all j, k with $j \neq k$. Consider two classifiers θ and ϕ. We say θ *strictly dominates* ϕ if all misclassification rates for θ are no worse than ϕ and at least one rate is strictly better. The set of all feasible classifiers such that no one is dominated by the other forms the *Pareto front.* Fieldsend and Everson (2005) present an evolutionary search algorithm to locate the Pareto front. This method is particularly useful when misclassification costs are not necessarily known.

More closely related to our work are the results of Lachiche and Flach (2003) and O'Brien and Gray (2005). Lachiche and Flach (2003) considered the case

[3] Assuming $c(C_j, C_j) = 0 \ \forall j$.

when the misclassification cost is known, and the goal is to find the optimal decision criterion that fits the training set. Recall that this can be solved optimally for the binary case. In particular, only one threshold θ is needed to make the decision for two-class problems. Since there are only $n+1$ possible thresholds for n examples, it is efficient enough to simply test all possibilities and select the one that gives the minimum average error (or cost). However, the situation is more complicated for multi-class problems. The main obstacle in the multi-class case is that the number of possible classification assignments grows exponentially in the number of instances: $\Omega(m^n)$.

Lachiche and Flach (2003) formulated the multi-class problem as follows. Suppose the multi-class learning algorithm will output a positive, real-valued function $f : \{x_1, \ldots, x_n\} \times \{C_1, \ldots, C_m\} \to \mathbb{R}^+$. Here, $f(x_i, C_j)$ gives the confidence that example x_i belongs to class C_j. The decision criterion simply assigns example x_i to the class with maximum score. Reweighting the classes involves defining a nonnegative weight vector $\vec{w} = (w_1, w_2, \ldots, w_m)$, and predicting the class for an example x as

$$h(x) = \operatorname*{argmax}_{1 \le j \le m} \{w_j f(x, C_j)\} \ .$$

It should be apparent that $\vec{w}$ has only $m-1$ degrees of freedom, so we can fix $w_1 = 1$.

Lachiche and Flach (2003) used a hill climbing or sequential optimization heuristic to find a good weight vector $\vec{w}$. In particular, they took advantage of the fact that the optimal threshold for the two-class problem can be found efficiently. For each coordinate in the weight vector, they mapped the problem to a binary problem. The algorithm starts by assigning $w_1 = 1$ and all other weights 0. It then tries to decide the weight for one class at a time as follows. Let $\mathcal{X}$ be the set of training examples and let p be the current class for which we want to assign a "good" weight w_p. Then the set of possible weights for w_p is

$$\left\{ \frac{\max_{j \in \{1,\ldots,p-1\}} f(x, C_j)}{f(x, C_p)} \middle| \, x \in \mathcal{X} \right\}.$$

It is not difficult to see that at any stage there are at most $\mathcal{O}(|\mathcal{X}|)$ possible weights that can influence the prediction. Thus choosing the optimal weight in this setting can be easily achieved by checking all possibilities. Overall, their algorithm runs in time $\Theta(mn \log n)$. Though there is no guarantee that this approach can find an optimal solution, they gave empirical results that it works well for optimizing 1BC, a logic-based Bayes classifier (Lachiche & Flach, 1999).

Although only briefly mentioned in Lachiche and Flach (2003), this ROC thresholding technique is quite extensible to cost-sensitive scenarios. O'Brien and Gray (2005) investigated the role of a cost matrix in partitioning the estimated class probability space and as a replacement for the weights. Assuming that M is a misclassification cost matrix, an optimal decision criterion would be

$$h(x) = \operatorname*{argmin}_{1 \le k \le m} \left\{ \sum_{1 \le j \le m} c(C_j, C_k)\, \hat{p}(x, C_j) \right\} \ .$$

If $\hat{p}(x, C_j)$ is a good probability estimate of example x belonging to class C_j, this prediction results in the lowest expected cost. However, if $\hat{p}(x, C_j)$ is not an accurate probability estimate, then to ensure optimality, the cost matrix M has to be altered accordingly. Thus the cost matrix M plays a similar role as Lachiche and Flach's weight vector in defining the decision boundary in estimated probability space. O'Brien and Gray (2005) defined several standard operations to manipulate the cost matrix M and proposed the use of a greedy algorithm to find the altered cost matrix (called a *boundary matrix*).

3. Our Contributions

In this section, we first present new mathematical programming formulations. In particular, we reformulate the objective function (1) given by Lachiche and Flach (2003) as a relaxed integer linear program as well as give a formulation that is a sum of linear fractional functions. We also describe a new heuristic algorithm approach, MetaClass. Finally (in Section 4) we present experimental results from these formulations. We give evidence that the objective function landscape for this problem is highly discontinuous and thus more amenable to global optimization methods such as genetic algorithms.

3.1. Mathematical Programming Formulations

3.1.1. Relaxed Integer Linear Program

We start by reformulating (1) as follows:

$$\operatorname*{minimize}_{\vec{w}, \vec{I}} \left\{ \sum_{1 \le j \le m} \frac{p(C_j)}{|C_j|} \sum_{1 \le k \le m} c(C_j, C_k) \sum_{x_i \in C_j} I_{i,k} \right\} , \tag{2}$$

where $C_j \subseteq \mathcal{X}$ is the set of instances of class j, $p(C_j)$ is the prior probability of class C_j, $c(C_j, C_k)$ is the cost of misclassifying an example from class C_j as C_k, and

$$I_{i,k} = \begin{cases} 1 & \text{if } w_k f(x_i, C_k) \ge w_l f(x_i, C_l), l \ne k \\ 0 & \text{otherwise.} \end{cases}$$

Here we assume $c(C_j, C_j) = 0$ for all C_j. Formalizing this as a constrained optimization problem, we want to minimize (2) subject to

$$I_{i,j} w_j f(x_i, C_j) = I_{i,j} \max_{1 \le k \le m}\{w_k f(x_i, C_k)\} \quad (3)$$
$$\textstyle\sum_{1 \le j \le m} I_{i,j} = 1 \quad (4)$$
$$I_{i,j} \in \{0, 1\} \quad (5)$$
$$w_j \ge 0 \quad (6)$$

where each constraint holds for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. Equation (3) allows only the class that has the max value of $w_k f(x_i, C_k)$ to be indicated by $\vec{I}$ to be the predicted class of example x_i and (4) forces exactly one class to be predicted per example x_i.

We can change the optimization problem in two ways to get an equivalent problem. First, we change the "=" in (3) to "≥". Second, we can relax (5) to be $I_{i,j} \in [0, 1]$. Note that (3) (even when amended with "≥") will only be satisfied if $I_{i,j} = 0$ for all C_j that don't maximize the RHS of (3). Thus, so long as we never have $w_k f(x_i, C_k) = w_{k'} f(x_i, C_{k'})$ for some $k \ne k'$, the relaxation is equivalent to the original problem. Further, even if there is such a tie for classes C_k and $C_{k'}$, it will not be an issue if the corresponding entries in the cost matrix are different, since an optimal solution will set $I_{i,k} = 1$ and $I_{i,k'} = 0$ if $c(C_j, C_k) < c(C_j, C_{k'})$. The potential problem of both $w_k f(x_i, C_k) = w_{k'} f(x_i, C_{k'})$ and $c(C_j, C_k) = c(C_j, C_{k'})$ is fixed by (after optimizing) checking for any $I_{i,k} \notin \{0, 1\}$ and arbitrarily choosing one to be 1 and the rest 0. Note that since there is a tie in this case, the prediction can go either way and the weight vector $\vec{w}$ returned is still valid.

Everything except (3) is linear. We now reformulate it. First, for each $i \in \{1, \ldots, n\}$, we substitute γ_i for $\max_{1 \le k \le m}\{w_k f(x_i, C_k)\}$:

$$I_{i,j} w_j f(x_i, C_j) \ge \gamma_i I_{i,j} \quad (7)$$
$$w_k f(x_i, C_k) \le \gamma_i \ , \quad (8)$$

for all $i \in \{1, \ldots, n\}$ and $j, k \in \{1, \ldots, m\}$ where each γ_i is a new variable. Obviously (8) is a linear constraint, but (7) is not even quasiconvex (Boyd & Vandenberghe, 2004). The complexity of this optimization problem motivates us to reformulate it a bit further.

Let us assume that $f(x_i, C_k) \in (0, 1]$ (e.g. if $f(\cdot, \cdot)$ are probability estimates from naïve Bayes or logistic regression). Now we can optimize (2) subject to:

$$\gamma_i - w_j f(x_i, C_j) + I_{i,j} \le 1 \quad (9)$$
$$\gamma_i \ge w_j f(x_i, C_j) \quad (10)$$
$$\textstyle\sum_{1 \le j \le m} I_{i,j} = 1 \quad (11)$$
$$I_{i,j} \in \{0, 1\} \quad (12)$$
$$w_j \ge 0 \quad (13)$$

for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$.

So long as $w_j f(x_i, C_j) \in (0, 1]$ and $I_{i,j} \in \{0, 1\}$ for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, this is another equivalent optimization problem, this time a $\{0, 1\}$ integer linear program. Unfortunately, we cannot relax (12) to $I_{i,j} \in [0, 1]$ as we did before to get an equivalent problem. But we still use the relaxation as a linear programming heuristic. To help avoid overfitting, we also add a linear regularization term to (2):

$$\underset{\vec{w}, \vec{I}}{\text{minimize}} \left\{ \sum_{1 \le j \le m} \frac{p(C_j)}{|C_j|} \sum_{1 \le k \le m} c(C_j, C_k) \sum_{x_i \in C_j} I_{i,k} \right. \\ \left. + \eta \|\vec{w} - \vec{1}\|_1 \right\} \quad (14)$$

where $\|\cdot\|_1$ is the 1-norm, $\vec{1}$ is the all-1s vector, and η is a parameter. This regularizer penalizes large deviations from the original classifier $f(\cdot, \cdot)$.

3.1.2. Sum of Linear Fractional Function Formulation

Another formulation comes from changing how predictions are made from deterministic to probabilistic. In this prediction model, given a new example x to predict on, first compute $w_k f(x, C_k)$ for each $k \in \{1, \ldots, m\}$. Then predict class C_k for example x with probability

$$\frac{w_k f(x, C_k)}{\sum_{1 \le \ell \le m} w_\ell f(x, C_\ell)} \ .$$

Assuming a uniform distribution over the data set, the expected cost of this predictor is

$$\sum_{1 \le j \le m} \frac{p(C_j)}{|C_j|} \sum_{1 \le k \le m} c(C_j, C_k) \sum_{x_i \in C_j} \varphi(i, k) \ , \quad (15)$$

where

$$\varphi(i, k) = \frac{w_k f(x_i, C_k)}{\sum_{1 \le \ell \le m} w_\ell f(x_i, C_\ell)}$$

subject to $w_k \ge 0$ for all $k \in \{1, \ldots, m\}$. We now have eliminated the variables $I_{i,k}$ and their integer constraints. However, we now have a nonlinear objective function in (15). Each individual term of the summation of (15) is a *linear fractional function*, which is quasiconvex and quasiconcave, and thus it is efficiently solvable optimally. However, the *sum of linear* *fractional functions* (SOLFF) problem is known to be hard (Matsui, 1996) and existing algorithms for this problem are inappropriate (they either restrict to few terms in the summation or to low-dimensional vectors). Instead, we apply a genetic algorithm to directly optimize (15).

3.2. The MetaClass Heuristic Algorithm

In addition to the linear programming formulations, we present a new algorithm that we call MetaClass (Algorithm 1). This algorithm is similar to that of Lachiche and Flach (2003) in that we reduce the multi-class problem to a series of two-class problems. However, we take what can be considered a top-down approach while the algorithm of Lachiche and Flach (2003) can be considered bottom-up. Moreover, MetaClass has a faster time complexity. The output of the algorithm is a decision tree with each internal node labeled by two *metaclasses* and a threshold value. Each leaf node is labeled by one of the classes in the original problem. At the root, the set of all classes is divided into two metaclasses. The criterion for this split may be based on any statistical measure, but for simplicity, experiments were performed by splitting classes so that each metaclass would have roughly the same number of examples. For each metaclass, our algorithm defines confidence functions f_1 and f_2 for each instance, which are simply the sum of the confidences of the classes in $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The ratio $F = \frac{f_1}{f_2}$ is used to find a threshold θ. We find θ by sorting the instances according to F and choose a threshold that minimizes error. (This threshold will be the average of $F(x_i)$ and $F(x_{i+1})$ for some instance x_i.) The situation for cost is slightly more complicated since it is not clear *which* class in the metaclass an example is misclassified as. Instead, we use the *average* cost of misclassifying instances into metaclasses in $\mathcal{C}_1$ and $\mathcal{C}_2$. In this case, a threshold is chosen that minimizes the overall cost. We recursively perform this procedure on the two metaclasses until there is only a single class, at which point a leaf is formed. The MetaClass algorithm is presented as Algorithm 1.

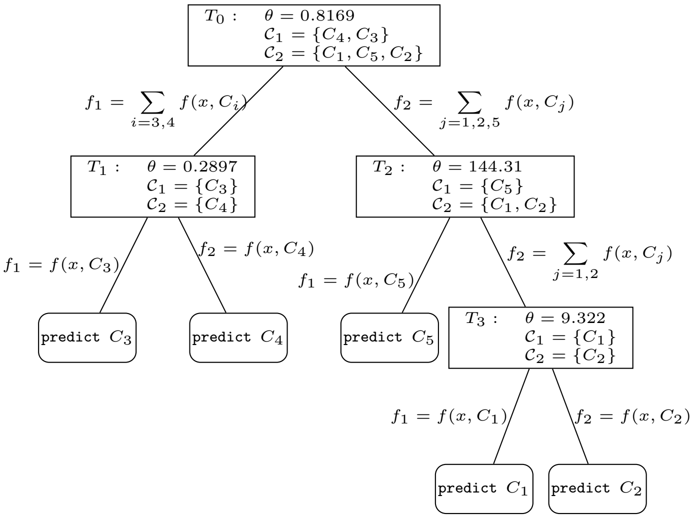

T_0 : $\theta = 0.8169$
$\mathcal{C}_1 = \{C_4, C_3\}$
$\mathcal{C}_2 = \{C_1, C_5, C_2\}$
$f_1 = \sum_{i=3,4} f(x, C_i)$
$f_2 = \sum_{j=1,2,5} f(x, C_j)$
T_1 : $\theta = 0.2897$
$\mathcal{C}_1 = \{C_3\}$
$\mathcal{C}_2 = \{C_4\}$
T_2 : $\theta = 144.31$
$\mathcal{C}_1 = \{C_5\}$
$\mathcal{C}_2 = \{C_1, C_2\}$
$f_1 = f(x, C_3)$
$f_2 = f(x, C_4)$
$f_1 = f(x, C_5)$
$f_2 = \sum_{j=1,2} f(x, C_j)$
predict C_3
predict C_4
predict C_5
T_3 : $\theta = 9.322$
$\mathcal{C}_1 = \{C_1\}$
$\mathcal{C}_2 = \{C_2\}$
$f_1 = f(x, C_1)$
$f_2 = f(x, C_2)$
predict C_1
predict C_2

Figure 1. Example run of MetaClass on Nursery, a 5-class problem.

```
Input  : A set of instances, $\mathcal{X} = \{x_1, \ldots, x_n\}$; a
         set of classes, $\mathcal{C} = \{C_1, \ldots, C_m\}$; a
         learned confidence function
         $f : \mathcal{X} \times \mathcal{C} \to \mathbb{R}^+$ and a tree node $T$
Output : A decision tree with associated weights.
Split $\mathcal{C}$ into two meta-classes $\mathcal{C}_1, \mathcal{C}_2$
foreach Instance $x_i \in \mathcal{X}$ do
    $f_1(x_i, \mathcal{C}_1) = \sum_{C_j \in \mathcal{C}_1} f(x_i, C_j)$
    $f_2(x_i, \mathcal{C}_2) = \sum_{C_j \in \mathcal{C}_2} f(x_i, C_j)$
    $F(x_i) = f_1(x_i)/f_2(x_i)$
end
Sort instances according to $F$
Compute a threshold $\theta$ that minimizes error/cost
w.r.t. $F$
Label $T$ with $\theta, \mathcal{C}_1, \mathcal{C}_2$
Create two children of $T$: $T_{\text{left}}, T_{\text{right}}$
Split $\mathcal{X}$ into two classes, $\mathcal{X}_1, \mathcal{X}_2$ according to $\mathcal{C}_1, \mathcal{C}_2$
Recursively perform this procedure on $\mathcal{X}_1, \mathcal{C}_1, T_{\text{left}}$
and $\mathcal{X}_2, \mathcal{C}_2, T_{\text{right}}$ until $|\mathcal{C}| = 1$
```

Algorithm 1: *MetaClass*

Figure 1 gives an example of a tree built by the MetaClass algorithm on the UCI (Blake & Merz, 2005) data set Nursery, a 5 class data set. At the root, the classes are divided into two metaclasses each with about the same number of examples represented in their respective classes. In this case, the threshold $\theta = 0.8169$ favors the sum of confidences in metaclass $\mathcal{C}_1 = \{C_4, C_3\}$ as an optimal weight.

Predictions for a new example y are made as follows. Starting at the root node, we traverse the tree towards a leaf. At each node T we compute the sum of confidences of y with respect to each associated metaclass. We traverse left or right down the tree depending on whether $f_1/f_2 \geq \theta$. When a leaf is reached, a final class prediction is made.

The number of nodes created by MetaClass is $\Theta(m)$, where m is the number of classes. At each node, the most complex step is sorting at most n instances according to the confidence ratio. Thus, the overall performance is bounded by $\Theta(n \log n \log m)$. Since for most applications, $n \gg m$, we may consider its actual running time to simply be $\mathcal{O}(n \log n)$. Classification is also efficient. At each node we compute a sum over an exponentially shrinking number of classes. The overall number of operations is thus

$$\sum_{i=0}^{\log(m)-1} \frac{m}{2^i} ,$$

which is linear in the number of classes: $\Theta(m)$.

4. Experimental Results

The following experiments were performed on 25 standard UCI data sets (Blake & Merz, 2005), using Weka's naïve Bayes (Witten et al., 2005) as the baseline classifier. We ran experiments evaluating improvements both in classification accuracy and total cost. We used 10-fold cross validation for error rate experiments (Table 1). For the cost experiments of Table 2, 10-fold cross validations were performed on 10 different cost matrices for each data set. Costs were integer values between 1 and 10 assigned uniformly at random. Costs on the diagonal were set to zero. The average cost per test instance was reported for each experiment. Table 2 gives the average cost over all 100 experiments per data set, per algorithm.

In both tables, for each data set m denotes the number of classes and NB indicates our baseline classifier's performance. For comparison, we have included wins and losses (and significance) for the algorithms reported by Lachiche and Flach (2003) and O'Brien and Gray (2005). Raw numbers are omitted since these results are not directly comparable to ours: in addition to being based on different data partitions, the results from Lachiche and Flach (2003) were from an optimization run on the base classifier 1BC (Lachiche & Flach, 1999). Moreover, the results in O'Brien and Gray (2005) (here, we have used one of their best formulations, "column multiply") pruned classes that did not have "sufficient" representation. Furthermore, Lachiche and Flach (2003) did not consider cost-sensitive experiments. Thus, the results in Table 2 for Lachiche & Flach are taken from the implementation and results reported by O'Brien and Gray.

The results of our experiments can be found in the last four columns of each table. Here, MC is the MetaClass algorithm (Algorithm 1). LP is a linear programming algorithm (MOSEK ApS, 2005) on Equation (14) with $\eta = 10^{-6}$. The first GA is the Sum of Linear Fractional Functions formulation (Equation (15)) using a genetic algorithm. The final column is a genetic algorithm performed on Equation (1). Both GA implementations were from Abramson (2005). Parameters for both used the default Matlab settings with a population size of 20, a maximum of 200 generations and a crossover fraction of 0.8. The algorithm terminates if no change is observed in 100 continuous rounds. In addition, the mutation function of Abramson (2005) is guaranteed to only generate feasible solutions (in our case, all weights must be nonnegative). Upon termination, a direct pattern search is performed using the best solution from the GA.

Data for some entries were not available and are denoted "n/a" (either the source did not report results or, in the case of our experiments, data sets were too large for Matlab). Therefore, for comparison it is important to note the *ratio* of significant wins to significant losses rather than merely total wins or losses. For all columns, **bold** entries indicate a significant difference to the baseline with at least a 95% confidence according to a Student's-t method. The overall best classifier for each data set is underlined.

Regarding classification error, in every case each algorithm showed some significant performance improvements. With the exception of LPR, all algorithms were competitive with no clear overall winner. However, Table 3.2 does, in fact, show a clear winner when costs are non-uniform. The success when using a GA on Equation (1) gives evidence that the objective function surface is likely to be very rough with many local minimums (it is certainly discontinuous given the use of the argmax function). This also may explain why other methods did not perform as well. The GA is searching globally; in contrast all other methods (including Lachiche and Flach (2003)) search locally. Even the integer linear programming relaxation, which in general has a good track record, came up short.

5. Conclusion & Future Work

When the cost model or class distribution of a learning problem deviates from the conditions under which a classifier f was trained, one may wish to re-optimize f. For two-class problems, it is well-known how to do this via ROC analysis, but the multi-class problem is more challenging. We presented multiple algorithms for the multi-class version of this problem and empirically showed their competitiveness. Direct optimization by a genetic algorithm was particularly effective.

Future work includes answering the question posed by Lachiche and Flach (2003): is this optimization problem computationally intractable? Assuming it is, then a more tractable and useful special case of this problem may be when the number of classes is restricted to a constant. In particular, can we find a provably optimal algorithm when the number of classes is 3?

Acknowledgments

The authors would like to thank Nicolas Lachiche for helpful correspondence. We also thank anonymous reviewers for helpful suggestions and feedback.

References

Abramson, M. A. (2005). Genetic algorithm and direct search toolbox. See http://www.mathworks.com/.

Bengio, S., Mariéthoz, J., & Keller, M. (2005). The expected performance curve. *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning* (pp. 9–16).

Blake, C., & Merz, C. (2005). UCI repository of machine learning databases.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Ferri, C., Hernández-Orallo, J., & Salido, M. (2003). Volume under the ROC surface for multi-class problems. *ECAI 2003* (pp. 108–120).

Fieldsend, J., & Everson, R. (2005). Formulation and comparison of multi-class ROC surfaces. *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning* (pp. 41–48).

Hand, D., & Till, R. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, *45*, 171–186.

Lachiche, N., & Flach, P. (1999). 1BC: A first-order bayesian classifier. *Proceedings of the 9th International Workshop on Inductive Logic Programming* (pp. 92–103).

Lachiche, N., & Flach, P. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. *Proceedings of the 20th International Conference on Machine Learning* (pp. 416–423).

Matsui, T. (1996). NP-hardness of linear multiplicative programming and related problems. *Journal of Global Optimization*, *9*, 113–119.

MOSEK ApS (2005). The MOSEK optimization tools version 3.2. See http://www.mosek.com/.

Mossman, D. (1999). Three-way ROCs. *Medical Decision Making*, 78–89.

O'Brien, D. B., & Gray, R. M. (2005). Improving classification performance by exploring the role of cost matrices in partitioning the estimated class probability space. *International Conference on Machine Learning, ICML, Workshop on ROC Analysis in Machine Learning* (pp. 79–86).

Srinivasan, A. (1999). *Note on the location of optimal classifiers in n-dimensional ROC space* (Technical Report PRG-TR-2-99). Oxford University Computing Laboratory, Oxford.

Witten, I. H., et al. (2005). Weka machine learning toolbox. See www.cs.waikato.ac.nz/ml/weka/.

Table 1. Error Rates. L&F are from Lachiche and Flach (2003) and O&G are from O'Brien and Gray (2005). W indicates that the optimized classifier performed better than its baseline and L indicates a performance loss. NB is our baseline. **Bold** font denotes a significant difference to the baseline with at least 95% confidence. The overall best classifier among our algorithms is underlined.

						LP	GA	GA
Data Set	m	L&F	O&G	NB	MC	Eq. (14)	Eq. (15)	Eq. (1)
Audiology	24	**W**	**L**	0.3095	**0.4816**	0.2869	**0.2826**	0.2872
Bridges 2 (material)	3	**W**	**W**	0.1582	**0.2436**	**0.2709**	0.1764	**0.2136**
Bridges 2 (rel-l)	3	W	tie	0.3164	0.3336	0.3081	0.3355	0.3455
Bridges 2 (span)	3	tie	n/a	0.2227	0.2245	0.2427	**0.4018**	n/a
Bridges 2 (type)	6	W	**W**	0.4564	0.4564	0.4654	0.4664	0.4655
Bridges 2 (t-or-d)	2	W	n/a	0.1755	0.1955	0.1754	0.1936	**0.2500**
Car	4	**W**	**W**	0.1464	**0.1267**	**0.1336**	0.1724	**0.1209**
Post-Op	3	W	n/a	0.4949	0.4881	0.4948	0.4990	0.4908
Horse-colic (code)	3	W	**L**	0.3173	0.3179	0.3172	0.2931	0.3120
Horse-colic (surgical)	2	tie	n/a	0.2089	**0.1710**	0.2089	**0.1791**	**0.1791**
Horse-colic (site)	63	**W**	**L**	0.7634	0.7770	0.7634	0.7444	0.7661
Horse-colic (subtype)	2	**W**	**W**	0.0027	0.0027	0.0027	0.0027	0.0027
Horse-colic (type)	8	W	**L**	0.0409	0.0272	0.0409	0.0327	0.0354
Credit	2	L	n/a	0.2490	0.2570	0.2490	**0.2720**	0.2580
Dermatology	6	L	tie	0.0273	0.0192	0.0273	0.0273	0.0165
Ecoli	8	L	tie	0.1516	0.1608	0.1545	0.1640	**0.1368**
Flags	8	L	tie	0.3761	0.4213	0.3707	0.3705	0.3808
Glass	7	L	**L**	0.5236	0.5472	0.5235	**0.4260**	0.4959
Mushroom	2	**W**	n/a	0.0420	**0.0181**	**0.0374**	**0.0192**	**0.0178**
Nursery	5	**W**	**W**	0.0968	**0.0849**	n/a	n/a	**0.0847**
Image Segmentation	7	L	W	0.1974	**0.1485**	0.1974	**0.1260**	**0.1727**
Solar Flare (common)	8	**W**	**W**	0.2364	**0.1745**	0.2364	**0.1708**	**0.1980**
Solar Flare (moderate)	6	**W**	n/a	0.0732	**0.0356**	0.0731	**0.0338**	**0.0507**
Solar Flare (severe)	3	**W**	n/a	0.0282	**0.0085**	**0.0234**	**0.0047**	**0.0207**
Vote	2	**L**	n/a	0.0965	0.1081	0.0964	0.0942	0.0965
Win/Loss		15/8	7/6	–	11/12	6/5	13/11	15/9
Significant Win/Loss		9/1	6/4	–	8/2	3/1	7/3	8/3

Table 2. Costs. L&F and O&G are both from O'Brien and Gray (2005).

Data Set	m	L&F	O&G	NB	MC	LP (14)	GA (15)	GA (1)
Audiology	24	**L**	**L**	1.7720	**2.7935**	1.6386	1.7245	**1.5194**
Bridges 2 (material)	3	**L**	L	0.8611	**1.3542**	**1.4725**	1.0059	**1.2181**
Bridges 2 (rel-l)	3	**L**	**L**	1.9355	1.9441	1.9003	**2.5553**	1.9875
Bridges 2 (span)	3	n/a	n/a	1.1872	**1.6153**	1.2930	**1.9934**	**1.4491**
Bridges 2 (type)	6	**L**	**L**	2.4585	2.3160	2.6020	**2.7605**	2.5220
Bridges 2 (t-or-d)	2	n/a	n/a	0.9655	0.8946	0.9568	1.0794	0.9705
Car	4	**W**	**W**	0.8484	0.8898	0.8074	**1.6665**	**0.6523**
Post-Op	3	n/a	n/a	2.9242	3.0450	**2.9993**	**3.6611**	**2.7057**
Horse-colic (code)	3	**W**	**W**	1.7915	1.7243	1.7863	1.7950	1.6989
Horse-colic (surgical)	2	n/a	n/a	1.3788	**1.0060**	1.3890	**1.6364**	**1.0629**
Horse-colic (site)	63	**L**	**L**	4.0892	4.2372	4.1084	**4.2647**	4.0809
Horse-colic (subtype)	2	**W**	**W**	0.0114	0.5741	0.0114	0.0113	0.0114
Horse-colic (type)	8	**W**	**W**	0.2225	0.1447	0.2172	0.1704	0.1970
Credit	2	n/a	n/a	1.3203	**1.0444**	**1.3951**	**2.1906**	**1.0531**
Dermatology	6	**L**	L	0.1744	0.1105	0.1564	0.1775	0.1147
Ecoli	8	**L**	**L**	0.8105	0.8514	0.8766	**1.2116**	0.7678
Flags	8	**W**	**W**	2.1590	2.2803	2.1408	2.3013	**1.9888**
Glass	7	**L**	**L**	3.1308	2.9330	3.1301	**3.4910**	**2.6720**
Mushroom	2	n/a	n/a	0.1930	**0.0994**	**0.1784**	**0.1262**	**0.1031**
Nursery	5	**W**	**W**	0.5565	0.6651	n/a	n/a	**0.4634**
Image Segmentation	7	**W**	**W**	1.0855	**0.8416**	1.0855	1.0989	**0.8952**
Solar Flare (common)	8	**W**	**W**	1.3080	**4.1595**	**1.3199**	**1.1622**	**0.9844**
Solar Flare (moderate)	6	n/a	n/a	0.4644	**5.6085**	0.4628	**0.2749**	**0.2652**
Solar Flare (severe)	3	n/a	n/a	0.1682	**6.0877**	0.1556	**0.0800**	**0.1070**
Vote	2	n/a	n/a	0.4510	**0.3770**	0.4510	**0.5346**	0.4577
Win/Loss		8/8	8/8	–	11/14	12/9	6/17	19/6
(Sig)		8/8	8/6	–	5/6	1/4	4/11	13/2

A Framework for Comparative Evaluation of Classifiers in the Presence of Class Imbalance

William Elazmeh[†] WELAZMEH@SITE.UOTTAWA.CA
Nathalie Japkowicz[†] NAT@SITE.UOTTAWA.CA
Stan Matwin[†‡] STAN@SITE.UOTTAWA.CA
[†] School of Information Technology and Engineering, University of Ottawa
[‡] Institute of Computer Science, Polish Academy of Sciences, Poland

Abstract

Evaluating classifier performance with ROC curves is popular in the machine learning community. To date, the only method to assess confidence of ROC curves is to construct ROC bands. In the case of severe class imbalance, ROC bands become unreliable. We propose a generic framework for classifier evaluation to identify the confident segment of an ROC curve. Confidence is measured by Tango's 95%-confidence interval for the difference in classification errors in both classes. We test our method with severe class imbalance in a two-class problem. Our evaluation favors classifiers with low numbers of classification errors in both classes. We show that our evaluation method is more confident than ROC bands when faced with severe class imbalance.

Table 1. The statistical proportions in a confusion matrix.

	Predicted +	Predicted -	total
Class +	a (q_{11})	b (q_{12})	a+b
Class -	c (q_{21})	d (q_{22})	c+d
total	a+c	b+d	n

1. Motivation

Recently, the machine learning community has increased the focus on classifier evaluation. Evaluation schemes that compute accuracy, precision, recall, or F-score have been shown to be insufficient or inappropriate (Ling et al., 2003; Provost & Fawcett, 1997). Furthermore, the usefulness of advanced evaluation measures, like ROC curves (Cohen et al., 1999; Provost & Fawcett, 1997; Swets, 1988) and cost curves (Drummond & Holte, 2000; Drummond & Holte, 2004), deteriorates in the presence of a limited number of positive examples. The need for confidence in classifier evaluation in machine learning has lead to the construction of ROC confidence bands. Methods in (Macskassy et al., 2005; Macskassy & Provost, 2004) construct ROC bands by computing confidence intervals for points along the ROC curve. These methods are either parametric (making assumptions of data distributions), or non-parametric and rely on carefully crafted sampling methods. When faced with severe class imbalance, sampling methods become unreliable, especially when the data distribution is unknown (Macskassy & Provost, 2004). In fact, with severe imbalance, the entire issue of evaluation becomes a serious challenge even when making assumptions of data distributions (Drummond & Holte, 2005). In contrast, biostatistical and medical domains impose strong emphasis on error estimates, interpretability of prediction schemes, scientific significance, and confidence (Motulsky, 1995) whilst machine learning evaluation measures fail to provide such guarantees. Consequently, the usefulness of some machine learning algorithms remains inadequately documented and unconvincingly demonstrated. Thus, despite their interest in using learning algorithms, biostatisticians remain skeptical of their evaluation methods and continue to develop customized statistical tests to measure characteristics of interest. Our work adopts Tango's test (Tango, 1998) from biostatistics in an attempt to provide confidence in classifier evaluation. Tango's test is a non-parametric confidence test designed to measure the difference in binomial proportions in paired data. Computing the confidence based on the positive or negative rates (using a or d of the confusion matrix in

Appearing in Proceedings of the third Workshop on ROC Analysis in Machine Learning, Pittsburgh, USA, 2006. Copyright 2006 by the author(s)/owner(s).

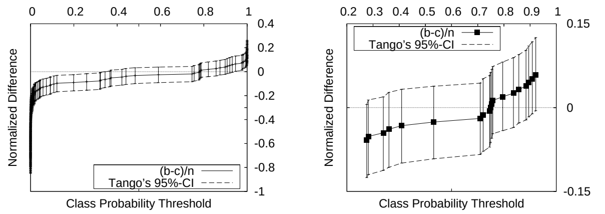


Figure 1. $\frac{b-c}{n}$ and Tango's 95%-confidence intervals for ROC points. Left: all the ROC points. Right: only confident ROC points whose Tango's intervals contain 0.

table 1) can be influenced by class imbalance in favor of the majority class. Alternatively, applying a statistical significance test to those entries (b or c) that resist such influence may provide a solution. Hence, to counter the class imbalance, we favor classifiers with similar normalized number of errors in both classes, rather than similar error rates to avoid the imbalance.

In this paper; (1) we propose a framework for classifier evaluation that identifies confident points along an ROC curve using a statistical confidence test. These points form a confident ROC segment to which we recommend restricting the evaluation. (2) Although our framework can be applied to any data, this work focuses on the presence of severe imbalance where ROC bands, ROC curves and AUC struggle to produce meaningful assessments. (3) We produce a representation of classifier performance based on the average difference in classification errors and the Area Under the Confident ROC Segment. We present experimental results that show the effectiveness of our approach in severe imbalanced situations compared to ROC bands, ROC curves, and AUC. Having motivated this work, subsequent sections present discussions of classification error proportions in both classes (in section 2), our evaluation framework (in section 3), and our experimental results (in section 4) followed by conclusions and future work (in section 5). We review Tango's statistical test of confidence in appendix A (section 6).

2. Difference in Classification Errors

Common classifier performance measures in machine learning estimate classification accuracy and/or errors. ROC curves provide a visualization of a possible tradeoff between accuracy and error rates for a particular class. For the confusion matrix presented in table 1 on page 1, the ROC curve for the class + plots the true positive rate $\frac{a}{a+b}$ against the false positive rate $\frac{c}{c+d}$. When the number of positive examples is significantly lower than the number of negative examples, the row totals $a+b << c+d$. When changing the class probability threshold, the rate of change in the true positive rate climbs faster with each example than that of the false positives (due to using c and d). This inconsistent rate of change gives the majority class $(-)$ a clear advantage in the rates calculated for the ROC curve. Ideally, a classifier classifies both classes proportionally, but due to the severe imbalance, comparing the rates of accuracy and/or errors on both classes does not evaluate proportionally. We propose to favor the classifier that performs with similar number of errors in both classes to eliminate the use of the number of correctly classified examples (a and d) in the evaluation to avoid a large portion of examples in the majority class. In fact, our approach favors classifiers that have lower difference in classification errors in both classes, $\frac{b-c}{n}$. Furthermore, we normalize entries in the confusion matrix by dividing by the number of examples n so the difference $\frac{b-c}{n}$ remains within $[-1,+1]$.

ROC curves are generated by classifying examples while increasing class probability threshold T. When $T = 0$, all data examples are classified as +, thus, $a = |+|$ (the number of positives), $b = 0$, $c = |-|$, $d = 0$, and $\frac{b-c}{n} \in [-1, 0]$. Similarly, for $T = 1$, all examples are classified as $-$, then, $a = 0$, $b = |+|$, $c = 0$, $d = |-|$, and $\frac{b-c}{n} \in [0, +1]$. In fact, these two extreme negative and positive values of $\frac{b-c}{n}$ depend on class distributions in the data. Within these two extremes, $\frac{b-c}{n}$ exhibits a monotone behavior as the threshold varies from 0 to 1. This is illustrated in figure 1. For each threshold value $T := 0$ to 1, the classification produces a confusion matrix a, b, c, d. Initially, a and c are at their maximum values, while b and d are 0. As T increases, examples are classified in any combination of three possibilities; (1) c decreases when false positives become correctly classified, (2) b increases when true positives become misclassified, (3) or, b and c remain unchanged because examples are correctly classified. Since c never increases, b never decreases, and n is constant, then $\frac{b-c}{n}$ exhibits a monotone non-decreasing behavior for a classifier on a set of data. Our evaluation method computes Tango's 95%-confidence intervals for $\frac{b-c}{n}$ for ROC points. Those points whose confidence intervals include the value zero, show no evidence of statistically significant $\frac{b-c}{n}$ and are considered confident. This is explained in more details in the next section. In addition, Tango's confidence test is presented in (Tango, 1998) and is reviewed in appendix A (section 6).

$$1.\ ROC = \left\{ \begin{array}{|c|c|}\hline a_i & b_i \\ \hline c_i & d_i \\ \hline \end{array} \;\middle|\; \begin{array}{l} t_i \in T, i = 1, \cdots, |T|, \\ \begin{array}{|c|c|}\hline a_i & b_i \\ \hline c_i & d_i \\ \hline \end{array} = K(D, t_i), \\ 0 \leq T_i \leq 1 \end{array} \right\}$$

$$2.\ S = \left\{ \begin{array}{|c|c|}\hline a_i & b_i \\ \hline c_i & d_i \\ \hline \end{array} \;\middle|\; \begin{array}{l} (u_i, l_i) = Tango_\alpha(b_i, c_i, n), \\ \begin{array}{|c|c|}\hline a_i & b_i \\ \hline c_i & d_i \\ \hline \end{array} \in ROC, \\ 0 \in [u_i, l_i], \\ i = 1, \cdots, |ROC|, n = |D| \end{array} \right\}$$

$$3.\ CAUC = \begin{cases} 0 & \text{if } S = \text{empty} \\ AUC(S) & \text{if } S \neq \text{empty} \end{cases}$$

$$4.\ AveD = \frac{1}{m} \sum_{i=1}^{m} \frac{b_i - c_i}{n} \qquad \begin{array}{l} \forall \begin{array}{|c|c|}\hline a_i & b_i \\ \hline c_i & d_i \\ \hline \end{array} \in S, \\ m = |S| \end{array}$$

Figure 2. Evaluating classifier K (on data D with T class probability thresholds) by Tango at confidence level (1-α). S contains confident ROC points, $CAUC$ is the area under S, and $AveD$ is the average error difference.

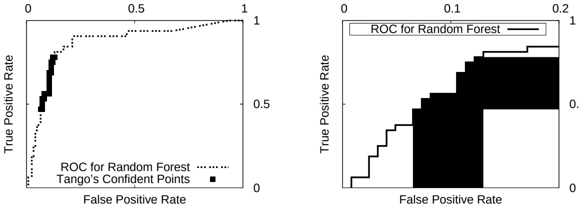


Figure 3. Sample Confident ROC segment (left). Area under ROC segment (right).

3. The Proposed Method of Evaluation

Presented in figure 2, our evaluation consists of four steps: **(1)** Generate an ROC curve for a classifier K applied on test examples D with increasing class probability thresholds t_i (0 to 1). **(2)** For each resulting point (a confusion matrix along the ROC curve), apply Tango's test to compute the 95%-confidence interval $[u_i, l_i]$, within which lies the point of the observed difference $\frac{b_i - c_i}{n}$. If $0 \in [u_i, l_i]$, then this point is identified as a confident point and is added into the set of confident points S. Points in S form the confident ROC segment illustrated in the left plot of figure 3. Our framework is generic and accommodates a test of choice provided that it produces a meaningful interpretation of results. **(3)** Compute $CAUC$ the area under the confident ROC segment S, shown in the right plot of figure 3. **(4)** Compute $AveD$ the average normalized difference ($\frac{b-c}{n}$) for all points in S. In our experiments, we plot the area under the confident ROC segment $CAUC$ against the average observed classification difference $AveD$. Lower values for $AveD$ suggests low classification difference and higher values for $CAUC$ indicate larger confident ROC segment. An effective classifier shows low $AveD$ and high $CAUC$.

Table 2. UCI data sets (Newman et al., 1998)

Data Set	Training	Testing
`dis`	45(+)/(-)2755	13(+)/(-)959
`hypothyroid`	151(+)/(-)3012	–
`sick`	171(+)/(-)2755	13(+)/(-)959
`sick-euthyroid`	293(+)/(-)2870	–
`SPECT`	40(+)/(-)40	15(+)/(-)172
`SPECTF`	40(+)/(-)40	55(+)/(-)214

4. Experiments

Having presented our evaluation framework, we now present an overview of our experiments and their data sets followed by an assessment of results to motivate conclusions. The data sets, listed in table 2, are selected from the UCI-Machine Learning repository (Newman et al., 1998) and consist of examples of two-class problems. They are severely imbalanced with the number of positive examples reaching as low as 1.4% (`dis`) and not exceeding 26% (`spectf`). Only (`spect`) and (`spectf`) data sets have a balanced training set and imbalanced testing set. On these data sets, we train four classifiers and compare their performances as reported by the ROC, by the AUC, and by our method. If testing data sets are unavailable, we use cross-validation of 10 folds. Using `Weka 3.4.6` (Witten & Frank, 2005), we build a decision stump classifier without boosting (`S`), a decision tree (`T`), a random forest (`F`), and a Naive Bayes (`B`) classifier. The rationale is to build classifiers for which we can expect a ranking of performance. A decision stump built without boosting is a decision tree with one test at the root (only 2 leave nodes) and is expected to perform particularly worse than a decision tree. Relatively, a decision tree is a stronger classifier since it is more developed and has more leave nodes that cover the training examples. The random forest classifier is a reliable classifier and is expected to outperform a single decision tree. Finally, the naive Bayes classifier tends to minimize classification error and is expected to perform reasonably well when trained on a balanced training set.

We first investigate the usefulness of ROC confidence bands on data with imbalance. Figure 4 shows the ROC confidence bands for our four classifiers on the most imbalanced `dis` data set. These bands are gen-

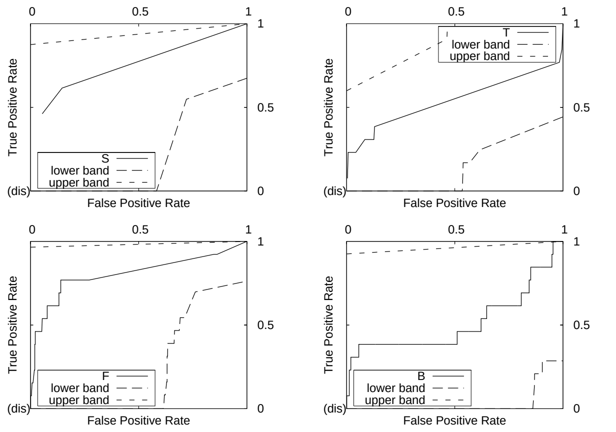


Figure 4. ROC confidence bands for decision stump (S), decision tree (T), random forest (F), and naive Bayes (B) on (`dis`) data set. The bands are wide and are not very useful.

Table 3. AUC values for decision stump (S), decision tree (T), random forest (F), and naive Bayes (B) on data sets.

Data Set	(S)	(T)	(F)	(B)
dis	0.752	0.541	**0.805**	0.516
hypothyroid	0.949	0.936	**0.978**	0.972
sick	0.952	0.956	**0.997**	0.946
sick-euthyroid	0.931	0.930	**0.978**	0.922
spect	0.730	0.745	0.833	**0.835**
spectf	0.674	0.690	**0.893**	0.858

erated using the empirical fixed-width method (Macskassy & Provost, 2004) at the 95% level of confidence (like Tango's test, this method of generating ROC bands does not make assumptions of the underlying distributions of the data). We claim that with severe imbalance, sampling-based techniques do not work. Clearly, the generated bands are very wide and contain more than 50% of the ROC space proving that they are not very useful. This result is also consistent on the other data sets.

Next, we consider the ROC curves of our four classifiers on all data sets shown in figure 5. Recall, ROC curves are compared by being more dominantly placed towards the north-west of the plot (higher true positive rate and lower false positive rate). We observe that the decision stump (S) performs the same or better than the decision tree (T) on all data sets. In addition, the random forest (F), consistently, outperforms the naive Bayes (B). In fact, (F) shows the best performance on most data sets. When we consider the AUC values of these classifiers, shown in table 3, (S) has similar or higher AUC values than (T). Furthermore, the AUC of (F) is, clearly, higher than that of the others on most the data sets (the bold numbers in table 3). When trained on a balanced data set (`SPECT`), (F) and (B) classifiers perform significantly better than the others.

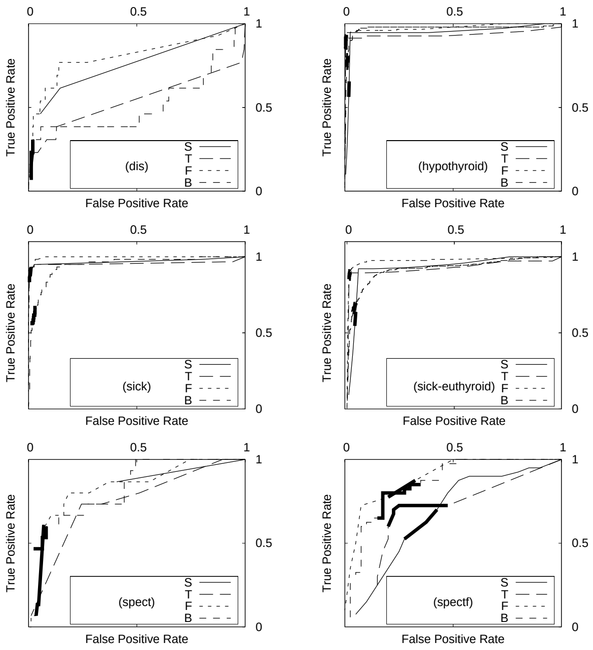


Figure 5. ROC curves for decision stump (S), decision tree (T), random forest (F), and naive Bayes (B) on all data set. The dark segments are Tango's confident points.

In contrast, the results obtained by our proposed evaluation measure are presented in figure 6. Each plot in the figure reports our evaluation of the four classifiers on each data set. The x-axis represents the average normalized classification difference $\frac{b-c}{n}$ for those confident points on the ROC. The y-axis represents the area under the confident segment of the ROC. This area includes the TP area (vertical area) and the FP area (the horizontal area) as illustrated in figure 3 on page 3. Classifiers placed towards the top-left corner perform better (bigger area under the confident ROC segment and less difference in classification error) than those placed closer to the bottom right corner (smaller confident area and higher difference in classification error). Classifiers that fail to produce confident points on their ROC curves are excluded from the plots. The decision stump (S) fails to produce confident points along its ROC, therefore, it does not appear in any of the plots in the left column of figure 6. This is consistent with our expectation of it being less effective. In fact, plots in the right column of the same figure show that (S) also performs poorly producing higher classi-

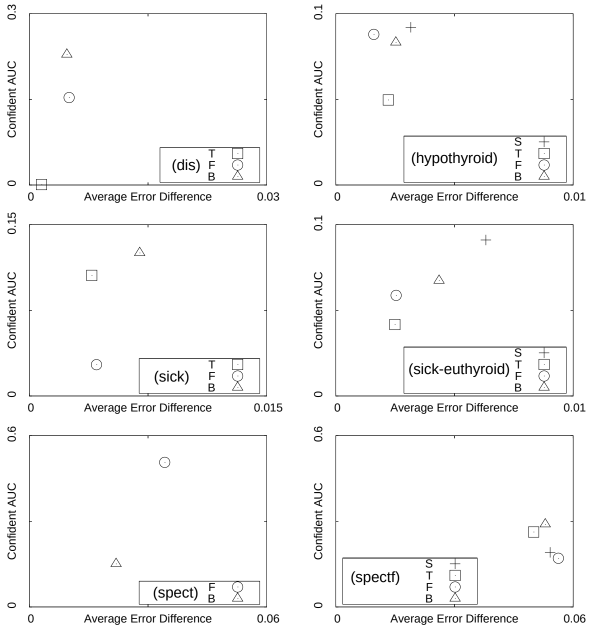


Figure 6. Our evaluation for decision stump (S), decision tree (T), random forest (F), and naive Bayes (B) on our data sets. The y-axis shows the area under the confident ROC segment and the x-axis shows the average observed classification difference $\frac{b-c}{n}$.

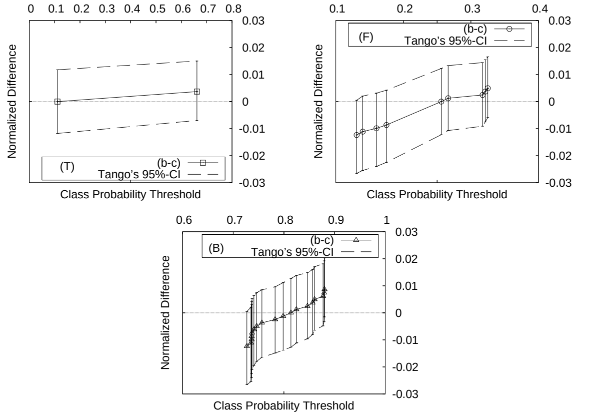


Figure 7. Tango's 95%-confidence intervals for ROC points of decision tree (T), random forest (F), and naive Bayes (B) on (`dis`) set. The center points are ($\frac{b-c}{n}$). (T) has a wide range of thresholds (x-axis).

fication difference and/or covering smaller area under its confident ROC segment. In fact, even when (S) has higher confident AUC than (T), in the right plots of figure 6, (S) still shows a significantly higher difference in classification error than that of (T). The decision tree (T), on the other hand, performs well in most cases and outperforms all other classifiers in the bottom right plot in figure 6. (T) certainly outperforms the (S) which contradicts observations based on the ROCs and AUCs. Furthermore, (T) fails to produce confident points on the (`spect`) data set (bottom left plot of the same figure). Perhaps, since (`spect`) is a binary data set extracted from the continuous (`spectf`) set, this may suggest that the extraction process hinders the decision tree learning. (F) and (B) classifiers appear reasonably consistent on all data sets with (B) being particularly strong on the (`dis`) data set. However, the surprise is (B) showing significantly higher confident AUC than (F) on all data sets with the exception of the `spect` data set in the bottom left plot of figure 6. Moreover, (B) shows significantly better performance particularly on the (`dis`) data set.

Our results, clearly, contradict conclusions based on the ROC and AUC evaluations. Therefore, we investigate those confident points along the ROCs for two situations. First, when the four classifiers are trained and tested on the same imbalanced `dis` data set using cross-validation. Second, when the four classifiers are trained on a balanced training set and are tested on an imbalanced testing `SPECTF` data set. For the first situation (`dis` data set), the ROC curves reveal that three of the classifiers produce confident classification points in the bottom left section of the ROC space (see the bold segments in the top left plot of figure 5). These confident points are detected by our method at the 95% level of confidence and are consistent with having severely imbalanced data sets. When we consider the corresponding Tangos 95%-confidence intervals for these classifier (see figure 7), we see that confident points produced by (T) cover a wider range of probability threshold (0.1 to 0.65 on the x-axis of the top left plot) with a low classification difference (y-axis). This indicates added confidence in (T)'s performance. (T) produces only two points which may be due to the very low number of positive examples. Alternatively, despite generating many more confident points, (F) and (B) classifiers show higher variations of classification difference for a much narrower range of thresholds values. At the least, this indicates a distinction between these classifiers.

For the second situation (`SPECTF` data set), consider the ROC curves in the bottom right plot of figure 5. (T) and (B), clearly, outperform (S) and (F) on this data set. Tango's 95%-confidence intervals of the confident ROC points (shown in figure 8) show that (T) and (B) outperform the other classifiers. When trained on the balanced `spectf` data set, (T) shows the least difference in classification error and has a significantly wider range of threshold values in which it produces

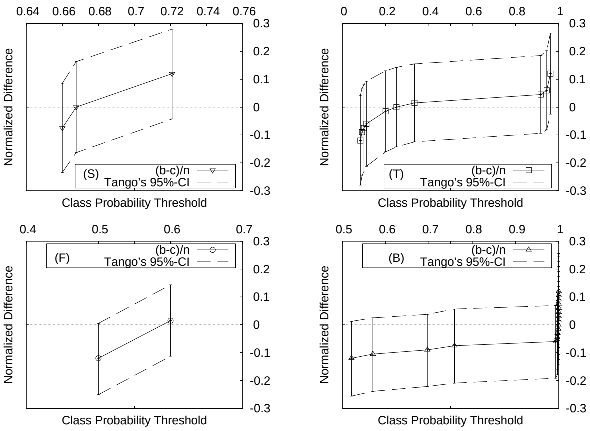


Figure 8. Tango's 95%-confidence intervals for ROC points of decision stump (S), decision tree (T), random forest (F), and naive Bayes (B) on (`spectf`) set. The center points are $(\frac{b-c}{n})$. (T) and (B) have a wider range of thresholds (x-axis) and produce more confident points.

many confident points (0 to 1 along the x-axis of the top right plot in figure 8). Also in this figure, (S) and (T) produce classification points that have exactly zero classification difference while the other two come close to the zero classification difference.

5. Conclusions and Future Work

We propose a method to address classifier evaluation in the presence of severe class imbalance with significantly fewer positive examples. In this case, our experiments show that ROC confidence bands fail to provide meaningful results. We propose a notion of statistical confidence by using a statistical tests, borrowed from biostatistics, to compute the 95%-confidence intervals on the difference in classification. Our framework incorporates this evaluation test into the space of the ROC curves to produce confidence oriented evaluation. Our method results in the presentation of the tradeoff between classification difference and area under the confident segment of the ROC curve. Our experiments show that our method is more reliable than general ROC and AUC measures.

In the future, we plan to compare our evaluation results to other methods of generating ROC bands to show further usefulness of our framework. Also, it can be useful to compute confidence bands or intervals for these proposed confident ROC segments. This remains a difficult task because the confidence in our method is computed on the classification difference which may not map easily to the ROC space. We plan to investigate the feasibility of mapping the confidence intervals from this work into the ROC space. This may be interesting particularly when there is no danger of imbalance. Although this work addresses the case of severe imbalance in the data, Tango's test of confidence can still be applied to balanced data sets. We plan to explore our framework in balanced situations with the aim to drive useful and meaningful evaluation metrics to provide confidence and reliability. Furthermore, Tango's test is a clinical equivalence test. This may possibly provide the basis to derive a notion of equivalence on classification.

Acknowledgments

The authors thank James D. Malley at the National Institute of Health for his suggestions and communications. In addition, we acknowledge the support of The Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Cohen, W. W., Schapire, R. E., & Singer, Y. (1999). Learning to order things. *Journal of Artificial Intelligence Research*, 243–270.

Dietterich, T. G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, *10*, 1895–1923.

Drummond, C., & Holte, R. C. (2000). Explicitly representing expected cost: An alternative to roc representation. *the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 198–207.

Drummond, C., & Holte, R. C. (2004). What roc curves can't do (and cost curves can). *ECAI'2004 Workshop on ROC Analysis in AI.*

Drummond, C., & Holte, R. C. (2005). Severe class imbalance: Why better algorithms aren't the answer. *Proceedings of the 16th European Conference of Machine Learning*, 539–546.

Everitt, B. S. (1992). *The analysis of contingency tables.* Chapman-Hall.

Ling, C. X., Huang, J., & Zang, H. (2003). Auc: a better measure than accuracy in comparing learning algorithms. *Canadian Conference on AI*, 329–341.

Macskassy, S. A., & Provost, F. (2004). Confidence bands for roc curves: Methods and empirical study. *in Proceedings of the 1st Workshop on ROC Analasis in AI (ROCAI-2004) at ECAI-2004.*

Macskassy, S. A., Provost, F., & Rosset, S. (2005). Roc confidence bands: An empirical evaluation. *in Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, 537 – 544.

Motulsky, H. (1995). *Intuitive biostatistics.* Oxford University Press, New York.

Newcombe, R. G. (1998a). Improved confidence intervals for the difference between binomial proportions based on paired data. *Statistics in Medicine*, *17*, 2635–2650.

Newcombe, R. G. (1998b). Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in Medicine*, *17*, 857–872.

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences.

Provost, F., & Fawcett, T. (1997). Analysis and visualization f classifier performance: Comparison under imprecise class and cost distributions. *the Third International Conference on Knowledge Discovery and Data Mining*, 34–48.

Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, 1285–1293.

Tango, T. (1998). Equivalence test and confidence interval for the difference in proportions for the paired-sample design. *Statistics in Medicine*, *17*, 891–908.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques.* 2nd Edition, Morgan Kaufmann.

6. Appendix A: Tango's Confidence Intervals

Clinical trials, case-control studies, and sensitivity comparisons of two laboratory tests are examples of medical studies that deal with the difference of two proportions in a paired design. Tango's test (Tango, 1998) builds a model to derive a one-sided test for equivalence of two proportions. Medical equivalence is defined as no more than 100Δ percent inferior, where $\Delta(> 0)$ is a pre-specified acceptable difference. Tango's test also derives a score-based confidence interval for the difference of binomial proportions in paired data. Statisticians have long been concerned with the limitations of hypothesis testing used to summarize data (Newcombe, 1998b). Medical statisticians prefer the use of confidence intervals rather than p-values to present results. Confidence intervals have the advantage of being close to the data and on the same scale of measurement, whereas p-values are a probabilistic abstraction. Confidence intervals are usually interpreted as margin of errors because they provide magnitude and precision. A method deriving confidence intervals must be a priori reasonable (justified derivation and coverage probability) with respect to the data (Newcombe, 1998b).

The McNemar test is introduced in (Everitt, 1992) and has been used to rank the performance of classifiers in (Dietterich, 1998). Although inconclusive, the study showed that the McNemar test has low Type I error with high power (the ability to detect algorithm differences when they do exist). For algorithms that can be executed only once, the McNemar test is the only test that produced an acceptable Type I error (Dietterich, 1998). Despite Tango's test being an equivalence test, setting the minimum acceptable difference Δ to zero produces an identical test to the McNemar test with strong power and coverage probability (Tango, 1998). In this work, we use Tango's test to compute confidence intervals on the difference in classification errors in both classes with a minimum acceptable difference $\Delta = 0$ at the (1-α) confidence level. Tango makes few assumptions; (1) the data points are representative of the class. (2) The predictions are reasonably correlated with class labels. This means that the misclassified positives and negatives are relatively smaller than the correctly classified positives and negatives respectively. In other words, the classifier does reasonable well on both classes, rather than performing a random classification. We consider classifier predictions and class labels as paired machines that fit the matched paired design. As shown in table 1 on page 1, entries a and d are the informative or the discordant pairs indicating the agreement portion $(q_{11} + q_{22})$, while b and c are the uninformative or concordant pairs representing the proportion of disagreement $(q_{12} + q_{21})$ (Newcombe, 1998a). The magnitude of the difference δ in classifications errors can be measured by testing the null hypothesis $H_0 : \delta = q_{12} - q_{21} = 0$. This magnitude is conditional on the observed split of b and c (Newcombe, 1998a). The null hypothesis H_0 is tested against the alternative $H_1 : \delta \neq 0$. Tango's test derives a simple asymptotic (1-α)-confidence interval for the difference δ and is shown to have good power and coverage probability. Tango's confidence intervals can be computed by:

$$\frac{b - c - n\delta}{\sqrt{n(2\hat{q_{21}} + \delta(1-\delta))}} = \pm Z_{\frac{\alpha}{2}} \qquad (1)$$

where $Z_{\frac{\alpha}{2}}$ denotes the upper $\frac{\alpha}{2}$-quantile of the normal distribution. In addition, $\hat{q_{21}}$ can be estimated by the maximum likelihood estimator for q_{21}:

$$\hat{q_{21}} = \frac{\sqrt{W^2 - 8n(-c\delta(1-\delta))} - W}{4n} \qquad (2)$$

where $W = -b - c + (2n - b + c)\delta$. Statistical hypothesis testing begins with a null hypothesis and searches for sufficient evidence to reject that null hypothesis. In this case, the null hypothesis states that there is no difference, or $\delta = 0$. By definition, a confidence interval includes plausible values for the null hypothesis. Therefore, if the zero is not included in the computed interval, then the null hypothesis $\delta = 0$ is rejected. On the other hand, if the zero value is included in the interval, then we do not have sufficient evidence to reject the difference being zero, and the conclusion is that the difference can be of any value within the confidence interval at the specified level of confidence (1-α).

Tango's test of equivalence can reach its limits in two cases; (1) when the values of b and c are both equal to zero where the Z statistic does not produce a value. This case occurs when we build a perfect classifier and is consistent with the test not using the number of correctly classified examples a and d. (2) The values b and c differ greatly. This is consistent with the assumption that the classifier is somewhat reasonably good, i.e. the classifier is capable of detecting a reasonable portion of the correct classifications in the domain. In both cases of limitations, the confidence intervals are still produced and are reliable (Tango, 1998) but may be wider in range. Tango's confidence intervals are shown not to collapse nor they exceed the boundaries of the normalized difference of $[-1, 1]$ even for small values of b and c.

Cost Curves for Abstaining Classifiers

Caroline C. Friedel CAROLINE.FRIEDEL@BIO.IFI.LMU.DE
Institut für Informatik, Ludwig-Maximilians-Universität München, Amalienstr. 17, 80333 München, Germany

Ulrich Rückert RUECKERT@IN.TUM.DE
Stefan Kramer KRAMER@IN.TUM.DE
Institut für Informatik/I12, Technische Universität München, Boltzmannstr. 3, 85748 Garching b. München, Germany

Abstract

We present abstention cost curves, a new three-dimensional visualization technique to illustrate the strengths and weaknesses of abstaining classifiers over a broad range of cost settings. The three-dimensional plot shows the minimum expected cost over all ratios of false-positive costs, false-negative costs and abstention costs. Generalizing Drummond and Holte's cost curves, the technique allows to visualize optimal abstention settings and to compare two classifiers in varying cost scenarios. Abstention cost curves can be used to answer questions different from those addressed by ROC-based analysis. Moreover, it is possible to compute the volume under the abstention cost curve (VACC) as an indicator of the classifier's performance across all cost scenarios. In experiments on UCI datasets we found that learning algorithms exhibit different "patterns of behavior" when it comes to abstention, which is not shown by other common performance measures or visualization techniques.

1. Introduction

In many application areas of machine learning it is not sensible to predict the class for each and every instance, no matter how uncertain the prediction is. Instead, classifiers should have the opportunity to abstain from risky predictions under certain conditions. Our interest in abstaining classifiers is motivated by specific applications, for instance in chemical risk assessment, where it is considered harmful to predict the toxicity or non-toxicity of a chemical compound if the prediction is weak and not backed up by sufficient training material.

Abstaining classifiers can easily be derived from non-abstaining probabilistic or margin-based classifiers by defining appropriate thresholds which determine when to classify and when to refrain from a prediction. The lower and upper thresholds, within which no classifications are made, constitute a so-called abstention window (Ferri et al., 2004). Making use of abstention windows, a recent approach based on ROC analysis (Pietraszek, 2005) derives an optimal abstaining classifier from binary classifiers. In this approach the thresholds can be determined independently of each other from the convex hull of ROC curves. However, ROC-based approaches assume at least known misclassification costs. Moreover, classifiers and optimal abstention thresholds cannot be compared directly for a range of possible cost matrices, as it is usually done in cost curves (Drummond & Holte, 2000).

In this paper, we propose an alternative approach to ROC-based analysis of abstaining classifiers based on cost curves. The advantage of cost curves is that cost-related questions can be answered more directly, and that the performance over a range of cost scenarios can be visualized simultaneously. The proposed generalization of cost curves plots the optimal expected costs (the z-axis) against the ratio of false positive costs to false negative costs (the x-axis) and the ratio of abstention costs to false negative costs (the y-axis). The fundamental assumption is that abstention costs can be related to misclassification costs. As pointed out by other authors (Pietraszek, 2005), unclassified instances might take the time or effort of other classifiers (Ferri et al., 2004), or even human experts. Another scenario is that a new measurement has to be made for

Appearing in *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, Pittsburgh, PA, 2006.

the instance to be classified. Thus, abstention costs link misclassification costs with attribute costs. Consequently, the setting is in a sense related to active learning (Greiner et al., 2002). Along those lines, we also assume that abstention costs are the same independently of the true class: Not knowing the class, the instances are handled in the very same way.

We devised a non-trivial, efficient algorithm for computing the three-dimensional plot in time linear in the examples and in the number of grid points (Friedel, 2005). The algorithm takes advantage of dependencies among optimal abstention windows for different cost scenarios to achieve its efficiency. However, the focus of this paper is not on the algorithm, but on actual abstention cost curves of diverse classifiers on standard UCI datasets. We present abstention cost curves as well as "by-products", showing the abstention rates and the location of the abstention window (the lower and upper interval endpoints). Moreover, a new aggregate measure, the volume under the abstention cost curve (VACC), is presented. VACC is related to the expected abstention costs, if all cost scenarios are equally likely.

2. Abstaining in a Cost-Sensitive Context

Before going into detail, we need to specify some basic concepts and introduce the overall setting. First of all, we assume that a classifier Cl has been induced by some machine learning algorithm. Given an instance x taken from an instance space $\mathcal{X}$, this classifier assigns a class label $y(x)$ taken from the target class $\mathcal{Y} = \{P, N\}$, where P denotes the *positive* class and N the *negative* class. To avoid confusion, we use capital letters for the actual class and lowercase letters for the labels assigned by the classifier. We would now like to analyze this classifier on a validation set $S = \{s_1, s_2, \ldots, s_r\}$ containing r instances with classes $\{y_1, y_2, \ldots, y_r\}$. As argued in the work on ROC curves (e.g. in (Provost & Fawcett, 1998)), it can make sense to use a different sampling bias for the training set than for the validation set. In this case, the class probabilities in the validation set might differ from the class probabilities of the training set or the true class probabilities. Thus, we do not explicitly assume, that the validation set shows the same class distribution as the training set, even though this is the case in many practical applications. However, we demand that the classifier outputs the predicted class label as well as some confidence score for each instance in the validation set. For simplicity we model class label and confidence score as one variable, the *margin*. The margin $m(s)$ of an instance s is positive, if the predicted class is p and negative otherwise. The absolute value of the margin is between zero and one and gives some estimate of the confidence in the prediction. Thus, the margin $m(s)$ of an instance s ranges from -1 (clearly negative) over 0 (equivocal) to +1 (clearly positive).

Applying the classifier to the validation set, yields a sequence of r (not necessarily distinct) margin values $M = (m(s_1), m(s_2), \ldots, m(s_r))$. Sorting this sequence in ascending order yields a characterization of the uncertainty in the predictions. The certain predictions are located at the left and right end of the sequence and the uncertain ones somewhere in between. Based on the information in this sequence one can then allow the classifier Cl to abstain for instances with margin values between a lower threshold l and an upper threshold u. Any such ordered pair of thresholds constitutes an *abstention window* $a := (l, u)$. A specific *abstaining classifier* is defined by an abstention window a and its prediction on an instance x is given as

$$\pi(a, x) = \begin{cases} p & \text{if } m(x) \geq u \\ \perp & \text{if } l < m(x) < u \\ n & \text{if } m(x) \leq l \end{cases} \tag{1}$$

where $\perp$ denotes "don't know".

As both the upper and lower threshold of an abstention window are real numbers, the set of possible abstention windows is uncountably infinite. Therefore, we have to restrict the abstention windows considered in some way. If we are given the margin values as a sorted vector $(m_1, \ldots, m_k)$ of distinct values – i.e., $m_1 < \cdots < m_k$ – it is sensible to choose the thresholds just in between two adjacent margin values. To model this, we define a function $v : \{0, \ldots, k\} \to \mathcal{R}$ which returns the center of the margin with index i and the next margin to the right. We extend the definition of v to the case where $i < 1$ or $i = k$ to allow for abstention windows that are unbounded on the left or on the right:

$$v(i) = \begin{cases} \frac{m_i + m_{i+1}}{2} & \text{if } 1 \leq i < k \\ -\infty & \text{if } i = 0 \\ +\infty & \text{if } i = k. \end{cases} \tag{2}$$

Note that the original margin sequence may contain the same margin value more than once, but v is defined only on the $k \leq n$ distinct margin values. The set of abstention windows $\mathcal{A}(Cl)$ for a classifier Cl is then $\mathcal{A}(Cl) := \{(v(i), v(j)) | 0 \leq i \leq j \leq k\}$. Where the classifier is clear from the context, we omit it and denote the set just by $\mathcal{A}$.

The performance of an abstention window is assessed in terms of expected cost on the validation set. To calculate this, we need information about the costs associated with each combination of true target class and predicted target class. For our purposes, the costs are given in a *cost matrix* C such that $C(\theta, \pi)$ is the cost of labeling an instance of true class $\theta \in \{P, N\}$ with $\pi \in \{p, n, \perp\}$:

$$C := \begin{pmatrix} C(P,p) & C(P,n) & C(P,\perp) \\ C(N,p) & C(N,n) & C(N,\perp) \end{pmatrix} \tag{3}$$

As the relative frequency on the validation set can be considered as a probability measure, we use conditional probabilities to denote the classification/misclassification rates of an abstention window $a = (l, u)$ on the validation set S. For example, the *false positive rate* of the abstention window a on S is denoted by

$$P_{S,a}(p|N) := \frac{\big|\{s \in S | y(s) = N \wedge \pi(a,s) = p\}\big|}{\big|\{s \in S | y(s) = N\}\big|} \tag{4}$$

Similarly, we have the *true positive rate* $P_{S,a}(p|P)$, the *false negative rate* $P_{S,a}(n|P)$, the *positive abstention rate* $P_{S,a}(\perp|P)$, the *true negative rate* $P_{S,a}(n|N)$, and the *negative abstention rate* $P_{S,a}(\perp|N)$. With this we can calculate the *expected cost* of an abstention window a on S for cost matrix C as the sum of the products of cost and probability over all events:

$$\mathbf{EC}(C, a, S) := \sum_{\theta \in \{N,P\}} \sum_{\pi \in \{n,p,\perp\}} C(\theta, \pi) P_{S,a}(\pi|\theta) P(\theta). \tag{5}$$

In this equation $P(\theta)$ denotes the probability of an example belonging to class $\theta \in \{N, P\}$. In most applications this is just the fraction of positive and negative examples in the validation set. Sometimes, one might want to use other values for those quantities, for example to accommodate for a resampling bias.

For a given cost matrix C, we are primarily interested in the *optimal abstention window* $a_{opt} := \text{argmin}_{a \in \mathcal{A}} \mathbf{EC}(C, a, S)$, that is, the abstention window with the lowest expected cost on the validation set. We observe that the optimal abstention window does not depend on the absolute values of the costs, but only on the relation of the individual costs to each other and the class probabilities $P(P)$ and $P(N)$. For example, multiplying all values in the cost matrix by a constant factor c_m does not change the optimal window. Similarly, adding a constant c_P to the upper row and a constant c_N to the lower row of the cost matrix also has no effect on the optimal abstention window. Let C' denote C with c_P added to the upper row and c_N added to the lower row. Then:

$$\begin{aligned} &\mathbf{EC}(C', a) = \\ &\quad P(P) \sum_{\pi \in \{n,p,\perp\}} (C(P,\pi) + c_P) P(\pi|P) \\ &\quad + P(N) \sum_{\pi \in \{n,p,\perp\}} (C(N,\pi) + c_N) P(\pi|N) \\ &\quad = \mathbf{EC}(C, a) + P(P)c_P + P(N)c_N \end{aligned}$$

Thus, $\text{argmin}_{a \in \mathcal{A}} \mathbf{EC}(C', a, S) = \text{argmin}_{a \in \mathcal{A}} \mathbf{EC}(C, a, S)$ and the optimal abstention window remains the same. Consequently, we can transform any cost matrix in a normal form C' by adding $c_P = -C(P, p)$ and $c_N = -C(N, n)$ to the upper and lower rows respectively and then multiplying with $c_m = 1/(C(P, n) - C(P, p))$. This "normalization" operation does not change the optimal abstention window, but it ensures that $C'(P, p) = C'(N, n) = 0$ and that $C'(P, n) = 1$. In the following we always assume a normalized cost matrix C' such that the optimal abstention window depends only on the relative false positive costs $C'(N, p)$ and abstention costs $C'(P, \perp)$ and $C'(N, \perp)$:

$$C' := \begin{pmatrix} 0 & 1 & C'(P,\perp) \\ C'(N,p) & 0 & C'(N,\perp) \end{pmatrix} \tag{6}$$

In many applications abstaining on an instance results in additional tests. As the true class of an instance is not known at that point, the cost of such a test is the same for both types of instances, i.e. the cost of abstention is independent of the true class of an instance. In the following we will therefore focus on cases where $C'(P, \perp) = C'(N, \perp) := C'(\perp)$[1]. This means that the optimal window of a given cost matrix in normal form is uniquely determined by just two parameters $\mu := C'(N, p)$ and $\nu := C'(\perp)$. The *normalized expected cost* of an abstention window a can then be written as a function of μ and ν:

$$\begin{aligned} &c(a, \mu, \nu) := \\ &P_{S,a}(n|P)P(P) + \mu P_{S,a}(p|N)P(N) + \nu P_{S,a}(\perp) \end{aligned} \tag{7}$$

In this problem formulation μ represents the false positive costs relative to the false negative costs, while ν controls the abstention costs relative to the false negative costs. As it turns out, abstention does not make sense for all possible settings of μ and ν. For instance, if ν is greater than μ, we can do better by classifying an instance as positive instead of abstaining. The

[1] If this condition is not fulfilled, it is is still possible to compute optimal abstention windows. However, the computational efficiency suffers from more complicated cost settings.

following lemma quantifies this phenomenon. For the sake of simplicity, we use the fractions of positive and negative instances in the validation set for $P(P)$ and $P(N)$. Therefore, we can determine $P_{S,a}(n|P)P(P)$, $P_{S,a}(p|N)P(N)$ and $P_{S,a}(\perp)$ by counting the occurrences of each event and then dividing by the number of instances r.

Lemma 1. *Let S, μ and ν be defined as before. If $\nu > \frac{\mu}{1+\mu}$, the optimal abstention window a_{opt} is empty, i.e. $l_{opt} = u_{opt}$ (proof omitted).*

3. Cost Curves for Abstaining Classifiers

If the cost matrix and the class probabilities in a learning setting are known exactly, one can determine the optimal abstention window a_{opt} simply by calculating the expected costs for all windows. However, for most applications costs and class distributions are uncertain and cannot be determined exactly. In such a setting one would like to assess the performance of an abstaining classifier for a broad range of cost settings. Even in the case of non-abstaining classifiers one might want to illustrate a classifier's behavior for varying cost matrices or class distributions. The two most prominent visualization techniques to do so are ROC curves (Provost & Fawcett, 1998) and cost curves (Drummond & Holte, 2000). In the following we present a novel method that allows to visualize the performance of abstaining classifiers. In principle, one could extend ROC curves or cost curves with a third dimension to accomodate for abstention. However, the meaning of the new axis in such an "extended" cost curve is not very intuitive, making it rather hard to interpret. Since visualization tools rely on easy interpretability, we follow a different approach[2].

The presented cost curve simply plots the normalized expected cost as given in equation (7). It is created by setting the x-axis to μ, the y-axis to ν and the z-axis to the normalized expected cost. Without loss of generality, we assume that the positive class is always the one with highest misclassification costs, so that $\mu \leq 1$ (if this is not the case, just flip the class labels). Furthermore, we can safely assume that $\nu \leq 1$, because otherwise the optimal abstention window is empty (as stated by lemma 1).

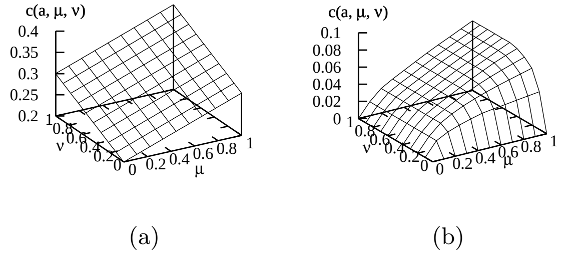


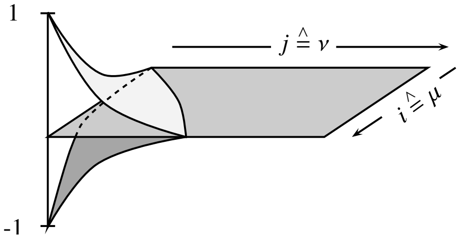
Figure 1: Example cost curves for uncertain costs, but fixed class distributions. (a) shows a cost curve for a specific abstention window, (b) a cost curve for an example classifier.

We can apply the cost curves in two ways. In the first case, we plot the normalized expected cost against the false positive and abstention costs for one fixed abstention window a. Then the resulting cost curve is just a plane, because $z = c(a, x, y)$ is linear in its parameters (see Figure 1(a)). This illustrates the performance of a classifier for one particular abstention window. In the second case, the cost curve is the lower envelope of all abstention windows, i.e. $z = \min_{a \in \mathcal{A}(Cl)} c(a, x, y)$ (see Figure 1(b)). This scenario is well suited for comparing two classifiers independently of the choice of a particular abstention window. For easier analysis, the curves can be reduced to two dimensions by color coding of the expected cost (see Section 4).

Using the information from cost curves, several questions can be addressed. First, we can determine for which cost scenarios one abstaining classifier Cl_s outperforms another classifier Cl_t. This can be done by examining a so-called differential cost curve $D(s,t)$, which is defined by $d_{i,j}(s,t) := k_{i,j}(s) - k_{i,j}(t)$. $d_{i,j}(s,t)$ is negative for cost scenarios for which Cl_s outperforms Cl_t and positive otherwise. Obviously, we can also compare a non-trivial classifier with a trivial one, which either always abstains or always predicts one of the two classes. Second, we can determine which abstention window should be chosen for certain cost scenarios by plotting the lower and the upper threshold of the optimal window for each cost scenario. Third, we can plot the abstention rate instead of expected costs in order to determine where abstaining is of help at all.

Although cost curves are continuous in theory, the visualization on a computer is generally done by calculating the z-values for a grid of specific values of x and y. The number of values chosen for x and y determines the resolution of the grid and is denoted as Δ. For computational considerations, we can thus define a cost curve for a classifier Cl as a $\Delta \times \Delta$ matrix $K(p)$ with $k_{i,j}(p) := min_{a \in \mathcal{A}(Cl)} c(a, i/\Delta, j/\Delta)$

[2] Technically, the presented cost curve assumes a fixed class distribution to allow for easier interpretation. We feel that the gain in interpretability outweighs the need for this additional assumption. In some settings cost curves that extend (Drummond & Holte, 2000) might be more suited; see (Friedel, 2005, section 3.4) for an elaborate comparison with the cost curves presented in this paper.



(a)

Figure 2: Schematic illustration of optimal abstention windows (upper threshold above the plane, lower threshold below) for various μ and ν. For the same μ and $\nu_1 < \nu_2$, the optimal window for ν_2 is contained in the window for ν_1.

for $0 \leq i, j \leq \Delta$. Calculating such a cost curve for moderately high values of Δ can be computationally demanding, as we have to determine the optimal abstention window for a large number of cost settings.

A naive algorithm would compute the cost curve by calculating the expected cost for each possible abstention window for each cost scenario. As the number of abstention windows is quadratic in the number of instances, this results in an algorithm in $O(\Delta^2 n^2)$. Our more efficient algorithm (Friedel, 2005) for computing cost curves largely relies on two observations:

1. The optimal abstention window a_{opt} can be computed in linear time by first determining the optimal threshold for zero abstention for the respective μ, and then finding the best abstention window located around this threshold.

2. for fixed μ and $\nu_1 < \nu_2$, the optimal abstention window for ν_2 is contained in the optimal abstention window for ν_1.

Thus, the optimal thresholds and abstention windows are arranged as illustrated by the schematic drawing in Figure 2: The plane in the center gives the optimal threshold between positive and negative classification; above we have the upper threshold of the optimal abstention window, and below the lower threshold. Based on these observations, it possible to design an efficient algorithm linear in the number of examples: In the first step, the optimal thresholds for non-abstention and the various values of μ are computed. Subsequently, the precise upper and lower thresholds around the optimal threshold found in the first step are determined.

4. Experiments

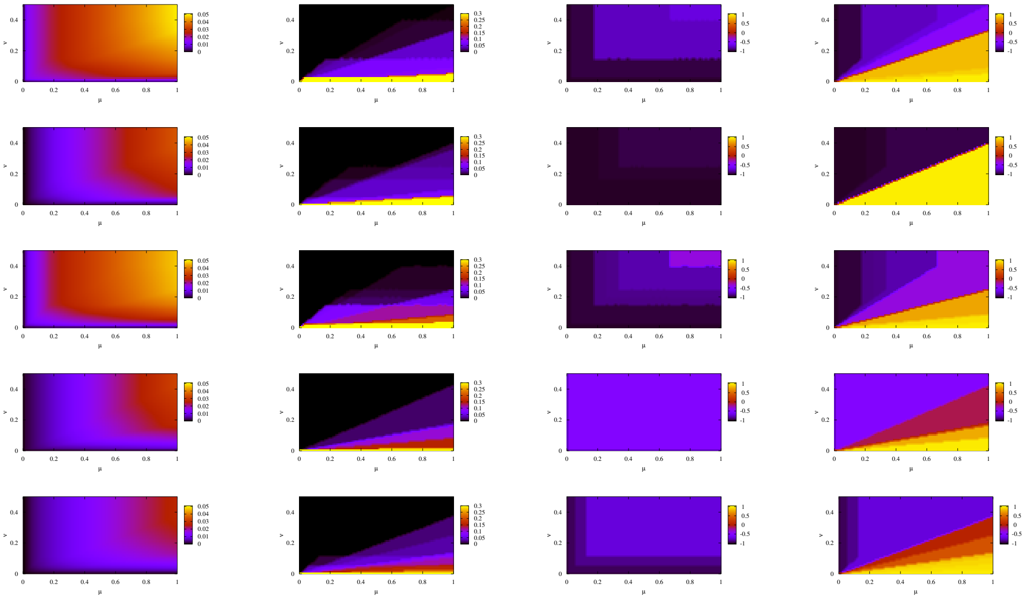
To analyze and visualize the abstention costs, we chose six two-class problems from the UCI repository: breast-w, bupa, credit-a, diabetes, haberman and vote. Five different machine learning algorithms, as implemented in the WEKA workbench (Witten & Frank, 2005), were applied to those datasets: J48, Naive Bayes (NB), PART, Random Forests (RF) and Support Vector Machines (SVM).

Alg.	Acc. (%)	AUC	VACC	Nrm. Acc.	Nrm. AUC	Nrm. VACC
breast-w						
J48	95	0.96	0.032	0.98	0.96	1.00
NB	96	0.98	0.018	0.99	0.99	0.58
PART	95	0.97	0.030	0.98	0.98	0.95
RF	95	0.99	0.016	0.98	0.99	0.50
SVM	97	0.99	0.014	1.00	1.00	0.44
bupa						
J48	65	0.67	0.16	0.97	0.90	0.93
NB	55	0.64	0.18	0.82	0.87	1.00
PART	62	0.67	0.17	0.93	0.91	0.97
RF	67	0.74	0.15	1.00	1.00	0.84
SVM	64	0.70	0.17	0.95	0.95	0.96
credit-a						
J48	87	0.89	0.082	1.00	0.97	0.88
NB	78	0.90	0.093	0.90	0.98	1.00
PART	85	0.89	0.089	0.98	0.98	0.95
RF	85	0.91	0.088	0.99	1.00	0.94
SVM	85	0.86	0.081	0.98	0.95	0.87
diabetes						
J48	73	0.75	0.15	0.96	0.90	1.00
NB	76	0.82	0.14	0.99	0.98	0.92
PART	74	0.79	0.14	0.96	0.95	0.94
RF	75	0.78	0.15	0.98	0.94	0.98
SVM	76	0.83	0.13	1.00	1.00	0.87
haberman						
J48	69	0.61	0.12	0.93	0.87	1.00
NB	75	0.65	0.11	1.00	0.93	0.95
PART	71	0.59	0.11	0.96	0.84	0.95
RF	68	0.65	0.12	0.91	0.93	1.00
SVM	74	0.70	0.11	1.00	1.00	0.96
vote						
J48	97	0.97	0.021	1.00	0.98	0.44
NB	90	0.97	0.046	0.93	0.98	1.00
PART	97	0.95	0.022	1.00	0.96	0.48
RF	96	0.98	0.021	1.00	0.99	0.44
SVM	96	0.99	0.022	0.99	1.00	0.47

Table 1: Summary of quantitative results of five learning algorithms applied to six UCI datasets

Our starting point is a summary of all quantitative results from ten-fold cross-validation on the datasets (see Table 1).[3] In the table, the predictive accuracy, the area under the (ROC) curve (AUC) and the volume under the abstention cost curve (VACC) are shown. The volume under the abstention cost curve can be

[3] In the experiments, we assume that the class distribution observed in the data resembles the true class distribution. Experiments assuming a uniform distribution (50:50) changed the absolute VACC numbers, but not their ordering.



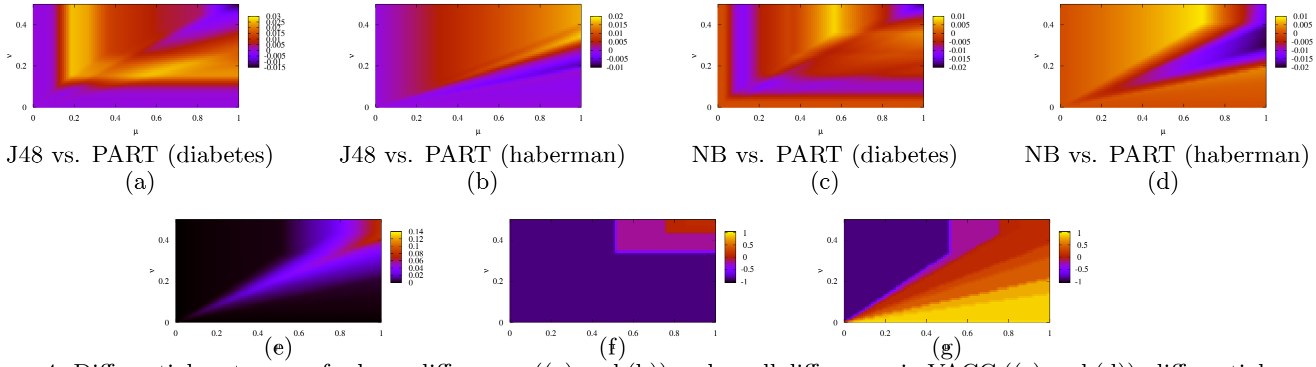
abstention cost curve *abstention rate* *lower threshold* *upper threshold*

Figure 3: Abstention cost curves, optimal abstention rates and lower/upper thresholds of the optimal abstention window on the breast-w dataset. From top to bottom: J48, NB, PART, RF, and SVM (as in Table 1)

defined as the double integral over μ and ν. VACC is related to the expected value of the abstention costs if all cost scenarios are equally likely. Moreover, the normalized values of those measures are given, that is, the value of the measure divided by the maximum over the classifiers' performance for the particular dataset. The normalized values are given to facilitate an easier comparison between the measures.

Overall, one can see that VACC in fact captures a different aspect than accuracy or AUC. In the following, we discuss the quantitative results from the table one by one. On breast-w, the VACC measure indicates significant differences in terms of abstention costs, which is neither reflected in predictive accuracy nor in AUC. For instance, we can see that there is a clear order over the classifiers from different learning algorithms: SVMs perform best, followed by RF and NB, whereas PART and J48 lag behind. This is also illustrated by the plots in Figure 3, which are discussed below. On the bupa dataset, NB performs worst and RF performs best according to all measures. However, the differences are not equally visible in all measures (see, e.g., RF vs. SVM or, vice versa, NB vs. PART). On credit-a, the comparison between J48 and NB hints at a marked difference in accuracy and VACC, not shown by AUC. PART vs. SVM is a different case: Comparable values for accuracy and AUC, but a considerable gap in VACC. For the diabetes data, a distinct difference is detected for RF vs. SVM in AUC/VACC, but not in terms of accuracy. On the haberman dataset, the variation in the quantitative results is negligible (for details, see below). Finally, the results on vote reveal that NB performs dramatically worse than all other approaches, perhaps due to the violated independence assumption on this particular dataset. This drop in performance is particularly visible in the VACC results.

Next, we have a closer look at the abstention cost curves and derived plots for all five learning algorithms on the breast cancer data (see Figure 3). In the leftmost column, the optimal abstention costs over all cost scenarios are visualized. Note that all plots are cut at $\nu = 0.5$, because for greater values of abstention costs, the abstention window is already degenerate, with $l = u$. The plots reflect the numbers from Table 1 adequately, but additionally show in which regions of the space the majority of costs occur. The second column from the left visualizes the abstention rate, that is, the fraction of instances the classifiers leaves unclassified. For instance, we can infer that PART should refrain from 10% to 15% of the predictions if the abstention costs are about one tenth of the false negative costs. The two right-most colums visualize the lower and the upper interval endpoints of the abstention window. To enable a visual comparison, all curves are plotted on the same scale. Considerable dif-



J48 vs. PART (diabetes)
(a)

J48 vs. PART (haberman)
(b)

NB vs. PART (diabetes)
(c)

NB vs. PART (haberman)
(d)

(e)

(f)

(g)

Figure 4: Differential cost curves for large differences ((a) and (b)) and small differences in VACC ((c) and (d)), differential cost curve SVMs vs. trivial classifier(s) on bupa (e), lower (f) and upper (g) thresholds of abstention window

ferences in the classifiers' abstention behavior become apparent.

In the plots, the isolines of l and u have a remarkably different shape. This can be explained as follows: First, both the upper and lower thresholds increase not continuously with ν or μ, but in steps. This is due to the fact that a critical value has to be reached for the cost of abstaining or classifying the instances between different threshold values, before thresholds are adjusted. Second, we observe that for values of ν for which abstaining is too expensive, the upper and the lower threshold are equal, as shown before.

The threshold shows a different behaviour only for those values of ν and μ that allow abstaining. In this range, the lower threshold depends only on the ratio between false negative costs (which are constant) and abstaining costs, and is thus independent of the false positive costs. The upper threshold on the other hand depends on both the abstaining costs ν and the false positive costs μ. In the same way as the lower threshold is effectively not affected by changes in μ in the range for which abstaining is reasonable, the upper threshold is not affected by changes in the false negative costs, which can easily be confirmed by switching the positive and negative labels.

Next, we take a look at *differential cost curves*. Differential cost curves are a tool for the practitioner to see in which regions of the cost space one classifier is to be preferred over another. In Figure 4, differential cost curves with large differences in VACC (upper row, (a) and (b)) and small differences in VACC (upper row, (c) and (d)) are shown. In Figure 4(a) and (b), J48 decision trees have smaller abstention costs than PART rules only in the bluish areas of the space. Differential cost curves also shed light on differences that do not appear in VACC, if a classifier is dominating in one region as it is dominated in another (Figure 4 (c) and (d)). The regions can be separated and quite distant in cost space, as illustrated by Figure 4 (c). The differential cost curve of NB vs. PART on haberman (Figure 4 (d)) demonstrates that even for datasets with no clear tendencies in accuracy, AUC or VACC, the plot over the cost space clearly identifies different regions of preference not shown otherwise.

Another interesting possibility is the comparison with the trivial classifier that always predicts positive, negative, or always abstains. In Figure 4 (e), we compare SVMs with trivial classifiers on the bupa dataset. In the black areas near the left upper and the right lower corner, the trivial classifer performs better than the SVM classifier. To explain this, we take a look at the lower and upper thresholds of the abstention window in Figure 4 (f) and (g). Strikingly, we find that in the upper left part $l = u = -1$, that is, everything is classified as positive, because false positives are very inexpensive compared to false negatives. However, in the lower right part $l = -1$ and $u = 1$, i.e., not a single prediction is made there, because abstention is inexpensive.

It is clear that the discussion of the above results remains largely on a descriptive level. However, ideally we would like to explain or even better, predict the behavior of classifiers on particular datasets. Unfortunately, this is hardly ever achieved in practice: In the majority of cases it is not possible to explain the error rate or AUC for a particular machine learning algorithm on a particular dataset at the current state of the art. To learn more about the behavior of the abstention cost curve and the VACC measure, we performed preliminary experiments with J48 trees, varying the confidence level for pruning, and SVMs, varying the penalty/regularization parameter C. Over all datasets, we observed only small, gradual shifts in VACC and in the shape of the curves. While it is hard to detect a general pattern, it is clear that no abrupt changes occur. It was also striking to see that the changes over varying parameter values were consistent for both learning schemes. It seems that the VACC depends, to some extent, on the noise level of a dataset.

5. Related Work

The trade-off between coverage and accuracy has been addressed several times before, such as in articles by (Chow, 1970), who described an optimum rejection rule based on the Bayes optimal classifier, or (Pazzani et al., 1994), who showed that a number of machine learning algorithms can be modified to increase accuracy at the expense of abstention. Tortorella (Tortorella, 2005) and Pietraszek (Pietraszek, 2005) use ROC analysis to derive an optimal abstaining classifier from binary classifiers. Pietraszek extends the cost-based framework of Tortorella, for which a simple analytical solution can be derived, and proposes two models in which either the abstention rate or the error rate is bounded in order to deal with unknown abstention costs. Nevertheless, all of these ROC-based approaches assume at least known misclassification costs. In contrast, abstention cost curves, as shown in this paper, visualize optimal costs over a range of possible cost matrices. Ferri and Hernández-Orallo (Ferri & Hernández-Orallo, 2004) introduce additional measures of performance for, as they call it, cautious classifiers, based on the confusion matrix. Our definition of an abstention window can be considered as a special case of Ferri and Hernández-Orallo's model for the two-class case. However, no optimization is performed when creating cautious classifiers and only the trade-off between abstention rate and other performance measures such as accuracy is analyzed. Cautious classifiers can be combined in a nested cascade to create so-called delegating classifiers (Ferri et al., 2004). Cost-sensitive active classifiers (Greiner et al., 2002) are related to abstaining classifiers as they are allowed to demand values of yet unspecified attributes, before committing themselves to a class label based on costs of misclassifications and additional tests.

6. Conclusion

In this paper, we adopted a cost-based framework to analyze and visualize classifier performance when refraining from prediction is allowed. We presented a novel type of cost curves that makes it possible to compare classifiers as well as to determine the cost scenarios which favor abstention if costs are uncertain or the benefits of abstaining are unclear. In comprehensive experiments, we showed that adding abstention as another dimension, the performance of classifiers varies highly depending on datasets and costs. Viewing the optimal abstention behavior of various classifiers, we are entering largely unexplored territory. We performed preliminary experiments to shed some light on the dependency of VACC on other quantities, such as the noise level in a dataset. However, more work remains to be done to interpret the phenomena shown by the curves. Finally, we would like to note that another, more qualitative look at abstention is possible. In particular on structured data, refraining from classification is advisable if the instance to be classified is not like any other instance from the training set.

References

Chow, C. K. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, *16*, 41–46.

Drummond, C., & Holte, R. C. (2000). Explicitly representing expected cost: An alternative to ROC representation. *Proc. of the 6th International Conf. on Knowledge Discovery and Data Mining* (pp. 198–207).

Ferri, C., Flach, P., & Hernández-Orallo, J. (2004). Delegating classifiers. *Proc. of the 21st International Conf. on Machine Learning.*

Ferri, C., & Hernández-Orallo, J. (2004). Cautious classifiers. *Proceedings of the ROC Analysis in Artificial Intelligence, 1st International Workshop* (pp. 27–36).

Friedel, C. C. (2005). On abstaining classifiers. Master's thesis, Ludwig-Maximilians-Universität, Technische Universität München.

Greiner, R., Grove, A. J., & Roth, D. (2002). Learning cost-sensitive active classifiers. *Artificial Intelligence*, *139*, 137–174.

Pazzani, M. J., Murphy, P., Ali, K., & Schulenburg, D. (1994). Trading off coverage for accuracy in forecasts: Applications to clinical data analysis. *Proceedings of the AAAI Symposium on AI in Medicine* (pp. 106–110). Standford, CA.

Pietraszek, T. (2005). Optimizing abstaining classifiers using ROC analysis. *Proceedings of the* 22nd *International Conference on Machine Learning.*

Provost, F. J., & Fawcett, T. (1998). Robust classification systems for imprecise environments. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 706–713).

Tortorella, F. (2005). A ROC-based reject rule for dichotomizers. *Pattern Recognition Letters*, *26*, 167–180.

Witten, I., & Frank, E. (2005). *Data mining: Practical machine learning tools with java implementations.* Morgan Kaufmann, San Francisco.

Upper and Lower Bounds of Area Under ROC Curves and Index of Discriminability of Classifier Performance

Shane T. Mueller STMUELLE@INDIANA.EDU
Indiana University, Department of Psychological and Brain Sciences, 1101 East 10th Street, Bloomington, IN 47404

Jun Zhang JUNZ@UMICH.EDU
University of Michigan, Department of Psychology, 530 Church Street Ann Arbor, MI 48109-1043

Abstract

Area under an ROC curve plays an important role in estimating discrimination performance – a well-known theorem by Green (1964) states that ROC area equals the percentage of correct in two-alternative forced-choice setting. When only single data point is available, the upper and lower bound of discrimination performance can be constructed based on the maximum and minimum area of legitimate ROC curves constrained to pass through that data point. This position paper, after reviewing a property of ROC curves parameterized by the likelihood-ratio, presents our recently derived formula of estimating such bounds (Zhang & Mueller, 2005).

1. Introduction

Signal detection theory (Green & Swets, 1966) is commonly used to interpret data from tasks in which stimuli (e.g., tones, medical images, emails) are presented to an operator (experimenter, medical examiner, classification algorithm), who must determine which one of two categories (high or low, malignant or benign, junk or real) the stimulus belongs in. These tasks yield a pair of measures of behavioral performance: the Hit Rate (H), also called "true positive" rate, and the False Alarm Rate (F), also called "false positive" rate. (The other two rates, those of Miss or "false negative" and of Correct Rejection or "true negative", are simply one minus H and F, respectively.) H and F are typically transformed into indices of sensitivity and bias based on assumptions about an underlying statistical model. A curve $c \mapsto (F(c), H(c))$ in the ROC (Receiver-Operating Characteristic) space is a collection of hit and false-alarm rates while the operator/receiver modifies the cutoff criterion c of accepting the input stimulus as belonging to one category versus another; often c is the likelihood ratio of the evidence favoring the two corresponding hypotheses, or a monotonic transformation thereof. In the machine learning context, we map the "operator/receiver" in the SDT sense to a "classification algorithm" or simply an "algorithm", the "stimulus" as an "input instance" or simply "instance" which carries one of the two class labels, and view c as a parameter of the algorithm which biases the output of the algorithm to favor one category or the other; the optimal setting of c is related to the cost structure, i.e., individual payoffs related to correct and incorrect classifications.

A well-known result in SDT is Green's Theorem, which relates the discrimination accuracy performance of an operator to the area under the operator's (i.e., the classification algorithm's) ROC curve. This so-called ROC area is thus a compact measure of how discriminable a classification algorithm is between binary-class inputs. Consequently, the performance of different algorithms can be compared by comparing their respective ROC areas.

Often, algorithms reported in the literature may not contain a tradeoff analysis of the Hit and False Alarm rates produced by varying parameters corresponding to the algorithm's bias. In these cases, the entire ROC curve of an algorithm may not be available — in some cases, only a few or even a single point (called "data point") in the ROC space is available. In this case, performance comparison across different algorithms becomes a question of comparing areas of possible ROC curves constrained to pass through these limited data

Appearing in *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, Pittsburgh, USA, 2006.

points.

In the mathematical psychology community, the problem of estimating area of ROC curves constrained to pass through a single data point is particularly well studied (Norman, 1964; Pollack & Norman, 1964; Pollack & Hsieh, 1969; Grier, 1971; Smith, 1995; Zhang & Mueller, 2005). These estimates of the ROC area do not assume the ROC curves to arise from any specific class of parametric models, and so these estimates are often referred to as a "non-parametric" indices of an operator's discriminability (sensitivity).[1] Typically, the upper and lower bounds of discriminability were obtained by considering the maximal and minimum ROC areas among the class of "admissible" ROC curves satisfying the data constraint. Interestingly, though the basic idea was very simple and advanced over 40 years ago (Pollack & Norman, 1964), the popular formula to calculate this index (Grier, 1971), dubbed A' in psychometrics and cognitive psychology literature, turned out to be erroneous, at least insofar as its commonly understood meaning is concerned; moreover, its purported correction (Smith, 1995), dubbed A'', also contained an error. These formulae incorrectly calculated the upper bound of admissible ROC curves, using either an ROC curve that was not admissible (Pollack & Norman, 1964), or one that was not the maximum for some points (Smith, 1995). Zhang and Mueller (2005) rectified the error and gave the definite answer to the question of non-parametric index of discriminability based on ROC areas.

In this note, we first review the notion of "proper" (or "admissible") ROC curves and prove a lemma basically stating that all ROC curves are proper/admissible when the likelihood functions (for the two hypotheses) used to construct the ROC curve are parameterized by the likelihood ratio (of those hypotheses). We then review Green's Theorem, which related area under an ROC curve to percentage correct in a two-alternative discrimination task. Finally, we present the upper and lower bounds on a 1-point constrained ROC area and reproduce some of the basic arguments underlying their derivation. All technical contents were taken from Zhang and Mueller (2005).

2. Slope of ROC curve and likelihood ratio

Recall that, in the traditional signal detection framework, an ROC curve $u_c \mapsto (F(u_c), H(u_c))$ is parameterized by the cutoff criteria value u_c along the measurement (evidence) axis based on which categorization decision is made. Given underlying signal distribution $f_s(u)$ and noise distribution $f_n(u)$ of measurement value u[2], a criterion-based decision rule, which dictates a "Yes" decision if $u > u_c$ and a "No" decision if $u < u_c$, will give rise to

$$\begin{aligned} H(u_c) &= \Pr(\text{Yes}|s) = \Pr(u > u_c|s) = \int_{u_c}^{\infty} f_s(u)du, \\ F(u_c) &= \Pr(\text{No}|s) = \Pr(u > u_c|n) = \int_{u_c}^{\infty} f_n(u)du. \end{aligned} \tag{1}$$

As u_c varies, so do H and F; they trace out the ROC curve. Its slope is

$$\left.\frac{dH}{dF}\right|_{F=F(u_c),H=H(u_c)} = \frac{H'(u_c)}{F'(u_c)} = \frac{f_s(u_c)}{f_n(u_c)} \equiv l(u_c)\ .$$

With an abuse of notation, we simply write

$$\frac{dH(u)}{dF(u)} = l(u)\,. \tag{2}$$

Note that in the basic setup, the likelihood ratio $l(u)$ as a function of decision criterion u (whose optimal setting depends on the prior odds and the payoff structure) need not be monotonic. Hence, the ROC curve $u \mapsto (F(u), H(u))$ need not be concave. We now introduce the notion of "proper (or admissible) ROC curves".

Definition 2.1. A *proper* (or *admissible*) ROC curve is a piece-wise continuous curve defined on the unit square $[0,1] \times [0,1]$ connecting the end points (0,0) and (1,1) with non-increasing slope.

The shape of a proper ROC curve is necessarily concave (downward-bending) connecting (0,0) and (1,1). It necessarily lies above the line $H = F$. Next we provide a sufficient and necessary condition for an ROC curve to be proper/admissible, that is, a concave function bending downward.

Lemma 2.2. An ROC curve is proper if and only if the likelihood ratio $l(u)$ is a non-decreasing function of decision criterion u.

[1] Though no parametric assumption is invoked in the derivation of these indices, the solution itself may correspond to certain models of underlying likelihood process, see MacMillan and Creelman, 1996. In other words, parameter-free here does not imply model-free.

[2] In machine learning applications, "signal" and "noise" simply refer the two category classes of inputs, and "signal distribution" and "noise distribution" are likelihood functions of the two classes.

Proof. Differentiate both sides of (2) with respect to u

$$\frac{dF}{du} \cdot \frac{d}{dF}\left(\frac{dH}{dF}\right) = \frac{dl}{du}.$$

Since, according to (1)

$$\frac{dF}{du} = -f_n(u) < 0,$$

therefore

$$\frac{dl}{du} \geq 0 \Longleftrightarrow \frac{d}{dF}\left(\frac{dH}{dF}\right) \leq 0$$

indicating that the slope of ROC curve is non-increasing, i.e., the ROC curve is proper. $\diamond$

Now it is well known (see Green & Swets, 1966) that a monotone transformation of measurement axis $u \mapsto v = g(u)$ does not change the shape of the ROC curve (since it is just a re-parameterization of the curve), so a proper ROC curve will remain proper after any monotone transformation. On the other hand, when $l(u)$ is not monotonic, one wonders whether there always exists a parameterization of any ROC curve to turn it into a proper one. Proposition 1 below shows that the answer is positive — the parameterization of the two likelihood functions is to use the likelihood ratio itself!

Proposition 2.3. (Slope monotonicity of ROC curves parameterized by likelihood-ratio). The slope of an ROC curve generated from a pair of likelihood functions $(F(l_c), H(l_c))$, when parameterized by the likelihood-ratio l_c as the decision criterion, equals the likelihood-ratio value at each criterion point l_c

$$\frac{dH(l_c)}{dF(l_c)} = l_c. \tag{3}$$

Proof. When likelihood-ratio l_c is used the decision cutoff criterion, the corresponding hit rate (H) and false-alarm rate (F) are

$$\begin{aligned} H(l_c) &= \int_{\{u:l(u)>l_c\}} f_s(u)du, \\ F(l_c) &= \int_{\{u:l(u)>l_c\}} f_n(u)du. \end{aligned}$$

Note that here u is to be understood as (in general) a multi-dimensional vector, and du should be understood accordingly. Writing out $H(l_c + \delta l) - H(l_c) \equiv \delta H(l_c)$ explicitly,

$$\delta H(l_c) = \int_{\{u:l(u)>l_c+\delta l\}} f_s(u)du - \int_{\{u:l(u)>l_c\}} f_s(u)du$$
$$= -\int_{\{u:l_c<l(u)<l_c+\delta l\}} f_s(u)du \simeq -\int_{\{u:l(u)=l_c\}} f_s(u)\,\delta u$$

where the last integral $\int \delta u$ is carried out on the set $\partial \equiv \{u : l(u) = l_c\}$, i.e., across all u's that satisfy $l(u) = l_c$ with given l_c. Similarly,

$$\delta F(l_c) \simeq -\int_{\{u:l(u)=l_c\}} f_n(u)\,\delta u\,.$$

Now, for all $u \in \partial$

$$\frac{f_s(u)}{f_n(u)} = l(u) = l_c$$

is constant, from an elementary theorem on ratios, which says that if $a_i/b_i = c$ for $i \in I$ (where c is a constant and I is an index set), then $(\sum_{i\in I} a_i)/(\sum_{i\in I} b_i) = c$,

$$\frac{\delta H(l_c)}{\delta F(l_c)} = \frac{\int_\partial f_s(u)\,\delta u}{\int_\partial f_n(u)\,\delta u} = \left.\frac{f_s(u)\,\delta u}{f_n(u)\,\delta u}\right|_{u\in\partial} = l_c\,.$$

Taking the limit $\delta l \to 0$ yields (3). $\diamond$.

Proposition 2.3 shows that the slope of ROC curve is always equal the likelihood-ratio value regardless how it is parameterized, i.e., whether the likelihood-ratio is monotonically or non-monotonically related to the evidence u and whether u is uni- or multi-dimensional. The ROC curve is a signature of a criterion-based decision rule, as captured succinctly by the expression

$$\frac{dH(l)}{dF(l)} = l\,.$$

Since $H(l)$ and $F(l)$ give the proportion of hits and false alarms when a decision-maker says "Yes" whenever the likelihood-ratio (of the data) exceeds l, then $\delta H = H(l + \delta l) - H(l)$, $\delta F = F(l + \delta l) - F(l)$ are the amount of hits and false-alarms if he says "Yes" only when the likelihood-ratio falls within the interval $(l, l+\delta l)$. Their ratio is of course simply the likelihood-ratio.

Under the likelihood-ratio parameterization, the signal distribution $f_s(l) = -dH/dl$ and the noise distribution $f_n(l) = -dF/dl$ can be shown to satisfy

$$E_s\{l\} = \int_{l=0}^{l=\infty} l f_s(l)dl \geq 1 = \int_{l=0}^{l=\infty} l f_n(l)dl = E_n\{l\}.$$

The shape of the ROC curve is determined by $H(l)$ or $F(l)$. In fact, its curvature is

$$\kappa = \frac{d}{dl}\left(\frac{dH}{dF}\right) / \left(1 + \left(\frac{dH}{dF}\right)^2\right) = \frac{1}{1+l^2}\,.$$

3. Green's Theorem and area under ROC curves

The above sections studies the likelihood-ratio classifier in a single-instance paradigm — upon receiving an input instance, the likelihood functions in favor of each hypothesis are evaluated and compared with a pre-set criterion to yield a decision of class label. Both prior odds and payoff structure can affect the optimal setting of likelihood ratio criterion l_c by which class label is assigned. On the other hand, in two-alternative force choice paradigms with two two instances, each instance is drawn from one category, and the operator must match them to their proper categories. For example, an auditory signal may be present in one of two temporal intervals, and the operator must determine which interval contains the signal and which contains noise. In this case, the likelihood-ratio classifier, after computing the likelihood-ratios for each of the instances, simply compares the two likelihood-ratio values l_a and l_b, and matches them to the two class labels based on whether $l_a < l_b$ or $l_a > l_b$. It turns out that the performance of the likelihood-ratio classifier under the single-instance paradigm ("detection paradigm") and under the two-instance forced-choice paradigm ("identification paradigm") are related by a theorem first proven by Green (1964).

Proposition 3.1. (Green, 1964). Under the likelihood-ratio classifier, the area under an ROC curve in a single-observation classification paradigm is equal to the overall probability correct in the two-alternative force choice paradigm.

Proof. Following the decision rule of the likelihood-ratio classifier, the percentage of correctly ("PC") matching the two input instances to the two categories is

$$\begin{aligned}\text{PC} &= \int\int_{0\le l_b \le l_a \le \infty} f_s(l_a)\, f_n(l_b)\, dl_a\, dl_b \\ &= \int_0^\infty \left(\int_{l_b}^\infty f_s(l_a)\, dl_a\right) f_n(l_b)\, dl_b \\ &= \int_{l_b=0}^{l_b=\infty} H(l_b)\, dF(l_b) = \int_{F=0}^{F=1} H dF,\end{aligned}$$

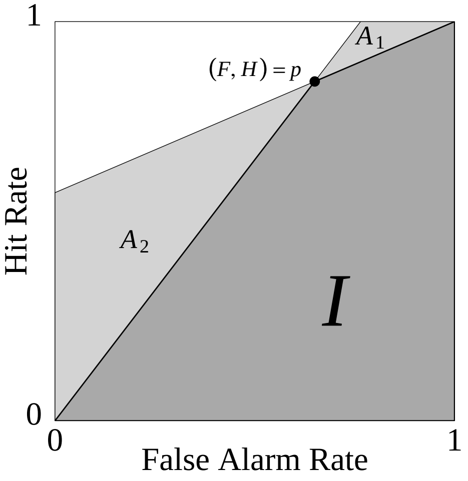
which is the area under the ROC curve $l_c \mapsto (F(l_c), H(l_c))$. ⋄

Green's Theorem (Proposition 3.1) motivates one to use the area under an ROC curve to as a measure of discriminability performance of the operator. When multiple pairs of hit and false alarm rates $(F_i, H_i)_{i=1,2,\cdots}$ (with $F_1 < F_2 < \cdots, H_1 < H_2 < \cdots$) are available, all from the same operator but under manipulation of prior odds and/or payoff structure and with the constraints

$$0 \le \cdots \le \frac{H_3 - H_2}{F_3 - F_2} \le \frac{H_2 - H_1}{F_2 - F_1} \le \infty,$$

then it is possible to construct proper ROC curves passing through these points, and the bounds for their area can be constructed. The question of finding the areal bounds of ROC curves passing through a single data point has received special attention in the past (since Norman, 1964), because as more data points are added, the uncertain in ROC area (difference between the upper and lower bounds of area measure) decreases. We discuss the bounds of 1-point constrained ROC area in the next sections.

Figure 1. Proper ROC curves through point p must lie within or on the boundaries of the light shaded regions A_1 and A_2. The minimum-area proper ROC curve through p lies on the boundary of region I.

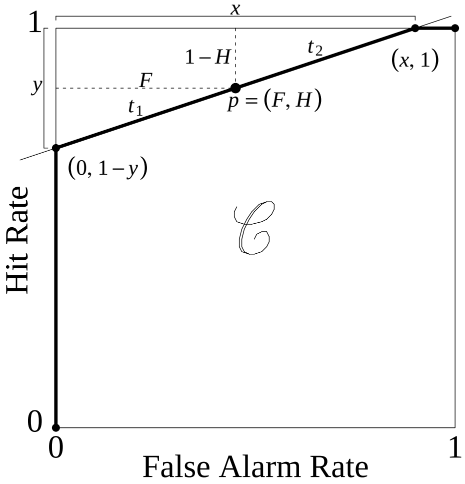


4. ROC curves constrained to pass through a data point

When the data point $p = (F, H)$ is fixed, the non-increasing property of the slope (Corollary 1) immediately leads to the conclusion that all proper ROC curves must fall within or on the bounds of light shaded regions A_1 and A_2 (shown in Figure 1). This observation was first made in Norman (1964). The proper ROC curve with the smallest area lies on the boundary between I and A_1 (to the right of p) and A_2 (to the left of p), whereas the proper ROC curve with the largest area lies within or on the boundaries of A_1 and A_2.

Pollack and Norman (1964) proposed to use the average of the areas $A_1 + I$ and $A_2 + I$ as an index of discriminability (so-called A'), which turns out to equal

Figure 2. Example of a proper ROC curve through p. The ROC curve $\mathcal{C}$, a piecewise linear curve denoted by the dark outline, is formed by following a path from $(0, 0)$ to $(0, 1-y)$ to $(x, 1)$ (along a straight line that passes through $p = (F, H)$) and on to $(1, 1)$.



$1/2 + (H-F)(1+H-F)/(4H(1-F))$ (Grier, 1971). However, the A' index was later mistakenly believed to represent the *average* of the maximal and minimum areas of proper ROC curves constrained to pass through $p = (F, H)$. Rewriting

$$\frac{1}{2}((A_1 + I) + (A_2 + I)) = \frac{1}{2}(I + (A_1 + A_2 + I)),$$

the mis-conceptualization probably arose from (incorrectly) taking the area $A_1 + A_2 + I$ to be the maximal area of 1-point constrained proper ROC curves while (correcting) taking the are I to be the minimal area of such ROC curves, see Figure 1. It was Smith (1995) who first pointed out this long, but mistakenly-held belief, and proceeded to derive the true upper bound (maximal area) of proper ROC curves, to be denoted A_+. Smith claimed that, depending on whether p is to the left or right of the negative diagonal $H + F = 1$, A_+ is the larger of $I + A_1$ and $I + A_2$. This conclusion, unfortunately, is still erroneous when p is in the upper left quadrant of ROC space (i.e., $F < .5$ and $H > .5$) — in this region, neither $I + A_1$ nor $I + A_2$ represents the upper bound of all proper ROC curves passing through p.

5. Lower and upper bound of area of 1-point constrained proper ROC curves

The lower bound A_- of the area of all proper ROC curves constrained to pass through a given point $p = (F, H)$ can be derived easily (the area labelled as I in Figure 1):

$$A_- = \frac{1}{2}(1 + H - F).$$

In Zhang and Mueller (2005), the expression was derived for the upper bound A_+ of such ROC area.

PROPOSITION 5.1. (Upper Bound of ROC Area). The areal upper bound A_+ of proper ROC curves constrained to pass through one data point $p = (F, H)$ is

$$A_+ = \begin{cases} 1 - 2H(1-F) & \text{if} \quad F < 0.5 < H\,, \\ \frac{1-F}{2H} & \text{if} \quad F < H < 0.5\,, \\ 1 - \frac{1-H}{2(1-F)} & \text{if} \quad 0.5 < F < H\,. \end{cases}$$

Proof. See Zhang and Mueller (2005). ⋄

The ROC curve achieving the maximal area generally consists of three segments (as depicted in Figure 2), with the data point p *bisecting* the middle segment – in other words, $t_1 = t_2$ in Figure 2. When p falls in the $F < H < 0.5$ ($0.5 < F < H$, resp) region, then the vertical (horizontal, resp) segment of the maximal-area ROC curve degenerates to the end point $(0, 0)$ ($(1, 1)$, resp), corresponding to $y = 1$ ($x = 1$, resp) in Figure 2.

With the upper and lower bounds on ROC area derived, Figure 3 plots the difference between these bounds — that is, the uncertainty in the area of proper ROC curves that can pass through each point. The figure shows that the smallest differences occur along the positive and negative diagonals of ROC space, especially for points close to $(0, 1)$ and $(.5, .5)$. The points where there is the greatest difference between the lower and upper bounds of ROC area are near the lines $H = 0$ and $F = 1$. Thus, data observed near these edges of ROC space can be passed through by proper ROC curves with a large variability of underlying areas. Consequently, care should be taken when trying to infer the ROC curve of the observer/algorithm when the only known data point regarding its performance (under a single parameter setting) falls within this region.

By averaging the upper and lower bound $A = (A_+ + A_-)/2$, we can derive the (non-parametric) index of discriminability performance

$$A = \begin{cases} \frac{3}{4} + \frac{H-F}{4} - F(1-H) & \text{if} \quad F \le 0.5 \le H\,; \\ \frac{3}{4} + \frac{H-F}{4} - \frac{F}{4H} & \text{if} \quad F < H < 0.5\,; \\ \frac{3}{4} + \frac{H-F}{4} - \frac{1-H}{4(1-F)} & \text{if} \quad 0.5 < F < H\,. \end{cases}$$

One way to examine A is to plot the "iso-discriminability" curve, i.e, the combinations of F and

Figure 3. Difference between the lower and upper bounds of area of proper ROC curves through every point in ROC space. Lighter regions indicate smaller differences.

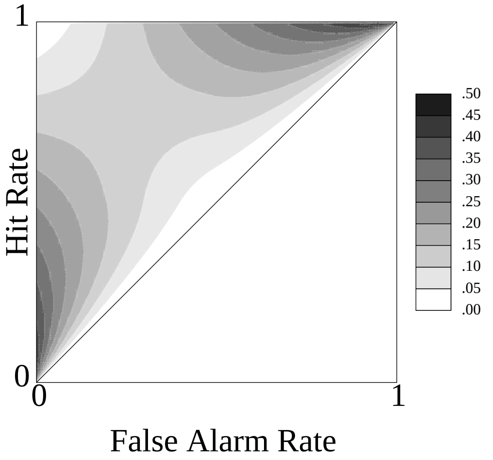
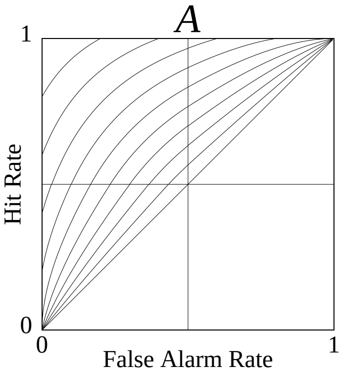


Figure 4. Iso-discriminability contours in ROC space. Each line corresponds to combinations of F and H that produce equal values of A, in increments of 0.05.



H will produce a given value of A. The topography of A in ROC space can be mapped by drawing isopleths for its different constant values. Figure 4 shows these topographic maps for A.

Finally, since the slope of any proper ROC curve is related to the likelihood ratio of the underlying distributions, we can construct an index of decision bias (Zhang & Mueller, 2005), denoted b, as being orthogonal to the slope of the constant-A curve (called b):

$$b = \begin{cases} \frac{5-4H}{1+4F} & \text{if} \quad F \leq 0.5 \leq H\,; \\ \frac{H^2+H}{H^2+F} & \text{if} \quad F < H < 0.5\,; \\ \frac{(1-F)^2+1-H}{(1-F)^2+1-F} & \text{if} \quad 0.5 < F < H\,. \end{cases}$$

6. Conclusion

We showed that the relationship of ROC slope to likelihood-ratio is a fundamental relation in ROC analysis, as it is invariant with respect to any continuous reparameterization of the stimulus, including non-monotonic mapping of uni-dimensional and multi-dimensional evidence in general. We provided an upper bound for the area of proper ROC curves passing through a data point and, together with the known lower bound, a non-parametric estimate of discriminability as defined by the average of maximal and minimum ROC areas.

References

Green, D. M. (1964). General prediction relating yes-no and forced-choice results. *Journal of the Acoustical Society of America, A*, *36*, 1024.

Green, D. M., & Swets, J. A. (1964). *Signal detection theory and psychophysics.* New York: John Wiley & Sons.

Grier, J. B. (1971). Nonparametric indexes for sensitivity and bias: computing formulas. *Psychological Bulletin*, *75*, 424–429.

Macmillan, N. A., & Creelman, C. D. (1996). Triangles in roc space: History and theory of "non-parametric" measures of sensitivity and response bias. *Psychonomic Bulletin & Review*, *3*, 164–170.

Norman, D. A. (1964). A comparison of data obtained with different false-alarm rates. *Psychological Review*, *71*, 243–246.

Peterson, W. W., Birdsall, T. G., & Fox, W. C. (1954). The theory of signal detectability. *Transactions of the IRE Professional Group on Information Theory* (pp. 171–212).

Pollack, I., & Hsieh, R. (1969). Sampling variability of the area under roc curve and d'_e. *Psychological Bulletin*, *1*, 161–173.

Pollack, I., & Norman, D. A. (1964). Non-parametric analysis of recognition experiments. *Psychonomic Science*, *1*, 125–126.

Smith, W. D. (1995). Clarification of sensitivity measure A'. *Journal of Mathematical Psychology*, *39*, 82–89.

Zhang, J., & Mueller, S. T. (2005). A note on roc analysis and non-parametric estimate of sensitivity. *Psychometrika*, *70*, 145–154.

Applying REC Analysis to Ensembles of Sigma-Point Kalman Filters

Aloísio Carlos de Pina LONG@COS.UFRJ.BR
Gerson Zaverucha GERSON@COS.UFRJ.BR
Department of Systems Engineering and Computer Science, COPPE/PESC, Federal University of Rio de Janeiro, C.P.68511 - CEP. 21945-970, Rio de Janeiro, RJ, Brazil

Abstract

The Sigma-Point Kalman Filters (SPKF) is a family of filters that achieve very good performance when applied to time series. Currently most researches involving time series forecasting use the Sigma-Point Kalman Filters, however they do not use an ensemble of them, which could achieve a better performance. The REC analysis is a powerful technique for visualization and comparison of regression models. The objective of this work is to advocate the use of REC curves in order to compare the SPKF and ensembles of them and select the best model to be used.

1. Introduction

In the past few years, several methods for time series prediction were developed and compared. However, all these studies based their conclusions on error comparisons.

Results achieved by Provost, Fawcett and Kohavi (1998) raise serious concerns about the use of accuracy, both for practical comparisons and for drawing scientific conclusions, even when predictive performance is the only concern. They indicate ROC analysis (Provost & Fawcett, 1997) as a superior methodology than the accuracy comparison in the evaluation of classification learning algorithms. Receiver Operating Characteristic (ROC) curves provide a powerful tool for visualizing and comparing classification results. A ROC graph allows the performance of multiple classification functions to be visualized and compared simultaneously and the area under the ROC curve (AUC) represents the expected performance as a single scalar.

But ROC curves are limited to classification problems. Regression Error Characteristic (REC) curves (Bi & Bennett, 2003) generalize ROC curves to regression with similar benefits. As in ROC curves, the graph should characterize the quality of the regression model for different levels of error tolerance.

The Sigma-Point Kalman Filters (SPKF) (van der Merwe & Wan, 2003) is a family of filters based on derivativeless statistical linearization. It was shown that Sigma-Point Kalman Filters achieve very good performance when applied to time series (van der Merwe & Wan, 2003).

Current research on time series forecasting mostly relies on use of Sigma-Point Kalman Filters, achieving high performances. Although most of these works use one of the filters from the SPKF family, they do not use an ensemble (Dietterich, 1998) of them, which could achieve a better performance. Therefore, the main goal of this paper is to advocate the use of REC curves in order to compare ensembles of Sigma-Point Kalman Filters and choose the best model to be used with each time series.

This paper is organized as follows. The next section has a brief review of REC curves. Then, a summary of the main characteristics of the Sigma-Point Kalman Filters is presented in Section 3. An experimental evaluation comparing the REC curves provided by each algorithm and ensembles of them is reported in Section 4. Finally, in Section 5, the conclusions and the plans for future research are presented.

2. Regression Error Characteristic Curves

Results achieved by Provost, Fawcett and Kohavi (1998) indicate ROC analysis (Provost & Fawcett, 1997) as a superior methodology to the accuracy comparison in the evaluation of classification learning algorithms. But ROC curves are limited to classification problems. Regression Error Characteristic (REC) curves (Bi & Bennett, 2003) generalize ROC curves to regression with similar benefits.

The REC curve is a technique for evaluation and comparison of regression models that facilitates the visualization of the performance of many regression functions simultaneously in a single graph. A REC graph contains one or more monotonically increasing curves (REC curves), each corresponding to a single regression model.

Appearing in Proceedings of the third Workshop on ROC Analysis in Machine Learning, Pittsburgh, USA, 2006. Copyright 2006 by the author(s)/owner(s).

One can easily compare many regression functions by examining the relative position of their REC curves. The shape of the curve reveals additional information that can be used to guide modeling.

REC curves plot the error tolerance on the x-axis and the accuracy of a regression function on the y-axis. Accuracy is defined as the percentage of points predicted within the tolerance. A good regression function provides a REC curve that climbs rapidly towards the upper-left corner of the graph, in other words, the regression function achieves high accuracy with a low error tolerance.

In regression, the residual is the analogous concept to the classification error in classification. The residual is defined as the difference between the predicted value $f(\mathrm{x})$ and actual value y of response for any point (x, y). It could be the squared error $(y - f(\mathrm{x}))^2$ or absolute deviation $| y - f(\mathrm{x}) |$ depending on the error metric employed. Residuals must be greater than a tolerance e before they are considered as errors.

The area over the REC curve (AOC) is a biased estimate of the expected error for a regression model. It is a biased estimate because it always underestimates the actual expectation. If e is calculated using the absolute deviation (AD), then the AOC is close to the mean absolute deviation (MAD). If e is based on the squared error (SE), the AOC approaches the mean squared error (MSE). The evaluation of regression models using REC curves is qualitatively invariant to the choices of error metrics and scaling of the residual. The smaller the AOC is, better the regression function will be. However, two REC curves can have equal AOC's but have different behaviors. The one who climbs faster towards the upper-left corner of the graph (in other words, the regression function that achieves higher accuracy with a low error tolerance) may be preferable. This kind of information can not be provided by the analysis of an error measure.

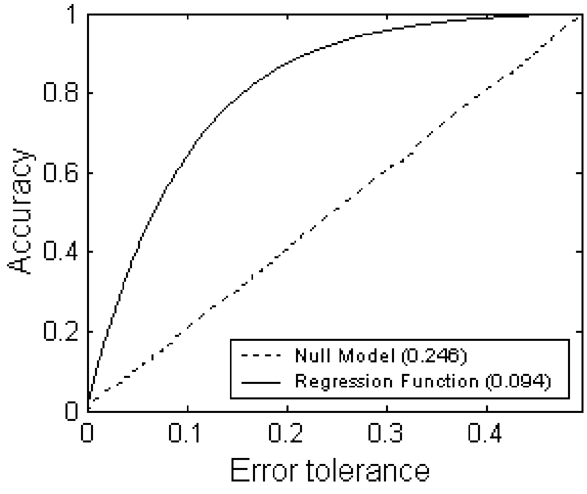


Figure 1. Example of REC graph.

In order to adjust the REC curves in the REC graph, a null model is used to scale the REC graph. Reasonable regression approaches produce regression models that are better than the null model. The null model can be, for instance, the mean model: a constant function with the constant equal to the mean of the response of the training data.

An example of REC graph can be seen in Figure 1. The number between parentheses in the figure is the AOC value for each REC curve. A regression function dominates another one if its REC curve is always above the REC curve corresponding to the other function. In the figure, the regression function dominates the null model, as should be expected.

3. Sigma-Point Kalman Filters

It is known that for most real-world problems, the optimal Bayesian recursion is intractable. The Extended Kalman Filter (EKF) (Jazwinsky, 1970) is an approximate solution that has become one of the most widely used algorithms with several applications.

The EKF approximates the state distribution by a Gaussian random variable, which is then propagated through the “first-order” linearization of the system. This linearization can introduce large errors which can compromise the accuracy or even lead to divergence of any inference system based on the EKF or that uses the EKF as a component part.

The Sigma-Point Kalman Filters (SPKF) (van der Merwe & Wan, 2003), a family of filters based on derivativeless statistical linearization, achieve higher performance than EKF in many problems and are applicable to areas where EKFs can not be used.

Instead of linearizing the nonlinear function through a truncated Taylor-series expansion at a single point (usually the mean value of the random variable), SPKF rather linearize the function through a linear regression between r points, called sigma-points, drawn from the prior distribution of the random variable, and the true nonlinear functional evaluations of those points. Since this statistical approximation technique takes into account the statistical properties of the prior random variable the resulting expected linearization error tends to be smaller than that of a truncated Taylor-series linearization.

The way that the number and the specific location of the sigma-points are chosen, as well as their corresponding regression weights, differentiate the SPKF variants from each other. The SPKF Family is composed by four algorithms: Unscented Kalman Filter (UKF), Central Difference Kalman Filter (CDKF), Square-root Unscented Kalman Filter (SR-UKF) and Square-root Central Difference Kalman Filter (SR-CDKF).

Now we will present a brief overview of the main characteristics of the Sigma-Point Kalman Filters. See (van der Merwe & Wan, 2003) for more details.

3.1 The Unscented Kalman Filter

The Unscented Kalman Filter (UKF) (Julier, Uhlmann & Durrant-Whyte, 1995) derives the location of the sigma-points as well as their corresponding weights so that the sigma-points capture the most important statistical properties of the prior random variable x. This is achieved by choosing the points according to a constraint equation which is satisfied by minimizing a cost-function, whose purpose is to incorporate statistical features of x which are desirable, but do not necessarily have to be met. The necessary statistical information captured by the UKF is the first and second order moments of $p(x)$.

3.2 The Central Difference Kalman Filter

The Central Difference Kalman Filter (CDKF) (Ito & Xiong, 2000) is another SPKF implementation, whose formulation was derived by replacing the analytically derived first and second order derivatives in the Taylor series expansion by numerically evaluated central divided differences. The resulting set of sigma-points for the CDKF is once again a set of points deterministically drawn from the prior statistics of x. Studies (Ito & Xiong, 2000) have shown that in practice, just as UKF, the CDKF generates estimates that are clearly superior to those calculated by an EKF.

3.3 Square-Root Forms of UKF and CDKF

SR-UKF and SR-CDKF (van der Merwe & Wan, 2001) are numerically efficient square-root forms derived from UKF and CDKF respectively. Instead of calculating the matrix square-root of the state covariance at each time step (a very costly operation) in order to buid the sigma-point set, these forms propagate and update the square-root of the state covariance directly in Cholesky factored form, using linear algebra techniques. This also provides more numerical stability.

The square-root SPKFs (SR-UKF and SR-CDKF) achieve equal or slightly higher accuracy when compared to the standard SPKFs. Besides, they have lower computational cost and a consistently increased numerical stability.

4. Experimental Evaluation

Since the experiments described in (Bi & Bennett, 2003) used just one data set and their results were only for REC demonstration, we first did tests with two well-known regression algorithms using 25 regression problems, in order to better evaluate the REC curves as a tool for visualizing and comparing regression learning algorithms.

Then we present the results of the comparison by using REC curves of SPKFs and EKF applied to time series and finally we investigate the use of an ensemble method (stacking (Wolpert, 1992)) with the tested models, evaluating it with REC curves, as suggested by Bi and Bennett (2003). In this work, 12 time series with real-world data were used in order to try to establish a general ranking among the models tested. The names and sizes of the used time series are shown in Table 1. All data are differentiated and then the values are rescaled linearly to between 0.1 and 0.9. As null model we choose the mean model, a constant function with the constant equal to the mean of the response of the training data.

Table 1. Time series used in the experimental evaluation.

Time series	Data points
A[1]	1000
Burstin[2]	2001
Darwin[2]	1400
Earthquake[2]	2097
Leuven[3]	2000
Mackey-Glass[4]	300
Series 1[5]	96
Series 2 [5]	96
Series 3 [5]	96
Soiltemp[2]	2306
Speech[2]	1020
Ts1 [2]	1000

4.1 Preliminary Results with Regression

Initial experiments were carried out in order to reinforce the conclusions reached out by Bi and Bennett (2003) in favor of the use of REC curves as a mean to compare regression algorithms (similarly to arguments for ROC curves in classification).

We have used REC curves in order to compare the performance of the Naive Bayes for Regression (Frank, Trigg, Holmes & Witten, 2000) to the performance of Model Trees (Quinlan, 1992). Naive Bayes for Regression (NBR) uses the Naive Bayes methodology for numeric prediction tasks by modeling the probability distribution of the target value with kernel density estimators. Model Tree predictor is a state-of-the-art method for regression. Model trees are the counterpart of

[1] Data from a competition sponsored by the Santa Fe Institute. (http://www-psych.stanford.edu/%7Eandreas/Time-Series/SantaFe)

[2] Data from the UCR Time Series Data Mining Archive (Keogh & Folias, 2002).

[3] Data from the K.U. Leuven competition. (ftp://ftp.esat.kuleuven.ac.be/pub/sista/suykens/workshop/datacomp.dat)

[4] Numerical solution for the Mackey-Glass delay-differential equation.

[5] Data of monthly electric load forecasting from Brazilian utilities (Teixeira & Zaverucha, 2003).

decision trees for regression tasks. They have the same structure as decision trees, but employ linear regression at each leaf node to make a prediction. In (Frank, Trigg, Holmes & Witten, 2000) an accuracy comparison of these two learning algorithms is presented and its results show that Model Trees outperform NBR significantly for almost all data sets tested.

The 25 regression data sets used in this study were obtained from the UCI Repository of Machine Learning Databases (Blake & Merz, 2006). With 16 of the data sets the Model Tree predictor clearly outperforms NBR, as can be seen, for instance, in Figure 2. The number between parentheses in the figure is the AOC value for each REC curve. Note that the REC curve for Model Tree covers completely the REC curve for NBR, becoming clear the superiority of the former algorithm when applied to this specific data set.

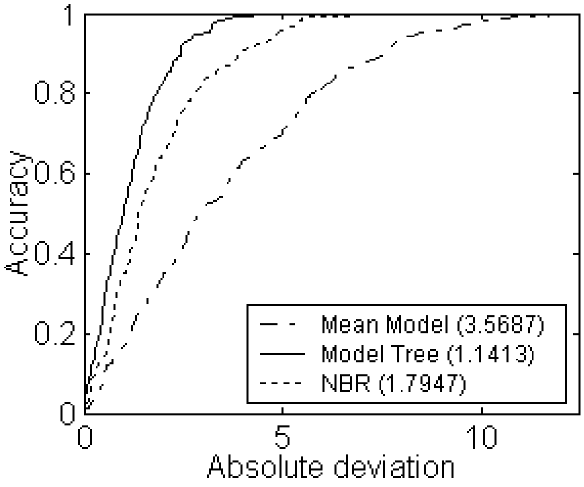


Figure 2. REC graph used to compare the performances of NBR and Model Tree when applied to data set pwLinear.

4.2 Comparing SPKFs by means of REC Curves

First, we have compared UKF and CDKF with their square-root forms, SR-UKF and SR-CDKF respectively. As expected, the REC curves for UKF and for SR-UKF are very similar. This means that the difference between the performances of the models provided by UKF and SR-UKF was negligible. The same fact could be verified with the REC curves for CDKF and SR-CDKF. Therefore, because of these results and the other advantages mentioned before in Section 3, we have continued our experiments only with the square-root forms of the SPKF.

By analyzing the generated REC graphs, we could verify that, for most time series, the model provided by SR-UKF dominates the models provided by SR-CDKF and EKF, that is, the REC curve for the SR-UKF model is always above the REC curves for SR-CDKF and EKF. Therefore, the model provided by SR-UKF would be preferable. An example is shown in Figure 3.

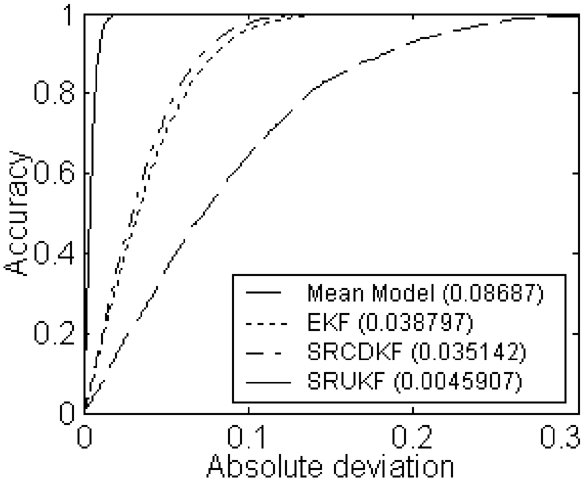


Figure 3. EKF and SPKFs applied to Burstin time series.

SR-UKF was outperformed by SR-CDKF only for the Mackey-Glass time series (Figure 4). Note that the curves cross each other at error tolerance of 0.7. SR-CDKF and EKF achieved similar performances for almost all time series, as can be seen, for instance, in Figure 5. However, the analysis of the AOC's gives a small advantage to SR-CDKF. The lower performance of EKF when compared to the others is probably caused by the non-linearity of the series. Therefore, SR-UKF consistently showed to be the best alternative to use with these series, followed by SR-CDKF and EKF, in this order. The Model Tree predictor and NBR were also tested for the prediction of the time series, but both provided poor models.

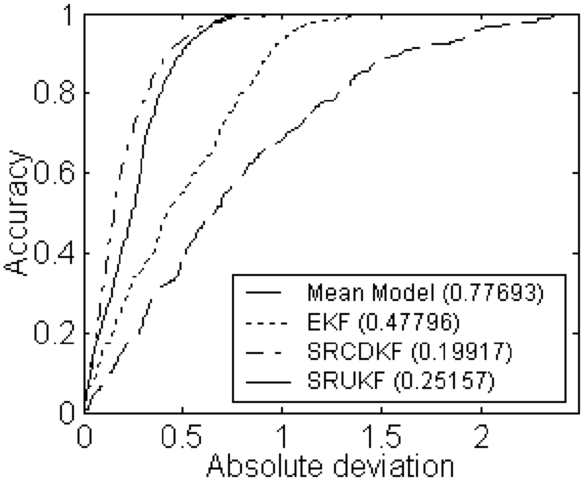


Figure 4. EKF and SPKFs applied to Mackey-Glass time series.

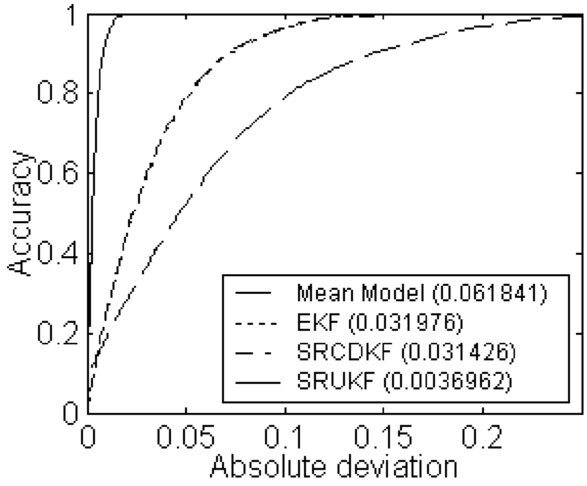


Figure 5. EKF and SPKFs applied to Earthquake time series.

4.3 Stacking of Sigma-Point Kalman Filters

Stacking (Wolpert, 1992) is an ensemble method (Dietterich, 1998) used to combine different learning algorithms. It works as follows. Suppose we have a set of different learning algorithms and a set of training examples. Each of these algorithms, called base learners, is applied to the training data in order to produce a set of hypotheses. The results computed by this set of hypotheses are combined into new instances, called meta-instances. Each "attribute" in the meta-instance is the output of one of the base learners and the class value is the same of the original instance. Another learning algorithm, called meta-regressor (or meta-classifier, for classification), is trained and tested with the meta-instances and provides the final result of the stacking.

We have used stacking to build ensembles of SPKFs and EKF. A Model Tree predictor was chosen as a meta-regressor not only because it achieved good results in the initial experiments, but also because it is a state-of-the-art regression method and it has already been successfully used as a meta-classifier for stacking (Dzeroski & Zenko, 2004), outperforming all the other combining methods tested.

Table 2. Stackings built.

Stackings	Base learners
Stacking 1	EKF, SR-CDKF
Stacking 2	EKF, SR-UKF
Stacking 3	SR-CDKF, SR-UKF
Stacking 4	EKF, SR-CDKF, SR-UKF

In order to determine which subset of algorithms can provide the best ensemble, we built four models by stacking: one containing the square-root SPKFs and EKF, and the others leaving one of them out. If we were testing several algorithms we could use a method to build the ensembles (Caruana & Niculescu-Mizil, 2004). Table 2 shows the stackings built: Stacking 1 is composed by EKF and SR-CDKF, Stacking 2 is composed by EKF and SR-UKF, Stacking 3 is composed by SR-CDKF and SR-UKF, and Stacking 4 is composed by EKF, SR-CDKF and SR-UKF. The REC curves show that all stackings that have the SR-UKF as a base learner achieve similar high performances. This can be seen, for example, in Figure 6.

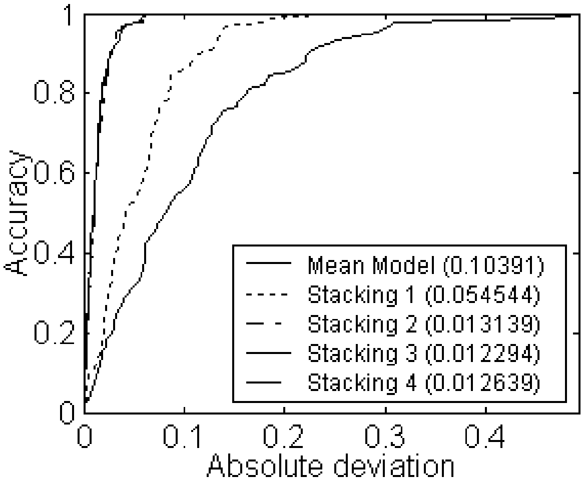


Figure 6. Stackings applied to Series 2 time series.

Table 3 shows the AOC values of the REC curves provided for the stackings with SR-UKF as a base learner. By analyzing the values we can see that among the three stackings that contain the SR-UKF, those who have SR-CDKF as a base learner achieve a slightly better performance. Since the number of time series for which Stacking 3 achieved the best performance is almost the same number of time series for which Stacking 4 was the best, we have considered that the inclusion of EKF as a base learner does not compensate the overhead in terms of computational cost. Thus, the model chosen as the best is that provided by Stacking 3 (SR-CDKF and SR-UKF as base learners).

Table 3. AOC's of the REC curves provided for the stackings with SR-UKF as a base learner.

Time series	Stacking 2	Stacking 3	Stacking 4
A	0.001366	0.001497	**0.001310**
Burstin	0.001740	**0.001613**	0.001740
Darwin	**0.013934**	0.014069	0.014052
Earthquake	0.000946	**0.000943**	0.000946
Leuven	0.005172	0.005190	**0.005142**
Mackey-Glass	0.228064	0.133420	**0.128672**
Series 1	0.001167	0.001306	**0.001111**
Series 2	0.013139	**0.012294**	0.012639
Series 3	0.000800	**0.000717**	0.000767
Soiltemp	0.000884	**0.000780**	0.000782
Speech	0.000714	0.000713	**0.000706**
Ts1	0.005010	0.005044	**0.004881**

By comparing the best stacking model (SR-CDKF and SR-UKF as base learners and Model Tree predictor as meta-regressor) to the best individual algorithm (SR-UKF) we could verify that the stacking achieved a significantly higher performance for all time series tested. This can be clearly noted in Figure 7.

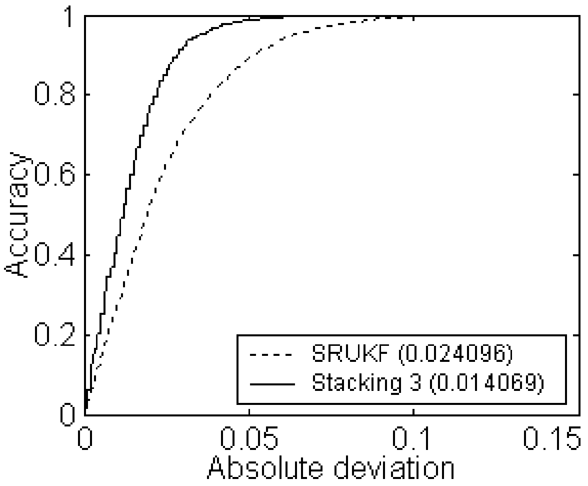


Figure 7. SR-UKF and Stacking 3 applied to Darwin time series.

5. Conclusions and Future Works

We have used REC curves in order to compare the SPKF family of filters (state-of-the-art time series predictors) and ensembles of them, applied to real-world time series.

The results of the experiments pointed SR-UKF as the best SPKF to use for forecasting with the series tested. Further experiments showed that a stacking composed by SR-CDKF and SR-UKF as base learners and a Model Tree predictor as meta-regressor can provide a performance statistically significantly better than that provided by the SR-UKF algorithm working individually. The REC curves showed to be very efficient in the comparison and choice of time series predictors and base learners for ensembles of them.

Currently, we are conducing tests with REC curves in order to compare Particle Filters (Doucet, de Freitas & Gordon, 2001), sequential Monte Carlo based methods that allows for a complete representation of the state distribution using sequential importance sampling and resampling. Since Particle Filters approximate the posterior without making any explicit assumption about its form, they can be used in general nonlinear, non-Gaussian systems. As a future work we intend to investigate further the use of ensembles with SPKFs, as well as with Particle Filters.

Acknowledgments

We would like to thank Eibe Frank for providing the source code for the NBR algorithm. The Sigma-Point Kalman Filters were obtained from the ReBEL Toolkit, under Academic License from OGI School of Science and Engineering, Rudolph van der Merwe and Eric A. Wan. The implementation of Model Trees was obtained from the WEKA System (http://www.cs.waikato.ac.nz/~ml/weka/). The authors are partially financially supported by the Brazilian Research Agency CNPq.

References

Bi, J., & Bennett, K. P. (2003). Regression Error Characteristic Curves. *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)* (pp. 43-50). Washington, DC.

Blake, C. L., & Merz, C. J. (2006). UCI Repository of Machine Learning Databases. Machine-readable data repository. University of California, Department of Information and Computer Science, Irvine, CA. [http://www.ics.uci.edu/~mlearn/MLRepository.html]

Caruana, R., & Niculescu-Mizil, A. (1997). An Empirical Evaluation of Supervised Learning for ROC Area. *Proceedings of the First Workshop on ROC Analysis (ROCAI 2004)* (pp. 1-8).

Dietterich, T. G. (1998). Machine Learning Research: Four Current Directions. *The AI Magazine*, *18*, 97-136.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag.

Dzeroski, S., & Zenko, B. (2004). Is Combining Classifiers with Stacking Better than Selecting the Best One?. *Machine Learning*, *54*, 255-273.

Frank, E., Trigg, L., Holmes, G., & Witten, I. H. (2000). Naive Bayes for Regression. *Machine Learning*, *41*, 5-25.

Ito, K., & Xiong, K. (2000). Gaussian Filters for Nonlinear Filtering Problems. *IEEE Transactions on Automatic Control*, *45*, 910-927.

van der Merwe, R., & Wan, E. (2001). Efficient Derivative-Free Kalman Filters for Online Learning. *Proceedings of the 9th European Symposium on Artificial Neural Networks (ESANN'2001)*. Bruges.

van der Merwe, R., & Wan, E. (2003). Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. *Proceedings of the Workshop on Advances in Machine Learning*. Montreal, Canada.

Jazwinsky, A. (1970). *Stochastic Processes and Filtering Theory*. New York: Academic Press.

Julier, S., Uhlmann, J., & Durrant-Whyte, H. (1995). A New Approach for Filtering Nonlinear Systems. *Proceedings of the American Control Conference* (pp. 1628-1632).

Keogh, E., & Folias, T. (2002). The UCR Time Series Data Mining Archive. University of California, Computer Science & Engineering Department, Riverside, CA.

Provost, F., & Fawcett, T. (1997). Analysis and Visualization of Classifier Performance: Comparison Under Imprecise Class and Cost Distributions. *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'97)* (pp. 43-48). AAAI Press.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Classifiers. *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)* (pp. 445-453). San Francisco: Morgan Kaufmann.

Quinlan, J.R. (1992). Learning with Continuous Classes. *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence* (pp. 343-348). Singapore: World Scientific.

Teixeira, M., & Zaverucha, G. (2003). Fuzzy Bayes and Fuzzy Markov Predictors. *Journal of Intelligent and Fuzzy Systems*, *13*, 155-165.

Wang, Y., & Witten, I. H. (1997). Induction of Model Trees for Predicting Continuous Classes. *Proceedings of the poster papers of the European Conference on Machine Learning.* University of Economics, Faculty of Informatics and Statistics, Prague.

Wolpert, D. (1992). Stacked generalization. *Neural Networks*, *5*, 241-260.

An Analysis of Reliable Classifiers through ROC Isometrics

Stijn Vanderlooy S.VANDERLOOY@CS.UNIMAAS.NL
Ida G. Sprinkhuizen-Kuyper KUYPER@CS.UNIMAAS.NL
Evgueni N. Smirnov SMIRNOV@CS.UNIMAAS.NL
MICC-IKAT, Universiteit Maastricht, PO Box 616, 6200 MD Maastricht, The Netherlands.

Abstract

Reliable classifiers abstain from uncertain instance classifications. In this paper we extend our previous approach to construct reliable classifiers which is based on isometrics in Receiver Operator Characteristic (ROC) space. We analyze the conditions to obtain a reliable classifier with higher performance than previously possible. Our results show that the approach is generally applicable to boost performance on each class simultaneously. Moreover, the approach is able to construct a classifier with at least a desired performance per class.

1. Introduction

Machine learning classifiers were applied to various classification problems. Nevertheless, only few classifiers were employed in domains with high misclassification costs, e.g., medical diagnosis and legal practice. In these domains it is desired to have classifiers that abstain from uncertain instance classifications such that a desired level of reliability is obtained. These classifiers are called reliable classifiers.

Recently, we proposed an easy-to-visualize approach to reliable instance classification (Vanderlooy et al., 2006). Classification performance is visualized as an ROC curve and a reliable classifier is constructed by skipping the part of the curve that represents instances difficult to classify. The transformation to the ROC curve of the reliable classifier was provided. An analysis showed when and where this new curve dominates the original one. If the underlying data of both curves have approximately equal class distributions, then dominance immediately results in performance increase. However, in case of different class distributions and a performance measure that is class-distribution dependent, dominance of an ROC curve does not always guarantee an increase in performance.

In this paper we analyze for which performance metrics the approach boosts performance on each class simultaneously. We restrict to widely used metrics characterized by rotating linear isometrics (Fürnkranz & Flach, 2005). Furthermore, skew sensitive metrics are used to generalize the approach to each possible scenario of error costs and class distributions.

This paper is organized as follows. Section 2 provides terminology and notation. Section 3 gives a brief background on ROC curves. Sections 4 and 5 introduce skew sensitive evaluation and isometrics, respectively. Section 7 defines reliable classifiers and their visualization in ROC space. In Section 8 we provide our main contribution. Section 9 concludes the paper.

2. Terminology and Notation

We consider classification problems with two classes: positive (p) and negative (n). A discrete classifier is a mapping from instances to classes. Counts of true positives, false positives, true negatives and false negatives are denoted by TP, FP, TN, and FN, respectively. The number of positive instances is $P = TP + FN$. Similarly, $N = TN + FP$ is the number of negative instances.

From these counts the following statistics are derived:

$$tpr = \frac{TP}{TP + FN} \qquad tnr = \frac{TN}{TN + FP}$$
$$fpr = \frac{FP}{FP + TN} \qquad fnr = \frac{FN}{TP + FN}$$

True positive rate is denoted by tpr and true negative rate by tnr. False positive rate and false negative rate are denoted by fpr and fnr, respectively. Note that $tnr = 1 - fpr$ and $fnr = 1 - tpr$.

Most classifiers are rankers or scoring classifiers. They

Appearing in *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, Pittsburgh, USA, 2006.

output two positive values $l(x|p)$ and $l(x|n)$ that indicate the likelihood that an instance x is positive and negative, respectively. The score of an instance combines these values as follows:

$$l(x) = \frac{l(x|p)}{l(x|n)} \tag{1}$$

and can be used to rank instances from most likely positive to most likely negative (Lachiche & Flach, 2003).

3. ROC Curves

The performance of a discrete classifier can be represented by a point (fpr, tpr) in ROC space. Optimal performance is obtained in $(0, 1)$. Points $(0, 0)$ and $(1, 1)$ represent classifiers that always predict the negative and positive class, respectively. The ascending diagonal connects these points and represents the strategy of random classification.

A threshold on the score $l(x)$ transforms a scoring classifier into a discrete one. Instances with a score higher than or equal to this threshold are classified as positive. The remaining instances are classified as negative. An ROC curve shows what happens with the corresponding confusion matrix for each possible threshold (Fawcett, 2003). The convex hull of the ROC curve (ROCCH) removes concavities.

Theorem 1 *For any point (fpr, tpr) on an ROCCH a classifier can be constructed that has the performance represented by that point.*

Provost and Fawcett (2001) prove this theorem. For simplicity of presentation, in the following we will assume that ROC curves are convex and all points can be obtained by a threshold.

4. Skew Sensitive Evaluation

The metrics *tpr*, *fpr*, *tnr*, and *fnr* evaluate performance on a single class. This follows from the confusion matrix since values are used from a single column. In most cases a metric is desired that indicates performance on both classes simultaneously. Unfortunately, such metric assumes that the class distribution of the application domain is known and used in the test set.

Accuracy is a well-known example. Provost et al. (1998) showed that classifier selection with this metric has two severe shortcomings with regard to class and error costs distributions.

To overcome these problems, Flach (2003) considers class and error costs distributions as a parameter of performance metrics. Evaluation with these metrics is called skew sensitive evaluation. The parameter is called the skew ratio and expresses the relative importance of negative versus positive class:

$$c = \frac{c(p,n)}{c(n,p)} \frac{P(n)}{P(p)} \tag{2}$$

Here, $c(p,n)$ and $c(n,p)$ denote the costs of a false positive and false negative, respectively[1]. The probabilities of a positive and negative instance are denoted by $P(p) = \frac{P}{P+N}$ and $P(n) = \frac{N}{P+N}$, respectively. The class ratio is then $\frac{P(n)}{P(p)} = \frac{N}{P}$.

From Eq. 2 it is clear that we can cover all possible scenarios of class and cost distributions by a single value of c used as parameter in the performance metric. If $c < 1$ ($c > 1$), then the positive (negative) class is most important.

In the following we assume without restriction that c is the ratio of negative to positive instances in the test set, i.e., $c = \frac{N}{P}$. The reader should keep in mind that our results are also valid for $c = \frac{c(p,n)}{c(n,p)} \frac{N}{P}$.

5. ROC Isometrics

Classifier performance is evaluated on both classes. We define a positive (negative) performance metric as a metric that measures performance on the positive (negative) classifications. The skew sensitive metrics used in this paper are summarized in Table 1. An explanation of these metrics follows.

ROC isometrics are collections of points in ROC space with the same value for a performance metric. Flach (2003) and Fürnkranz and Flach (2005) investigate isometrics to understand metrics. However, isometrics can be used for the task of classifier selection and to construct reliable classifiers (see Section 6).

Table 1 also shows the isometrics for the performance metrics. They are obtained by fixing the performance metric and rewriting its equation to that of a line in ROC space. Varying the value of the metric results in linear lines that rotate around a single point in which the metric is undefined.

5.1. Precision

Positive precision, $prec_p^c$, is defined as the proportion of true positives to the total number of positive classifications. The isometrics are linear lines that rotate around the origin $(0, 0)$.

[1] Benefits of true positives and true negatives are incorporated by adding them to the corresponding errors. This operation normalizes the cost matrix such that the two values on the main diagonal are zero.

Table 1. Performance metrics and corresponding isometrics defined in terms of *fpr*, *tpr*, $c = \frac{N}{P}$, $\alpha \in \mathbb{R}^+$, and $\hat{m} = \frac{m}{P+N}$.

Metric	Indicator	Formula	Isometric
Pos. precision	$prec_p^c$	$\frac{tpr}{tpr+c\,fpr}$	$tpr = \frac{prec_p^c}{1-prec_p^c}\ c\,fpr$
Neg. precision	$prec_n^c$	$\frac{tnr}{tnr+\frac{1}{c}fnr}$	$tpr = \frac{1-prec_n^c}{prec_n^c}\ c\,fpr + 1 - \frac{1-prec_n^c}{prec_n^c}\ c$
Pos. *F*-measure	$F_p^{c,\alpha}$	$\frac{\left(1+\alpha^2\right)tpr}{\alpha^2+tpr+c\,fpr}$	$tpr = \frac{F_p^{c,\alpha}}{1+\alpha^2-F_p^{c,\alpha}}c\,fpr + \frac{\alpha^2\ F_p^{c,\alpha}}{1+\alpha^2-F_p^{c,\alpha}}$
Neg. *F*-measure	$F_n^{c,\alpha}$	$\frac{\left(1+\alpha^2\right)tnr}{\alpha^2+tnr+\frac{1}{c}fnr}$	$tpr = \frac{1+\alpha^2-F_n^{c,\alpha}}{F_n^{c,\alpha}}c\,fpr + 1 + \frac{(1+\alpha^2)(F_n^{c,\alpha}-1)}{F_n^{c,\alpha}}c$
Pos. *gm*-estimate	$gm_p^{c,\hat{m}}$	$\frac{tpr+\hat{m}}{tpr+c\,fpr+\hat{m}(1+c)}$	$tpr = \frac{gm_p^{c,\hat{m}}}{1-gm_p^{c,\hat{m}}}c\,fpr + \frac{\hat{m}\left(gm_p^{c,\hat{m}}(1+c)-1\right)}{1-gm_p^{c,\hat{m}}}$
Neg. *gm*-estimate	$gm_n^{c,\hat{m}}$	$\frac{tnr+\hat{m}}{tnr+\frac{1}{c}fnr+\hat{m}\frac{1+c}{c}}$	$tpr = \frac{1-gm_n^{c,\hat{m}}}{gm_n^{c,\hat{m}}}c\,fpr + 1 - \frac{1-gm_n^{c,\hat{m}}}{gm_n^{c,\hat{m}}}c + \frac{\hat{m}\left(gm_n^{c,\hat{m}}(1+c)-c\right)}{gm_n^{c,\hat{m}}}$

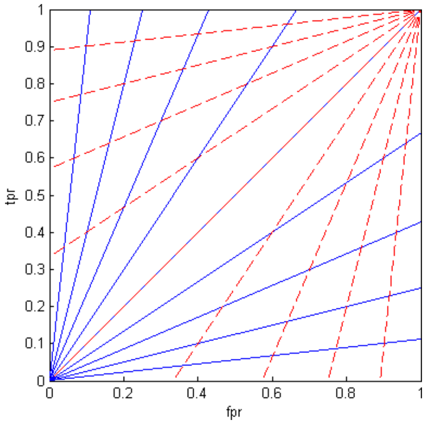


Figure 1. Precision isometrics in ROC space: solid lines are $prec_p^1$-isometrics and dashed lines are $prec_n^1$-isometrics.

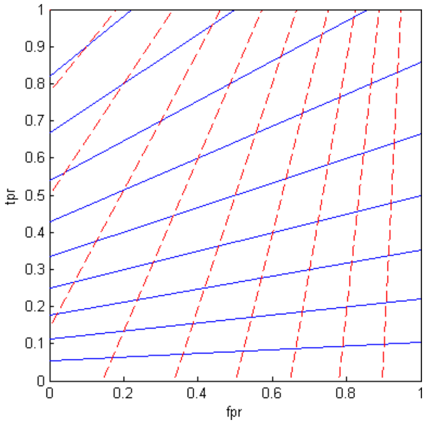


Figure 2. F-measure isometrics in ROC space: solid lines are $F_p^{1,1}$-isometrics and dashed lines are $F_n^{1,1}$-isometrics.

The case of negative precision, $prec_n^c$, is similar. Corresponding isometrics rotate around point $(1,1)$. Figure 1 shows $prec_p^c$-isometrics and $prec_n^c$-isometrics for $c = 1$. In this and subsequent figures the value of the performance metric is varied from 0.1 to 0.9.

5.2. F-measure

Positive precision is maximized when all positive classifications are correct. To know if $prec_p^c$ uses enough positive instances to be considered as reliable, it is combined with *tpr*. Note that $prec_p^c$ and *tpr* are antagonistic, i.e., if $prec_p^c$ goes up, then *tpr* usually goes down (and vice versa).

Rijsbergen (1979) introduced the positive F-measure for the trade-off between these metrics:

$$F_p^{c,\alpha} = \frac{(1+\alpha^2)\ prec_p^c\ tpr}{\alpha^2\ prec_p^c + tpr} = \frac{\left(1+\alpha^2\right)tpr}{\alpha^2 + tpr + c\,fpr} \qquad (3)$$

where the parameter α indicates the importance given to $prec_p^c$ relative to *tpr*. If $\alpha < 1$ ($\alpha > 1$) then *tpr* is less (more) important than $prec_p^c$. If $\alpha = 1$, then both terms are equally important.

The isometrics of $F_p^{c,\alpha}$ are linear lines rotating around $(-\frac{\alpha^2}{c}, 0)$. Therefore, they can be seen as a shifted version of the positive precision isometrics. The larger c and/or the smaller α, the smaller the difference with $prec_p^c$-isometrics.

Similar to $F_p^{c,\alpha}$ the negative F-measure is a metric for the trade-off between $prec_n^c$ and *tnr*. Isometrics are a shifted version of the $prec_n^c$-isometrics and rotate around $(1, 1+\alpha^2 c)$. Figure 2 shows $F_p^{c,\alpha}$-isometrics and $F_n^{c,\alpha}$-isometrics for $c = 1$ and $\alpha = 1$ in the relevant region $(0,1) \times (0,1)$ of ROC space.

5.3. Generalized m-estimate

The m-estimate computes a precision estimate assuming that m instances are a priori classified. One of the main reasons why it is favored over precision is that it is less sensitive to noise and more effective in avoiding overfitting (Fürnkranz & Flach, 2005; Lavrac & Dzeroski, 1994, Chapters 8-10). This is especially true if the metric is used for the minority class when the class distribution is very skewed.

The positive m-estimate assumes that m instances are a priori classified as positive. These instances are distributed according to the class distribution in the training set:

$$gm_p^{c,m} = \frac{TP + m\frac{P}{P+N}}{TP + FP + m} \qquad (4)$$

or equivalently:

$$gm_p^{c,m} = \frac{tpr + \frac{m}{P+N}}{tpr + c\,fpr + \frac{m}{P}} \qquad (5)$$

To eliminate absolute numbers P and N we define $\hat{m} = \frac{m}{P+N}$ and obtain the formula in Table 1. Fürnkranz and Flach (2005) call this metric the positive gm-estimate (generalized m-estimate) since $\hat{m}$ defines the rotation point of the isometrics (see below)[2].

The isometrics of the $gm_p^{c,\hat{m}}$-estimate rotate around $(-\hat{m}, -\hat{m})$. If $\hat{m} = 0$, then we obtain $prec_p^c$-isometrics. For $\hat{m} \rightarrow \infty$ the performance metric converges to $\frac{1}{1+c} = P(p)$ and the corresponding isometric is the ascending diagonal.

The case of the negative gm-estimate is similar. The rotation point of the isometrics is $(1+\hat{m}, 1+\hat{m})$. Figure 3 shows $gm_p^{c,\hat{m}}$-isometrics and $gm_n^{c,\hat{m}}$-isometrics for $c = 1$ and $\hat{m} = 0.1$.

For simplicity of presentation, in the following the isometric of a positive (negative) performance metric is simply called a positive (negative) isometric.

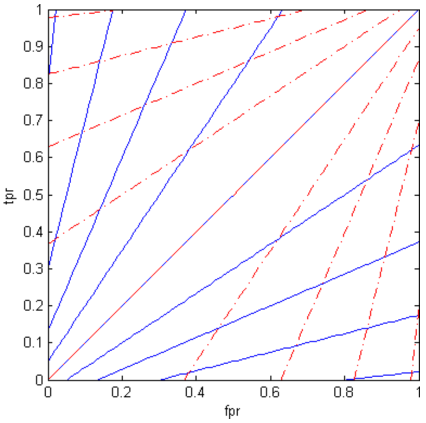
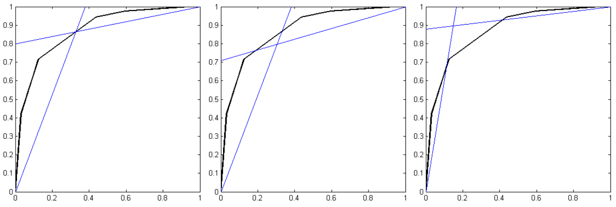


Figure 3. Generalized m-estimate isometrics in ROC space: solid lines are $gm_p^{1,0.1}$-isometrics and dashed lines are $gm_n^{1,0.1}$-isometrics.



(a) (b) (c)

Figure 4. Location of intersection between a positive and negative isometric: (a) Case 1, (b) Case 2, and (c) Case 3.

6. Classifier Design through Isometrics

In Vanderlooy et al. (2006) we used precision isometrics as a tool to design classifiers. We generalize this approach to include all isometrics defined in Section 5.

For specific skew ratio, a positive isometric is build with a desired positive performance. By definition, the intersection point (fpr_a, tpr_a) with an ROCCH represents a classifier with this performance. Similarly, the intersection point (fpr_b, tpr_b) of a negative isometric and the ROCCH represents a classifier with negative performance defined by that isometric. If we assume that the positive and negative isometrics intersect each other in the relevant region of ROC space, then three cases can be distinguished to construct the desired classifier (see Figure 4).

Case 1: the isometrics intersect on the ROCCH
The discrete classifier corresponding to this point has the performance defined by both isometrics. Theorem 1 guarantees that we can construct it. Therefore, the isometrics provide an approach to construct a classifier with a desired performance per class.

Case 2: the isometrics intersect below the ROCCH
This classifier can also be constructed. However, the classifiers corresponding to any point on the ROCCH between (fpr_b, tpr_b) and (fpr_a, tpr_a) have better performance.

Case 3: the isometrics intersect above the ROCCH
There is no classifier with the desired performances. To increase performance instances between (fpr_a, tpr_a) and (fpr_b, tpr_b) are not classified. In case

[2] The gm-estimate of Fürnkranz and Flach (2005) is more general than ours since they also vary $a = \frac{1}{P+N}$ in Eq. 5.

of more than one intersection point for the positive (negative) isometric and the ROCCH, the intersection point with highest *tpr* (lowest *fpr*) is chosen such that $fpr_a < fpr_b$. Then, the number of unclassified instances is minimized. The resulting classifier is called a reliable classifier.

7. Reliable Instance Classification

A scoring classifier is almost never optimal: there exists negative instances with higher score than some positive instances. A reliable classifier abstains from these uncertain instance classifications. It simulates the behavior of a human expert in fields with high error costs. For example, in medical diagnosis an expert does not state a possibly incorrect diagnosis but she says “I do not know” and performs more tests.

Similar to Ferri and Hernández-Orallo (2004), we define a reliable classifier as a filtering mechanism with two thresholds $a > b$. An instance x is classified as positive if $l(x) \geq a$. If $l(x) \leq b$, then x is classified as negative. Otherwise, the instance is left unclassified. Unclassified instances can be rejected, passed to a human, or to another classifier (Ferri et al., 2004). Pietraszek (2005) chooses a and b to minimize expected cost, also considering the abstention costs. Here, we focus on performance on the classified instances.

Counts of unclassified positives and unclassified negatives are denoted by UP and UN, respectively. Unclassified positive rate and unclassified negative rate are then defined as follows:

$$upr = \frac{UP}{TP+FN+UP} \tag{6}$$

$$unr = \frac{UN}{FP+TN+UN} \tag{7}$$

We define thresholds a and b to correspond with points (fpr_a, tpr_a) and (fpr_b, tpr_b), respectively. The ROC curve of the reliable classifier is obtained by skipping the part between (fpr_a, tpr_a) and (fpr_b, tpr_b). By definition we have:

$$upr = tpr_b - tpr_a \tag{8}$$

$$unr = fpr_b - fpr_a \tag{9}$$

The transformation from the original ROC curve to that of the reliable classifier is given in Theorem 2.

Theorem 2 *If the part between points (fpr_a, tpr_a) and (fpr_b, tpr_b) of an ROC curve is skipped with $0 < upr < 1$ and $0 < unr < 1$, then points (fpr_x, tpr_x) on this curve between $(0, 0)$ and (fpr_a, tpr_a) are transformed into points (fpr'_x, tpr'_x) such that:*

$$fpr'_x = \frac{fpr_x}{1 - unr} \ , \ tpr'_x = \frac{tpr_x}{1 - upr} \tag{10}$$

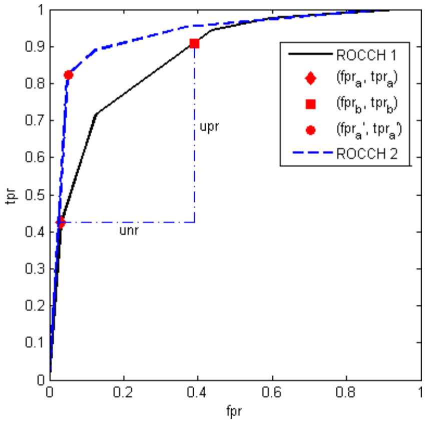


Figure 5. ROCCH 2 is obtained by not covering the part between (fpr_a, tpr_a) and (fpr_b, tpr_b) of ROCCH 1. The length of the horizontal (vertical) line below ROCCH 1 equals *unr* (*upr*).

Points (fpr_x, tpr_x) between (fpr_b, tpr_b) and $(1, 1)$ are transformed into points (fpr'_x, tpr'_x) such that:

$$fpr'_x = 1 - \frac{1 - fpr_x}{1 - unr} \ , \ tpr'_x = 1 - \frac{1 - tpr_x}{1 - upr} \tag{11}$$

The proof is in Vanderlooy et al. (2006). Note that the transformations of (fpr_a, tpr_a) and (fpr_b, tpr_b) are the same point on the new ROC curve. Figure 5 shows an example of a transformation. The intersection points are obtained with precision isometrics for $c = 1$, $prec_p^c = 0.93$, and $prec_n^c = 0.87$.

Theorem 3 *If the original ROC curve is convex, then the ROC curve obtained by not considering the points between (fpr_a, tpr_a) and (fpr_b, tpr_b) is also convex.*

We proved this theorem in Vanderlooy et al. (2006). There, we also analyzed when and where the original ROCCH is dominated by that of the reliable classifier. Note that the underlying data of both ROCCHs can have a different class distribution when $upr \neq unr$. For skew insensitive metrics or when $upr \approx unr$, dominance of a ROCCH will immediately result in performance increase. In the next Section 8 we analyze when the skew sensitive performance metrics in Table 1 can be boosted by abstention.

8. Effect on Performance

We defined (fpr_a, tpr_a) and (fpr_b, tpr_b) as intersection points of an ROCCH and positive and negative isometric, respectively. The type of isometrics defines the effect on the performance of the reliable classifier corresponding to (fpr'_a, tpr'_a) as defined in Theorem 2.

8.1. Precision

Theorem 4 provides an easy and computationally efficient approach to construct a classifier with a desired precision per class.

Theorem 4 *If points (fpr_a, tpr_a) and (fpr_b, tpr_b) are defined by an $prec_p^c$-isometric and $prec_n^c$-isometric respectively, then the point (fpr'_a, tpr'_a) has the precisions of both isometrics.*

The proof of this theorem and also of following theorems are included in the appendix. Since isometrics of skew sensitive performance metrics are used, the approach does not commit to costs and class distributions[3]. Thus, when the application domain changes a new reliable classifier can be constructed from the original ROC curve only.

Theorem 4 together with the next Theorem 5 provides an approach to construct a classifier with desired accuracy. This approach overcomes the problems with accuracy explained in Section 4. From the proof it follows that if the precisions are not equal, then the accuracy is bounded by the smallest and largest precision.

Theorem 5 *If point (fpr'_a, tpr'_a) has $prec_p^c = prec_n^c$, then the accuracy in this point equals the precisions.*

8.2. F-measure

Theorem 6 shows that also the F-measure can be boosted on both classes if a part of an ROC curve is not covered. In this case, the resulting classifier has higher performance than defined by both isometrics. Figure 6 gives an example where positive (negative) performance is increased with approximately 5% (10%).

Theorem 6 *If points (fpr_a, tpr_a) and (fpr_b, tpr_b) are defined by an $F_p^{c,\alpha}$-isometric and $F_n^{c,\alpha}$-isometric respectively, then the point (fpr'_a, tpr'_a) has higher performance than defined by both isometrics.*

8.3. Generalized m-estimate

To analyze the effect of abstention on the gm-estimate, we can consider the number of a priori classified instances m to be fixed or the parameter $\hat{m}$ to be fixed.

Consider the case when m is not changed after transformation. In this case upr and unr can change the distribution of a priori instances over the classes. If $upr < unr$, then the distribution of these instances in the positive gm-estimate moves to the true positives resulting in higher performance. For the negative gm-estimate, the distribution moves to the false negatives resulting in lower performance. The case of $upr > unr$ is the other way around. Therefore, an increase in performance in both classes is only possible iff $upr = unr$.

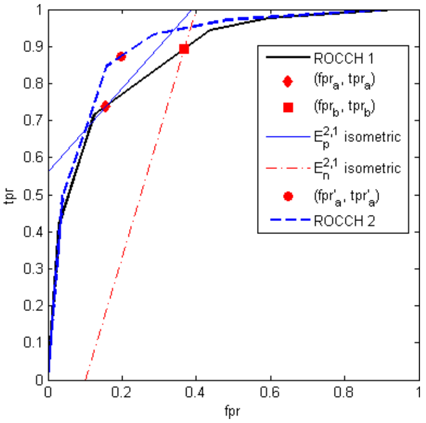


Figure 6. Designing with F-measure isometrics: $F_p^{2,1} = 0.72$ in (fpr_a, tpr_a) and $F_n^{2,1} = 0.75$ in (fpr_b, tpr_b). The reliable classifier represented by (fpr'_a, tpr'_a) has $F_p^{1.84,1} = 0.7693$ and $F_n^{1.84,1} = 0.8597$. The abstention is represented by $upr = 0.1541$ and $unr = 0.2116$.

For the case when $\hat{m}$ is not changed after transformation, a similar reasoning results in improvement of the positive gm-estimate if $upr \leq unr$ and $tpr_a \geq fpr_a$. The latter condition holds for all points on the ROCCH. Similarly, improvement in the negative gm-estimate occurs if $upr \geq unr$ and $tpr_b \geq fpr_b$. Thus, we find the following theorems for the gm-estimate.

Theorem 7 *If point (fpr_a, tpr_a) is defined by an $gm_p^{c,\hat{m}}$-estimate isometric with $m > 0$ and if $upr \leq unr$, then the point (fpr'_a, tpr'_a) has at least the positive performance defined by that isometric.*

Theorem 8 *If point (fpr_b, tpr_b) is defined by an $gm_n^{c,\hat{m}}$-estimate isometric with $m > 0$ and if $upr \geq unr$, then the point (fpr'_a, tpr'_a) has at least the negative performance defined by that isometric.*

Corollary 1 *If points (fpr_a, tpr_a) and (fpr_b, tpr_b) are defined by an $gm_p^{c,\hat{m}}$-estimate isometric and $gm_n^{c,\hat{m}}$-estimate isometric respectively with $m > 0$ and if $upr = unr$, then the point (fpr'_a, tpr'_a) has at least the gm-estimates of both isometrics.*

We suggest to use the gm-estimate for the minority class only and to use a normal precision for the majority class. From Theorems 7 and 8, if the minority

[3] Remember that, although our proofs use the simplest case $c = \frac{N}{P}$, the results are also valid for $c = \frac{c(p,n)}{c(n,p)}\frac{N}{P}$.

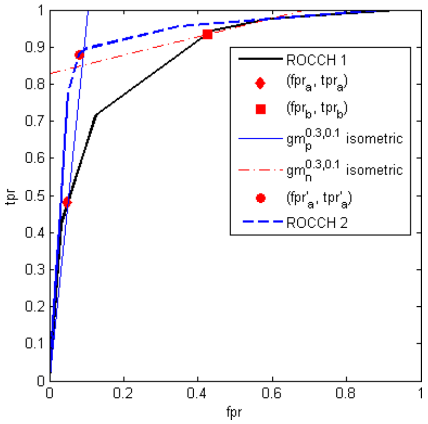


Figure 7. Designing with precision and *gm*-estimate isometrics: $prec_p^{0.3} = 0.97$ in (fpr_a, tpr_a) and $gm_n^{0.3,0.1} = 0.55$ in (fpr_b, tpr_b). The reliable classifier represented by (fpr'_a, tpr'_a) has $prec_p^{0.3} = 0.97$ and $gm_n^{0.34,0.18} = 0.5584$. The abstention is represented by $upr = 0.4549$ and $unr = 0.3763$.

class is the positive (negative) class, then we need an abstention characterized by $upr \leq unr$ ($upr \geq unr$).

Figure 7 shows an example with fixed m and the negative class as minority class. Therefore, we want that the $gm_n^{c,\hat{m}}$-estimate isometric covers a large part in ROC space and consequently the condition $upr \geq unr$ is easily satisfied.

9. Conclusions

A reliable classifier abstains from uncertain instance classifications. Benefits are significant in application domains with high error costs, e.g., medical diagnosis and legal practice. A classifier is transformed into a reliable one by not covering a part of its ROC curve. This part is defined by two isometrics indicating performance on a different class.

In case of a classifier and corresponding reliable classifier, dominance of an ROC curve immediately represents an increase in performance if the underlying data of both curves have approximately equal class distributions. Since this assumption is too strong, we analyzed when performance can be boosted by abstention.

We showed how to construct a (reliable) classifier with a desired precision per class. We did the same for accuracy. For the F-measure a classifier is obtained with at least the desired performance per class. To prevent a possible performance decrease with the *gm*-estimate, we propose to use it for the minority class and to use a normal precision for the majority class.

We may conclude that the proposed approach is able to boost performance on each class simultaneously. Benefits of the approach are numerous: it guarantees a classifier with an acceptable performance in domains with high error costs, it is efficient in terms of time and space, classifier independent, and it incorporates changing error costs and class distributions easily.

Acknowledgments

We thank the reviewers for useful comments and suggestions. This work was partially supported by the Dutch Organization for Scientific Research (NWO), grant nr: 634.000.435.

References

Fawcett, T. (2003). *ROC graphs: Notes and practical considerations for researchers* (Technical Report HPL-2003-4). HP Laboratories.

Ferri, C., Flach, P., & Hernández-Orallo, J. (2004). Delegating classifiers. *Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence (ROCAI-2004)* (pp. 37–44).

Ferri, C., & Hernández-Orallo, J. (2004). Cautious classifiers. *Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence (ROCAI-2004)* (pp. 27–36).

Flach, P. (2003). The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)* (pp. 194–201).

Fürnkranz, J., & Flach, P. (2005). Roc ‘n’ rule learning – towards a better understanding of covering algorithms. *Machine Learning*, *58*, 39–77.

Lachiche, N., & Flach, P. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)* (pp. 416–423).

Lavrac, N., & Dzeroski, S. (1994). *Inductive logic programming: Techniques and applications*. Ellis Horwood, New York.

Pietraszek, T. (2005). Optimizing abstaining classifiers using ROC analysis. *Proceedings of the 22th International Conference on Machine Learning (ICML-2005)* (pp. 665–672).

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, *42*, 203–231.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *Proceedings of the 15th International Conference on Machine Learning (ICML-1998)* (pp. 43–48).

Rijsbergen, C. V. (1979). *Information retrieval.* Department of Computer Science, University of Glasgow. 2nd edition.

Vanderlooy, S., Sprinkhuizen-Kuyper, I., & Smirnov, E. (2006). Reliable classifiers in ROC space. *Proceedings of the 15th Annual Machine Learning Conference of Belgium and the Netherlands (BENELEARN-2006)* (pp. 113–120).

A. Proofs

Proof of Theorem 4

The positive precisions in (fpr_a, tpr_a) and (fpr'_a, tpr'_a) are defined as follows:

$$prec_p^c\,(fpr_a, tpr_a) = \frac{tpr_a}{tpr_a + c\, fpr_a} \tag{12}$$

$$prec_p^{c'}\left(fpr'_a, tpr'_a\right) = \frac{tpr'_a}{tpr'_a + c'\, fpr'_a} \tag{13}$$

with $c' = c\frac{1-unr}{1-upr}$. Substitution of Eq. 10 in Eq. 13 results in Eq 12. In a similar way, Eq. 11 is used to show that the negative precisions in (fpr_b, tpr_b) and (fpr'_b, tpr'_b) are the same. The theorem follows since $(fpr'_b, tpr'_b) = (fpr'_a, tpr'_a)$. □

Proof of Theorem 5

Since the positive precision and negative precision in (fpr'_a, tpr'_a) are equal, we can write:

$$tpr'_a = a\left(tpr'_a + c'\, fpr'_a\right) \tag{14}$$

$$tnr'_a = a\left(tnr'_a + \frac{1}{c'} fnr'_a\right) \tag{15}$$

with $a = prec_p^{c'} = prec_n^{c'}$. It follows that:

$$tpr'_a + c'\, tnr'_a = a\left(tpr'_a + c'\, fpr'_a + c'\, tnr'_a + fnr'_a\right) \tag{16}$$

or equivalently:

$$a = \frac{tpr'_a + c'\, tnr'_a}{tpr'_a + c'\, fpr'_a + c'\, tnr'_a + fnr'_a} \tag{17}$$

and this is the accuracy with skew ratio c'. □

Proof of Theorem 6

The positive F-measures in (fpr_a, tpr_a) and (fpr'_a, tpr'_a) are defined as follows:

$$F_p^{c,\alpha}\,(fpr_a, tpr_a) = \frac{\left(1+\alpha^2\right) tpr_a}{\alpha^2 + tpr_a + c\, fpr_a} \tag{18}$$

$$F_p^{c',\alpha}\left(fpr'_a, tpr'_a\right) = \frac{\left(1+\alpha^2\right) tpr'_a}{\alpha^2 + tpr'_a + c'\, fpr'_a} \tag{19}$$

Using Eq. 10 and $c' = c\frac{1-unr}{1-upr}$, the right-hand side of Eq. 19 becomes:

$$\frac{\left(1+\alpha^2\right) tpr_a}{\alpha^2(1-upr) + tpr_a + c\, fpr_a} \tag{20}$$

It follows that $F_p^{c',\alpha}\left(fpr'_a, tpr'_a\right) > F_p^{c,\alpha}\,(fpr_a, tpr_a)$ since $0 < upr < 1$. The case of the negative F-measure is similar. □

Proof of Theorem 7

The positive gm-estimates in (fpr_a, tpr_a) and (fpr'_a, tpr'_a) are defined as follows:

$$gm_p^{c,\hat{m}}\,(fpr_a, tpr_a) = \frac{tpr+\hat{m}}{tpr+c\, fpr+\hat{m}(1+c)} \tag{21}$$

$$gm_p^{c',\hat{m}'}\left(fpr'_a, tpr'_a\right) = \frac{tpr'+\hat{m}'}{tpr'+c'\, fpr'+\hat{m}'(1+c')} \tag{22}$$

with $\hat{m} = \frac{m}{P+N}$, and $c' = c\frac{1-unr}{1-upr}$.

Case 1: m is not changed after transformation

In this case we can write $\hat{m}' = \frac{m}{P(1-upr)+N(1-unr)}$. Substitution of Eq. 10 in Eq. 22 results in the following right-hand side:

$$\frac{tpr + m\frac{1-upr}{P(1-upr)+N(1-unr)}}{tpr + c\, fpr + \hat{m}(1+c)} \tag{23}$$

Clearly, $gm_p^{c',\hat{m}'}\left(fpr'_a, tpr'_a\right) \geq gm_p^{c,\hat{m}}\,(fpr_a, tpr_a)$ iff:

$$\frac{1-upr}{P(1-upr)+N(1-unr)} \geq \frac{1}{P+N} \tag{24}$$

This holds iff $upr \leq unr$.

Case 2: $\hat{m}$ is not changed after transformation

Substitution of Eq. 10 in Eq. 22 with fixed $\hat{m}$ results in the following right-hand side:

$$\frac{tpr + \hat{m}(1-upr)}{tpr + c\, fpr + \hat{m}(1-upr+c(1-unr))} \tag{25}$$

Straightforward computation results in $gm_p^{c',\hat{m}}\left(fpr'_a, tpr'_a\right) \geq gm_p^{c,\hat{m}}\,(fpr_a, tpr_a)$ iff:

$$\hat{m}(unr-upr) + (tpr_a\; unr - fpr_a\; upr) \geq 0 \tag{26}$$

This holds if $upr \leq unr$ and $tpr_a \geq fpr_a$.□

Proof of Theorem 8

The proof is similar to that of Theorem 7. □

A Comparison of Different ROC Measures for Ordinal Regression

Willem Waegeman WILLEM.WAEGEMAN@UGENT.BE
Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 913, B-9052 Ghent, Belgium

Bernard De Baets BERNARD.DEBAETS@UGENT.BE
Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, B-9000 Ghent, Belgium

Luc Boullart LUC.BOULLART@UGENT.BE
Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 913, B-9052 Ghent, Belgium

Abstract

Ordinal regression learning has characteristics of both multi-class classification and metric regression because labels take ordered, discrete values. In applications of ordinal regression, the misclassification cost among the classes often differs and with different misclassification costs the common performance measures are not appropriate. Therefore we extend ROC analysis principles to ordinal regression. We derive an exact expression for the volume under the ROC surface (VUS) spanned by the true positive rates for each class and show its interpretation as the probability that a randomly drawn sequence with one object of each class is correctly ranked. Because the computation of VUS has a huge time complexity, we also propose three approximations to this measure. Furthermore, the properties of VUS and its relationship with the approximations are analyzed by simulation. The results demonstrate that optimizing various measures will lead to different models.

1. Introduction

In multi-class classification labels are picked from a set of unordered categories. In metric regression labels might take continuous values. Ordinal regression can be located in between these learning problems because here labels are chosen from a set of ordered categories. Applications of ordinal regression frequently arise in domains where humans are part of the data generation process. When humans assess objects for their beauty, quality, suitability or any other characteristic, they really prefer to qualify them with ordinal labels instead of continuous scores. This kind of datasets is obtained in information retrieval and quality control, where the user or the human expert frequently evaluates objects with linguistic terms, varying from “very bad” to “very good” for example. Also in medicine and social sciences, where many datasets originate by interaction with humans, ordinal regression models can be used.

In these applications of ordinal regression one is often the most interested in a subset of the classes. In many cases these classes of interest are the “extreme” categories, such as the documents with the highest relevance to the query or the products with the lowest quality. Moreover, there is often an unequal number of training objects for the different categories in real-world ordinal regression problems. The overall classification rate or mean absolute error are in these cases not the most pertinent performance measures. Criteria such as the area under the *receiver operating characteristics* (ROC) curve — which is related to defining an optimal ranking of the objects — are more appropriate. This article aims to discuss possible extensions of ROC analysis for ordinal regression.

Nowadays the area under the ROC curve is used as a standard performance measure in many fields where a binary classification system is needed. A ROC curve is created by plotting the *true positive rate* (TPR) versus

Appearing in *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, Pittsburgh, USA, 2006. Copyright 2006 by the author(s)/owner(s).

	$\widehat{y}=-1$	$\widehat{y}=1$	
$y=-1$	TN	FP	n_-
$y=1$	FN	TP	n_+
	NP	PP	n

Table 1: Confusion matrix for a two class classification problem of size n

the *false positive rate* (FPR). The TPR (or *sensitivity*) and the FPR (also known as 1 - *specificity*) are computed from the confusion matrix or contingency table (shown in Table 1). Sensitivity is defined as the number of positive predicted examples from the positive class TP divided by the number of positive examples n_+ and specificity is defined as the number of negative predicted examples TN from the negative class divided by the number of negative examples n_-:

$$Sens = TPR = \frac{TP}{TP + FN} \tag{1}$$

$$Spec = TNR = 1 - FPR = \frac{TN}{TN + FP} \tag{2}$$

With a classifier that estimates a continuous function f, the class prediction $\widehat{y}$ for an object x is obtained by the following rule:

$$\widehat{y} = \operatorname{sgn}(f(x) + b) \tag{3}$$

The points defining the ROC curve can then be computed by varying the threshold b from the most negative to the most positive function value and the *area under the ROC curve* (AUC) gives an impression of quality of the classifier. It has been shown [Cortes & Mohri, 2003, Yan et al., 2003] that the AUC is equivalent to the *Wilcoxon-Mann-Whitney* statistic:

$$WMW = AUC(f) = \frac{1}{n_- n_+} \sum_{i=1}^{n_-} \sum_{j=1}^{n_+} I_{f(x_i)<f(x_j)} \tag{4}$$

The value of the indicator function I will be one when its argument is true and zero otherwise. The measure $AUC(f)$ can be seen as a nonparametric estimate for the probability that the function value of an object randomly drawn from the negative class is strictly smaller than the function value of an object randomly drawn from the positive class:

$$AUC(f) = P(f(x_i) < f(x_j) \mid y_i = -1 \wedge y_j = 1)) \tag{5}$$

2. ROC Measures for Ordinal Regression

Recently, different approaches have been proposed to extend ROC analysis for multi-class classification. In the most general case, the *volume under the ROC surface* (VUS) has to be minimized in multi-class classification. The ROC surface can be seen as a Pareto front, where each objective corresponds to one dimension. In case there are more then two classes (let's say r), then the number of objectives depends on the multi-class method that is used:

- For a *one-versus-all* method, r functions f_k are estimated that try to separate objects of class k from the other classes. As a consequence misclassification costs for each class are fixed and the corresponding ROC surface will have r dimensions representing the true positive rates TPR_k for each class [Flach, 2004]. ROC points are here obtained by varying the thresholds b_k in the prediction rule $\widehat{y} = \operatorname{argmax}_k f_k(x) + b_k$.
- For a *one-versus-one* method, a function f_{kl} is estimated for each pair of classes, which allows to specify the cost for a misclassification of an object of class k predicted as class l. The corresponding ROC space is in this case spanned by $\frac{r(r-1)}{2}$ objectives [Ferri et al., 2003]. A prediction for new instances is done by majority voting over all $\frac{r(r-1)}{2}$ classifiers based on the outcomes $\operatorname{sgn}(f_{kl} + b_{kl})$.

In ordinal regression the picture is slightly different. The vast majority of existing methods for ordinal regression — including traditional statistical methods like cumulative logit models and their variants [Agresti, 2002], kernel methods [Chu & Keerthi, 2005, Shashua & Levin, 2003] and bayesian approaches [Chu & Gharhamani, 2005] — fit in general one function f to the data together with $r-1$ thresholds b_k for r ordered classes. New observations can then be classified by predicting them into the class k for which it holds that

$$b_{k-1} < f(x) \leq b_k \text{ with } b_0 = -\infty \text{ and } b_r = +\infty. \tag{6}$$

The simplicity of this kind of models has as disadvantage that one can not control the cost of misclassifying an object of a given class into another specified class. In other words, like in *one-versus-all* multi-class classification only r objectives can be simultaneously minimized. Therefore one could wonder whether a *one-versus-one* approach could be useful for ordinal regression. However, the answer is negative because it would lead to a more complex model with more variables to be estimated. Fortunately, Misclassification costs are always proportional to the absolute difference between the real and the predicted class, so defining a loss function with this property will solve the problem [Rennie & Srebro, 2005].

We will further assume that the misclassification costs are fixed for each class (they are always to proportional to the absolute difference between the real and the predicted label). Like in binary classification, we want a model f that imposes an optimal ranking of the data objects. There are several ways to define an optimal ranking. By analogy with (5) an optimal ranking could here be defined as a ranking that maximizes the joint probability that an r-tuple $(x_1, ..., x_r)$ is correctly ordered where each element x_k is randomly drawn from class k. This probability is given by

$$P(\bigwedge_{k=1}^{r-1} (f(x_k) < f(x_{k+1}) \mid y_k = k) \tag{7}$$

and it can be estimated for a given model by counting the number of ordered r-tuples occurring in the training dataset, i.e.

$$OrdTuples(f) = \frac{1}{\prod_{k=1}^{r} n_k} \sum_{y_{j_1} < \ldots < y_{j_r}} I_{f(x_{j_1}) < \ldots < f(x_{j_r})} \tag{8}$$

Here n_k stands for the number of objects with label k. It is straightforward to see that $OrdTuples(f)$ reduces to (4) in case of two classes. Furthermore, we can show the following.

Theorem 2.1 *Given a continuous function f that imposes a ranking over a dataset with r ordered classes, OrdTuples(f) is the volume under the ROC surface ($VUS_{ord}(f)$) spanned by the true positive rates for each class.*

In statistics there has some related work on this topic. [Dreisetl et al., 2000] derive formulas for the variance of VUS_{ord} and the covariance between two volumes in the three class case. This work is extended to the general r-class case in [Nakas & Yiannoutsos, 2004]. They conclude that bootstrapping is preferred over U-statistics for large values of n and r. In this article we focus more on the use of $VUS_{ord}(f)$ as performance measure for ordinal regression problems.

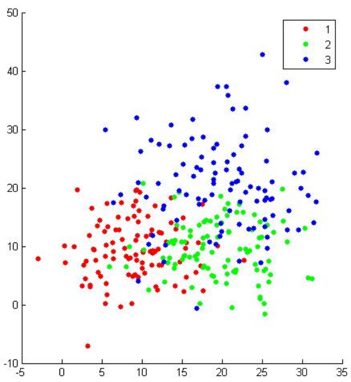


Figure 1: Synthetic dataset

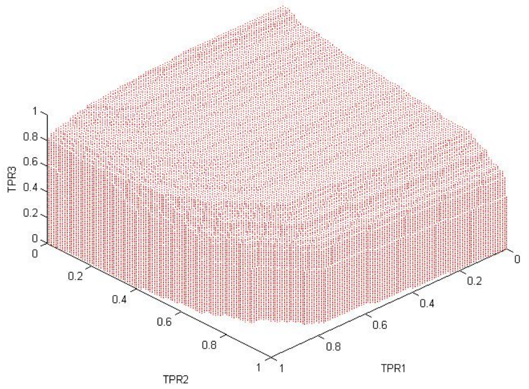
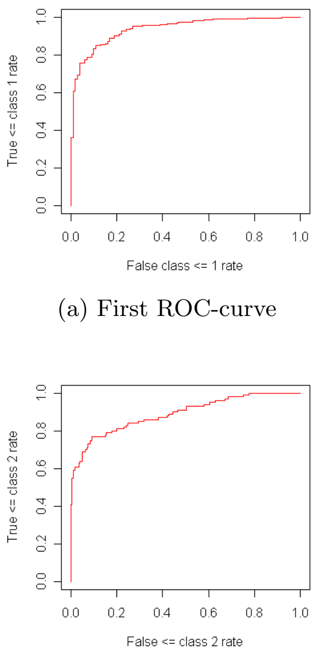


Figure 2: 3D ROC surface for the synthetic dataset

For three ordered classes the ROC surface can be visualized. We have constructed this ROC surface for a synthetic dataset. We sampled $3 * 100$ instances from 3 bivariate Gaussian clusters with consecutive ranks. The mean of the clusters was set to (10,10), (20,10) and (20,20) respectively, σ_1 and σ_2 were set to 5 for the first two clusters and were set to 7 for the last cluster. ρ was fixed to 0. This dataset is visualized in Figure 1. We used the support vector ordinal regression algorithm of [Chu & Keerthi, 2005] to estimate the function f, without looking at the thresholds. The ROC surface is shown in Figure 2.

Optimizing the AUC instead of accuracy has been suggested for binary classification, for example with gradient descent or a quadratic solver. However, the computation of $VUS_{ord}(f)$ has a large time complexity. The function I is evaluated $\prod_{k=1}^{r} n_k$ times, which is exponential in the number of classes r. As a consequence, minimizing $VUS_{ord}(f)$ will lead to a hard optimization problem.

We will look at approximations of $VUS_{ord}(f)$ which can be more easily transformed into a suitable loss function. The biggest problem is that all $r - tuples$ need to be verified. Much would be gained if only pairs of function values have to be correctly ranked in each evaluation of I. This is another way of evaluating the imposed ranking. We discuss here three approximations of VUS_{ord} that all reduce to I-evaluations



(a) First ROC-curve

(b) Second ROC-curve

Figure 3: The three dimensional ROC surface approximated by a set of two ROC-curves for the synthetic dataset.

with only one condition.

The first approximation $Cons(f)$ is directly deduced from the way the majority of existing ordinal regression models are constructed. With a function f and $r-1$ thresholds one could look at threshold b_k as providing the separation between the consecutive ranks k and $k+1$. Varying this threshold will change the proportion between objects predicted lower than or equal to class k and objects predicted higher than class k. This corresponds to measuring the non-weighted sum of $r-1$ two-dimensional ROC curves representing the trade-off between consecutive classes:

$$Cons(f) = \frac{1}{r-1}\sum_{l=1}^{r-1} AUC_l(f) \tag{9}$$

$$AUC_l(f) = \frac{1}{\sum_{i=1}^{l} n_i \sum_{j=l+1}^{n} n_j} \sum_{i:y_i \leq l} \sum_{j:y_j > l} I_{f(x_i)<f(x_j)}$$

The two ROC curves belonging to the synthetic dataset are shown in figure 3.

For a second approximation of $VUS_{ord}(f)$ we looked at the statistical literature. In nonparametric statistics the *Jonckheere-Terpstra* test is known as a more powerful alternative for a *Kruskal-Wallis* test for testing

$$H_0: \quad \mu_1 \leq \mu_2 \leq \ldots \leq \mu_r \tag{10}$$

versus the one side alternative

$$H_a: \quad \mu_1 \geq \mu_2 \geq \ldots \geq \mu_r \tag{11}$$

if there is a cdf F for which $F_k(x) = F(x-\mu_k)$). It is composed of a set of one sided WMW-tests:

$$JT = \sum_{i<j} WMW_{ij} \tag{12}$$

JT computes the WMW statistic for all possible pairs of classes, which is the same as computing the AUC for each pair of classes. This has been done for *one-versus-one* multi-class classification [Hand & Till, 2001], which gives rise to the following approximation:

$$Ovo(f) = \frac{2}{r(r-1)} \sum_{l<k} AUC_{lk}(f) \tag{13}$$

$$AUC_{lk}(f) = \frac{1}{n_l n_k} \sum_{i:y_i=l} \sum_{j:y_j=k} I_{f(x_i)<f(x_j)}$$

A third measure could exist of counting the number pairs that are correctly ranked among all possible pairs of data objects:

$$Pairs(f) = \frac{1}{\sum_{k<l} n_k n_l} \sum_{i=1}^{n} \sum_{j=1;y_i<y_j}^{n} I_{f(x_i)<f(x_j)} \tag{14}$$

A loss function based on (14) is used in the ordinal regression method of [Herbrich et al., 2000]. The difference with $Ovo(f)$ is that here a weighted average of the ROC areas for each of pair of classes is taken. The weights are the prior π_k probabilities of observing an object of class k, i.e.

$$Pairs(f) = \frac{2}{r(r-1)} \sum_{l<k} \pi_k \pi_l AUC_{lk}(f) \tag{15}$$

3. Simulation experiments

To see the characteristics of the different measures, we conducted some simulation experiments. In the first experiment we wanted to find out which values are obtained for different levels of separability and for an increasing number of classes. Therefore we assume that the function values of the model f can be represented by a distribution with cdf $F(x)$, in which the function values for the objects of class k are distributed with cdf $F_k(x) = F(x-kd)$. Furthermore we chose to sample

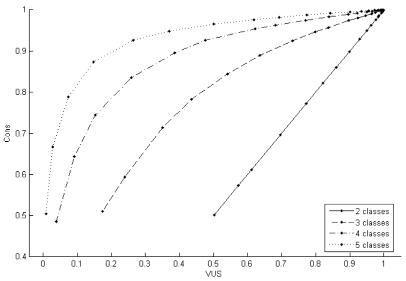


Figure 4: Relation between $VUS_{ord}(f)$ and $Cons(f)$ for $r = 1, ..., 5$ and $d = 0, ..., 5$ with step size 0.25. The values are averaged over 20 runs.

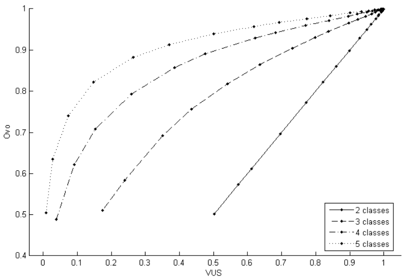


Figure 5: Relation between $VUS_{ord}(f)$ and $Ovo(f)$ for $r = 1, ..., 5$ and $d = 0, ..., 5$ with step size 0.25. The values are averaged over 20 runs.

from a Gaussian distribution with standard deviation $\sigma = 1$. So the function values conditioned on the labels are normally distributed with equidistant ordered means. Repeatedly 100 data points were sampled from each class while we increased the distance d between the means of consecutive clusters. We started at $d = 0$ (random classifier) and stopped at $d = 5$ (as good as perfect separation) with step size 0.25.

The results obtained for $VUS_{ord}(f)$, $Cons(f)$ and $Ovo(f)$ are graphically compared. In this simulation all classes have the same prior of occurring, so $Ovo(f)$ and $Pairs(f)$ will always have the same value. Consequently the results for $Pairs(f)$ are omitted. The relationship between $VUS_{ord}(f)$ and $Cons(f)$ on the one side and between $VUS_{ord}(f)$ and $Ovo(f)$ on the other side are shown in Figures 4 and 5. One can see that, as expected, the relation between $VUS_{ord}(f)$ and the other two measures is without doubt nonlinear. The expected value for $VUS_{ord}(f)$ heavily depends on the number of classes, while this is not the case for the approximations. The approximations all take an average over a set of two dimensional ROC-curves, so their expected value is never lower than a half, irrespective of the number of classes. Nevertheless, one can also see that $VUS_{ord}(f)$ converges rapidly to one when the distance between the subsequent means increases. In addition, $Cons(f)$ and $Ovo(f)$ behave quite similar in this simulation. This is also shown in Figure 6. Their observed values become more dissimilar when the number of classes increases.

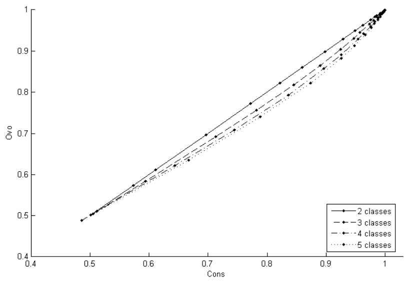


Figure 6: Relation between $Cons(f)$ and $Pairs(f)$ for $r = 1, ..., 5$ and $d = 0, ..., 5$ with step size 0.25. The values are averaged over 20 runs.

In a second experiment we wanted to investigate whether optimizing the various performance measures would lead to the same model. For two measures M_1 and M_2 this implies that

$$\forall f, f^* \in \mathcal{H} : M_1(f) < M_1(f^*) \Leftrightarrow M_2(f) < M_2(f^*) \tag{16}$$

$$\forall f, f^* \in \mathcal{H} : M_1(f) = M_1(f^*) \Leftrightarrow M_2(f) = M_2(f^*) \tag{17}$$

The following experiment was set up to test whether this property holds for the four measures. All measures only quantify the quality of the ordering of a dataset for a function f. For a dataset of size n there are $n!$ possible rankings of the objects, so evaluating them all is computationally intractable. Therefore we sampled randomly 1000 rankings from all possible orderings of the dataset. We assumed we had 50 samples per class with four ordered classes, resulting in a sample size of 200 objects and 200! possible rankings. The results are given in Figure 7, which shows the distributions of all measures together with pairwise scatter plots. All classes again have the same prior of occurring, so $Ovo(f)$ and $Pairs(f)$ have a perfect correlation. This is however not true for the other measures. One can clearly see that for no pair of measures conditions (16) or (17) hold. In general, $VUS_{ord}(f), Cons(f)$ and $Ovo(f)$ will have different maxima over a hypothesis space $\mathcal{H}$ and a given dataset. So, optimizing one of the proposed approximations of $VUS_{ord}(f)$ will give rise to different classifiers.

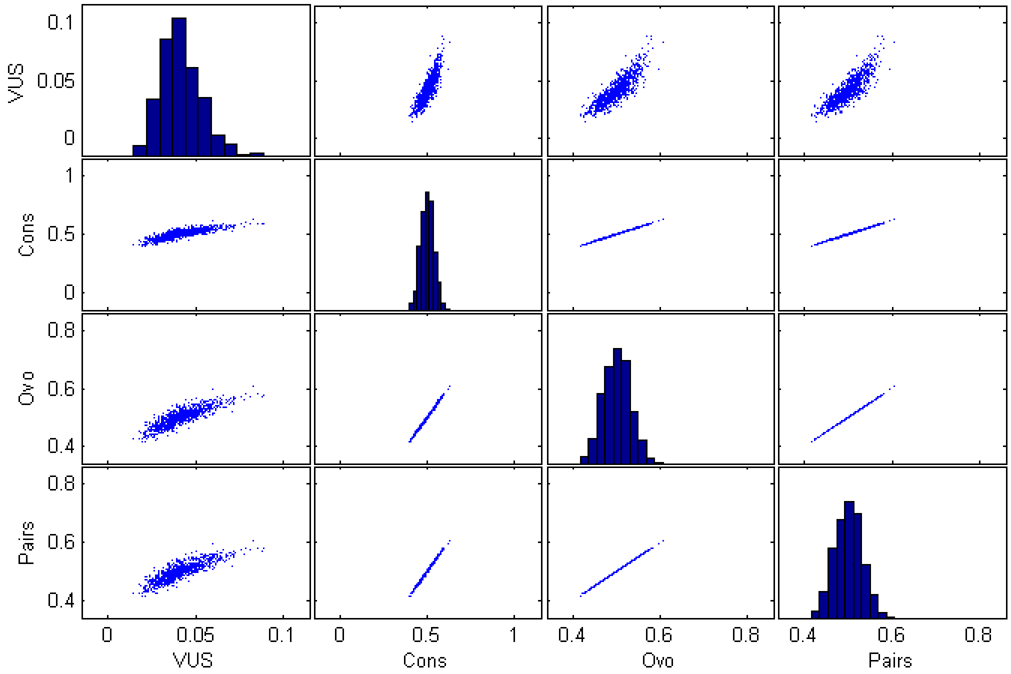


Figure 7: Histograms and pairwise scatter plots for all the measures.

4. Discussion and further research

In this article we argued that accuracy or mean absolute error are not the most powerful performance measures to evaluate ordinal regression models when misclassification costs are not equal for each class or when the data is unbalanced. Therefore we proposed some new measures, which extend binary and multiclass ROC analysis to ordinal regression. They all measure the quality of the ranking imposed by an ordinal regression model. First of all we showed that counting the number of ordered r-tuples in the ranking is equivalent to the area under the r-dimensional ROC curve spanned by the true positive rates of all classes. However, $VUS_{ord}(f)$ can't be transformed easily into a suitable loss function for learning algorithms, so three approximations were also analyzed. By simulation we showed that these four measures in general have a different distribution and that none of them is a monotone function of another. Further research will be devoted to converting measures like the area under the ROC curve into a loss function for a learning algorithm and to further analyse the characteristics of the presented measures.

Acknowledgments

The authors would like to thank Bruno Bauwens for useful discussions. Willem Waegeman is supported by a grant of the "Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen)".

References

Agresti, A. (2002). *Categorical Data Analysis, 2nd version*. John Wiley and Suns Publications.

Chu, W., & Gharhamani, Z. (2005). Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, *6*, 1019–1041.

Chu, W., & Keerthi, S. (2005). New approaches to support vector ordinal regression. *Proceedings of the International Conference on Machine Learning, Bonn, Germany* (pp. 321–328).

Cortes, C., & Mohri, M. (2003). AUC optimization versus error rate minimization. *In Advances in Neural Information Processing Systems, Vancouver, Canada*. The MIT Press.

Dreisetl, S., Ohno-Machado, L., & Binder, M. (2000).

Comparing three-class diagnostic tests by three-way roc analysis. *Medical Decision Making*, *20*, 323–331.

Ferri, C., Hernandez-Orallo, J., & Salido, M. (2003). Volume under ROC surface for multi-class problems. *In Proceedings of the European Conference on Machine Learning, Dubrovnik, Croatia* (pp. 108–120).

Flach, P. (2004). The many faces of ROC analysis in machine learning. Tutorial presented at the European Conference on Machine Learning, Valencia, Spain.

Hand, D. J., & Till, R. J. (2001). A simple generalization of the area under the ROC curve for multiple class problems. *Machine Learning*, *45*, 171–186.

Herbrich, R., Graepel, T., & Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers* (pp. 115–132). The MIT Press.

Nakas, C. T., & Yiannoutsos, C. T. (2004). Ordered multiple-class roc analysis with continuous measurements. *Statistics in Medicine*, *22*, 3437–3449.

Rennie, J. D. M., & Srebro, N. (2005). Loss functions for preference levels: Regression with discrete, ordered labels. *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, Edinburgh, Scotland* (pp. 180–186).

Shashua, A., & Levin, A. (2003). Ranking with large margin principle: Two approaches. *Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, Canada* (pp. 937–944). Cambridge MA: MIT Press.

Yan, L., Dodier, R., Mozer, M. C., & Wolniewiecz, R. (2003). Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. *Proceedings of the International Conference on Machine Learning, Washington D. C., USA* (pp. 848–855).

Upper and Lower Bounds of Area Under ROC Curves and Index of Discriminability of Classifier Performance

Shane T. Mueller STMUELLE@INDIANA.EDU
Indiana University, Department of Psychological and Brain Sciences, 1101 East 10th Street, Bloomington, IN 47404

Jun Zhang JUNZ@UMICH.EDU
University of Michigan, Department of Psychology, 530 Church Street Ann Arbor, MI 48109-1043

Abstract

Area under an ROC curve plays an important role in estimating discrimination performance – a well-known theorem by Green (1964) states that ROC area equals the percentage of correct in two-alternative forced-choice setting. When only single data point is available, the upper and lower bound of discrimination performance can be constructed based on the maximum and minimum area of legitimate ROC curves constrained to pass through that data point. This position paper, after reviewing a property of ROC curves parameterized by the likelihood-ratio, presents our recently derived formula of estimating such bounds (Zhang & Mueller, 2005).

1. Introduction

Signal detection theory (Green & Swets, 1966) is commonly used to interpret data from tasks in which stimuli (e.g., tones, medical images, emails) are presented to an operator (experimenter, medical examiner, classification algorithm), who must determine which one of two categories (high or low, malignant or benign, junk or real) the stimulus belongs in. These tasks yield a pair of measures of behavioral performance: the Hit Rate (H), also called "true positive" rate, and the False Alarm Rate (F), also called "false positive" rate. (The other two rates, those of Miss or "false negative" and of Correct Rejection or "true negative", are simply one minus H and F, respectively.) H and F are typically transformed into indices of sensitivity and bias based on assumptions about an underlying statistical model. A curve $c \mapsto (F(c), H(c))$ in the ROC (Receiver-Operating Characteristic) space is a collection of hit and false-alarm rates while the operator/receiver modifies the cutoff criterion c of accepting the input stimulus as belonging to one category versus another; often c is the likelihood ratio of the evidence favoring the two corresponding hypotheses, or a monotonic transformation thereof. In the machine learning context, we map the "operator/receiver" in the SDT sense to a "classification algorithm" or simply an "algorithm", the "stimulus" as an "input instance" or simply "instance" which carries one of the two class labels, and view c as a parameter of the algorithm which biases the output of the algorithm to favor one category or the other; the optimal setting of c is related to the cost structure, i.e., individual payoffs related to correct and incorrect classifications.

A well-known result in SDT is Green's Theorem, which relates the discrimination accuracy performance of an operator to the area under the operator's (i.e., the classification algorithm's) ROC curve. This so-called ROC area is thus a compact measure of how discriminable a classification algorithm is between binary-class inputs. Consequently, the performance of different algorithms can be compared by comparing their respective ROC areas.

Often, algorithms reported in the literature may not contain a tradeoff analysis of the Hit and False Alarm rates produced by varying parameters corresponding to the algorithm's bias. In these cases, the entire ROC curve of an algorithm may not be available — in some cases, only a few or even a single point (called "data point") in the ROC space is available. In this case, performance comparison across different algorithms becomes a question of comparing areas of possible ROC curves constrained to pass through these limited data

Appearing in *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, Pittsburgh, USA, 2006.

points.

In the mathematical psychology community, the problem of estimating area of ROC curves constrained to pass through a single data point is particularly well studied (Norman, 1964; Pollack & Norman, 1964; Pollack & Hsieh, 1969; Grier, 1971; Smith, 1995; Zhang & Mueller, 2005). These estimates of the ROC area do not assume the ROC curves to arise from any specific class of parametric models, and so these estimates are often referred to as a "non-parametric" indices of an operator's discriminability (sensitivity).[1] Typically, the upper and lower bounds of discriminability were obtained by considering the maximal and minimum ROC areas among the class of "admissible" ROC curves satisfying the data constraint. Interestingly, though the basic idea was very simple and advanced over 40 years ago (Pollack & Norman, 1964), the popular formula to calculate this index (Grier, 1971), dubbed A' in psychometrics and cognitive psychology literature, turned out to be erroneous, at least insofar as its commonly understood meaning is concerned; moreover, its purported correction (Smith, 1995), dubbed A'', also contained an error. These formulae incorrectly calculated the upper bound of admissible ROC curves, using either an ROC curve that was not admissible (Pollack & Norman, 1964), or one that was not the maximum for some points (Smith, 1995). Zhang and Mueller (2005) rectified the error and gave the definite answer to the question of non-parametric index of discriminability based on ROC areas.

In this note, we first review the notion of "proper" (or "admissible") ROC curves and prove a lemma basically stating that all ROC curves are proper/admissible when the likelihood functions (for the two hypotheses) used to construct the ROC curve are parameterized by the likelihood ratio (of those hypotheses). We then review Green's Theorem, which related area under an ROC curve to percentage correct in a two-alternative discrimination task. Finally, we present the upper and lower bounds on a 1-point constrained ROC area and reproduce some of the basic arguments underlying their derivation. All technical contents were taken from Zhang and Mueller (2005).

2. Slope of ROC curve and likelihood ratio

Recall that, in the traditional signal detection framework, an ROC curve $u_c \mapsto (F(u_c), H(u_c))$ is parameterized by the cutoff criteria value u_c along the measurement (evidence) axis based on which categorization decision is made. Given underlying signal distribution $f_s(u)$ and noise distribution $f_n(u)$ of measurement value u[2], a criterion-based decision rule, which dictates a "Yes" decision if $u > u_c$ and a "No" decision if $u < u_c$, will give rise to

$$\begin{aligned} H(u_c) &= \Pr(\text{Yes}|s) = \Pr(u > u_c|s) = \int_{u_c}^{\infty} f_s(u)du, \\ F(u_c) &= \Pr(\text{No}|s) = \Pr(u > u_c|n) = \int_{u_c}^{\infty} f_n(u)du. \end{aligned} \tag{1}$$

As u_c varies, so do H and F; they trace out the ROC curve. Its slope is

$$\left.\frac{dH}{dF}\right|_{F=F(u_c),H=H(u_c)} = \frac{H'(u_c)}{F'(u_c)} = \frac{f_s(u_c)}{f_n(u_c)} \equiv l(u_c)\ .$$

With an abuse of notation, we simply write

$$\frac{dH(u)}{dF(u)} = l(u)\,. \tag{2}$$

Note that in the basic setup, the likelihood ratio $l(u)$ as a function of decision criterion u (whose optimal setting depends on the prior odds and the payoff structure) need not be monotonic. Hence, the ROC curve $u \mapsto (F(u), H(u))$ need not be concave. We now introduce the notion of "proper (or admissible) ROC curves".

Definition 2.1. A *proper* (or *admissible*) ROC curve is a piece-wise continuous curve defined on the unit square $[0,1] \times [0,1]$ connecting the end points (0,0) and (1,1) with non-increasing slope.

The shape of a proper ROC curve is necessarily concave (downward-bending) connecting (0,0) and (1,1). It necessarily lies above the line $H = F$. Next we provide a sufficient and necessary condition for an ROC curve to be proper/admissible, that is, a concave function bending downward.

Lemma 2.2. *An ROC curve is proper if and only if the likelihood ratio $l(u)$ is a non-decreasing function of decision criterion u.*

[1] Though no parametric assumption is invoked in the derivation of these indices, the solution itself may correspond to certain models of underlying likelihood process, see MacMillan and Creelman, 1996. In other words, parameter-free here does not imply model-free.

[2] In machine learning applications, "signal" and "noise" simply refer the two category classes of inputs, and "signal distribution" and "noise distribution" are likelihood functions of the two classes.

Proof. Differentiate both sides of (2) with respect to u

$$\frac{dF}{du} \cdot \frac{d}{dF}\left(\frac{dH}{dF}\right) = \frac{dl}{du}.$$

Since, according to (1)

$$\frac{dF}{du} = -f_n(u) < 0,$$

therefore

$$\frac{dl}{du} \geq 0 \Longleftrightarrow \frac{d}{dF}\left(\frac{dH}{dF}\right) \leq 0$$

indicating that the slope of ROC curve is non-increasing, i.e., the ROC curve is proper. $\diamond$

Now it is well known (see Green & Swets, 1966) that a monotone transformation of measurement axis $u \mapsto v = g(u)$ does not change the shape of the ROC curve (since it is just a re-parameterization of the curve), so a proper ROC curve will remain proper after any monotone transformation. On the other hand, when $l(u)$ is not monotonic, one wonders whether there always exists a parameterization of any ROC curve to turn it into a proper one. Proposition 1 below shows that the answer is positive — the parameterization of the two likelihood functions is to use the likelihood ratio itself!

Proposition 2.3. (Slope monotonicity of ROC curves parameterized by likelihood-ratio). The slope of an ROC curve generated from a pair of likelihood functions $(F(l_c), H(l_c))$, when parameterized by the likelihood-ratio l_c as the decision criterion, equals the likelihood-ratio value at each criterion point l_c

$$\frac{dH(l_c)}{dF(l_c)} = l_c. \tag{3}$$

Proof. When likelihood-ratio l_c is used the decision cutoff criterion, the corresponding hit rate (H) and false-alarm rate (F) are

$$\begin{aligned} H(l_c) &= \int_{\{u:l(u)>l_c\}} f_s(u)du, \\ F(l_c) &= \int_{\{u:l(u)>l_c\}} f_n(u)du. \end{aligned}$$

Note that here u is to be understood as (in general) a multi-dimensional vector, and du should be understood accordingly. Writing out $H(l_c + \delta l) - H(l_c) \equiv \delta H(l_c)$ explicitly,

$$\delta H(l_c) = \int_{\{u:l(u)>l_c+\delta l\}} f_s(u)du - \int_{\{u:l(u)>l_c\}} f_s(u)du$$

$$= -\int_{\{u:l_c<l(u)<l_c+\delta l\}} f_s(u)du \simeq -\int_{\{u:l(u)=l_c\}} f_s(u)\,\delta u$$

where the last integral $\int \delta u$ is carried out on the set $\partial \equiv \{u : l(u) = l_c\}$, i.e., across all u's that satisfy $l(u) = l_c$ with given l_c. Similarly,

$$\delta F(l_c) \simeq -\int_{\{u:l(u)=l_c\}} f_n(u)\,\delta u\,.$$

Now, for all $u \in \partial$

$$\frac{f_s(u)}{f_n(u)} = l(u) = l_c$$

is constant, from an elementary theorem on ratios, which says that if $a_i/b_i = c$ for $i \in I$ (where c is a constant and I is an index set), then $(\sum_{i\in I} a_i)/(\sum_{i\in I} b_i) = c$,

$$\frac{\delta H(l_c)}{\delta F(l_c)} = \frac{\int_\partial f_s(u)\,\delta u}{\int_\partial f_n(u)\,\delta u} = \left.\frac{f_s(u)\,\delta u}{f_n(u)\,\delta u}\right|_{u\in\partial} = l_c\,.$$

Taking the limit $\delta l \to 0$ yields (3). $\diamond$.

Proposition 2.3 shows that the slope of ROC curve is always equal the likelihood-ratio value regardless how it is parameterized, i.e., whether the likelihood-ratio is monotonically or non-monotonically related to the evidence u and whether u is uni- or multi-dimensional. The ROC curve is a signature of a criterion-based decision rule, as captured succinctly by the expression

$$\frac{dH(l)}{dF(l)} = l\,.$$

Since $H(l)$ and $F(l)$ give the proportion of hits and false alarms when a decision-maker says "Yes" whenever the likelihood-ratio (of the data) exceeds l, then $\delta H = H(l + \delta l) - H(l)$, $\delta F = F(l + \delta l) - F(l)$ are the amount of hits and false-alarms if he says "Yes" only when the likelihood-ratio falls within the interval $(l, l+\delta l)$. Their ratio is of course simply the likelihood-ratio.

Under the likelihood-ratio parameterization, the signal distribution $f_s(l) = -dH/dl$ and the noise distribution $f_n(l) = -dF/dl$ can be shown to satisfy

$$E_s\{l\} = \int_{l=0}^{l=\infty} l f_s(l)dl \geq 1 = \int_{l=0}^{l=\infty} l f_n(l)dl = E_n\{l\}.$$

The shape of the ROC curve is determined by $H(l)$ or $F(l)$. In fact, its curvature is

$$\kappa = \frac{d}{dl}\left(\frac{dH}{dF}\right) / \left(1 + \left(\frac{dH}{dF}\right)^2\right) = \frac{1}{1+l^2}\,.$$

3. Green's Theorem and area under ROC curves

The above sections studies the likelihood-ratio classifier in a single-instance paradigm — upon receiving an input instance, the likelihood functions in favor of each hypothesis are evaluated and compared with a pre-set criterion to yield a decision of class label. Both prior odds and payoff structure can affect the optimal setting of likelihood ratio criterion l_c by which class label is assigned. On the other hand, in two-alternative force choice paradigms with two two instances, each instance is drawn from one category, and the operator must match them to their proper categories. For example, an auditory signal may be present in one of two temporal intervals, and the operator must determine which interval contains the signal and which contains noise. In this case, the likelihood-ratio classifier, after computing the likelihood-ratios for each of the instances, simply compares the two likelihood-ratio values l_a and l_b, and matches them to the two class labels based on whether $l_a < l_b$ or $l_a > l_b$. It turns out that the performance of the likelihood-ratio classifier under the single-instance paradigm ("detection paradigm") and under the two-instance forced-choice paradigm ("identification paradigm") are related by a theorem first proven by Green (1964).

Proposition 3.1. (Green, 1964). Under the likelihood-ratio classifier, the area under an ROC curve in a single-observation classification paradigm is equal to the overall probability correct in the two-alternative force choice paradigm.

Proof. Following the decision rule of the likelihood-ratio classifier, the percentage of correctly ("PC") matching the two input instances to the two categories is

$$\begin{aligned}
\text{PC} &= \int\int_{0\le l_b \le l_a \le \infty} f_s(l_a)\, f_n(l_b)\, dl_a\, dl_b \\
&= \int_0^{\infty} \left(\int_{l_b}^{\infty} f_s(l_a)\, dl_a \right) f_n(l_b)\, dl_b \\
&= \int_{l_b=0}^{l_b=\infty} H(l_b)\, dF(l_b) = \int_{F=0}^{F=1} H dF,
\end{aligned}$$

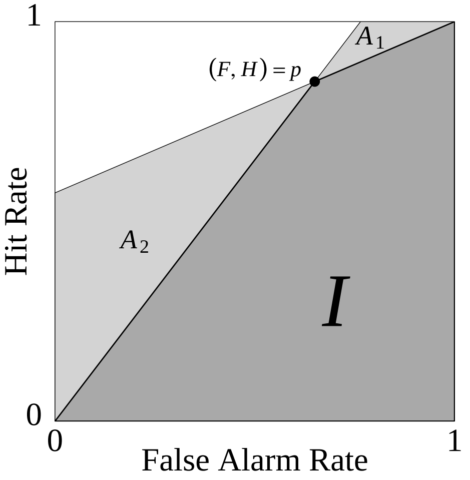
which is the area under the ROC curve $l_c \mapsto (F(l_c), H(l_c))$. ◇

Green's Theorem (Proposition 3.1) motivates one to use the area under an ROC curve to as a measure of discriminability performance of the operator. When multiple pairs of hit and false alarm rates $(F_i, H_i)_{i=1,2,\cdots}$ (with $F_1 < F_2 < \cdots, H_1 < H_2 < \cdots$) are available, all from the same operator but under manipulation of prior odds and/or payoff structure and with the constraints

$$0 \le \cdots \le \frac{H_3 - H_2}{F_3 - F_2} \le \frac{H_2 - H_1}{F_2 - F_1} \le \infty,$$

then it is possible to construct proper ROC curves passing through these points, and the bounds for their area can be constructed. The question of finding the areal bounds of ROC curves passing through a single data point has received special attention in the past (since Norman, 1964), because as more data points are added, the uncertain in ROC area (difference between the upper and lower bounds of area measure) decreases. We discuss the bounds of 1-point constrained ROC area in the next sections.

Figure 1. Proper ROC curves through point p must lie within or on the boundaries of the light shaded regions A_1 and A_2. The minimum-area proper ROC curve through p lies on the boundary of region I.

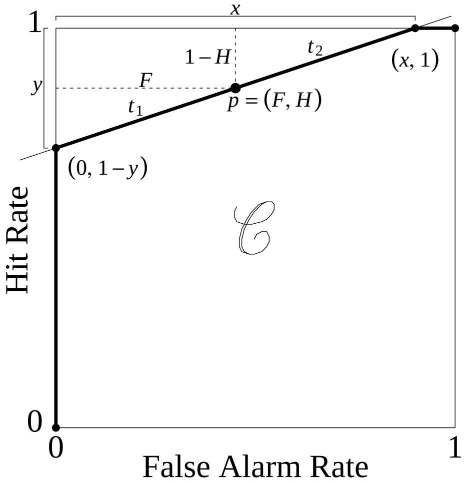


4. ROC curves constrained to pass through a data point

When the data point $p = (F, H)$ is fixed, the non-increasing property of the slope (Corollary 1) immediately leads to the conclusion that all proper ROC curves must fall within or on the bounds of light shaded regions A_1 and A_2 (shown in Figure 1). This observation was first made in Norman (1964). The proper ROC curve with the smallest area lies on the boundary between I and A_1 (to the right of p) and A_2 (to the left of p), whereas the proper ROC curve with the largest area lies within or on the boundaries of A_1 and A_2.

Pollack and Norman (1964) proposed to use the average of the areas $A_1 + I$ and $A_2 + I$ as an index of discriminability (so-called A'), which turns out to equal

Figure 2. Example of a proper ROC curve through p. The ROC curve $\mathcal{C}$, a piecewise linear curve denoted by the dark outline, is formed by following a path from $(0,0)$ to $(0,1-y)$ to $(x,1)$ (along a straight line that passes through $p=(F,H)$) and on to $(1,1)$.



$1/2+(H-F)(1+H-F)/(4H(1-F))$ (Grier, 1971). However, the A' index was later mistakenly believed to represent the *average* of the maximal and minimum areas of proper ROC curves constrained to pass through $p=(F,H)$. Rewriting

$$\frac{1}{2}((A_1+I)+(A_2+I))=\frac{1}{2}(I+(A_1+A_2+I)),$$

the mis-conceptualization probably arose from (incorrectly) taking the area A_1+A_2+I to be the maximal area of 1-point constrained proper ROC curves while (correcting) taking the are I to be the minimal area of such ROC curves, see Figure 1. It was Smith (1995) who first pointed out this long, but mistakenly-held belief, and proceeded to derive the true upper bound (maximal area) of proper ROC curves, to be denoted A_+. Smith claimed that, depending on whether p is to the left or right of the negative diagonal $H+F=1$, A_+ is the larger of $I+A_1$ and $I+A_2$. This conclusion, unfortunately, is still erroneous when p is in the upper left quadrant of ROC space (i.e., $F<.5$ and $H>.5$) — in this region, neither $I+A_1$ nor $I+A_2$ represents the upper bound of all proper ROC curves passing through p.

5. Lower and upper bound of area of 1-point constrained proper ROC curves

The lower bound A_- of the area of all proper ROC curves constrained to pass through a given point $p=(F,H)$ can be derived easily (the area labelled as I in Figure 1):

$$A_-=\frac{1}{2}(1+H-F).$$

In Zhang and Mueller (2005), the expression was derived for the upper bound A_+ of such ROC area.

Proposition 5.1. (Upper Bound of ROC Area). The areal upper bound A_+ of proper ROC curves constrained to pass through one data point $p=(F,H)$ is

$$A_+=\begin{cases}1-2H(1-F) & \text{if} \quad F<0.5<H\,,\\ \frac{1-F}{2H} & \text{if} \quad F<H<0.5\,,\\ 1-\frac{1-H}{2(1-F)} & \text{if} \quad 0.5<F<H\,.\end{cases}$$

Proof. See Zhang and Mueller (2005). ⋄

The ROC curve achieving the maximal area generally consists of three segments (as depicted in Figure 2), with the data point p *bisecting* the middle segment – in other words, $t_1=t_2$ in Figure 2. When p falls in the $F<H<0.5$ ($0.5<F<H$, resp) region, then the vertical (horizontal, resp) segment of the maximal-area ROC curve degenerates to the end point $(0,0)$ ($(1,1)$, resp), corresponding to $y=1$ ($x=1$, resp) in Figure 2.

With the upper and lower bounds on ROC area derived, Figure 3 plots the difference between these bounds — that is, the uncertainty in the area of proper ROC curves that can pass through each point. The figure shows that the smallest differences occur along the positive and negative diagonals of ROC space, especially for points close to $(0,1)$ and $(.5,.5)$. The points where there is the greatest difference between the lower and upper bounds of ROC area are near the lines $H=0$ and $F=1$. Thus, data observed near these edges of ROC space can be passed through by proper ROC curves with a large variability of underlying areas. Consequently, care should be taken when trying to infer the ROC curve of the observer/algorithm when the only known data point regarding its performance (under a single parameter setting) falls within this region.

By averaging the upper and lower bound $A=(A_++A_-)/2$, we can derive the (non-parametric) index of discriminability performance

$$A=\begin{cases}\frac{3}{4}+\frac{H-F}{4}-F(1-H) & \text{if} \quad F\le 0.5\le H\,;\\ \frac{3}{4}+\frac{H-F}{4}-\frac{F}{4H} & \text{if} \quad F<H<0.5\,;\\ \frac{3}{4}+\frac{H-F}{4}-\frac{1-H}{4(1-F)} & \text{if} \quad 0.5<F<H\,.\end{cases}$$

One way to examine A is to plot the "iso-discriminability" curve, i.e, the combinations of F and

Figure 3. Difference between the lower and upper bounds of area of proper ROC curves through every point in ROC space. Lighter regions indicate smaller differences.

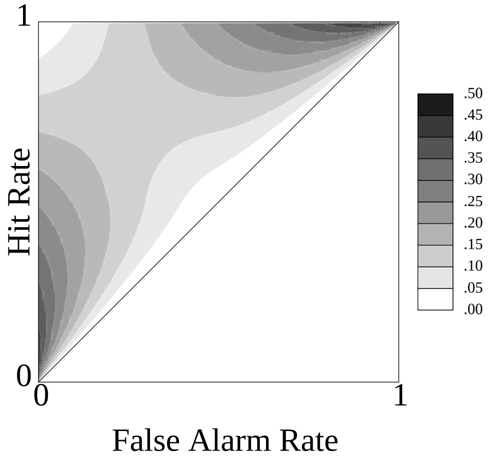
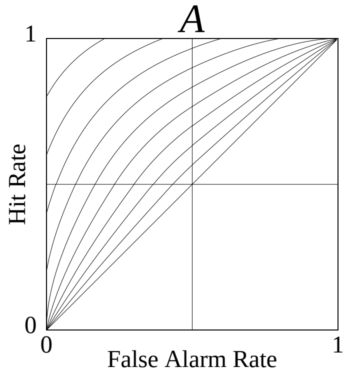


Figure 4. Iso-discriminability contours in ROC space. Each line corresponds to combinations of F and H that produce equal values of A, in increments of 0.05.



H will produce a given value of A. The topography of A in ROC space can be mapped by drawing isopleths for its different constant values. Figure 4 shows these topographic maps for A.

Finally, since the slope of any proper ROC curve is related to the likelihood ratio of the underlying distributions, we can construct an index of decision bias (Zhang & Mueller, 2005), denoted b, as being orthogonal to the slope of the constant-A curve (called b):

$$b = \begin{cases} \frac{5-4H}{1+4F} & \text{if} \quad F \le 0.5 \le H\,; \\ \frac{H^2+H}{H^2+F} & \text{if} \quad F < H < 0.5\,; \\ \frac{(1-F)^2+1-H}{(1-F)^2+1-F} & \text{if} \quad 0.5 < F < H\,. \end{cases}$$

6. Conclusion

We showed that the relationship of ROC slope to likelihood-ratio is a fundamental relation in ROC analysis, as it is invariant with respect to any continuous reparameterization of the stimulus, including non-monotonic mapping of uni-dimensional and multi-dimensional evidence in general. We provided an upper bound for the area of proper ROC curves passing through a data point and, together with the known lower bound, a non-parametric estimate of discriminability as defined by the average of maximal and minimum ROC areas.

References

Green, D. M. (1964). General prediction relating yes-no and forced-choice results. *Journal of the Acoustical Society of America, A*, *36*, 1024.

Green, D. M., & Swets, J. A. (1964). *Signal detection theory and psychophysics.* New York: John Wiley & Sons.

Grier, J. B. (1971). Nonparametric indexes for sensitivity and bias: computing formulas. *Psychological Bulletin*, *75*, 424–429.

Macmillan, N. A., & Creelman, C. D. (1996). Triangles in roc space: History and theory of “nonparametric” measures of sensitivity and response bias. *Psychonomic Bulletin & Review*, *3*, 164–170.

Norman, D. A. (1964). A comparison of data obtained with different false-alarm rates. *Psychological Review*, *71*, 243–246.

Peterson, W. W., Birdsall, T. G., & Fox, W. C. (1954). The theory of signal detectability. *Transactions of the IRE Professional Group on Information Theory* (pp. 171–212).

Pollack, I., & Hsieh, R. (1969). Sampling variability of the area under roc curve and d'_e. *Psychological Bulletin*, *1*, 161–173.

Pollack, I., & Norman, D. A. (1964). Non-parametric analysis of recognition experiments. *Psychonomic Science*, *1*, 125–126.

Smith, W. D. (1995). Clarification of sensitivity measure A'. *Journal of Mathematical Psychology*, *39*, 82–89.

Zhang, J., & Mueller, S. T. (2005). A note on roc analysis and non-parametric estimate of sensitivity. *Psychometrika*, *70*, 145–154.