Optimization of the Area Under the ROC Curve

Cristiano Leite Castro
crislcastro@gmail.com

Antonio Padua Braga
apbraga@cpdee.ufmg.br

Universidade Federal de Minas Gerais.
Depto. Engenharia Eletrônica, LITC.
Av. Antônio Carlos, 6.627 - Campus UFMG Pampulha 30.161-970,
Belo Horizonte, MG, Brasil

Abstract

In this paper, we propose a new binary classification algorithm (AUCtron), based on gradient descent learning, that directly optimizes AUC (Area Under the ROC Curve). We compare it with a linear classifier and with AUCsplit proposed by [5]. The AUCtron algorithm implicitly considers class prior probabilities in the decision criteria. Our results demonstrated that AUC is a sensitive enough metric that when used in small and imbalanced data sets may lead to a better separation.

1. Introduction

In many real-world binary classification problems, classification accuracy (correct classification rate) is not the most important measure to evaluate classifier performance. For instance, in medical decision-making problems, ranking the observations is more important than classifying each observation as positive or negative. Consider, for example, the list of scores (posterior probabilities) returned by a diagnostic classifier. Such a list may contain several patients but, in practice, only the most critical ones should be selected for immediate treatment, what turns out that, for this kind of problem, ranking the patients is more critical than classifying them as sick or healthy.

Moreover, the accuracy and the error rate can be misleading when the prior probabilities of the classes are too different [1, 16, 13]. For example, it is straightforward to create a classifier having 98% accuracy (or 2% error rate) if the data set has a majority class with 98% of the total number of observations, by simply classifying every new observation as belonging to the majority class. In medical decision-making problems, for instance, small, missing values and heavily imbalanced data sets are common.

In fact, the accuracy (or error rate) considers different classification errors as equally important. In general, many traditional learning algorithms are not prepared to induce a classifier when the class distribution is highly imbalanced. Frequently, the classifier has a good classification accuracy for the majority class (false positive error rate), but its accuracy for the minority class (false negative error rate) is unacceptable. An alternative way of evaluating performance of a classifier that does not confuse the false positive and false negative error rates is the Receiver Operating Characteristic (ROC) curves [3].

ROC Curves were originally developed in signal detection theory and over the last few years have been used in many applications of machine learning and data mining for model evaluation and selection [15, 5, 4]. The ROC curve for a binary classification plots the true positive rate as a function of the false positive rate. Assuming that a classifier produces a continuous output (posterior probabilities), then the output must be thresholded to label each observation as positive or negative. Thus, for each setting of the decision threshold, a new point (true positive and false positive rate) is obtained. By varying the decision threshold over a range of the classifier output (from 0 to 1, for example), the ROC curve is produced. Figure 1, shows an example of a ROC curve. The more inclined the curve is toward the upper left corner, the better is the classifier's ability to discriminate between positive and negative classes.

The ROC curve has the advantage of being independent of class prior probabilities. In addition, the area under the ROC curve (AUC) is a robust measure of classifier discrimination performance, regardless of the decision threshold, being also closely related to the ranking quality of the classification as shown more formally in Section 2. The ROC curve and the AUC have been used extensively to evaluate classifier performance, mainly when classification problems have imbalanced classes and it is aimed at attaining as many as possible observations of the minority class above the threshold (ranking quality). However, the usual objective function optimized by most classification algorithms is the error rate and not AUC. These algorithms indirectly op-

1522-4899/08 $25.00 © 2008 IEEE
DOI 10.1109/SBRN.2008.25

IEEE computer society

timize AUC by minimizing the error rate, what is likely to produce suboptimal results.

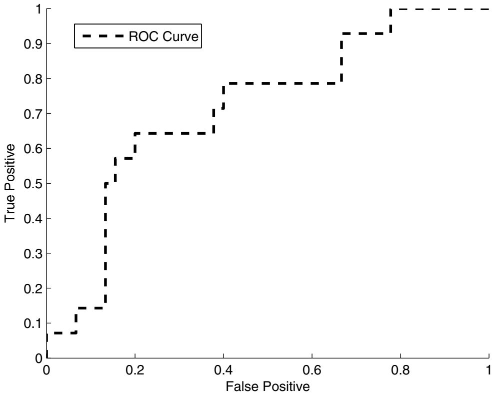

Figure 1. Typical ROC curve.

Recently, several algorithms have been proposed to address this problem. Many algorithms can be extended to incorporate unbalanced misclassification costs via linear loss functions in context of Support Vector Machines (SVMs) [14, 10]. Also, multi-objectives evolutionary algorithms have been proposed for optimizing sensitivity (true positive rate) and specificity (false negative rate) [18, 6]. Methods for optimizing the AUC value locally have been developed in the context of decision trees [5]. Other algorithms have been proposed to obtain approximations of the global AUC value such as [19, 8, 17, 9], but, in general these algorithms did not obtain AUC values significantly better than those obtained by an algorithm designed to minimize the error rate. In addition, for non-linear metrics like AUC, the few previous attempts towards their direct optimization showed their computational difficulty.

In this paper, we present a detailed analysis of the relationship between AUC and accuracy values based on the expressions of the expected value and the variance of the AUC for a fixed error rate, as proposed by [2]. Moreover, we propose a new binary classification algorithm, based on gradient descent learning, that directly optimizes AUC from a global approximation proposed by [19]. We compare it with a linear classifier and with AUCsplit proposed by [5]. The experiments were performed using several data sets from the UC Irvine Repository. Most of the data sets chosen for the experiments were from medical decision-making problems.

2 AUC as a Performance Measure

This Section formally presents the definition and the properties of the Area under ROC curve (AUC).

2.1 Definition of AUC

Consider a data set of iid observations, drawn from a population. P observations belong to the minority, or positive class, and Q to the majority, or negative class. The positive observations are denoted by vectors $(\vec{x_j^+}, j = 1 \dots P)$, and the negative ones by $(\vec{x_k^-}, k = 1 \dots Q)$, where each element of the vector $\vec{x_j^+}$ or $\vec{x_k^-}$ represents the value of a feature for the corresponding observation. The data set has m feature variables, so $\vec{x_j^+} = \{\vec{x_{ij}^+}, i = 1 \dots m\}$ and $\vec{x_{ij}^+}$ is the j^{th} instance of random variable X_i^+. Equivalent definitions hold for the majority class.

The AUC of a classifier on a given data set can be expressed as the probability $P(f(\vec{X^+}) > f(\vec{X^-}))$, where $f(\vec{X^+})$ is the random variable corresponding to the distribution of the outputs (scores) of the classifier for the positive observations and $f(\vec{X^-})$ the one corresponding to the negative observations. This expression is equivalent to value of the Wilcoxon-Mann-Whitney Statistic [12, 19] illustrated by Equation 1, which represents the expression of that probability in the discrete case.

$$\widehat{AUC}(f) = \frac{1}{PQ}(\sum_{j=1}^{P}\sum_{k=1}^{Q} g(f(\vec{x_j^+}) - f(\vec{x_k^-}))) \tag{1}$$

where the function $g(x)$ is defined by,

$$g(x) = \begin{cases} 0 & \text{if } x < 0, \\ 0.5 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{2}$$

Thus, the AUC can be viewed as a measure based on pairwise comparisons between classifications of the two classes. With a perfect ranking, all positive observations are ranked higher than the negative ones and $\widehat{AUC}(f) = 1$.

2.2 Properties of the AUC

For a given classification threshold θ, given that all observations with score $\geq \theta$ are labeled positive and other observations are labeled negative, one can determine which observations are labeled correctly and which are errors. If k is the total number of errors in the data set, the classification accuracy is $1 - \frac{k}{P+Q}$. In contrast, the AUC takes into consideration both k and the scores of the observations which are classified incorrectly. Thus, different classifiers may have the same error rate but different AUC values. Based on these assumptions, [2] proposed the first exact expression of the expected value and the variance of the AUC for a fixed error rate.

Assuming that the number of errors k is fixed. For a given binary classifier with P positive observations and Q

negative observations, there may be x, $0 \leq x \leq k$, false positive examples and, therefore, $k-x$ false negative examples. For a fixed x, the average value of AUC is given by Equation 3,

$$\langle A\rangle_x = \left(1 - \frac{\frac{x}{Q} + \frac{k-x}{P}}{2}\right) \tag{3}$$

Note that Equation 3 shows that the average AUC value for a given x is simply one minus the average of the accuracy rates for the positive and negative classes.

As mentioned earlier, an arbitrary reordering of the observations with outputs of more than θ clearly does not affect the accuracy but leads to different AUC values. Similarly, one may reorder the observations with scores less than θ without changing the error rate. Consequently, to calculate the average overall possible values of x, it is necessary to weight the expression of Equation 3 with the total number of possible classifications for a given x. Hence, for a given binary classifier with P positive examples and Q negative examples, the expected value of AUC over all classifications with k errors is given by Equation 4 [2],

$$\langle A\rangle = \frac{\sum_{x=0}^{k} \binom{M}{x}\binom{N}{k-x}\left(1 - \frac{\frac{x}{Q}+\frac{k-x}{P}}{2}\right)}{\sum_{x=0}^{k} \binom{M}{x}\binom{N}{k-x}}, \tag{4}$$

where M and N are the total numbers of observations classified as positive and negative respectively.

The variance of the AUC $\sigma^2(A)$ over all classifications with k errors is given by Equation 5 [2].

$$\sigma^2(A) = F\left(\left(1 - \frac{\frac{x}{Q}+\frac{k-x}{P}}{2}\right)^2\right) - F\left(\left(1 - \frac{\frac{x}{Q}+\frac{k-x}{P}}{2}\right)\right)^2 + F\left(\frac{Px^2 + Q(k-x)^2 + (P(P+1)x)}{12P^2Q^2}\right) + F\left(\frac{Q(k-x)(Q+1) - 2x(k-x)(P+Q+1)}{12P^2Q^2}\right) \tag{5}$$

The same methodology is used to calculate the variance of the AUC. For each term Y of the Equation 5, it is necessary to compute the average overall possible values of x. As before, we begin by determining the average of the term Y for a given value of x and then use the function $F(.)$, defined by Equation 6, to weight this average with the number of possible ranks of the x misclassified observations, i.e., $\langle Y\rangle = F(\langle Y\rangle_x)$. More details about the expressions of the expected value and the variance of the AUC can be found on [2].

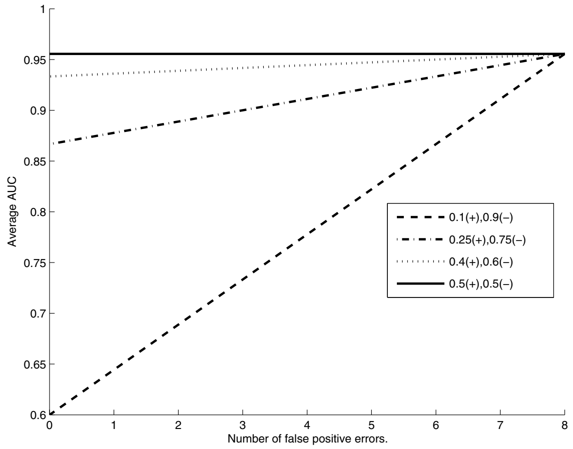

Figure 2. Average value of the AUC as a function of the number of false positive errors.

$$F(Z) = \frac{\sum_{x=0}^{k} \binom{M}{x}\binom{N}{k-x} Z}{\sum_{x=0}^{k} \binom{M}{x}\binom{N}{k-x}} \tag{6}$$

3 Methodology and Experiments

In this Section, we described how the experiments were performed. Firstly, we presented a detailed analysis of the relationship between classification accuracy and AUC. After that, we designed an efficient algorithm which directly optimizes the AUC based on gradient descent and applied it to solve binary classification problems extracted from UCI repository.

3.1 Classification Accuracy and AUC

AUC differs from accuracy because it is sensitive to class imbalance and also to the rate of errors occurring in each class. To argue this point, we conducted a detailed analysis of the relationship between AUC and accuracy. Firstly, the average AUC was calculated for four data sets with different class ratios $r = \frac{P}{P+Q}$. For each value of r, the number of false positive errors x was decremented by uniform intervals. The total number of errors k was kept constant. The results of this experiment are illustrated in Figure 2.

Observe that when prior probabilities of the classes are very different ($P \neq Q$), the average AUC decreases with the number of false positives. Beforehand, the accuracy assumes constant values for each r since the total number of errors k and the total number of observations was kept invariable.

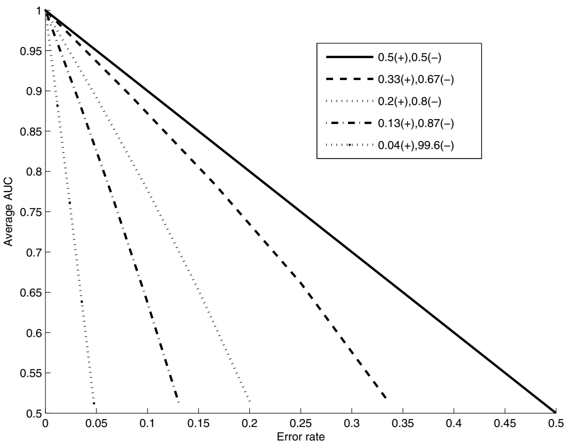

Figure 3. Average value of the AUC as a function of the error rate.

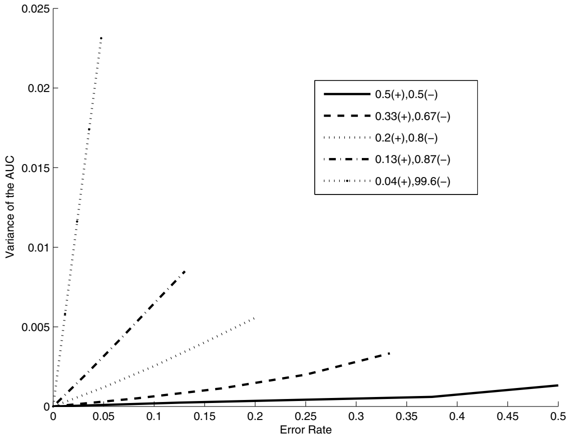

Figure 4. Average value of the AUC as a function of the error rate.

Figures 3 and 4 show, respectively, the average and the variance of the AUC as a function of the error rate. In figure 3, the curves were obtained by increasing the accuracy from r to 1. Notice that when $P = Q$, balanced classes, the accuracy coincides with the average AUC. In Figure 4, each curve was obtained by fixing the ratio class r and varying the number of errors k from 1 to the size of the minority class.

3.2 AUCtron Algorithm

Most traditional learning algorithms are designed to optimize the mean square error (mse). This corresponds to one single point on the ROC curve. However, it is desirable to optimize the classifier over a wide range of points of the ROC Curve. The AUC is a metric that simulates the overall ROC measure. Here, we propose an efficient algorithm that directly optimizes $\widehat{AUC}(f)$ of Equation 1. A smoothing method, the polynomial function in Equation 7, was used to approximate the non differentiable function $\widehat{AUC}(f)$ replacing the indicator function $g(x)$ of Equation 2 [19].

$$R(x) = \begin{cases} (-(x-\gamma))^p & \text{if } x < \gamma, \\ 0 & : \text{otherwise.} \end{cases} \tag{7}$$

Thus, we designed an algorithm called AUCtron that optimizes directly AUC from a global approximation $\widehat{AUC_{pol}}(f)$. AUCtron was used to train a binary classifier based on Perceptron nodes with logistic sigmoid (softmax) outputs. Therefore, formally we have to optimize the problem defined as,

$$\vec{w}_{opt} = argmin \ \widehat{AUC_{pol}}(f) \tag{8}$$

The goal of the learning process is to obtain the vector $\vec{w}_{opt} = [b_{opt}; w_{opt}]$ that minimizes the objective function $\widehat{AUC_{pol}}(f)$ and consequently maximizes the ROC curve.

To optimize the objective function, AUCtron uses the gradient descent method. We calculate the gradient vector for each epoch (iteration) and take a small step in its opposite direction. The calculating of optimal step, or learning rate η, is performed by Golden Section method [11]. Thus, we guarantee that the value of $\widehat{AUC_{pol}}(f)$ decreases until a minimum is achieved. It was observed that for a wide variety of data sets local minimums are unlikely. Therefore, the selection of a starting point for AUCtron should have minimal impact on the final solution. It may however have an impact on computational complexity.

[19] observed that $\widehat{AUC_{pol}}(f)$ is quite insensitive to γ parameter of Equation 7. In general, one can choose a value between 0.1 and 0.7 for γ. Also, we have found that $p = 2$ or $p = 3$ achieves similar, and generally the best results. The AUCtron is described by Algorithm 1,

4 Results

Table 1 compares AUC values obtained by AUCtron, MSEtron and AUCsplit algorithms. The AUCsplit algorithm proposed by [5] locally optimizes the AUC in context of decision trees. The MSEtron is the typical algorithm of Perceptron that uses the mean squared error as objective function [7]. All the experiments were performed using data sets from the UC Irvine Repository. Table 2 shows the values of sensitivity and specificity achieved by AUCtron and MSEtron. The results presented in Tables 1 and 2 represent the average values of several executions of algorithms for different training and test sets. The class ratios were kept invariable for training and test sets for all experiments.

Algorithm 1 : Pseudocode of AUCtron Algorithm

```
$\vec{w}_1 \Leftarrow [b_{ini}; w_{ini}]$ {Initializes parameters}
$epoch \Leftarrow 1$

while $(epoch < MaxEpochs)$ do

    $\vec{\nabla} AUC_{epoch} \Leftarrow Gradient(\vec{w}_{epoch})$
    $\eta_{epoch} \Leftarrow GoldenSection(\vec{w}_{epoch}, \vec{\nabla} AUC_{epoch})$
    $\vec{\Delta} w_{epoch} \Leftarrow \eta_{epoch} \cdot \vec{\nabla} AUC_{epoch}$
    $\vec{w}_{epoch+1} \Leftarrow \vec{w}_{epoch} - \vec{\Delta} w_{epoch}$
    $epoch \Leftarrow epoch + 1$

end while

return $\vec{w}_{opt} \Leftarrow \vec{w}_{epoch}$
```

Table 1. Comparison of AUC values between all the algorithms. The values for AUCsplit are from [5].

Data Set	ratio (%)	AUCsplit AUC (%)	MSEtron AUC (%)	AUCtron AUC (%)
Breast	23.7	59.3	72.2	80.1
Pima	34.9	76.7	77.9	79.3
SPECTF	20.4	-	84.1	83.7
Ionosphere	35.9	89.7	82.0	83.4
Cleveland	39.6	-	86.9	91.2
Hungarian	41.3	-	83.8	87.8

Table 2. Comparison of sensitivity and specificity between MSEtron and AUCtron algorithms.

Data Set	MSEtron		AUCtron	
	sens	spec	sens	spec
Breast	0.43	0.87	0.79	0.64
Pima	0.45	0.87	0.55	0.91
SPECTF	0.38	0.91	0.75	0.73
Ionosphere	0.63	1.0	0.71	0.94
Cleveland	0.52	0.88	0.89	0.82
Hungarian	0.46	0.79	0.84	0.81

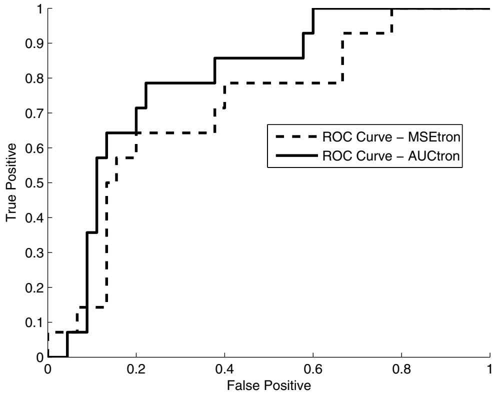

Figure 5. The ROC curves over Breast Cancer wpbc data set for AUCtron and MSEtron algorithms.

5 Analysis of Results and Discussion

As shown in Figure 3, the average AUC monotonically increases with the accuracy. For $P = Q$, it coincides with the accuracy. In the other words, there does not seem to advantage in designing specific learning algorithms for maximizing the AUC, i.e., a classification algorithm minimizing mean squared error (mse) can optimize the AUC. However, Figure 4 demonstrates that, when prior probabilities of the classes are very different ($P \neq Q$) and the total number of errors k is high, classifiers with the same accuracy exhibit noticeably different AUC values (high variance). This motivates the design of algorithms that optimize the AUC rather than doing so indirectly by minimizing the mean squared error (mse). This assumption is reinforced by Table 1 which shows that, in general, the learning algorithm AUCtron developed to directly optimize the AUC achieved better results than MSEtron and AUCsplit algorithms. Figure 5 compares the ROC curves obtained from a test set for Breast Cancer wpbc data set. Notice that the AUCtron algorithm generates a better ROC curve than MSEtron based on mse.

Figure 2 illustrates that when estimated priors of the classes are different, the average AUC decreases as the false negative rate increases. Usually, the misclassification cost for minority class is much higher than misclassification cost for the majority class. That is the norm for most applications with small imbalanced data sets. The AUCtron algorithm considers estimated priors in the classification process of observations. The surface of separation obtained in the input space was set to minimize the number of false negatives errors and consequently to maximize the number of correct positive classifications. This statement is confirmed by results presented in Table 2. Observe that the true positive rates (sensitivity) for all data sets increased significantly when the algorithm AUCtron was utilized. The ranking quality improved perceptibly and a better balance between sensitivity and specificity was achieved.

6 Conclusion

It is known that representativeness in a training set is the most important feature to achieve classifiers with high generalization performance. However in most classification problems, representativeness is not only expensive but often a very difficult task. In general, data sets available are small, sparse, with missing values and with heavily imbalanced estimated prior probabilities.

Usually the representativeness problem in data sets is resolved using learning algorithms with techniques based on complexity control such as minimizing the magnitude of parameters, maximizing separation margins and use of regularization terms. Other techniques involve re-sampling or partition strategies. When applied to small data sets problems, these algorithms tend to smooth the response. One of the possible explanations is the asymptotic boundaries imposed by the reduced size of the training and validation data sets. In addition, these algorithms do not take into consideration the differences between the class distributions during the decision process.

Beforehand, the binary classifiers based on Bayes rule [7] will consider estimated priors in the decision criteria. The surface of separation obtained by a bayesian classifier bend in the direction of the minority or positive class. When priors are equal or are not considered in decision making the classifier becomes smoother.

Therefore when analyzing small imbalanced data sets we are faced with the problem of which is the best separation surface. The decision as to whether class distribution should be taken into consideration must be contemplated. Here we have demonstrated that AUC is a sensitive enough metric that when used in small data sets may lead to a better separation.

References

[1] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition, Vol.30 No.7 pgs. 1145-1159*, 1997.

[2] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. *In Advances in Neural Information Processing Systems No. 16. pgs. 313-320. Cambridge MA: MIT Press.*, 2004.

[3] J. P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

[4] T. Fawcett. Roc graphs: notes and pratical consideration for researchers. Technical report, HP Laboratories, 2004.

[5] C. Ferri, P. Flach, and J. Hernandez-Orallo. Learning decision trees using the area under roc curve. *Proceedings of the 19th International Conference on Machine Learning pgs. 139-146*, 2002.

[6] L. Graning, Y. Jin, and B. Sendhoff. Generalization improvement in multi-objective learning. *International Joint Conference on Neural Networks, Vancouver, BC, Canada, pgs. 4839-4846*, 2006.

[7] S. Haykin. *Neural Networks: Principles and Practice*. Bookman, 2001.

[8] A. Herschtal and B. Raskutti. Optimizing area under the roc curve using gradient descent. *Proceedings of the 21st International Conference on Machine Learning, Banff, Canada.*, 2004.

[9] A. Herschtal, B. Raskutti, and P. K. Campbell. Area under roc optimisation using a ramp approximation. *Proceedings of the 6th SIAM International Conference on Data Mining, April 20-22, 2006, Bethesda, MD, USA*, 2006.

[10] Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning, 46, pgs. 191-202.*, 2002.

[11] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, Stanford, California, 2 edition, 1984.

[12] H. B. Mann and D. R. Whitney. On a test wheter one of two random variables is stochastically larger than the other. *Annals of Math. Statistics, 18, pgs. 50 - 60.*, 1947.

[13] M. C. Monard and G. E. Batista. Learning with skewed class distributions. *Advances in Logic, Artificial Intelligence and Robotics, pgs. 173-179.*, 2002.

[14] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge based approach - a case study in intensive care monitoring. In *Proceedings of the 16th International Conference on Machine Learning, Bled, Slovenia, pgs. 268-277*, 1999.

[15] M. C. Mozer, R. Dodier, M. D. Colagrosso, C. Guerra-Salcedo, and R. Wolniewicz. Prodding the roc curve: constrained optimization of classifier performance. *In Neural Information Processing Systems 2001, Vancouver, British Columbia, Canada*, 2001.

[16] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th International Conference on Machine Learning, Madison, Wisconsin, USA*, 1998.

[17] A. Rakotomamonjy. Svms and area under roc curve. Technical report, PSI-INSA de Roun, 2004.

[18] M. S. Sanchez, M. C. Ortiz, L. A. Sarabia, and R. Llet. On pareto optimal fronts for deciding about sensitivity and especificity in class-modelling problems. *Analytica Chimica Acta 544 pgs. 236-245*, 2005.

[19] L. Yan, R. Doldier, M. C. Mozer, and R. Wolniewicz. Optimizing classifier performance via approximation to the wilcoxon-mann-witney statistic. *Proceedings of the 20th International Conference on Machine Learning pgs. 848-855, Menlo Park, CA*, 2003.