Smooth Receiver Operating Characteristics (*smROC*) Curves

William Klement[1], Peter Flach[2], Nathalie Japkowicz[1], and Stan Matwin[1,3]

[1] School of Electrical Engineering and Computer Science, University of Ottawa, Canada
{klement,stan,nat}@site.uottawa.ca
[2] Computer Science, Bristol University, BS8 1UB United Kingdom
Peter.Flach@bristol.ac.uk
[3] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. Supervised learning algorithms perform common tasks including classification, ranking, scoring, and probability estimation. We investigate how scoring information, often produced by these models, is utilized by an evaluation measure. The ROC curve represents a visualization of the ranking performance of classifiers. However, they ignore the scores which can be quite informative. While this ignored information is less precise than that given by probabilities, it is much more detailed than that conveyed by ranking. This paper presents a novel method to weight the ROC curve by these scores. We call it the Smooth ROC (*smROC*) curve, and we demonstrate how it can be used to visualize the performance of learning models. We report experimental results to show that the *smROC* is appropriate for measuring performance similarities and differences between learning models, and is more sensitive to performance characteristics than the standard ROC curve.

1 Introduction

Supervised learning algorithms perform common learning tasks including classification, ranking, scoring, and probability estimation. This paper investigates how scoring information, often produced by such algorithms, may be utilized by the performance evaluation measure. A scoring model estimates scores on the training data and assigns them to testing data to express class memberships, and sometimes, these may represent probabilities where Brier Score is used to assess their quality. However, when the scores are not probabilities, they are more than ranks; they aren't just ordinals but numbers expressed on some scale. The scale may be unbounded and may not be additive, eg. the score could be a likelihood ratio which is multiplicative. Many applications, particularly in medicine, employ scores that are highly meaningful for the users and are not probabilities, eg. ICU scoring systems. In these cases, the task is usually reduced to a ranking or a classification by ignoring the magnitudes of scores.

The Receiver Operating Characteristics (ROC) curves are commonly used to visualize the ranking performancs of classifiers. However, they ignore the

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part II, LNAI 6912, pp. 193–208, 2011.

scores which, we argue, are quite informative. For instance, the scores convey information as to how close two data points may be from one another within a given rank. While such information is less precise than that of probabilities, it is much more detailed than performance information conveyed by ranking.

To illustrate this concept, consider the problem of assessing similarities and differences among user preferences. For example, Anna and Jan are asked to make movie recommendations based on their preferences. They are both given the same list of n movies to which they assign a *positive*, or a *negative* recommendation along with a continuous score (between zero and one). The score indicates the degree to which they like or dislike the movie, with value 1 indicating the maximum *"liking"* and 0 the maximum *"disliking"*. The task is to examine recommendations and preferences made by Anna and by Jan in search for similarities and differences between their assessments of the movie collection. The consideration of both criteria, recommendations and preferences, makes this task considerably more complex. For instance, Anna positively recommends a movie with a score of 0.52 because although she does not like the movie per se, she finds this movie worthy of recommendation. However, Jan gives the same movie a negative recommendation but assigns to it a score of 0.6, because while he likes it, he does not find it worthy of recommendation. Clearly, if we only compare their recommendations, we may draw a conclusive disagreement. Similarly, their scores depict a disagreement in the opposite direction. The issue becomes: do they really disagree? Her lower score suggests agreement with Jan's decision of a *not-so-good* movie, but his high score indicates his inclination to a *not-so-bad* movie. Although their assessments may appear to disagree, there is, in fact, a substantial agreement among them. This agreement, or the lack there of, between Anna and Jan is expressed as a combination of both a binary decision and a continuous score, which is far from being probabilistic in nature.

To compare Anna's results to Jan's we would plot and compare two ROC curves (or two *smROC* curves for that matter). The standard ROC method only compares positive to negative recommendations. The AUC, in this case, depicts how they agree on separating the positive from the negative recommendations only because the magnitude of their preference scores are excluded from the analysis. The proposed method addresses this issue of assessing both decisions and scores using a single evaluation measure based on ROC analysis. The area under the proposed curve (the *smAUC*) will depict the separation of scores in the context of the binary decision. This problem reaches several domains including: search engines where the results of a query may (or may not) be relevant and are strongly (or weakly) related to a query, recommendation systems as illustrated in the above example, medical decision making where a physician is interested in the presence (or absence) of a condition along with its associated severity score, and finally, in bioinformatics where a genetic sample may be analyzed for up (or down) regulation of a particular gene at a high (or low) levels of gene expression.

The *smROC* is a novel method that extends the ROC curve to include the scores as smoothing weights added to its line segments. We, therefore, call it the Smooth ROC (*smROC*) curve. This proposed *smROC* method measures similarities and differences in how Anna and Jan make recommendation decisions and assign scores and can be used to compare scoring classifiers, but this is hardly the only application. The *smROC* method can also be used to measure similarities and differences among data points. Optimizing such an agreement, or the lack thereof, will offer substantially more informative views of various abilities of supervised machine learning methods.

The organization of this paper follows the presentation of Section 2 to motivate the research and to discuss related work, Section 3 describes how to construct *smROC* curves and shows calculations of *smAUC*, and Section 4 presents experimental results that demonstrate the superiority of *smROC* over the standard ROC in measuring performance similarities and differences among scoring classifiers. Finally, Section 5 concludes with a brief discussion of future work.

2 Motivation and Related Work

Common supervised learning tasks (Figure 1) convey diverse information of class memberships of data points. Classification is a categorization of points into classes with yes/no decisions. Binary classification is a special case of making decisions on two classes. Ordinal classification extends the multi-class settings and imposes an order on the classes. However, its outcome remains a classification. Ranking yields an order of data points, and intuitively, a good ranker places the positives towards the top and the negatives towards the bottom of an ordered list. Ranking can be depicted by a simple order or by assigning ranks to data points, the latter can also be viewed as scores. Scoring enables the learning model to convey its confidence in class memberships [7]. However, interpreting scores, in general, requires a considerable amount of information related to the definition of the underlying scoring function, which is usually difficult to obtain and is highly uncertain. Instead, some algorithms estimate class memberships probabilities to induce this information. Thus, a probability estimation task may be considered an informed case of a scoring task that involves modeling the posterior probability distribution of class memberships given the training data.

The ability to distinguish between data points depends on the granularity of predictions made by the model and is known as the "discriminancy" of a

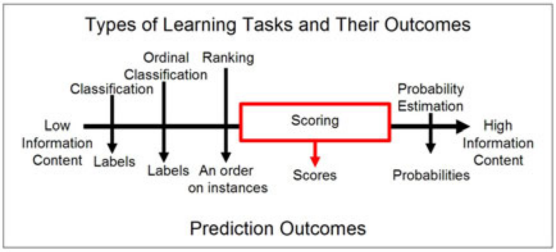
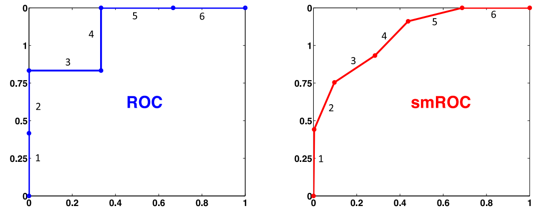


Fig. 1. Information content of common machine learning tasks

performance measure [11] where ties between predictions represent a difficult challenge. While classification distinguishes between positives and negatives, ranking separates instances based on how high (or low) they are placed in an ordered list. Yet, gaps between the ranked instances are determined by their availability rather than by the magnitudes of class memberships usually conveyed by the scores. Obviously, a prediction conveyed by a continuous-value score is less likely to produce ties than a binary decision. The latter is the least informative decision whose outcomes offer the least granularity. Probability estimation is the most informative task designed to describe the probability distribution learned from data. Its wealth of information offers a high degree of granularity due to exploiting powerful statistical principles. These methods, we find, are restrictive, inflexible, and parametric in nature. For instance, most machine learning methods assume the class distribution invariant between training and testing [8], and the quality of probability estimation is affected by this assumption [2]. Furthermore, the naive Bayesian learning method assumes the independence of attributes that describe the data. In practice, such assumptions are often violated [2,9,10], and in many cases, the raw scores provide poor estimates of true probabilities because some models are prone to estimating poor probabilities [5]. Or, it may be unnecessary to treat the scores as calibrated posterior probabilities. In these cases, the scores are merely used to construct the ROC curve for the purpose of performance analysis [5]. This use of the scores represents a reduction in information conveyed by the model (Figure 1) and we argue that it omits information relevant to the performance analysis by eliminating the magnitudes of the scores. The *smROC* curve evaluates such scores, particularly when their estimates are poor or violate assumptions required by probabilistic methods.

The scope of performance measures is usually restricted by information made available by the model. Scalar evaluation metrics such as accuracy, precision, recall, AUC, and MSE (or Brier score [3]) compute a summary of performance insensitive to characteristics of class memberships of individual points. The ROC method addresses this issue but ignores the magnitudes of the scores. Their exclusion can result in plotting identical ROC curves for multiple learning models irrespective of differences in their score assignments. Wu et al. [15] propose the sAUC method to incorporate score margins into the standard AUC and to detect the existence of fixed-size score gaps between positive and negative data points. Effectively, this measures how quickly the AUC, under the standard ROC curve, deteriorates when the positive scores are decreased. The *smAUC* method differs from sAUC in several ways. While *smAUC* weights the standard ROC curve by the absolute values of the scores, sAUC relies on score differences being greater than a fixed gap. In addition, the sAUC measures these score gaps only between positive and negative data points, which limits the assessment to the positives versus the negatives [15]. In contrast, the *smROC* relies on the score values themselves which enables the visualization of score differences and gaps of any size between any pairs of points regardless of their class, i.e., the *smAUC* can distinguish between data points in one class. Going back to our example, if Anna positively recommends two movies at scores 0.7 and 0.99, the sAUC will not



Anna's Assessment		
i	*Decision*	*Score*
1.	+	0.99
2.	+	0.70
3.	-	0.60
4.	+	0.51
5.	-	0.20
6.	-	0.00

Fig. 2. In the ROC space, it's difficult to distinguish between movies 1 and 2. Anna recommends both but she likes movie 1 almost 30% more than 2. The *smROC* plots the line segments with slopes proportional to the scores. Visually, movies 1 and 2 have different slopes in the *smROC* space. Similarly, scores of movies 5 and 6 result in different slopes. Anna likes movie 1 the most and 6 the least.

compare these two movies because they are both positives. The *smROC* curve represents individual movies by line segments whose slope are proportional to the corresponding score value (Figure 2). Therefore, the area under this curve will be affected by the exact margin of these two points. If we examine the definition of the *smAUC* (in the next section), it is clear that the *smAUC* compares all pairs of data points, and thus, these two movies will contribute their score magnitudes to the *smAUC*. This means that the *smAUC* reports a different kind of performance information than the sAUC. The *smAUC* metric depicts the performance of ranking weighted by the magnitude of confidence in predictions (conveyed by the class membership scores). The sAUC relates to the behavior of the AUC, under the ROC, when the positive scores are decreased.

A recent study [14] argues that soft variations of the AUC metric contribute little to the AUC for the purpose of model selection, because they favor models that generate large (rather than small) score margins. In this paper, we argue that such *soft* analysis of the ROC can be used to understand the behavior of scores. The *smROC* method not only produces a visualization of the scores (as opposed to their margins), it also detects similarities among score values. Such analyses are not limited to model selection, they can help compare scores assigned to data points as well. This aspect of performance cannot be measured by the standard AUC metric.

3 Constructing a Smooth ROC Curve

Let X be a data set that contains $n = n^+ + n^-$ (positive and negative) points, and let x_i be the i^{th} point whose label is $C_i \in \{+, -\}$ and S_i be the positive class membership score assigned to x_i. Finally, let m^+ and m^- be the average positive and negative scores respectively. Algorithm 1 begins at the origin and incrementally plots the ROC curve by examining points in X in a decreasing order of their S_i scores [5] (Figure 2). For a positive x_i, the curve climbs one step upwards, and for a negative x_i, the curve runs one step to the right. The vertical

Algorithm 1. Incrementally plotting of the ROC curve [5]

```
Input: $n = n^+ + n^-$ positive and negative points, $S_i \in [0,1]$ scores of $n$ points in a decreasing order, and $C_i \in \{+,-\}$ Labels.
for $i = 1$ to $n$ do {start at the origin (0,0)}
    if scores are tied between $y$ positives and $x$ negatives then
        simultaneously move up by $\frac{y}{n^+}$ and right by $\frac{x}{n^-}$
    else if $C_i = +$ then
        move up by $\frac{1}{n^+}$
    else if $C_i = -$ then
        move right by $\frac{1}{n^-}$
    end if
end for
```

and horizontal step sizes are $\frac{1}{n^+}$ and $\frac{1}{n^-}$ in the ROC space. To incorporate S_i scores into this curve, we modify Algorithm 1 to produce Algorithm 2. Our approach relies on altering the step sizes in the space proportionally to the score magnitudes. This modification preserves properties and characteristics of the ROC curve so that ROC analysis remain valid.

A scoring classifier estimates class membership scores and assigns them to data points. Classifications are obtained by imposing a threshold on these scores. While negative points whose scores lie above the threshold result in false positive errors, positives with scores below that produce false negative errors. We argue that such errors can be blamed not only on the choice of threshold value, but also on the *inappropriateness* of scores (see definition 1). More specifically, *inappropriate* scores result from assigning low scores to positives and/or from assigning high scores to negatives. The identification of high and low scores relies on a midpoint value in the range of the scores. When the scores are calibrated, this midpoint lies naturally at 0.5. Otherwise, it can be estimated. The latter is discussed in details later in this section.

Definition 1. *For a midpoint Mid, a class membership score S_i is appropriate for data point x_i if S_i is greater or equal to Mid when x_i is positive, or if S_i is less than Mid when x_i is negative. Otherwise, $(1 - S_i)$ is appropriate for x_i.*

To deal with score appropriateness, two separate confusion matrices are needed as shown in Figure 3. The separate consideration of classification accuracy for points whose scores are appropriate or inappropriate depicts the ability of the model to counter score inappropriateness. Algorithm 2 treats appropriate scores differently from inappropriate ones. For appropriate scores, it climbs vertically proportionally to the magnitude S_i while running horizontally proportionally to $(1 - S_i)$. (the normalization factors $\propto_h$ and $\propto_v$ are omitted at this point for simplicity). The first plot from the left in Figure 3 shows a positive data point with an appropriate score ($S_i > Mid$ assuming $Mid = 0.5$). This is depicted by a higher rise than run. When $S_i < Mid$, x_i is more likely to be negative (the second plot from the left in the same figure), and the score is deemed appropriate when the $C_i = -$ because S_i agrees with the C_i. While S_i indicates,

(Score Appropriateness)

	Scores	
Label	High	Low
$+$	yes	no
$-$	no	yes

(Accuracy of Appropriate Scores)

		Predicted	
Score	Label	Y	N
High	$+$	correct	incorrect
Low	$-$	incorrect	correct

(Accuracy of Inappropriate Scores)

		Predicted	
Score	Label	Y	N
High	$-$	incorrect	correct
Low	$+$	correct	incorrect

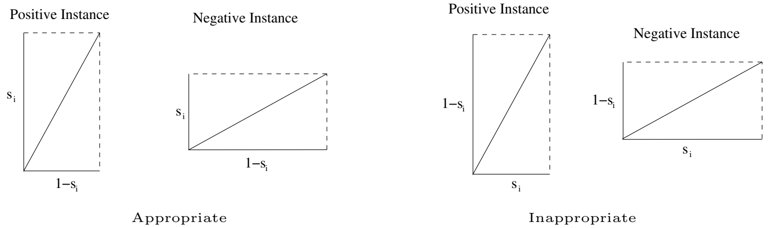


Fig. 3. The modified step size is based on score appropriateness

in the appropriate case, whether the x_i is positive or negative, $1 - S_i$ contradicts the label $C_i \in \{0, 1\}$. Thus, we plot a vertical climb proportional to S_i to show a gain in performance, and impose a horizontal penalty proportional to $1 - S_i$. Inappropriate scores must be treated differently. For instance, a positive point may be assigned a low score, and/or a negative point may be assigned a high score. These two situations are considered inappropriate, which we account for by reversing the score plotting strategy as shown in the two plots on the right of Figure 3. A positive point that is assigned $S_i < Mid$ conveys that x_i is more likely to be negative and contradicts the positive label. Similarly, a negative x_i that is assigned ($S_i > Mid$) suggests that x_i is more likely to be positive. In both cases, S_i contradicts the label and $1-S_i$ agrees with the label. Such contradictions merit adjustments. Therefore, Algorithm 2 plots a performance gain in the form of a vertical climb and proportional to $1 - S_i$, and it impose a penalty as a horizontal run proportional to S_i. Finally, the complete curve is assembled by connecting individual vectors resulting from the successive assessments of individual points similar to Algorithm 1.

The main difference between the two algorithms lies in the adjustment of the step-size. In the standard ROC curve (Algorithm 1) individual line segments either rise by $\frac{1}{n^+}$ or run by $\frac{1}{n^-}$ but not both. In the *smROC*, progress is made in both directions simultaneously using the scores (as described above) and using $\propto_v$ and $\propto_h$ as the vertical and horizontal normalization factors respectively. Algorithm 2 climbs vertically by $\frac{S_i}{\propto_v}$ while simultaneously running horizontally by $\frac{(1-S_i)}{\propto_h}$ when the scores are appropriate, and when they aren't, this algorithm climbs up by $\frac{(1-S_i)}{\propto_v}$ and simultaneously runs by $\frac{S_i}{\propto_h}$. Such a curve is illustrated in Figure 2. The remaining issues include the calculations of Mid, $\propto_v$, and $\propto_h$.

The midpoint Mid is necessary for the assessment of score appropriateness to separate high from low scores. When the scores $S_i \in [0, 1]$ are calibrated, it is natural to use $Mid = \frac{max(S)-min(S)}{2} = \frac{(1-0)}{2} = 0.5$. However, when the scores are uncalibrated, we estimate the midpoint between the average positive score

Algorithm 2. Incrementally Plotting the *smROC* curve

```
 1: Input: $n = n^+ + n^-$ positive and negative points, $S_i \in [0,1]$ scores of $n$ points in
    a decreasing order, $C_i \in \{+,-\}$ Labels, $Mid$ = Equation 1, $\propto_v$ = Equation 2, and
    $\propto_h$ = Equation 4.
 2: for $i = 1$ to $n$ do {start at the origin (0,0)}
 3:    if scores are tied (above $Mid$) between $y$ positives and $x$ negatives then
 4:       Move up $\frac{yS_i+x(1-S_i)}{\propto_v}$ and move right $\frac{y(1-S_i)+xS_i}{\propto_h}$
 5:    else if scores are tied (below $Mid$) between $y$ positives and $x$ negatives then
 6:       Move up $\frac{y(1-S_i)+xS_i}{\propto_v}$ and move right $\frac{yS_i+x(1-S_i)}{\propto_h}$
 7:    else if $(C_i = +)$AND$(S_i > Mid)$ then
 8:       Move up $\frac{S_i}{\propto_v}$ and move right $\frac{(1-S_i)}{\propto_h}$
 9:    else if $(C_i = -)$AND$(S_i < Mid)$ then
10:       Move up $\frac{S_i}{\propto_v}$ and move right $\frac{(1-S_i)}{\propto_h}$
11:    else if $(C_i = +)$AND$(S_i < Mid)$ then
12:       Move up $\frac{(1-S_i)}{\propto_v}$ and move right by $\frac{S_i}{\propto_h}$
13:    else if $(C_i = -)$AND$(S_i > Mid)$ then
14:       Move up $\frac{(1-S_i)}{\propto_v}$ and move right by $\frac{S_i}{\propto_h}$
15:    end if
16: end for
```

m^+ and the average negative score m^- for a given data set X. When the scores S_i are calibrated, and if the class distribution $c = \frac{n^+}{n^-} = 1$, then, it can be shown that $m^+ + \frac{m^-}{c} = 1$. This becomes obvious when $S_i \in \{0,1\}$ and $c = 1$ which gives $n^+m^+ + n^-m^- = n^+$. However, in the general case where scores S_i are not calibrated and $c \neq 1$, $m^+ + \frac{m^-}{c}$, reduces to $\frac{sum(S)}{n^+}$. Therefore, we set Mid to the midpoint as per Equation 1. This estimation of Mid is data specific and is based on scores produced by the model. Alternate methods of estimating Mid remain under investigation.

$$Mid = \frac{1}{2}(m^+ + \frac{m^-}{c}) = \frac{sum(S)}{2n^+} \tag{1}$$

To ensure that Algorithm 2 makes progress vertically and horizontally in the unit square of the space, the step-size must be normalized. These vertical and horizontal normalization factors are represented by $\propto_v$ and $\propto_h$ respectively. We now show their calculations using H^+, L^-, L^+, and H^- as the sets of: positives where $S_i \geq Mid$, negatives with $S_i < Mid$, negatives of $S_i \geq Mid$, and positives where $S_i < Mid$ respectively. Algorithm 2 plots the curve in steps proportional to S_i or $1 - S_i$ scores. We divide each step by the total score contributions in either direction to normalize them. To determine the vertical normalization factor $\propto_v$, we add up scores contributing to the upwards progress. Lines 7 and 9 of Algorithm 2 show this for points in H^+, and in L^- respectively, they contribute their S_i scores towards a vertical climb upwards. Lines 11 and 13 show that points in L^+, and in H^- respectively, contribute their $1 - S_i$ scores also in the vertical direction. Therefore, this total sum of scores contributions in the vertical direction is computed by Equation 2 as the sum of the positive contributions $\varTheta(x_i)$ for $i = 1 \dots n$.

$$\propto_v = \sum_{i=1}^{|H^+|} S_i + \sum_{i=1}^{|L^-|} S_i + \sum_{i=1}^{|L^+|}(1 - S_i) + \sum_{i=1}^{|H^-|}(1 - S_i) = \sum_{i=1}^{n} \varTheta(x_i) \tag{2}$$

$$\varTheta(x_i) = \begin{cases} s_i & \text{if } x_i \in \{H^+ \cup L^-\} \text{ (Appropriate Scores)} \\ 1 - s_i & \text{if } x_i \in \{H^- \cup L^+\} \text{ (Inappropriate Scores)} \end{cases} \tag{3}$$

The horizontal normalization factor $\propto_h$ is the sum of all scores contributions towards shifts (to the right) in the horizontal direction. Instances in H^+, and in L^-, respectively on Lines 7 and 9 of Algorithm 2, contribute their $(1 - S_i)$ scores along the horizontal direction. In addition, points in L^+, and in H^-, respectively on Lines 11 and 13 of Algorithm 2, contribute their S_i scores also to the horizontal progression. Therefore, the horizontal normalization factor $\propto_h$ is computed by Equation 4 as the sum of the negative contributions $1 - \varTheta(x_i)$ for $i = 1 \dots n$.

$$\propto_h = \sum_{i=1}^{|H^+|}(1 - S_i) + \sum_{i=1}^{|L^-|}(1 - S_i) + \sum_{i=1}^{|L^+|} S_i + \sum_{i=1}^{|H^-|} S_i = \sum_{i=1}^{n}(1 - \varTheta(x_i)) \tag{4}$$

The area under the *smROC* curve (*smAUC*) is calculated using Equation 5. The *smAUC* is based on accumulating the product of positive score contributions ($\varTheta(x_i)$) by the negative score contributions for all data points ranked lower than x_i. The latter is computed by Equation 6. A special case occurs when x_i is compared to itself, only one half of the product contributes toward the area under the curve in the second case of Equation 6.

$$smAUC = \frac{1}{\propto_v \propto_h} \sum_{i=1}^{n} \sum_{j=1}^{n} \varTheta(x_i) \varPsi(x_i, x_j) \tag{5}$$

where:

$$\varPsi(x_i, x_j) = \begin{cases} 1 - \varTheta(x_i) & \text{for } (S_i > S_j) \text{ and } (i \neq j) \\ \frac{1}{2}(1 - \varTheta(x_i)) & \text{for } i = j \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

The *smAUC* represents the separation between the total positive contribution of scores $\varTheta(x_i)$ and the total negative contribution of scores $1 - \varTheta(x_i)$ for all instances $i = 1 \dots n$ in their ranking order $(S_i > S_j)$. This suggests that *smAUC* favors scores that result in higher separation of classes weighted by the magnitudes of the scores. Finally, it can be shown that when the scores S_i are zeros and ones, the *smROC* and the *smAUC* will reduce to the standard ROC and the standard AUC respectively. This discussion is omitted due to space limitations.

Table 1. UCI binary classification data [1]

Abbr.	Data Set Name	n	$\|+\|$	$\|-\|$	Features	%+
prom	promoters	106	53	53	57	50
echo	echocardiogram	132	43	88	7	33
hepa	hepatitis	155	32	123	19	21
prks	parkinsons	195	147	48	22	75
hart	statlog heart	270	120	150	13	44
hrth	heart disease hungarian	294	188	106	13	64
hors	horse-colic reduced	296	188	108	21	64
habr	haberman	306	81	225	3	26
iono	ionosphere	351	225	126	34	64
vots	house-votes-84	435	168	267	16	39
jcrx	japanese crx	690	307	383	15	44
aust	statlog australian	690	307	383	14	44
wisc	breast cancer wisc	699	241	458	9	34
blod	blood transfusions	748	178	570	4	24
diab	pima-indians-diabetes	768	268	500	8	35
mamo	mammographic masses	945	434	511	5	46
tic	tic-tac-toe	958	626	332	9	65
ger	statlog german	1000	700	300	20	70
oz8h	ozone eighthr	2534	160	2374	72	6
oz1h	ozone onehr	2536	73	2463	72	3
chss	chess kr-vs-kp	3196	1669	1527	36	52
ads	internet ad	3279	459	2820	1558	14
spam	spambase	4601	1813	2788	57	39
mush	mushroom	8124	3916	4208	23	48
mgic	magic04	19020	12332	6688	10	65
adlt	census adult	32562	7841	24720	14	24

4 Experiments

The objective of this experiment is to illustrate the ability of the *smROC* curve to measure performance similarities and differences of scoring classifiers, and to demonstrate its superiority over the standard ROC curve. Therefore, we first construct learning models that are expected to produce similar performance, and we assess their performance in the *smROC* space and in the standard ROC. We wish to show that the *smROC* methods captures their performance similarities with higher performance sensitivity than the standard ROC. Then second, we generate performance data for two learning methods, Naive Bayes (NB) and Probability Estimating Trees (PET – unpruned decision trees with Laplace correction [12]), which are known to produce different scores when applied to the same data. Again, we compare their performance analysis in the *smROC* space and in the standard ROC space to demonstrate that the *smROC* method captures score differences better than the standard ROC.

To simulate score similarities, we rely on the consistency of the learning method by fixing the learning algorithm, as well as, the training/testing data

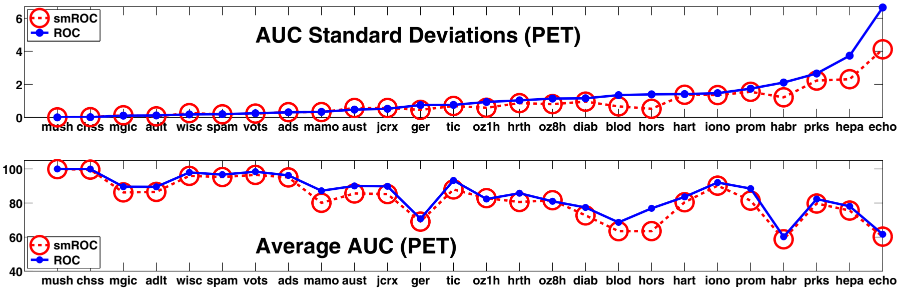


Fig. 4. Similar PET Models from ten runs of 10-fold cross-validation

distribution. The idea is to construct, more-or-less, similar classifiers from randomized versions of the same data. For instance, a collection of ten Naive Bayes classifiers should produce similar class membership scores in multiple runs of 10-fold cross-validation applied to the same data. In this case, performance variations occur due to the random splitting of data into 10 folds. The same argument applies to building ten PETs in the same way. We use the ROC and the *smROC* analysis to show that classifiers in the same group produce similar scores. As for differences, we rely on the same two methods, PET and Naive Bayes, to produce significantly different scores from each other as evidence suggests [12,13,6,16]. Therefore, to measure performance differences, we compare the pairwise performance of PET and Naive Bayes over the multiple runs of 10-fold cross-validation, they should produce different performance. We then compare how well the standard ROC curve and the *smROC* curve capture these differences. An issue we need to consider is the construction the pairs of models which should be derived from identical data sets in each run. This accomplished by controlling the seeds when we randomly split the data into folds.

For the purpose of this experiment, we use benchmark data sets listed in Table 1, which are obtained from the machine learning repository [1]. For evaluation, we construct the *smROC* and the standard ROC curves (two curves) for the two learning methods resulting from each of the ten runs on every data set (twenty-six sets). This generates over one thousand curves for us to analyze. To make this analysis manageable, we summarize the curves by their respective area under the curve. Thus, we calculate the *smAUC* and AUC respectively. And for ease of presentation, we plot the average and standard deviation of AUC and of *smAUC* for each data set. When assessing similarities. we plot these values for each model separately, but for differences, we plot these values (average and standard deviation) for their observed pairwise difference.

4.1 Performance Similarities

Figure 4 shows the standard deviation and the average area under the curves (*smROC* and ROC) generated by the PET method over ten runs of 10-fold cross-validation. We observe that the dashed curve is consistently below (or sometimes

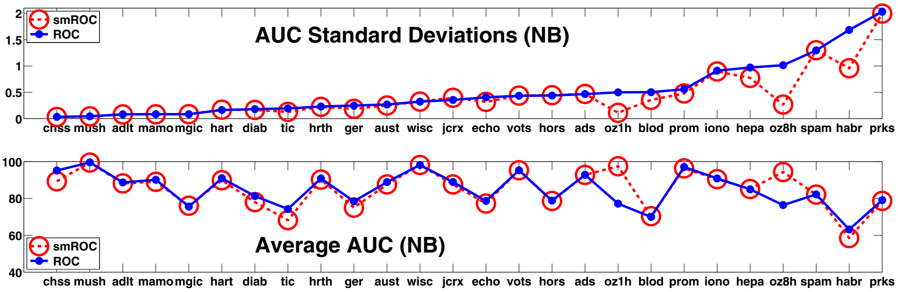


Fig. 5. Similar Naive Bayes models from ten runs of 10-fold cross-validation

the same) as the solid curve in both plots. For data sets that produce higher standard deviation (in the top plot), the standard deviation of the area under the *smROC* curve is lower than that under the standard ROC. When this standard deviation is low, both ROC and *smROC* curves show the same low standard deviation. This suggests that when variations occur among similar models, the *smROC* captures more similarity than the standard ROC. The higher standard deviation of area under the ROC curve can be attributed to the exclusion of score magnitudes, which results in an over/under estimate of separation between ranks (in the ROC space, the area under the curve is estimated by a discrete indicator function, whereas, the *smAUC* follows the separation between score values). This observation is supported by the bottom plot (in the same figure) which shows the average area under the curve. The average area under the *smROC* curve is generally lower than that under the standard ROC in the figure. This makes sense in the context of the *smROC* being a kind of smoothing of the ROC curve (weighted by score magnitudes). The PET learning method uses unpruned decision trees with Laplace correction for smoothing the scores. Thus, we expect its associated *smROC* curves to be smoother than the corresponding ROC curves. In a sense, the consistently lower *smAUC* is saying that PET scores are consistently smoother than what the standard ROC shows.

Similar observations can be made in Figure 5. For data sets with low standard deviation, the *smROC* and the standard ROC produce similar standard deviations of the area under their respective curves for similar Naive Bayes models. As this standard deviation increases, the *smAUC* produces lower standard deviation (in the top plot of the figure). The average area under the *smROC* curve is the same, or less, than that under the standard ROC curve for most data sets. One exception is the `oz1h` and `oz8h` data sets (they are both obtained form the same *Ozone* domain [1]). For these sets, the average *smAUC* is substantially higher than the standard AUC. Furthermore, the standard deviation of the area under the *smROC* curve for these two sets is also substantially lower than that under the standard ROC curve (see the top plot of the same figure, the corresponding open circles are much lower than the solid curve). This suggests that the ranking order of data points fails to correctly separate the two classes.

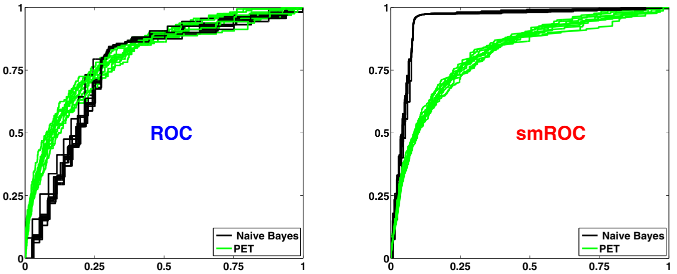


Fig. 6. Naive Bayes and PET models constructed from `oz8h` data

However, the combination of score magnitudes along with the ranking performance produce better class separation. Perhaps, the standard ROC curve is faced with classification errors due to the scores being just above, or just below, the classification threshold. Consequently, the scores appear in disagreement with class labels (this is similar to the movie recommendation example presented in the introduction). Such errors are compensated for when the magnitudes of the scores are used as weights by the *smROC* analysis.

4.2 Detecting Differences

In this section, we compare the performance of two learning methods, Naive Bayes and PET, which are known to be different in how they produce class membership scores. We measure their performance on our data sets in both the *smROC* and the standard ROC spaces. Figure 6 presents the ROC and the *smROC* curves for the two learning methods applied to the `oz8h` data. Comparing the ROC curves leads to the conclusion that both methods accomplish comparable performance because Naive Bayes (the dark curves) and PET (the light curves) are close to each other. In addition, if we compare the average area under their curves respectively, we measure a small difference between them, well, smaller difference than that observed in the *smROC* space. The reason the two plots differ is due to including score magnitudes in the construction of the *smROC* curves. The left plot of Figure 6 shows that the ROC curve is insensitive to differences in scores. All we can see is that the PET curves are visually smoother than those of the Naive Bayes'. However, if we use the AUC metric, this difference becomes far less obvious. Alternatively, the *smROC* curves show that scores produced by Naive Bayes are high for positive examples (the steep vertical rise) and they are low for negative instances (the consistent horizontal run in the top right). The strong change of direction along the *smROC* curves associated with naive Bayes indicates a substantial gap in the scores. The *smROC* curves associated with the PET models (the light curves) appear smoother with a comparable area under the curve in both spaces. This illustrates how the *smROC* curve depicts ranking information but adds score magnitudes. These magnitudes have a little smoothing effect on the standard ROC in the case of PETs.

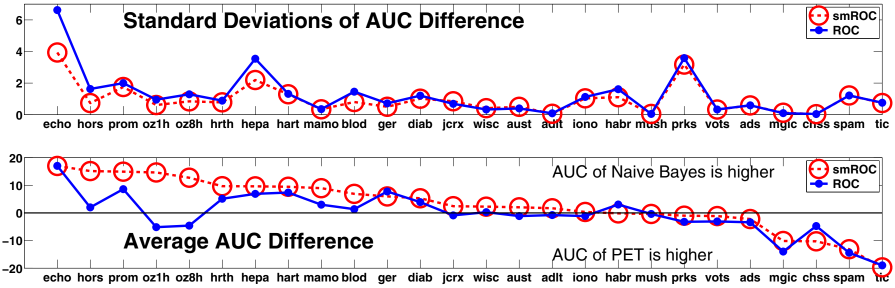


Fig. 7. AUC difference for Naive Bayes and PET models

However, in the case of Naive Bayes, the score magnitudes amplify its classification ability but expose the *definite* nature of its scores (definite, or appropriate, because positive examples have high scores and negatives have low scores). This highlights a significant difference between the two learning methods that the standard ROC shows little to no sensitivity to. In fact, if we rely on the area under the curve metric to understand such differences, the $smAUC$ is more favorable due to its sensitivity to the scores. The standard AUC fails to measure differences in scores which presents an interesting argument for studies such as [14]. Vanderlooy and Hüllermeier [14] suggest that soft variations of the AUC offer little to no improvement when it comes to model selection. This may well be the case when the primary interest is classification or ranking for that matter. However, when it comes to examining a scoring method where the scores are of interest, our results show that the $smAUC$, a soft variation of the standard AUC, is able to measure scoring differences that are buried in the ROC space due to the exclusion of their magnitudes. This paper argues and shows that these differences are important. For instance, the scoring behavior of Naive Bayes and decision trees have been the focus of several studies [12,13,6,16]. Our $smROC$ curve depicts these differences with ease, and moreover, the $smAUC$ represents a metric sensitive to such characteristics. If the $smAUC$ metric enables these studies to investigate the scoring behavior of learning methods with ease.

Lets consider the average and standard deviations of the difference in the AUC and in the $smAUC$, respectively, for our two learning methods. Figure 7 shows the average pairwise difference of the area under the curve between Naive Bayes and PET resulting from this experiment. We now describe how we compute this difference; in each of the ten runs of 10-fold cross-validation, we construct a Naive Bayes model and a PET model from the same sample of data (randomly split into folds using the same seed). Then, we construct their ROC and $smROC$ curves, and we compute their respective areas under the curves and record their difference. Namely, we subtract the area covered by the curve associated with PET from the area covered by the curve associated with Naive Bayes. At the end, we compute the average and the standard deviation of these recorded differences in their respective spaces. Finally, we plot these results for all data sets (see Figure 7). It is clear that the average AUC difference in both spaces agree when this difference is in favor of the PET models (the two curves agree when they are close or below the solid line of 0 difference in the right half of the bottom plot of the Figure 7). In addition, if we examine the standard deviations for the same data sets in the top plot, we see that both AUC and *smAUC* produce similar standard deviations of this difference. This suggests that PET models that perform better than Naive Bayes produce solid, and consistent, ranking and scoring performance observed in both spaces. However, when the balance tips in favor of the Naive Bayes models, the scores become more appropriate (as we define them). The *smAUC* measures a substantially higher difference than the standard AUC. This is illustrated by the average difference in the ROC space being below that of the *smROC* curve, and sometimes, the former crosses below the zero line (see the bottom plot of the same figure). Furthermore, the observed standard deviation of this AUC difference is higher for ROC curves than for *smROC* curves (see the corresponding standard deviations in the top plot of the same figure). This suggests that the ROC curve struggles to measure this difference between the two models. Thus, the use of the standard AUC fails to detect these differences because they are excluded. The *smAUC*, however, measures these differences clearly and with lower standard deviations. Since it favors appropriate scores, these results suggest that Naive Bayes produces scores useful for classification but they are far from being smooth.

5 Conclusions

This paper presents a novel evaluation measure, the *smROC* curve, to incorporate class membership scores into the ROC curve. Based on a categorization of common machine learning tasks, which include classification, ranking, scoring and probability estimation, we argue that class membership scores convey valuable information relevant to the performance. Ignoring them, as the standard ROC does, results in a reduction of information expressed by the model.

Our results show that the *smROC* is effective in measuring performance similarities and differences among learning models. The *smROC* is sensitive to performance characteristics related to how a learning model assigns class membership scores to data points. The results demonstrate that the *smROC* curve measures the performance with less variations than the standard ROC curve, and it detects performance differences more consistently than the standard ROC. These results are statistically significant using the paired t-test. However, significance results are omitted due to space limits. Therefore, the *smROC* method enhances the ROC method, and captures specialized performance information with a higher granularity while remaining more abstract than dealing with probability estimates. Future research directions include investigating alternate methods of computing the mid point used to assess score appropriateness, analyzing the effect of varying this midpoint for all values between zero and one, and exploring other advantages of the *smROC* curve. These include its sensitivity to changes in the domain, i.e., it can be shown that the *smROC* method is sensitive to changes in the data distribution. It can be argued that measuring these differences between training and testing represents a significant accomplishment.

Acknowledgement. This research has been supported by the Natural Sciences and Engineering Research Council of Canada and the Ontario Centres of Excellence.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Bennett, P.N.: Using Asymmetric Distributions to Improve Text Classifier Probability Estimates. In: Proceedings of ACM SIGIR 2003, pp. 111–118 (2003)
3. Brier, G.: Verification of Forecasts Expressed in Terms of Probabilities. Monthly Weather Review 78, 1–3 (1950)
4. DeGroot, M., Fienberg, S.: The Comparison and Evalution of Forecasters. The statistician 32, 12–22 (1983)
5. Fawcett, T., Niculescu-Mizil, A.: PAV and the ROC Convex Hull. Machine Learning 68(1), 97–106 (2007)
6. Ferri, C., Flach, P., Hernandez-Orallo, J.: Improving the AUC of Probabilistic Estimation Trees. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 121–132. Springer, Heidelberg (2003)
7. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. Technical Report HPL-2003-4, HP Labs (2003)
8. Forman, G.: Counting Positives Accurately Despite Inaccurate Classification. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 564–575. Springer, Heidelberg (2005)
9. Greiner, R., Su, X., Shen, B., Zhou, W.: Structural Extension to Logistic Regression: Discriminative Parameter Learning of Belief Net Classifiers. Machine Learning 59(3), 213–235 (2005)
10. Grossman, D., Domingos, P.: Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. In: Proceedings of ICML 2004, pp. 361–368 (2004)
11. Ling, C.X., Huang, J., Zhang, H.: AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In: Proceedings of Canadian AI 2003, pp. 329–341 (2003)
12. Margineantu, D.D., Dietterich, T.G.: Improved Class Probability Estimates from Decision Tree Models. Nonlinear Estimation and Classification 171, 169–184 (2002)
13. Provost, F., Domingos, P.: Tree Induction for Probability-Based Ranking. Machine Learning 52, 199–215 (2003)
14. Vanderlooy, S., Hullermeier, E.: A Critical Analysis of Variants of the AUC. Machine Learning 72(3), 247–262 (2008)
15. Wu, S., Flach, P.A., Ferri, C.: An Improved Model Selection Heuristic for AUC. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 478–489. Springer, Heidelberg (2007)
16. Zhang, H., Su, J.: Learning Probability Decision Trees for AUC. Pattern Recognition Letters 27, 892–899 (2006)