ROC Analysis of Classifiers in Machine Learning: A Survey

Technical report MM-1/2011

Matjaž Majnik, Zoran Bosnić

University of Ljubljana, Faculty of Computer and Information Science, Tržaška cesta 25, Ljubljana, Slovenia

Abstract

The use of ROC (receiver operating characteristics) analysis as a tool to evaluate performance of classification models in machine learning has been increasing in the last decade. Among the most notable advances in this area is that the two-class ROC analysis has been extended to multi-class problems and that the ROC analysis in cost-sensitive learning has been considered, as well. Methods exist which take instance-varying costs into account. The purpose of our paper is to present a survey of this field with the aim to gather important achievements in one place. In the paper, we present application areas of the ROC analysis in machine learning, describe its problems and challenges and provide a summarized list of alternative approaches to ROC analysis. In addition to presented theory, we also provide a couple of examples intended to illustrate the described approaches.

Keywords: ROC analysis, ROC, performance, machine learning, classification
PACS: 07.05.Kf, 07.05.Mh, 07.05.Tp
2000 MSC: 68T01, 68T05, 68W40, 68T99

1. Introduction

Receiver operating characteristics (ROC) analysis is a method for evaluating, comparing and selecting classifiers on the basis of their performance. First known application of ROC analysis took place during Second World War when it was employed for the processing of radar signals. Later its use began in the signal detection theory for illustrating the compromise between hit rates and false alarm rates of classifiers [32, 17]. Other fields to which ROC analysis has

Email address: matjaz.majnik@gmail.com, zoran.bosnic@fri.uni-lj.si (Matjaž Majnik, Zoran Bosnić)

been introduced include psychophysics [32], medicine (various medical imaging techniques for diagnostic purposes, including computed tomography, mammography, chest x-rays [51] and magnetic resonance imaging [43], diverse methods in epidemiology [36]) and social sciences. An extensive list of ROC analysis sources to support decision making in medicine has been published, consisting of over 350 entries divided into several sections [62].

For over a decade, ROC analysis is gaining popularity more intensely also in the field of machine learning. First applications date back to late 1980's when ROC curves were demonstrated to be applicable to the rating of algorithms [48]. In the present, their use as a metric for assessing machine learning algorithms has become almost indispensable. Introduction to the use of ROC analysis in research (with the stress on machine learning) can be found in [20].

Research in the field of ROC analysis has been developing lively in many directions as well as the ROC curves are used in various applications: the two-class ROC methodology has been generalized to handle the multi-class problems, extensions of the basic ROC curves and the AUC metric were proposed, advanced alternatives to the basic ROC graphs appeared etc. Since the topic has been studied by many researchers from the distinct points of view, some questions arose which have to be answered in the future. The purpose of this paper is to present a survey on these issues in the context of machine learning which could guide a reader to the literature with more exhaustive explanations.

In this paper, the terms "ROC curve", "ROC graph" and "ROC analysis" are sometimes used interchangeably, though the "ROC analysis" is the most general, depicting the whole field of study, while the "ROC curve" in fact denotes a curve on the "ROC graph" (chart). The paper is structured as follows. In Section 2, the basic features of ROC analysis are discussed. Section 4 summarizes various applications of the analysis for distinct purposes. Further, Section 5 presents some problems and beneficial improvements of the basic ROC methodology, Section 6 sheds light on alternative visualizations to the ROC graphs, and Section 7 concludes the paper.

2. Definition of the ROC space

ROC analysis in its original two-class form is used to deal with the two-class classification problems. A set of instances with known classes is given, each of them belonging either to the positive or the negative class. The terms *positive* and *negative* originally stem out from the early medical applications, where the examples describing the patients with some present observed medical phenomenon (e.g. an illness) were denoted as *positive*, and the rest of the patients as *negative*. After the learning phase, a classifier should be able to predict a class value of some new, unseen instances.

Since the predicted classes of instances are not necessarily same as true classes, a matrix is used to keep a record of the prediction errors. This matrix is called a *contingency table* or a *confusion matrix* (since it represents the confusion between classes) and is shown in Fig. 1. There are four possible outputs for the classification of each instance, as follows. If the instance is positive

and is classified as such then we denote it as *true positive* (TP). If a classifier made a mistake and classified the instance as negative, we call it *false negative* (FN). Similarly, if the instance is negative and was also classified as negative we denote it as *true negative* (TN) and in the case of a misclassification we call it *false positive* (FP). The correct classifier decisions thus lie on the diagonal of the contingency table, while other table elements represent the number of misclassifications. A contingency table is a source for the further calculation of knowledge evaluation measures, including the *true positive rate (TPR)* and the *false positive rate (FPR)*, which are defined in the Fig. 2. In some fields of study, TPR is also called *sensitivity* or *recall* as well as the term *specificity* denotes the *true negative rate (TNR)*.

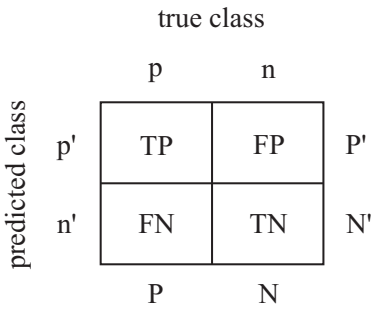


Figure 1: Structure of the contingency table for binary classification problems

$$\text{sensitivity} = \text{true positive rate} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{false positive rate} = \frac{\text{FP}}{\text{N}}$$

$$\text{specificity} = \text{true negative rate} = \frac{\text{TN}}{\text{N}} = \frac{\text{TN}}{\text{FP} + \text{TN}} = 1 - \text{FPR}$$

$$\text{false negative rate} = \frac{\text{FN}}{\text{P}}$$

Figure 2: Performance measures relevant for the ROC analysis

A ROC graph for the original two-class problems is defined as the two dimensional plot by representing FPR (=1-specificity) on its x-axis against TPR (sensitivity) on the y-axis. A performance of a particular classifier, represented by its sensitivity and specificity, is denoted as a single point on the ROC graph. There are some basic characteristic points on the ROC graph. The point with coordinates (0,0) ($TPR = 0$, $FPR = 0$) represents a classifier which never predicts a positive class. While such a classifier would never misclassify a negative instance as positive, it is not the best choice, since it would never make a single correct classification of a positive instance neither. Its relative in (1,1) represents the opposite situation ($TPR = 1$, $FPR = 1$) as it classifies all instances

as positive, thus producing also a high number of false positives. Classifiers in $(0, 0)$ and $(1, 1)$ are also called the *default classifiers*. In $(0, 1)$ the perfect classifier is located ($TPR = 1$, $FPR = 0$). While it is not realistic to expect such a performance of any classifier on a real-world problem it represents a goal at which the induction of classifiers should aim. Classifiers which are located on the diagonal line have the same performance as the random guessing. It is said they have no information about the problem. Useful classifiers are located above the diagonal. Those under it are performing worse than random guessing. Nevertheless, they can be made useful very easily by inverting their predictions. Such classifiers are said to have useful information but are employing it in a wrong way [27].

Very useful feature of ROC curves is that they remain unchanged when altering class distribution. ROC curve is based on TPR and FPR values. Class distribution is the proportion of positive instances (left column in Fig. 1) to negative instances (right column in Fig. 1). The fact that TPR and FPR are each calculated from values of one column makes ROC curves independent of class distribution.

There are, roughly speaking, two kinds of classifiers: discrete and probabilistic (scoring). For every instance, a discrete classifier predicts a class, whereas a probabilistic one issues a value which is class probability (in the strict sense) or class score (not calibrated). If a probabilistic classifier, for example, returns two scores where the score for the first class is greater than the second, this indicates that the first class has higher probability as well. A disadvantage, however, is that scores from different classifiers cannot be compared to each other in contrast to predicted probabilities which have a common interpretation. Nevertheless, the important property of ROC curves is that they measure the capability of classifiers to output good scores [20]. Analyzed classifiers thus do not have to produce exact probabilities, all they have to do is discriminate positive instances from negative ones.

To construct an ROC curve of a probabilistic classifier we first have to sort instances according to their scores. Then we start with the instance having the highest score and sweep over other instances in a decreasing manner. Drawing process starts in $(0, 0)$. In every step we check whether the instance's true class is positive or negative - if positive, we move one unit up, if negative, one unit to the right. The process is described also as applying different *thresholds* on scores, and terminates when the upper right corner in $(1, 1)$ is reached. Horizontal and vertical unit sizes are inversely proportional to the number of negative and positive instances in the dataset, respectively . All points, obtained by the process, are finally connected to form an ROC curve. Thus, by varying the threshold on scores one may obtain different (FPR,TPR) points, what may be seen as drawing an ROC curve.

Table 1 and Fig. 3 show an illustrative example of the ROC drawing process. The Tab. 1 contains 20 examples along with their predicted classifier score and the true class. The dataset is balanced, containing 10 positive and 10 negative examples which makes the construction of our sample ROC curve more clear, since horizontal and vertical unit sizes are both equal 0.1. In our example,

applying threshold between examples 4 and 5 yields a situation with 4 true positive classifications and 0 false positive, which we denote as a point $(0, 0.4)$. After moving the threshold between examples 5 and 6, we obtain the point $(0.1, 0.4)$ since the example 5 is misclassified as negative. We proceed in a similar way until the point $(1, 1)$ is reached, obtaining the ROC curve shown in Fig. 3. For simplicity, we presumed equal misclassification costs with their value being 1. In the case of unbalanced class distribution or unequal misclassification costs ROC space features change, e.g. expected costs of classifiers on the diagonal are different.

Table 1: An example of a class distribution

example no.	classifier score	true class	example no.	classifier score	true class
1	0.95	p	11	0.57	n
2	0.92	p	12	0.55	p
3	0.90	p	13	0.54	p
4	0.86	p	14	0.52	n
5	0.80	n	15	0.50	n
6	0.73	p	16	0.48	n
7	0.71	p	17	0.47	p
8	0.64	n	18	0.44	n
9	0.61	p	19	0.38	n
10	0.60	n	20	0.35	n

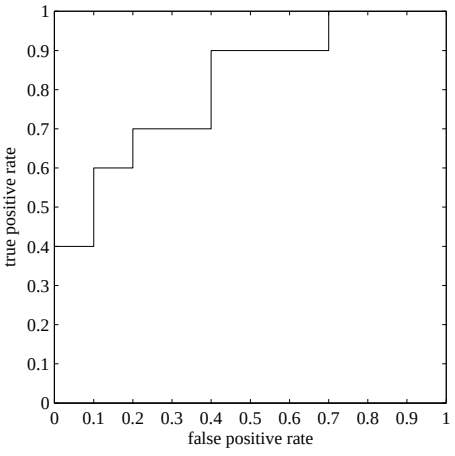


Figure 3: ROC curve for a sample class distribution

Since a discrete classifier is represented by only one point on an ROC graph,

one may alternatively construct a very approximate ROC curve by connecting this point with the points denoting both default classifiers. Another, better option is to analyze the classifier's decision process and adapt it to issue scores in addition to class predictions. When the scores are obtained the same procedure of constructing an ROC curve is employed as in the case of probabilistic classifiers.

A ROC graph example with four classifiers is given in Fig. 4. Classifier A is by far better than other three classifiers. ROC curves of classifiers B and C cross - each of these two is superior to the other for some deployment contexts. Classifier D is of no use as its performance is no better than chance.

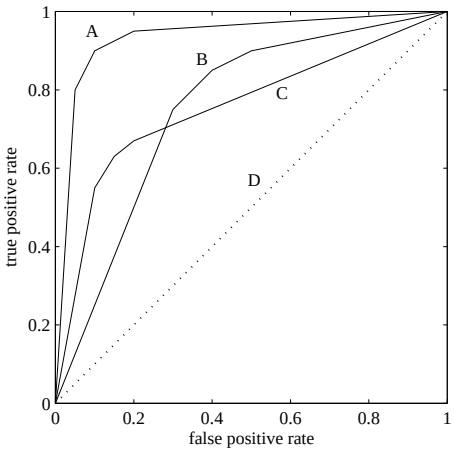


Figure 4: ROC graph with four classifiers

Comparing a pair of classifiers through their ROC curves can be non-trivial task when no classifier is dominating the other. To this end, ROC analysis defines another measure of classification model performance: *area under the ROC curve (AUC)* [34]. Statistical meaning of the AUC is the following: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [20]. This statistical property is often referred to as the probabilistic form of AUC measure. In [7] the use of AUC as a performance measure for machine learning algorithms is investigated. AUC and overall accuracy metrics are compared. It is shown that AUC has some convenient features: a standard error decreasing when both AUC and the number of test samples increases; it is independent of a decision threshold; it is invariant to prior class probabilities; and it indicates to what degree the negative and positive classes are separated.

AUC is related to many well-known measures. It is equivalent to the Wilcoxon statistic [34] and to the Mann-Whitney statistic. Moreover, it is related to the Gini index [8]. The value of AUC is in the range from 0 to 1. Since any useful

classification model should lie above the diagonal of an ROC graph, AUC of such a model exceeds the value 0.5.

3. Engaging classifiers into the ROC construction process

Below, the procedure of generating ROC curves for some typical classifiers is described briefly. When planning to draw a ROC curve, the main issue with the classifiers is how to obtain the scores for their test instances. As already mentioned, some classifiers yield such score values in their original setting, while the others output discrete predictions which should be manipulated to output scores. It is presumed that each classifier is already trained on instances of the training set. The statistics about class distribution of training instances may then be used in calculations of a score output for individual test instances.

After the scores are obtained, we may generate a ROC curve by applying one of the following two equivalent processes: (1) we may draw a ROC curve as described in the previous example, or (2) alternatively, alter the threshold on some parameter of a classifier explicitly, in a systematic manner. For every threshold value, the contingency matrix is recalculated, TPR and FPR is recomputed and the corresponding point added to the ROC space. The set of points is finally connected to form an ROC curve. When the threshold is applied to the predicted score for the positive class, it should be noted, that by increasing the threshold, values of TPR and FPR decrease (each at its own rate), as less instances are classified as positive. Consequently, we move over an ROC curve in the south-west direction.

In the following, we briefly summarize how the classifier scores are obtained from the most commonly used classification models:

- naive Bayes (NB) by default outputs a score in the interval [0.00, 1.00] for every test instance. The threshold may then be employed on these probability estimates,
- decision Tree (DT) in its basic form only returns a class label for each test instance. However, the proportion of training instances in the leaf (i.e. class distribution) to which a test instance has fallen may be used as a score. DTs that estimate class probabilities are also known as probability estimation trees (PETs), and are further discussed in [44], a work on probability estimates in decision trees. The threshold is set on the proportion of positive instances in the leaf,
- artificial Neural Network (ANN) yields a score for every test instance. Common technique is to set the threshold on the output node (in the interval [0.00, 1.00]). Yet another strategy is to scale the bias input weights of nodes on the first hidden layer of the network, as presented in [59]. In this work, methods for ROC curves construction for ANN classifiers are analyzed and it is claimed that ROC curves generated by the latter method have higher value of AUC and, moreover, a better distribution of operating points,

- in the case of k-Nearest Neighbors (KNN), a score for a test instance may be associated to the proportion of its neighbors belonging to the positive class, i.e. class distribution. The threshold on the number of neighbors needed to classify a test instance to a positive class (i.e. the number of votes) is varied from 1 to k, where k is the fixed number of neighbors taken in consideration. In this manner, the ROC curve for a given k is constructed. If the most appropriate value of k is unknown, a useful optimization technique is to repeat the process for different values of k. As a result, a set of ROC curves is acquired. A curve with the highest AUC may be chosen as the best option, and the value of the corresponding k as the optimal number of the considered neighbors. In [58] k is varied from 1 to 200,

- Support Vector Machine (SVM) outputs scores by default. The threshold is simply set on the decision function [53].

While instance statistics are handy basis for score acquisition, they are not the only option. Discrete classifiers may be transformed to scoring ones also by classifier aggregation or a combination of scoring and voting [20]. A review of methods for generating ROC curves for various classifiers may also be found in [58].

4. Applications

The use of ROC analysis in machine learning is heterogeneous and turns out to be very convenient when class distributions and misclassification costs are unknown (at training time). It is applied to model evaluation, comparison, selection, presentation, construction and combination. Thus, it is used as a post-processing technique and as a method that is actively taking part in the model construction to improve a given model. In the following we describe the most common application areas of ROC analysis in machine learning. Activities connected with the fulfillment of ROC-related tasks are not strictly independent and some level of overlapping is present. For instance, *model improvement* (see Sect. 5) may be recognized as a task related to model construction as well as model combination since the goal in both cases is to gain a model with better performance.

4.1. Model evaluation and comparison

ROC curves facilitate the task of *model evaluation and comparison*. Classifiers may be evaluated by merely observing their location in the ROC space. Given two classifiers, the former can be evaluated as better if it is situated more northern or western (or both) than the latter. If operating characteristics (i.e. class distribution and misclassification costs connected with each class) are unknown at the time of evaluation, global measure of performance in the form of AUC may be employed. The use of AUC as a performance measure for machine

learning algorithms is advocated in [7] and has already been discussed in Section 2.

ROC space can further be used for the redefinition of machine learning metrics [29]. The theory which would define the use of different metrics for various goals is discussed. It is claimed that such a theory is missing and the choice of which metric to use in a certain context is historically determined. As the main tool, authors use *isometrics*, i.e. sets of points for which a metric has the same value (employed also in [46] as iso-performance lines), and 2D ROC space. The latter is derived from 3D ROC space, represented by FPR on x-axis, TPR on y-axis, and relative frequency of positives (i.e. $P/(P+N)$) on z-axis, by discarding z-axis and introducing the *skew ratio*, a parameter summarizing operating characteristics. It is then argued that the defining characteristic of a metric is its effective skew landscape, i.e. the slope of its isometric at any point in 2D ROC space. As a result, a simplification of the F-measure and a version of the Gini splitting criterion which is said to be invariant to the class and cost distributions have been derived.

ROC analysis has been an interesting research topic during last decades and has been studied by many individuals from various perspectives. As a result, some disagreements like the following emerged. In [56], cautiousness is recommended when employing ROC analysis for classifier evaluation at varying class distributions. It is argued that ROC analysis cannot guarantee accurate evaluation of classifiers at unstable class distributions if a) the classes contain causally dependent subclasses whose frequencies may vary at different rates between the base and target data, or b) if there are attributes upon which the classes are causally dependent. A reply to the issues above can be found in [22]. It is argued that assertions in [56] are mainly related to only one of the two general domain types. Some real-world domains of the second type are given where ROC analysis is expected to be valid in spite of varying class distributions.

4.2. ROC convex hull and model selection

Applying ROC analysis to *model selection* enables the selection of an optimal model after information about the operating characteristics of the model deployment is obtained. The final choice of which model is the most appropriate is thus postponed until the deployment stage. A common selection procedure is described in [26]. Class distribution and error costs are combined to determine the slope of an auxiliary line positioned on an arbitrary location on the ROC graph. Afterwards, the line should be shifted in direction of the upper-left corner of the ROC space, until it touches the ROC convex hull (ROCCH, defined below) in one single point (i.e. the line becomes a tangent to the ROCCH). This point represents the optimal classifier for given operating characteristics.

ROCCH itself is a line connecting classifiers which may be optimal for some operating characteristic. This line is convex and no classifier can exist in the part of ROC space above it. Given a set of classifiers, their ROCCH can be generated by the following procedure. Classifiers have to be sorted according to their TPR values and plotted on an ROC graph. Afterwards, the line is constructed starting in (0,0) and connecting consequent points on the graph in

a way that the slope is always maximal possible. By repeating this step, one finally reaches the default classifier in (1,1), and all classifiers with minimum expected cost are located on the ROCCH. As an ROC curve, ROCCH is also useful when target conditions are completely unknown, representing a global performance indicator over all possible conditions. Figure 5 shows ROC curves of classifiers B and C together with their ROCCH. The key feature of ROCCH is that for some deployment contexts it may perform better than (and always at least equal to) the best of its constituent classifiers. In the figure such a case may be observed in the FP interval [0.15, 0.40].

High level of similarity and principal differences between ROC convex hull and ROC curve should be noted. Both are always monotonically non-decreasing, while a convex hull, as the name suggests, has to be convex as well. Nevertheless, some authors use a term "curve" when actually having in mind "convex hull", thus attention should be given when this distinction may be of importance. Transforming an ROC curve into ROC convex hull is feasible since given any two classifiers in ROC space, arbitrary classifier on the line connecting these two may be obtained by assigning a weight to each of them and then choosing one randomly (according to the weights) every time a new example is processed.

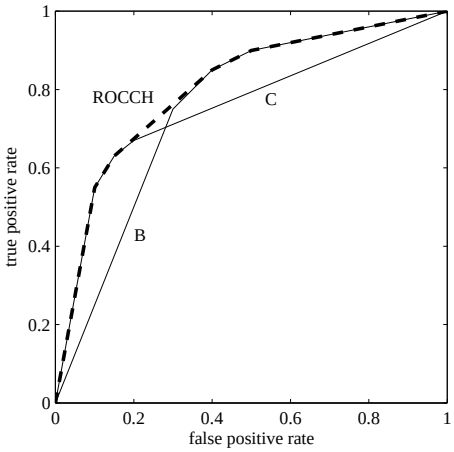


Figure 5: ROC convex hull

4.3. Model presentation

ROC analysis is based on graphical principles what makes it an appropriate tool for *model presentation*. As such, it visualizes performance and facilitates other activities concerning classifiers' analysis, e.g. model evaluation and selection, as well as our notion of a classification model. It may be combined with other visualization techniques, several of which are listed in Section 6, e.g., AUC has been put in the role of the y-axis of a learning curve technique [6].

4.4. Model construction

Learning algorithms may be adjusted to aim at constructing classifiers with good ROC curves instead of optimizing other criteria. In [23], the procedure implementing this idea for induction of decision trees is shown. The goal is to generate a decision tree for which its set of derived trees will result in an ROCCH with the maximum area under it. A novel splitting criterion for decision trees based on the AUC measure is defined. The criterion chooses the split which has the highest local AUC and is argued to be the first splitting criterion based on the estimated probabilities that is not a weighted average of impurities of the children.

A method with similar aims of obtaining multiple classification models from a single one is presented in [5]. These models focus on various target conditions and as such cover different areas of an ROC graph. The method is based on the finding that classifiers can carry additional information apart from their coordinates in the ROC space. Classifiers may often be divided into sub-components, each of them a classifier itself. As before, the approach is discussed more deeply for decision trees, but may be employed to other classification schemes, as well. Firstly, the leaves are ordered with regard to their probability of predicting a positive class. Predictions (positive and negative) in the leaves are then systematically varied, yielding new (biased) decision trees. It is argued that the method can only improve the ROCCH (as in the worst case the original classifier will dominate all derived ones) while at the same time being computationally cheap.

Another technique optimizing ROCCH has been presented in [50], using inductive logic programming (ILP) as the main tool. Background knowledge used to construct the models is usually obtained without having in mind some particular target conditions in which such models will operate. Employing irrelevant information may result in suboptimal or even incorrect models, thus only an appropriate subset of all background knowledge should be considered for given conditions. Therefore, the main idea is to construct a convex hull by repeatedly running an ILP system on various subsets of background information.

Further, ROC curves may be used to experimentally find more suitable decision thresholds for naive Bayes classifier [37]. Authors treat the threshold as an additional parameter of a model which should be learned from given data. Posterior estimates in naive Bayes classifier are scores, not real probabilities, and in the case that independence assumptions do not hold, probability estimates of the classifier will be inaccurate. In such a case, there is no well-grounded reason to predict the class whose posterior probability is higher than 0.5 (in a two-class problem). The algorithm for a two-class case is an adaptation of the algorithm for drawing ROC curves. The one for multi-class problems is implemented using weights (one per class) which are determined by greedy hillclimbing search. Its weaknesses are that finding a local optimum is not guaranteed and that the result may change while altering the order of classes. Searching for globally optimal solution has not been taken into account since it would result in an algorithm being computationally intractable. However, both algorithms are ca-

pable of considering non-uniform misclassification costs and are applicable to the recalibration of all learning methods which return class scores as a result.

4.5. Model combination

The idea of *model combination* is to combine a set of classifiers to obtain a hybrid model which demonstrates improved performance with respect to its component classifiers. One may combine different models, or alternatively, a single model with various parameter settings. A common principle which may be employed to this end is ROCCH [46]. With this technique it is possible to obtain a combined model which will classify at least as well as the best of its constituent models for all possible operating characteristics. Theorem in [46] states that a hybrid model can achieve a tradeoff between true and false positive rates given by any point on the ROCCH, not only the vertices (the latter represent given classifiers), which results in a fact that sometimes a hybrid can actually be superior to the best classifier known. ROCCH only includes classifiers that are optimal for some (FP,TP) pair and discards all other sub-optimal models which contributes to its spacial efficiency.

Another approach to model combination is given in [28]. Two methods (model assembly algorithms) are proposed to discover concavities in ROC curves and repair them. The idea is in adapting the predictions of questionable quality. This active manipulation of model performance is claimed to be a novelty. The goal of *SwapOne* algorithm is to enlarge the AUC of a probabilistic classifier by taking three models from different thresholds into account. A hybrid model is constructed by combining two better models and an inversion of the inferior one. The second algorithm, named *SwapCurve*, is designed to enlarge the AUC of a probabilistic classifier by detecting a section of ROC curve that is under its convex hull. Afterwards, the ranks of the instances in that section are inverted. The algorithm is claimed to be applicable to any model that computes a score or rank.

5. Improvements of the basic ROC technique

Since the basic ROC analysis is not able to answer all questions pertaining to classifiers' performance sufficiently, some extensions, approximations and other improvements have emerged over time. Some have been found to be useful while the practicability of others remains unclear. In this Section the following advantageous upgrades of the basic ROC approach are discussed: ROC convex hull, confidence intervals and confidence bands, extensions and approximations of ROC analysis to the multi-class case, variants of original ROC curves and AUC metric that takes scores and instance varying costs into account, and improvements that help to increase efficiency.

5.1. Robustness of classifiers

The *ROC convex hull (ROCCH)* determines the subset of classifiers that may be optimal. In this manner, a hybrid classifier, optimal for any target

conditions, including imprecise class distributions and misclassification costs, may be constructed. In model selection, method selects the potentially optimal classifier for given target conditions. ROCCH is argued to be a more robust tool for measuring classifier performance than single-number measures, e.g. classification accuracy and AUC [47]. The authors claim that classification accuracy is not appropriate for evaluation and comparison of classifiers when none of the considered classifiers dominates the others on a full range of operating conditions. In such cases, it may happen that the classifier with the highest value of classification accuracy will not be the one with minimum cost for specific target conditions. Without providing these conditions no single-number metric may possibly be trustworthy and ROCCH is argued to be able to tackle the problem its own way. Ranges of operating conditions where some particular classifier is optimal may be specified by expressing them in the form of slopes of tangents to ROCCH. As a result, a table of such regionally optimal classifiers is obtained. The claims are supported by the experimental study of classifiers, such as decision tree, naive Bayes and k-nearest neighbor, applied to ten UCI repository datasets. ROCCH method has been furher tested on other datasets, including one from a credit domain and two from a direct marketing, and the performance of various classifiers has been compared through misclassification costs [4].

A method to facilitate various tasks, robust to imprecise class distributions and misclassification costs, has originally been presented in [45, 46]. Auxiliary lines (mentioned before) also known as *iso-performance lines* have a characteristic that the expected cost of all classifiers located on it is equal. These lines are used to separate the performance of a classification model from the specific class and cost distributions. Similar approaches may as well be found in other techniques. *Cost curves* do not represent expected costs through the slopes of iso-performance lines but rather in an explicit form [16]. As such, they offer an alternative for visualization and are further discussed in Section 6. The *lower envelope* in cost space is equivalent to ROCCH in ROC space what is a consequence of duality of those two spaces. Each classification model defines a limit of a half-space and a lower envelope is then created by intersecting half-spaces of all given classifiers. The envelope may be employed as a tool for selecting the optimal model for the specific operating characteristic, i.e. the one minimizing cost.

A *PR space* (see Sect. 6) equivalent to ROCCH is presented in [12] as *achievable PR curve*. Algorithm for computing achievable PR curves is given there - it is based on computing the ROCCH in ROC space and transforming it to PR space. Firstly, points are transferred using simple contingency table calculations, and secondly, values between the newly computed points are interpolated. The latter step is not so straightforward as in ROC space, since values should not be linearly interpolated.

5.2. Reliability

For comparison and evaluation of two (or more) ROC curves, the AUC metric is traditionally applied. Since AUC is completely valid only when one

classification model dominates the other, the need for more reliable (statistical) approaches has evolved.

One of the approaches is introduction of one-dimensional confidence intervals for ROC curves which may be constructed by *Vertical Averaging (VA)* [47]. The method is designed on the principle of sweeping over the FPR axis, fixing FPR at regular interval, and averaging TPR values of ROC curves being compared at each of these fixed FPRs (by computing the mean). From newly obtained (FPR,TPR) points, the averaged ROC curve is generated by employing linear interpolation (this is possible, since any classifier on the line connecting two other classifiers may be simulated, see Subsection 5.1). Finally, at these points of the curve, the vertical confidence intervals for the mean of TPR values are calculated. While the procedure is simple and such confidence intervals are suitable for maximizing TPR given some FPR value, authors state that they may not be fully appropriate for the task of minimum expected cost evaluation. A noticeable disadvantage is that generally FPR may not be controlled by a researcher [19]. An example of the VA method usage is given in Fig. 6 where confidence intervals are presented by vertical lines. Two dotted lines connecting all upper, respectively lower ends of confidence intervals represent confidence bands mentioned in the last paragraph of this subsection.

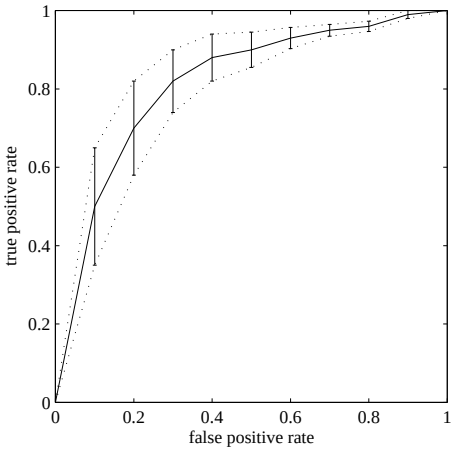


Figure 6: Vertical averaging

To overcome the potential disadvantage above, *Threshold Averaging (TA)* has been introduced [19]. Instead of fixing FPR, the technique is based on fixing the threshold of a scoring function. A set of thresholds is first generated. Afterwards, for each of these thresholds a relevant point on every ROC curve under comparison is located. The points (on different ROC curves) belonging to the same threshold value are then averaged using mean. In the resulting points confidence intervals may be calculated by applying standard deviation as in VA,

the difference is that now horizontal intervals may be produced as well. On the other hand, TA makes additional requirement that a classifier is able to issue scores. Moreover, since scores should not be directly compared across different classifiers, care should be taken when using this technique on their ROC curves, as the resulting averaged curve may be misleading. TA example may be seen in Fig. 7. As before, dotted lines denote the produced confidence bands.

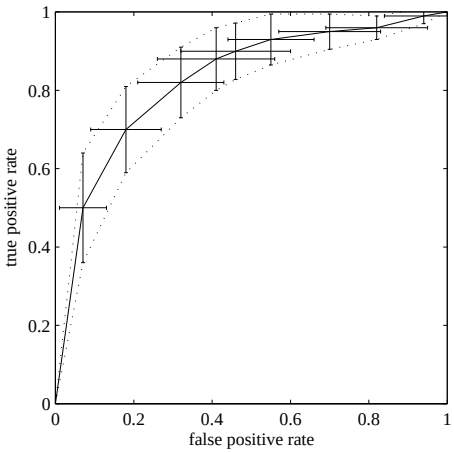


Figure 7: Threshold averaging

With the aim of obtaining a more robust statistical technique, methods for generating confidence bands for ROC curves are discussed in [40]. As ROC analysis has long been used by medical researchers, authors argue that it may beneficial to introduce these techniques and evaluate their fitness on various machine learning tasks. Authors in [40] present their methodology for constructing confidence bands by employing two existing machine learning techniques for generating confidence intervals, namely VA and TA, and introducing three techniques used in the medical field. The main idea of the methodology is that selected points produced by any of these methods are afterwards connected to form the upper and lower confidence bands of a ROC curve. An empirical evaluation has shown that bands generated by applying VA and TA methods are too tight. In the case of TA, this may also be a consequence of the procedure to convert confidence intervals into bands - horizontal intervals (FPR) have simply not been considered. The method which performs best in the evaluation is one of the three introduced from medicine - *Fixed-Width Bands (FWB)* method, of which an example is visualized in Fig. 8.

5.3. Generalization to multi-class problems

Original ROC analysis can only handle two-class decision problems. This is often satisfactory since numerous problems exist where the decision has to be

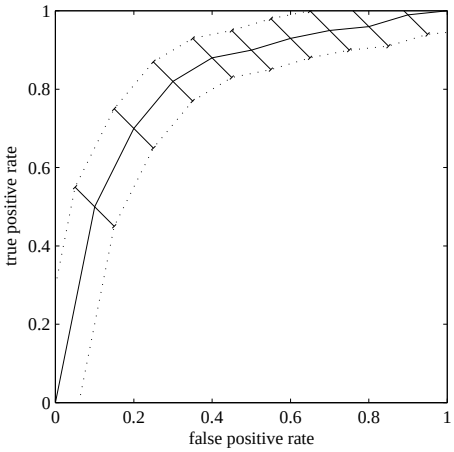


Figure 8: Fixed-width bands

made between two alternatives. Nevertheless, there are domains where three or more categorical alternatives should be considered. As the goal could be reached through the two-class ROC analysis by decomposing a multi-class problem to a number of binary problems, this often imposes a difficulty of dealing with such a system and understanding it. As a result, generalizations to multi-class ROC analysis have been developed. Since a contingency table for a n-class problem contains $n \cdot n$ elements (n correct classifications on the major diagonal, misclassifications elsewhere), the computational complexity becomes an important issue. The number of different misclassifications boosts up to $n^2 - n$, making it questionable how a reasonable visualization should look like. The outline of such issues can be found in [39].

ROC analysis has been extended to three-class decision problems in [42]. ROC surface can then be plotted in three dimensions. The *volume under the ROC surface (VUS)* for three-class problems is analogous to the AUC metric in two-class decision making and equals the probability that the classifier will correctly sort three items containing a randomly-selected instance from each of three classes. ROC surfaces can be compared via comparison of maximum information gain on each of them. A potential problem of three-class ROC analysis might be in estimating probabilities which is intellectually more complicated than in the two-class model. Domain experts in many fields of practice (e.g. physicians) make decisions between two classes without stating probabilities. Three-class case thus only adds difficulty of making good probability estimates. While with two classes estimates are needed for one pair of outcomes, in the three-class case the number of outcome pairs increases to three. The reliability (consistency from case to case) and validity (accordance with expert's opinion) of those estimates can thus become questionable.

It has been demonstrated that principles of ROCCH extend to multi-class problems and multi-dimensional convex hulls [49]. Optimal classifiers are shown to lie on the convex hull for arbitrary number of classes. In the latter work the author argues that given a finite set of points, the minimum value of any real-valued linear function is reached at the vertices of the convex hull of the points. This determines the position of classifiers with minimum cost and follows from results about convex sets that represent the base of linear programming algorithms.

One approach to the extension of AUC measure to the multi-class case is presented in [44]. All classes are, one by one, put in the role of a reference class (i.e. class 0), while at the same time, all other classes represent the alternative class (i.e. class 1). AUCs for all the resulting arrangements are calculated – put differently, the one-versus-all strategy is applied. Afterwards, the final AUC is obtained by computing the weighted average of above partial AUCs, where the weight of each AUC is proportional to the prevalence of its corresponding class in the data. The weakness of this approach is that the final multi-class AUC is sensitive to class distributions (and misclassification costs).

Another multi-class generalization of AUC is available in [33]. Since AUC itself does not offer an obvious extension to multi-class problems, it has been considered in the equivalent probabilistic form (see Section 2). A generalization is done by aggregation over all pairs of classes. Firstly, AUCs (more exactly, their approximations) for all possible combinations of (different) classes are computed. Then, the sum of these intermediate AUCs is divided by the number of all possible misclassifications. This averaging of pairwise comparisons is insensitive to class distributions and misclassification costs. One interesting feature of a new measure is, that its value becomes substantially higher for small improvements in pairwise separability. The extension is invariant to monotonic transformations of estimated probabilities. It should be noted that authors, in the view of representation adopted in our article, use swapped axes on an ROC graph.

An extension to the AUC measure in the form of VUS has been presented in [25]. The trivial classifiers, minimum and maximum VUS and the equations are given in the paper. The procedure of computing the polytopes (a multi-dimensional geometric object with flat sides; a polygon is a polytope in two dimensions) for the set of classifiers is presented. It consists of forming constraints (linear inequations) which are then solved by Hyperpolyhedron Search Algorithm (HSA). Volume of the hyperpolyhedron is computed using QHull algorithm [3]. This way, VUS of any classification model for any number of classes may be gained. The disadvantage of this technique is its inefficiency for a higher number of classes. It should be mentioned that authors use different representation of a ROC graph by plotting FNR on y-axis against FPR on x-axis. In this case, the goal becomes the minimization of area under the ROC curve what is essentially equivalent to maximization of the area *above* the ROC curve (AAC). However, the authors are consistent with the terminology and refer to the AAC as AUC.

Considering costs, ROC analysis has also been extended through the perspective of an optimization problem [18]. The latter has been used to define ROC

surface for the n class decision problem with the aim of minimizing $n \cdot (n-1)$ misclassification rates. Final solution to the problem is finding the optimal trade-off surface between different types of misclassification, known as *Pareto front*. Pareto front is represented by a set of all *Pareto optimal* solutions, each of them, in turn, being a solution not dominated by any other possible solution. An evolutionary algorithm for locating the Pareto front (based on greedy search) has been presented. In addition, a multi-class generalization of the Gini coefficient has been proposed. A similar multi-class generalization of ROC analysis to [18] is given in [49] (already mentioned in this subsection). It says that if classifiers for n classes are considered to be points with coordinates assigned by their $n \cdot (n-1)$ misclassification rates, then optimal classifiers lie on the convex hull of these points.

Cost-sensitive optimization is further discussed in [38]. It is argued that no method is known for handling multi-class ROC analysis for large numbers of classes. Algorithm in [18], for instance, is claimed to be only tested on domains with few classes and may become intractable since it uses sampling of operating points. A pairwise approximation is introduced to this end, which examines interactions between operating weight pairs (two-class ROC curves). The algorithm considers the most interacting pairings and the most expensive errors. Since some interactions are removed, the method becomes extensible to a large number of classes. Greedy hill-climbing algorithm in [37] can also be used for cost-sensitive optimization. One deficiency of the algorithm is that finding a local optimum is not guaranteed. Moreover, the result is influenced by the order of classes.

Yet another extension is available in [13]. It is demonstrated how a multi-class classifier can be directly optimized by maximizing VUS. This is accomplished in two steps. Firstly, the discrete U-statistic (that is equivalent to VUS) is approximated in a continuous way. Secondly, the resulting approximation is maximized by gradient ascent algorithm. The drawback of this approach is in its exponential time complexity.

5.4. Considering Scores

AUC ignores the scores (i.e. posterior probabilities of the positive class) and considers only ranks of the scores (i.e. order). Since a part of the information is ignored, this may lead to suboptimal results, for instance, overfitting the test set when selecting classifiers with high AUCs. Nonetheless, the advantage of the original AUC is the independence of any distribution assumptions. Intuitively, a kind of trade-off should be reached.

One option on how to evaluate the success of learning is to replace AUC by using some another measure, e.g. information gain, Brier score or LogLoss. However, these alternatives ignore the order. Another solution is to apply confidence bands for ROC curves [40]. Effect of those bands on AUC is, nevertheless, claimed to be unclear in the case that probabilities are non-uniformly distributed from 1 to 0 [24]. The third possible approach is to use one of the four evaluation methods developed by various authors, described in the following. These

measures consider both – scores and ranking – and are derived from the basic AUC metric.

Like with the basic AUC, their values can be calculated directly without the explicit construction of corresponding ROC curve variants. All variants also underestimate the value of the basic AUC. AUC may be generalized to a form in which basic metric and its variants can be expressed more uniformly. Such a generalization has been carried out in [54]. To calculate the value of any AUC variant for a given set of instances, we traverse all possible pairs of one positive and one negative instance and accumulate values of the *difference function* (called also the *modifier function* in [54]) for every pair. Difference function handles the difference of scores of the two instances in the individual pair. Afterwards, we divide the intermediate result with the number of all possible pairs. The generalized form of AUC may thus be interpreted as the mean value of the difference function for a set of instances. It is obvious that AUC and its variants diverge only in their difference functions, i.e. the way how the score difference is dealt with. Difference function for the basic AUC is the step function.

The proposed four variants of AUC metrics are briefly summarized in the following. Table 2 shows an example of the computed AUC variants for 15 different example outcomes for classifying 5 examples. The first table column provides the predicted probabilities and the instances' true classes; the following table columns illustrate how various AUC variants differently evaluate the classification quality. The AUC variants are:

1. In [24] the first variant, called *probabilistic area under the ROC curve (probAUC)*, is presented. Its difference function is defined using the probability and uniform (or alternatively normal) distribution. The *probAUC* metric may be interpreted as the mean of the (i) average predicted probability of belonging to the positive class issued for positives and (ii) the average predicted probability of belonging to the negative class issued for negatives. Although *probAUC* for the most part underestimates the value of AUC, the opposite may happen as well (as seen for the instances no. 11, 12 and 14 in Table 2). Since such performance can no longer be visualized by the basic ROC curves, an approach for drawing probROC curves (with area under the curve equal to the *probAUC*) is given. Authors find it useful to maintain the principle of ROC curves as they offer a way of selecting a classification threshold during the algorithm execution. As well as the ROC curves, the probROC curves are also constructed in ROC space. When constructing the probROC curves, probabilities denote the curve intervals. These curves have usually smoother shapes, main distinctions can especially be noticed in cases when the original ROC curve is unreliable (few instances, small differences between scores). For larger instance sets, the probROC curve behaves similar to the basic ROC curve.
2. The second variant is *scored area under the ROC curve (scorAUC)* [61, 60]. Its value is equal to the area under the *scorROC* curve. The scorROC curve illustrates how quickly the AUC deteriorates if positive scores are

decreased, i.e. how sensitive is classifier to the shift of score values, and *scorAUC* accumulates this information into a numerical metric. The scorROC curve is constructed in a space not much alike to the ROC space - here, the x-axis denotes value of the parameter τ, indicating the decrease degree of the positive scores, and y-axis denotes the AUC value of such a modified set. The difference function for the *scorAUC* is the step function, weighted by the difference of the scores. The *scorAUC* always underestimates the value of AUC metric - their values are equal only when a classifier issues perfect scores for the set of instances, i.e. predicts 1 for every positive instance and 0 for every negative instance (as seen from the example in Tab. 2). Since the *scorAUC* employs both, scores and ranks, the authors argue that it could be used as a statistic for testing the diversity of two samples (similar to the Wilcoxon-Mann-Whitney statistic).

3. Third variant, named *softened area under the ROC curve (sondAUC)*, has been proposed in [35]. The *sondAUC* is actually a generalized version of *scorAUC*. Difference functions are the same with the one exception - in the case of *sondAUC*, the difference between scores is raised to the power of some chosen parameter q, which is not the case with the *scorAUC*. The purpose of the exponent q is to regulate sensitivity and robustness to ranking alterations, which gives the user greater flexibility when selecting classifiers. By increasing q, the *sondAUC* usually becomes more sensitive and less robust to variations in ranking order. If the value of parameter q is set to 0, the *sondAUC* becomes the basic AUC metric.
4. The last variant is called *soft AUC (softAUC)* [9]. Besides considering the scores, the main motivation for defining such a metric was its continuity and differentiability; the resulting *softAUC* therefore possesses all these features. The difference function of the *softAUC* is a sigmoidal function (more precisely, a logistic function) with parameter β. The sigmoid has a role of approximating the step function smoothly and converges to the latter as $\beta \to \infty$, i.e. the *softAUC* becomes equal to the basic AUC in such a case.

Methods of drawing *probROC* and *scorROC* curves have been presented together with their corresponding AUC variants, while the notion of *sondROC* and *softROC* curves has not been explicitly mentioned. All four variants are softer than the basic AUC as they aim at smoothing the difference function of AUC. As such, they are intuitively expected to be more robust for small data sets. Difference functions of *probAUC*, *scorAUC* and *softAUC* variants are visualized in [54].

Example 1. *In Tab. 2 we provide an artificial example, intended to illustrate behaviors of different AUC variants. The table contains 15 example data sets comprised of five examples. Each example is represented by its probability of belonging to the positive class (denoted with a real number) as predicted by a*

Table 2: Instance sets presented by predicted probabilities and true classes, with calculated values of AUC and its variants for each set

#	set of instances	AUC	probAUC	scorAUC	sondAUC	softAUC	q	β
1a	1.00+ 1.00+ 1.00+ 0.00− 0.00− 0.00−	1.000	1.000	1.000	1.000	1.000	1/7	20.0
1b	1.00+ 1.00+ 1.00+ 0.00− 0.00− 0.00−	1.000	1.000	1.000	1.000	0.999	1/7	7.0
1c	1.00+ 1.00+ 1.00+ 0.00− 0.00− 0.00−	1.000	1.000	1.000	1.000	0.881	1/7	2.0
1d	1.00+ 1.00+ 1.00+ 0.00− 0.00− 0.00−	1.000	1.000	1.000	1.000	0.731	1/7	1.0
1e	1.00+ 1.00+ 1.00+ 0.00− 0.00− 0.00−	1.000	1.000	1.000	1.000	0.599	1/7	0.4
2	0.97+ 0.95+ 0.92+ 0.09− 0.06− 0.05−	1.000	0.940	0.880	0.982	0.998	1/7	7.0
3	0.94+ 0.94+ 0.94+ 0.58− 0.58− 0.58−	1.000	0.680	0.360	0.864	0.926	1/7	7.0
4	0.94+ 0.88+ 0.82+ 0.61− 0.59− 0.55−	1.000	0.648	0.297	0.839	0.883	1/7	7.0
5	0.90+ 0.70+ 0.60+ 0.40− 0.10− 0.00−	1.000	0.783	0.567	0.912	0.955	1/7	7.0
6	1.00+ 1.00+ 1.00+ 0.90− 0.90− 0.90−	1.000	0.550	0.100	0.720	0.668	1/7	7.0
7	0.60+ 0.57+ 0.56+ 0.54− 0.52− 0.51−	1.000	0.527	0.053	0.651	0.592	1/7	7.0
8	0.95+ 0.83+ 0.77− 0.75+ 0.69− 0.40−	0.889	0.612	0.226	0.707	0.766	1/7	7.0
9a	0.61+ 0.61+ 0.61+ 0.60− 0.60− 0.60−	1.000	0.505	0.010	0.215	0.517	1/3	7.0
9b	0.61+ 0.61+ 0.61+ 0.60− 0.60− 0.60−	1.000	0.505	0.010	0.398	0.517	1/5	7.0
9c	0.61+ 0.61+ 0.61+ 0.60− 0.60− 0.60−	1.000	0.505	0.010	0.518	0.517	1/7	7.0
9d	0.61+ 0.61+ 0.61+ 0.60− 0.60− 0.60−	1.000	0.505	0.010	0.736	0.517	1/15	7.0
9e	0.61+ 0.61+ 0.61+ 0.60− 0.60− 0.60−	1.000	0.505	0.010	0.995	0.517	1/1001	7.0
10	1.00+ 0.80− 0.60+ 0.25− 0.20+ 0.00−	0.667	0.625	0.344	0.593	0.681	1/7	7.0
11	1.00+ 0.90− 0.65− 0.56+ 0.43+ 0.00−	0.556	0.573	0.271	0.487	0.574	1/7	7.0
12	0.61− 0.61− 0.61− 0.60+ 0.60+ 0.60+	0.000	0.495	0.000	0.000	0.483	1/7	7.0
13	1.00+ 1.00+ 1.00+ 1.00− 1.00− 1.00−	0.500	0.500	0.000	0.000	0.500	1/7	7.0
14	0.90− 0.77+ 0.65− 0.56+ 0.43+ 0.22−	0.444	0.498	0.136	0.368	0.482	1/7	7.0
15	1.00− 1.00− 1.00− 0.00+ 0.00+ 0.00+	0.000	0.000	0.000	0.000	0.001	1/7	7.0

classifier (i.e. scores), and a sign denoting true classes (+ denotes the positive class and − denotes the negative class). For each data set, the values of the basic AUC and its four variants are calculated. A couple of data sets (no. 1 and 9) are used several times (thus annotated with suffixes a-e) for different values of parameters q and β which are required for the calculation of the sondAUC and softAUC, respectively. The data sets are ordered from intuitively the best to intuitively the worst (in the decreasing order). The table serves as an illustration of how basic AUC and its variants evaluate some possible instance sets. The most obvious characteristics are exposed as follows:

- *Data sets 1a-1e represent an ideal case in which the classifier issues a score 1 for every positive instance and score 0 for every negative instance. Taking a look on these sets, an effect of varying a parameter β may be seen. As $\beta \to \infty$, the softAUC converges to the basic AUC metric. In the opposite extreme, as $\beta \to 0$, the value of softAUC approaches 0.5. For the further example sets, we have chosen a fixed value of $\beta = 7$, as the softAUC behaves similar to the basic AUC, yet sufficiently distinct to deserve its own focus of observation.*

- *In data sets 2-7, all examples have still been perfectly ranked, although the margin between positive and negative instances has become more narrow. Comparing the sets 4 and 5, it may be observed that the latter is somewhat better calibrated than the former. On the other hand, as the score difference (the margin) between positives and negatives is slightly larger in the data set 4 ($0.82 - 0.61 = 0.21$ in contrast to $0.60 - 0.40 = 0.20$), we may say that this classifier slightly better separates the positive from the*

negative instances. Nevertheless, all AUC variants evaluate the data set 5 as more preferable than the data set 4. Here, the calibration predominates over the difference between the positive and the negative scores. Considering the data set 3, values of all AUC variants are still rather lower than for the data set 5, even though the score margin increased to remarkable 0.36. *Of all variants, it seems that the scorAUC relies on the calibration the most.*

- *In the data set 6, a narrow margin between positive and negative instances (*0.10*) is perceived as undesirable by all AUC variants. Classifier's predictions are in this case fully consistent - all positive instances are equipped with the score 1.00 and all negative with the score 0.90. Its ranking performance on a given set is perfect, while the only deficiency seems to be the improper calibration. The basic AUC metric only considers ranking, while its variants consider the calibration, as well. On account of the narrow score margin, the bad calibration may outweigh the correct ranking order in evaluation by the AUC variants. Examples of this phenomenon are given below (see description of the data set 14). Since an accurate classifier does not have to be well calibrated, and besides, there exist methods calibrating classifiers, we may say that in this specific case the basic AUC presents the classifier's quality more credibly than its variants.*

- *Another similar example follows from comparing the data sets 7 and 8. A ranker with an ideal performance may be seen in the data set 7. Although the classifier in the data set 8 does not rank all instances perfectly by misclassifying two instances with a small difference in scores, it still performs very well. Thus, the better calibrated classifier for the data set 8 has been evaluated with higher grades by all AUC variants, most noticeably by the scorAUC.*

- *The data sets 9a-9e demonstrate the influence of the parameter q on the behavior of the sondAUC metric. When $q = 1$, the sondAUC is identical to the scorAUC. As $q \to 0$, the sondAUC becomes more similar to the basic AUC metric. When choosing the value of q as a rational number between 0 and 1, some care should be taken, as with the even denominators of the fraction a problem with calculating a root of a negative number could arise (occurs always when some instances are improperly ranked). For example, the value of q may be set to $\frac{1}{3}$ or $\frac{1}{5}$, but not to $\frac{1}{2}$. In the following example data sets, we have decided to use a fixed $q = \frac{1}{7}$ as it seems to offer an appropriate balance between robustness and sensitivity.*

- *The data sets 9 and 12 expose the main disadvantage of AUC measure of which its variants strive to overcome - the unreliability for the (small) sets where the differences between the predicted scores are negligible. As we can see, a tiny variation in score values results in an enormous change of the AUC value (since AUC fully trusts the ranking order). AUC variants, on the other hand, issue quite consistent values for both, very similar sets.*

This shows a typical example of when it is more advisable to rely on the evaluation results of the AUC variants.

- *The data set 13 represents a case when the observed classifier is of no use - it issues the score 1 for every instance it sees and obviously does not separate the positive instances from the negative ones. A classifier does not differentiate between the positive and the negative instances in the data set 14, neither. It gives an impression that score values are issued randomly. Considering the data set 14 which depicts a classifier of essentially inferior quality to the ideal rankers of data sets 6 and 7, we observe that the scorAUC metric reported higher values in the former than in the latter case. Similarly, the obtained probAUC and scorAUC values for the data set 11 are higher compared to the data set 6. This kind of behavior where the correct ranking is outweighed by the bad calibration may also be noticed for the softAUC metric while comparing the data sets 10 and 6. Finally, the data set 15 is the worst possible, though an ideal case (the data set 1) is retrieved trivially by inverting the classifier's decisions (i.e. interpret 0 as a positive instance and 1 as a negative).*

A much more extensive analysis of the three AUC variants, namely *probAUC*, *scorAUC* and *softAUC*, has been provided in [54], where their performance is claimed to be questionable. It is argued that none of the variants should surpass the basic AUC, at least when applying them to evaluation and selection of classifiers. The variants are all claimed to be biased with the variance possible in either direction, what makes their theoretical foundations unsteady. Nevertheless, the *probAUC* and the *softAUC* with appropriately chosen parameter values are recognized as exact approximations of the basic AUC metric.

5.5. Considering Instance-Varying Costs

One of the limitations of ROC graphs is their inability of handling problems where misclassification costs vary from example to example of the same class. Such costs are called *instance-varying costs* (also, *example-specific costs*) and appear quite often in real-world problems. ROC graphs use true and false positive rates to construct a curve and assume that errors of one type are all equal.

A *ROCIV graph*, a transformation of the original ROC graph, is an approach to struggle with instance-varying costs [21]. The intuitive interpretation of a ROCIV curve is that the axes are scaled by example costs within each class. In such manner, the y-axis represents the true positive benefit, while the x-axis represents the false positive cost. A ROCIV curve is constructed similarly as a ROC curve: for each positive (negative) instance, its benefits (costs) are incremented accordingly. The interpretation of the area under the ROCIV curve (AUCIV) is related to the one of the original AUC and equals the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance given that each is chosen in proportion to their costs. If instance costs in a given class are all equal, the ROCIV graph is

identical to the ROC graph. ROCIV graphs may offer more accurate picture of classifier quality in domains where error costs are not uniform within a single class and may prefer different classifiers (in different regions of the problem space) than traditional ROC graphs. On the other hand, the original ROC graphs presume that true and false positive rates of the test set will be similar to those in the training set. In the case of ROCIV graphs a new presumption arises, that the instance costs will also be similar. This new presumption is in contradiction with the feature of ROC curves that they are cost-invariant, and represents a potential drawback. A ROCIV curve thus becomes sensitive to variations in inter-class misclassification costs, but remains insensitive to intra-class variations. Inter-class error cost distributions in training and test sets should therefore be additionally checked for consistency.

5.6. Efficient computation of the AUC

Repetitive computations of AUC can be relatively time-consuming. Since such computations are important for many techniques, such as the methods for direct optimization of AUC, aspirations for more effective algorithms emerged.

In [9], a polynomial approximation of AUC has been presented. Similarly as in the case of AUC variants, the only distinction between the exact (basic) AUC and its approximation is in the difference function: step function is replaced by a general form of a polynomial. The degree of a polynomial should be chosen deliberately, optimizing the relation between accuracy and performance. The approximation is claimed to be more accurate than sampling, being computable in one pass over the database and thus having linear time-complexity.

6. Alternatives to the tools of ROC analysis

In this section, some other techniques which may be used instead of the ROC graphs are listed and discussed. Some of them are strongly related to the ROC graphs while still providing an alternative form of presentation which may be favorable in particular domains.

1. *Detection error tradeoff curve (DET curve)* is an approach to performance comparison, presented in [41]. It is based on the principles of the ROC curves, nonetheless, in DET space, y-axis denotes FNR instead of TPR (as shown in Fig. 9d). This way the error rates are plotted on the both axes which have normal deviate scale. Such a scale makes curves in DET space nearly straight lines. If classifiers perform well enough, the plot may be limited to the lower-left quadrant. Both modifications spread the curves and facilitate their evaluation. In this form of representation an ideal classifier is situated in the lower-left corner.
2. *Loss comparison plot (LC plot)* and *Loss comparison index (LC index)* have been presented as another alternative [1, 2]. LC plots are intended to compare the classification models by explicitly showing cost values for which some individual model is better (shown in Fig. 10a). The triangular form represents belief distribution of a quotient of misclassification

costs, i.e., what is the confidence that some particular quotient value will appear, and its area is defined to be equal 1. LC index, further, is a comparative measure of the classifiers' performance which makes it possible to employ any available information about the relative importance of two misclassification types. LC index does not represent an absolute index of performance and should not be interpreted as an expected loss. Both techniques are of a benefit if some information about the ratio of misclassification costs, including the interval of possible ratio values and the most probable ratio value, is available.

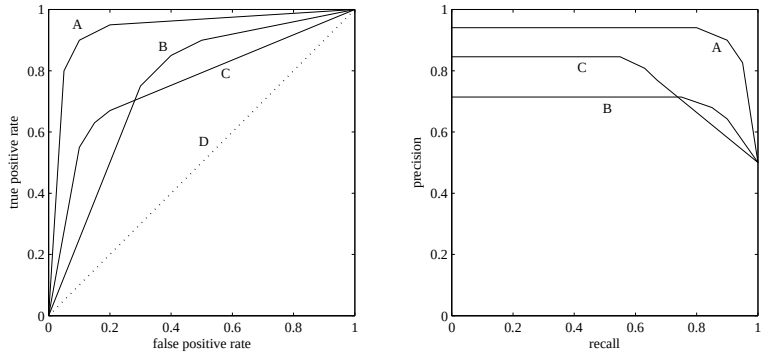
3. *Lift chart* is a technique similar to the ROC graphs and is commonly used in some branches of data mining [57]. It is defined by TP on its y-axis and the size of a subset (proportion) on its x-axis (as shown in Fig. 9e) which makes it sensitive to the variations in class distribution.
4. *Calibration chart* (also calibration plot, calibration graph) is an approach that demonstrates how well a given classification model is calibrated, and can be used to recalibrate it [10]. Calibration chart plots the actual probability on y-axis against the predicted probability on x-axis (as shown in Fig. 10c) and is sensitive to changes in the class distribution. It cannot reflect the sole quality of classification but is able to recognize a classifier's bias, which can be used as a performance measure [55].
5. *Learning curve plot* is another graphical technique for visualization of a classifier's performance. x-axis measures the size of a training set while y-axis indicates the performance of a classification model (as shown in Fig. 10b). As such, a learning curve depicts how the amount of learning depends on the number of instances in the training set and shows when the continuation of learning has no further effects. It can be used for comparing classifiers built on different data set sizes. As a measure of classifier performance (y-axis) the AUC can be applied [11, 52]. Learning curve with the AUC on the y-axis is further studied in [6], where methods for incremental updating of exact AUC learning curves and calculating their approximations are given.
6. *Cost curve* is a method developed with the aim of redressing the tools of ROC analysis [16] and seems to be one of the most promising among various alternatives to ROC graphs [14, 15]. Information implicitly contained in ROC graphs is presented explicitly with cost curves, the approach is therefore claimed to be more clear. Ranges of class distributions and misclassification costs for which one classifier is superior to other models may be easily obtained, as well as quantitative differences between them. On the x-axis, the probability-cost function for positive instances is represented, while the y-axis represents the normalized expected cost (as shown in Fig. 9f). The area under such a curve measures the final expected cost. An ideal zero-cost classifier lies on the x-axis. Both representations are dual, i.e. a point in the ROC space may be translated into a line in the cost space, whereas a line in the ROC space converts into a point in the cost space.

7. *PN graph* is a graphical technique in close relation to ROC graphs. It plots TP on y-axis against FP on x-axis (as shown in Fig. 9c) and can be transformed to an ROC graph by scaling both axes to the interval [0.00, 1.00]. Both relatives have been compared in [30] and [31]. Acronym PN is derived from the titles of both axes, as the y and x are sometimes labeled as "P" for covered positive examples and "N" for covered negative examples, respectively.
8. *Precision-recall curve (PR curve)* is an alternative which is often used in information retrieval and can be beneficial in cases when the class distribution is highly skewed (i.e., highly imbalanced). In PR space, precision on the y-axis is plotted against recall (which is equal to TPR) on the x-axis (as shown in Fig. 9b). An ideal classifier is located in the upper-right corner of the PR space. PR and ROC curves are compared and studied in [12], where it has been demonstrated that a curve of a given classifier dominates in the ROC space if and only if it dominates in the PR space as well. An important difference is that in the PR space the curve should not be constructed by linearly interpolating values between two points, i.e. it is incorrect to simply connect two (distant) points with a straight line.

Example 2. *We provide an example case in which we compare three classifiers trained on an imaginary set of 100 positive and 100 negative instances. We presume their misclassification costs to be all equal, i.e. all having value of* 1*. With such a set-up, a high level of similarity among alternatives is revealed. It should be noted, however, that quite a different picture may be seen when the class skewness gets high or cost of a false positive substantially differs from the cost of a false negative. The described ROC alternatives present classifiers' performance from different angles and thus may be helpful in better understanding of a given problem. They are applicable to all the application areas of ROC analysis (see Section 4), e.g. model evaluation and presentation.*

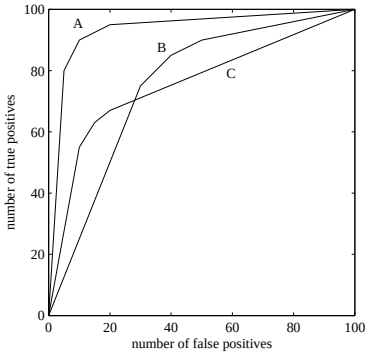
Some of the alternative techniques are computed based on the same data as the ROC curves, i.e. a contingency table, and can be easily transformed from one representation to another. Such techniques are shown in Fig. 9. On the other hand, the second group of techniques measures substantially different features and requires other information, as well (e.g. classifier scores, performance measured while varying the size of a training set, etc.). Some of such performance curves are shown in Fig. 10. A transformation to some of them may only be accomplished if all the needed information is available. In all representations, the performance of the same classifiers A, B and C (same classifiers as in Figs. 3 and 9a) is visualized in different ways.

To present a meaningful LC plot (Fig. 10a), we change our initial presumption that misclassification costs for both types of misclassification are equal. In our example we presume that the misclassification of negative instances (FPs) is between two and five times as serious as misclassification of positive instances (FNs), with the most probable ratio being three. These three values determine the location of three key points on the LC plot. As FPs are more costly than

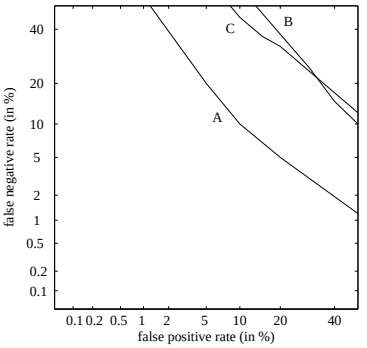


(a) ROC graph

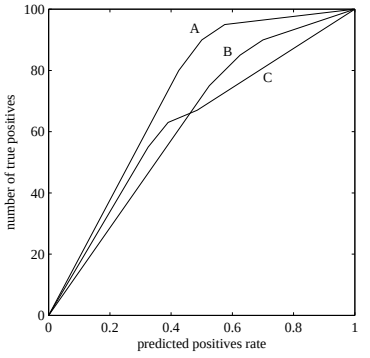
(b) Precision-Recall curve



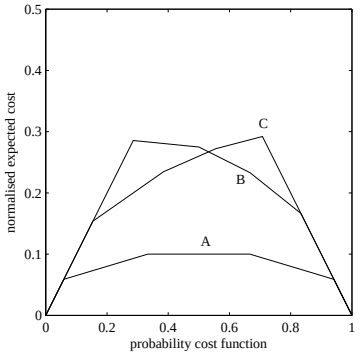
(c) PN graph



(d) Detection error tradeoff curve

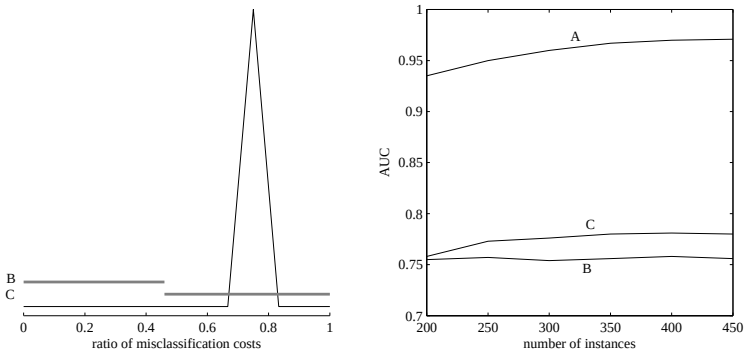


(e) Lift chart



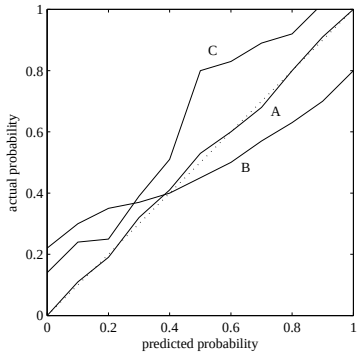
(f) Cost curve

Figure 9: Alternatives to ROC graphs



(a) Loss comparison plot

(b) Learning curve plot



(c) Calibration chart

Figure 10: Alternatives to ROC graphs which are defined using additional data

FNs, the goal should be to stay on the left side of an ROC graph, near the ordinal axis. In this area classifier C is superior to classifier B (A is not included in LC plot since it is superior to both, B and C, for any context) and this is exactly what the LC plot reveals.

Learning curves in Fig. 10b reveal the information which is not contained in a single ROC graph, since the latter assumes a fixed number of training examples. A ROC graph with a fixed number of instances thus represents only a single point on each learning curve in a learning curve plot. In Fig. 10b, only three points are therefore actually related to our ROC graph example in Fig. 9a, namely those three which represent AUC value for 200 instances. In a similar way, the calibration chart in Fig. 10c is based on the data not contained in the ROC graphs, but rather on the additional information gained from the data set.

7. Conclusion

In our paper we presented the basic concepts of ROC analysis in the area of machine learning. We explained the basic notions of this approach as well as the well-known ways of how the ROC curves and the AUC measure can be employed to tackle different classifier optimization problems.

Important improvements of the basic two-class ROC curves and the AUC metric, including their generalizations to the multi-class case, have been mentioned. Further, we shed some light on alternative approaches. The resulting survey provides relevant information gathered on one place, and may serve as a signpost to other articles where the topic is discussed in greater detail.

Since the ROC analysis is still becoming increasingly popular in the field of machine learning, this review shall still be complemented with other approaches and possible improvements. This, as well as analyzing applications of ROC analysis in other machine learning areas (e.g. in regression), is the intended focus of our further work.

References

[1] N.M. Adams, D.J. Hand, Comparing classifiers when the misallocation costs are uncertain, Pattern Recognition 32 (1999) 1139–1147.

[2] N.M. Adams, D.J. Hand, An improved measure for comparing diagnostic tests, Computers in Biology and Medicine 30 (2000) 89–96.

[3] C.B. Barber, D.P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software 22 (1996) 469–483.

[4] R. Bettinger, Cost-sensitive classifier selection using the ROC convex hull method, Computing Science and Statistics 35 (2003) 142–153.

[5] H. Blockeel, J. Struyf, Deriving biased classifiers for better ROC performance, Informatica 26 (2002) 77–84.

[6] R.R. Bouckaert, Efficient AUC learning curve calculation, in: Proceedings of the Nineteenth Australian Joint Conference on Artificial Intelligence, pp. 181–191.

[7] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (1997) 1145–1159.

[8] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees, Wadsworth International Group, Belmont, CA, USA, 1984.

[9] T. Calders, S. Jaroszewicz, Efficient AUC optimization for classification, in: Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 42–53.

[10] I. Cohen, M. Goldszmidt, Properties and benefits of calibrated classifiers, in: Proceedings of the Eigth European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 125–136.

[11] M. Culver, D. Kun, S. Scott, Active learning to maximize area under the ROC curve, in: Proceedings of the Sixth International Conference on Data Mining, pp. 149–158.

[12] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the Twenty-third International Conference on Machine Learning, ACM Press, New York, NY, USA, 2006, pp. 233–240.

[13] S. Dreiseitl, Training multiclass classifiers by maximizing the volume under the ROC surface, in: Proceedings of the Eleventh International Conference on Computer Aided Systems Theory, pp. 878–885.

[14] C. Drummond, R. Holte, What ROC curves can't do (and cost curves can), in: Proceedings of the First Workshop ROC Analysis in Artificial Inteligence, pp. 19–26.

[15] C. Drummond, R. Holte, Cost curves: An improved method for visualizing classifier performance, Machine Learning 65 (2006) 95–130.

[16] C. Drummond, R.C. Holte, Explicitly representing expected cost: An alternative to ROC representation, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, NY, USA, 2000, pp. 198–207.

[17] J.P. Egan, Signal detection theory and ROC analysis, Series in Cognition and Perception, Academic Press, New York, NY, USA, 1975.

[18] R.M. Everson, J.E. Fieldsend, Multi-class ROC analysis from a multi-objective optimisation perspective, Pattern Recognition Letters, special issue on ROC analysis 27 (2006) 918–927.

[19] T. Fawcett, ROC graphs: Notes and practical considerations for data mining researchers, Technical Report HPL-2003-4, HP Laboratories, Palo Alto, CA, USA, 2003.

[20] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters, special issue on ROC analysis 27 (2006) 861–874.

[21] T. Fawcett, ROC graphs with instance-varying costs, Pattern Recognition Letters, special issue on ROC analysis 27 (2006) 882–891.

[22] T. Fawcett, P.A. Flach, A response to webb and ting's on the application of ROC analysis to predict classification performance under varying class distributions, Machine Learning 58 (2005) 33–38.

[23] C. Ferri, P. Flach, J. Hernández-Orallo, Learning decision trees using the area under the ROC curve, in: Proceedings of the Nineteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2002, pp. 139–146.

[24] C. Ferri, P. Flach, J. Hernández-Orallo, A. Senad, Modifying ROC curves to incorporate predicted probabilities, in: Proceedings of the Second Workshop on ROC Analysis in Machine Learning.

[25] C. Ferri, J. Hernández-Orallo, M.A. Salido, Volume under the ROC surface for multi-class problems, in: Proceedings of the Fourteenth European Conference on Machine Learning, pp. 108–120.

[26] P. Flach, H. Blockeel, C. Ferri, J. Hernández-Orallo, J. Struyf, Decision support for data mining: Introduction to ROC analysis and its application, in: Data Mining and Decision Support: Aspects of Integration and Collaboration, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003, pp. 81–90.

[27] P. Flach, S. Wu, Repairing concavities in ROC curves, in: Proceedings of the 2003 UK Workshop on Computational Intelligence, pp. 38–44.

[28] P. Flach, S. Wu, Repairing concavities in ROC curves, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Professional Book Center, Denver, CO, USA, 2005, pp. 702–707.

[29] P.A. Flach, The geometry of ROC space: Understanding machine learning metrics through ROC isometrics, in: Proceedings of the Twentieth International Conference on Machine Learning, AAAI Press, Menlo Park, CA, USA, 2003, pp. 194–201.

[30] J. Fürnkranz, P. Flach, An analysis of rule learning heuristics, Technical Report CSTR-03-002, Department of Computer Science, University of Bristol, Bristol, UK, 2003.

[31] J. Fürnkranz, P.A. Flach, An analysis of rule evaluation metrics, in: Proceedings of the Twentieth International Conference on Machine Learning, AAAI Press, Menlo Park, CA, USA, 2003, pp. 202–209.

[32] D.M. Green, J.A. Swets, Signal detection theory and psychophysics, John Wiley and Sons, New York, NY, USA, 1966.

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

[34] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29–36.

[35] J. Huang, C.X. Ling, Partial ensemble classifiers selection for better ranking, in: Proceedings of the Fifth IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, 2005, pp. 653–656.

[36] T.D. Koepsell, N.S. Weiss, Epidemiologic methods: Studying the occurrence of illness, Oxford University Press, New York, NY, USA, 2003.

[37] N. Lachiche, P. Flach, Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves, in: Proceedings of the Twentieth International Conference on Machine Learning, AAAI Press, Menlo Park, CA, USA, 2003, pp. 416–423.

[38] T.C.W. Landgrebe, R.P.W. Duin, Approximating the multiclass ROC by pairwise analysis, Pattern Recognition Letters 28 (2007) 1747–1758.

[39] T. Lane, Extensions of ROC analysis to multi-class domains, in: Proceedings of the ICML-2000 Workshop on Cost-Sensitive Learning.

[40] S.A. Macskassy, F. Provost, Confidence bands for ROC curves: Methods and an empirical study, in: Proceedings of the First Workshop on ROC Analysis in Artificial Intelligence, pp. 61–70.

[41] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, The DET curve in assessment of detection task performance, in: Proceedings of EuroSpeech, pp. 1895–1898.

[42] D. Mossman, Three-way ROC's, Medical Decision Making 19 (1999) 78–89.

[43] N.A. Obuchowski, Receiver operating characteristic curves and their use in radiology, Radiology 229 (2003) 3–8.

[44] F. Provost, P. Domingos, Well-trained PETs: Improving probability estimation trees, CeDER Working Paper IS-00-04, Stern School of Business, New York University, New York, NY, USA, 2000.

[45] F. Provost, T. Fawcett, Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, USA, 1997, pp. 43–48.

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

[47] F.J. Provost, T. Fawcett, R. Kohavi, The case against accuracy estimation for comparing induction algorithms, in: Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1998, pp. 445–453.

[48] K.A. Spackman, Signal detection theory: Valuable tools for evaluating inductive learning, in: Proceedings of the Sixth International Workshop on Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1989, pp. 160–163.

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

[50] A. Srinivasan, Extracting context-sensitive models in inductive logic programming, Machine Learning 44 (2001) 301–324.

[51] J.A. Swets, Measuring the accuracy of diagnostic systems, Science 240 (1988) 1285–1293.

[52] D.M.J. Tax, R.P.W. Duin, Learning curves for the analysis of multiple instance classifiers, in: Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, pp. 724–733.

[53] F. Tortorella, A ROC-based reject rule for support vector machines, in: Proceedings of the Third International Conference on Machine Learning and Data Mining, Springer-Verlag, Berlin, Germany, 2003, pp. 106–120.

[54] S. Vanderlooy, E. Hüllermeier, A critical analysis of variants of the AUC, Machine Learning 72 (2008) 247–262.

[55] M. Vuk, T. Curk, ROC curve, lift chart and calibration plot, Metodološki zvezki 3 (2006) 89–108.

[56] G.I. Webb, K.M. Ting, On the application of ROC analysis to predict classification performance under varying class distributions, Machine Learning 58 (2005) 25–32.

[57] I. Witten, E. Frank, Data mining: Practical machine learning tools and techniques with java implementations, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2000.

[58] K. Woods, Computer-aided diagnosis and automated screening of digital mammograms, Annual Report DAMD17-94-J-4328, University of South Florida, Tampa, FL, USA, 1995.

[59] K. Woods, K.W. Bowyer, Generating ROC curves for artificial neural networks, IEEE Transactions on Medical Imaging 16 (1997) 329–337.

[60] S. Wu, P. Flach, C. Ferri, An improved model selection heuristic for AUC, in: Proceedings of the Eighteenth European Conference on Machine Learning, pp. 478–489.

[61] S. Wu, P.A. Flach, Scored and weighted AUC metrics for classifier evaluation and selection, in: Proceedings of the Second Workshop on ROC Analysis in Machine Learning.

[62] K.H. Zou, Receiver operating characteristic (ROC) literature research, `http://splweb.bwh.harvard.edu:8000/pages/ppl/zou/roc.html`, 2002.