Intelligent Data Analysis 17 (2013) 531–558
DOI 10.3233/IDA-130592
IOS Press

ROC analysis of classifiers in machine learning: A survey

Matjaž Majnik* and Zoran Bosnić
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia

Abstract. The use of ROC (Receiver Operating Characteristics) analysis as a tool for evaluating the performance of classification models in machine learning has been increasing in the last decade. Among the most notable advances in this area are the extension of two-class ROC analysis to the multi-class case as well as the employment of ROC analysis in cost-sensitive learning. Methods now exist which take instance-varying costs into account. The purpose of our paper is to present a survey of this field with the aim of gathering 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

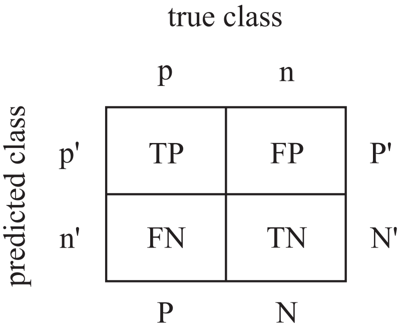
1. Introduction

Receiver operating characteristics (ROC) analysis is a methodology for evaluating, comparing and selecting classifiers on the basis of their predicting 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 [18,34]. Other fields to which ROC analysis has been introduced include psychophysics [34], medicine (various medical imaging techniques for diagnostic purposes, including computed tomography, mammography, chest x-rays [53] and magnetic resonance imaging [45], and also diverse methods in epidemiology [38]) 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 [64].

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 [50]. In the present, these curves already represent one of the standard metrics for assessing machine learning algorithms. A detailed introduction to the use of ROC analysis in research (with stress on machine learning) may be found in [21].

ROC analysis is already used in various applications and research in the field is lively developing in many directions: the two-class ROC methodology has been generalized to handle multi-class problems, extensions of basic ROC curves and the AUC metric have been proposed, advanced alternatives to basic

*Corresponding author: Matjaž Majnik, Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, Ljubljana, Slovenia. Tel.: +386 1 4768459; Fax: +386 1 4768498; E-mail: matjaz.majnik@fri.uni-lj.si.



		true class		
		p	n	
predicted class	p'	TP	FP	P'
	n'	FN	TN	N'
		P	N	

Fig. 1. Structure of a contingency table for binary classification problems.

ROC graphs appeared etc. Since the topic has been studied by many researchers from distinct points of view, questions came up which led to several polemics. 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” denotes a curve on an “ROC graph” (chart). The paper is structured as follows. In Section 2, the basic features of ROC analysis are discussed. Section 3 shortly describes procedures of ROC curve construction for some well-known classifiers and Section 4 summarizes various applications of the ROC 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 ROC graphs, and Section 7 concludes the paper.

2. Basic definitions

ROC analysis in its original form is used to deal with two-class classification problems. In the following subsections, main tools of this methodology are presented. At the end of the section an explanatory example is provided.

2.1. Contingency table

Suppose 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 early medical applications, where the instances 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 predicted classes of given instances are not necessarily same as true classes, a matrix is used to keep a record of the number of 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 a 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 number of correct classifier decisions thus lies on the main diagonal of a contingency table, while other table elements represent a number of misclassifications.

A contingency table is a source for calculating further knowledge evaluation measures, including the *true positive rate (TPR)*, *false positive rate (FPR)*, *true negative rate (TNR)* and *false negative rate(FNR)*. They are defined as $TPR = \frac{TP}{P}$, $FPR = \frac{FP}{N}$, $TNR = \frac{TN}{N}$ and $FNR = \frac{FN}{P}$, respectively, where $P = TP + FN$ and $N = FP + TN$. As we may observe in Fig. 1, P is the number of all instances which are actually positive and N the number of all instances which are actually negative. In some fields of study, TPR is also called *sensitivity* or *recall* as well as the term *specificity* denotes TNR.

2.2. ROC graphs and ROC curves

An *ROC graph* for original two-class problems is defined as a two-dimensional plot which represents TPR (sensitivity) on y-axis in dependence of FPR (= 1-specificity) on x-axis. Performance of a particular classifier, represented by its sensitivity and specificity, is denoted as a single point on an ROC graph. There are some basic characteristic points on a graph of this type. 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 usually not a good choice, since it would never make a single correct classification of a positive instance neither. Its relative in the point (1,1) represents the opposite situation (TPR = 1, FPR = 1) as it classifies all instances as positive, thus also producing a possibly high number of false positives. The classifiers in (0,0) and (1,1) are called *default classifiers*. In (0,1) the perfect classifier is located (TPR = 1, FPR = 0). While it is not realistic to expect such performance from 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 ascending diagonal of an ROC graph have the same performance as random guessing. For such classifiers we say, that they have no information about the problem. Useful classifiers are located above the ascending 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 [29].

An *ROC curve* is a curve on an ROC graph with start point in (0,0) and end point in (1,1). Drawing procedure for this curve depends on the type of classifiers we want to evaluate. In view of the amount of returned information, classifiers may roughly be divided into three groups: discrete (predicting a class membership), scoring (predicting a class score) and probability estimating (predicting a class probability). The score is defined as posterior probability (not necessarily calibrated) of the positive class. We should note, that a class score offers more information than a class membership. Similarly, the amount of information contained in a class probability is higher than in a class score. Main reason for the use of scores is that good probability estimates are not always available, for example, in a case of small amount of learning data. The meaning of scores may be interpreted as follows: if a classifier returns scores for two instances where the score of the first instance is greater than the score of the second, this indicates that the first instance has a 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. Procedures of drawing ROC curves for classifiers of above-mentioned types are discussed in Sections 2.2.1 and 2.2.2. The construction of ROC curves for some concrete classifiers is further described in Section 3.

We should have in mind the important property of ROC curves – they measure the capability of classifiers to output good scores [21]. Analyzed classifiers thus do not have to produce exact probabilities,

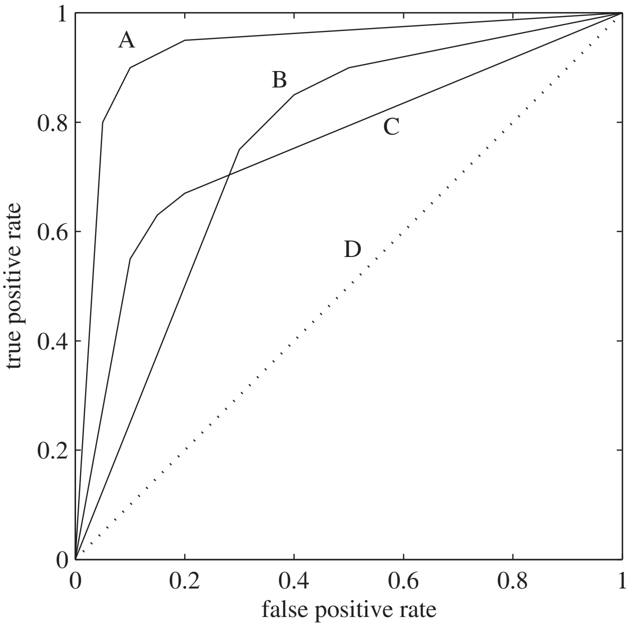


Fig. 2. An ROC graph with four ROC curves.

all they have to do is discriminate positive instances from negative ones. Another useful feature of ROC curves is that they remain unchanged when altering class distribution. Class distribution is the proportion of positive instances (left column in Fig. 1) to negative instances (right column in Fig. 1). An ROC curve is based on TPR and FPR values and since TPR and FPR are each calculated from values of one column, ROC curves are consequently independent of class distribution. The fact that ROC curves take into consideration sensitivity (i.e. TPR) and specificity (i.e. $TNR = 1 - \text{FPR}$) also represents an advantage of these curves over simpler evaluation measures, such as classification accuracy.

An example of an ROC graph with four different ROC curves each representing one classifier is given in Fig. 2. Classifier A is by far better than the other three classifiers. ROC curves of classifiers B and C cross – each of these two is superior to the other for some deployment contexts (i.e. combinations of class distribution and misclassification costs). Classifier D is of no use as its performance is no better than chance.

2.2.1. ROC curve construction for scoring and probability estimating classifiers

To construct an ROC curve of a scoring classifier we first have to sort instances according to their scores. We then draw the first point at (0,0) and select the instance having the highest score. We check whether the instance's true class is positive or negative – if it is positive, we move one unit up on the graph, if negative, one unit to the right. Horizontal and vertical unit sizes are inversely proportional to the number of negative and positive instances in the data set, respectively. We repeat this step by taking successive instances in a decreasing order and moving correspondingly over an ROC graph. The process terminates as the upper right corner in (1,1) is reached. All points, obtained by the process, are finally connected to form an ROC curve. The process may also be interpreted as applying different values of a *threshold* on scores. By varying this threshold one may obtain different (FPR,TPR) points, what may be seen as drawing an ROC curve. The procedure for probability estimating classifiers is exactly the same.

2.2.2. *ROC curve construction for discrete classifiers*

Since a discrete classifier is represented by only one point on an ROC graph, one may construct a very approximate ROC curve by connecting this point to the points of both default classifiers. However, a much 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 scoring classifiers.

2.3. *AUC measure*

We may want to compare more than two ROC curves. If the number gets high, visual comparison of these curves may become a non-trivial task. This is especially true in the case that many of them intersect (meaning that the underlying classifiers do not dominate each other). To this end, another measure of classification model performance has been introduced in ROC analysis: *Area Under the ROC Curve (AUC)*. The purpose of this measure is to summarize individual ROC curves in the form of numerical information. Comparison of the quality of classifiers thus reduces to comparison of numerical values.

Statistical meaning of the AUC measure is the following: AUC of a classifier is equivalent to the probability that the classifier will evaluate a randomly chosen positive instance as better than a randomly chosen negative instance [21]. This statistical property is often referred to as the probabilistic form of the AUC measure. It originates from signal detection theory and was introduced to machine learning community mainly through the use of ROC analysis in radiology. In [34] an experiment employing two-alternative forced choice (2AFC) technique (commonly used in psychophysics) was performed. As a result, the meaning of the AUC measure was determined to be the probability of correctly distinguishing a random pair of one normal and one abnormal sample in a 2AFC task.

AUC is related to other well-known measures. It is equivalent to the Wilcoxon statistic and to the Mann-Whitney statistic [3]. Further, the AUC is related to the Gini index [9]. In [36] the relation between statistical properties of the AUC and those of the Wilcoxon statistic are discussed in detail.

Value of the AUC measure may be calculated using the formula below:

$$AUC = \frac{\sum_{\text{over all pairs}} \mathit{if}\,(\mathit{difference} \geqslant 0; 1; 0)}{\mathit{number\ of\ all\ pairs}} \qquad (1)$$

where the sum passes over all pairs of one positive and one negative instance. Value of the variable *difference* is equal to the difference between the score of a positive and the score of a negative instance (in exactly this order) in an individual pair. Conditional statement is in the form

$\mathit{if}\,(\mathit{condition}; a; b)$

where a is the value returned when a condition is met and b the value returned when a condition is not met.

To be convinced that the value of AUC of some ROC curve may be calculated using Eq. (1), we may consider y and x axes of an ROC graph to be divided to P and N sections, respectively, where P is the number of all positive instances and N the number of all negative instances. An ROC graph may thus be seen as composed of $P \cdot N$ rectangles (i.e. P rows and N columns). If we then have a set of instances sorted according to their scores in decreasing manner, the value of AUC is calculated as follows. For every positive instance, we count the number of negative instances which have lower score than the chosen positive instance. We accumulate the sum. At the end, we divide the final sum with the number of all pairs of one positive and one negative instance ($= P \cdot N$), and finally we obtain the value of AUC.

The gain of one positive instance may thus be regarded as one row on an ROC graph. The scalar value of the AUC metric thus exactly corresponds to what may graphically be seen as the portion of the area of an ROC graph lying under an ROC curve.

Value of the AUC lies on the interval from 0 to 1. Since any useful classification model should lie above the ascending diagonal of an ROC graph, AUC of such models exceeds the value of 0.5.

In [8] the use of AUC as a performance measure for machine learning algorithms is investigated. AUC and overall accuracy measures are compared. It is shown that AUC has some convenient features: standard error decreasing when AUC and the number of test samples increase; 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.

While there are several possible measures that allow users to measure association between sensitivity and specificity (various information measures, such as mutual information gain etc.), AUC additionally provides a geometrical interpretation of the ROC graph. As there is no general rule specifying which measure has advantages or disadvantages in particular problem domains and using particular models, it is up to a user to select such a measure which has a required interpretation.

2.4. *An example*

In this subsection we present an illustrative example of the ROC drawing process and AUC calculation. Suppose the following set of instances is given:

(0.95p 0.71p 0.70n 0.64p 0.61n 0.48n 0.47p 0.35n)

It contains 8 instances represented by their scores (predicted by a classifier) and true classes. The letter ‘p’ stands for the positive class while the letter ‘n’ stands for the negative class. The data set is balanced, containing 4 positive and 4 negative instances what results in the fact that the horizontal and vertical unit sizes are both equal 0.25. The starting point is in (0,0). In our example, applying the threshold between the scores of instances 1 and 2 yields a situation with 1 true positive and 0 false positive classifications, which we denote as a point (0,0.25) on an ROC graph, i.e. we move one unit up. In next step we fix the threshold between the instances 2 and 3. The number of true positives increases to 2 while there are still no false positives, what results in moving up for another unit to a point (0,0.50). After fixing the threshold between the scores of instances 3 and 4, we obtain the point (0.25,0.50) on the graph, i.e. we move one unit to the right, since the instance 3 whose true class is negative has higher score than some positive instances. In a similar way we move up and right over the graph until we reach the point (1,1). As a result, the ROC curve shown in Fig. 3 is drawn in a step-by-step manner. For simplicity, we presumed equal misclassification costs with their value being 1. In the case of unequal misclassification costs or unbalanced class distribution ROC space features change, e.g. expected costs of classifiers on the ascending diagonal are different.

We may calculate the value of the AUC measure for such a set as follows. As already mentioned, the set contains 4 positive and 4 negative instances. Consequently, there exist $4 \cdot 4 = 16$ possible pairs of scores of one positive and one negative instance and the corresponding ROC graph in Fig. 3 is divided to 16 squares of equal size. The final value of the AUC measure for the set above is obtained in the following manner.

1. The difference between the score of instance 1 (p) and the score of any negative instance is always positive, as the former is higher than any of the latter. The partial sum thus equals 4. Similar is true for all pairs in which the score of instance 2 (p) appears. The partial sum added is again 4 and the total sum increases to 8. If we divide the ROC graph in Fig. 3 into 4 rows and 4 columns, we may see the result in the lower two rows of the graph (i.e., the rows are “full”).

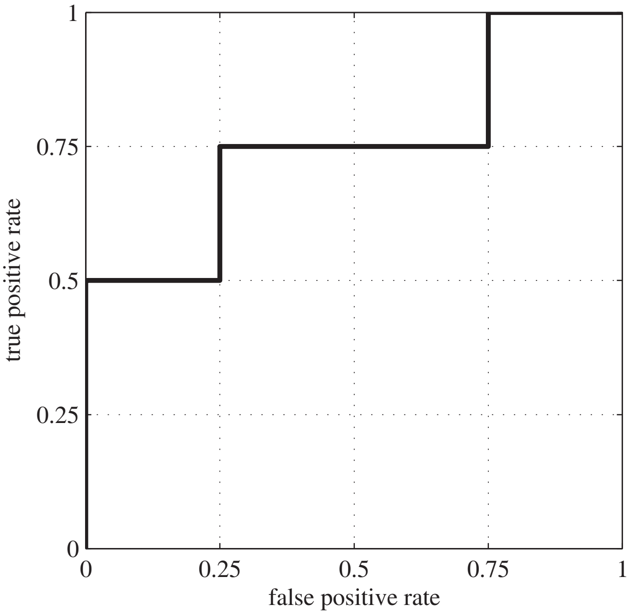


Fig. 3. The ROC curve for a sample set of instances.

2. The difference between the score of instance 4 (p) and scores of individual negative instances is positive for all negative instances except the instance 3 (n) as the latter has higher score than instance 4 (p). The partial sum added equals 3 and the total sum increases to 11. Here we "lose" one square on the graph (the third row from the bottom).
3. The difference between the score of instance 7 (p) and scores of individual negative instances is only positive for instance 8 (n) and negative for all other negative instances. The partial sum added thus equals 1 and the total sum increases to 12. Here we "lose" another three squares on the graph (the uppermost row).
4. We divide the total sum (i.e. the sum in the numerator of the fraction in formula 1) with the number of all pairs (which in our case equals 16). We finally obtain $AUC = 0.75$.

3. Construction of ROC curves for commonly used classifiers

In this section, the procedure of generating ROC curves for some typical classifiers is described briefly. When planning to draw an ROC curve, the main issue with the classifiers is how to obtain 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 a 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 an ROC curve by applying one of the following two equivalent processes: (1) we may traverse the scores as described in the previous section, or (2) alternatively, alter the threshold on some parameter (not necessarily a score) of a classifier explicitly, in a systematic manner. For every threshold value, the contingency matrix is recalculated, TPR and FPR values are recomputed and the corresponding point is added to an ROC graph. 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 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. A threshold may then be employed on these score values.
- Decision Tree (DT) in its basic form only returns a class label (prediction of a class membership) 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 [46], a work on probability estimates in decision trees. The threshold is set on the proportion of positive instances in a 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 a network, as presented in [61]. In the latter work, methods of ROC curves construction for ANN classifiers are analyzed and the curves generated by the described method are claimed to 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 the 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 considered neighbors. In [60] k is, for example, varied from 1 to 200.
- Support Vector Machine (SVM) outputs scores by default. The threshold may simply be set on the decision function [55].

While instance statistics are a handy basis for score acquisition, they are not the only option. Discrete classifiers may also be transformed to scoring ones by classifier aggregation or a combination of scoring and voting [21]. A review of methods for generating ROC curves for various classifiers may also be found in [60].

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, presentation, comparison, selection, 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 Section 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, comparison, selection and presentation

ROC curves facilitate the task of *model evaluation*. Classifiers may be evaluated by merely observing their location on an ROC graph. 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 the AUC measure may be employed. The use of AUC as a performance measure for machine learning algorithms is advocated in [8] and has already been discussed in Section 2.

In view of model evaluation, ROC space has even be used for the redefinition of other machine learning metrics. The theory which would define the use of different metrics for various goals is discussed in [31]. The authors believe 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, they use *isometrics* (i.e. sets of points for which a metric has the same value) and the 2D ROC space. Isometrics have originally been presented in [47,48] as *iso-performance lines*. These auxiliary lines have a characteristic that the expected cost of all classifiers located on an individual line is equal. They are used to separate the performance of a classification model from the specific class and cost distributions. They are also robust to imprecise class distributions and misclassification costs. The 2D ROC space, on the other hand, is derived from the 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 the operating characteristics. The effective skew landscape of a metric (i.e. the slope of its isometric at any point in 2D ROC space) is found to be its defining characteristic. As a result, a simplification of the F-measure and a version of the Gini splitting criterion invariant to the class and cost distributions have been derived.

Another task to which ROC analysis may be applied is *model comparison*. Given two discrete classifiers, the former may be evaluated as better if it is situated more northern or western (or both) than the latter. When comparing two scoring (or probability estimating) classifiers, the former classifier may be determined as better only if its ROC curve is strictly above the ROC curve of the latter. Instead of visually comparing ROC curves we may rather compare corresponding values of the AUC measure. In the case that ROC curves cross we should be aware that the classifier with the highest AUC value will not necessarily be the optimal one for some specific combination of class distribution and misclassification costs.

Employing ROC analysis for *model selection* then enables the selection of an optimal model (after the 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 procedure for selecting the potentially optimal classifier is described in [28]. Class distribution and error costs are combined to determine the slope of an auxiliary line positioned on an arbitrary location on an ROC graph. Afterwards, the line is shifted in the direction of the upper-left corner of an ROC graph, until it touches the ROC convex hull (ROCCH, defined in Subsection 5.1) in one single point (i.e. the line becomes a tangent to the ROCCH). This point represents the optimal classifier for given operating characteristics. Suitability of the ROCCH for tasks of model selection has been tested in [5] where the performance of various classifiers has been compared through the perspective of misclassification costs. The ROCCH has been recognized as a robust alternative to the AUC measure.

Since ROC analysis is based on graphical principles it is an appropriate tool for *model presentation*. As such, it visualizes performance and facilitates a general notion of classification models. It may be combined with other visualization techniques, several of which are listed in Section 6, e.g., AUC may be put in the role of the y-axis of a learning curve plot [7].

4.2. Model construction and model combination

Learning algorithms may be adjusted to aim at constructing classifiers with good ROC curves instead of optimizing other criteria. ROC analysis may thus be employed for *model construction*. The idea of *model combination*, on the other hand, 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. As we may combine individual models to construct some other models, both activities may be highly overlapping.

A common principle which may be employed for model combination is ROCCH [48]. 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. A theorem in [48] 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. Such a hybrid classifier may be optimal for any target conditions, including imprecise class distributions and misclassification costs. ROCCH only includes classifiers that are optimal for some *(FP,TP)* pair and discards all other sub-optimal models what contributes to its spacial efficiency.

Another approach to model combination is given in [30]. 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. The goal of the *SwapOne* algorithm is to enlarge the AUC of a probability estimating 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 probability estimating classifier by detecting a section of an ROC curve that is under its convex hull. Afterwards, the ranks of the instances in that section are inverted. The algorithm is applicable to any model issuing scores.

In [24], the procedure implementing the above-mentioned ideas for a construction 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 presented as 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 [6]. 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 may carry additional information apart from their coordinates on an ROC graph. Classifiers may often be divided into sub-components, each of them being 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 the positive class. Predictions (positive and negative) in the leaves are then systematically varied, yielding new (biased) decision trees. The method may only improve the ROCCH (as in the worst case the original classifier will dominate all the derived ones) while at the same time being computationally cheap.

Another technique optimizing ROCCH has been presented in [52], using inductive logic programming (ILP) as the main tool. Background knowledge used to construct 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 also be used to experimentally find more suitable decision thresholds for the naive Bayes classifier [39]. The authors treat the threshold as an additional parameter of a model which should be learned from the given data. Posterior estimates in the naive Bayes classifier are scores, not real probabilities, and in the case that independence assumptions do not hold, these probability estimates 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 (discussed in Section 2.2.1). The one for multi-class problems is implemented using weights (one per class) which are determined by greedy hill-climbing 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 a computationally intractable algorithm. However, both algorithms are capable of considering non-uniform misclassification costs and are applicable to the recalibration of all learning methods which return class scores as a result.

4.3. Polemics in the research community

ROC analysis has been an interesting research topic during last decades and has been studied by many researchers from various perspectives. As a result, some disagreements like the following emerged. In [58], 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 these issues may be found in [23] and states that the assertions in [58] 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.

5. Improvements of basic ROC techniques

Since the basic ROC analysis is not able to answer satisfactorily all questions pertaining to classifiers' performance, some extensions, approximations and other improvements have emerged over time. Some have been found to be useful while the practicability of others has not been generally acknowledged. 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 the ROC analysis to the multi-class case, variants of original ROC curves and the AUC metric taking scores and instance varying costs into account, and improvements that help to increase efficiency.

5.1. ROC convex hull

The *ROC convex hull (ROCCH)* is a line connecting classifiers which may be optimal for some operating characteristics. This line is convex and no classifier may exist in the part of an ROC graph above it. Given a set of classifiers, their ROCCH may 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 a graph in a way that the slope is always

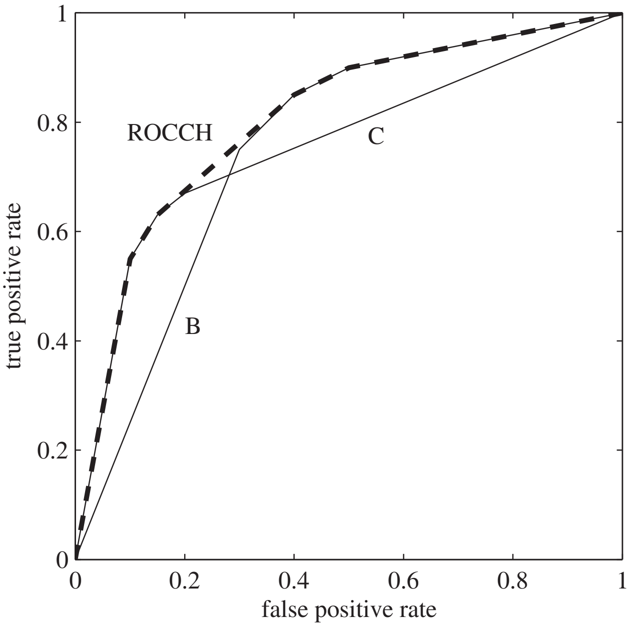


Fig. 4. An ROC convex hull.

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 4 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].

A high level of similarity and the principal differences between an ROC convex hull and an 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 the 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 an ROC convex hull is feasible since given any two classifiers on an ROC graph, 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 instance is processed.

5.2. Robustness and reliability

ROCCH is a more robust tool for measuring classifier performance than the AUC [49]. The AUC is a single-number measure and, as already mentioned, is not appropriate for an 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 such a single-number measure will not be the one with minimum cost for specific target conditions. Without providing these conditions no single-number metric may be absolutely trustworthy. In the case of ROCCH, ranges of operating conditions where some particular classifier is optimal may be specified by expressing them in

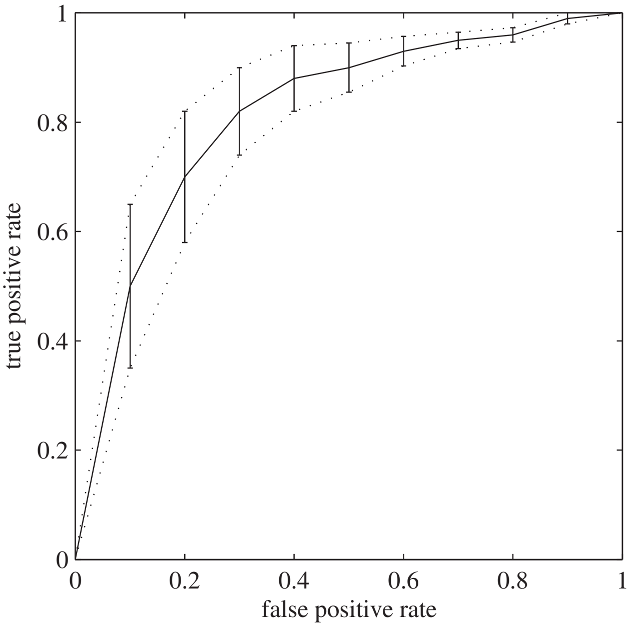


Fig. 5. Vertical averaging.

the form of slopes of tangents to the ROCCH. As a result, a table of such regionally optimal classifiers is obtained. The approach has been supported by the experimental study of classifiers, such as decision tree, naive Bayes and k-nearest neighbor, applied to several UCI repository data sets.

Another, more statistically oriented approach, is the introduction of one-dimensional confidence intervals for ROC curves which may be constructed by *Vertical Averaging (VA)* [49]. 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). Finally, at these points of the curve, vertical confidence intervals for the mean of TPR values are calculated. While the procedure is simple and such confidence intervals are suitable for maximizing the TPR given some FPR value, the authors state that it may not be fully appropriate for the task of minimum expected cost evaluation. A noticeable disadvantage is the fact that the FPR may generally not be controlled by a researcher [20]. An example of the VA method application is given in Fig. 5 where confidence intervals are presented by vertical lines. Two dotted lines connecting all upper respectively lower ends of confidence intervals represent a confidence band described in the last paragraph of this subsection.

To overcome the potential disadvantage above, *Threshold Averaging (TA)* has been introduced [20]. Instead of fixing the 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 the 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 the VA method. In addition, horizontal intervals may now be produced as well. On the other hand, TA makes additional requirement that classifiers have to be able to issue scores. Since scores should not be directly compared across different classifiers, care should be taken

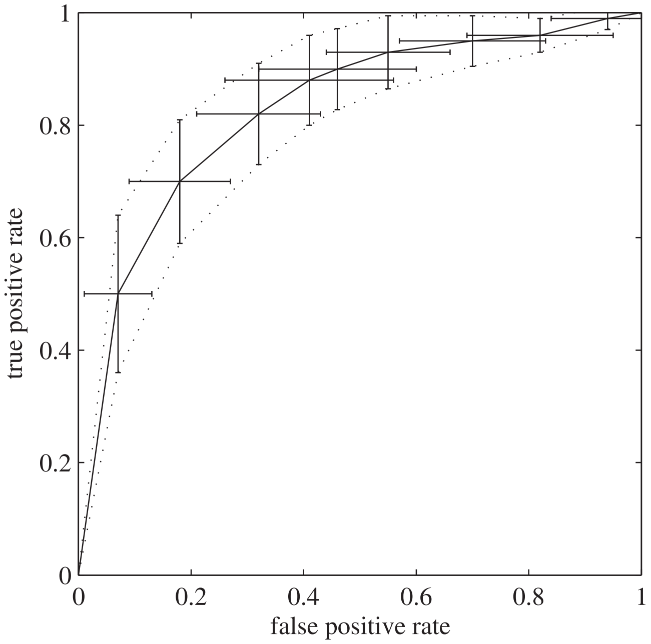


Fig. 6. Threshold averaging.

when using this technique on their ROC curves, as the resulting averaged curve may be misleading. An example of the TA may be seen in Fig. 6. As before, dotted lines denote a produced confidence band.

With the aim of obtaining a more robust statistical technique, methods for generating confidence bands for ROC curves are discussed in [42]. As ROC analysis has long been used by medical researchers, the authors assume that it may be beneficial to introduce techniques from the medical field and evaluate their fitness on various machine learning tasks. The authors then 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 an 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 that has been used 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. 7. One disadvantage of confidence bands may be an inappropriate effect of these bands on the AUC in the case that probabilities are non-uniformly distributed from 1 to 0 [25].

5.3. Generalizations to multi-class problems

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

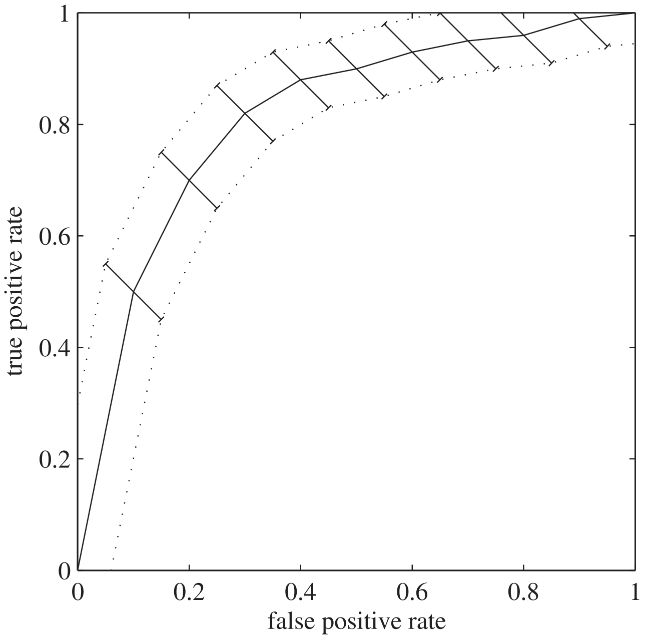


Fig. 7. Fixed-width bands.

As a result, generalizations to multi-class ROC analysis have been developed. If a multi-class problem consists of k classes, there are now $k \cdot k$ possible classifications, considering k options for the true class and k options for the predicted class. This calls for different methods of visualization as simple two-dimensional graphs cannot represent such complex information. In case the number of classes gets high, the computational complexity may also become important. These issues are discussed in more depth in [41].

ROC analysis has been extended to three-class decision problems in [44]. In this way, an ROC surface may 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 a 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 a two-class model. Domain experts in many fields of practice (e.g. physicians) make decisions between two classes without stating probabilities. A 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 a 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 may thus become questionable.

An extension of the AUC measure in the form of the VUS has also been presented in [27]. The default classifiers, minimum and maximum VUS, and the equations are derived. The procedure of computing polytopes (multi-dimensional geometric objects with flat sides – a polygon is a polytope in two dimensions) for a set of classifiers is presented. It consists of forming constraints (linear inequations) which are then solved by the Hyperpolyhedron Search Algorithm (HSA). Volume of the hyperpolyhedron is computed using QHull algorithm [4]. In this way, VUS of any classification model for any number of classes may be gained. A disadvantage of this technique is its inefficiency for a higher number of classes. It should be mentioned that the authors use different representation of an ROC graph by plotting FNR on the y-axis against FPR on the x-axis. In this case, the goal becomes the minimization of the area under the ROC curve what is essentially equivalent to the maximization of the area *above* the ROC curve (AAC). However, the authors are consistent with the terminology and refer to the AAC as AUC.

It is demonstrated in [14] how a multi-class classifier can be directly optimized by maximizing the VUS. This is accomplished in two steps. Firstly, the discrete U-statistic (which is equivalent to the VUS) is approximated in a continuous way. Secondly, the resulting approximation is maximized by the gradient ascent algorithm. The drawback of this approach lies in its exponential time complexity.

Another approach to an extension of the AUC measure to the multi-class case is presented in [46]. 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). Values of the AUC 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.

Further multi-class generalization of the AUC is available in [35]. 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 the new measure is, that its value may become substantially higher for small improvements in pairwise separability. The extension is invariant to monotonic transformations of estimated probabilities. It should be noted that the authors, in view of the representation adopted in our paper, use swapped axes on an ROC graph.

It has been demonstrated that principles of the ROCCH extend to multi-class problems and multi-dimensional convex hulls [51]. 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 the minimum cost and follows from the features of convex sets that represent the base of linear programming algorithms. As a result, if classifiers for n classes are considered to be points with coordinates assigned by their $n \cdot (n-1)$ misclassification rates, then optimal classifiers are shown to lie on the convex hull of these points. This holds for an arbitrary number of classes.

Considering costs, the ROC analysis has also been extended through the perspective of an optimization problem [19]. The latter has been used to define an ROC surface for the n-class decision problem with the aim of minimizing $n \cdot (n-1)$ misclassification rates. The final solution to the problem is finding the optimal trade-off surface between different types of misclassifications, known as a *Pareto front*. The 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.

Cost-sensitive optimization is further discussed in [40] in view of a lack of methods for handling multi-class ROC analysis for large numbers of classes. Algorithm in [19], for instance, has only been 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 larger number of classes.

Table 1
Sets of instances presented by predicted scores and true classes, with calculated values of the 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

5.4. *Considering scores*

AUC ignores scores (i.e. posterior probabilities of the positive class) and considers only ranks of scores (i.e. an order). Since a part of information is ignored, this may lead to suboptimal results, for instance, overfitting to a test set when selecting classifiers with high values of the AUC. A different option of how to evaluate a success of learning is to replace the AUC by using some other measure, e.g. information gain, Brier score or LogLoss. However, these alternatives ignore the order and will not be further discussed in this work. Another 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. However, we should be aware of the advantage of the original AUC, which is its independence of any distribution assumptions.

Like with the basic AUC, values of these variants can be calculated directly without the explicit construction of corresponding ROC curve variants. 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 [56]. To calculate a 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 the value of the *difference function* (called also the *modifier function* in [56]) over all pairs. The difference function handles a difference of scores of two instances in an individual pair. Afterwards, we divide the intermediate result with the number of all possible pairs. The generalized form of the AUC may thus be interpreted as a mean value of the difference function for a set of instances. It is obvious that the AUC and its variants only diverge in their difference functions, i.e. the way how a score difference is dealt with. Difference function for the basic AUC is the step function.

The proposed four variants of the AUC metric are briefly summarized in the following. Table 1 shows an example of computed values of these variants for 15 different score sets in the classification of 6 instances. The second table column provides the predicted scores and the instances' true classes. The following table columns illustrate how various AUC variants differently evaluate the quality of classifiers. The AUC variants are:

1. In [25] the first variant, called *probabilistic area under the ROC curve (probAUC)*, is presented. Its difference function is defined using a) uniform distribution, and b) 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 the AUC, the opposite may happen as well (as seen for the sets no. 11, 12 and 14 in Table 1). Since such a performance can no longer be visualized by basic ROC curves, an approach for drawing probROC curves (with the area under the curve equal to the *probAUC*) is given. The 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 ROC curves, probROC curves are also constructed in the ROC space. When constructing probROC curves, probabilities denote curve intervals. These curves have usually smoother shapes, main distinctions may especially be noticed in cases when the original ROC curve is unreliable (few instances, small differences between scores). For larger instance sets, an probROC curve behaves similar to a basic ROC curve.
2. The second variant is *scored area under the ROC curve (scorAUC)* [62,63]. Its value is equal to the area under the *scorROC* curve. A scorROC curve illustrates how quickly AUC deteriorates if positive scores are decreased, i.e. how sensitive is a classifier to a shift of score values, and *scorAUC* accumulates this information into a numerical metric. A scorROC curve is constructed in a space not much alike to the ROC space. In case of scorROC graphs, the x-axis denotes a value of the parameter τ, indicating a decrease degree of positive scores, and y-axis denotes an AUC value of such a modified set. The difference function for the *scorAUC* is the step function, weighted by a difference of scores. The *scorAUC* always underestimates a value of the AUC metric – their values are equal only when a classifier issues perfect scores for a set of instances, i.e. predicts 1 for every positive instance and 0 for every negative instance (as seen for the set no. 1 in Table 1).
3. Third variant, named *softened area under the ROC curve (sondAUC)*, has been proposed in [37]. The *sondAUC* is actually a generalized version of the *scorAUC*. Difference functions are the same with one exception – in case of *sondAUC*, a difference between scores is raised to a power of some chosen parameter q, which is not a case with the *scorAUC*. The purpose of the exponent q is to regulate sensitivity and robustness to ranking alterations, what gives greater flexibility to a user when selecting classifiers. By increasing q, the *sondAUC* usually becomes more sensitive and less robust to variations in ranking order. If value of the parameter q is set to 0, the *sondAUC* becomes equivalent to the basic AUC metric.
4. The last variant is called the *soft AUC (softAUC)* [10]. Besides considering scores, the main motivation for defining such a metric were features like continuity and differentiability. The developed variant *softAUC* possesses both of them. The difference function of the *softAUC* is a sigmoidal function (more precisely, a logistic function) with a 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 AUC variants are softer than the basic AUC as they aim at smoothing the difference function of the AUC. As such, they are intuitively expected to be more robust for small data sets. Since all four variants employ both, scores and ranks, any of them may possibly also be used as a statistic for testing the diversity of two samples (similar as the Wilcoxon-Mann-Whitney statistic). Difference functions of *probAUC*, *scorAUC* and *softAUC* variants are visualized in [56].

5.4.1. An example

In Table 1 we provide an artificial example, intended to illustrate behaviors of different AUC variants. The table contains 15 example data sets comprised of six instances. Each instance is represented by its (not necessarily calibrated) probability of belonging to the positive class (denoted with a real number) as predicted by a classifier (i.e. a score), and a sign denoting the true class (+ denotes the positive class and – denotes the negative class). For each set of instances, values of the basic AUC and its four variants are calculated. A couple of 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 a calculation of the *sondAUC* and *softAUC*, respectively. The sets of instances are ordered from intuitively the best to intuitively the worst (in a decreasing order). The table serves as an illustration of how basic AUC and its variants evaluate some possible sets of instances. The most obvious characteristics are exposed as follows.

- Sets of instances 1a-1e represent an ideal case in which a classifier issues a score 1 for every positive instance and a score 0 for every negative instance. Taking a look on the calculations for 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 further sets of instances, we have chosen a fixed value of $\beta = 7$, as at this value the *softAUC* behaves similar to the basic AUC, yet sufficiently distinct to deserve its own focus of observation.
- In sets of instances 2–7, all instances 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 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 positive from negative instances. Nevertheless, all AUC variants evaluate the set 5 as more preferable than the set 4. Here, calibration predominates over a difference between positive and negative scores. Considering the set 3, values of all AUC variants are still rather lower than for the set 5, even though the score margin increased to remarkable 0.36. Of all variants, it seems that the *scorAUC* relies on calibration the most.
- In the set of instances 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 improper calibration. The basic AUC metric only considers ranking, while its variants consider calibration, as well. On account of a narrow score margin, bad calibration may even outweigh a correct ranking order in an evaluation done by the AUC variants. Examples of this phenomenon are given below (see the description of the 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 may present the classifier's quality more credibly than its variants.
- Another similar example follows from comparing the sets 7 and 8. A ranker with an ideal performance may be seen in the set 7. Although the classifier in the set 8 only misclassifies two instances with a small difference in scores, it does not rank all instances perfectly. Nevertheless, the better calibrated classifier for the set 8 has been evaluated with higher grades by all AUC variants, most noticeably by the *scorAUC*.

- The sets 9a-9e demonstrate an influence of the parameter q on 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 a 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 may arise (occurs always when some instances are improperly ranked). For example, a value of q may be set to $\frac{1}{3}$ or $\frac{1}{5}$, but not to $\frac{1}{2}$. In other sets of instances, we have decided to use a fixed value $q = \frac{1}{7}$ as it seems to offer an appropriate balance between robustness and sensitivity.
- The sets 9 and 12 expose the main disadvantage of the AUC measure which the derived variants strive to overcome – the unreliability for (small) sets of instances where differences between predicted scores are negligible. As we can see, a tiny variation in score values results in an enormous change of the AUC value (since the 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 may be more advisable to rely on evaluation results of the AUC variants.
- The set 13 represents a case when the observed classifier is of no use – it issues a score 1 for every instance it sees and obviously does not separate positive instances from negative ones. A classifier does not differentiate between positive and negative instances in the set 14, neither. It gives an impression that score values are issued randomly. Considering the set 14 which depicts a classifier of essentially inferior quality to the ideal rankers of 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 set 11 are higher than for the set 6. This kind of behavior where the correct ranking is outweighed by bad calibration may also be noticed for the *softAUC* metric while comparing the sets 10 and 6. Finally, the set 15 is the worst possible, though an ideal case (the set 1) is retrieved trivially by inverting the classifier's decisions (i.e. interpreting 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 [56], where their performance is claimed to be questionable. The authors believe that none of the variants should surpass the basic AUC, at least when applying them to evaluation and selection of classifiers. The variants may be biased with the variance possible in either direction. 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 limitations of ROC graphs is their inability of handling problems where misclassification costs vary from one instance to another of a 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.

An *ROCIV graph*, a transformation of the original ROC graph, is an approach to deal with instance-varying costs [22]. An intuitive interpretation of an ROCIV curve is that the axes are scaled by example costs within each class. In such a manner, the y-axis represents a true positive benefit, while the x-axis represents a false positive cost. An ROCIV curve is constructed similarly as an 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 a probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance given that each is chosen in a 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 a feature of ROC curves that they are cost-invariant, and represents a potential drawback. An 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*

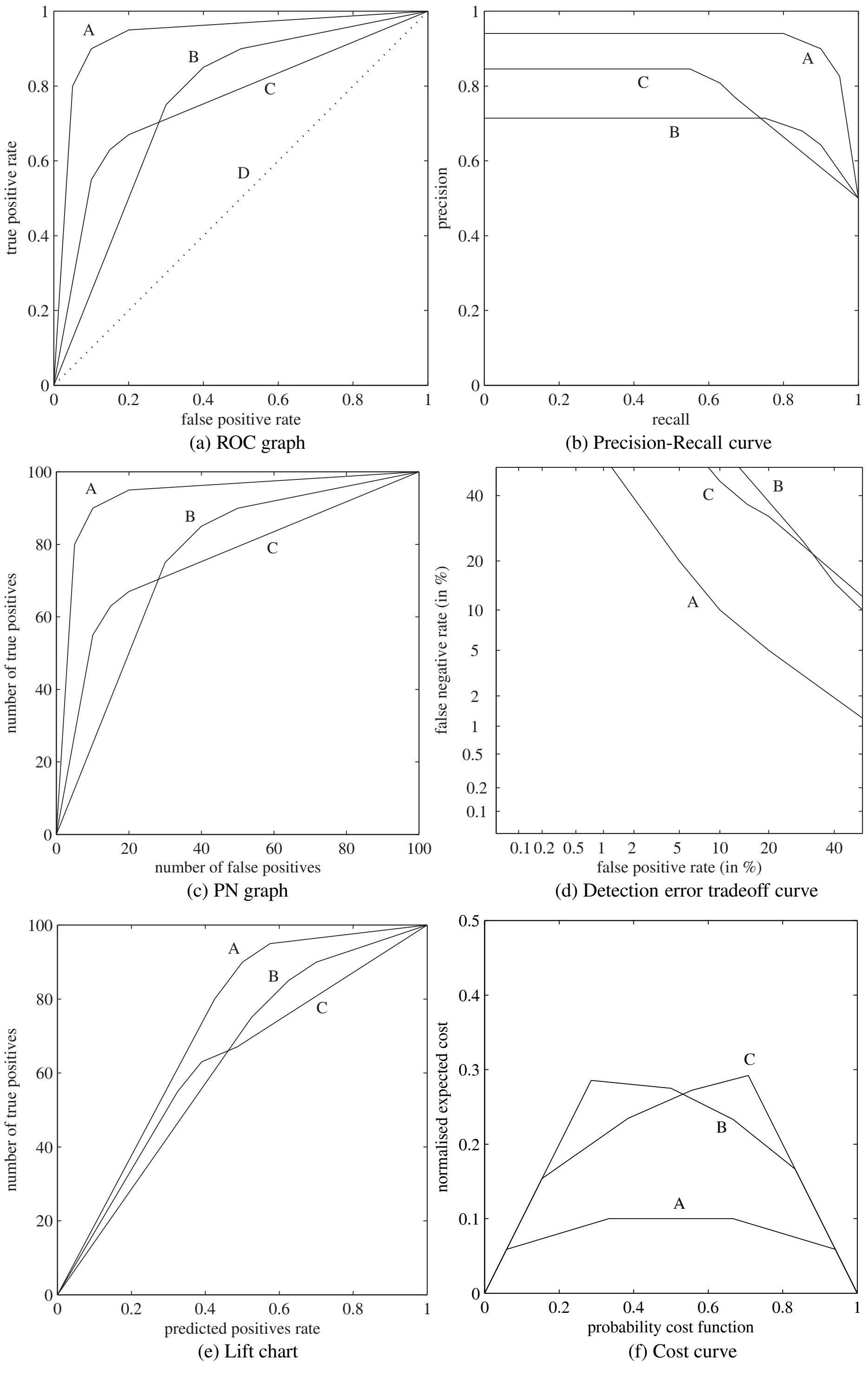
Repetitive computations of the AUC may be relatively time-consuming. Since computations of this kind are important in many techniques, such as, for example, methods for direct optimization of the AUC, aspirations for more effective algorithms emerged.

In [10], a polynomial approximation of the 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: the step function is replaced by a general form of a polynomial. A degree of a polynomial should be chosen deliberately, optimizing the relation between accuracy and performance. The approximation is believed to be more accurate than sampling and at the same time being computable in one pass over a database (thus having linear time-complexity).

6. Alternatives to tools of the ROC analysis

In this section, some other techniques which may be used instead of ROC graphs are listed and discussed. Some of them are strongly related to 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 [43]. It is based on principles of ROC curves, nonetheless, in the DET space, the y-axis denotes FNR instead of TPR (as shown in Fig. 8d). In this way error rates are plotted on both axes which have normal deviate scale. Such a scale makes curves in the DET space nearly straight lines. If classifiers perform well enough, the plot may be limited to the lower-left quadrant. Both modifications spread 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 classification models by explicitly showing cost values for which some individual models are better (shown in Fig. 9a). The triangular form represents a 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 classifiers' performance which makes it possible to employ any available information about 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 benefit if some information about a ratio of misclassification costs, including an interval of possible ratio values or the most probable ratio value, is available.



(a) ROC graph

(b) Precision-Recall curve

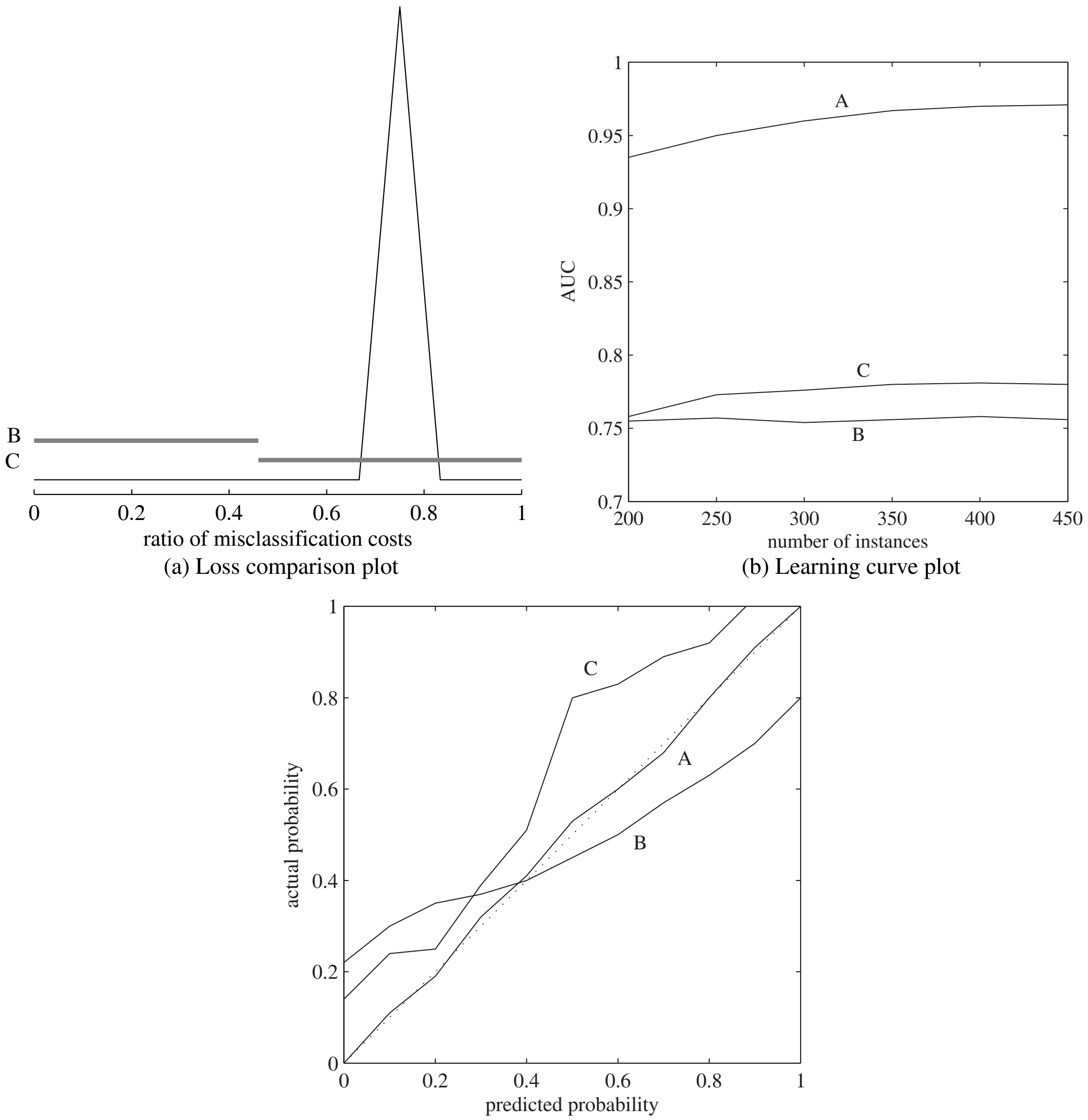
(c) PN graph

(d) Detection error tradeoff curve

(e) Lift chart

(f) Cost curve

Fig. 8. Alternatives to ROC graphs.



(a) Loss comparison plot

(b) Learning curve plot

(c) Calibration chart

Fig. 9. Alternatives to ROC graphs which are plotted using additional data.

3. *Lift chart* is a technique similar to ROC graphs and is commonly used in some branches of data mining [59]. It is defined by the TP on its y-axis and the size of a subset (proportion) on its x-axis (as shown in Fig. 8e) what makes it sensitive to 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 may be used for its recalibration [11]. Calibration chart plots an actual probability on the y-axis against a predicted probability on the x-axis (as shown in Fig. 9c) and is sensitive to changes in a class distribution. It cannot reflect a sole quality of classification but is able to recognize a classifier's bias, which can be used as a performance measure [57].

5. *Learning curve plot* is another graphical technique for visualization of classifier's performance. The x-axis measures a size of a training set while the y-axis indicates performance of a classification model (as shown in Fig. 9b). As such, a learning curve depicts how the amount of learning depends on a number of instances in a training set and shows when the continuation of learning has no further effects. It may be used for comparing classifiers built on different sizes of a data set. The AUC measure may be applied on the y-axis as a measure of classifier performance [12,54]. Learning curve with the AUC on the y-axis is further studied in [7], 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 [17] and seems to be one of the most promising among various alternatives to ROC graphs [15,16]. *Cost curves* do not represent expected costs through slopes of iso-performance lines but rather in an explicit form [17]. As such, they offer an alternative for visualization. Information implicitly contained in ROC graphs is presented explicitly with cost curves. 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. 8f). 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. The *lower envelope* in cost space is equivalent to the ROCCH in the 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 a specific operating characteristic, i.e. the one minimizing cost.
7. *PN graph* is a graphical technique in close relation to ROC graphs (the acronym PN is derived from the titles of both axes, as the y and x are sometimes labeled as 'P' denoting covered positive instances and 'N' denoting covered negative instances, respectively). It plots the TP on the y-axis against the FP on the x-axis (as shown in Fig. 8c) 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 [32,33].
8. *Precision-recall curve (PR curve)* is an alternative which is often used in information retrieval and can be beneficial in cases when a class distribution is 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. 8b). An ideal classifier is located in the upper-right corner of the PR space. PR and ROC curves are compared and studied in [13], 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 a 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. In the same paper, a PR-space equivalent to the ROCCH is presented as *achievable PR curve* and an algorithm for computing such achievable PR curves is given. The algorithm is based on computing the ROCCH in the ROC space and transforming it to the 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 as straightforward as in the ROC space, since values, as already mentioned, should not be linearly interpolated.

6.1. An example

We provide an illustration in which we compare three classifiers trained on an example 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 the alternatives is revealed. It should be noted, however, that quite a different picture may be seen when class imbalance gets high or the 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 may thus be helpful in better understanding of a given problem. They are applicable to all application areas of the ROC analysis (see Section 4), e.g. model evaluation and presentation.

Some of the alternative techniques are computed on the base of same data as ROC curves, i.e. a contingency table, and may be transformed easily from one representation to another. Such techniques are shown in Fig. 8. On the other hand, a 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. 9. A transformation to some of them may only be accomplished if all the needed information is available. In all the representations, the performance of the classifiers A, B and C (same classifiers as in Figs 2 and 8a) is visualized in different ways.

To present a meaningful LC plot (Fig. 9a), 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 3. These three values determine the location of three key points on an LC plot. As FPs are more costly than FNs, the goal should be to stay on the left side of an ROC graph, near the ordinal axis. In this area, the classifier C is superior to the classifier B (A is not included in the 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. 9b reveal information which is not contained in a single ROC graph, since the latter assumes a fixed number of training instances. An 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. 9b, only three points are therefore actually related to our ROC graph example in Fig. 8a, namely those three which represent the AUC value for 200 instances. In a similar way, the calibration chart in Fig. 9c is based on data not contained in ROC graphs, but rather on additional information gained from a data set.

7. Conclusion

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

Important improvements of 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.

During future development of ROC analysis it would be, in the face of recent findings, beneficial to periodically compare tools of ROC analysis with other widely-used measures of classification performance and verify their advantages and disadvantages from the theoretical as well as experimental point of view, similarly as the authors in [26].

Next task might be trying to solve remaining issues in multi-class ROC analysis, especially finding ways of more intuitive visualization and more efficient calculations.

It would be useful to check whether an introduction of further extensions and improvements from other fields where ROC analysis is extensively used (especially medical decision making) may be advantageous in the area of machine learning. One option would be, for instance, to examine potential benefits of an introduction of time-dependent ROC analysis.

Furthermore, it would be advantageous to consider how in a balanced and theoretically well-founded manner take scores of instances into account. A new derivative of the AUC measure might be developed as a result which would eliminate most of deficiencies of the original AUC and its existing variants. This is also a part of our future work.

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 the ROC analysis in other machine learning areas (e.g. in regression), is the intended focus of our further work.

References

[1] N.M. Adams and D.J. Hand, Comparing classifiers when the misallocation costs are uncertain, *Pattern Recognition* **32**(7) (1999), 1139–1147.

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

[3] D. Bamber, The area above the ordinal dominance graph and the area below the receiver operating characteristic graph, *Journal of Mathematical Psychology* **12**(4) (1975), 387–415.

[4] C.B. Barber, D.P. Dobkin and H. Huhdanpaa, The quickhull algorithm for convex hulls, *ACM Transactions on Mathematical Software* **22**(4) (1996), 469–483.

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

[6] H. Blockeel and J. Struyf, Deriving biased classifiers for better ROC performance, *Informatica* **26**(1) (2002), 77–84.

[7] R.R. Bouckaert, Efficient AUC learning curve calculation. In *Proceedings of the Nineteenth Australian Joint Conference on Artificial Intelligence*, 2006, pp. 181–191.

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

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

[10] T. Calders and S. Jaroszewicz, Efficient AUC optimization for classification. In *Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2007, pp. 42–53.

[11] I. Cohen and M. Goldszmidt, Properties and benefits of calibrated classifiers. In *Proceedings of the Eigth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2004, pp. 125–136.

[12] M. Culver, D. Kun and S. Scott, Active learning to maximize area under the ROC curve. In *Proceedings of the Sixth International Conference on Data Mining*, 2006, pp. 149–158.

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

[14] 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*, 2007, pp. 878–885.

[15] C. Drummond and R. Holte, What ROC curves can't do (and cost curves can). In *Proceedings of the First Workshop ROC Analysis in Artificial Inteligence*, 2004, pp. 19–26.

[16] C. Drummond and R. Holte, Cost curves: An improved method for visualizing classifier performance, *Machine Learning* **65**(1) (2006), 95–130.

[17] C. Drummond and 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*, New York, NY, USA, 2000. ACM Press, pp. 198–207.

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

[19] R.M. Everson and J.E. Fieldsend, Multi-class ROC analysis from a multi-objective optimisation perspective, *Pattern Recognition Letters, special issue on ROC analysis* **27**(8) (2006), 918–927.

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

[21] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters, special issue on ROC analysis* **27**(8) (2006), 861–874.

[22] T. Fawcett, ROC graphs with instance-varying costs, *Pattern Recognition Letters, special issue on ROC analysis* **27**(8) (2006), 882–891.

[23] T. Fawcett and 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**(1) (2005), 33–38.

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

[25] C. Ferri, P. Flach, J. Hernández-Orallo and A. Senad, Modifying ROC curves to incorporate predicted probabilities. In *Proceedings of the Second Workshop on ROC Analysis in Machine Learning*, 2005.

[26] C. Ferri, J. Hernández-Orallo and R. Modroiu, An experimental comparison of performance measures for classification, *Pattern Recognition Letters* **30**(1) (2009), 27–38.

[27] C. Ferri, J. Hernández-Orallo and M.A. Salido, Volume under the ROC surface for multi-class problems. In *Proceedings of the Fourteenth European Conference on Machine Learning*, 2003, pp. 108–120.

[28] P. Flach, H. Blockeel, C. Ferri, J. Hernández-Orallo and 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.

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

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

[31] 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*, Menlo Park, CA, USA, 2003, pp. 194–201. AAAI Press.

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

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

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

[35] D.J. Hand and R.J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, *Machine Learning* **45**(2) (2001), 171–186.

[36] J.A. Hanley and B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology* **143**(1) (1982), 29–36.

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

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

[39] N. Lachiche and 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*, Menlo Park, CA, USA, 2003, pp. 416–423. AAAI Press.

[40] T.C.W. Landgrebe and R.P.W. Duin, Approximating the multiclass ROC by pairwise analysis, *Pattern Recognition Letters* **28**(13) (2007), 1747–1758.

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

[42] S.A. Macskassy and F. Provost, Confidence bands for ROC curves: Methods and an empirical study. In *Proceedings of the First Workshop on ROC Analysis in Artificial Intelligence*, 2004, pp. 61–70.

[43] A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, The DET curve in assessment of detection task performance. In *Proceedings of EuroSpeech*, 1997, pp. 1895–1898.

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

[45] N.A. Obuchowski, Receiver operating characteristic curves and their use in radiology, *Radiology* **229**(1) (2003), 3–8.

[46] F. Provost and 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.

[47] F. Provost and 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*, Menlo Park, CA, USA, 1997, pp. 43–48. AAAI Press.

[48] F. Provost and T. Fawcett, Robust classification for imprecise environments, *Machine Learning* **42**(3) (2001), 203–231.

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

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

[51] 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.

[52] A. Srinivasan, Extracting context-sensitive models in inductive logic programming. *Machine Learning* **44**(3) (2001), 301–324.

[53] J.A. Swets, Measuring the accuracy of diagnostic systems, *Science* **240**(4857) (1988), 1285–1293.

[54] D.M.J. Tax and 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*, 2008, pp. 724–733.

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

[56] S. Vanderlooy and E. Hüllermeier, A critical analysis of variants of the AUC. *Machine Learning* **72**(3) (2008), 247–262.

[57] M. Vuk and T. Curk, ROC curve, lift chart and calibration plot, *Metodološki zvezki* **3**(1) (2006), 89–108.

[58] G.I. Webb and K.M. Ting, On the application of ROC analysis to predict classification performance under varying class distributions, *Machine Learning* **58**(1) (2005), 25–32.

[59] I. Witten and 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.

[60] 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.

[61] K. Woods and K.W. Bowyer, Generating ROC curves for artificial neural networks, *IEEE Transactions on Medical Imaging* **16**(3) (1997), 329–337.

[62] S. Wu, P. Flach and C. Ferri, An improved model selection heuristic for AUC. In *Proceedings of the Eighteenth European Conference on Machine Learning*, 2007, pp. 478–489.

[63] S. Wu and 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*, 2005.

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