A Survey on Graphical Methods for Classification Predictive Performance Evaluation

Ronaldo C. Prati, Gustavo E.A.P.A. Batista, and Maria Carolina Monard

Abstract—Predictive performance evaluation is a fundamental issue in design, development, and deployment of classification systems. As predictive performance evaluation is a multidimensional problem, single scalar summaries such as error rate, although quite convenient due to its simplicity, can seldom evaluate all the aspects that a complete and reliable evaluation must consider. Due to this, various graphical performance evaluation methods are increasingly drawing the attention of machine learning, data mining, and pattern recognition communities. The main advantage of these types of methods resides in their ability to depict the trade-offs between evaluation aspects in a multidimensional space rather than reducing these aspects to an arbitrarily chosen (and often biased) single scalar measure. Furthermore, to appropriately select a suitable graphical method for a given task, it is crucial to identify its strengths and weaknesses. This paper surveys various graphical methods often used for predictive performance evaluation. By presenting these methods in the same framework, we hope this paper may shed some light on deciding which methods are more suitable to use in different situations.

Index Terms—Machine learning, data mining, performance evaluation, ROC curves, cost curves, lift graphs.

1 Introduction

Classifier performance depends fundamentally on the characteristics of the data to be classified. Broadly speaking, the no-free-lunch theorem for supervised machine learning [1], [2] states that, if all possible hypothesis are equally likely, the average performance of two different classifiers over all possible problems are equivalent. As a corollary of the no-free-lunch theorem, there is no single classifier that works best on all given problems. Many approaches for constructing classifiers have been proposed, including tree classifiers, neural networks, support vector machines, nearest neighbor methods, Naïve Bayes methods, linear and quadratic discriminant analysis, to name but a few.

Some attempts have been made aiming to predict whether some approach for constructing a classifier may perform well for a given domain [3], either from a theoretical or empirical point of view. One example is [4], where the authors have derived metrics based on the eigenvector decompositions of matrices commonly used in generalized linear discriminant analysis procedures. Based on these metrics, the authors carried out a theoretical analysis that demonstrate where and why the eigen-based linear equations beneath these methods do not work, so that it is possible to predict where these methods would not perform well. Furthermore, these metrics can be used to design more robust algorithms, which best fit the data at hand [5]. Unfortunately, this analysis could not be generalized for approaches other than the eigen-based linear methods. In addition, some approaches for classifier construction are based on soft computing approaches, and theoretical analysis is impractical. Thus, determining a suitable classifier for a given problem is essentially an empirical enterprise where evaluation procedures play a major role.

Therefore, evaluating the predictive performance of classification systems is an issue of great importance in machine learning, data mining, and pattern recognition as it is often used as the main indicator of predictive systems' quality. One of the important issues in performance evaluation is selecting the criterion to measure classifier performance. However, this task is not as trivial as it would seem at a first glance. Even the most widely used methods such as measuring accuracy or error rate on a test set (even using resampling techniques such as cross validation) have severe limitations [6]. Two of the most prominent limitations of these measures are that they do not consider misclassification costs, and can be misleading when classes have very different prior probabilities. Furthermore, they do not take into account the fact that, in real-world problems, class prevalence and misclassification costs are likely to change due to the inherent evolution of the process which generates the data [7]. The Area under the ROC curve (AUC), another performance measure that has been widely used in recent years, has also been criticized as it may use different misclassification cost distributions for different classifiers [8]. Moreover, in application oriented research, quality measures should reflect the concerns of the end users which are typically hard to model precisely [9].

- *R.C. Prati is with the Centro de Matemática, Computação e Cognição, Universidade Federal do ABC, Rua Santa Adélia, 166 Bairro Bangu, CEP 09.210-170, Santo André, São Paulo, Brazil. E-mail: ronaldo.prati@ufabc.edu.br.*
- *G.E.A.P.A. Batista and M.C. Monard are with the Departamento de Ciência da Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Avenida Trabalhador São-carlense, 400-Centro, Caixa Postal 668, CEP 13560-970, São Carlos, São Paulo, Brazil. E-mail: {gbatista, mcmonard}@icmc.usp.br.*

Manuscript received 19 May 2009; revised 24 Nov. 2009; accepted 21 May 2010; published online 24 Feb. 2011.
Recommended for acceptance by L. Wang.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2009-05-0445.
Digital Object Identifier no. 10.1109/TKDE.2011.59.

Due to these and other reasons, assessing the quality of a predictive system has been receiving much attention in recent years (see, for instance, the series of four recent workshops on this subject, the first two held together with the 2006 [10] and 2007 [11] editions of the Association for the Advancement of Artificial Intelligence—AAAI—conferences, and the latter two held together with the 2008 [12] and 2009 [13] editions of the International Conference on Machine Learning—ICML). One lesson that emerges from these concerns is that the prediction performance is an inherently multifaceted quantity. That is to say that, although it is quite common to express performance in terms of a single scalar (i.e., one-number) quantity (e.g., error rate, precision, recall, and so on) these considerations are likely to be insufficient. Any attempt to reduce the performance evaluation to a single scalar number might lose some information, imposing an arbitrarily chosen compromise among the components of performance. Although attractive from a practical standpoint, these reductions will necessarily give incomplete pictures of prediction performance. To sum up, although a single scalar measure may capture some aspects of performance, it does not capture all the aspects [14]. Therefore, a complete and reliable analysis must consider all the various components of performance quality.

Over the last years, various studies have pointed out alternative methods to evaluate the performance of predictive systems. Some of these methods are based on graphical or diagram evaluation and are used as an alternative to scalar performance measures because they can display more of the multidimensionality and complexity of the evaluation of the underlying problem. The main advantage of graphical methods resides in their ability to depict the trade-offs between evaluation aspects in a multidimensional space rather than reducing these aspects to an arbitrarily chosen (and often biased) single scalar measure. However, graphical methods are not as easy to interpret and analyze as single scalar values are [15].

The purpose of this paper is two-fold: the first one is to put various graphical methods for evaluating predictive systems, which are often used in machine learning, data mining, and pattern recognition, in the same framework, so that the reader can get an overall idea of how to use and interpret them. The methods often used in machine learning and pattern recognition include precision-recall [16] and ROC graphs [17], as well as some methods based on expected profit or return, such as lift and return of investment (ROI) graphs [18], or costs, such as cost curves [19]. These methods are related to evaluating the discrimination aspect of predictive systems. The second purpose is to call the attention of the machine learning, data mining, and pattern recognition communities to other methods used in other areas (mainly weather forecasting) aimed to evaluate other aspects of predictive performance evaluations. To the best of our knowledge, these methods are seldom used in machine learning, data mining, and pattern recognition contexts.

The outline of the paper is as follows: Section 2 introduces the problem of predictive performance evaluation. Section 3 reviews some graphical models for evaluating discrete binary classifiers. These methods are extended in Section 4, which approaches the problem of graphical evaluation of rated or ranked predictions as well as in Section 5, which deals with continuous predictions. Section 6 highlights the main advantages and drawbacks of each method described in this paper. Section 7 shows an illustrative example of using graphs to analyze predictive performance, and Section 8 concludes the work.

2 Predictive Performance Evaluation

Let X be the true label of an instance (the "ground truth") and Y the prediction made by a predictive model. The purpose of predictive performance evaluation is to assess the agreement between X and Y. Each of these quantities (X and Y) might be continuous, ordinal, or categorical. The prediction of continuous variables is given the name of regression; prediction of ordinal variables are often called ordinal regression, rating or ranking; and prediction of categorical variables is called classification or categorization.

Although it is often assumed that there is a direct correspondence between the types of X and Y, i.e., both X and Y are of the same type, this restriction is not imperative. One may have, for example, a continuous observation $X \in \Re$ and a discrete prediction $Y \in \{c_1, c_2, \ldots, c_k\}$, where $k \in \aleph$ is the number of possible predicted categories (classes). In this particular example, we are dealing with a regression problem through discretization [20]. Other examples are using ranking-based evaluation of regression models [21] or a continuous-based function which aims to minimize classification error for crisp classifiers [22].

In this paper, we are concerned about classification problems, i.e., problems where the labels are categorical. Furthermore, as most graphical evaluation methods are suitable for binary classification problems, we constraint our discussion to two-class problems, which are given the general class labels **positive** and **negative**. Therefore, in this paper we are dealing with binary classification problems (real class X is a discrete variable that can assume one of two possible values) while prediction Y given by the predictive models might be used in three possible situations:

- **Classification.** In this case, we are dealing with standard "crisp" classifiers, i.e., classifiers which only predict the class label, and Y may assume discrete values (one of the class values).
- **Ranking.** While a crisp classifier aims to distinguish instances from each class, a ranker orders instances from high to low expectation where the instance is of a certain class—generally the positive class. Most classification models in machine learning output some score of positiveness, and hence can be used as rankers. Conversely, any ranker can be turned into a classifier if we have some instance-independent means of splitting the ranking into positive and negative segments. This could be a fixed score threshold or a percentage of cases that should be classified as positive. In this case, we are interested in two-tier ordering, where we would like to place as many cases of the target class as possible at the top of the rank.
- **Probability estimation.** In this case, Y is a continuous variable that somehow estimates the likelihood of a given instance, to be classified in one of the

TABLE 1
Guidelines for Choosing a Graphical Evaluation Tool for Assessing Binary Classification Models

Predicted variable	User objectives	Graphic method tool to use
Classes	Compare the performance of each class of two or more classifiers independently from class distributions or misclassification costs.	ROC graphs (Section 3.2)
	Relate the error rate (or normalized expected cost) with all possible combinations of class prevalence and misclassification costs (operational conditions), and vice-versa.	Cost lines (Section 3.3)
Rankings	Examine the tradeoff between the ability of a classifier to correctly rank positive cases in front of the negative cases independently of class proportions or misclassification costs.	ROC curves (Section 4.1)
	Visualize the performance in terms of error rate (or normalized expected cost) of all classifiers that can be derived by assuming different operational conditions, and vice-versa.	Cost curves (Section 4.2)
	Analyze the performance of how the cases from one of the classes (the positive class) appear at the top positions of the ranking.	Precision-recall curves (Section 4.3)
	Evaluate the likelihood of identifying a large number of positive cases by following the rank instead of picking up a random sample of instances.	Lift curves (Section 4.4)
	Estimate the expected return of investment when a fixed cost should be spent to reach the top ranked cases.	ROI curves (Section 4.5)
Probabilities	Evaluate whether the predicted probabilities are well calibrated.	Reliability diagram (Section 5.2)
	Identify regions where the predicted probabilities are degrading performance when compared to a reference classifier which always predicts a class with constant probability.	Attributes diagram (Section 5.3)
	Show the extent to which the predictions discriminate between the classes.	Discrimination diagram (Section 5.4)

classes. The continuous variable might be (re)scaled to represent probabilities. However, unless we are interested in calibrated probability estimates, this is an unnecessary step.

Regardless of the type of Y variable, the joint probability distribution of observations and predictions—$p(X, Y)$—encapsulates all the components of performance. However, different graphical tools are necessary to analyze the different types of Y. This paper presents graphical tools that can be used to evaluate predictive models for each type of the predicted Y variable. Table 1 summarizes the methods presented in this paper according to how the Y value can be interpreted, as well as the user objectives in carrying out the evaluation. This table can be used to provide some guidance on how the paper can be consulted.

3 Graphical Evaluation Methods for Discrete Predictions

For binary discrete observations and predictions, the joint probability distribution $p(X, Y)$ can be cross tabulated into a 2×2 contingency table, as shown in Table 2. In this table, x and $\overline{x}$ represent the events $X = \text{positive}$ and $X = \text{negative}$ and y and $\overline{y}$ represent the events $Y = \text{positive}$ and $Y = \text{negative}$, respectively, meaning the actual/predicted class is positive/negative. $TPos$, $FPos$, $TNeg$, and $FNeg$ represent true/false positive/negative example counts, respectively. Pos, Neg, $PPos$, and $PNeg$ represent the number of examples actual/predicted as positives/negatives, respectively. N is the sample size.

TABLE 2
A 2×2 Contingency Table

X \ Y	y	$\overline{y}$	
x	$TPos$	$FNeg$	Pos
$\overline{x}$	$FPos$	$TNeg$	Neg
	$PPos$	$PNeg$	N

The two inner rows correspond to actual classes, while the two inner columns correspond to predicted classes.

From a contingency table, the joint probability distribution for each pair of events can be easily estimated by dividing the respective inner cells by the sample size. For instance, $p(X = x, Y = y) = \frac{TPos}{N}$. The joint probability distribution of observations and predictions—$p(X, Y)$—encapsulates all the components of performance. However, the information contained in $p(X, Y)$ is more accessible when this distribution is factored into conditional and marginal distributions. Two such factorizations can be identified, which follows from the basic laws of probability

$$p(X|Y) = \frac{p(X,Y)}{p(Y)}, \text{ and}$$
$$p(Y|X) = \frac{p(X,Y)}{p(X)},$$

where $p(X|Y)$ is the conditional probability of observing X given that the prediction is Y and $p(Y|X)$ is the conditional probability of predicting Y given that the observation is X.

The former factorization is known in statistics as reliability or predictive value (in machine learning, it is also known as precision). Note that, in this case, the direction of the probability is from prediction to truth. This probability is of particular interest to the classifier's user, since it provides the probability of correct classification given a prediction. Therefore, these measures can be understood as a confidence value for a given prediction. The latter factorization is often referred to as likelihoods, since it specifies the likelihoods that a particular prevision was made given the occurrence of a specific observation. The direction of this probability is from truth to prediction, and is of principal utility when evaluating

predictive systems. These likelihoods indicate the extent to which predictions discriminate among the values of X.

3.1 Cost-Sensitive Learning

For binary classification, two types of errors may occur: false positives and false negatives. Most learning systems deal with these errors as equally costly and try to minimize the overall error rate. Furthermore, a cost-sensitive learning system can be used in applications where the *misclassification costs* are known. A misclassification cost is simply a value that is assigned as a penalty for making a mistake. In this case, misclassification costs can be used in substitution for the error rate, and a cost-sensitive learning system attempts to reduce the cost of misclassified examples instead of classification errors.

Usually, a *cost matrix* is used to define the costs associated to a domain. A cost matrix is similar to a contingency table. If the values on the main diagonal are represented with negative costs, then these values can be interpreted as *gains* or *profits*. Each entry of a cost matrix defines a constant cost/profit for each type of error/hit that can be made by a classifier. Given a contingency table and a cost matrix, the expected cost, EC,[1] can be computed using

$$EC = \sum_{X \in \{x, \overline{x}\}} \sum_{Y \in \{y, \overline{y}\}} p(X, Y) c(X, Y), \tag{1}$$

where $p(X, Y)$ is the corresponding cell in the contingency table divided by N and $c(X, Y)$ is the cost/profit for that type of classification.

Some learning systems are not able to integrate cost information into the learning process. However, there is a simple and general method to make any learning system cost sensitive for a binary class problem if costs are known and are constant [23]. The idea is to change the class distributions in the training set toward the most costly class. Suppose that the positive class is five times more costly than the negative class. If the number of positive examples are artificially increased by a factor of five, then the learning system, aiming to reduce the number of classification errors, will come up with a classifier that is skewed toward the avoidance of errors in the positive class, since any such errors are penalized five times more. In [24], a theorem is provided that shows how to change the proportion of positive and negative examples in order to make optimal cost-sensitive classifications for a concept-learning problem. Moreover, a general method to make a learning system cost sensitive is presented in [25]. This method has the advantage of being applicable to multiclass problems.

3.2 ROC Graph

As we are concerned with two class problems, two conditional probabilities[2]—$p(y|x)$ or true positive rate (tpr) and $p(y|\overline{x})$ or false positive rate (fpr)—are sufficient to provide all the information needed for the evaluation of a binary classifier. A ROC graph is just a plot of these two probabilities—fpr on the x-axis and tpr on the y-axis. A binary "discrete" classifier produces a single point with coordinates (fpr, tpr) in the ROC space.

Some points in the ROC space are worth noticing. The lower left corner $(0, 0)$ represents the classifier which always predicts negative; such a classifier does not produce any false positive errors, although it is not able to classify any positive ones as well. The upper right corner $(1, 1)$ represents the opposite strategy of unconditionally classifying every case as positive. The upper left corner $(0, 1)$ represents perfect classification, while the lower right corner $(1, 0)$ represents the always wrong classifier. Any classifier which performs no better or worse than chance falls in a point in the ascending diagonal. Classifiers which perform worse than random appear in the lower right triangle formed by the points $(0, 0), (1, 1)$, and $(1, 0)$. For this reason, this area is usually empty. A point in the ROC space is better than another if it is to the northwest (tpr is higher and fpr is lower) of the first.

One advantage of ROC graphs is that they can visualize and organize classifiers' performance without considering class distributions or misclassification costs. This ability is very important when investigating learning algorithms in skewed class distributions or cost-sensitive learning situations. The performance of a set of classifiers can be graphed, and as long as the class conditional likelihoods do not change, the graph will remain invariant with respect to the class skew and misclassification costs (operational conditions). As these operational conditions change, the region of interest may change, but the graph itself does not change.

In [6], it is shown that the operating conditions may be easily transformed into the so-called expected cost isoperformance line in the ROC space. Two points in this space, (fpr_1, tpr_1) and (fpr_2, tpr_2), have the same expected cost if

$$\frac{tpr_2 - tpr_1}{fpr_2 - fpr_1} = \frac{c(x, \overline{y}) p(\overline{x})}{c(\overline{x}, y) p(x)} = m. \tag{2}$$

This equation defines the slope m of a total cost isoperformance line. All classifiers lying at points on a line of slope m have the same expected cost (1). Each set of class priors and misclassification costs defines a family of isoperformance lines. The "more northwest" a line is (having a larger TP-intercept) the better, because it corresponds to classifiers with a lower expected cost. This implies that, regardless of operational conditions, a classifier is potentially optimal if and only if it lies on the convex hull of the set of points in the ROC space. The convex hull of the set of points in the ROC space is called the ROC convex hull (ROCCH) of the corresponding set of classifiers.

An example of a ROC graph is shown in Fig. 1. In this graph, five hypothetical classifiers are depicted: A, B, C, D, and E. The graph also shows the ROCCH. The convex hull is bounded only by the trivial points $(0, 0)$ and $(1, 1)$ and by the points A, B, and C. Points D and E are not in the ROCCH and therefore are suboptimal. Thus, as we are looking for optimal classification performance, classifiers D and E can be entirely removed from consideration. That is to say that there are no combinations of class and cost distributions in which classifiers D and E present lower expected costs than A, B, or C.

However, the choice among A, B, and C depends on information regarding operational conditions. For example, the slope of the line segment connecting the origin to

1. Note that EC degenerates to the error rate if all the misclassification costs are set to 1.

2. To simplify the notation, from hereafter we shall use $p(x, y)$ meaning $p(X = x, Y = y)$, $c(x, y)$ meaning $c(X = x, Y = y)$, and so forth.

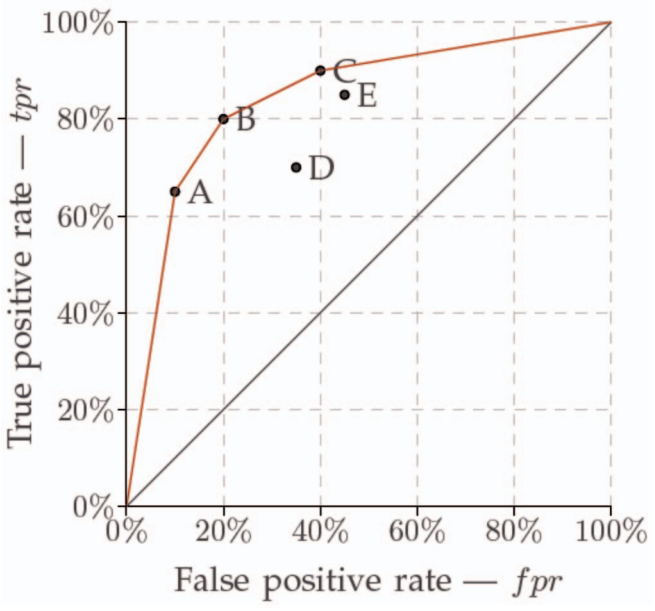

Fig. 1. ROC graph.

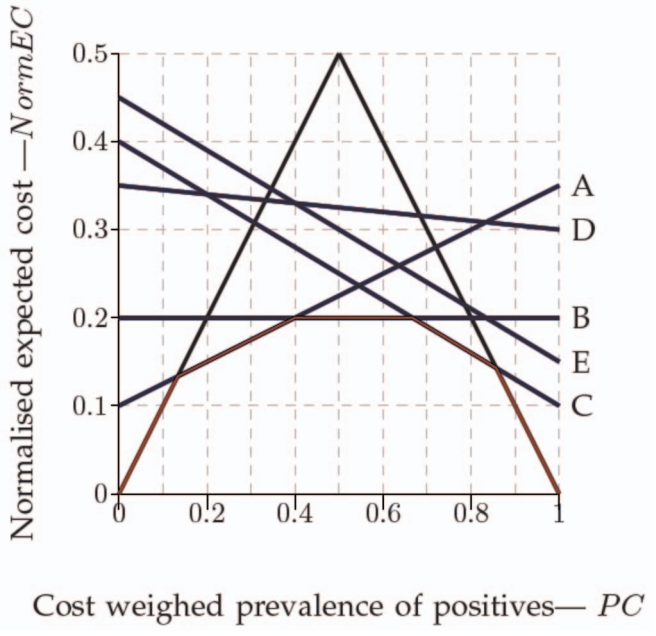

Fig. 2. Cost lines.

point A is 6.5. If committing a false positive error is at least 6.5 times more costly than committing a false negative, or the proportion of negative cases is at least 6.5 times greater than the positives, or any combination of these two factors lead to a slope m (2) greater than 6.5, the trivial classifier of never issuing a positive classification (the point $(0,0)$ in the graph) would be preferred for A, B, and C. If the slope is exactly 6.5, either the trivial classifier of always classifying everything as negative or the A classifier would have the same expected cost. The difference is how these two approaches perform in the different classes, although the expected cost (which is a weighed average of each class performance) is the same. In fact, we can achieve any point in between these two classifiers by randomly alternating between them. The points can be obtained by varying the probability in which we choose one of the classifiers.

3.3 Cost Lines

In [19], a modification of ROC graphs in order to facilitate the reading of the classifiers' expected costs is proposed. The authors proposed a point-line transformation from the ROC space so that a point in that space is represented by a line in the cost space. The y-axis in the cost space is related to the expected cost (1). The expected cost is normalized so that the maximum possible expected cost is 1. The x-axis is related to the proportion of positive cases (prevalence of positive) multiplied by its respective cost. This prevalence weighed by cost is also normalized by the maximum cost so that it is scaled to 0-1. The formulas of the y- and x-axes are given by (3) and (4), respectively,

$$NormEC = \frac{EC}{maxEC}, \tag{3}$$

$$PC = \frac{p(x)c(\overline{x},y)}{maxEC}, \tag{4}$$

where $maxEC$ is given by

$$maxEC = p(x)c(\overline{x},y) + p(\overline{x})c(x,\overline{y}). \tag{5}$$

The same set of classifiers represented in the ROC space in Fig. 1 is depicted in the cost space in Fig. 2. Note that each classifier represented by a point in the ROC space is now represented by a line in the cost space. The always negative trivial classifier, represented by the point $(0,0)$ in the ROC space, is represented by the ascending diagonal in the cost space. In the same way, the always positive classifier, represented by the point $(1,1)$ in the ROC space, is represented by the descending diagonal in the cost space. On the other hand, random performance, which is represented by the ascending diagonal in the ROC space, is represented by the point $(0.5,0.5)$ in the cost space.

Note that the expected cost performance, which could only be implicitly observed in the ROC space by taking into account slopes and isoperformance lines, can be easily read directly from the graph in the cost space. Indeed, this is one of the main advantages of this sort of graph. As class or misclassification distributions change (a situation that may occur in practice, e.g., in an epidemic crisis in a medical domain), one can look at the graph of the corresponding PC (4) and see what classifier is the best one for the new operating conditions. For instance, by reading the cost graph in Fig. 2, we can see that the always negative classifier is optimal whenever the PC lies between 0 and 0.13. Furthermore, classifier A is optimal whenever the PC is between 0.13 and 0.4, and so forth. Similar to the ROCCH, where all optimal classifiers lie in the ROC convex hull, in the cost space the optimal classifiers form a lower envelope of various cost lines.

Although cost curves are very convenient for reading expected cost performance, this comes with a price: the performance in each one of the classes, which can be easily seen in the ROC graph, is not accessible in a cost graph. This is because the expected cost is a trade-off between the performance in both classes.

In a sense, ROC and cost graphs are complementary approaches. ROC graphs are very convenient in helping us to understand class conditional performance and working

on possible ways to overcome possible deficiencies of a classifier in a particular class (e.g., in problems where there is a strong imbalance between classes). On the other hand, cost curves are very convenient in helping us to understand classifiers' performance in their deployment environment.

4 Graphical Evaluation Methods for Ranked Predictions

Ranking is prevalent in real-world applications where it is required to order—or rank—cases rather than simply classifying each case. An example is a recommendation system, where the aim is to obtain a ranked list of goods—say books or movies—a customer is likely to enjoy, based on his/her preferences. Another example is direct mail marketing, where the vendor would like to have a ranked list of potential customers regarding their likelihood of purchasing a product after receiving a catalog. This list could be used to mail the catalog only to the top best potential customers rather than mailing it to all the customers, thus maximizing the expected profit.

Given a set of cases we are interested in, a rank is an ordered list of these cases. As we are dealing with binary class problems, we are interested in a bipartite ranking, where we would like to rank cases from one of the classes (the target class) higher than the cases of the other class. In other words, we are interested in two-tier ordering, where we would like to place as many cases of the target class as possible at the top of the rank.

Only a few learning systems are able to directly produce rankings. The most used approach is to use the continuous output given by some systems and use this output to order the cases and produce the ranking. In other words, we ignore the face value given by these systems and only take into account the ordering of the set of cases they define. The reason for this is manifold [21]: such as in the examples at the beginning of this section, ranking may be the real goal in building the prediction model; ranking-based measures are quite interpretable; ranking-based evaluation is robust. In fact, the scores produced by these learning systems are sometimes "biased" or "uncalibrated," and an additional step is often required to make these scores more useful. Some graphical methods to assess the quality of continuous scores will be discussed in Section 5, while this section describes graphical methods which can be applied to ranked predictions.

4.1 ROC Curves

As mentioned in Section 3.2, a binary "crisp" classifier—one that predicts only the class label—produces a single point, represented by the pair (fpr, tpr) in the ROC space. On the other hand, a ranking function may be thresholded to produce a binary classifier by predicting the top n-percent of the cases as positive. By varying this percentage from 0 to 100 percent, we can produce various points so that a curve in the ROC space can be traced. A similar procedure can be used to obtain ROC curves of a scoring classifier. In this case, we may vary the threshold from $-\infty$ to $+\infty$ to obtain the points in the ROC space that compose the ROC curve. Note that, as we are only interested in true positive and false positive rates, the values of the scores are not important.

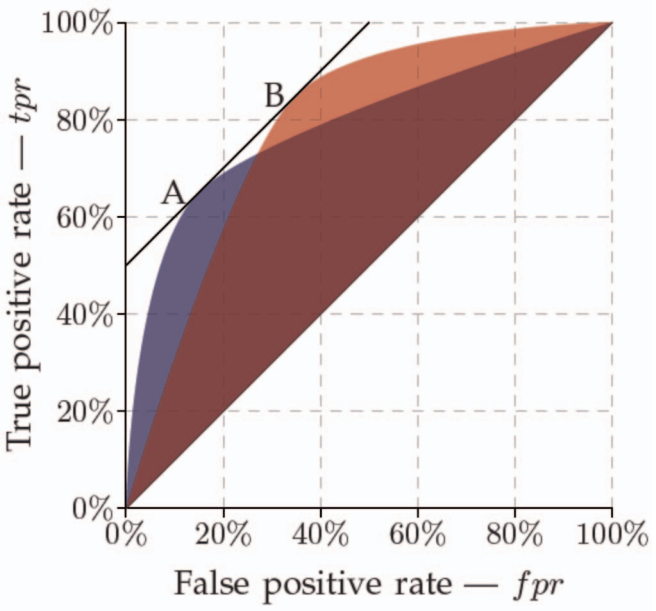

Fig. 3. ROC curves.

Therefore, from a ROC perspective, a scoring classifier is equivalent to a ranking function. In other words, we can think of a ROC curve as a parametrized set of classifiers, where each parameter value produces a point in the ROC space. The curve is obtained by joining all these points.

The sharper the curve bends, the greater the ability in putting positive cases at the top of the rank. The crisper the curve flattens toward the (0,1)-(1,1) line, the greater the ability in leaving negative cases at the bottom part of the rank. If the curve hits the upper left corner (0,1) point, there is a perfect ranking. An example of two ROC curves is shown in Fig. 3. For each curve, the two probabilities vary together from the lower left corner, where both true and false positive rates are near 0, as they would be for a very strict threshold, to the upper right corner, where both rates are near 1, as they would be for a very lenient threshold. In between, the curve would rise smoothly, having a decreasing slope, to represent all possible thresholds. Hence, the curve is independent of whatever threshold is chosen in a particular task.

Analyzing the curves, we can conclude that curve A is better at the top part of the rank (it is better in grouping positive cases at the top of the rank), and thus it might be appropriate in problems such as information retrieval, where we are interested in classifying the positive cases better. Curve B is better at the bottom of the rank (it is better in grouping negative cases at the bottom of the rank), and thus might be interesting in, for instance, some medical problems where a cheap test can be used to exclude people who do not have a certain disease and a more costly test can be further applied to the remaining people. The line tangenting both curves is the line where the expected cost is the same for both models.

A single measure of ranking performance can be derived by calculating the AUC. This area can be interpreted as the probability of a randomly chosen positive case being ranked higher than a randomly chosen negative case. Furthermore, the AUC is numerically equivalent to the Wilcoxon signed rank test and correlated to the Gini index [26]. However, the

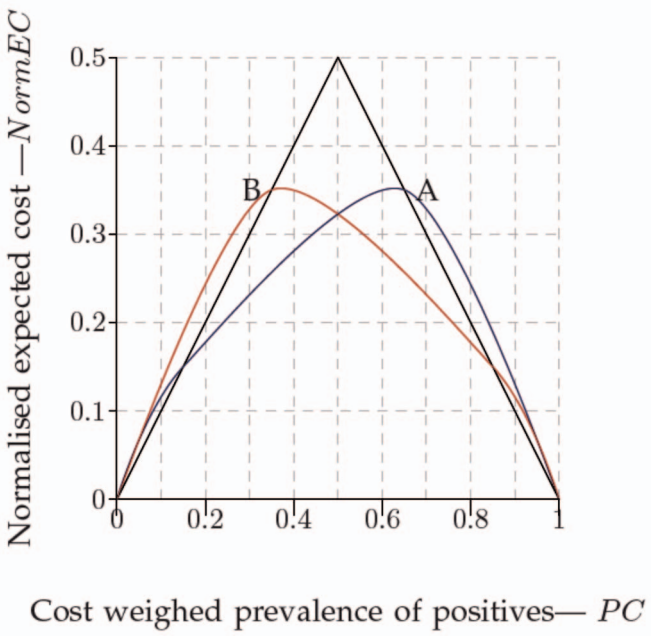

Fig. 4. Cost curves.

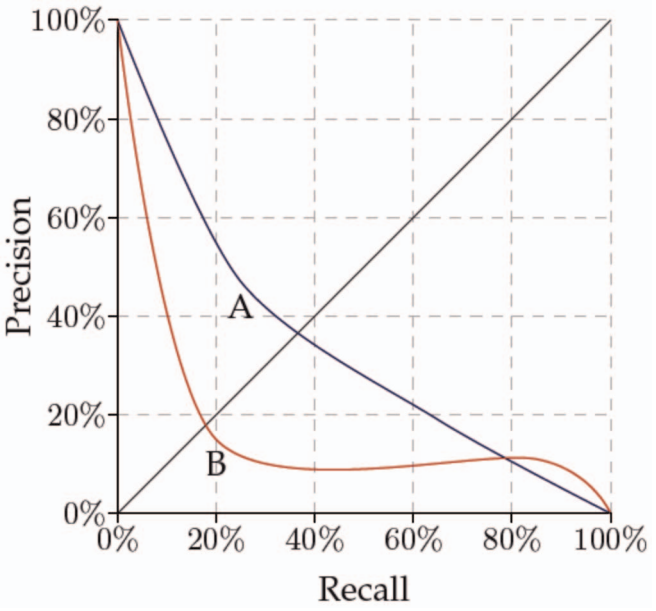

Fig. 5. Precision-recall curve.

AUC also has some drawbacks: for example, both curves in Fig. 3 have the same AUC value, although one might be better than the other, depending on the circumstance. This is because the AUC integrates over all possible thresholds, and thus treats equally misrankings at the top or at the bottom of the ranking [27].

4.2 Cost Curves

As for ROC graphs for binary predictions, where each classifier in the ROC space has its correspondent line in the cost space, we can derive a cost curve correspondent to a ROC curve. The process is very similar to the generation of a ROC curve: by varying the percentage of cases classified as positive from 0 to 100 percent. This percentage is used as a parameter so that each possible parameter value produces a line in the cost space. A set of points in the ROC space is a set of cost lines, one for each ROC point [19].

Fig. 4 presents the cost curves corresponding to the ROC curves shown in Fig. 3. For the sake of visualization, the cost lines which compose the cost curves are omitted and only the cost curves are shown. As in the cost lines case (Section 3.3), the reading of the expected cost is facilitated. Furthermore, the reading of the operating range is quite simple as well. For instance, model A is outperformed by the always negative classifier as long as PC is lower than 0.15. Model A is better than the other if and only if the PC ranges from 0.15 and 0.5, and so forth.

In spite of its advantages, as in the case of cost lines, cost curves have their drawbacks. We are not able to see the performance with respect to each class individually. Furthermore, we are aware of the concavity which lies between the two ROC curves and the line with the same expected cost (Fig. 3). This is because this entire line in the ROC space is mapped to the point where both cost curves intercept in the cost space. Although the expected cost is the same in the entire line in the ROC space, we can achieve different performances in both classes (the expected cost is a weighed mean of the performance in both classes) and this cannot be inferred by the cost curve. In other words, cost curves and ROC graphs are not competitors but are, in fact, complementary approaches. The remaining part of this section presents other complementary approaches which explore different aspects of ranking evaluation.

4.3 Precision-Recall Curves

Precision-recall curves are often used in information retrieval applications to evaluate ranked retrieval performance results [16]. This is because information retrieval tasks are often characterized by a large skew in the class distribution, i.e., the number of negative cases heavily outnumbers the number of positive cases. Furthermore, in information retrieval tasks, the positive class is of more interest than the negative one. This particular characteristic of information retrieval tasks makes the area of interest in a ROC graph compressed to a small corner in the lower left side of the ROC space.

The difference between the ROC and precision-recall space is that the x-axis, represented by the fpr in the ROC space, is replaced by tpr (recall) and the y-axis is represented by the precision (positive predictive value), as shown in (6). The objective of this axis setup is to make differences in the area of interest clearer than in the ROC space. However, precision-recall curves are dependent on prior knowledge about class distributions

$$precision = \frac{p(x,y)}{p(y)}. \tag{6}$$

Fig. 5 shows the precision-recall corresponding to the ROC curves shown in Fig. 3, for an arbitrarily chosen prevalence of positives of 5 percent. Analyzing the curves, in spite of both models having the same AUC, we can see that model A is better at identifying positives than negatives (has higher precision) in almost all the x-axis. Furthermore, due to the low prevalence of positives, this difference is much more significant in the precision-recall

space than in the ROC space. The higher the prevalence of the positives, the closer the curves in the precision-recall space are. This is because it is easy to obtain high precision (say 96 percent) in domains where the prevalence of positives is also high (say 95 percent).

4.4 Lift Graph

Lift graphs are frequently used by the database marketing community [28]. Similarly to ROC graphs, lift graphs associate the true positive rate (tpr) with the y-axis. However, $p(y)$ is associated to the x-axis instead of the false positive rate (fpr). This change makes lift graphs sensitive to operational conditions, since

$$p(y) = p(y|x)p(x) + p(y|\overline{x})p(\overline{x}). \quad (7)$$

Therefore, $p(y)$ can be derived from tpr ($p(y|x)$) and fpr ($p(y|\overline{x})$) and the positive ($p(x)$) and negative ($p(\overline{x})$) class prevalence. Furthermore, as the positive class prevalence tends to 1, $p(y)$ tends to tpr and the lift curve approaches the increasing diagonal line. As the positive class prevalence tends to 0, $p(y)$ tends to fpr, and the lift curve approaches the ROC curve.

As with ROC graphs, a crisp classifier corresponds to a point in a lift graph. However, a set of points can be generated by varying the percentage of cases classified as positive. A *lift curve* is defined as the convex hull of all points generated. Fig. 6 depicts two lift curves corresponding to the ROC curves in Fig. 3. The positive class prevalence is 5 percent in Fig. 6a and 50 percent in Fig. 6b. Similar to the ROC graph, the upward diagonal line represents a random classifier.

Lift graphs are popular in database marketing as marketing campaigns, such as mail campaigns, usually have very low response rates. Therefore, mass mailing, *i.e*, mailing all prospects in a customer database, is not usually profitable. An alternative, known as direct mailing, is to rank the prospects so that the top ranked prospects are more likely to purchase the offered product, and the campaign is restricted to contact a set of likely respondents. Thus, $p(y)$ is associated with the percentage prospects contacted, which is a fraction of all prospects, and tpr is associated with the percentage of respondents, which is a fraction of all positive respondents. Therefore, a lift curve shows the relationship between a set of top ranked examples and the number of positive examples in this set, expressed as a percentage of the total number of positive examples. In Fig. 6a, mailing the 10 percent top-ranked prospects will reach 50 percent of the respondents for classifier A and 28 percent for B. On the other hand, mailing the 40 percent top-ranked will reach 78 percent for A and 87 percent for B.

4.5 ROI Graph

Return of Investment (ROI) graphs are similar to lift graphs. However, ROI graphs associate the total expected profit (TEP), given by (8), to the y-axis

$$TEP = N \sum_{X \in \{x,\overline{x}\}} \sum_{Y \in \{y,\overline{y}\}} p(X,Y)c(X,Y)p(X), \quad (8)$$

where N is the sample size, $p(X,Y)$ is the corresponding cell in the contingency table divided by N, $c(X,Y)$ is the cost/profit for that type of classification and $p(X)$ represents the class prevalence.

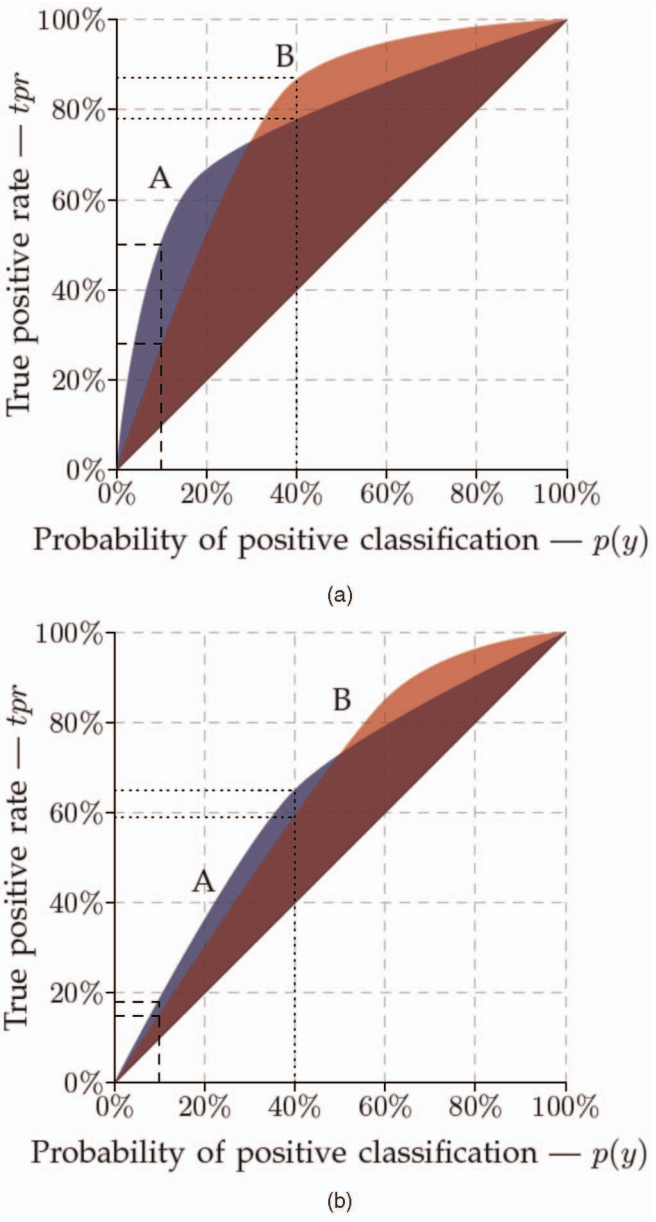

Fig. 6. A lift graph. (a) Prevalence of positives at 5 percent. (b) Prevalence of positives at 50 percent.

In order to compute the TEP, $c(X,Y)$ should associate positive values with profits and negative values with costs. An association of negative values with profits and positive values with costs changes (8) to calculate the total expected cost (TEC). ROI graphs are limited to domains in which costs and class prevalence are constant and can be estimated confidently in advance.

In the marketing campaign example, suppose that the cost matrix is given by Table 3. In this example, \$50.00 is the profit obtained by selling one unit of the advertized product and \$3.00 is the cost of sending each mail. It is assumed that there is no cost involved in not sending a mail.

Fig. 7 depicts two ROI curves corresponding to the ROC curves in Fig. 3. The positive class prevalence is 0.05, thus these classifiers have the same operational conditions as the

TABLE 3
A Cost Matrix for the Marketing Campaign Example

X \ Y	y	$\overline{y}$
x	50	-3
$\overline{x}$	0	0

lift curves shown in Fig. 6a. In addition, the sample size N was fixed in 100,000 prospects. As in ROC and lift graphs, the diagonal line of a ROI graph shows the performance obtained by a random classifier.

Fig. 7 shows that campaigns which contact a small number of prospects are more likely to be profitable if classifier A is used. For instance, the total expected profit for a campaign that mails 11 percent of all prospects is \$96,500 for classifier A and \$41,500 for classifier B. However, classifier B outperforms classifier A for larger campaigns. For instance, if 42 percent of all prospects are mailed, classifier A will have a total expected profit of \$81,000 and classifier B a profit of \$103,500. The downward diagonal line indicates that any random classifier is not able to provide a profitable campaign.

A ROI curve presents a maximum point that provides a maximum return of investment. From this point, an optimum number of prospects to be contacted can be estimated. The graph in Fig. 7 shows that classifier A provides the highest total expected profit of \$119,750 when 18 percent of all prospects are mailed.

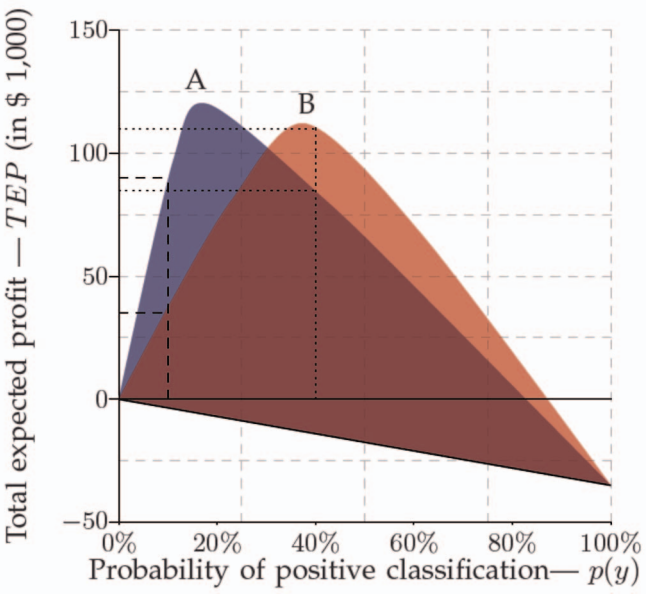

Fig. 7. A ROI graph.

5 Graphical Evaluation Methods for Continuous Predictions

For most practical problems where a predictive system outputs a continuous score, ranking performance evaluation is sufficient for evaluating performance [21]. This is true in cases where the main objective is to assess the discrimination ability of the system, i.e., how the system is able to separate cases of one class from the other by assigning high scores to one of the classes and low scores to the other. For these cases, the graphical methods presented in Section 4 can be used to carry out the analysis.

Common to all these methods is the fact that they completely ignore the magnitude of the predictions, and only take into account the relative ordering among the cases. However, in some situations we do need to take the face value of the prediction into account. One such situation is related to reliable probability predictions. In this case, besides discrimination, we are also interested in other aspects of performance that can be measured by taking into account the magnitude of the precision. This section presents some graphical methods often used in weather forecast analyses, which can be used to evaluate predictive systems.

5.1 Calibration and Refinement: The Brier Score

One of the most used measures to evaluate probabilistic forecasts in binary events is the Brier score (BS) [29]. The Brier score is essentially the mean squared error of the probability predictions considering that the observation is $x = 1$, if the event occurs (a positive class case), and that the observation is $\overline{x} = 0$ if the event does not occur (a negative class case). It can be estimated as the average deviation between predicted probabilities for a set of events and their prediction, i.e.,

$$BS = \frac{1}{N}\sum_{n=1}^{N}(y_n - x_n)^2,$$

where x_n is either 1 or 0 whether a case is negative or not.

The Brier score is the analogous of the mean squared error in regression analysis. A minimum Brier score of zero is obtained for a perfect (deterministic) system in which $y_n = x_n$ for all n. This system issues a probability prediction of 1(0) for each positive (negative) case. On the other hand, the Brier score takes its maximum value of one for a systematically erroneous (although perfect resolution) classifier which predicts the wrong class with confidence.

An instructive algebraic decomposition of the Brier score is derived in [30]. It is related to two aspects of probabilistic performance evaluation, namely calibration and refinement. Suppose there are t distinct values of y_j for $j = 1, \ldots, t$, and the set of cases with identical associated probability value y_j is n_j. Thus, $N = \sum_{j=1}^{t} |n_j|$. Therefore, the set of N predictions can be divided into t subsets, where each subset consists of the n_j predictions. In this context, and as the Brier score is quadratic, it can be usefully decomposed into the sum of three parts

$$BS = \underbrace{p(x)(1-p(x))}_{\text{Uncertainty}} + \underbrace{\frac{1}{N}\sum_{j=1}^{t}|n_j|(y_j - p(x|j))^2}_{\text{Reliability}} - \underbrace{\frac{1}{N}\sum_{j=1}^{t}|n_j|(p(x|j) - p(x))^2}_{\text{Resolution}}, \quad (9)$$

where $p(x|j)$ is the probability that the true class is x when the forecaster predicts x with probability j.

One may note that this decomposition is very similar to the bias-variance decomposition for regression analysis. However, instead of computing the mean squared error of the continuous prediction *versus* the continuous observation, we calculate the mean of the observations (the relative frequency of the examples of the positive class) stratified/conditioned on the different forecast probabilities. The first term in (9) depends only on the variability of the observations and cannot be influenced by the predictions, and is often called observational uncertainty term. The second and third terms in (9) are related to reliability (bias of conditional means, or conditional bias) and resolution (variance of conditional means, or conditional variance) of probabilistic predictions. The definitions of these two terms formalize the interpretations of calibration and refinement.

The reliability term in (9) summarizes the calibration of the predictions. It consists of a weighed average of squared differences between the prediction probability $p(x|j)$ and the relative frequencies of the predicted event of each subsample y_j. For perfect reliable predictions, the subsample relative frequency is exactly equal to the prediction probability. Furthermore, for reliable well-calibrated predictions, all the squared differences in the reliability term will be near zero and their weighed average will be small.

The resolution term in (9) summarizes the resolution of the predictions. It is aimed to assess the ability of the predictive systems to discern subsamples to which predictions substantially differ from the observed proportion of cases in the population. Resolution, also known as refinement, scores the usefulness of each forecast. For instance, in a place that rains 50 percent of the time, a forecaster that always announces rain with 50 percent of confidence is calibrated, yet not very useful [31]. Mathematically, the resolution term is a weighed average of the squared differences between the subsample relative frequencies and that of the overall population. Thus, if the prediction subsamples have substantially different relative frequencies than the overall distribution, the resolution term will be large. This is a desirable situation, as the resolution term is subtracted in (9). Conversely, if the proportion of cases in the prediction subsamples are very similar to the overall proportion, this term will be small and the predictions will weakly distinguish the classes.

5.2 Reliability Diagram

As we are arguing throughout this paper, a single scalar summary such as the Brier score (or its components) can provide a convenient idea about some aspect of performance evaluation, but a comprehensive appreciation of the predictive system should consider the full joint distribution of predictions and observations. The reliability diagram is a graph that shows this joint distribution in order to highlight the calibration and refinement components of probabilistic binary predictors. The reliability diagram has some resemblance with quantile-quantile plots in the regression analysis.

The reliability diagram contains two curves. The first one is a plot of the calibration function. It is a plot of $p(x|j)$ distribution by j, and measures the degree to which predictions agree with the observed frequency of positive cases given the predictions. An ideally reliable prediction should be the main diagonal upward line. Well-calibrated predictions should produce curves close to this line, yielding a small reliability term (which is a weighed average of the squared vertical distances between the curve and the main diagonal). Lines lying entirely above (below) represent positive (negative) biased predictions. Flat lines or line segments represent regions of poor resolution.

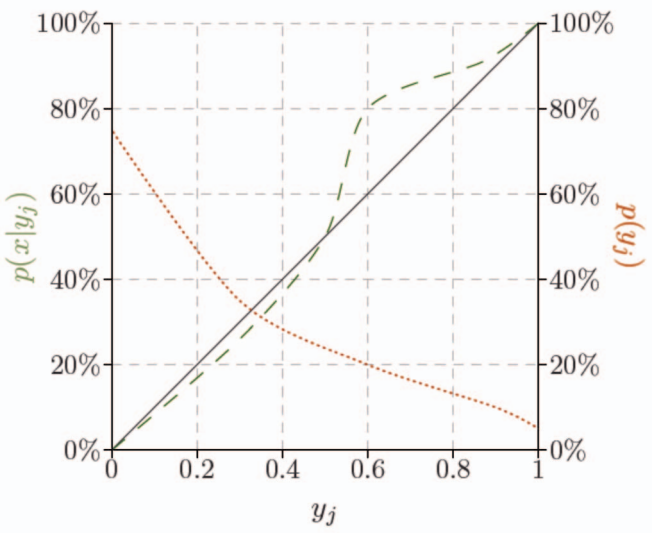

Fig. 8. Reliability diagram.

The second curve in the reliability diagram is a plot of $p(y_j)$ distribution by j, and is related to the refinement component of the Brier score. The dispersion of the refinement distribution reflects the overall confidence of the predictive system: predictions which are often extreme (i.e., specifying probabilities close to 0 or 1) show high confidence. On the other hand, predictions that deviate rarely and little from their average value show little confidence.

Fig. 8 is an example of a reliability diagram. The green dashed line is relative to the calibration component while the dotted red line corresponds to the refinement component. As can be observed from the graph, apart from a deviation around $j = 0.6$ and $j = 0.8$, the predictions are quite reliable. Furthermore, the predictions present a good degree of refinement, as the $p(y)$ distribution is concentrated at lower values of j.

5.3 Attributes Diagram

The Brier skill score (BSS) is a normalization of the Brier score with respect to some reference score, given by

$$BSS = 1 - \frac{BS}{BS_{ref}},$$

and it is aimed to assess the relative improvement of the predictive system over some reference predictive system. The Brier skill score also has a calibration-refinement decomposition, which corresponds to the reliability and resolution terms divided by the first term of (9)

$$BSS = \frac{Resolution - Reliability}{Uncertainty} = RES_{ref} - REL_{ref}.$$

The attributes diagram [32] is an extension of the reliability diagram in order to provide some reference lines related to the algebraic decomposition of the Brier score and

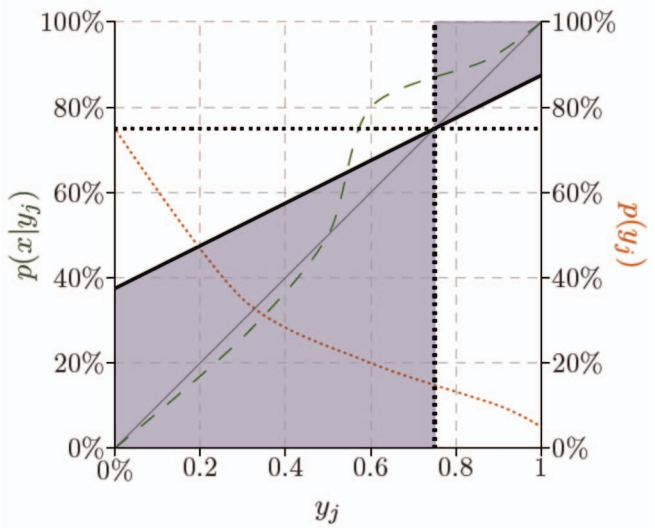

Fig. 9. Attributes diagram.

the Brier skill score, using the trivial (which always assigns probabilities equal to the class proportion) predictive system as a reference. The purpose of the attributes diagram is to provide a geometrical framework that incorporates other aspects (or "attributes," as called in weather forecasting verification) of interest when evaluating probabilistic predictive systems. The attributes diagram is basically the reliability diagram with the addition of three lines.

The first one is the no-resolution line (the line where the resolution term in (9) is equal to zero). This is a horizontal line where $p(x|j) = p(x)$, i.e., the proportion of cases in each subsample j is the same as the overall population. Points falling into the nonresolution line indicate predictions y_i that are unable to discern if a case is more or less likely to be a positive case than chance. The nonresolution line intercepts the main diagonal at $(p(x), p(x))$. On the other hand, the line with maximum resolution is the vertical line, intercepting the main diagonal at the same point. This is the second reference line in the attributes diagram. The last one is the line forming a 45 degree between the main diagonal and the nonresolution line, which represents the points where the accuracy is lower than the predictive model used as a reference (the Brier skill score is zero, and thus $RES_{ref} = REL_{ref}$).

An example of an attributes diagram is shown in Fig. 9. It is the same reliability diagram shown in Fig. 8 with the addition of the three reference lines, using the trivial model as a reference (classifying all cases as positive with probability 1).

The shaded region in Fig. 9 has the following interpretation: points outside this region contribute positively to the Brier score. The demonstration is rather tricky and can be found in [32]. Nevertheless, the intuition is that when the point falls outside this region, the reliability term is larger than the resolution term (recall that the 45 degree line bisecting the main diagonal and the nonresolution line refers to points where $RES_{ref} = REL_{ref}$), and thus contributes positively to the Brier score.

5.4 Discrimination Diagram

The discrimination diagram is a plot of $p(y)$ by $p(y|x)$. As the depicted functions are conditional on X, each discrimination diagram contains two curves—one for $p(y|x)$ and another for $p(y|\overline{x})$. Thus, the discrimination diagram consists of superimposed plots of the two likelihood distributions as functions of the forecast probability y. Ideally, the distributions $p(y|x)$ and $p(y|\overline{x})$ would not overlap. In this case, it is possible to discriminate perfectly between the two classes. In real-world problems, however, generally there is some amount of overlap.

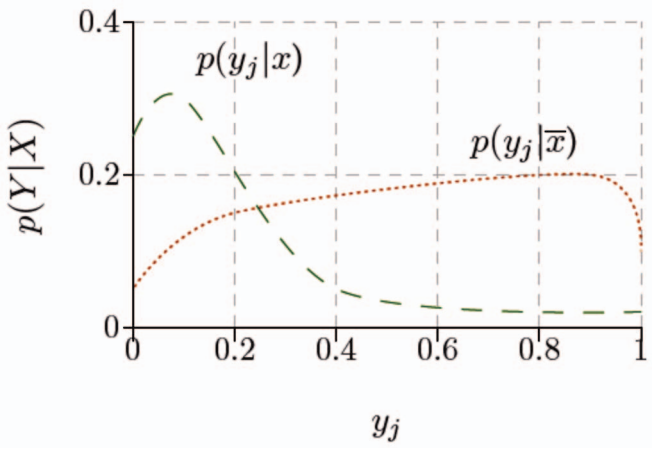

Fig. 10. Discrimination diagram.

Fig. 10 shows an example of a discrimination diagram. The red dotted line represents the density function for the positive class, and the green dashed line represents the density function for the negative class. Analyzing this figure, some pieces of information might be obtained. Clearly, the conditional probabilities given to positive cases are greater for smaller prediction probabilities, and the conditional probabilities given to negative cases are greater for intermediate and larger probability predictions. The likelihood distributions in Fig. 10 overlap somewhat, although they show substantial nonoverlapped areas, indicating substantial separation between positive and negative cases.

The separation of the two likelihoods in a discrimination diagram can be summarized by the difference between their means, called the discrimination distance [33]. This distance is zero if the two likelihood distributions are the same (i.e., if the predictive system cannot discriminate between the classes), and increases as the two likelihood distributions become more distinct. In the limit, this distance is equal to 1 for perfect predictions.

The discrimination diagram has some similarities with the ROC curve. The discrimination distance is equivalent to a parametrized version of the AUC, assuming normal distribution for the likelihoods. Furthermore, the overlapped area is equivalent to the AUC of the corresponding ROC graphs. In fact, the ROC graph is a plot of the two conditional probabilities in the discrimination diagram. However, as the ROC curve does not take into account the face value of the predictions, in a sense, the discrimination diagram has more information than a ROC graph, especially if we take the face predicted value as an estimator of how good a prediction is. On the other hand, ROC graphs have the intuitive interpretation of dominance of one predictive system over another, which cannot be easily generalized for the discrimination diagram.

6 Discussion

Graphical diagrams can be used to analyze the quality of predictive systems. These tools provide a means of identifying possible deficiencies of performance in the predictive models—for example, a tendency to perform well in one of the classes and poorly in the other. Furthermore, the feedback provided by these methods can be used to guide further research to improve predictions. The methods presented in this paper focus on different aspects of prediction, and therefore are suitable for different analyses. The key point of selecting an appropriate graphical method is to identify its strengths and weaknesses. Another point is to know which prediction aspect we are interested in evaluating.

6.1 Advantages and Drawbacks

The main advantages and drawbacks of each graphical method described in this paper, as well as their corresponding evaluating aspects, are summarized next.

- **ROC graph.** By decoupling performance evaluation in true and false positive rates, a ROC graph facilitates the performance evaluation for each class. As long as the class conditional likelihoods do not change, this evaluation is independent from class prevalence and misclassification costs. Furthermore, the ROC convex hull can be used to rule out suboptimal classifiers for all combinations of class prevalence and misclassification costs. A drawback is that the error rate (or expected cost) cannot be easily obtained from the graph.
- **Cost lines.** The aim of cost lines is primarily to facilitate the assessment of the classifier performance in terms of error rate (or expected cost). This is accomplished by graphing the normalized prevalence of positives weighed by its respective cost on the x-axis by the normalized expected cost on the y-axis. As ROC graphs, this evaluation is independent from class prevalence and misclassification costs, as long as the class conditional likelihoods do not change and the suboptimal classifiers lie outside the lower envelope formed by the optimal ones (under different operational conditions). However, although this approach overcomes the difficulty in reading the expected cost in a ROC graph, it does not allow for the evaluation for each class.
- **ROC curves.** ROC curves inherit the qualities and drawbacks of ROC graphs. Furthermore, as the process of derivating the curve involves ordering the cases according to their likelihood to be in the positive class, a ROC curve can evaluate how cases are ranked disregarding the proportion of cases between the classes. The area under the ROC curve can also be used as an index of the quality of the predictive system instead of the error rate or expected cost.
- **Cost curves.** Like the ROC curve, the cost curve inherits the qualities and drawbacks of the cost lines. Another drawback is that the number of cost lines which forms the cost curve is quite large (there are as many lines as there are points in the corresponding cost curve), which makes the visualization of a single line clumsy. Furthermore, although quite useful for analyzing overall classification performance, the cost curves are not as appropriate as ROC curves for evaluating ranking performance.
- **Precision-recall curves.** The precision-recall curve is an interesting approach if we are primarily interested in analyzing performance in one of the classes, especially when dealing with highly skewed domains. However, the precision-recall curve is dependent on the class prevalence, as the curve changes with different class proportions.
- **Lift curve.** A lift curve can be seen as a variation of the ROC curve. It is very useful to visualize how many cases of the positive class will be correctly classified as we make the criteria to assign a positive classification more lenient, since it shows the relation between the top ranked examples and the number of positive examples. A drawback is that lift curves are dependent on class distributions.
- **ROI curve.** A ROI curve can be seen as a specialization of cost curves, without taking into account normalized costs. It can be very useful when costs are known, as it can be used to determine optimal response rates. However, its use is limited to situations where costs can be determined in advance.
- **Reliability diagram.** A reliability diagram can be used to evaluate whether a probabilistic predictive system is well calibrated or not, and how many of these predictions are spread over the prediction variable Y. To this end, it uses calibration and refinement decomposition of the Brier score. Reliability diagrams reveal reliability by plotting the observed relative frequencies versus the forecast values. However, this approach is sensitive to the amount of different forecast values.
- **Attributes diagram.** An attributes diagram extends the reliability diagram to incorporate three lines used as a reference (generally related to the trivial classifier) aimed to assess how the predictions improve with respect to this reference. These lines form a region in which points falling outside this region degrade the Brier score, and thus have poor probability estimates. However, the reference classifier must be specified.
- **Discrimination diagram.** A discrimination diagram is a plot of the class conditional likelihoods as a function of predictions. It can be used to show the behavior of these conditional likelihoods to discriminate among the classes. It has some similarities with the ROC curve, although the discrimination diagram makes the face value of the prediction explicit, while in the ROC curve this is an implicit parameter. However, the dominance of one prediction over another cannot be easily evaluated as in the ROC space.

As can be seen from this summary, various methods have some properties in common. Furthermore, it is not possible to claim the superiority of one method over the others, as they generally produce complimentary views about the predictive systems performance. In other words, a complete analysis for a given problem should consider more than one graph in order to have a clear picture of the performance of the predictive systems under evaluation.

6.2 Limitations

Although graphical methods have numerous interesting properties, they have some limitations, briefly discussed in this section.

- **Necessity of visual inspection.** One of them is that evaluation by means of graphics requires visual inspection, as well as interpretation of the results and may carry out some degree of subjectiveness. On the other hand, as scalars are totally ordered, scalar measures are objective, and results are easier to be compared directly than when using graphical methods (when comparing two predictive models, one should only compare two numbers and choose the predictive model which produced the best score rather than analyzing and interpreting two different curves). However, as argued throughout this paper, evaluation has numerous concerns that should be considered, and it is not possible to capture all these concerns in a single scalar measure. Furthermore, graphical analysis provides a health of information when compared to a single scalar measure. For instance, when comparing two predictive models, it is more valuable to know that the curve corresponding to one of them dominates the other in the ROC space than to know that the area under the ROC curve of one of them is higher than the other. The former (dominance) imply the latter (higher AUC), but the latter does not implies the former. Another example is that two predictive models with the same AUC may differ in their curves.
- **Difficulty of comparison among different data sets.** This requirement of visual inspection also poses some difficulties in conducting experiments where many data sets are used. Comparing many predictive system over the same data set is straightforward. For a single data set, the user can plot as many predictive systems as she/he wants in the same graph. This might make the interpretation difficult, as plotting too many lines in the same graph might make it unclear, but it does not imply increasing the number of graphs. On the other hand, when the user has numerous data sets to test, she/he should plot one graph for each data set. This is because, for the graphs discussed in this paper, it is meaningless to plot curves from different data sets in the same graph.
- **Multi-class problems.** Another drawback is that the graphical tools discussed in this paper are limited to two class problems. In some cases, this might be overcome by using the one-against-all approach, where one of the classes is designated as the "positive" class, while the remaining classes are collapsed as the "negative" class. If necessary, the process can be repeated by choosing a different class as the positive one. Another approach is to extend the methods to the multiclass problem. In [34], an extension of reliability diagrams to the multiclass setting is presented. However, for some graphs, this might not be a trivial task. In the ROC graph, for instance, the complexity of constructing the graph grows exponentially as the number of classes increases. Some approaches to three-class problems were developed [35]. Nevertheless, for a large number of classes alternative approaches should be used to reduce the number of dimensions, as the one proposed in [36], for example.
- **Variability of the test set.** Finally, it should be kept in mind that these graphs are constructed from a test set which is a sample of a population. Therefore, it is expected to have some uncertainty due to this particular sample, and inference techniques should be used to extrapolate from this sample to the population. In general, resampling techniques can often be used to derivate confidence bounds for the graphs, and average techniques can be used to produce curves with error bars, as discussed in [17] for the case of ROC graphs.

7 An Illustrative Example

In this section, we provide an illustrative example of using graphs to analyze classification performance. To this end, we selected "The Insurance Company (TIC) Benchmark" data set [37]. This data set was used in the 2000 meeting of the Computational Intelligence and Learning (Coil '2000) challenge. It contains information about customers consisting of 86 variables which include product usage data and socio-demographic data derived from zip area codes. The data come from a real-world business problem of predicting which customers are potentially interested in buying a caravan insurance policy. The training set contains 5,822 descriptions of customers, including the information of whether or not they have a caravan insurance policy. There are 348 customers who bought the policy and 5,474 who did not. The test set contains 4,000 customers, where 3,762 did not buy the policy and the remaining 238 did.

We used the 3.5.8 version of the Weka data mining suit [38] to generate predictive models for this data set, using only the training data to build the models. These models were then used to assign labels, ranks, and scores to the instances in the test set. To induce the models, we used Naïve Bayes, a Bayesian Network using the BMA estimator [39] and NBTree, a decision tree with Naïve Bayes as a probability estimator on the leaves [40]. All other parameters were left with default values.

First, we consider only the assignment of crisp binary labels for the instances in the test set (i.e., the classifier predicts whether or not a customer would buy a caravan insurance policy). The error rates for each model are 21.43 percent for the model induced by Naïve Bayes, 14.77 percent for the Bayesian Network, and 6.05 percent for the NBTree.

Figs. 11 and 12 show the ROC and cost graphs for the models induced for this data set. Analyzing the ROC graph in Fig. 11, we can see that the model induced by the NBTree is the most conservative in terms of predicting that a customer would buy an insurance policy (and thus commits to less false positive errors, although the number of true positive is also low), while the models induced by Naïve Bayes and the Bayesian Network are more lenient (i.e., they achieve a higher true positive rate than the NBTree does paying the price of a larger false positive rate).

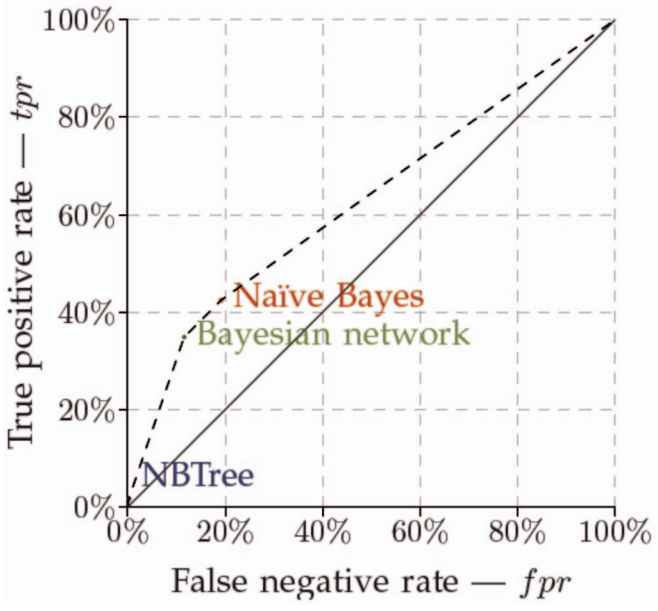

Fig. 11. ROC graph.

The low error rate of the model induced by the NBTree has two reasons. The conservative approach of assigning a positive classification is one of them. Associated to the fact that positive examples have low prevalence in the data set, predicting the negative class more often has advantages in terms of minimizing classification error rates. Indeed, numerous learning algorithms available in Weka (including decision trees, support vector machines, and others) executed using default parameters generated the trivial model of always predicting the negative class.

The dashed line in Fig. 11 shows the ROC convex hull for these three models. As can be seen in the figure, the convex hull is formed by all points. Thus, without knowing the operational conditions we cannot discard any model.

Fig. 12 shows the cost graphs for the three models. As can be seen from the graph, the NBTree has quite a close performance to the classifier which always predicts negative (represented by the main diagonal in the cost graph). The lower envelope (the dashed line in the cost graph) shows the operational condition, expressed in terms of cost-weighed prevalence of positives (PC), in which each classifier is optimal when compared to the others. For PC ranging from 0 to approximately 0.11, the classifier which always predicts negative (all the customers would not buy the insurance policy) is better. The NBTree has an optimal performance for PC ranging from approximately 0.11 to 0.25 (the difference is quite small and it is hard to see it in the graph). The Bayesian network has an optimal performance for PC ranging from approximately 0.25 to 0.50 while Naïve Bayes has optimal performance for PC ranking from 0.50 to 0.58. Predicting positive (all the customers would buy the insurance policy) is the better classifier for PC ranging from nearly 0.58 to 1.

The ROC and cost graphs presented in Figs. 11 and 12 show two different views for crisp binary classifiers. While ROC graphs focus on decoupling positive hits/mistakes, grasping all possible combination of costs and class distribution, cost graphs try to show whether particular

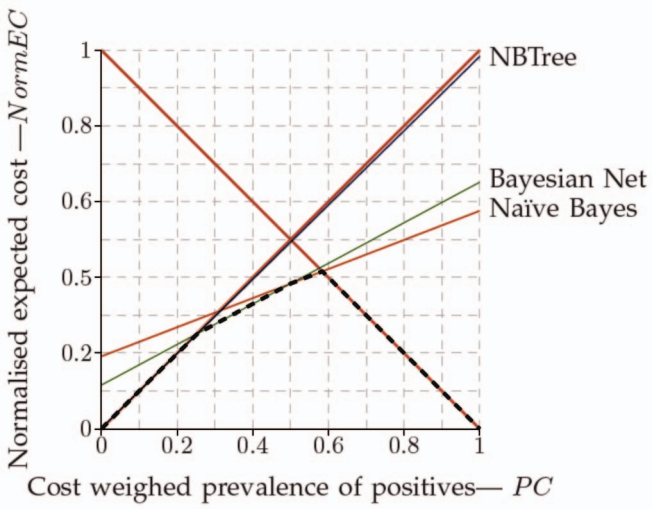

Fig. 12. Cost lines.

combinations of cost and class distribution would commit to a better performance of a particular classifier.

We continue our example by analyzing models where it is possible to order examples rather than only assign a crisp label to them. Fig. 13 shows the ROC curves obtained for each induced model (note that, as we are constructing a ranking instead of only assigning a label, we now have a curve rather than a single point in the ROC space).

Analyzing the curves in Fig. 13, we can see that the model induced by the Bayesian Network dominates the other two in almost the entire space (the line representing the NBTree is slightly over the line representing the Bayesian Network near the lower left corner, and the line representing the Naïve Bayes is over the line of the Bayesian Network when the false positive rate is near 50 percent in the graph). The areas under the ROC curve are 70.7, 68.8, and 65.5 percent for the models induced by the Bayesian Network, Naïve Bayes, and NBTree, respectively.

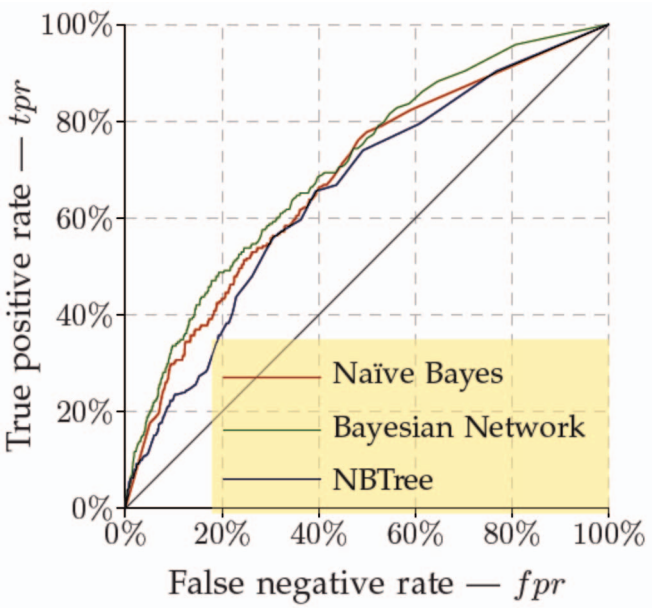

Fig. 13. ROC curves.

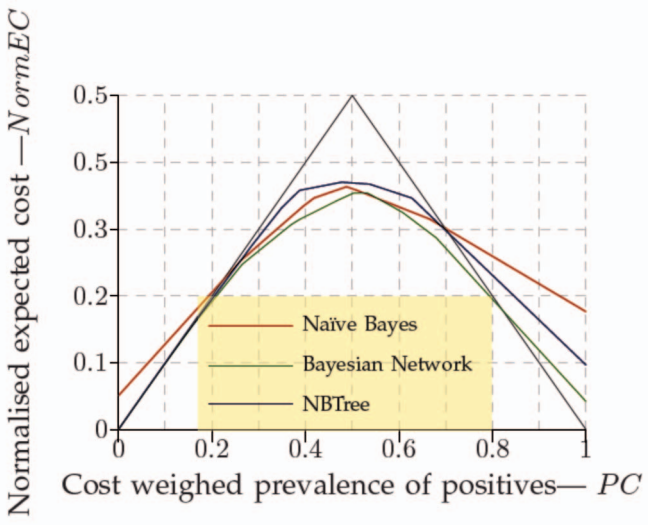

Fig. 14. Cost lines.

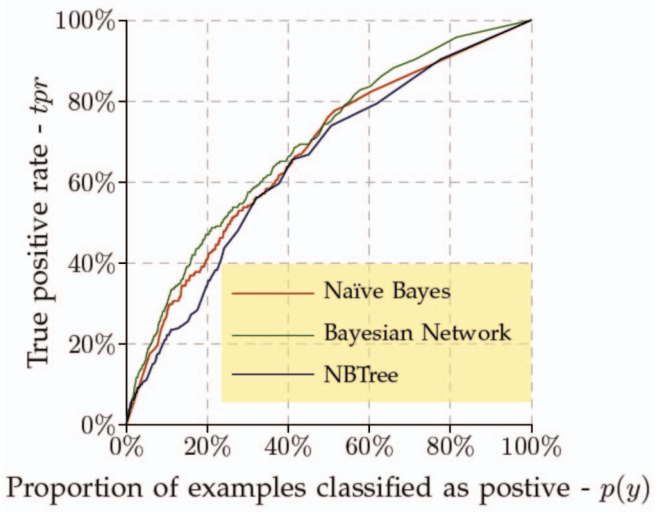

Fig. 16. Lift graph.

Fig. 14 shows the cost curves formed by the lower envelope for the three models. These lines show the operational condition clearly (indicating the combination of class distribution and costs represented by PC in the graph). The models induced by the NBTree and the Bayesian Network have a very similar performance in terms of normalized expected costs ($NormEC$) for PC ranging from 0 to 0.2. They are only slightly better than the classifier which always predicts negative. It is interesting to note that the model induced by Naïve Bayes has a worse performance than always predicting negative for PCs ranging from 0 to 0.25. The model induced by the Bayesian Network dominates (has a lower expected cost) the other three in almost all the PCs ranging from near 0.2 to near 0.8. The model induced by the Naïve Bayes only has a smaller $NormEC$ for PCs close to 0.55. The models induced by the Naïve Bayes and the NBTree have a worse performance than always predicting positive for PCs higher than 0.7, while the model induced by the Bayesian Network has a worse performance than always predicting positive for PCs higher than 0.8.

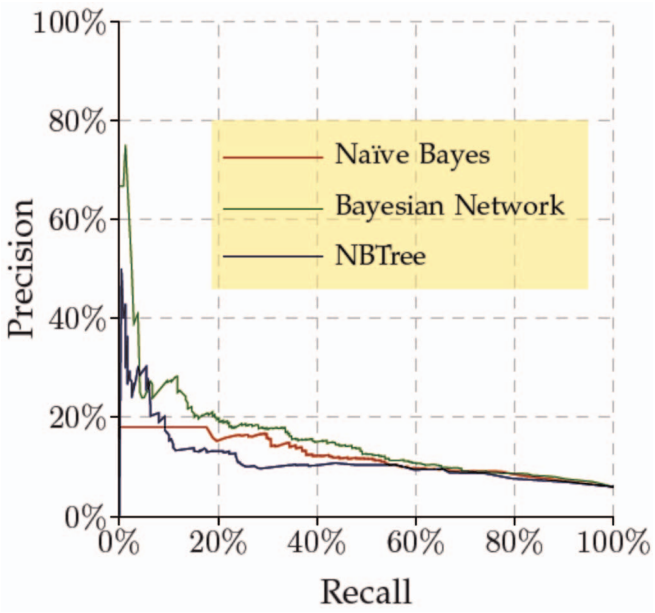

Fig. 15. Precision-recall graph.

The precision-recall curves for the models are shown in Fig. 15. Analyzing the figure, we can see that the Naïve Bayes presents low precision rates when the recall is below 10 percent. In other words, the Naïve Bayes is less effective in distinguishing examples of the positive class at the top part of the rank. Apart from recall rates near 5-10 percent, where the NBTree has higher precision rates, in general, the Bayesian Network has the highest precision rates for almost all levels of recall. However, the precision rate of the NBTree drops sharply and it presents the lower precision rates when the recall is higher than 10 percent. When the recall rate is higher than 70 percent, the precision rate of all models tends to be very similar. One advantage of the precision-recall curves over ROC curves is that the differences at the top part of the rank are highlighted, as illustrated in Fig. 15. In the ROC curve, these points are concentrated in the lower left hand corner of the graph.

Fig. 16 shows the lift curve for the three models. As can be observed the lift curve is very similar to the ROC curve shown in Fig. 13. This is due to the low prevalence of positive cases in the data set.

Assuming a profit of US $50.00 per insurance policy sold and a cost of US $3.00 per mailed customer, Fig. 17 shows the ROI graph for each model evaluated in this example. This graph can be interpreted as the net profit as a function of the mailed customers, according to the ranking given by each model. As can be seen in the graph, the highest profit (about US $3,700.00) would be obtained using the model induced by the Bayesian Network. This profit can be obtained by mailing 20 percent of the top ranked customers. There is a plateau (with a profit slightly oscillating between US $3,500.00 to 3,700.00) between the 20 to 45 percent top ranked customers for this model. A similar profit would be obtained by the model induced by Naïve Bayes only if about 50 percent of the customers were mailed using this model. It is worth noticing that the choice between these two models would depend on the objectives of the insurance company. If the company would like to achieve the highest profit by mailing the lowest number of customers, the model induced by the Bayesian

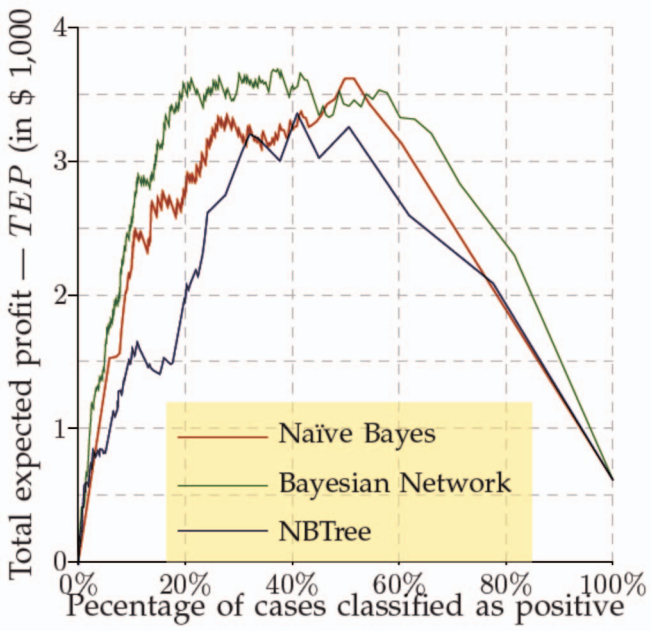

Fig. 17. Return of investment graph.

Network would be the most appropriate. On the other hand, a similar profit could be achieved by mailing customers according to the model induced by Naïve Bayes, in case the company does not care about junk mailing a larger number of customers to get a larger number of positive answers (for instance, to have a larger database in order to offer other products).

We conclude the analysis of this example by examining whether the face numerical values predicted by each model has meaningful information. We start by analyzing the reliability diagram shown in Fig. 18. For each model, this diagram has two lines (the continuous and the dashed one). The continuous line is related to the prediction reliability. Reliability can be interpreted as whether the probabilities predicted by the model mean what they say. In other words, does the accuracy predicted by a given model match the probabilities it predicts? (For instance, does the model

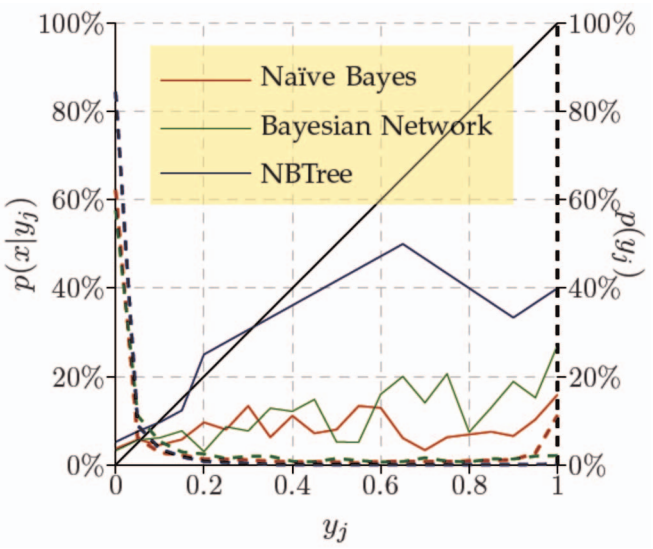

Fig. 18. Reliability diagram.

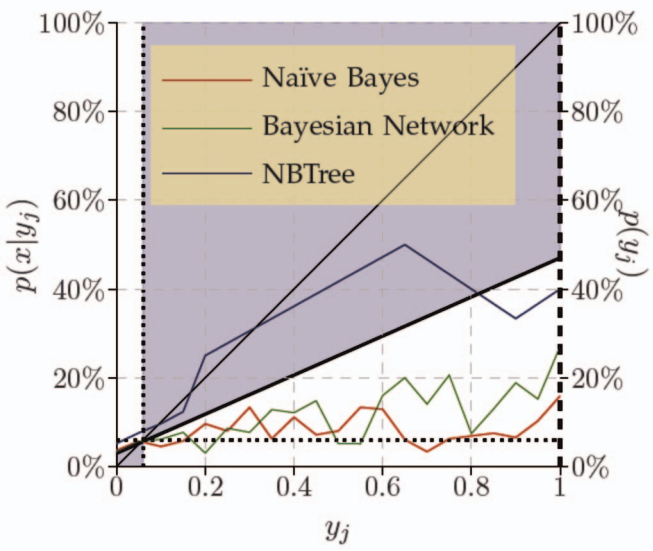

Fig. 19. Attributes diagram.

correctly predict 80 percent of the cases of a given class when it predicts this class with 80 percent probability?). A perfect reliable model is represented by the ascending diagonal in the graph. As can be seen in Fig. 18, the models are overconfident, i.e., the accuracy of the forecasts is generally lower than the probability they predict. The most reliable is the model induced by the NBTree, while the models induced by the Naïve Bayes and the Bayesian Network have similar reliability. Observe that all methods have similar refinement, as can be seen by the dashed lines in the graph shown in Fig. 18. The model induced by Naïve Bayes is the one which has better refinement, as it has peaks either near zero or one, while values within them form a valley, forming a "U" shaped curve.

Fig. 19 shows the attribute diagram for the three models. As described earlier, this graph is the reliability diagram with additional lines added so that it can be analyzed when compared to a model which always predicts the positive class. For the sake of visualization, the refinement lines were removed. The graph makes it clear why the NBTree has a better resolution, as it has a larger proportion inside the gray shaded area (recall that the proportion outside this line contributes to increasing the Brier Score). On the other hand, the lines representing the Naïve Bayes and the Bayesian Network models are almost entirely outside the gray shaded region, and thus have a higher Brier Score than the model induced by the NBTree.

The last graph presented in this section is the discrimination diagram, shown in Fig. 20. For the sake of visualization, we present three graphs, one for each induced model. Figs. 20a, 20b, and 20c correspond to the discrimination diagrams of the models induced by the Naïve Bayes, the Bayesian Network, and NBTree, respectively. The continuous lines correspond to the positive class, while the negative class is represented by the dashed line. As can be seen from the graphs, the three models have poor resolution, as the lines overlap a wide proportion in the three graphs. Furthermore, the graphs indicate that a hybrid approach which combines the three models would lead to an increase in performance. This is because Naïve Bayes has a better discrimination at the

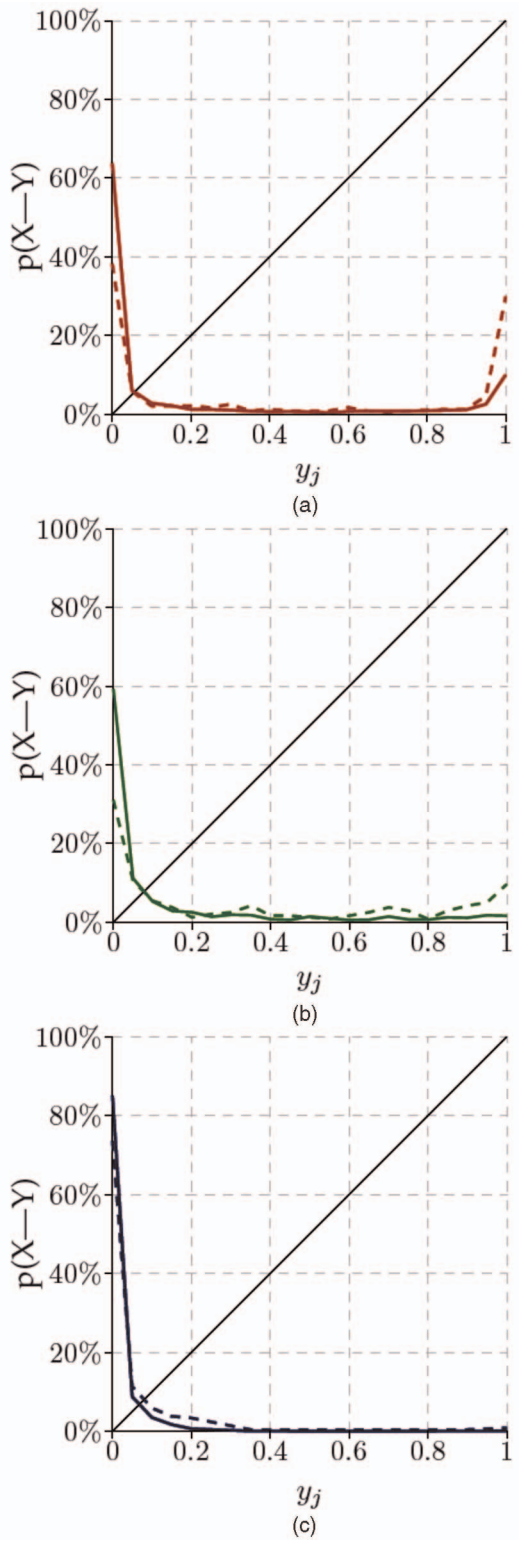

Fig. 20. Discrimination graph. (a) Naïve Bayes. (b) Bayesian Network. (c) NBTree.

ends, NBTree has a better discrimination when $p(y)$ is around 20 percent, and the Bayesian Network has a region with better discrimination when $p(y)$ ranges from 60 to 80 percent. This hybrid approach should map the values of $p(y)$ to the corresponding values of the classifier scores, and then choose the corresponding classifier according to the values corresponding to the best discrimination ranges for the classifiers.

This example illustrates how evaluating performance has several aspects which could be taken into account when evaluating a predictive model. Although the NBTree has the lowest error rate among the three induced models, this is due to its conservative approach which predicts the majority class more often. Nevertheless, the NBTree might not be a better choice if we take different cost scenarios for this problem. Indeed, it has the lowest area under the ROC curve, and the ROC and Cost graphs show that the Bayesian Network dominates the NBTree model and has a performance similar or better than the model induced by Naïve Bayes. The scores predicted by each model do not predict reliable probabilities and should not be interpreted as such. Overall, the models have good refinement although they present poor discrimination ability. As stated earlier, the graphs also indicate that a hybrid model could be useful in increasing the classification performance for the problem.

8 Concluding Remarks

This paper presents a review of various graphical tools that can be used when evaluating classification predictive systems. These methods have obvious benefits for machine learning and data mining research, as they are able to provide detailed feedback on a number of performance dimensions, as well as suggesting ways in which predictive systems can be improved. Furthermore, they are also valuable to machine learning and data mining practitioners as they can help them to understand various characteristics of different systems and to choose among them.

Summarizing, on one hand, if we are primarily interested in the discriminability of the predictive system, then the face values of the predictions are not important. In this case, we can use ROC graphs to evaluate performance in each class, and precision-recall graphs if we are interested in one (often low prevalent) class. Furthermore, we might analyze the expected cost or profit of each model using cost curves, lift, and ROI graphs. On the other hand, if we are interested in reliable and calibrated predictions, then the face values of predictions are important. In this case, we should use reliability and attributes diagrams.

By presenting various graphical performance evaluation methods in the same framework, we hope this paper might shed some light on deciding which methods to use. Nevertheless, we would like to point out that the methods presented in this paper are by no means exhaustive. Other methods may supplement those presented here and may provide additional insights into the behavior of predictive systems. Moreover, specific needs may motivate extensions to these methods.

Acknowledgments

This research was supported by the Brazilian Research Councils FAPESP, CNPq, and Fundação ParqueTecnológico Itaipu—FPTI/Brazil.

References

[1] C. Schaffer, "A Conservation Law for Generalization Performance," *Proc. 11th Int'l Conf. Machine Learning (ICML '94)*, pp. 259-265, 1994.

[2] D.H. Wolpert, "The Lack of a Priori Distinctions between Learning Algorithms," *Neural Computation,* vol. 8, pp. 1341-1390, 1996.
[3] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining.* Springer, 2009.
[4] A.M. Martínez and M. Zhu, "Where are Linear Feature Extraction Methods Applicable?" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1934-1944, Dec. 2005.
[5] M. Zhu and A.M. Martínez, "Subclass Discriminant Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 8, pp. 1274-1286, Aug. 2006.
[6] F.J. Provost, T. Fawcett, and R. Kohavi, "The Case against Accuracy Estimation for Comparing Induction Algorithms," *Proc. 15th Int'l Conf. Machine Learning (ICML '98),* pp. 445-453, 1998.
[7] J.C. Xue and G.M. Weiss, "Quantification and Semi-Supervised Classification Methods for Handling Changes in Class Distribution," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '09).* pp. 897-906, 2009.
[8] D.J. Hand, "Measuring Classifier Performance: A Coherent Alternative to the Area under the Roc Curve," *Machine Learning,* vol. 77, no. 1, pp. 103-123, 2009.
[9] P. Datta, "Business Focused Evaluation Methods: A Case Study," *Proc. Third European Conf. Principles of Data Mining and Knowledge Discovery (PKDD '99),* pp. 316-322, 1999.
[10] C. Drummond, W. Elazmeh, N. Japkowicz, and P. Cochairs, "2006 AAAI Workshop Evaluation Methods for Machine Learning," Technical Report WS-06-06, AAAI press, 2006.
[11] C. Drummond, W. Elazmeh, N. Japkowicz, and S.A. Macskassy, "2007 AAAI Workshop Evaluation Methods for Machine Learning II," Technical Report WS-07-05, AAAI Press, 2007.
[12] W. Klement, C. Drummond, N. Japkowicz, and S. Macskassy, "The Third Workshop Evaluation Methods for Machine Learning," http://www.site.uottawa.ca/ICML09WS/index.html, 2008.
[13] W. Klement, C. Drummond, N. Japkowicz, and S. Macskassy, "The Fourth Workshop Evaluation Methods for Machine Learning," *Proc. 26th Ann. Int'l Conf. Machine Learning (ICML '09),* http://www.site.uottawa.ca/ICML09WS/index.html, 2009.
[14] C. Drummond, "Machine Learning as an Experimental Science (Revisited)," *Proc. AAAI Workshop Evaluation Methods for Machine Learning (Technical Report WS-06-06),* 2006.
[15] C. Drummond and N. Japkowicz, "Warning: Statistical Benchmarking is Addictive. Kicking the Habit in Machine Learning," *J. Experimental and Theoretical Artificial Intelligence,* vol. 22, no. 1, pp. 67-80, 2009.
[16] J. Davis and M. Goadrich, "The Relationship between Precision-Recall and ROC Curves," *Proc. 23rd Int'l Conf. Machine Learning (ICML '06),* pp. 233-240, 2006.
[17] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006.
[18] M.C. Monard and G.E.A.P.A. Batista, "Graphical Methods for Classifier Performance Evaluation," *Proc. Advances in Logic, Artificial Intelligence and Robotics (LAPTEC '2003),* pp. 59-67, 2003.
[19] C. Drummond and R.C. Holte, "Cost Curves: An Improved Method for Visualizing Classifier Performance," *Machine Learning,* vol. 65, no. 1, pp. 95-130, 2006.
[20] L. Torgo and J. Gama, "Regression Using Classification Algorithms," *Intelligent Data Analysis,* vol. 1, nos. 1-4, pp. 275-292, 1997.
[21] S. Rosset, C. Perlich, and B. Zadrozny, "Ranking-Based Evaluation of Regression Models," *Knowledge and Information Systems,* vol. 12, no. 3, pp. 331-353, 2007.
[22] K.-A. Toh and H.-L. Eng, "Between Classification-Error Approximation and Weighted Least-Squares Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 4, pp. 658-669, Apr. 2008.
[23] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Wadsworth Int'l Group, 1984.
[24] C. Elkan, "The Foundations of Cost-Sensitive Learning," *Proc. 17th Int'l Joint Conf. Artificial Intelligence (IJCAI '01),* pp. 973-978, 2001.
[25] P. Domingos, "Metacost: A General Method for Making Classifiers Cost-Sensitive," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99),* pp. 155-164, 1999.
[26] A.P. Bradley, "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern Recognition,* vol. 30, no. 7, pp. 1145-1159, 1997.
[27] N.M. Adams and D.J. Hand, "Comparing Classifiers When the Misallocation Costs are Uncertain," *Pattern Recognition,* vol. 32, no. 7, pp. 1139-1147, 1999.
[28] C.X. Ling and C. Li, "Data Mining for Direct Marketing: Problems and Solutions," *Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98),* pp. 73-79, 1998.
[29] G.W. Brier, "Verification of Forecasts Expressed in Terms of Probability," *Monthly Weather Rev.,* vol. 78, no. 1, pp. 1-3, 1950.
[30] A.H. Murphy, "A New Vector Partition of the Probability Forecasts," *J. Applied Meteorology,* vol. 12, no. 4, pp. 595-560, 1976.
[31] I. Cohen and M. Goldszmidt, "Properties and Benefits of Calibrated Classifiers," *Proc. Eighth European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '04),* pp. 125-136, 2004.
[32] W. Hsu and A.H. Murphy, "The Attributes Diagram: A Geometrical Framework for Assessing the Quality of Probability Forecasts," *Int'l J. Forecasting,* vol. 2, no. 3, pp. 285-293, 1986.
[33] D.S. Wilks, *Statistical Methods in the Atmospheric Sciences,* second ed. Elsevier, 2006.
[34] T.M. Hamill, "Reliability Diagrams for Multicategory Probabilistic Forecasts," *Weather and Forecasting,* vol. 12, no. 4, pp. 736-741, 1996.
[35] D. Mossman, "Three-Way ROCs," *Medical Decision Making,* vol. 19, no. 1, pp. 78-89, 1999.
[36] T.C. Landgrebe and R.P. Duin, "Efficient Multiclass ROC: Approximation by Decomposition via Confusion Matrix Perturbation Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 5, pp. 810-822, May 2008.
[37] P. van der Putten and M. van Someren, Eds., *CoIL Challenge 2000: The Insurance Company Case.* Published by Sentient Machine Research, http://www.liacs.nl/~putten/library/cc2000/, 2000.
[38] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Morgan Kaufmann, 2005.
[39] R.R. Bouckaert, "Bayesian Networks in Weka," Technical Report 14/2004, Computer Science Dept., Univ. of Waikato, 2004.
[40] R. Kohavi, "Scaling up the Accuracy of Naïve Bayes Classifiers: A Decision-Tree Hybrid," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD '96),* pp. 202-207, 1996.

Ronaldo C. Prati received the PhD degree in computer science in 2006. He is a reader in computer science at Centro de Matemática, Computação and Cognição, Universidade Federal do ABC, Santo André, São Paulo, Brazil. His research interests include machine learning and data mining.

Gustavo E.A.P.A. Batista received the PhD degree in computer science in 2003. He is a reader in computer science at Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, São Carlos, São Paulo, Brazil. His research interests include machine learning and data mining.

Maria Carolina Monard received the PhD degree in informatics in 1980. She is a professor of computer science at Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, São Carlos, São Paulo, Brazil. She is the leader of Laboratório de Inteligência Computacional (LABIC). Her research interests include machine learning and data mining.