Mach Learn (2006) 65:95–130
DOI 10.1007/s10994-006-8199-5

Cost curves: An improved method for visualizing classifier performance

Chris Drummond · Robert C. Holte

Received: 18 May 2005 / Revised: 23 November 2005 / Accepted: 5 March 2006 / Published online: 8 May 2006

Abstract This paper introduces cost curves, a graphical technique for visualizing the performance (error rate or expected cost) of 2-class classifiers over the full range of possible class distributions and misclassification costs. Cost curves are shown to be superior to ROC curves for visualizing classifier performance for most purposes. This is because they visually support several crucial types of performance assessment that cannot be done easily with ROC curves, such as showing confidence intervals on a classifier's performance, and visualizing the statistical significance of the difference in performance of two classifiers. A software tool supporting all the cost curve analysis described in this paper is available from the authors.

Keywords Performance evaluation · Classifiers · ROC curves · Machine learning

1. Introduction

Performance evaluation is crucial at many stages in classifier development. The process of designing a new classification algorithm, or extracting a specific model from data, is typically iterative. Each iteration will alter the classifier significantly, so it must be re-evaluated to determine the impact on performance. At the end of the development process, it is important to show that the final classifier achieves an acceptable level of performance and that it represents a significant improvement over existing classifiers.

Editors: Tom Faweett

C. Drummond (✉)
Institute for Information Technology,
National Research Council Canada,Ontario,
Canada, K1A 0R6
e-mail: chris.drummond@nrc-cnrc.gc.ca

R. C. Holte
Department of Computing Science,
University of Alberta, Edmonton,
Alberta, Canada, T6G 2E8
e-mail: holte@cs.ualberta.ca

To evaluate a classifier, we need an estimate of future performance. If the work is motivated by a particular application, some idea of the classifier's eventual use may suggest an application-specific measure. When designing and testing new classifier algorithms in a research setting, the appropriate measure of performance is not so clear. The machine learning community has traditionally used error rate (or accuracy) as its default performance measure. Recently, however, area under the ROC curve (AUC) has been used in some studies (Bradley, 1997; Karwath & King, 2002; Weiss and Provost, 2003; Yan et al., 2003). In cost-sensitive learning, expected cost under a range of cost matrices has been the preferred measure (Bradford et al., 1998; Domingos, 1999; Margineantu & Dietterich, 2000).

The shortcomings of using accuracy have been pointed out by others (Hand, 1997; Provost et al., 1998). The most fundamental shortcoming is the simple fact that a single, scalar performance measure cannot capture all aspects of the performance differences between two classifiers. Even when there are only two classes, there are two different types of errors that can occur in any combination. The performance of a two-class classifier is therefore characterized by a pair of numbers. Any single scalar measurement must lose some of this information, imposing a one-dimensional ordering on what is fundamentally two dimensions of performance. This criticism is not specific to error rate, it applies equally strongly to AUC and to any other scalar performance measure.

A possible attraction of scalar measures is that, when comparing classifiers, the number of wins, losses and draws can be easily counted and tabulated. This often gives an apparently definitive answer to which classifier is better, allowing authors to claim their algorithm is the best overall. We feel however that a serious shortcoming of scalar measures is that they fail to give any indication of the circumstances under which one classifier outperforms another. Scalar performances are totally ordered, leading to conclusions that one classifier is either better or worse than another, or that there is no significant difference. Yet it often happens that one classifier is superior to another in some circumstances and inferior in others, and existing performance measures give no assistance in identifying the circumstances in which a particular classifier is superior.

An important example of this failing occurs when the cost of misclassifying examples in one class is much different than the cost of misclassifying examples in the other class, or when one class is much rarer than the other (Japkowicz et al., 1995; Kubat et al., 1998; Ling & Li, 1998). A scalar measure can give the expected performance given a probability distribution over costs and class ratios, but it will not indicate for which costs and class ratios one classifier outperforms the other. Adams and Hand (1999) emphasize this point, and specifically mention AUC as being no better than other scalar measures in this regard:

> "... AUC is not an ideal measure of performance. If the ROC curves of two classifiers cross, then one classifier will be superior for some values of the cost ratio and the other classifier will be superior for other values. If it were known that the actual cost ratio for a problem led to a classification threshold which lay to one side or the other of the crossing point, even if one did not know the precise position of the actual cost ratio, then a summary measure integrating over all possible values of the cost ratio would be worthless." (p. 1141)

An alternative to scalar measures which overcomes all these difficulties when there are two classes has been known for decades: ROC plots. An ROC plot is a two-dimensional plot, with the misclassification rate of one class ("negative") on the x-axis and the accuracy of the other class ("positive") on the y-axis. Not only does an ROC plot preserve all performance-related information about a classifier, it also allows key relationships between the performance of several classifiers to be seen instantly by visual inspection. For example, if classifier C1

"dominates" classifier C2 (has better accuracy on both classes) C1's ROC plot will be above and to the left of C2's. If C1 is superior to C2 in some circumstances but inferior in others, their ROC plots will cross. Interpreted correctly, ROC plots show the misclassification costs of a classifier over all possible class distributions and all possible assignments of misclassification costs[1].

ROC analysis was introduced to the data mining and machine learning communities by Provost and Fawcett (1997). Despite its advantages, it has not been adopted as the standard method of performance evaluation in either of these scientific communities even though two-class classification is an extremely common task in the research literature.

We believe the reason for this is that, despite its strengths, ROC analysis is inadequate for the needs of data mining and machine learning researchers in several crucial respects. It does not allow researchers to do the kind of analysis they currently do with scalar measures. In particular, ROC plots do not allow any of the following important experimental questions to be answered by visual inspection:

- what is classifier C's performance (expected cost) given specific misclassification costs and class probabilities?
- for what misclassification costs and class probabilities does classifier C outperform the trivial classifiers that assign all examples to the same class?
- for what misclassification costs and class probabilities does classifier C1 outperform classifier C2?
- what is the difference in performance between classifier C1 and classifier C2?
- what is the average of performance results from several independent evaluations of classifier C (e.g. from 10-fold cross-validation)?
- what is the 90% confidence interval for classifier C's performance?
- for what misclassification costs and class probabilities is the difference in performance between classifier C1 and classifier C2 statistically significant?

In this paper we present a different way of visualizing classifier performance, cost curves (Drummond & Holte, 2000a), that allows all of these questions to be answered instantly by visual inspection, while retaining almost all the attractive features of ROC plots. A software tool based on cost curves is freely available and provides touch-of-a-button visualization for all these questions.[2]

The paper is organized around these questions. After a brief review of ROC curves (see Fawcett (2003) for a more in-depth tutorial), cost curves are introduced. Then a section is devoted to each of the questions. The limitations of cost curves and the circumstances in which ROC curves are more useful than cost curves are discussed in Section 6.

2. ROC curves

The basis for any evaluation or visualization of a 2-class classifier's performance are the numbers in the confusion matrix, as illustrated in Fig. 1(a). The inner bold box is a 2 by 2 confusion matrix, where the rows represent the actual class of an instance and the columns the predicted class. Typically these numbers are obtained by applying the classifier to a set of test

[1] "All" distributions and costs with certain standard restrictions. These are discussed later in the paper.

[2] http://www.aicml.ca/research/demos/content/demo_template.php?num=4. This tool goes far beyond the existing cost curve capabilities implemented in the popular Weka tool set (Witten & Frank, 2005)

Act. \ Pred.	Pos (+)	Neg (−)	
Pos (+)	16	4	20
Neg (−)	4	6	10

Act. \ Pred.	Pos (+)	Neg (−)	
Pos (+)	P(+|+) TP	P(−|+) FN	P(+)
Neg (−)	P(+|−) FP	P(−|−) TN	P(−)

Fig. 1 (a) Example confusion matrix — (b) *TP*, *FP* and other rates

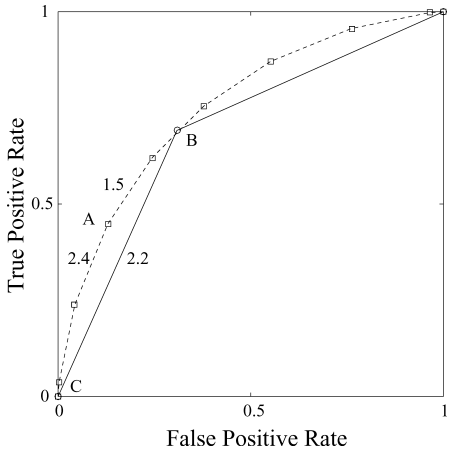


Fig. 2 Example ROC points (A, B, C) and ROC curves (dashed and solid).The numbers are the slopes of the line segments they are beside

examples and counting how many examples fall into each cell. Dividing the entries in each row by the row total gives an estimate of the predictive characteristics of the classifier—the probability that the classifier will make a particular prediction given an example from a particular class, as shown in Figure 1(b). These characteristics are called the "true positive" rate (*TP*), "false positive" rate (*FP*), "true negative" rate (*TN*) and "false negative" rate (*FN*).

ROC space has 2 dimensions, with *TP* on the *y*-axis and *FP* on the *x*-axis. A single confusion matrix thus produces a single point in ROC space. For example, the point labeled B in Fig. 2 is the ROC point for a classier with $FP = 0.35$ and $TP = 0.7$. An ROC curve is formed from a set of such points, such as the points on the dashed curve in Fig. 2. A common assumption in traditional ROC analysis is that these points are samples of a continuous curve in a specific parametric family. Therefore standard curve fitting techniques can be used as means of interpolating between known points (Swets, 1988). In the machine learning literature it is more common to take a non-parametric approach and join the ROC points by line segments, as was done to create both ROC curves in Fig. 2.

The method used to generate the set of ROC points for a given classifier (or learning algorithm) depends on the classifier. Some classifiers have parameters for which different settings produce different ROC points. For example, with naive Bayes (Duda & Hart, 1973; Clark & Niblett, 1989) an ROC curve is produced by varying its threshold parameter. If such a parameter does not exist, algorithms such as decision trees can be modified to include costs producing different trees corresponding to different points (Breiman et al., 1984). The counts at the leaves may also be modified, thus changing the leaf's classification, allowing a single tree to produce multiple points (Bradford et al., 1998; Ferri et al., 2002). Alternatively the class frequencies in the training set can be changed by under- or oversampling to simulate a change in class priors or misclassification costs (Domingos, 1999).

An ROC curve implicitly conveys information about performance across "all" possible combinations of misclassification costs and class distributions, with certain standard restrictions. For class distributions "all" means any prior probabilities for the classes while keeping the class-conditional probabilities, or likelihoods, constant (Webb and Ting, 2005). For costs "all" means all combinations of costs such that a misclassification is more costly than a correct classification.

An "operating point" is a specific combination of misclassification costs and class distributions. Knowing a classifier's error rate across the full range of possible operating points is important for two reasons. One reason is that the class distribution and costs can change with time, or with the location where the classifier is deployed. The second reason is that the distribution in the datasets used for training and evaluating the classifier may not reflect reality. For example, consider the two credit application datasets in the UCI repository (Newman et al., 1998). Positive examples in these datasets represent credit applications that were approved. In the Japanese credit dataset 44.5% of the examples are positive but in the German credit dataset 70% of the examples are positive. This might reflect a true difference in the two countries, but a plausible alternative explanation is that neither proportion reflects reality. An extreme case of a dataset not representing the true class distribution is the Splice dataset in the UCI repository. It has an equal number of positive (classes I/E and E/I) and negative (class "neither") examples, whereas in actual DNA sequences the ratio is more like 1:20 (Saitta & Neri, 1998).

One point in ROC space dominates another if it is above and to the left, i.e. has a higher true positive rate and a lower false positive rate. If point A dominates point B, A will have a lower expected cost than B for all operating points. One set of ROC points, A, dominates another set, B, when each point in B is dominated by some point in A and no point in A is dominated by a point in B.

ROC curves can be used to select the set of system parameters (or the individual classifier) that gives the best performance for a particular operating point (Halpern et al., 1996). In advance of knowing the operating point, one can compute the upper convex hull of the ROC points defined by the system parameters (Provost & Fawcett, 1997). The set of points on the convex hull dominates all the other points, and therefore are the only classifiers that need be considered for any given operating point.

The slope of the segment of the convex hull connecting the two vertices (FP_1, TP_1) and (FP_2, TP_2) is given by the left-hand side of the following equation:

$$\frac{TP_1 - TP_2}{FP_1 - FP_2} = \frac{p(-) * C(+|-)}{p(+) * C(-|+)} \qquad (1)$$

The right-hand side defines the set of operating points where (FP_1, TP_1) and (FP_2, TP_2) have the same expected cost (Hilden & Glasziou, 1996; Provost & Fawcett, 1997). Here $p(a)$ is the probability of a given example being in class a, and $C(a|b)$ is the cost incurred if an example in class b is misclassified as being in class a. The slope of the line joining the ROC points for two classifiers therefore fully characterizes the operating points for which the two classifiers have the same expected cost. If an operating point defines a slope different than this slope, one of the classifiers will outperform the other at this operating point. For example, consider the point labeled A on the dashed ROC curve in Fig. 2. The slopes of the line segments incident to A are the numbers, 1.5 and 2.4, shown in the figure. If the right-hand side of Eq. (1) evaluates to a value between 1.5 and 2.4 classifier A will give the best performance of any of the classifiers on this ROC curve. But if it evaluates to a value outside this range, classifier A will perform worse than one or more other classifiers on this curve.

The solid lines in Fig. 2 are produced by joining the ROC point for classifier B to the ROC points for the trivial classifiers: point (0,0) represents classifying all examples as negative; point (1,1) represents classifying all points as positive. The slopes of these lines define the set of operating points for which classifier B is potentially useful, its operating range. For operating points outside this set, classifier B will be outperformed by one of the trivial classifiers.

ROC analysis does not directly commit to any particular measure of performance. This is sometimes considered an advantageous feature of ROC analysis. For example, Van Rijsbergen (1979) quotes Swets (1967) who argues that this is useful because it measures "discrimination power independent of any 'acceptable criterion' employed". Provost and Fawcett (1998) substantiate this argument by showing that ROC dominance implies superior performance for a variety of commonly-used performance measures. The ROC representation allows an experimenter to see quickly if one classifier dominates another, and, using the convex hull, to identify potentially optimal classifiers without committing to a specific performance measure.

3. Cost curves

Cost curves, unlike ROC curves, are specifically designed for a specific performance measure—expected cost. To begin the discussion of cost curves, we will assume $C(-|+) = C(+|-)$, i.e. that misclassification costs are equal and can therefore be set to 1. In this setting, expected cost is simply error rate and an operating point is defined by $p(+)$, the probability of an example being from the positive class.

It is important to be perfectly clear about the exact meaning of $p(+)$, because in the lifetime of a classifier learned from training data there are at least three distinct sets of examples that might each have a different proportion of positive examples. $p_{train}(+)$ is the percentage of positive examples in the dataset used to learn the classifier. $p_{test}(+)$ is the percentage of positive examples in the dataset used to build the classifier's confusion matrix. $p_{deploy}(+)$ is the percentage of positive examples when the classifier is deployed (put to use). It is $p_{deploy}(+)$ that should be used for $p(+)$, because it is the performance during deployment that we wish to estimate of a classifier. However, because we do not necessarily know $p_{deploy}(+)$ at the time we are learning or evaluating the classifier we would like to visualize the classifier's performance across all possible values of $p_{deploy}(+)$. Cost curves do precisely that.

In the simplified setting where $C(-|+) = C(+|-)$, a cost curve plots error rate as a function of $p(+)$. Error rate is the y-axis in the plot, $p(+)$ is the x-axis. The extreme values on the x-axis represent the situations where all the examples to which the classifier will be applied are in the same class. $x = p(+) = 0$ means that all these examples are negative, $x = p(+) = 1$ means they are all positive. When $x = 0$ a classifier's overall error rate is simply its error rate on the negatives, since no positive examples will be presented to the classifier. When $x = 1$ its overall error rate is its error rate on the positives. Joining these two points by a straight line plots its overall error rate as a function of $p(+)$.

The dashed line in Fig. 3(a) is the estimated error rate of the decision stump produced by 1R (Holte, 1993) for the Japanese credit dataset over the full range of possible $p(+)$ values. The solid line in Fig. 3(a) gives the same information for the decision tree C4.5 (Quinlan, 1993) learned from the same training data. In this plot, we can instantly see the relation between 1R and C4.5's error rate across the full range of deployment situations. The vertical difference between the two lines is the difference between their error rates at a specific operating point. The intersection point of the two lines is the operating point where 1R's stump and C4.5's tree perform identically. This occurs at $x = p(+) = 0.445$. For larger values of $p(+)$ 1R's

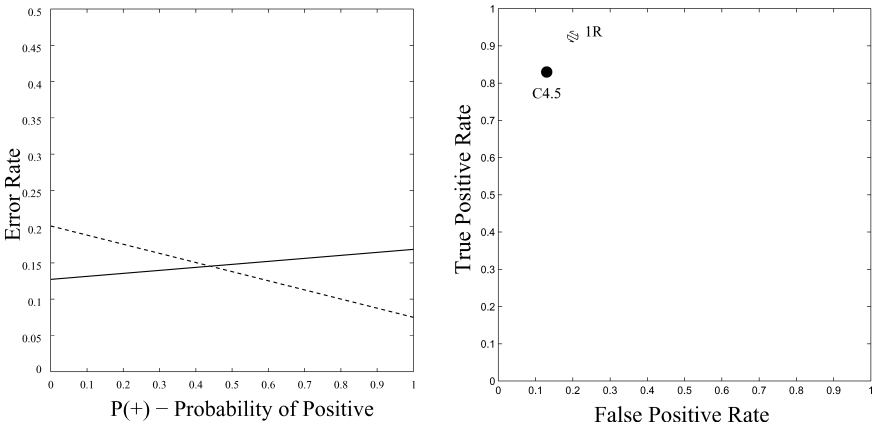


Fig. 3 1R (dashed) and C4.5 (solid) on the Japanese credit data. (a) Error rate as a function of $p(+)$ — (b) Corresponding ROC points

error rate is lower than C4.5's, for smaller values the opposite is true. It is much harder to extract the same information from the corresponding ROC plot (Fig. 3(b)).

Mathematically, ROC curves and cost curves are very closely related: there is a bidirectional point/line duality[3] between them. This means that a point in ROC space is represented by a line in cost space, a line in ROC space is represented by a point in cost space, and vice versa.

The point (FP, TP) in ROC space is a line in cost space that has $y = FP$ when $x = 0$ and $y = FN = 1 - TP$ when $x = 1$. The equation for this line is given by the following equation:

$$Y = error\ \ rate = (FN - FP) * p(+) + FP \qquad (2)$$

A line in ROC space with slope S and y-intercept TP_o is converted to a point in cost space using the following equation:

$$X = p(+) = \frac{1}{1+S} \qquad (3)$$
$$Y = error\ \ rate = (1 - TP_o) * p(+)$$

Both these operations are invertible. Their inverses map points (lines) in cost space to lines (points) in ROC space. Given a point (X, Y) in cost space, the corresponding line in ROC space is:

$$TP = (1/X - 1) * FP + (1 - Y/X)$$

Given a line $aX + b$ in cost space, the corresponding point in ROC space is:

$$FP = b, \quad TP = 1 - (b + a)$$

[3] For an introduction to point/line duality see (Preparata & Shamos, 1988).

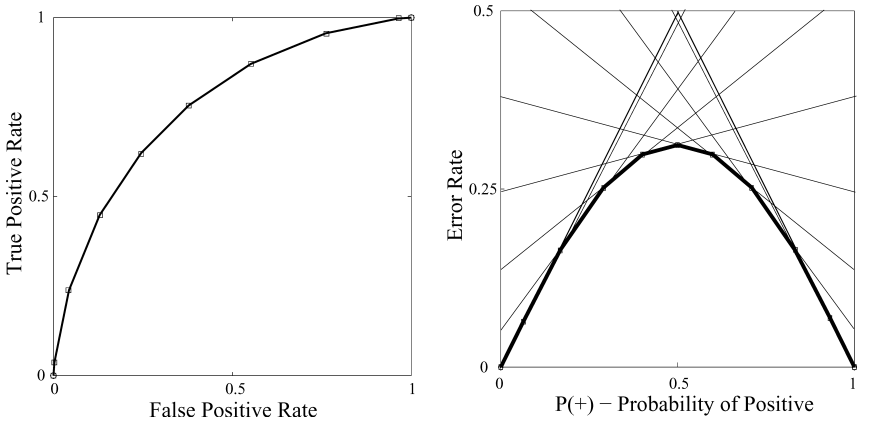


Fig. 4 (a) Ten ROC points and their ROC convex hull — (b) Corresponding set of cost lines and their lower envelope

As discussed in Section 2, the slopes of lines in ROC space play a crucial role in ROC analysis. Equation 3 shows that each slope, S, in ROC space gets mapped to a distinct X value in cost space. The awkward process of visualizing slopes in ROC space therefore becomes the simple visual process of inspecting cost curves at a particular point on the x-axis. Thus our representation exploits certain natural human visual strengths; this is perhaps the main advantage of cost curves.

3.1. The lower envelope of several cost lines

As we have just seen, a single classifier, or confusion matrix, which would be a single point, (FP, TP), in ROC space, is a straight line, joining (0, FP) to (1, FN), in cost space. A set of points in ROC space, the basis for an ROC curve, is a set of cost lines, one for each ROC point. For example, the bold curve in Fig. 4(a) is an ROC curve based on ten ROC points (including the two trivial points, (0,0) and (1,1)). Each of those points becomes a cost line in Fig. 4(b).

The notion of dominance in ROC space has an exact counterpart in cost space. Cost curve $C1$ dominates cost curve $C2$ if $C1$ is below (or equal to) $C2$ for all x values. This means there is no operating point for which $C2$ outperforms $C1$.

The related ROC concept of upper convex hull also has an exact counterpart for cost curves: the lower envelope. The lower envelope at any given x value is defined as the lowest y value on any of the given cost curves at that x. Because of the duality between ROC space and cost space, the line segments making up the lower envelope precisely correspond to the points on the ROC convex hull and the vertices of the lower envelope correspond to the line segments on the ROC hull. The bold line in Fig. 4(b) is the lower envelope of the cost lines in the figure. Visually, the lower envelope stands out very clearly, especially when there are a large number of cost lines.

Just as an ROC curve is a curve constructed piecewise from a set of ROC points, a cost curve is a curve constructed piecewise from a set of cost lines. The lower envelope is one way to construct a cost curve from a set of cost lines; other methods are described in Section 5.

3.2. Taking costs into account

So far in this section, we have assumed that the cost of misclassifying positive examples, $C(-|+)$, is the same as the cost of misclassifying negative examples, $C(+|-)$. We will now eliminate this assumption and explain how to draw cost curves in the general case of arbitrary misclassification costs and class distributions. The only change needed to take costs into account is to generalize the definitions of the x-axis and y-axis.

To derive the general definition of the y-axis in our plots, we start with the formula for expected cost based on the standard cost model (see Appendix A). We assume that all costs are finite. We also assume that the cost of correctly classifying an example is always less than the cost of misclassifying it. We take this as the definition of correctly classifying the example. Thus the costs we consider are always strictly greater than zero. So the best possible classifier, which classifies every example correctly, has an expected cost of 0:

$$E[Cost] = FN * p(+) * C(-|+) + FP * p(-) * C(+|-) \qquad (4)$$

where $p(-) = 1 - p(+)$.

We apply one additional normalization, so that the maximum possible expected cost is 1. The maximum expected cost occurs when all instances are incorrectly classified, i.e. when FP and FN are both one. In this case Eq. (4) simplifies to

$$max\ E[Cost] = p(+) * C(-|+) + p(-) * C(+|-)$$

We define normalized expected cost, `Norm (E[Cost])`, by dividing the right hand side of Eq. (4) by the maximum possible expected cost, resulting in the following equation.

$$Norm(E[Cost]) = \frac{FN * p(+) * C(-|+) + FP * p(-) * C(+|-)}{p(+) * C(-|+) + p(-) * C(+|-)} \qquad (5)$$

With this normalization, a classifier's y-value indicates the fraction of the difference between the maximum and minimum possible costs that will be incurred if the classifier is used. This is the natural extension of error rate to normalized costs.

The x-axis is also redefined to include misclassification costs. We multiply $p(+)$ by $C(-|+)$ and normalize so that x ranges from 0 to 1. We refer to the normalized version of $p(+) * C(-|+)$ as $PC(+)$, "PC" standing for "probability times cost". The following equation is the formal definition (here a is a class, either $+$ or $-$):

$$PC(a) = \frac{p(a) * C(\overline{a}|a)}{p(+) * C(-|+) + p(-) * C(+|-)}$$

By definition, $PC(-) = 1 - PC(+)$. When misclassification costs are equal $PC(+) = p(+)$. In general, $PC(+) = 0$ corresponds to the cases when $p(+) = 0$ or $C(-|+) = 0$. These are the extreme cases when cost is only incurred by misclassifying negative examples - either because positive examples never occur ($p(+) = 0$) or because it costs nothing to misclassify them. $PC(+) = 1$ corresponds to the other extreme, the cases when $p(-) = 0$ or $C(+|-) = 0$.

Rewriting Eq. (5) with $PC(a)$ replacing the normalized version of $p(a) * C(\overline{a}|a)$ produces the following:

$$\texttt{Norm(E[Cost])} = FN * PC(+) + FP * PC(-) \qquad (6)$$

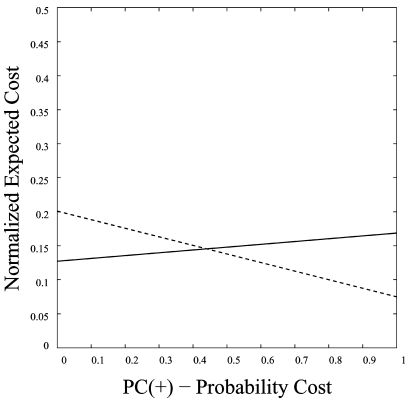


Fig. 5 Japanese credit—cost lines for 1R (dashed line) and C4.5 (solid line)

which shows that normalized expected cost is a linear combination of the false positive and false negative rates.

Because $PC(+) + PC(-) = 1$, we can rewrite Eq. (6) into the more common algebraic formula for a line. This results in

$$\texttt{Norm(E[Cost])} = (FN - FP) * PC(+) + FP$$

which is the straight line representing the normalized expected cost for a classifier. At the limits of the normal range of $PC(+)$ this equation simplifies to the following:

$$\texttt{Norm(E[Cost])} = \begin{cases} FP, & \text{when } PC(+) = 0 \\ FN, & \text{when } PC(+) = 1 \end{cases}$$

To plot a classifier on the cost graph, we draw two points, $y = FP$ when $x = 0$ and $y = FN$ when $x = 1$ and join them by a straight line. Note that this is exactly how we plotted the cost line for a classifier in the simpler setting where costs were ignored. The new cost lines therefore are identical to the ones in the previous sections. All that has changed is the interpretation of the axes. For example, Fig. 5 shows the cost lines for 1R and C4.5 on the Japanese credit dataset taking costs into account. It is identical to Fig. 3(a) except for the labels on the axes and the fact that each line now represents the expected cost of the classifier over the full range of possible class distributions and misclassification costs. We typically restrict the range of the *y*-axis to 0 to 0.5, as the lower half of the figure is typically the most interesting part. Sometimes for clarity we rescale the *y*-axis to range from 0 to 1. We indicate this by adding "(Rescaled Y)" to the captions of the relevant figures.

The conclusions we drew from Fig. 3(a) about which classifier was better for which operating points are therefore equally true, with appropriate re-interpretation, when costs are taken into account. Those conclusions would now read, with the changes in bold font, "In this plot we can instantly see the relation between 1R and C4.5's **normalized expected cost** across the full range of deployment situations. The vertical difference between the two lines is the difference between their **normalized expected costs** at a specific operating point. The intersection point of the two lines is the value of **PC**(+) where 1R's stump and C4.5's tree perform identically. This occurs at x=**PC**(+)=0.445. For larger values of **PC**(+) 1R's **normalized expected cost** is lower than C4.5's, for smaller values the opposite is true."

Likewise, if we simply replace "error rate" and "p(+)" by "normalized expected cost" and "PC(+)", respectively, in Eqs. (2) and (3), we get fully general equations relating ROC space to cost space.

3.3. Related visualization techniques

Some previously published visualization methods are closely related to cost curves: the "regret graphs" (or "tent graphs") of Hilden and Glasziou (1996), the performance graphs used by Turney (1995), and the "Loss Difference Plots" and "LC index" of Adams and Hand (1999).

In a regret graph, exact values for the misclassification costs, $C(-|+)$ and $C(+|-)$, are assumed to be known. Operating points are therefore defined by $p_{deploy}(+)$, which serves as the x-axis for a regret graph. Knowing $C(-|+)$ and $C(+|-)$ also allows the y-axis (performance) to be expressed in absolute terms, rather than the relative cost that is used for the y-axis in cost curves. The only cost normalization is to subtract the minimum possible cost from all costs, so that zero is the minimum possible cost in a regret graph. Thus, regret curves are identical to cost curves with specific values for $C(-|+)$ and $C(+|-)$ substituted into the definitions of `Norm(E[Cost])` and $PC(+)$. If these values are not known exactly, regret graphs cannot be used, but cost curves can.

Hilden and Glasziou (1996) point out that the exact costs of misclassification, in their medical application, can vary from patient to patient, and therefore each patient will have a different regret graph. What cost curve analysis makes clear is that certain conclusions can be drawn about classifiers before costs are known. For example, if classifier C1 dominates classifier C2, there are no costs for which C2 will outperform C1, and the minimum vertical distance between the two cost curves is a lower bound on the relative advantage of C1 over C2.

The clinical decision-making setting in which regret graphs are described provides an interesting alternative interpretation that could be applied to cost curves as well. For example, the x-axis in a regret graph is interpreted as the decision-maker's uncertainty about a binary condition, and regret graphs can be used to assess the usefulness of tests designed to reduce this uncertainty as well as of therapeutic actions.

Turney (1995) discusses the difficulties of using absolute cost in comparative studies involving several datasets and proposes instead to use a normalized cost he calls "standard cost". This differs from our normalized expected cost in a crucial way — "standard cost" chooses as its reference worst-case classifier the trivial classifier that makes the fewest errors, as opposed to the classifier that misclassifies every example. It then multiplies this error rate by the worst possible misclassification cost. We believe this is undesirable for two important reasons. First, it does not simplify to error rate when misclassification costs are equal. Second, it requires knowing which class is more frequent, and knowing the maximum misclassification cost exactly. Like "regret graphs", Turney's graphs do not represent performance across the full range of possible operating points. In Turney's experiments where $C(+|-)$ and $C(+|-)$ are different (Turney's Section 4.2.3), the class probabilities are assumed to be exactly known and the x-axis is simply the cost ratio, $C(-|+)/C(+|-)$, which is plotted on a logarithmic scale to cope with the fact that this ratio can grow arbitrarily large (see Turney's Fig. 5). In cost curves a logarithmic scale is unnecessary because $X = PC(+)$ is normalized to be between 0 and 1.

Adams and Hand (1999) sketch briefly, and then reject, a performance visualization technique, loss difference plots, that differs from cost curves in two respects. The first difference is that their discussion focuses entirely on the misclassification costs, which Adams

and Hand denote c_0 and c_1. Class probabilities are never mentioned after the introductory remarks. This difference can easily be eliminated by defining $c_0 = p(+) * C(-|+)$ and $c_1 = p(-) * C(+|-)$ instead of the literal readings $c_0 = C(-|+)$ and $c_1 = C(+|-)$. With these definitions the x-axis in a loss difference plot, the normalized ratio of the misclassification costs, $c_0/c_0 + c_1$, is identical to the x-axis for cost curves.

The key difference between loss difference plots and cost curves is the *y*-axis (performance). In loss difference plots it is on an absolute scale, as it is in regret graphs. However, Adams and Hand do not assume the exact values for c_0 and c_1 are known, they only assume their ratio is known. Knowing only the ratio of the costs makes absolute cost measurements impossible: $c_0 = 2$ and $c_1 = 1$ has the same ratio as $c_0 = 20$ and $c_1 = 10$, but the absolute cost of the latter is ten times that of the former. This leads Adams and Hand to reject loss difference plots as meaningless.

The LC Index is the alternative Adams and Hand advocate instead of a loss difference plot. This has the same *x*-axis as a loss difference plot (and as a cost curve). The *y*-axis simply records which classifier has the best performance for a given x-value. It is not a numerical axis, the LC index gives no indication at all of the magnitude of difference in performance.

We agree with Adams and Hand on two central points: (1) that we want a visualization of performance that does not require exact knowledge of $C(-|+)$ and $C(+|-)$, and (2) that it is impossible to compute absolute performance measurements without knowing $C(-|+)$ and $C(+|-)$ exactly. But we do not agree that one must abandon all quantitative performance comparison. In cost curves, the *y*-axis is normalized—cost is measured relative to the minimum and maximum possible costs. This is a meaningful measure when the cost ratios are known even if absolute costs are not known. For example, the normalized cost when $c_0 = 2$ and $c_1 = 1$ is precisely the same as when $c_0 = 20$ and $c_1 = 10$ because scaling the costs also scales the minimum and maximum possible costs and the scale factor cancels. In using cost curves this normalization of costs must always be kept in mind. From a cost curve no statement at all can be made about the absolute cost that will be incurred if a certain classifier is used at a specific operating point. To make such a statement requires knowledge of the magnitude of the costs, and that cannot be extracted from the cost curve itself.

A valuable contribution by Adams and Hand (1999) is a method for using approximate knowledge of $PC(+)$ in selecting among classifiers. They argue that domain experts can often put a rough probability distribution, $prob(x)$, over the set of possible operating points, the *x*-axis. Then the expected performance of a classifier can be defined with respect to $prob(x)$. This technique applies to cost curves as well as to LC indexes (Ting, 2002). The area under a cost curve is the expected cost of the classifier assuming that all possible probability-cost values are equally likely, i.e. that $prob(x)$ is the uniform distribution.[4] More generally, expected cost can be computed using the following equation:

$$TEC = \int_o^1 Norm(E[Cost(x)]) * prob(x)dx$$

for any specific probability distribution $prob(x)$. This also allows a comparison of two classifiers when one does not strictly dominate the other and some knowledge of $prob(x)$ is

[4] This is in contrast with the area under the ROC curve which does not measure cost but is rather a ranking measure.

available. The difference in area under the two curves gives the expected advantage of using one classifier over another.

4. Classifier performance analysis using ROC curves and cost curves

Machine learning researchers have certain requirements for the visualization tool they use. Ideally, everything they currently do with scalar performance measures, such as error rate or AUC, would be possible by simple visual inspection of 2-d plots. In this section, we show that the seven specific questions posed in the introduction can all be answered readily by visual inspection of cost curves and not as readily by inspecting ROC curves. Each question is discussed in a separate subsection.

4.1. Visualizing classifier performance

This subsection addresses the question: what is classifier C's performance (expected cost) given specific misclassification costs and class probabilities?

Figure 6 shows a set of ROC points for C4.5 on the sonar data set from the UCI collection. Each point corresponds to a classifier trained using a different class distribution produced by undersampling the training set. Even though ROC analysis does not commit to any particular measure of performance it is still possible to read certain performance-related information from this figure. For example, certain ROC points are obviously dominated by others, and from the visually obvious fact that all the ROC points are well above the chance line (the diagonal joining (0,0) to (1,1)) one can easily see that this decision tree's overall performance is good.

Figure 7 illustrates two simple geometric constructions for reading quantitative performance information from an ROC plot for specific operating conditions. Both begin by drawing an iso-performance line through the ROC point with a slope representing the operating conditions (recall Eq. (1)). The construction shown on the left side of Fig. 7 then draws a line segment from the point (0, 1) that meets the iso-performance line at a right angle. The length

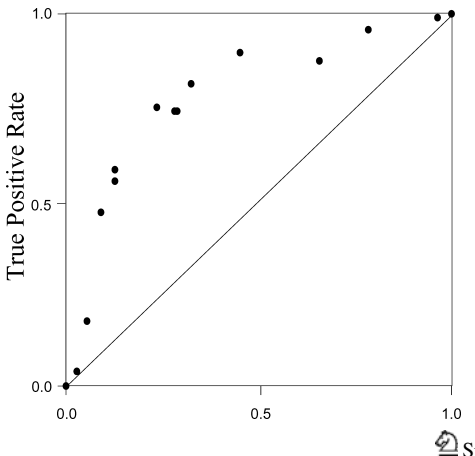


Fig. 6 ROC points for C4.5 on the sonar dataset

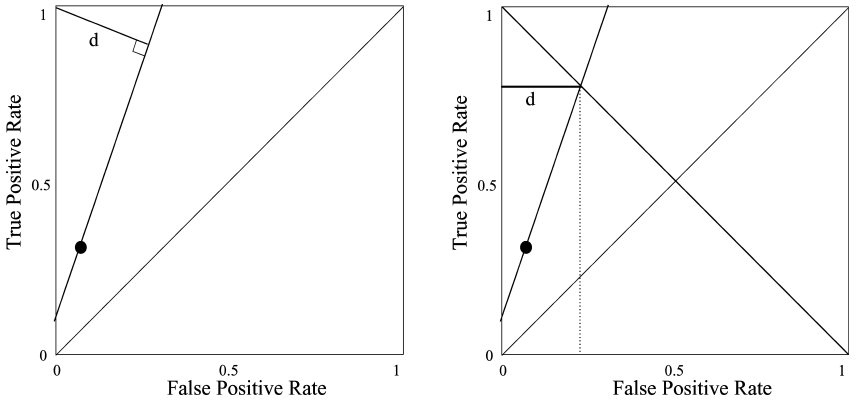
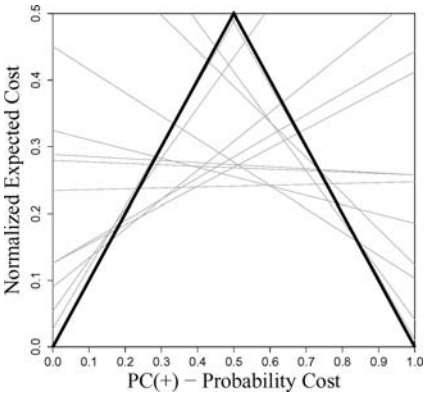


Fig. 7 Two methods for obtaining quantitative performance information from an ROC point

Fig. 8 Cost curves corresponding to Fig. 6



of this line segment is proportional to normalized expected cost.[5] The construction shown on the right, due to Peter Flach (2003, 2004), draws a horizontal line segment from the *y*-axis to the point where the iso-performance line intersects the diagonal line $TP = 1 - FP$. The length of this line segment is exactly equal to normalized expected cost. These constructions have the disadvantage that neither the iso-performance line nor the line segments added by the constructions are an intrinsic part of an ROC curve. Moreover, they cannot be added easily by the naked eye upon a casual inspection of an ROC curve. They also change depending on the operating conditions. So the ROC curve cannot be augmented with this information until a specific operating condition is defined.

Figure 8 shows the cost lines for the classifiers whose ROC points are shown in Fig. 6. Each cost line in Fig. 8 corresponds to one of the individual ROC points in Fig. 6. All the conclusions drawn from the ROC plot, and more, can be made from a quick visual inspection of this cost curve plot. The lower envelope is visually obvious. From this one can quickly see that, with an appropriately chosen level of undersampling, C4.5's decision tree will never have a normalized expected cost higher than 25%. One can also see that there are many choices of classification threshold that give near-optimal performance when $PC(+)$ is near 0.5.

[5] Thanks to an anonymous referee for pointing out this construction.

4.2. Comparing a classifier to the trivial classifiers

This subsection addresses the question: for what misclassification costs and class probabilities does classifier C outperform the trivial classifiers that assign all examples to the same class?

In an ROC plot, points (0,0) and (1,1) represent the trivial classifiers: (0,0) represents classifying all examples as negative, and (1,1) represents classifying all points as positive. The diagonal line connecting these points represents chance behavior. The cost curves for these classifiers are the diagonal lines from (0, 0) to (1, 1) (the always-negative classifier) and from (0, 1) to (1, 0) (the always-positive classifier). The operating range of a classifier is the set of operating points or $PC(+)$ values, for which the classifier outperforms both the trivial classifiers.

In an ROC curve the operating range of a classifier is defined by the slopes of the lines connecting the classifier's ROC point to (0,0) and (1,1). Slopes are notoriously difficult to judge visually. In general, the operating range cannot be read off an ROC curve precisely, but it can be roughly estimated by the minimum FP value and the maximum TP value of the non-trivial classifiers on the convex hull. For example, the dotted lines in Fig. 9(a) show the approximate operating range for this set of classifiers to be $0.05 < PC(+) < 0.94$.

In a cost curve, the operating range is defined by the interval of $PC(+)$ values between the points where the cost curve intersects the diagonal line for the always-positive classifier and the diagonal line for the always-negative classifier. The vertical lines in Fig. 9(b) show the operating range for this classifier exactly, as opposed to the approximate method for ROC curves just described. We see that the actual operating range is narrower, $0.14 < PC(+) < 0.85$.

In Fig. 8 the diagonals representing the trivial classifiers are shown as bold lines. We can see exactly where the lower envelope, or any individual classifier, intersects these diagonals, and directly read off the precise operating range. For the lower envelope this is $0.17 < PC(+) < 0.84$. Moreover, the vertical distance below the diagonal at a given $PC(+)$ value within the operating range is the quantitative performance advantage of the classifier over the trivial classifier at that operating point.

Cost curves make it so easy to see a classifier's operating range that performances worse than the trivial classifiers cannot be overlooked. ROC curves and other commonly used meth-

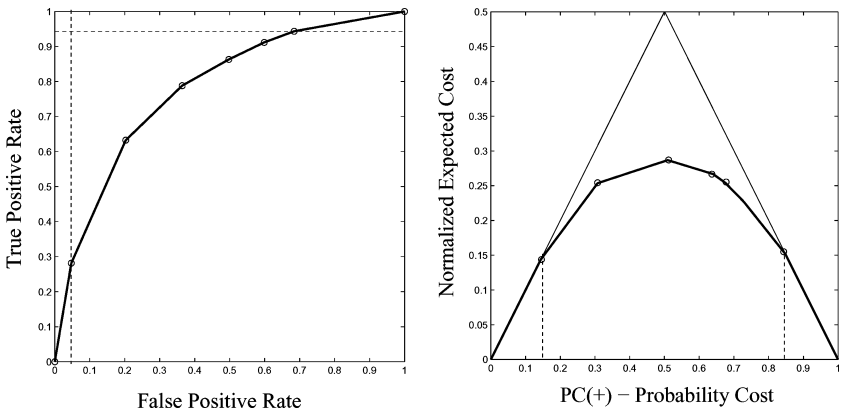


Fig. 9 (a) Estimate of the operating range of an ROC curve – (b) Exact operating range of the corresponding cost curve

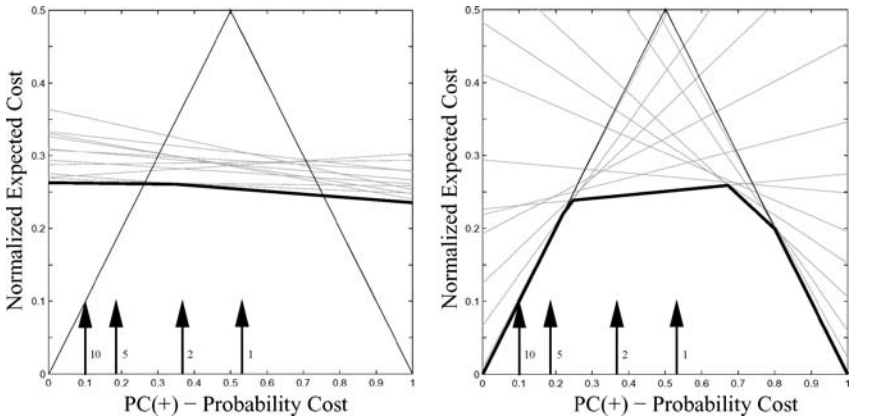


Fig. 10 (a) Cost curves for various oversampling ratios — (b) Cost curves for various undersampling ratios

ods of presenting performance results do not have this property, which might possibly result in performances worse than the trivial classifiers being published without being noticed. For example, Domingos (1999) compares MetaCost, oversampling and undersampling on a variety of datasets for four cost ratios—1:1, 2:1, 5:1 and 10:1. In Fig. 10 we have approximately reproduced its oversampling and undersampling experiments on the sonar dataset. Our purpose is not to single out these experiments or to question their results but rather to show how easy it is to overlook classifiers' operating ranges with the performance evaluation methods currently in use.

Each gray line in Fig. 10(a) is a cost curve of the decision tree created by C4.5 for a different oversampling ratio[6]. The arrows in Fig. 10 indicate the $PC(+)$ values for the four cost ratios. They are not exactly at the ratios themselves because $p(+)$ in this domain is 0.5337, not 0.5. As can be seen, the operating range for the lower envelope of the classifiers produced by oversampling is rather narrow, $0.28 < PC(+) < 0.75$ and does not include the two most extreme cost ratios (5:1 and 10:1) used to evaluate the systems. Fig. 10(b) shows the same information but now the training set is produced by undersampling instead of oversampling[7]. The operating range is slightly wider than for oversampling, $0.25 < PC(+) < 0.82$, but still does not include cost ratios 5:1 and 10:1. Although it makes sense to oversample or undersample with these ratios, or even more extreme ones, for training purposes, it does not make sense to test these classifiers with cost ratios this extreme, because none of these classifiers should be used for these cost ratios.

4.3. Choosing between classifiers

This subsection addresses the question: for what misclassification costs and class probabilities does classifier C1 outperform classifier C2 ?

[6] To generate the training sets, instances are duplicated for one of the classes up to the floor of the desired ratio. The remaining fraction is chosen randomly from the training data for that class.

[7] The difference in performance of the two sampling methods is vividly seen in the cost curves. For further discussion see Drummond and Holte (2003).

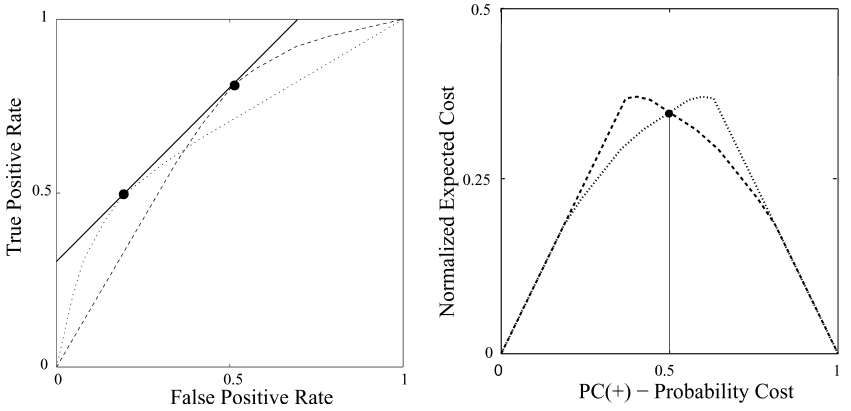


Fig. 11 (a) Two ROC curves that cross — (b) Corresponding cost curves

If the ROC curves for two classifiers cross, each classifier is better than the other for a certain range of operating points. Identifying this range visually is not easy in an ROC diagram and, perhaps surprisingly, the intersection point of the ROC curves has little to do with the range[8]. Consider the ROC curves for two classifiers, the dotted and dashed curves of Fig. 11(a). The solid line is the iso-performance line tangent to the two ROC curves. Its slope represents the operating point at which the two classifiers have equal performance. For operating points corresponding to steeper slopes, the classifier with the dotted ROC curve performs better than the classifier with the dashed ROC curve. The opposite is true for operating points corresponding to shallower slopes. This information might be extracted from the graph and tabulated (Provost et al., 1998, Table 1)

Fig. 11(b) shows the cost curves corresponding to the ROC curves in Fig. 11(a). The vertical line drawn down from the intersection point of the two cost curves indicates the operating point where the two curves have equal performance. This $PC(+)$ value exactly corresponds to the slope of the iso-performance line in Fig. 11(a). It can immediately be seen that the dotted line has a lower expected cost and therefore outperforms the dashed line when $PC(+) < 0.5$ (approximately) and vice versa. Overall, cost curves show for what range of values one classifier outperforms another much more clearly than ROC curves. For the latter, one has to either add iso-performance lines or look up the values in a table.

4.4. Comparing classifier performance

This subsection addresses the question: what is the difference in performance between classifier C1 and classifier C2?

Figures 12(a) and (b) illustrate how much more difficult it is to compare classifiers with ROC curves than with cost curves. Although it is obvious in the ROC diagram that the dashed

[8] This point maps to a straight line in cost space that is not of any special significance nor visually apparent. This ROC point's only distinction in ROC space is to be both a weighted average of two points, A1 and A2 (on opposite sides of the intersection point), on one ROC curve and also to be a weighted average of two points, B1 and B2, on the other ROC curve. In cost space, the line corresponding to this point has the same distinction: it can be expressed as a weighted average of the cost lines corresponding to A1 and A2, and it can also be expressed as a weighted average of the cost lines corresponding to B1 and B2.

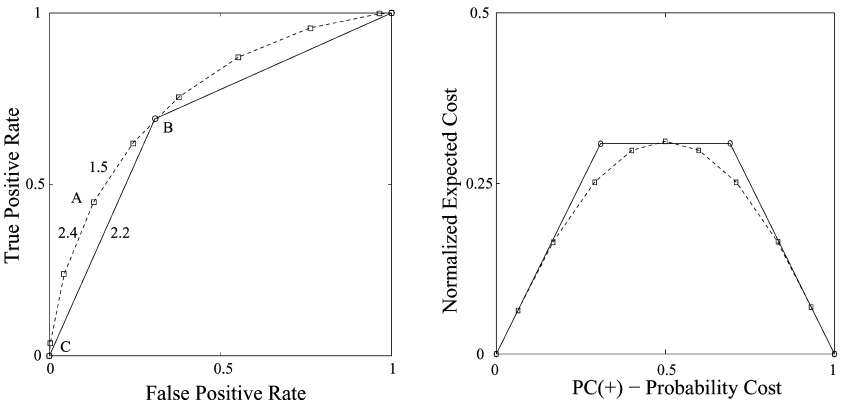


Fig. 12 (a) Two ROC curves whose performance is to be compared — (b) Corresponding cost curves

curve is better than the solid one, it is not easy, visually, to determine by how much. One might be tempted to take the Euclidean distance normal to the lower curve to measure the difference. But this would be wrong on two counts. First, the difference in expected cost is the weighted Manhattan distance between two classifiers not the Euclidean distance. Equation 7 gives the difference, each cost calculated using Eq. (4) with TP replacing $(1 - FN)$. One can interpret this graphically as the distance between the two points traveling first in the X direction and second in the Y direction. The distances must be "weighted" to account for the relevant prior probabilities and misclassification costs.

$$E[Cost_1] - E[Cost_2] = (TP_2 - TP_1)\underbrace{*p(+) * C(-|+)}_{w_+}$$
$$+(FP_1 - FP_2)\underbrace{*p(-) * C(+|-)}_{w_-} \quad (7)$$

Second, the performance difference should be measured between the appropriate classifiers on each ROC curve—the classifiers out of the set of possibilities on each curve that would be used at each given operating point. This is a crucial point that will be discussed in Section 5. To illustrate how intricate this is, suppose the two classes are equiprobable but that the ratio of the misclassification costs might vary. In Fig. 12(a) for a cost ratio of 2.1 the classifier marked A on the dashed curve should be compared to the one marked B on the solid curve. But if the ratio was 2.3, A should be compared to the trivial classifier marked C on the solid curve at the origin.

The dashed and solid cost curves in Fig. 12(b) correspond to the dashed and solid ROC curves in Fig. 12(a). The horizontal line atop the solid cost curve corresponds to classifier B. Classifier C is the trivial "always negative" classifier, the diagonal cost curve rising from (0,0) towards (1,1). The vertical distance between the cost curves for the dashed and solid classifiers directly indicates the performance difference between them. The dashed classifier outperforms the solid one—has a lower or equal expected cost—for all values of $PC(+)$. The maximum difference is about 20% (0.25 compared to 0.3), which occurs when $PC(+)$ is about 0.3 or 0.7. Their performance difference is negligible when $PC(+)$ is near 0.5, less than 0.2 or greater than 0.8.

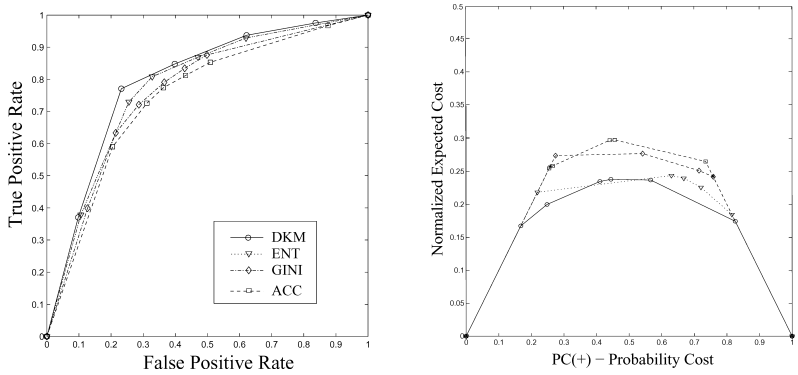


Fig. 13 Various C4.5 splitting criteria on the sonar dataset (a) ROC curves — (b) Corresponding cost curves

The difficulty of comparison is even worse with data from real experiments. The ROC curves in Fig. 13(a) show the performance of the decision trees built on the Sonar dataset by C4.5 with different splitting criteria (Drummond & Holte, 2000b). The ROC curves are close together and somewhat tangled, making visual analysis difficult. These are typical of the comparative experiments in machine learning. While it is clear that the DKM splitting criterion dominates the others, there is no indication of how much better DKM is than them or how much their performances differ from one another.

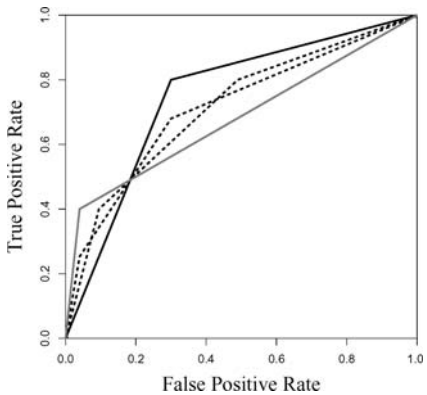
Figure 13(b) shows the corresponding cost curves. The tangled ROC curves are now cleanly separated. Although DKM dominates, it can now be seen that its performance differs little from ENT's over a fairly broad range, $0.3 < PC(+) < 0.6$. These two splitting criteria have similar operating ranges and are clearly superior to the other two. It can also be clearly seen that GINI dominates ACC over most of their operating range.

4.5. Averaging cost curves

This subsection addresses the question: what is the average of performance results from several independent evaluations of classifier C (e.g. the results of 10-fold cross-validation)?

Each solid line in Fig. 14 is an ROC curve based on a single non-trivial classifier. One is based on the point $(FP_1, TP_1) = (0.04, 0.4)$, the other is based on the point $(FP_2, TP_2) = (0.3, 0.8)$. We assume that they are the result of learning or testing from different random

Fig. 14 Vertical and horizontal averages of two ROC curves



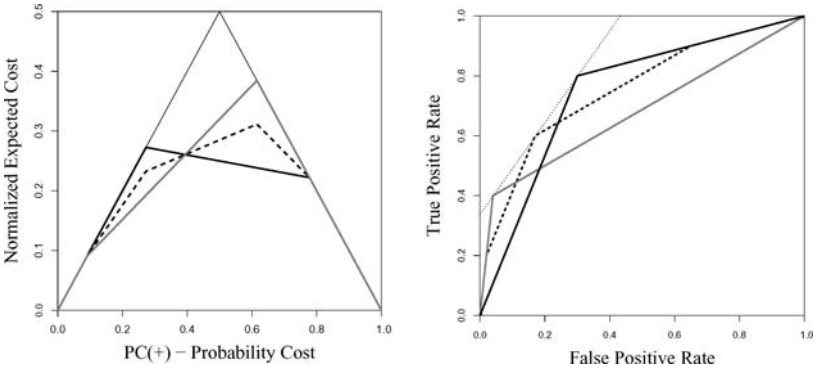


Fig. 15 (a) Vertical average of the cost curves corresponding to the ROC curves in Fig. 14 — (b) Corresponding ROC curve

samples, or some other cause of random fluctuation in performance, and therefore their average can be used as an estimate of expected performance. The question is, how exactly shall the "average" of the two curves be calculated?

There is no universally agreed-upon method of averaging ROC curves. Swets and Pickett (1982) suggest two methods, pooling and "averaging", and Provost et al. (1998) propose an alternative averaging method. The Provost et al. method is to regard y, here the true positive rate, as a function of x, here the false positive rate, and to compute the average y value for each x value. We call this method "vertical averaging". In Fig. 14 the vertical average is one of the dotted lines in between the two ROC curves. The other dotted line is the "horizontal" average—the average false positive rate (x) for each different true positive rate (y). As the figure illustrates, these two averages are intrinsically different.

An important shortcoming of these methods of averaging is that the performance (error rate, or cost) of the average curve is not the average performance of the two given curves. The easiest way to see this is to consider the iso-performance line that connects the central vertices of the two ROC curves in Fig. 14. The vertical and horizontal averages do not touch this line; they are well below it.

Now consider what vertical averaging would do in cost space, where each x value is an operating point and y is performance (normalized expected cost). The vertical average of two cost curves is the average performance at each operating point—precisely what we wish to estimate. The solid lines in Fig. 15(a) are the ROC curves from Fig. 14 translated into cost curve lower envelopes. The expected performance based on these two cost curves is given by the bold dotted line. Translating this average cost curve into ROC space produces the dashed line in Fig. 15(b). Note that its central vertex is on the iso-performance line joining the vertices of the two given ROC curves (in fact, midway between them on this line, by definition).

The reason these different averaging methods do not produce the same result is that they differ in how points on one curve are put into correspondence with points on the other curve. With vertical averaging of ROC curves, points correspond if they have the same FP value. With vertical averaging of cost curves, points correspond if they have the same operating point. Other methods of averaging ROC curves, such as horizontal averaging, put points on the two curves into correspondence in yet other ways. Only the cost curve method of averaging has the property that the expected cost of the average of the given curves is the average of the curves' expected costs.

4.6. Confidence intervals on cost lines

This subsection addresses the question: what is the 90% confidence interval for classifier C's performance?

Although there has been considerable investigation of confidence intervals for ROC curves, none of it directly addresses this question. The vast majority of the ROC literature on confidence intervals investigates confidence intervals on the ROC curve itself, constructed in either a point-wise manner (Dukic & Gatsonis, 2003; McNeil & Hanley, 1984; Platt et al., 2000; Tilbury et al., 2000; Zou et al., 1997) or as a global confidence band (Dukic & Gatsonis, 2003; Jensen et al., 2000; Ma & Hall, 1993). Macskassy et al. (2005) give a good review of this work accessible to a machine learning audience. These confidence intervals provide bounds within which a classifier's TP and FP are expected to co-vary, they do not directly provide bounds on the classifier's performance[9]. The confidence intervals in the ROC literature most closely related to performance are the confidence intervals placed on scalar, aggregate performance measures such as AUC (Agarwal et al., 2005; Bradley, 1997; Cortes & Mohri, 2005). As discussed in the introduction, these give little indication of expected performance under specific operating conditions.

Because cost curves plot performance as a function of operating conditions, confidence intervals in cost space will naturally indicate the bounds within which performance is expected to vary for every operating condition. In this subsection, and the next, we will restrict our attention to cost lines. Section 5 will address the general case of cost curves, such as lower envelopes, created by the piecewise construction of a single, hybrid classifier out of several classifiers.

The measure of classifier performance is derived from a confusion matrix produced from a sample of the data. As there is likely to be variation between samples, the measure itself is a random variable and some estimate of its variance is useful. This is typically done through a confidence interval calculated assuming that the distribution of the measure belongs to, or is closely approximated by, some parametric family such as Gaussian or Student's t. An alternative is to use computationally intense, non-parametric methods. Margineantu and Dietterich (2000) described how one such non-parametric approach, the bootstrap (Efron & Tibshirani, 1993), can be used to generate confidence intervals for predefined cost values. We use a similar technique, but for the complete range of class frequencies and misclassification costs.

The bootstrap method is based on the idea that subsamples generated from the available data are related to those data in the same way that the available data relate to the original population. Thus the variance of an estimate based on subsamples should be a good approximation to its true variance. A confidence interval is produced from new confusion matrices generated by resampling from the original matrix. The exact way bootstrapping is carried out depends on the source of the variation. As we have argued throughout this paper, we do not expect that the training set frequency represents the deployment frequency. So to generate confidence intervals, we assume the deployment frequency is fixed, but unknown, and will be determined at the time the classifier is used. We therefore do not need to account for any variance in the class frequency and can draw samples as if they come from two binomial distributions. This is often termed conditioning on the row marginals of the confusion matrix. This manner of generating new matrices is analogous to stratified cross validation as the class frequency is guaranteed to be identical in every sample.

[9] This was first observed by Provost et al. (1998).

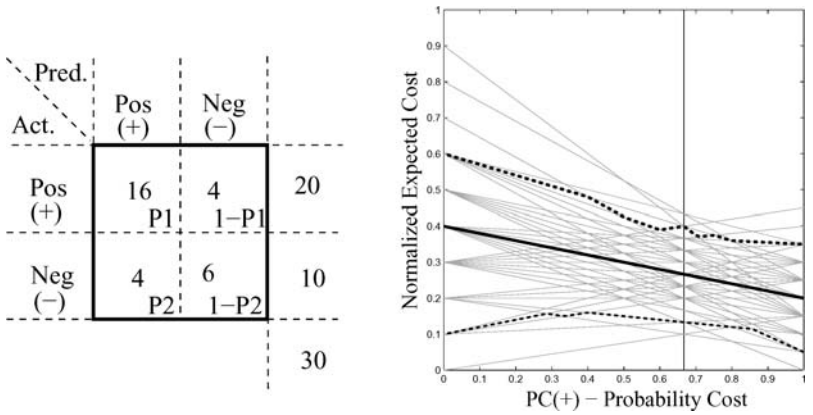


Fig. 16 (a) Binomial sampling resamples each row of the confusion matrix independently — (b) Resulting cost curve confidence

To illustrate the sampling, suppose we have the confusion matrix of Fig. 16(a). There are 30 instances, 20 of which are positive and 10 negative. The classifier correctly labels 16 out of 20 of the positive class, but only 6 out of 10 of the negative class. We can normalize the rows by the row totals, 20 and 10, the number of positive and the number of negative instances respectively, resulting in two independent binomials with probabilities $P1$ and $P2$. New matrices are produced by randomly sampling according to these binomial distributions until the number of positive and negative instances equal the corresponding row totals, producing a new confusion matrix with the same row totals as the original.

For each new confusion matrix, a gray line is plotted representing the new estimate of classifier performance, as shown in Fig. 16(b). For ease of exposition, we generated 100 new confusion matrices (typically at least 500 are used for an accurate estimate of variance). To find the 90% confidence limits, if we had values just for one specific x-value, the fifth lowest and fifth highest value could be found. A simple, but inefficient way to do this is to find the fifth lowest and highest line segment for many different $PC(+)$ values. A more sophisticated algorithm, which runs in $O(n^2)$ time (n is the number of lines) is given in Miller et al. (2001). It is noticeable in Fig. 16(b) that there are only a limited number of FP and FN values achieved. As there are ten negatives, the values on the y-axis at the left hand side must be multiples of 1/10. As there are twenty positives, those on the right hand side are multiples of 1/20.

The confidence interval, the bold dashed lines in Fig. 16(b), tends to be broad at each end and narrower near the class distribution of the test set, the vertical solid line. The explanation for this is as follows. The standard deviation of the error rate, or normalized expected cost, is given by the following equation:

$$\sigma_{NEC} = \left(p(+)^2 * \sigma_{FN}^2 + (1 - p(+))^2 * \sigma_{FP}^2\right)^{\frac{1}{2}} \qquad (8)$$

As Eq. (8) is quadratic in $p(+)$ (our x-axis, if costs are equal) it has a single minimum. We can find that minimum by differentiating over $p(+)$, and setting the value to zero, as in the

following equation:

$$0 = p(+) * \sigma^2_{FN} + (p(+) - 1) * \sigma^2_{FP}$$

The variance of *FN* and *FP* are dependent on the variance of the binomial distributions and are given by the following equation:

$$\sigma^2_{FN} = \frac{P1 * (1 - P1)}{m}, \sigma^2_{FP} = \frac{P2 * (1 - P2)}{n}$$

In this equation m is the number of positive examples in the test set and n is the number of negative examples.

If the two binomial proportions are equal, $P1 = P2 = P$, then the minimum occurs at the $p(+)$ value given in the following equation:

$$\begin{aligned} 0 &= p(+)\frac{P * (1 - P)}{m} + (p(+) - 1)\frac{P * (1 - P)}{n} \\ 0 &= \frac{p(+)}{m} + \frac{p(+) - 1}{n} \\ p(+) &= \frac{m}{m + n} \end{aligned}$$

This is just the number of positives over the total number of instances, in other words the class distribution of the test set. If the proportions are different, $P1 \neq P2$, the minimum occurs between the class distribution and whichever one of FN and FP has the smaller variance.

4.7. Testing if performance differences are significant

This subsection addresses the question: for what misclassification costs and class probabilities is the difference in performance between classifier C1 and classifier C2 statistically significant?

Although it is useful to have a confidence interval around the performance estimate of a single classifier, often we want to compare two classifiers and determine if the difference in their performance is statistically significant. For example, consider the two classifiers whose cost curves are shown as thick dashed lines in Fig. 17(a). The lower line dominates the upper one, but is the difference in performance significant? The shaded area around the upper line indicates the 90% confidence interval for this classifier's performance. The confidence interval includes the lower line for $PC(+)$ values less than 0.5, suggesting that, at least for these $PC(+)$ values, the difference is not significant. In Fig. 17(b) the dark shaded region is the intersection of the 90% confidence intervals for the two classifiers. Clearly they overlap for every possible $PC(+)$ value.

We might conclude that there is not a statistically significant difference between the classifiers, but this would not be a sound conclusion. The confidence interval of the difference between these two classifiers' performances depends not only on the standard deviations of the two classifiers but also on their correlation. Equation (9) gives the standard deviation σ_Z of the difference, Z, of two random variables, X and Y, where ρ is the correlation.

$$Z = X - Y \qquad (9)$$

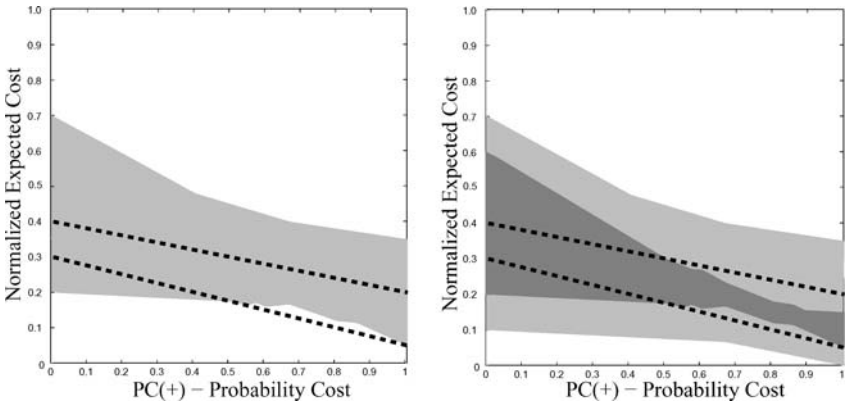


Fig. 17 (a) Confidence interval for the upper cost curve (rescaled Y)— (b) Union (light shade) and intersection (dark shade) of the confidence intervals for both cost curves (Rescaled Y)

$$\sigma_Z = \left(\sigma_X^2 + \sigma_Y^2 - 2\sigma_X * \sigma_Y * \rho\right)^{\frac{1}{2}}$$

$$\sigma_Z = \begin{cases} |\sigma_X - \sigma_Y|, & \text{when } \rho = 1 \\ (\sigma_X^2 + \sigma_Y^2)^{\frac{1}{2}}, & \text{when } \rho = 0 \\ \sigma_X + \sigma_Y, & \text{when } \rho = -1 \end{cases} \tag{10}$$

As Eq. (10) shows, if X and Y are perfectly positively correlated ($\rho = 1$) then the standard deviation of Z, their difference, depends on the absolute difference of their standard deviations. If they are uncorrelated ($\rho = 0$) Z's standard deviation is the Euclidean sum of X and Y's standard deviations. If they are perfectly negatively correlated ($\rho = -1$), Z's standard deviation is the sum of X and Y's standard deviations. Determining statistical significance, or lack thereof, by the overlap of two confidence intervals is equivalent to making the worst case assumption that they are perfectly negatively correlated.

Fortunately, a slight extension of the approach of Margineantu and Dietterich (2000) enables us to take the correlation of the classifiers correctly into account. The process is similar to that for a single classifier, but now a 3-dimensional matrix is used. This matrix is shown in the middle of Fig. 18. The matrix is split into two layers: the upper layer is for examples from the positive class, the lower layer is for examples from the negative class. In each layer, the columns represent how the first classifier labels instances and the rows represent how the second classifier labels the same instances. If the matrix is projected down to two dimensions, by summing over the other classifier, a confusion matrix for each classifier is produced.

We fix the number of examples in each class and then sample each layer as if it were a 4-valued multinomial until we have the requisite number of instances for each class. This is similar to the sampling used for a single classifier, except we now sample two multinomials rather than two binomials. Sampling in this way faithfully captures the correlation of the two classifiers, i.e. how they jointly classify particular instances, without having to explicitly measure the correlation coefficient ρ. After a new matrix is created by this resampling method, it is projected down to create a confusion matrix for each classifier, resulting in a cost line representing the performance of each classifier. These two lines are subtracted to create a single cost line representing the performance difference between the two classifiers. This

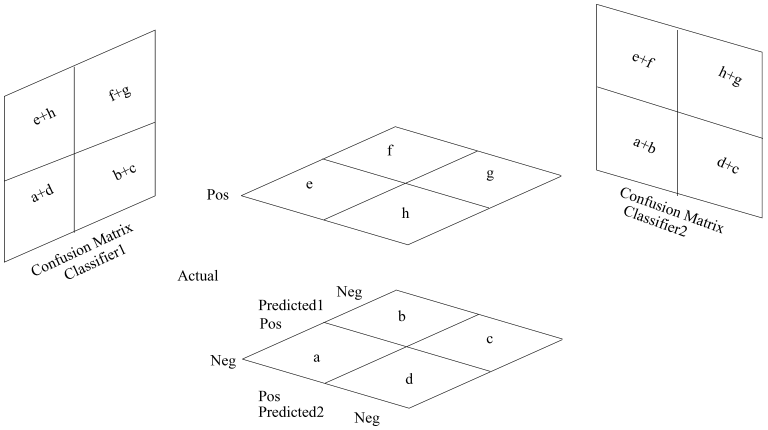


Fig. 18 3-dimensional confusion matrix for significance testing

process is then repeated many times to create a large set of performance-difference lines, from which a confidence interval for the difference is calculated.

The thick continuous line at the bottom of Fig. 19(a) represents the difference between the means of the two classifiers, which is the same as the mean difference. The shaded area represents the confidence interval of the difference, computed in the manner just described. As the difference can be negative, the *y*-axis has been extended downwards. Here we see that the confidence interval does not contain zero. So the difference between the classifiers is statistically significant for all possible operating points. Fig. 19(b) shows the same two classifiers but with the classifications less correlated. Notably, the confidence interval is much wider and includes zero, so the difference is not statistically significant. Thus cost curves give a nice visual representation of the significance of the difference in expected cost between two classifiers across the full range of misclassification costs and class frequencies. The cost curve representation also makes it clear that performance differences might be significant for some range of operating points but not others. An example of this is shown in Fig. 20, where the difference is significant only if $PC(+) > 0.7$.

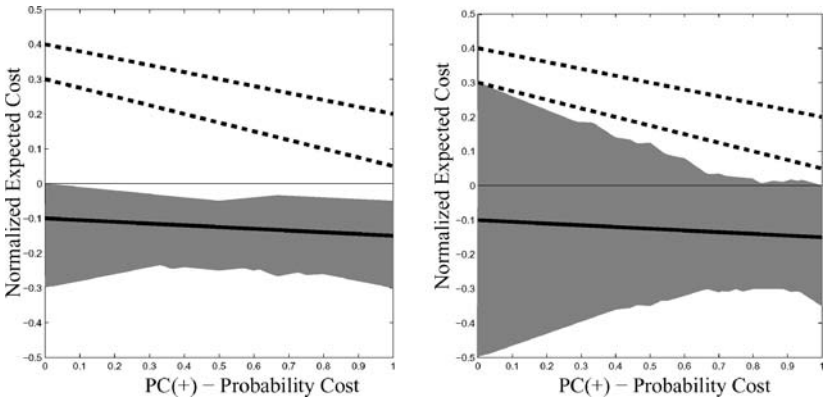


Fig. 19 Confidence interval around the difference in two classifiers' performance (a) High correlation — (b) Low correlation

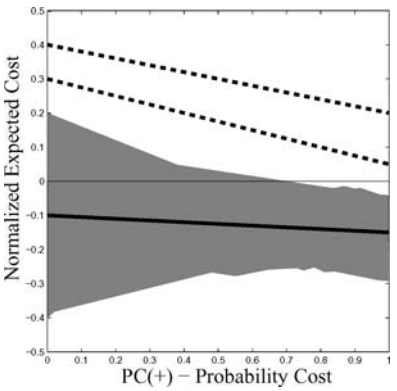


Fig. 20 Confidence interval for the difference, medium correlation

As with confidence intervals, significance testing for ROC curves has focused on properties other than performance. Metz and Kronman (1980) show how to test if the parameters defining two ROC curves in the binormal model are significantly different. Metz et al. (1983) describe a more general approach for significance testing of properties of ROC curves and applies it in three ways: (1) to test if the binormal parameters that define two ROC curves are different; (2) to test if two ROC curves have different *TP* values for a given *FP*; and (3) to test if the AUC of two ROC curves are different. By contrast, the significance test, just described for cost curves, determines if the performance of two classifiers is significantly different for each possible operating point.

5. Selection criteria for choosing a classifier given specific operating conditions

As noted in Section 3.1, it is often easy, by varying a parameter such as a classification threshold or cost matrix, or by varying the class distribution in the training set, to create a whole set of classifiers. These are visualized as a set of points in ROC space and as a set of lines in cost space. Provost and Fawcett (2001) observed that a hybrid classifier could be created out of a set of such classifiers by selecting among them based on the current operating conditions. For example, classifier A might be selected if $PC(+) < 0.5$, but classifier B selected if $PC(+) \geq 0.5$.

The criterion used to determine which classifiers are selected, given the current operating conditions, we refer to as the selection criterion. The first issue discussed in this section is how well the different selection criteria currently in use can be visualized in ROC space and cost space. The second issue addressed is how the methods defined above for computing confidence intervals and determining statistical significance for cost lines can be applied to hybrid classifiers.

5.1. Alternative selection criteria

This section discusses some commonly used selection procedures.

5.1.1. Performance-independent selection criteria

One commonly used selection criterion is to select the classifier whose parameter settings and training conditions most closely agree with the current operating conditions (Ting, 2004).

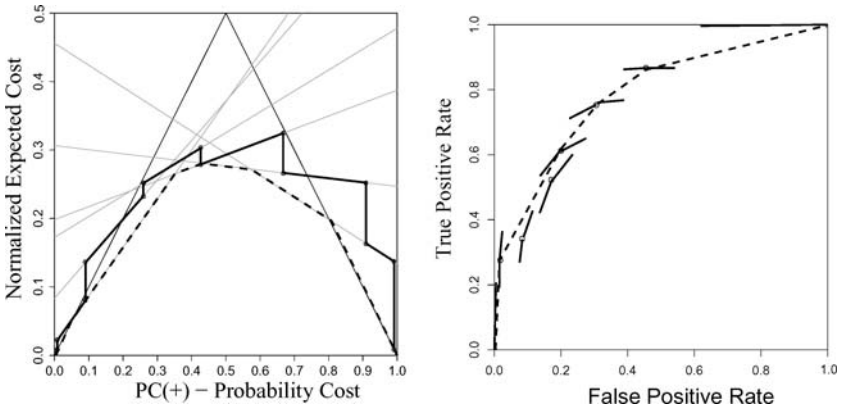


Fig. 21 (a) Hybrid cost curves formed by two different selection procedures (solid: performance-independent selection; dashed: cost-minimizing selection). (b) The corresponding ROC visualization of the two selection procedures

This is most clearly seen in studies of cost-sensitive learning, where the cost matrix used to train the classifier is the same as the cost matrix used to test it (Domingos, 1999; Kukar & Kononenko, 1998; Margineantu, 2002; Pazzani et al., 1994; Ting, 2000; Turney, 1995; Webb, 1996). Likewise, Zadrozny et al. (2003) and Radivojac et al. (2003) adjust the training set distributions in precise accordance with the costs used in testing the resulting classifiers. We call this kind of selection criterion "performance independent" because a classifier is selected without considering how well it will perform in the current operating conditions.

In cost space this criterion is easy to visualize, as seen in Fig. 21(a). The gray lines are the classifiers available for selection. Each line has a bold portion indicating the operating conditions in which it will be used as determined by the selection criterion defined above. Collectively these bold lines are a hybrid cost curve, an unbiased estimate of the hybrid classifier's performance.

This selection criterion can also be visualized in ROC space by drawing two line segments through each ROC point, with the slopes of the line segments representing the range of operating conditions for which that point would be selected. This is illustrated with the solid line segments in Fig. 21(b). It gives rise to the ROC convex hull (the dashed curve in Fig. 21(b)) only if this selection criterion happens to make cost-minimizing selections, which in general it will not do.

5.1.2. Cost-minimizing selection criterion

Although the preceding criterion, which matches training conditions to testing conditions, is widely thought to produce superior performance, it is not guaranteed to produce optimal performance. This can be seen by the fact that the solid line produced by this criterion in Fig. 21(a) is well above the lower envelope (the dashed line) for almost all possible operating points. The suboptimality of the performance-independent selection criterion arises in actual practice. The only large-scale study of this fact is by Weiss and Provost (2003). It is illustrated in Fig. 22, which shows the performance of Naive Bayes on the UCI Sonar dataset. The different classifiers are produced by varying Naive Bayes's threshold. The performance-independent criterion, in this case, is to set the threshold to correspond to the operating conditions. For example, if $PC(+) = 0.2$. the Naive Bayes threshold is set to 0.2. The solid

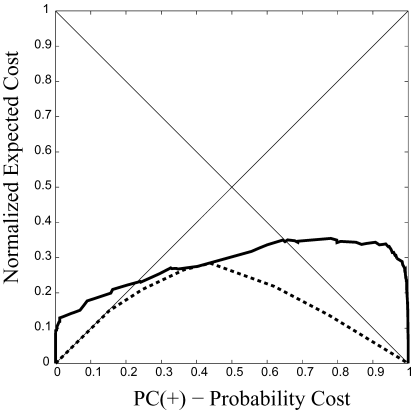


Fig. 22 Naive bayes on the sonar dataset: Hybrid cost curves formed by two different selection procedures (Rescaled Y) (solid: performance-independent selection; dashed: cost-minimizing selection)

line in Fig. 22 is the performance of the hybrid created by this selection procedure. It is clearly suboptimal for all operating conditions except in a narrow range around $PC(+) = 0.45$, and performs considerably worse than the trivial classifiers for $PC(+) < 0.2$ and $PC(+) > 0.7$.

An alternative criterion, implicitly suggested by the ROC convex hull (Provost & Fawcett, 2001), is to choose the classifier that performs best for each operating point regardless of its training conditions or parameter settings. There are few examples of the practical application of this technique. One example is (Fawcett and Provost, 1997), in which the decision threshold parameter was tuned to be optimal, empirically, for the test distribution. This criterion is visualized very well in both ROC space (the ROC convex hull) and cost space (the lower envelope).

5.1.3. Neyman-Pearson criterion

The Neyman-Pearson criterion comes from statistical hypothesis testing and minimizes the probability of a type two error for a maximum allowable probability of a type one error. For our purposes, this means fixing the maximum acceptable false positive rate and then finding the classifier with the largest true positive rate. ROC curves are ideally suited for this purpose because the *x*-axis of ROC space is precisely the quantity (false positive rate) constrained by the Neyman-Pearson criterion. The optimal classifier can found by drawing a vertical line for the particular value of *FP* until it intersects with the ROC convex hull, as shown by the dashed line in Fig. 23(a).

The procedure is equally easy in cost space. Remembering that the intersection of a classifier with the *y*-axis at $PC(+) = 0$ gives its false positive rate, a point can be placed on the *y*-axis representing the criterion. This is marked *FP* in Fig. 23(b). Immediately on either side of this point are the endpoints of two of the classifiers forming sides of the lower envelope. Connecting the new point to where those two classifiers intersect creates the classifier optimizing the Neyman-Pearson criterion. Extending it to where it crosses the line at $x = 1$ gives its *TP* value.

5.1.4. Workforce utilization

The workforce utilization criterion is based on the idea that a workforce can handle a fixed number of cases, factor *W* in Eq. (11). To keep the workforce maximally busy we want to select the best *W* cases. This is realized by the inequality condition of Eq. (11) and is the line

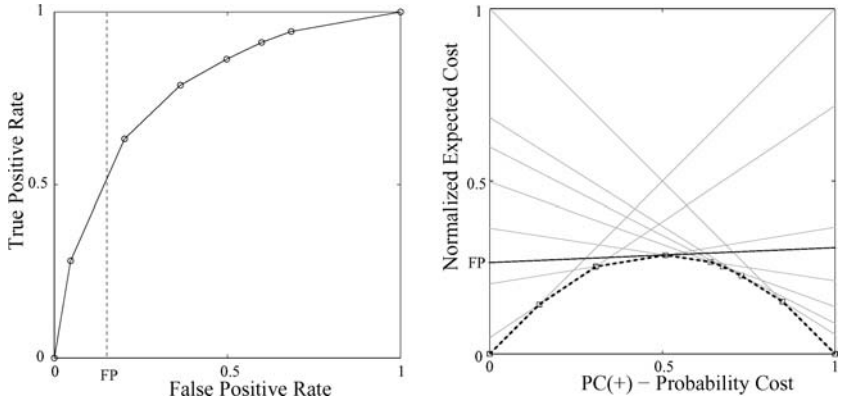


Fig. 23 Neyman-Pearson criterion (a) ROC curve – (b) Cost curve (Rescaled Y)

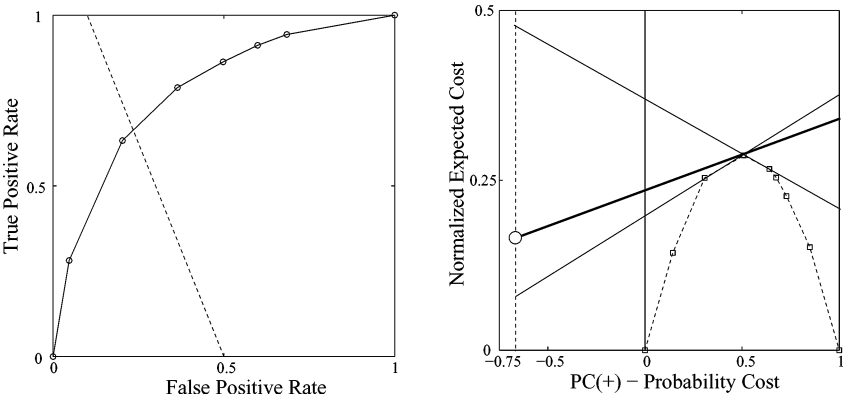


Fig. 24 Workforce utilization (a) ROC curve — (b) Cost curve

given by Eq. (12).

$$TP * P + FP * N \leq W \quad (11)$$

$$TP = -\frac{N}{P} * FP + \frac{W}{P} \quad (12)$$

Because Eq. (12) is a linear relation between the two axes defining ROC space, the workforce utilization criterion is as easily visualized in ROC space as the Neyman-Pearson criterion. It is a negatively sloped line in ROC space, such as the dashed line in Fig. 24(a), and the optimal classifier is the point where this line intersects the ROC convex hull.

Like all lines in ROC space, the workforce utilization criterion can be transformed into a point in cost space using Eq. (3)—the dashed line in Fig. 24(a) is transformed into the point shown as a circle on the left hand side of Fig. 24(b). Because the line's slope is negative, the transformation results in a negative $PC(+)$ value. A process similar to the one for the Neyman-Pearson criterion can now be applied—extend the cost curve lines for the classifiers until they have the same $PC(+)$ value as this point, indicated by the vertical dashed line.

Then pick the nearest classifiers above and below the point at that $PC(+)$ value, and draw a line from the point through the intersection point of those two classifiers. This, the bold line in Fig. 24(b), represents the classifier that optimizes workforce utilization. Unfortunately the point representing the workforce utilization criterion can be arbitrarily far outside the normal range for $PC(+)$, which can reduce the ease and effectiveness of visualizing this criterion. ROC curves are much better than cost curves for visualizing workforce utilization.

5.2. Analyzing hybrid classifiers

It is important to realize that the cost curve (or ROC curve) for a hybrid classifier built piecewise from the cost curves for the individual classifiers that make up the hybrid is not an unbiased estimate of performance except when a performance-independent selection criterion is used. The reason is that the other criteria use the FP and TP values of the underlying classifiers to create the hybrid, and therefore the data used to estimate these FP and TP values is part of the training set for the hybrid classifier, even if it was not used in training the base classifiers. To emphasize this difference we refer to these criteria as empirical selection criteria, in contrast to performance-independent criteria. To obtain unbiased estimates of the performance of a hybrid classifier constructed by empirical selection criteria, it is necessary to subject the hybrid to a new round of testing with fresh data (Bengio & Mariéthoz, 2004).

If one adopts an empirical selection criterion, it radically changes the way in which comparative machine learning experiments are conducted. The present practice is to compare learning algorithms on the same training set. This is correct if a performance-independent selection criterion is used to construct a hybrid, but is not correct if an empirical selection criterion is used. For the latter, one must attempt to find the optimal training sets and parameter settings for each operating point separately for each learning algorithm, and then compare the lower envelopes that result. This is necessary because, in general, the training set which produces the best performance for a given operating point will be different for the two learning algorithms and this method of comparison will therefore produce different (and more appropriate) results than present practice.

Having now reached an understanding that there are different selection procedures for constructing hybrids, and new methods required to estimate their performance, it is now straightforward to extend the methods of the preceding sections for computing confidence intervals and significance of difference from cost lines to hybrids. The cost curve for a hybrid classifier is piecewise linear and it is merely a matter of applying the previous techniques on a piecewise basis to construct a confidence interval for a hybrid classifier, or the significance of the difference between two hybrids.

6. Limitations of cost curves

In this section we summarize the main limitations of cost curves. We first discuss limitations that cost curves share with ROC curves and then discuss circumstances in which ROC curves are preferable to cost curves.

This paper has focused on classification problems in which there are only two classes. This is because of our emphasis on visual performance analysis. High-dimensional functions are notoriously difficult to visualize and the number of dimensions increases quadratically with the number of classes. Perhaps the best way to use cost curves, or ROC curves, to visualize performance with multiple classes is to project the high-dimensional space into a set of 2-dimensional spaces, such as 1-class-versus-all-others for each class.

It is important to note that the mathematics underlying cost curve techniques, the methods for averaging curves and computing operating ranges, and the bootstrap methods for computing confidence intervals and statistical significance all extend trivially to any number of classes, although the bootstrap methods will suffer from sparsity of data if the number of classes is large. A number of researchers have looked at extensions to ROC curves for more than two classes (Srinivasan, 1999; Ferri et al., 2003). As the duality between the two representations, ROC and Cost curves, also holds for higher dimensions (Pottmann, 2001), we expect that these extensions can be easily applied to our representation.

A second limitation shared by ROC analysis is the cost model used (see Appendix A). We have assumed that the cost of a misclassification is associated with a class, not an individual example (Zadrozny et al., 2003), and that costs are additive. Furthermore, the claim that cost curves (and ROC curves) represent performance for "all" class distributions is restricted to changes in class priors that leave the within-class distributions, or likelihoods, unchanged (Webb and Ting, 2005).

We are also tied to one particular, albeit very general, loss function. It is an interesting question if other loss functions might be used while still maintaining most of the benefits of cost curves. A new representation called "Expected Performance Curves" (Bengio et al., 2005) could be regarded as a generalization of cost curves using different loss functions. Cost curves are produced in their framework by setting $\alpha = PC(+)$, $V1 = FN$, $V2 = FP$ and $C = V = \alpha * FN + (1 - \alpha) * FP$. But the generalization comes at a cost, losing much of the power and intuitive feel of our representation. It is worth exploring if there is a middle ground where the flexibility of Expected Performance Curves is combined with the strengths of cost curves.

Another limitation that ROC analysis and cost curves share is the lack of any effective way to show, in a single plot, the performance results obtained on several different datasets. This difficulty follows from the fact that these techniques use two dimensions to present the performance on a single dataset. By contrast, scalar measures are one-dimensional leaving the second dimension free to be used creatively for comparing performance on multiple datasets (for example, see Fig. 3 in (Cohen, 1995)).

This paper has primarily been concerned to demonstrate that cost curves overcome several deficiencies of ROC curves while retaining most of their desirable properties. However, there are certain circumstances in which ROC curves are superior to cost curves for visualizing performance. One of these was mentioned earlier—Section 5.1.4 showed that ROC curves are very well suited to visualizing the workforce utilization criterion. Visualization of this criterion is possible with cost curves but not as conveniently.

Figure 25 shows the ROC curve and corresponding cost lines for Naive Bayes on the Sonar dataset. There are many distinct classification thresholds, and therefore many ROC points and cost lines. With such a large number of classifiers it is not easy to see the entirety of an individual classifier's cost line across the entire $PC(+)$ range. By contrast, a single ROC point can be seen equally easily no matter how many other points are displayed along with it. This weakness of cost curves can be remedied simply by highlighting individual classifiers that are of interest.

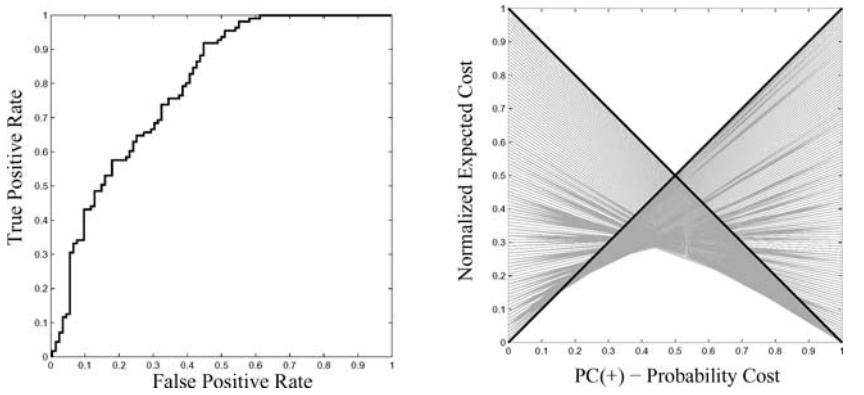


Fig. 25 (a) ROC curve for Naive bayes on the sonar dataset — (b) Corresponding cost lines (Rescaled Y)

The final advantage of ROC curves over cost curves is that ROC curves can be used when the task is to rank alternatives, instead of classifying them. Cost curves are only applicable to classification systems.[10]

7. Conclusions

This paper has presented cost curves, an alternative to ROC curves for visualizing the performance of 2-class classifiers in which performance (error rate or expected cost) is plotted as a function of the operating conditions—misclassification costs and class distributions summarized in a single number, $PC(+)$. Cost curves share many of ROC curves' desirable properties, but also visually support several crucial types of performance assessment that cannot be done easily with ROC curves, such as showing confidence intervals on a classifier's performance, and visualizing the statistical significance of the difference in performance of two classifiers. A software tool based on cost curves is freely available and provides touch-of-a-button visualization for all the analyses described in this paper.

Appendix

A. The cost model

We adopt in this paper the conventional cost model in which the cost associated with the classification of an example depends only on the example's class, and that the aggregate cost over a set of examples is the sum of their individual costs.

[10] This is the reason certain well-known visualization techniques have not been discussed in this paper, most notably precision-recall curves and lift curves. These techniques are for visualizing the performance of ranking systems, not classification systems. This paper is exclusively concerned with classification.

In the most general setting for two class problems, the cost matrix has four values and expected cost given by the following equation:

$$E[\texttt{Cost}] = \texttt{C}(+|+)\overbrace{*P(+|+)}^{TP}*P(+) + \texttt{C}(-|-)\overbrace{*P(-|-)}^{TN}*P(-)$$
$$+\texttt{C}(+|-)\underbrace{*P(+|-)}_{FP}*P(-) + \texttt{C}(-|+)\underbrace{*P(-|+)}_{FN}*P(+)$$

Substituting $1 - FN = TP$ and $1 - FP = TN$ and gathering the terms for FP and FN results in the following equation:

$$E[\texttt{Cost}] = \texttt{C}(+|+) * P(+) + \texttt{C}(-|-) * P(-)$$
$$+(\texttt{C}(-|+) - \texttt{C}(+|+)) * FN * P(+)$$
$$+(\texttt{C}(+|-) - \texttt{C}(-|-)) * FP * P(-)$$

The first two terms on the right hand side are independent of the classifier and represent the cost incurred when every example is correctly classified, i.e. when FP and FN are both zero. Since this cost-per-example is unavoidable, cost is usually taken to be the additional cost incurred above it. We can simplify further, by replacing the cost differences with single terms, resulting in the equation:

$$E[\texttt{Cost}] = \texttt{C}(-|+) * FN * P(+)$$
$$+\texttt{C}(+|-) * FP * P(-)$$

Acknowledgments We would like to thank the Natural Sciences and Engineering Research Council of Canada for the financial support that made this research possible, and Alberta Ingenuity for its funding of the Alberta Ingenuity Centre for Machine Learning. Thanks also to Peter Flach for in-depth discussions of many of the points presented in this paper.

References

Adams, N. M., & Hand, D. J. (1999). Comparing classifiers when misclassification costs are uncertain. *Pattern Recognition, 32*, 1139–1147.

Agarwal, S., Har-Peled, S., & Roth, D. (2005). A uniform convergence bound for the area under the ROC curve. In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (pp. 1–8).

Bengio, S., & Mariéthoz, J. (2004). The expected performance curve: a new assessment measure for person authentication. In: *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop* (pp. 9–16).

Bengio, S., Marithoz, J., & Keller, M. (2005). The expected performance curve. In: *Proceedings of the Second Workshop on ROC Analysis in ML* (pp. 9–16).

Bradford, J., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. E. (1998). Pruning decision trees with misclassification costs. In: *Proceedings of the Tenth European Conference on Machine Learning* (pp. 131–136).

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

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*(4), 261–283.

Cohen, W. (1995). Fast effective rule induction. In: *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 115–123).

Cortes, C., & Mohri, M. (2005). Confidence intervals for the area under the ROC curve. In: L.K. Saul, Y. Weiss, & L. Bottou, (eds.): *Advances in neural information processing systems 17*. MIT Press, (pp. 305–312).

Domingos, P. (1999) MetaCost: A general method for making classifiers cost-sensitive. In: *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*. (pp. 155–164).

Drummond, C., & Holte, R. C. (2000a). Explicitly representing expected cost: An alternative to ROC representation. In: *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*. (pp. 198–207).

Drummond, C., & Holte, R. C. (2000b). Exploiting the cost (In)sensitivity of decision tree splitting criteria. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. (pp. 239–246).

Drummond, C., & Holte, R. C. (2003). C4.5, Class imbalance, and cost sensitivity: why undersampling beats oversampling. In: *Proceedings of the Twentieth International Conference on Machine Learning: Workshop - Learning from Imbalanced Data Sets II*. (pp. 1–8).

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and scene analysis*. New York: Wiley.

Dukic, V., & Gatsonis, C. (2003) Meta-analysis of diagnostic test accuracy assessment studies with varying number of thresholds. *Biometrics 59*(4), 936–946.

Efron, B., & Tibshirani, R. (1993). *An Introduction to the bootstrap*. London: Chapman and Hall.

Fawcett, T. (2003). ROC Graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Labs.

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Journal of Data Mining and Knowledge Discovery 1*, 291–316.

Ferri, C., Flach, P. A., & Hernández-Orallo, J. (2002). Learning decision trees using the area under the ROC curve. In: *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 139–146).

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

Flach, P. (2003). The geometry of ROC Space: Understanding machine learning metrics through ROC isometrics'. In: *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 194–201).

Flach, P. A. (2004). ICML Tutorial: The many faces of ROC analysis in machine learning. http://www.cs.bris.ac.uk/∼flach/ICML04tutorial/index.html.

Halpern, E. J., Albert, M., Krieger, A. M., Metz, C. E., & Maidment, A. D. (1996). Comparison of receiver operating characteristic curves on the basis of optimal operating points. *Statistics for Radiologists, 3*, 245–253.

Hand, D. J. (1997). *Construction and assessment of classification rules*. New York: Wiley.

Hilden, J., & Glasziou, P. (1996). Regret graphs, diagnostic uncertainty, and youden's index. *Statistics in Medicine, 15*, 969–986.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*(1), 63–91.

Japkowicz, N., Myers, C., & Gluck, M. (1995). A novelty detection approach to classification. In: *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence* (pp. 518–523).

Jensen, K., Muller, H. H., & Schafer, H. (2000). Regional confidence bands for ROC curves. *Statistics in Medicine, 19*(4), 493–509.

Karwath, A., & King, R. D. (2002). Homology induction: The use of machine learning to improve sequence similarity searches. *BMC Bioinformatics 3*, 11.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning, 30*, 195–215.

Kukar, M., & Kononenko, I (1998). Cost-sensitive learning with neural networks. In: *Proceedings of the Thirteenth European Conference on Artificial Intelligence* (pp. 445–449).

Ling, C. X., & Li, C. (1998). Data mining for direct marketing: Problems and solutions. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 73–79).

Ma, G., & Hall, W. J. (1993). Confidence bands for receiver operating characteristic curves. *Medical Decision Making, 13*(3), 191–197.

Macskassy, S. A., Provost, F., & Rosset, S. (2005). ROC confidence bands: An empirical evaluation. In: *Proceedings of the Twenty-Second International Conference on Machine Learning* (pp. 537–544).

Margineantu, D. D. (2002). Class probability estimation and cost-sensitive classification decisions. In: *Proceedings of the Thirteenth European Conference on Machine Learning*. (pp. 270–281).

Margineantu, D. D., & Dietterich, T. G. (2000). Bootstrap methods for the cost-sensitive evaluation of classifiers. In: *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 582–590).

McNeil, B. J., & Hanley, J. A. (1984). Statistical approaches to the analysis of receiver operating characteristic (ROC) curves. *Medical Decision Making, 4*(2), 137–150.

Metz, C. E., & Kronman, H. B. (1980). Statistical significance tests for binormal ROC curves. *Journal of Mathematical Psychology, 22*(3), 218–243.

Metz, C. E., Wang, P. L., & Kronman, H. B., (1983). A new approach for testing the significance of differences between ROC curves measured from correlated data. In: *Proceedings of the Eighth Conference on Information Processing in Medical Imaging* (pp. 432–445).

Miller, K., Ramaswami, S., Rousseeuw, P., Sellares, T., Souvaine, D., Streinu, I., & Struyf, A. (2001). Fast implementation of depth contours using topological sweep. In: *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete algorithms* (pp. 690–699).

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. In: *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 217–225).

Platt, R. W., Hanley, J. A., & Yang, H. (2000). Bootstrap confidence intervals for the sensitivity of a quantitative diagnostic test. *Statistics in Medicine19* (3), 313–322.

Pottmann, H. (2001). Basics of projective geometry. An institute for mathematics and its applications tutorial. Geometric Design: Geometries for CAGD http://www.ima.umn.edu/multimedia/spring/tut7.html.

Preparata, F. P., & Shamos, M. I. (1988). *Computational Geometry, An Introduction*, Text and Monographs in Computer Science. New York: Springer-Verlag.

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 43–48).

Provost, F., & Fawcett, T. (1998). Robust classification systems for imprecise environments. In: *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. (pp. 706–713).

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

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In: *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 43–48).

Quinlan, J. R. (1993). *C4.5 Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann.

Radivojac, P., Sivalingam, K., & Obradovic, Z. (2003). Learning from class-imbalanced data in wireless sensor networks. In: *Proceedings of the Sixty-Second IEEE Semiannual Vehicular Technology Conference* (pp. 3030–3034).

Saitta, L., & Neri, F. (1998). Learning in the "Real World".*Machine Learning, 30*(2-3), 133–163.

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

Swets, J. A. (1967). *Information Retrieval Systems*. Cambridge, Massachusetts: Bolt, Beranek and Newman.

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

Swets, J. A., &. Pickett, R. M. (1982). *Evaluation of Diagnostic Systems : Methods from Signal Detection Theory*. New York: Academic Press.

Tilbury, J., Eetvelt, P. V., Garibaldi, J., Curnow, J., & Ifeachor, E. (2000). Receiver operating characteristic analysis for intelligent medical systems—a new approach for finding non-parametric confidence intervals. *IEEE Transactions Biomedical Engineering, 47*(7), 952–963.

Ting, K. M. (2000). An empirical study of metacost using boosting algorithms. In: *Proceedings of the Eleventh European Conference on Machine Learning* (pp. 413–425).

Ting, K. M. (2002). Issues in classifier evaluation using optimal cost curves. In: *Proceedings of The Nineteenth International Conference on Machine Learning* (pp. 642–649).

Ting, K. M. (2004). Matching model versus single model: A study of the requirement to match class distribution using decision trees. In: *Proceedings of the Fifteenth European Conference on Machine Learning* (pp. 429–440).

Turney, P. D. (1995). Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research, 2*, 369–409.

van Rijsbergen, C. J. (1979). *Information retrieval*. London: Butterworths.

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

Webb, G. I. (1996). Cost-sensitive specialization. In: *Proceedings of the Fourteenth Pacific Rim International Conference on Artificial Intelligence* (pp. 23–34).

Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research, 19*, 315–354.

Witten, I. H., & Frank, E.,(2005). *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann.

Yan, L., Dodier, R., Mozer, M. C., & Wolniewicz, R. (2003). Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In: *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 848–855).

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. In: *Proceedings of the Third IEEE International Conference on Data Mining* (pp. 435–442).

Zou, K. H., Hall, W. J., & Shapiro, D. E. (1997). Smooth non-parametric roc curves for continuous diagnostic tests. *Statistics in Medicine, 16*, 2143–56.