Cost-Sensitive Classifier Evaluation

Robert C. Holte
Department of Computing Science,
University of Alberta,
Edmonton, Alberta, Canada, T6G 2E8
holte@cs.ualberta.ca

Chris Drummond
Institute for Information Technology,
National Research Council Canada,
Ottawa, Ontario, Canada, K1A 0R6
Chris.Drummond@nrc-cnrc.gc.ca

ABSTRACT

Evaluating classifier performance in a cost-sensitive setting is straightforward if the operating conditions (misclassification costs and class distributions) are fixed and known. When this is not the case, evaluation requires a method of visualizing classifier performance across the full range of possible operating conditions. This paper reviews the classic technique for classifier performance visualization – the ROC curve – and argues that it is inadequate for the needs of researchers and practitioners in several important respects. It then shows that a different way of visualizing classifier performance – the cost curve introduced by Drummond and Holte at KDD'2000 – overcomes these deficiencies. A software package supporting all the cost curve analysis described in this paper is available by contacting the first author.

1. INTRODUCTION

In this paper[1], our focus is on the visualization of a classifier's performance. This is one of the attractive features of ROC analysis – the tradeoff between false positive rate and true positive rate can been seen directly. A good visualization of classifier performance allows an experimenter to immediately see how well a classifier performs and to compare two classifiers – to see when, and by how much, one classifier outperforms others.

We restrict the discussion to classification problems in which there are only two classes. The main point of this paper is to show that, even in this restricted case, ROC curves are not a good visualization of classifier performance. In particular, they do not allow any of the following important experimental questions to be answered visually:

- 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?
- 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. the results of 5-fold cross-validation)?
- what is the 90% confidence interval for classifier C's performance?
- what is the significance (if any) of the difference between the performance of classifier C1 and the performance of classifier C2?

The paper is organized around these questions. After a brief review of essential background material, there is a section devoted to each of these questions.

2. BACKGROUND

For 2-class classification problems ROC space is a two-dimensional plot with true positive rate (TP) on the y-axis and false positive rate (FP) on the x-axis. A single confusion matrix thus produces a single point in ROC space. An ROC curve is formed from a sequence of such points, including (0,0) and (1,1), connected by line segments. The method used to generate the sequence of points for a given classifier (or learning algorithm) depends on the classifier. For example, with Naive Bayes [5, 9] an ROC curve is produced by varying its threshold parameter.

An ROC curve implicitly conveys information about performance across all possible combinations of misclassification costs and class distributions[2]. We use the term "operating point" to refer to a specific combination of misclassification costs and class distributions.

One point in ROC space dominates another if it 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 points A is dominated by another B when each point in A is dominated by some point B and no point in B is dominated by a point in A.

[1] An early version of this paper appeared in [4].

[2] "All" distributions and costs with certain standard restrictions. For class distributions "all" means any prior probabilities for the classes while keeping the class-conditional probabilities, or likelihoods, constant [16]. For costs "all" means all combinations of costs such that a misclassification is more costly than a correct classification.

UBDM'05, August 21, 2005, Chicago, Illinois, USA.

Cost curves were introduced in [2]. Performance (expected cost normalized to be between 0 and 1) is plotted on the y-axis. Operating points are plotted on the x-axis after being normalized to be between 0 and 1 by combining the parameters defining an operating point in the following way:

$$PC(+) = \frac{p(+)\mathtt{C}(-|+)}{p(+)\mathtt{C}(-|+) + p(-)\mathtt{C}(+|-)} \quad (1)$$

where `C(-|+)`is the cost of misclassifying a positive example as negative, `C(+|-)`is the cost of misclassifying a negative example as positive, $p(+)$ is the probability of a positive example, and $p(-) = 1 - p(+)$. The motivation for this PC definition, and cost curves more generally, originates in the simple situation when misclassification costs are equal. In this case $PC(+) = p(+)$ and the y-axis becomes error rate, so the cost curve plots how error rate varies a function of the prevalence of positive examples. The PC definition generalizes this idea to the case when when misclassification costs are not equal. The PC formula is intimately tied to the definition of the slope of a line in ROC space, which plays a key role in ROC analysis. The x-axis of cost space is "slope in ROC space" normalized to be between 0 and 1 instead of being between 0 and infinity (historically this is how cost curves were invented).

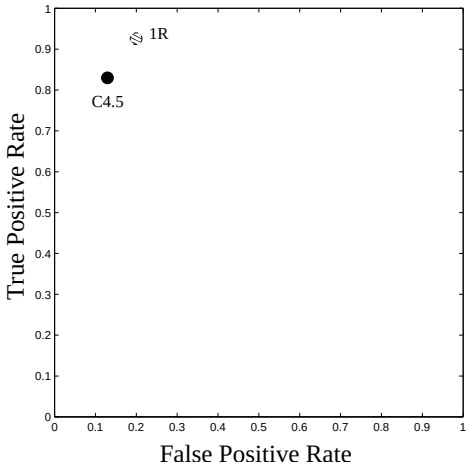

Figure 1: Two ROC points

There is a point/line duality between ROC space and cost space, meaning 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. A classifier represented by the point (FP,TP) in ROC space is a line in cost space that has $y = FP$ when $x = 0$ and $y = 1 - TP$ when $x = 1$. The set of points defining an ROC curve become a set of lines in cost space For example, Figure 1 shows the ROC points for two classifiers for the Japanese credit dataset from the UCI repository [1]: the dashed point is for the decision stump produced by 1R [7], the solid point is for the decision tree produced by C4.5 [12]. Each point becomes a line in cost space, as shown in Figure 2.

Given the cost lines for a set of classifiers, a cost curve is created by deciding which classifier to use for every possible operating point. If, for each operating point, the classifier is chosen that minimizes normalized expected cost, the resulting cost curve is the lower envelope of the given cost lines, the dual of the ROC convex hull.

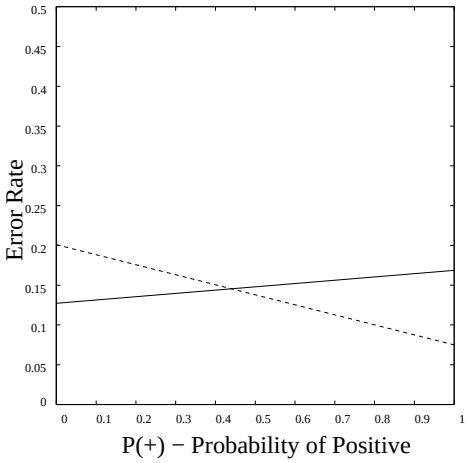

Figure 2: Corresponding Cost Lines

3. VISUALIZING PERFORMANCE

ROC analysis does not directly commit to any particular measure of performance. This is sometimes considered an advantageous feature of ROC curves. For example, Van Rijsbergen [15] quotes Swets [13] who argues that this is useful as it measures "discrimination power independent of any 'acceptable criterion' employed". Provost and Fawcett substantiate this argument by showing that ROC dominance implies superior performance for a variety of commonly-used performance measures [10]. The ROC representation allows an experimenter to see quickly if one classifier dominates another and therefore, using the convex hull, to identify potentially optimal classifiers visually without committing to a specific performance measure.

For example, Figure 3 shows a set of ROC points for C4.5 on the sonar data set from the UCI collection. Each point corresponds to a different setting of the classification threshold parameter. 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 the decision trees perform well overall.

Being independent of any particular performance measure can be a disadvantage when one has a particular performance measure in mind. ROC curves do not visually depict the quantitative performance of a classifier or the difference in performance between two classifiers.

The solid lines in Figure 4 are the the cost lines for the classifiers whose ROC points are shown in Figure 3. Each cost line in Figure 4 corresponds to one of the individual ROC points in Figure 3. 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 as is the fact that 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

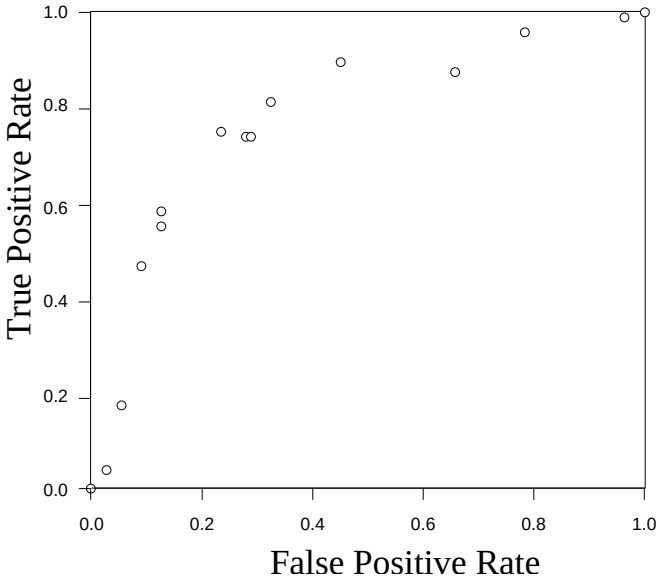

Figure 3: ROC Points for C4.5 on the Sonar dataset

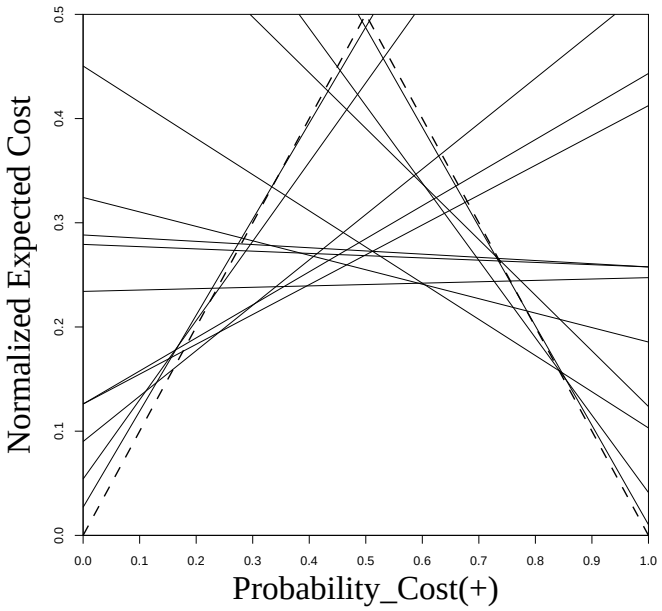

Figure 4: Cost Lines Corresponding to Figure 3

threshold that result in near-optimal normalized expected cost when $PC(+)$ is near 0.5.

4. COMPARING A CLASSIFIER TO THE TRIVIAL CLASSIFIERS

In an ROC diagram 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 cost lines for these classifiers are the dashed lines shown in Figure 4. The dashed line from (0,0) to (0.5,0.5) is the cost line for the classifier that classifies all examples as negative, and the diagonal line from (0.5,0.5) to (1,0) is the cost line for the classifier that classifies all examples as positive.

The operating range of a classifier is the set of operating points where it outperforms the trivial classifiers. A classifier should not be used outside its operating range, since one can obtain superior performance by assigning all examples to a single class.

The operating range of a classifier cannot be seen readily in an ROC curve. It is defined by the slopes of the lines tangent to the ROC curve and passing through (0,0) and (1,1). By contrast, a classifier's operating range can be immediately read off of a cost curve: it is defined by the PC values where the cost curve intersects the diagonal lines representing the trivial classifiers. For example, in Figure 4 it can be seen immediately that all the classifiers being considered perform worse than a trivial classifier when $PC < 0.15$ or $PC > 0.85$.

5. CHOOSING BETWEEN CLASSIFIERS

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 crossover point of the ROC curves has little to do with the range. Consider the ROC curves for two classifiers, the dotted and dashed curves of Figure 5. The solid line is the isoperformance 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.

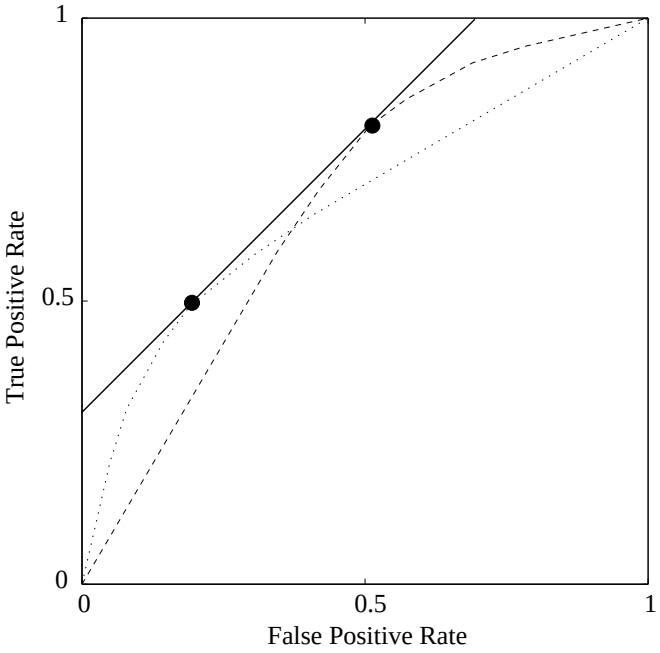

Figure 5: ROC Space Crossover

Figure 6 shows the cost curves corresponding to the ROC curves in Figure 5. It can immediately be seen that the dotted line has a lower expected cost and therefore outperforms the dashed line when $PC < 0.5$ and vice versa.

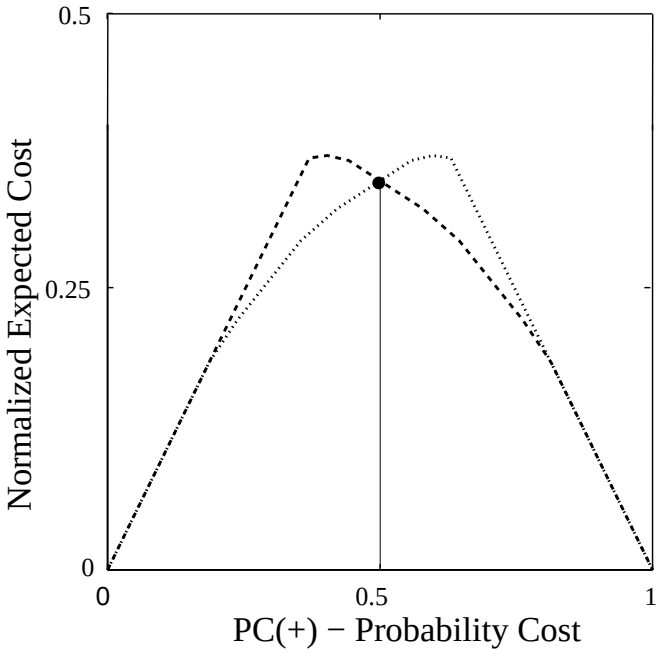

Figure 6: Corresponding Cost Space Crossover

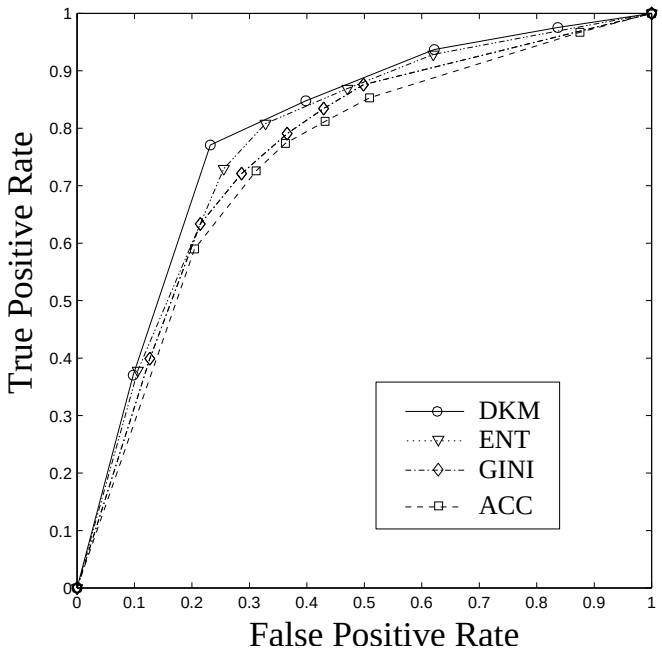

Figure 7: ROC Curves for Various C4.5 Splitting Criteria on the Sonar Dataset

6. COMPARING PERFORMANCE

The ROC curves in Figure 7 show the performance of the decision trees built on the Sonar dataset by C4.5 with different splitting criteria [3]. 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 8 shows the corresponding cost curves. The tangled ROC curves are now cleanly separated, and the vertical distance between two cost curves directly indicates the difference in their performance. 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.

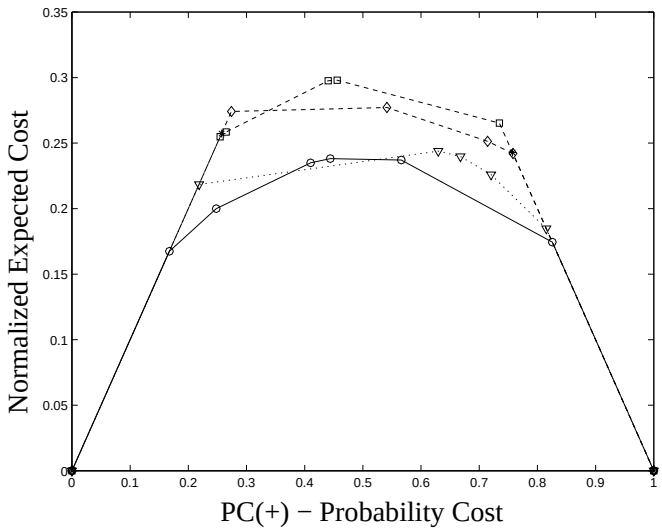

Figure 8: Cost Curves Corresponding to Figure 7

7. AVERAGING MULTIPLE CURVES

Each solid line in Figure 9 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 samples, or some other cause of random fluctuation in performance, and therefore their average can be used as an estimate of expected performance.

There is no universally agreed-upon method of averaging ROC curves. Swets and Pickett [14] suggest two methods, pooling and "averaging", and Provost et al. [11] 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 Figure 9 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).

An important shortcoming of all these methods of averaging ROC curves 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 isoperformance line that connects the central vertices of the two ROC curves in Figure 9. The vertical and horizontal averages do not touch this line, they are well below it.

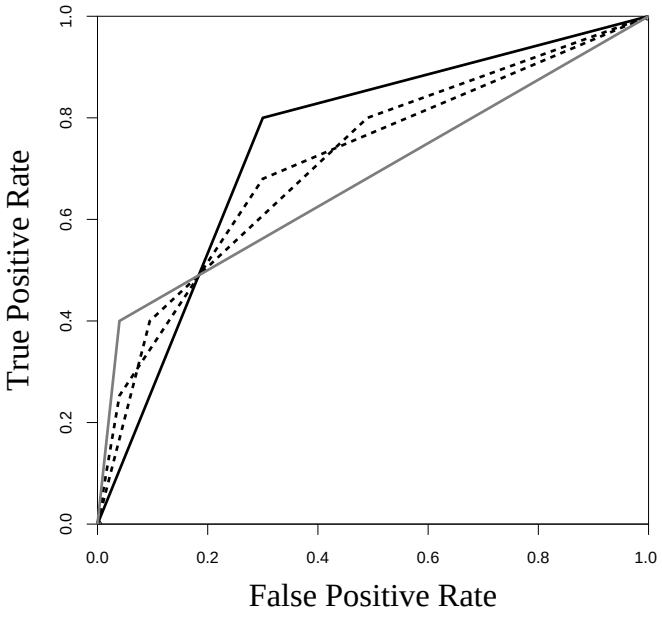

Figure 9: Vertical and Horizontal Averages of two ROC curves

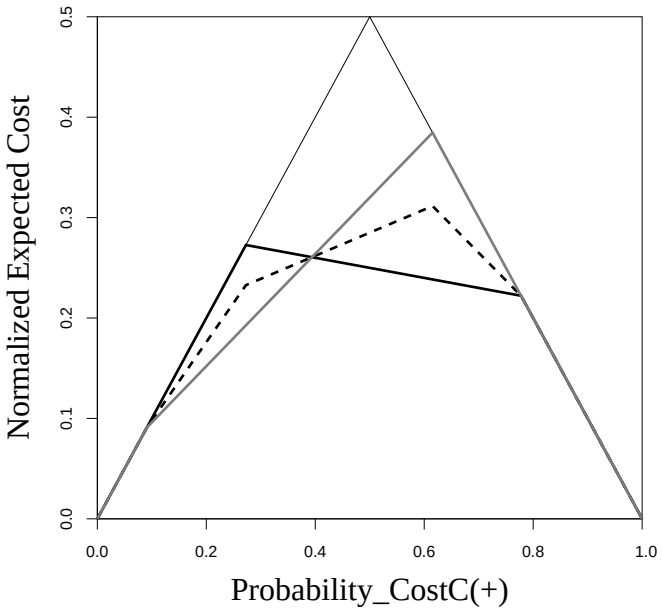

Figure 10: Average Cost Curves

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 Figure 10 are the ROC curves from Figure 9 translated into cost curve lower envelopes. The expected performance based on these two cost curves is given by the bold dotted line.

8. CONFIDENCE INTERVALS ON COSTS

The measure of classifier performance is derived from a confusion matrix produced from some sample of the data. As there is likely to be variation between samples, the measure is, itself, a random variable. So some estimate of its variance is useful, which usually takes the form of a confidence interval. The most common approach to producing a confidence interval is to assume that the distribution of the estimate belongs to, or is closely approximated by, some parametric family such as Gaussian or Student-t. An alternative, data driven, method has become popular in recent times which does not make any parametric assumptions. Margineantu and Dietterich [8] described how one such non-parametric approach called the bootstrap [6] can be used to generate confidence intervals for predefined cost values. We use a similar technique, but for the complete range of class distributions and misclassification costs.

The bootstrap method is based on the idea that new samples generated from the available data are related to that data in the same way that the available data relates to the original population. Thus the variance of an estimate based on the new samples should be a good approximation to its true variance. Confidence limits are produced by resampling from the original matrix to create numerous new confusion matrices. The exact way bootstrapping is carried out depends on the sampling scheme. We propose a resampling method analogous to stratified cross validation, in which the class frequency is guaranteed to be identical in every sample.

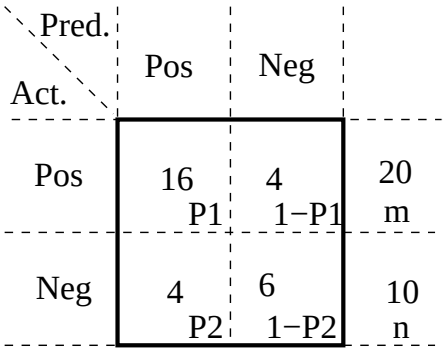

Figure 11: Binomial Sampling

For example, consider the confusion matrix of Figure 11. 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 fix the row totals at 20 and 10, and treat the two rows as independent binomial distributions with probabilities $P1 = 16/20 = 0.8$ and $P2 = 4/10 = 0.4$, respectively, of assigning a positive label to an example.

A new matrix is produced by randomly sampling according to these two binomial distributions until the number of positive and negative instances equal the corresponding row totals. For each new confusion matrix, a dotted line is plotted in Figure 12 representing the new estimate of classifier performance. For ease of exposition, we generated 100 new confusion matrices (typically at least 500 are used for an

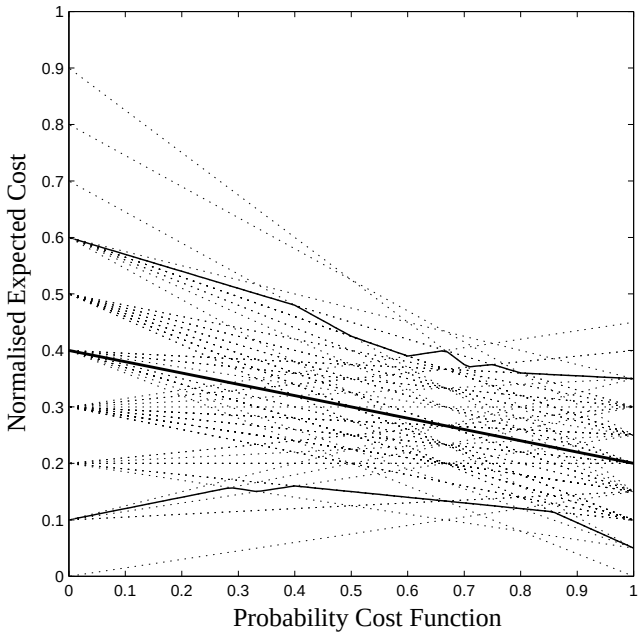

Figure 12: 90% Confidence Interval on a Cost Curve

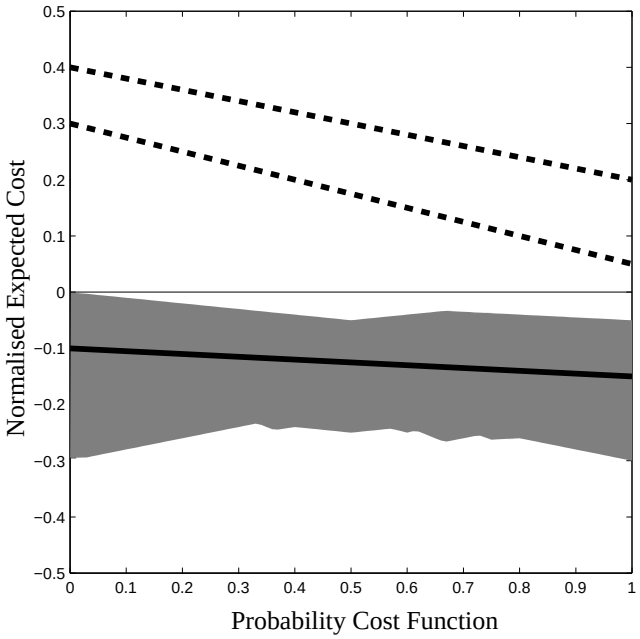

Figure 13: Confidence Interval for the Difference, High Correlation

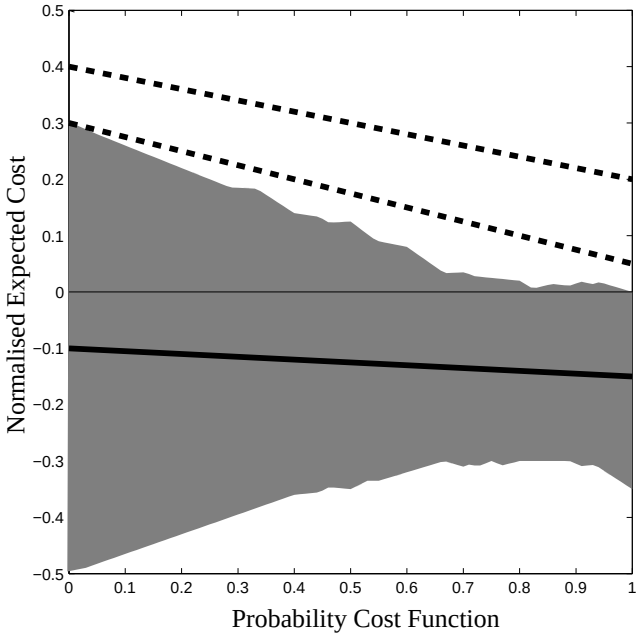

Figure 14: Confidence Interval for the Difference, Low Correlation

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. This process is repeated for each small increment in the PC(+) value. The centre bold line in Figure 12 represents the performance of the classifier based on the original confusion matrix. The other two bold lines are the upper and lower confidence limits for this classifier.

9. TESTING IF PERFORMANCE DIFFERENCES ARE SIGNIFICANT

The difference in performance of two classifiers is statistically significant if the confidence interval around the difference does not contain zero. The method presented in the previous section can be extended to do this, by resampling the confusion matrices of the two classifiers simultaneously, taking into account the correlation between the two classifiers. A single resampling thus produces a pair of confusion matrices, one for each classifier, and therefore two lines in cost space. However, instead of plotting the two lines, we plot the difference between the two lines. We can repeat this process a large number of times to get a large number of lines and then, as above, extract a 90% confidence interval from this set of lines. This is the confidence interval around the difference between the classifiers' performances.

The thick continuous line at the bottom of Figure 13 represents the mean difference between performance of the two classifiers (which are shown in the figure as bold dashed lines). The shaded area represents the confidence interval of the difference, calculated as just described. As the difference can range from -1 to $+1$ the y-axis has been extended. Here we see that the confidence interval does not contain zero, so the difference between the classifiers is statistically significant. Figure 14 shows two classifiers with the same individual confusion matrices but with their 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 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 Figure 15, where the difference is significant only if $PC > 0.7$.

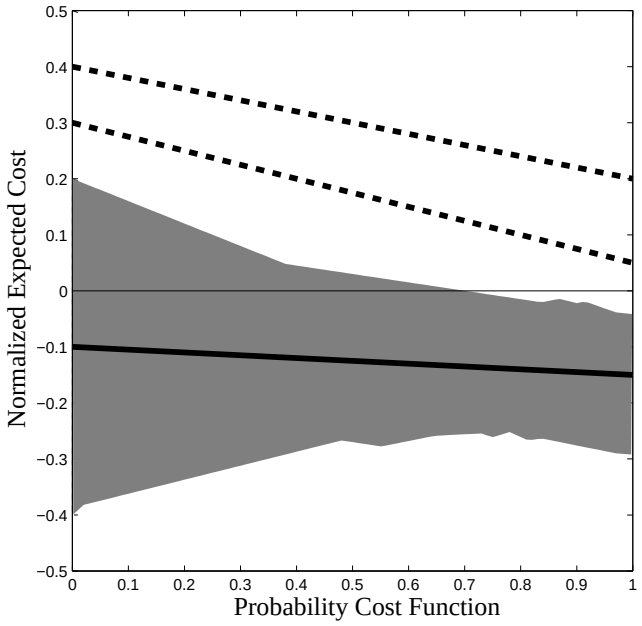

Figure 15: Confidence Interval for the Difference, Medium Correlation

10. CONCLUSIONS

This paper has demonstrated shortcomings of ROC curves for visualizing classifier performance, and shown that cost curves overcome these problems. We do not, however, contend that cost curves are always better than ROC curves. For example, for visualizing the workforce utilization measure of performance[10], ROC curves are distinctly superior to cost curves. But for many common visualization requirements, cost curves are by far the best alternative and we recommend their routine use instead of ROC curves for these purposes.

A software package supporting all the cost curve analysis described in this paper is available by contacting the first author.

11. ACKNOWLEDGEMENTS

We would like to acknowledge Alberta Ingenuity Fund for its support of this research through the funding of the Alberta Ingenuity Centre for Machine Learning.

12. REFERENCES

[1] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, University of California, Irvine, CA . www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[2] Chris Drummond and Robert C. Holte. Explicitly representing expected cost: An alternative to roc representation. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207, 2000.

[3] Chris Drummond and Robert C. Holte. Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 239–246, 2000.

[4] Chris Drummond and Robert C. Holte. What ROC Curves can't do (and Cost Curves can). In *Proceedings of the 1st Workshop on ROC Analysis in Artificial Intelligence (held in conjunction with ECAI 2004)*, pages 19–26, 2004.

[5] R. O. Duda and P. E. Hart. *Pattern Classification and scene analysis*. Wiley, New York, 1973.

[6] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, London, 1993.

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

[8] Dragos D. Margineantu and Thomas G. Dietterich. Bootstrap methods for the cost-sensitive evaluation of classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 582–590, 2000.

[9] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk. Reducing misclassification costs. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 217–225, San Francisco, 1994. Morgan Kaufmann.

[10] Foster Provost and Tom Fawcett. Robust classification systems for imprecise environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 706–713, Menlo Park, CA, 1998. AAAI Press.

[11] Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 43–48, San Francisco, 1998. Morgan Kaufmann.

[12] J. Ross Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[13] J. A. Swets. *Information Retrieval Systems*. Bolt, Beranek and Newman, Cambridge, Massachusetts, 1967.

[14] John A. Swets and Ronald M. Pickett. *Evaluation of diagnostic systems : methods from signal detection theory*. Academic Press, New York, 1982.

[15] C. J Van Rijsbergen. *Information retrieval*. Butterworths, London, 1979.

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