Analysis and Visualization of Classifier Performance with Nonuniform Class and Cost Distributions

Foster Provost
NYNEX Science and Technology
400 Westchester Avenue
White Plains, New York 10604
foster@nynexst.com

Tom Fawcett
NYNEX Science and Technology
400 Westchester Avenue
White Plains, New York 10604
fawcett@nynexst.com

Abstract

Applications of machine learning have shown repeatedly that the standard assumptions of uniform class distribution and uniform misclassification costs rarely hold. Little is known about how to select classifiers when error costs and class distributions are not known precisely at training time, or when they can change. We present a method for analyzing and visualizing the performance of classification methods that is robust to changing distributions and allows a sensitivity analysis if a range of costs is known. The method combines techniques from ROC analysis, decision analysis and computational geometry, and adapts them to the particulars of analyzing learned classifiers. We then demonstrate analysis and visualization properties of the method.

Introduction

Existing classifier learning methods are rarely sensitive to skewed class and cost distributions. Even methods that are sensitive break down when class and cost distributions are not known precisely at training time, which they rarely are, or when the distributions can change. Furthermore, when applying classification algorithms to real-world tasks, typically many different classfiers are generated by varying algorithm parameters, training distributions, or the learning algorithm itself. This leads potentially to a very large number of classifiers to be evaluated. It is important not only to identify the best classifier under fixed conditions, but also to understand how classifiers compare in general and how recommendations would change if conditions change.

This paper's contribution is a method for analyzing and visualizing classifier performance. The method accomodates imprecise and changing class and cost distributions. It can handle a large number of classifiers and allows sensitivity analysis if bounds can be placed on the distributions.

The *ROC convex hull* method combines techniques from ROC analysis, decision analysis and computational geometry. The method decouples classifiers' performance from specific class and cost distributions, and may be used to visualize the subset of methods that are optimal under any cost and class distribution assumptions. Constraints on the cost and class distributions sectionalize the hull into regions containing the optimal classifiers under the given constraints. The method is incremental and easily incorporates new and varied classifiers, including classifiers hand-crafted by human experts.

After discussing background, motivation, and existing methods from decision theory and ROC analysis, we present the ROC convex hull method, which is a hybrid of these techniques. We then demonstrate its usefulness visually through a series of examples taken from a fraud detection application.

The Inadequacy of Accuracy

A tacit assumption in the use of accuracy as a classifier evaluation metric is that the class distribution among examples is *constant and relatively balanced.* In the real world this is rarely the case. Classifiers are often used to sift through a large population of normal or uninteresting entities in order to find a relatively small number of unusual ones; for example, screening blood samples, looking for defrauded customers, or checking for defective parts. Because the unusual or interesting class is rare among the general population, the class distribution is very skewed (Ezawa, Singh, & Norton, 1996; Fawcett & Provost, 1996; Saitta, Giordana, & Neri, 1995).

Evaluation based on accuracy breaks down as the class distribution becomes more skewed. Consider a domain where the classes appear in a 999:1 ratio. A simple rule, always classify as the maximum likelihood class, yields a 99.9% accuracy. Presumably this is not satisfactory if a non-trivial solution is sought. Skews of 10^2 are common in fraud detection and skews greater than 10^6 have been reported in other classifier learning work (Clearwater & Stern, 1991).

Evaluation by classification accuracy also tacitly assumes *equal error costs*—that a false positive error is equivalent to a false negative error. In the real world this is rarely the case, because classifications lead to actions. Actions may be as disparate as informing a patient of a disease, routing a letter to a specific bin, or moving the control rods of a nuclear reactor. Actions have consequences, sometimes grave, and rarely are mistakes evenly weighted in their cost. Indeed, it is hard to imagine a domain in which a learning system may be indifferent to whether it makes a false positive or a false negative error. In such cases, accuracy maximization should be replaced with cost minimization.

The problems of unequal error costs and uneven class distributions are closely related. Indeed, it has been suggested that high-cost instances can be compensated for by increasing their prevalence in an instance set (Breiman, Friedman, Olshen, & Stone, 1984). Unfortunately, very little work has been published on either problem. There exist several dozen articles (Turney, 1996) in which techniques are suggested, but little is done to evaluate and compare them (the article of Pazzani, et al. (1994) being the exception). Furthermore, the literature provides little guidance in situations where cost and class distributions are not known precisely or can change.

Evaluating and Visualizing Classifier Performance

Let $p(P|I)$ be the estimated probability that instance I is positive. Let N_{FP} and N_{FN} be the total number of false positive and false negative errors, respectively, on a test set and let $c(FP)$ and $c(FN)$ be the cost of a single false positive and false negative error, respectively. The true positive rate, TP, and false positive rate, FP, of a classifier are:

$$TP = p(\text{Classify Positive} \mid \text{Positive}) \approx \frac{\text{positives correctly classified}}{\text{total positives}}$$

$$FP = p(\text{Classify Positive} \mid \text{Negative}) \approx \frac{\text{negatives incorrectly classified}}{\text{total negatives}}$$

If a classifier produces posterior probabilities, decision analysis gives us a way to produce cost-sensitive classifications from the classifier (Weinstein & Fineberg, 1980). Classifier error frequencies can be used to approximate probabilities (Pazzani et al., 1994). For an instance I, the decision to emit a positive classification is:

$$[1 - p(P|I)] \cdot c(FP) < p(P|I) \cdot c(FN)$$

Regardless of whether a classifier produces probabilistic or binary classifications, its cost on a test set can be evaluated empirically as:

$$\text{Cost} = N_{FP} \cdot c(FP) + N_{FN} \cdot c(FN)$$

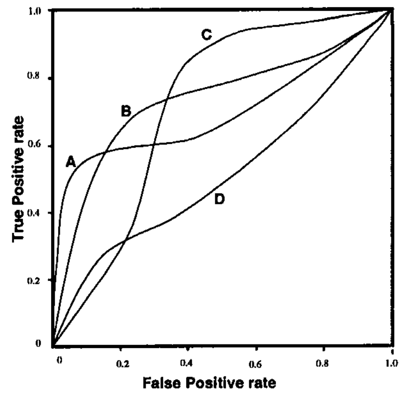

Figure 1: An ROC graph of four classifiers

Given a set of classifiers, a set of examples, and a precise cost function, most work on cost-sensitive classification uses an equation such as this to rank the classifiers according to cost and chooses the minimum.[1]

However, practical classifier evaluation is still problematic because the learning context is often uncertain. In practice, error costs can seldom be stated exactly; distributions and error costs change over time, and in some domains the actions of the classifier feed back to its environment. In our fraud detection work, the specification of costs is imprecise and class distributions change regularly (though short-term forecasts can be relatively good). One cannot ignore either type of distribution, nor can one assume that the distributions are precise and static.

A further limitation of decision-theoretic evaluation is that it does not provide a means of high-level visualization, and sensitivity analysis is complex. Identifying the best classifier under fixed conditions is important, but we must also understand how classifiers compare in general and how recommendations would change if conditions change—either slightly or dramatically. What is needed is a method of analysis of classifier performance that accomodates the dynamics and imprecision of real-world environments.

Receiver Operating Characteristic (ROC) graphs have long been used in signal detection theory to depict tradeoffs between hit rate and false alarm rate (Egan, 1975). ROC analysis has been extended for use in visualizing and analyzing the behavior of diagnostic systems (Swets, 1988), and is used for visualization in medicine (Beck & Schultz, 1986). ROC graphs are occasionally mentioned in classifier learning work (Ezawa et al., 1996; Catlett, 1995).

We will use the term *ROC space* to denote the

[1]For simplicity of presentation, we assume that correct classifications do not have associated costs and that benefit information is folded into error cost.

classifier performance space used for visualization in ROC analysis. On an ROC graph, TP is plotted on the Y axis and FP is plotted on the X axis. These statistics vary together as a threshold on a classifier's continuous output is varied between its extremes, and the resulting curve is called the ROC curve. An ROC curve illustrates the error tradeoffs available with a given classifier. Figure 1 shows a plot of the performance of four classifiers, A through D, typical of what we see in the creation of alarms for fraud detection (Fawcett & Provost, 1996).

For orientation, several points on an ROC graph should be noted. The lower left point $(0,0)$ represents the strategy of never alarming; this yields no false alarms but no true positives. Similarly, the upper right point $(1,1)$ represents the strategy of always alarming. The point $(0,1)$ represents perfection: no false alarms are issued and all possible true alarms are issued. The line $y = x$ (not shown) represents the strategy of random guessing.

Informally, one point in ROC space is better than another if it is to the northwest (TP is higher, FP is lower, or both). An ROC graph allows an informal visual comparison of a set of classifiers. In Figure 1, curve A is better than curve D because it dominates in all points.

ROC graphs illustrate the behavior of a classifier *without regard to class distribution or error cost*, and so they decouple classification performance from these factors. Unfortunately, while an ROC graph is a valuable visualization technique, ROC analysis does a poor job of aiding the choice of classifiers. Only when one classifier clearly dominates another over the entire performance space can it be declared better. Consider the classifiers shown in Figure 1. Which is best? The answer depends upon the performance requirements, *i.e.*, the error costs and class distributions in effect when the classifiers are to be used.

Some researchers advocate choosing the classifier that maximizes the product $(1 - FP) \cdot TP$. Geometrically, this corresponds to fitting rectangles under every ROC curve and choosing the rectangle of greatest area. This and other approaches (Swets, 1988; Beck & Schultz, 1986) calculate average performance over the entire performance space. These approaches may be appropriate if costs and class distributions are completely unknown, but typically some domain-specific information is available. In such cases, we can do better by folding class and cost distribution information into the visual analysis.

A Hybrid Method: The ROC Convex Hull

In this section, we combine decision analysis with ROC analysis, and adapt them for analyzing and visualizing the performance of learned classifiers. The method is based on three high-level principles. First, the ROC space is used to separate classification performance from class and cost distribution information. Second, decision-analytic information is projected onto the ROC space. Third, we use a convex hull to identify the subset of methods that are potentially optimal under any conditions.

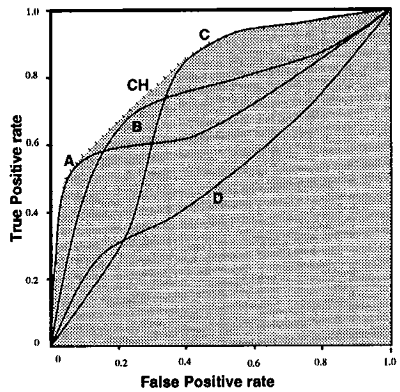

Figure 2: The ROC convex hull

Iso-performance lines

Decision theory gives us a precise way to quantify the performance of classifiers under fixed class and cost distribution assumptions, but organizing and visualizing this information can be difficult. By separating classification performance from class and cost distribution assumptions, we can project the decision goal onto ROC space for a neat visualization. Formally, let the prior probability of a positive example be $p(P)$, so the prior probability of a negative example is $p(N) = 1 - p(P)$; let the cost of a false positive error be $c(FP)$, and let the cost of a false negative error be $c(FN)$. The expected cost of the classifier represented by a point (TP,FP) in ROC space is:

$$p(P) \cdot (1 - TP) \cdot c(FN) + p(N) \cdot FP \cdot c(FP)$$

Therefore, two points, (TP_1,FP_1) and (TP_2,FP_2), have the same performance if

$$\frac{TP_2 - TP_1}{FP_2 - FP_1} = \frac{p(N)c(FP)}{p(P)c(FN)}$$

This equation defines the slope of an *iso-performance line*, *i.e.*, all classifiers corresponding to points on the line have the same expected cost. Each set of class and cost distributions defines a family of iso-performance lines. Lines "more northwest"—having a larger TP-intercept—are better because they correspond to classifiers with lower expected cost.

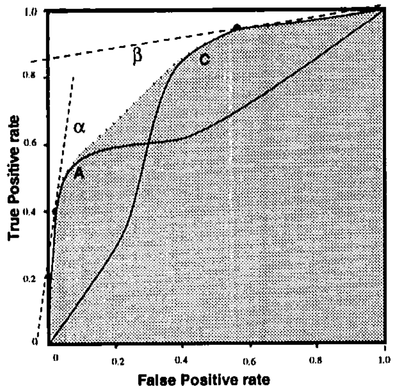

Figure 3: The ROC hull and two iso-performance lines

The ROC convex hull

Consider the ROC plots of a large set of classifiers under different conditions (parameter settings, training regimens, output threshold values, etc.). The optimal classifiers are those on the best iso-performance line defined by the target class and cost distributions. In most real-world cases the target distributions are not known precisely so it is valuable to be able to identify what subset of classifiers is potentially optimal under any conditions. Each set of conditions defines a family of iso-performance lines, and for a given family, the optimal methods are those that lie on the "most-northwest" iso-performance line. Thus, a classifier is potentially optimal if and only if it lies on the northwest boundary (*i.e.*, above the line y=x) of the convex hull of the set of points in ROC space. Space limitations prohibit a formal proof, but one can see that *if a point lies on the convex hull*, then there exists a line through that point such that no other line with the same slope (which defines a family of iso-performance lines) through any other point has a larger TP-intercept, and thus *the classifier represented by the point is optimal* under any distribution assumptions corresponding the that slope. *If a point does not lie on the convex hull*, then for any family of iso-performance lines there is another point that lies on an iso-performance line with the same slope but larger TP-intercept, and thus *the classifier can not be optimal.*

The convex hull of the set of points in ROC space will be called the *ROC convex hull* of the corresponding set of classifiers. Figure 2 shows the curves of Figure 1 with the ROC convex hull drawn (CH, the border between the shaded and unshaded areas). By the previous argument, neither B nor D is optimal under any circumstances, because none of the points of the ROC curve of either classifier lies on the convex hull. We can also remove from consideration any points of A and C that do not lie on the hull.

Consider these classifiers under two distribution scenarios. In each, negative examples outnumber positives by 10:1. In scenario $\mathcal{A}$, false positive and false negative errors have equal cost. In scenario $\mathcal{B}$, a false negative is 100 times as expensive as a false positive (*e.g.*, missing a case of fraud is much worse than a false alarm). Each scenario defines a family of iso-performance lines. The lines corresponding to scenario $\mathcal{A}$ have slope 10; those for $\mathcal{B}$ have slope $\frac{1}{10}$. Figure 3 shows the convex hull with two iso-performance lines, α and β, drawn on it. Line α is the "best" line with slope 10 that intersects the convex hull; line β is the best line with slope $\frac{1}{10}$ that intersects the convex hull. Each line identifies the optimal classifier under the given distribution.

Generating the ROC Convex Hull

We call the visualization and analysis of classifier performance based on the ROC convex hull and iso-performance lines the *ROC convex hull method.*

1. For each classifier, plot TP and FP in ROC space. For continuous-output classifiers, vary a threshold over the output range and plot the ROC curve.
2. Find the convex hull of the set of points representing the predictive behavior of all classifiers of interest. For n classifiers this can be done in $O(n \log(n))$ time by the QuickHull algorithm (Barber, Dobkin, & Huhdanpaa, 1993).
3. For each set of class and cost distributions of interest, find the slope (or range of slopes) of the corresponding iso-performance lines.
4. For each set of class and cost distributions, the optimal classifier will be the point on the convex hull that intersects the iso-performance line with largest TP-intercept (ranges of slopes specify hull segments).

Using the ROC Convex Hull

We are often faced with the task of managing and evaluating classifiers learned by a wide variety of different learning algorithms. We experiment with different algorithm parameters to determine the effect on classifier performance. For continuous-output classifiers we vary output threshold values. We also experiment with different training regimens; for example, should we use the distribution present in the sample at hand, or should we equalize class distributions for training?

The ROC Convex Hull can be used for managing the classifier set. Figures 2 and 3 demonstrate how the performance of multiple classifiers can be visualized under different cost and class distributions. We now demonstrate additional benefits of the method.

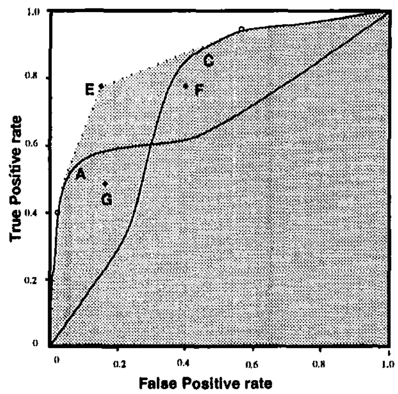

Figure 4: Classifiers E, F and G added

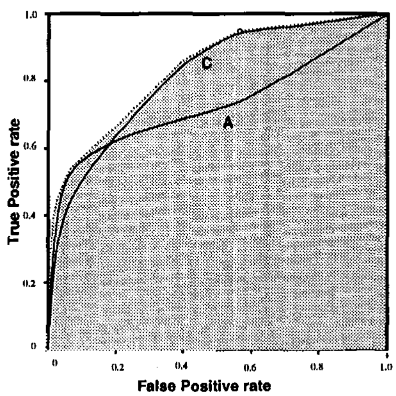

Figure 5: C nearly optimal through entire range

Managing a variety of classifiers

The ROC convex hull method accomodates both binary and continuous classifiers. Binary classifiers yield individual points in ROC space. Continuous classifiers produce continuous numeric outputs which can be thresholded, yielding a series of (FP, TP) pairs comprising an ROC curve. Each point may or may not contribute to the ROC convex hull. Figure 4 depicts the binary classifiers E, F and G added to the previous hull. E may be optimal under some circumstances because it contributes to the convex hull. Classifiers F and G will never be because they do not extend the hull.

New classifiers can be added incrementally to an ROC convex hull analysis, as demonstrated above with the addition of classifiers E,F, and G. Each new classifier either extends the existing hull or does not. In the former case the hull must be updated accordingly, but in the latter case the new classifer can be ignored. Therefore, the method does not require saving every classifier (or saving statistics on every classifier) for re-analysis under different conditions—only those points on the convex hull. No other classifiers can ever be optimal, so they need not be saved. For the purposes of comparison and sensitivity analysis, it is often useful to save the entire ROC curves for classifiers that contribute to the ROC convex hull. Figure 3 showed a situation in which no classifier is even close to optimal over the entire hull. By contrast, in Figure 5, while A and C are each optimal over a portion of the range, C is nearly optimal over the entire range.

Changing distributions and costs

Class and cost distributions that change over time necessitate the reevaluation of classifier choice. In fraud detection, costs change based on workforce and reimbursement issues; the amount of fraud changes monthly. With the ROC convex hull method, visualizing the effect of a new distribution involves only calculating the slope(s) of the corresponding iso-performance lines and intersecting them with the hull, as shown in Figure 3.

The ROC convex hull method scales gracefully to any degree of precision in specifying the cost and class distributions. If nothing is known about a distribution, the ROC convex hull shows all classifiers that may be optimal under any conditions. Figure 2 showed that, given classifiers A,B,C and D of Figure 1, only A and C can ever be optimal.

With complete information, the method identifies the optimal classifier(s). In Figure 3 we saw that classifier A (with a particular threshold value) is optimal under scenario $\mathcal{A}$ and classifier C is optimal under scenario $\mathcal{B}$. Next we will see that with less precise information, the ROC convex hull can show the set of possibly optimal classifiers.

Sensitivity analysis

Imprecise distribution information defines a *range* of slopes for iso-performance lines. This range of slopes intersects a segment of the ROC convex hull, which allows visual sensitivity analysis. For example, if the segment defined by a range of slopes corresponds to a single point in ROC space or a small threshold range for a single classifier, then there is no sensitivity to the distribution assumptions in question. Consider a scenario similar to $\mathcal{A}$ and $\mathcal{B}$ in that negative examples are 10 times as prevalent as positive ones. In this scenario, the cost of dealing with a false alarm is between \$5 and \$10, and the cost of missing a positive example is between \$500 and \$1000. This defines a range of slopes for iso-performance lines: $\frac{1}{20} \leq m \leq \frac{1}{5}$. Figure 6a depicts this range of slopes and the corresponding segment of the ROC convex hull. The figure shows that the choice of classifier is insensitive to changes within this range (and tuning of the classifier's threshold will be relatively small). Figure 6b depicts a scenario with a wider range of slopes: $\frac{1}{5} \leq m \leq 2$. The figure shows that under this scenario the choice of classifier is very sensitive to the distribution. Classifiers A,C, and E each are optimal for some subrange.

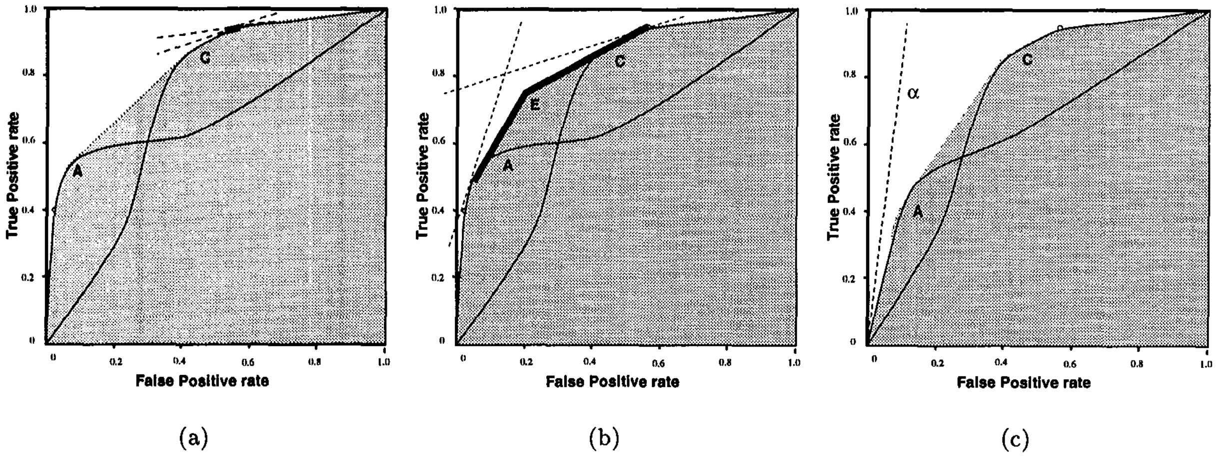

Figure 6: Sensitivity analysis using iso-performance lines. (a) low sensitivity, (b) high sensitivity, (c) when doing nothing is the best strategy

A particularly interesting question in any domain is, *When is doing nothing better than any of my available methods?* The ROC hull method gives a straightforward visualization of the answer. Consider Figure 6c. The point $(0,0)$ corresponds to doing nothing, *i.e.*, issuing negative classifications regardless of input. Any set of cost and class distribution assumptions for which the best hull-intersecting iso-performance line passes through the origin (*e.g.*, line α) defines a scenario where this null strategy is optimal. In the example of Figure 6c, the range of scenarios is small for which the null strategy is optimal; the slopes of the lines quantify the range.

Limitations and Implications

The ROC convex hull method enables analysis and visualization of classifier performance when class and cost distributions are not known precisely. However, the method has limitations with respect to universal use. In this paper, we have simplified by assuming there are only two classes and that costs do not vary within a given type of error. The first assumption is essential to the use of a two dimensional graph; the second assumption is essential to the creation of iso-performance lines. We have not investigated weakening these assumptions.

Furthermore, the method is based upon the maximization of expected value as the decision goal. Other decision goals are possible (Egan, 1975). For example, the *Neyman-Pearson observer* strategy tries to maximize the hit rate for a fixed false-alarm rate. In the ROC convex hull framework, a Neyman-Pearson observer would find the vertical line corresponding to the given FP rate, and intersect it with a "non-decreasing" hull, rather than the convex hull, (and move left horizontally, if possible).

The tradeoff between TP and FP rates is similar to the tradeoff between precision and recall, commonly used in Information Retrieval (Bloedorn, Mani, & MacMillan, 1996). However, precision and recall do not take into account the relative size of the population of "uninteresting" entities, which is necessary to deal with changing class distributions.

Existing cost-sensitive learners are brittle with respect to imprecise or changing distributions. Existing methods can be categorized into four categories: (i) the use of cost distribution in building a classifier, *e.g.*, for choosing splits in a decision tree (Breiman et al., 1984; Pazzani et al., 1994; Draper, Brodley, & Utgoff, 1994; Provost & Buchanan, 1992); (ii) the use of the cost distribution in post-processing the classifier, *e.g.*, for pruning a decision tree (Breiman et al., 1984; Pazzani et al., 1994; Draper et al., 1994), for finding rule subsets (Catlett, 1995; Provost & Buchanan, 1995), or for setting an output threshold; (iii) estimate the probability distribution and use decision-analytic combination (Pazzani et al., 1994; Catlett, 1995; Draper et al., 1994; Ezawa et al., 1996; Duda & Hart, 1973); and (iv) search for a bias with which a good classifier can be learned (Turney, 1995; Provost & Buchanan, 1995). Of these, only (iii) can handle changes in cost (or class) distribution without modifying the classifier. If any such method estimates the posterior probabilities perfectly, then it will give the optimal ROC curve under any class and cost distributions.

It is unlikely that a perfect method that is also practical will be found. Typically, no method dominates all others (Pazzani et al., 1994). Therefore robust methods for analysis and comparison of classifiers are needed. As future work, we propose a search similar to that taken by (iv) above, except that the goal is to find classifiers that extend the ROC convex hull. Thus, the resultant set of classifiers will be robust to imprecise and changing distributions.

Conclusion

The ROC convex hull method combines techniques from decision theory, ROC analysis and computational geometry, and adapts them to the analysis and visualization of classifier performance. The method has many desirable properties. It is graphical, intuitive, and provides a view of classifier performance both in general and under specific distribution assumptions. It is powerful enough to rule out classifiers over all possible scenarios. Due to its incremental nature, new classifiers can be incorporated easily, *e.g.*, when trying a new parameter setting, and it handles binary as well as continuous classifiers.

It has been noted many times that costs and class distributions are difficult to specify precisely. Classifier learning research should explore flexible systems that perform well under a range of conditions, perhaps for part of ROC space. We hope that our method for analysis of classifiers can help free researchers from the need to have precise class and cost distribution information.

References

Barber, C., Dobkin, D., & Huhdanpaa, H. (1993). The quickhull algorithm for convex hull. Tech. rep. GCG53, University of Minnesota. Available from `ftp://geom.umn.edu/pub/software/qhull.tar.Z`.

Beck, J. R., & Schultz, E. K. (1986). The use of ROC curves in test performance evaluation. *Arch Pathol Lab Med*, *110*, 13–20.

Bloedorn, E., Mani, I., & MacMillan, T. (1996). Machine learning of user profiles: Representational issues. In *Proceedings of Thirteenth National Conference on Artificial Intelligence*, pp. 433–438 Menlo Park, CA. AAAI Press.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees.* Wadsworth International Group, Belmont, CA.

Catlett, J. (1995). Tailoring rulesets to misclassificatioin costs. In *Proceedings of the 1995 Conference on AI and Statistics*, pp. 88–94.

Clearwater, S., & Stern, E. (1991). A rule-learning program in high energy physics event classification. *Comp Physics Comm*, *67*, 159–182.

Draper, B., Brodley, C., & Utgoff, P. (1994). Goal-directed classification using linear machine decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(9), 888–893.

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

Egan, J. P. (1975). *Signal Detection Theory and ROC Analysis.* Series in Cognitition and Perception. Academic Press, New York.

Ezawa, K., Singh, M., & Norton, S. (1996). Learning goal oriented bayesian networks for telecommunications risk management. In *Proceedings of IMLC-96*, pp. 139–147 San Francisco, CA. Morgan Kaufmann.

Fawcett, T., & Provost, F. (1996). Combining data mining and machine learning for effective user profiling. In *Proceedings of KDD-96*, pp. 8–13 Menlo Park, CA. AAAI Press.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. In *Proc. 11th International Conference on Machine Learning*, pp. 217–225. Morgan Kaufmann.

Provost, F., & Buchanan, B. (1992). Inductive policy. In *Proceedings of AAAI-92*, pp. 255–261. Menlo Park, CA: AAAI Press.

Provost, F., & Buchanan, B. (1995). Inductive policy: The pragmatics of bias selection. *Machine Learning*, *20*, 35–61.

Saitta, L., Giordana, A., & Neri, F. (1995). What is the "real-world"?. In *IMLC-95 Workshop on Applying Machine Learning in Practice (Tech Rept. AIC-95-023 Naval Research Lab)*, pp. 34–40.

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

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *JAIR*, *2*, 369–409.

Turney, P. (1996). Cost sensitive learning bibliography. `http://ai.iit.nrc.ca/bibliographies/cost-sensitive.html`.

Weinstein, M. C., & Fineberg, H. V. (1980). *Clinical Decision Analysis.* Philadelphi, PA: W. B. Saunders Company.