# Robust classification systems for imprecise environments

**Foster Provost**
Bell Atlantic Science and Technology
400 Westchester Avenue
White Plains, New York 10604
foster@basit.com

**Tom Fawcett**
Bell Atlantic Science and Technology
400 Westchester Avenue
White Plains, New York 10604
fawcett@basit.com

### Abstract

In real-world environments it is usually difficult to specify target operating conditions precisely. This uncertainty makes building robust classification systems problematic. We show that it is possible to build a hybrid classifier that will perform at least as well as the best available classifier for any target conditions. This robust performance extends across a wide variety of comparison frameworks, including the optimization of metrics such as accuracy, expected cost, lift, precision, recall, and workforce utilization. In some cases, the performance of the hybrid can actually surpass that of the best known classifier. The hybrid is also efficient to build, to store, and to update. Finally, we provide empirical evidence that a robust hybrid classifier is needed for many real-world problems.

## Introduction

Traditionally, classification systems have been built by experimenting with many different classifiers, comparing their performance and choosing the classifier that performs best. Unfortunately, this experimental comparison is often difficult in real-world environments because key parameters are not known. For example, the optimal cost/benefit tradeoffs and the target class distribution are seldom known precisely. This information is crucial to the choice of an optimal classifier.

We argue that it is possible and desirable to avoid committing to a single best classifier during system construction. Instead, a hybrid classification system can be built from the best available classifiers, and this hybrid will perform best under any target cost/benefit and class distributions. Target conditions can then be specified at run time. Moreover, in cases where precise information is still unavailable when the system is run (or if the conditions change dynamically during operation), the hybrid system can be tuned easily based on feedback from its actual performance.

The argument is structured as follows. First we sketch briefly the traditional approach to building such systems, in order to demonstrate that it is brittle under the types of imprecision common in real-world problems. In the subsequent sections we prove that the ROC Convex Hull, which is a method for comparing and visualizing classifier behavior in imprecise environments (Provost & Fawcett 1997), is also an elegant solution to the problem of building a robust classification system. The solution is elegant because the resulting hybrid classifier is robust for a wide variety of problem formulations, and it is efficient to build, to store, and to update. We then show that the hybrid can actually do better than the best known classifier in some situations. Finally, we provide empirical evidence that this type of system is needed for real-world problems.

## Brittle and robust classifiers

Consider a generic example. A systems-building team wants to create a system that will take a large number of instances and identify those for which an action should be taken. The instances could be potential cases of fraudulent account behavior, of faulty equipment, of responsive customers, of interesting science, etc. We consider problems for which the best method for classifying or ranking instances is not well-defined, so the system builders may consider AI methods such as expert systems, neural networks, learned decision trees and case-based systems as potential classification models. Ignoring for the moment issues of efficiency, the foremost question facing the system builders is: which of the available models performs "best" at classification?

Traditionally, an experimental approach has been taken to answer this question, because the distribution of instances is not known a priori, but usually can be sampled. The standard approach is to estimate the error rate of each model statistically and then to choose the model with the lowest error rate. This strategy is common in machine learning, pattern recognition, data mining, expert systems and medical diagnosis. In some cases, other measures

To appear in *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, 1998. 

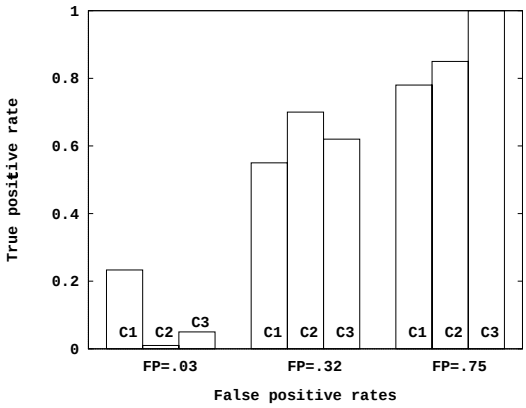


Figure 1: Three classifiers under three different Neyman-Pearson decision criteria

such as cost or benefit are used as well. Applied statistics provides methods such as cross-validation and the bootstrap for estimating model error rates and recent studies have compared the effectiveness of these different statistical methods (Salzberg 1997; Dietterich 1998).

Unfortunately, this experimental approach is brittle under two types of imprecision that are common in real-world environments. Specifically, costs and benefits usually are not known precisely, and class distributions often are known only approximately as well. This fact has been pointed out by many authors (Bradley 1997; Catlett 1995), and is in fact the concern of a large subfield of decision analysis (Weinstein & Fineberg 1980). Imprecision also arises because the environment may change between the time the system is conceived and the time it is used, and even as it is used. For example, levels of fraud and levels of customer responsiveness change continually over time and from place to place.

In this paper we address two-class problems. Formally, each instance $I$ is mapped to one element of the set $\{\mathbf{p}, \mathbf{n}\}$ of (correct) positive and negative classes. A *classification model* (or *classifier*) is a mapping from instances to predicted classes. To distinguish between the actual class and the predicted class of an instance, we will use the labels $\{\mathbf{Y}, \mathbf{N}\}$ for the classifications produced by a model. For our discussion, let $c(classification, class)$ be a two-place error cost function where $c(\mathbf{Y}, \mathbf{n})$ is the cost of a false positive error and $c(\mathbf{N}, \mathbf{p})$ is the cost of a false negative error. We represent class distributions by the classes' prior probabilities $p(\mathbf{p})$ and $p(\mathbf{n}) = 1 - p(\mathbf{p})$.

The traditional experimental approach is brittle because it chooses one model as "best" with respect to a specific set of cost functions and class distribution. If the target conditions change, this system may no longer perform optimally, or even acceptably. As an example, assume that we have a maximum false positive rate, $FP = p(\mathbf{Y}|\mathbf{n})$, that must not be exceeded. We want to find the classifier with the highest possible true positive rate, $TP = p(\mathbf{Y}|\mathbf{p})$, that does not exceed the $FP$ limit. This is the Neyman-Pearson decision criterion (Egan 1975). Three classifiers, under three such $FP$ limits, are shown in Figure 1. A different classifier is best for each $FP$ limit; any system built with a single "best" classifier is brittle if the $FP$ requirement can change.

We address this brittleness by extending the traditional comparison/selection framework to produce **robust classifiers**, defined as satisfying the following. *Under any target cost and class distributions, a robust classifier will perform at least as well as the best classifier for those conditions.* For this paper, statements about optimality are practical: the "best" classifier may not be the Bayes-optimal classifier, but it is better than all other known classifiers. Stating that a classifier is robust is stronger than stating that it is optimal. A robust classifier is optimal under all possible conditions.

Classification brittleness could be overcome by saving all possible classifiers (neural nets, decision trees, expert systems, probabilistic models, etc.) and then performing an automated run-time comparison under the desired target conditions. However, such a system is not feasible because of time and space limitations—there are myriad possible classification models, arising from the different data mining methods available under their many different parameter settings. Storing all the classifiers is not practical, and tuning the system by comparing classifiers on the fly under different conditions is not practical. Moreover, we will show that it is sometimes possible to do better than any of these classifiers.

## Hybrid classifiers using the ROC convex hull

We will show that robust hybrid classifiers can be built using the ROC Convex Hull (ROCCH).

**Definition 1** *Let* $\mathbf{I}$ *be the space of possible instances and let* $\mathbf{C}$ *be the space of sets of classification models. Let a* ***$\mu$-hybrid classifier*** *comprise a set of classification models* $\mathcal{C} \in \mathbf{C}$ *and a function*

$$\mu : \mathbf{I} \times \Re \times \mathbf{C} \rightarrow \{\mathbf{Y}, \mathbf{N}\}.$$

*A $\mu$-hybrid classifier takes as input an instance $I \in \mathbf{I}$ for classification and a number $x \in \Re$. As output, it produces the classification produced by $\mu(I, x, \mathcal{C})$.*

Things will get more involved later, but for the time being consider that each set of cost and class distributions defines a value for $x$, which is used to select the (predetermined) best classifier for those conditions. To build a $\mu$-hybrid classifier, we must define $\mu$ and the set $\mathcal{C}$. We would like $\mathcal{C}$ to include only those models that perform optimally under some conditions (class and cost distributions), since these will be stored by

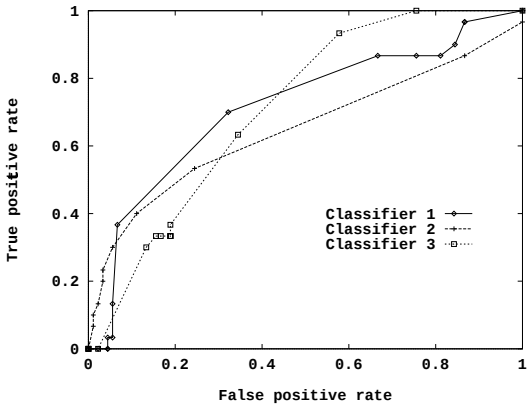


Figure 2: ROC graph of three classifiers

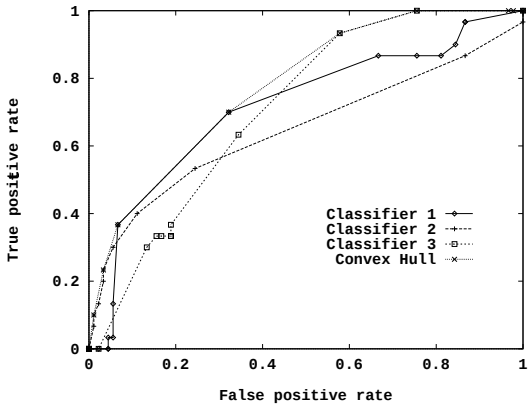


Figure 3: ROC curves with convex hull

the system, and we would like $\mu$ to be general enough to apply to a variety of problem formulations.

At this point, it is necessary to review briefly some of the basics of Receiver Operating Characteristic (ROC) analysis, a classic methodology from signal detection theory that is now common in medical diagnosis and has recently begun to be used more generally in AI classifier work (Swets 1988). *ROC space* denotes the coordinate system used for visualizing classifier performance. In ROC space, $TP$ is represented on the Y axis and $FP$ is represented on the X axis. Each classifier is represented by the point in ROC space corresponding to its $(FP, TP)$ pair. For models that produce a continuous output, e.g., an estimate of the posterior probability of an instance's class membership, these statistics vary together as a threshold on the output is varied between its extremes (each threshold value defines a classifier); the resulting curve is called the ROC curve. An ROC curve illustrates the error tradeoffs available with a given probabilistic model. Figure 2 shows a graph of three typical ROC curves; in fact, these are the complete ROC curves of the classifiers shown in Figure 1. ROC graphs illustrate the predictive behavior of a classifier *without regard to class distribution or error cost*, so they decouple classification performance from these factors.

As described in detail by Provost and Fawcett (1997) the ROCCH method takes as input a set of classifiers, along with their classification performance statistics, and plots them in ROC space. Then it finds the convex hull (Barber, Dobkin, & Huhdanpaa 1993) of the set of points in ROC space (the ROCCH). The convex hull of a set of points is the smallest convex set that contains the points. The ROCCH is the "northwest boundary" of the points in ROC space. The classifiers corresponding to the points comprising the ROCCH are the potentially optimal classifiers, because (roughly) points in ROC space that are more northwest are better. Figure 3 shows the three ROC curves with the convex hull drawn.

We claim that the models comprising the ROCCH can be combined to form an $\mu$-hybrid classifier that is an elegant, robust classifier.

**Definition 2** *The* ROCCH-hybrid *is a $\mu$-hybrid classifier where $\mathcal{C}$ is the set of classifiers that comprise the* ROCCH *and $\mu$ makes classifications using the classifier on the* ROCCH *with $FP = x$.*

Note that for the moment the ROCCH-hybrid is defined only for $FP$ values corresponding to ROCCH vertices.

## Robust classification

Our definition of robust classifiers was intentionally vague about what it means for one classifier to be better than another, because different situations call for different comparison frameworks. We now show that the ROCCH-hybrid is robust for a wide variety of comparison frameworks. We begin with minimizing expected cost, because the process of proving that the ROCCH-hybrid minimizes expected cost for any cost and class distributions provides a deep understanding of why the ROCCH-hybrid works. We then show that it is also robust for a variety of other practical metrics. After that we discuss how the ROCCH can be used in imprecise environments. Finally, we show that for some problems the ROCCH-hybrid can actually do better than all known classifiers.

### Minimizing expected cost

Decision analysis (Weinstein & Fineberg 1980) provides us with a method for determining when one classification model is better than another. Specifically, the expected cost of applying a classifier is:

$$p(\mathbf{p}) \cdot (1 - TP) \cdot c(\mathbf{N}, \mathbf{p}) \quad + \\ p(\mathbf{n}) \cdot FP \cdot c(\mathbf{Y}, \mathbf{n}) \qquad (1)$$

We now can show that the ROCCH-hybrid is robust for problems where the "best" classifier is the classifier with the minimum expected cost.

**Definition 3** *For minimizing expected cost, environmental conditions, viz., cost and class distributions, are translated to a single real value as follows:*

$$m_{mec} = \frac{c(\mathbf{Y},\mathbf{n})p(\mathbf{n})}{c(\mathbf{N},\mathbf{p})p(\mathbf{p})} \quad (2)$$

*This is the product of the cost ratio and the reciprocal of the class ratio. Every set of conditions will define an* $m_{mec} \geq 0$.

The slope of the ROCCH is an important tool in our argument. The ROCCH is a piecewise-linear, convex-down "curve." Therefore, as $x$ increases, the slope of the ROCCH is monotonically non-increasing with $k-1$ discrete values, where $k$ is the number of ROCCH component classifiers, including the degenerate classifiers that define the ROCCH endpoints. Where there will be no confusion, we use phrases such as "points in ROC space" as a shorthand for the more cumbersome "classifiers corresponding to points in ROC space." For this subsection, "points on the ROCCH" refer to vertices of the ROCCH.

**Definition 4** *For any real number* $m \geq 0$, ***the point where the slope of the*** ROCCH ***is*** $m$ *is one of the (arbitrarily chosen) endpoints of the segment of the* ROCCH *with slope* $m$*, if such a segment exists. Otherwise, it is the vertex for which the left adjacent segment has slope greater than* $m$ *and the right adjacent segment has slope less than* $m$.

For completeness, the leftmost endpoint of the ROCCH is considered to be attached to a segment with infinite slope and the rightmost endpoint of the ROCCH is considered to be attached to a segment with zero slope. Note that every $m \geq 0$ defines at least one point on the ROCCH.

**Lemma 1** *For any set of cost and class distributions, there is a point on the* ROCCH *with minimum expected cost.*
***Proof:*** *(by contradiction) Assume that for some conditions there exists a point* $\boldsymbol{C}$ *with smaller expected cost than any point on the* ROCCH*. By equations (1) and (2), a point* $(FP_2,TP_2)$ *has the same expected cost as a point* $(FP_1,TP_1)$*, if*

$$\frac{TP_2 - TP_1}{FP_2 - FP_1} = m_{mec}$$

*Therefore, for conditions corresponding to* $m_{mec}$*, all points with equal expected cost form a line, an* ***iso-performance line****, in ROC space with slope* $m_{mec}$*. Also by (1) and (2), points on lines with larger y-intercept have lower expected cost. Now, point* $\boldsymbol{C}$ *is not on the* ROCCH*, so it is either above the curve or below the curve. If it is above the curve, then the* ROCCH *is not a convex set enclosing all points, which is a contradiction. If it is below the curve, then the iso-performance line through* $\boldsymbol{C}$ *also contains a point* $\boldsymbol{P}$ *that is on the* ROCCH*. Since all points on an iso-performance line have the same expected cost, point* $\boldsymbol{C}$ *does not have smaller expected cost than all points on the* ROCCH*, which is also a contradiction.* □

Although it is not necessary for our purposes here, it can be shown that *all* of the minimum expected cost classifiers are *on* the ROCCH.

**Definition 5** *An iso-performance line with slope* $m$ *is an* ***m-iso-performance line.***

**Lemma 2** *For any cost and class distributions that translate to* $m_{mec}$*, a point on the* ROCCH *has minimum expected cost only if the slope of the* ROCCH *at that point is* $m_{mec}$.
***Proof:*** *(by contradiction) Suppose that there is a point* $\boldsymbol{D}$ *on the* ROCCH *where the slope is* not $m_{mec}$*, but the point does have minimum expected cost. By Definition 4, either (a) the segment to the left of* $\boldsymbol{D}$ *has slope less than* $m_{mec}$*, or (b) the segment to the right of* $\boldsymbol{D}$ *has slope greater than* $m_{mec}$*. For case (a), consider point* $\boldsymbol{N}$*, the vertex of the* ROCCH *that neighbors* $\boldsymbol{D}$ *to the left, and consider the (parallel)* $m_{mec}$*-iso-performance lines* $l_D$ *and* $l_N$ *through* $\boldsymbol{D}$ *and* $\boldsymbol{N}$*. Because* $\boldsymbol{N}$ *is to the left of* $\boldsymbol{D}$ *and the line connecting them has slope less than* $m_{mec}$*, the y-intercept of* $l_N$ *will be greater than the y-intercept of* $l_D$*. By the construction in the proof to Lemma 1, this means that* $\boldsymbol{N}$ *will have lower expected cost than* $\boldsymbol{D}$*, which is a contradiction. The argument for (b) is analogous (symmetric).* □

**Lemma 3** *If the slope of the* ROCCH *at a point is* $m_{mec}$*, then the point has minimum expected cost.*
***Proof:*** *If this point is the only point where the slope of the* ROCCH *is* $m_{mec}$*, then the proof follows directly from Lemma 1 and Lemma 2. If there are multiple such points, then by definition they are connected by an* $m_{mec}$*-iso-performance line, so they have the same expected cost, and once again the proof follows directly from Lemma 1 and Lemma 2.* □

It is straightforward now to show that the ROCCH-hybrid is robust for the problem of minimizing expected cost.

**Theorem 4** *The* ROCCH*-hybrid minimizes expected cost for any cost distribution and any class distribution.*
***Proof:*** *Because the* ROCCH*-hybrid is composed of the classifiers corresponding to the points on the* ROCCH*, this follows directly from Lemmas 1, 2, and 3.* □

## Optimizing other common metrics

The previous section showed that the ROCCH-hybrid is robust when the goal is to provide the minimum expected cost classification. The ROCCH-hybrid is robust for other common metrics as well. For example, even for accuracy maximization the preferred classifier may be different for different target class distributions. This is rarely taken into account in experimental comparisons of classifiers.

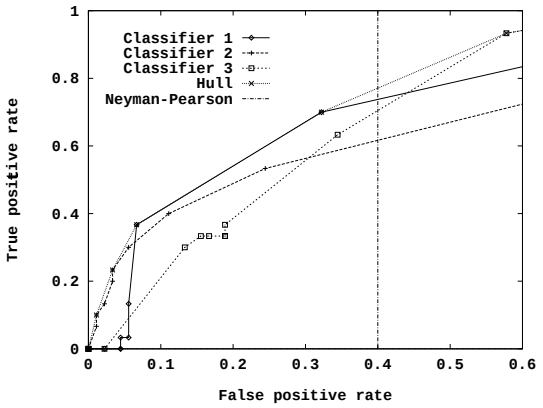


Figure 4: The ROC Convex Hull used to select a classifier under the Neyman-Pearson criterion

**Corollary 5** *The* ROCCH*-hybrid minimizes error rate (maximizes accuracy) for any target class distribution.*
***Proof:*** *Error rate minimization is cost minimization with uniform error costs.* □

An alternative metric, used often in ROC analysis to compare models, is the area under the ROC curve (AUC) (Bradley 1997). It is especially useful for situations where either the target cost distribution or class distribution is *completely* unknown. The AUC represents the probability that a randomly chosen positive instance will be rated higher than a negative instance, and thereby is also estimated by the Wilcoxon test of ranks (Hanley & McNeil 1982). A criticism of the use of AUC for model choice is that for specific target conditions, the classifier with the maximum AUC may be suboptimal (as we will see below). Fortunately, not only is the ROCCH-hybrid optimal for any specific target conditions, it has the maximum AUC.

**Theorem 6** *There is no classifier with AUC larger than that of the* ROCCH*-hybrid.*
***Proof:*** *(by contradiction) Assume the ROC curve for another classifier had larger area. This curve would have to have at least one point in ROC-space that falls outside the area enclosed by the* ROCCH*. This means that the convex hull does not enclose all points, which is a contradiction.* □

Recall that the Neyman-Pearson decision criterion specifies a maximum acceptable $FP$ rate. In standard ROC analysis, selecting the best classifier for the Neyman-Pearson criterion is easy: plot ROC curves, draw a vertical line at the desired maximum $FP$, pick ROC curve with the largest $TP$ at the intersection with this line.

For minimizing expected cost it was sufficient for the ROCCH-hybrid to choose a *vertex* from the ROCCH for any $m_{mec}$ value. For problem formulations such as the Neyman-Pearson criterion, the performance statistics at a non-vertex point on the ROCCH may be preferable (see Figure 4). Fortunately, with a slight extension, the ROCCH-hybrid can yield a classifier with these performance statistics.

**Theorem 7** *A* ROCCH*-hybrid can achieve the $TP$:$FP$ tradeoff represented by any point on the* ROCCH*, not just the vertices.*
***Proof:*** *(by construction) Extend $\mu(I,x,\mathcal{C})$ to non-vertex points as follows. Pick the point $P$ on the* ROCCH *with $FP = x$ (there is exactly one). Let $TP_x$ be the $TP$ value of this point. If $(x, TP_x)$ is a* ROCCH *vertex, use the corresponding classifier. If it is not a vertex, call the left endpoint of the hull segment $C_l$ and the right endpoint $C_r$. Let $d$ be the distance between $C_l$ and $C_r$, and let $p$ be the distance between $C_l$ and $P$. Make classifications as follows. For each input instance flip a weighted coin and choose the answer given by classifier $C_l$ with probability $\frac{p}{d}$ and that given by classifier $C_r$ with probability $1-\frac{p}{d}$. It is straightforward to show that $FP$ and $TP$ for this classifier will be $x$ and $TP_x$.* □

**Corollary 8** *For the Neyman-Pearson criterion, the performance of the* ROCCH*-hybrid is at least as good as that of any known classifier.*
***Proof:*** *Similar to minimum expected cost proofs—no classifier can be above the* ROCCH. □

There are still other realistic problem formulations. For example, consider the decision support problem of optimizing *workforce utilization*, in which a workforce is available to process a fixed number of cases. Too few cases will under-utilize the workforce, but too many cases will leave some cases unattended (expanding the workforce usually is not a short-term solution). If the workforce can handle $C$ cases, the system should present the best possible set of $C$ cases. This is similar to the Neyman-Pearson criterion, but with an absolute cutoff ($C$) instead of a percentage cutoff ($FP$).

**Theorem 9** *For workforce utilization, the* ROCCH*-hybrid will provide the best set of $C$ cases, for any choice of $C$.*
***Proof:*** *(by construction) The decision criterion is: maximize $TP$ subject to the constraint:*

$$TP \cdot P + FP \cdot N \leq C$$

*The optimal point is found by intersecting the constraint line with the* ROCCH*, because, similarly to the previous arguments, no classifier can be above the* ROCCH*. The rest follows from Theorem 7.* □

Similar arguments hold for many other comparison metrics. It can be shown that for maximizing lift (Berry & Linoff 1997), precision or recall, subject to absolute or percentage cutoffs on case presentation, the ROCCH-hybrid will provide the best set of cases.

As with minimizing expected cost, imprecision in the environment forces us to favor a robust solution for these other comparison frameworks. For many real-world problems, the precise desired cutoff will be unknown or will change (*e.g.*, because of fundamental uncertainty, variability in case difficulty or competing responsibiliities). The ROCCH-hybrid provides a robust

solution because it is the optimal solution for any cutoff. For example, for document retrieval the ROCCH-hybrid will yield the best $N$ documents for any $N$.

An apparent solution to the problem of robust classification is to use a system that ranks cases, rather than one that provides classifications, and just work down the ranked list of cases (the cutoff is implicit). However, for most practical situations, choosing the best ranking model is equivalent to choosing which classifier is best *for the cutoff that will be used.* Remember that ROC curves are formed from case rankings by moving the cutoff from one extreme to the other. For different cutoffs, implicit or explicit, different ranking functions perform better. This is exactly the problem of robust classification, and is solved elegantly by the ROCCH-hybrid—the ROCCH-hybrid comprises the set rankers that are best for all possible cutoffs. Formal arguments are beyond the scope of this paper, but as an example, consider two ranking functions $R_a$ and $R_b$. $R_a$ is perfect for the first 10 cases, and picks randomly thereafter. $R_b$ randomly chooses the first 10 cases, and ranks perfectly thereafter. $R_a$ is preferable for a cutoff of 10 cases and $R_b$ is preferable for much larger cutoffs.

## Using the ROCCH-hybrid

To use the ROCCH-hybrid for classification, we need to translate environmental conditions to $x$ values to plug into $\mu(I, x, \mathcal{C})$. For minimizing expected cost, equation 2 shows how to translate conditions to $m_{mec}$. For any $m_{mec}$, by Lemma 3 we want the $FP$ value of the point where the slope of the ROCCH is $m_{mec}$, which is straightforward to calculate. For the Neyman-Pearson criterion the conditions are defined as $FP$ values. For workforce utilization with conditions corresponding to a cutoff $C$, the $FP$ value is found by intersecting the line $TP \cdot P + FP \cdot N = C$ with the ROCCH.

We have argued that target conditions (misclassification costs and class distribution) are rarely known. It may be confusing that we now seem to require exact knowledge of these conditions. The ROCCH-hybrid gives us two important capabilities. First, the need for precise knowledge of target conditions is deferred until runtime. Second, in the absence of precise knowledge even at run-time, the system can be optimized easily with minimal feedback.

By using the ROCCH-hybrid, information on target conditions is not needed to train and compare classifiers. This is important because of temporal (or geographic) differences that may exist between training and use. Building a system for a real-world problem introduces a non-trivial delay between the time data are gathered and the time the learned models will be used. The problem is exacerbated in domains where error costs or class distributions change over time; even with slow drift, a brittle model may become suboptimal quickly. In these scenarios, costs and class distributions can be specified (or respecified) at run time with reasonable precision by sampling from the current population, and used to ensure that the ROCCH-hybrid always performs optimally.

In some cases, even at run time these quantities are not known exactly. A further benefit of the ROCCH-hybrid is that it can be tuned easily to yield optimal performance with only minimal feedback from the environment. Conceptually, the ROCCH-hybrid has one "knob" that varies $x$ in $\mu(I, x, \mathcal{C})$ from one extreme to the other. For any knob setting, the ROCCH-hybrid will give the optimal $TP$:$FP$ tradeoff for the target conditions corresponding to that setting. Turning the knob to the right increases $TP$; turning the knob to the left decreases $FP$. Because of the monotonicity of the ROCCH-hybrid, simple hill-climbing can guarantee optimal performance. For example, if the system produces too many false alarms, turn the knob to the left; if the system is presenting too few cases, turn the knob to the right.

## Beating the component classifiers

Perhaps surprisingly, in many realistic situations a ROCCH-hybrid system can do *better* than any of its component classifiers. Consider the Neyman-Pearson decision criterion. The ROCCH may intersect the $FP$-line *above* the highest component ROC curve. This occurs when the $FP$-line intersects the ROCCH between vertices; therefore, there is no component classifier that actually produces these particular ($FP$,$TP$) statistics (as in Figure 4).

**Theorem 10** *The* ROCCH*-hybrid can surpass the performance of its component classifiers for some Neyman-Pearson problems.*
***Proof:*** *For any non-vertex hull point ($x$,$TP_x$), $TP_x$ is larger than the $TP$ for any other point with $FP = x$. By Theorem 7, the* ROCCH*-hybrid can achieve any $TP$ on the hull. Only a small number of $FP$ values correspond to hull vertices.* □

The same holds for other problem formulations, such as workforce utilization, lift maximization, precision maximization, and recall maximization.

## Time and space efficiency

We have argued that the ROCCH-hybrid is robust for a wide variety of problem formulations. It is also efficient to build, to store, and to update.

The time efficiency of building the ROCCH-hybrid depends first on the efficiency of building the component models, which varies widely by model type. Some models built by machine learning methods can be built in seconds (once data are available). Hand-built models can take years to build. However, we presume that this is work that would be done anyway. The ROCCH-hybrid can be built with whatever methods are available, be there two or two thousand; as described below, as new classifiers become available, the ROCCH-hybrid can be updated incrementally. The time efficiency depends

also on the efficiency of the experimental evaluation of the classifiers. Once again, we presume that this is work that would be done anyway (more on this in Limitations). Finally, the time efficiency of the ROCCH-hybrid depends on the efficiency of building the ROCCH, which can be done in $O(N \log N)$ time using the Quick-Hull algorithm (Barber, Dobkin, & Huhdanpaa 1993) where $N$ is the number of classifiers.

The ROCCH is space efficient, too, because it comprises only classifiers that might be optimal under some target conditions.

**Theorem 11** *For minimizing expected cost, the* ROCCH*-hybrid comprises only classifiers that are optimal under some cost and class distributions.*
***Proof:*** *Follows directly from Lemmas 1–3 and Definitions 3 and 4.* □

The number of classifiers that must be stored can be reduced if bounds can be placed on the potential target conditions. As described by Provost and Fawcett (1997), ranges of conditions define segments of the ROCCH. Thus, the ROCCH-hybrid may need only a subset of $\mathcal{C}$.

Adding new classifiers to the ROCCH-hybrid is also efficient. Adding a classifier to the ROCCH will either (i) extend the hull, adding to (and possibly subtracting from) the ROCCH-hybrid, or (ii) conclude that the new classifiers are not superior to the existing classifiers in any portion of ROC space and can be discarded.

The run-time (classification) complexity of the ROCCH-hybrid is never worse than that of the component classifiers. In situations where run-time complexity is crucial, the ROCCH should be constructed without prohibitively expensive classification models. It will then find the best subset of the computationally efficient models.

## Empirical demonstration of need

Robust classification is of theoretical interest because it concentrates on weakening two very strong assumptions. We have argued that the assumptions of precise knowledge of cost and class distributions are too strong, citing evidence from AI work as well as a subfield of decision analysis dedicated to quantifying costs and benefits. However, might it not be that existing classifiers are already robust? For example, if a given classifier is optimal under one set of conditions, might it not be optimal under all?

It is beyond the scope of this paper to offer an in-depth experimental study of this question, but we can still provide solid evidence that the answer is "no." To this end, we refer to a comprehensive ROC analysis of medical domains recently conducted by Bradley (1997). His purpose was not to answer this question; fortunately his published results do anyway.

A classifier *dominates* if its ROC curve completely defines the ROCCH (which means dominating classifiers are robust and vice versa). If there exist more than a trivially few domains where no classifier dominates, then techniques like the ROCCH-hybrid are essential. Bradley studied six medical data sets, noting that "unfortunately, we rarely know what the individual misclassification costs are." He plotted the ROC curves of six classifier learning algorithms (two neural nets, two decision trees and two statistical techniques).

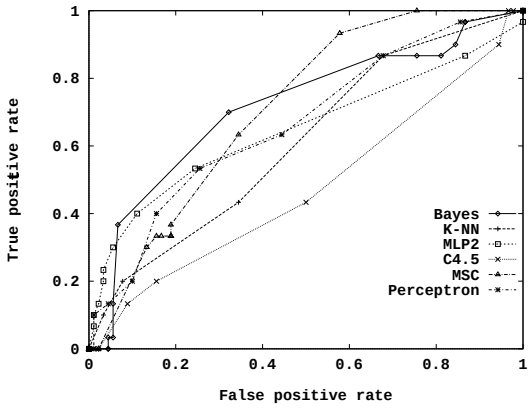


Figure 5: Bradley's classifier results for the heart bleeding data.

On *not one* of these data sets was there a dominating classifier. This means that for each domain, there exist different sets of conditions for which different classifiers are preferable. In fact, the three example classifiers used in this paper are the three best classifiers from Bradley's results on the heart bleeding data; his results for the full set of six classifiers can be found in Figure 5. Classifiers constructed for the Cleveland heart disease data, are shown in Figure 6.

Bradley's results show clearly that for many domains the classifier that maximizes any single metric—be it accuracy, cost, or the area under the ROC curve—will be the best for some cost and class distributions and will not be the best for others.[1] We have shown that the ROCCH-hybrid will be the best for all.

## Limitations

There are limitations to the ROCCH-hybrid as we have presented it here. We have defined it only for two-class problems. We believe that it can be extended to multi-class problems, but have not done so. It should be noted that the dimensionality of the "ROC-hyperspace" grows quadratically in the number of classes. We have also assumed constant error costs for a given *type* of error, e.g., all false positives cost the same. For some problems, different errors of the same type have different costs. In many cases, such a problem can be transformed into an equivalent problem with uniform intra-type error costs by duplicating

[1] More recently, an independent study showed a dominating classifier for only one of ten standard machine learning benchmarks (Provost, Fawcett, & Kohavi 1998).

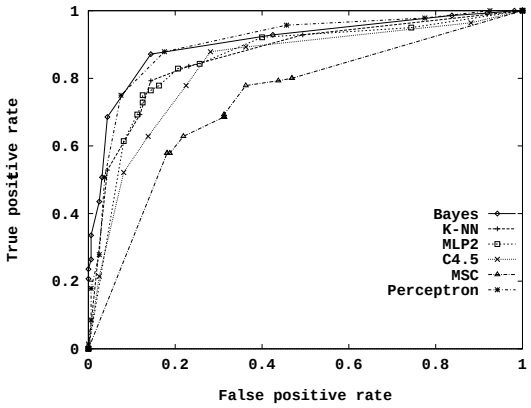


Figure 6: Bradley's classifier results for the Cleveland heart disease data

instances in proportion to their costs for evaluation.

We have also assumed for this paper that the estimates of the classifiers' performance statistics ($FP$ and $TP$) are very good. As mentioned above, much work has addressed the production of good estimates for simple performance statistics such as error rate. Much less work has addressed the production of good ROC curve estimates. It should be noted that, as with simpler statistics, care should be taken to avoid overfitting the training data and to ensure that differences between ROC curves are meaningful. Cross-validation with averaging of ROC curves is one possible solution (Provost, Fawcett, & Kohavi 1998).

Finally, we have addressed predictive performance and computational performance. These are not the only concerns in choosing a classification model. What if comprehensibility is important? The easy answer is that for any particular setting, the ROCCH-hybrid is as comprehensible as the underlying model it is using. However, this answer falls short if the ROCCH-hybrid is interpolating between two models or if one wants to understand the "multiple-model" system as a whole.

## Conclusion

The ROCCH-hybrid performs optimally under any target conditions for many realistic problem formulations. It is efficient to build in terms of time and space, and can be updated incrementally. Therefore, we conclude that it is an elegant, robust classification system.

The motivation for this work is fundamentally different from recent machine learning work on combining multiple models (Ali & Pazzani 1996). That work combines models in order to boost performance for a fixed cost and class distribution. The ROCCH-hybrid combines models for robustness across different cost and class distributions. These methods should be independent—multiple-model classifiers are candidates for extending the ROCCH. However, it may be that some multiple-model classifiers achieve increased performance for a specific set of conditions by (effectively) interpolating along edges of the ROCCH.

## Acknowledgements

We thank the many with whom we have discussed ROC analysis and classifier comparison, especially Rob Holte, George John, Ron Kohavi, Ron Rymon, and Peter Turney. We thank Andrew Bradley for supplying data from his analysis.

## References

Ali, K. M., and Pazzani, M. J. 1996. Error reduction through learning multiple descriptions. *Machine Learning* 24(3):173–202.

Barber, C.; Dobkin, D.; and Huhdanpaa, H. 1993. The quickhull algorithm for convex hull. Technical Report GCG53, University of Minnesota. Available from `ftp://geom.umn.edu/pub/software/qhull.tar.Z`.

Berry, M. J. A., and Linoff, G. 1997. *Data Mining Techniques: For Marketing, Sales, and Customer Support.* John Wiley & Sons.

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.

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

Dietterich, T. G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation.* to appear.

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

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

Provost, F., and 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 (KDD-97)*, 43–48. AAAI Press.

Provost, F.; Fawcett, T.; and Kohavi, R. 1998. Building the case against accuracy estimation for comparing induction algorithms. Submitted to IMLC–98. AAAI Press. Available from `http://www.croftj.net/~fawcett/papers/ICML98-submitted.ps.gz`.

Salzberg, S. L. 1997. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery* 1:317–328.

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

Weinstein, M. C., and Fineberg, H. V. 1980. *Clinical Decision Analysis.* Philadelphia, PA: W. B. Saunders Company.