Department of Computer Science,
School of Engineering and Computer Science and Mathematics,
University of Exeter. EX4 4QF. UK.
http://www.ex.ac.uk/secsm

UNIVERSITY of EXETER

Multi-class ROC analysis from a multi-objective optimisation perspective

Richard M. Everson and Jonathan E. Fieldsend
Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK
{R.M.Everson,J.E.Fieldsend}@exeter.ac.uk

5th April 2005

Abstract

The Receiver Operating Characteristic (ROC) has become a standard tool for the analysis and comparision of classifiers when the costs of misclassification are unknown. There has been relatively little work, however, examining ROC for more than two classes. Here we discuss and present a number of different extensions to the standard two-class ROC for multi-class problems.

We define the ROC surface for the Q-class problem in terms of a multi-objective optimisation problem in which the goal is to simultaneously minimise the $Q(Q-1)$ misclassification rates, when the misclassification costs and parameters governing the classifier's behaviour are unknown. We present an evolutionary algorithm to locate the Pareto front—the optimal trade-off surface between misclassifications of different types. The performance of the evolutionary algorithm is illustrated on a synthetic three class problem, for both k-nearest neighbour and multi-layer perceptron classifiers. Neuroscale is used to visualise the 5-dimensional front in two or three dimensions.

The use of the Pareto optimal surface to compare classifiers is discussed, together with Hand & Till's [2001] M measure of total class separability. We present a straightforward multi-class analogue of the Gini index. Also, we develop an evolutionary algorithm for the maximisation of M for the situation in which the parameters of the classifier can be varied. This is illustrated on various standard machine learning data sets.

1 Introduction

Classification or discrimination of unknown exemplars into two or more classes based on a 'training' dataset of examples, whose classification is known, is one of the fundamental problems in supervised pattern recognition. Given a classifier that yields estimates of the exemplar's probability of belonging to each of the classes and when the relative relative costs of misclassification are known, it is straightforward to determine the decision rule that minimises the average cost of misclassification. If the cost of misclassification is taken to be 1 and there is no penalty for a correct classification then the optimal rule becomes: assign to the class with the highest posterior probability. In practical situations, however, the true costs of misclassification are unequal and frequently unknown or difficult to determine [e.g. Bradley, 1997; Adams and Hand, 1999]. In such cases the practitioner must either guess the misclassification costs or explore the trade-off in classification rates as the decision rule is varied.

Receiver Operating Characteristic (ROC) analysis provides a convenient graphical display of the trade-off between true and false positive classification rates for two class problems [Provost and Fawcett, 1997]. Since its introduction in the medical and signal processing literatures [Hanley and McNeil, 1982; Zweig and Campbell, 1993] ROC analysis has become a prominent method for selecting an operating point; see [Flach et al., 2003] and [Hernández-Orallo et al., 2004] for a recent snapshot of methodologies and applications.

In this paper we extend the spirit of ROC analysis to multi-class problems by considering the trade-offs between the misclassification rates from one class into each of the other classes. Rather than considering the true and false positive rates, we consider the multi-class ROC surface to be the solution of the multi-objective optimisation problem in which these misclassification rates are simultaneously optimised. Srinivasan [1999] has discussed a similar formulation of multi-class ROC, showing that if classifiers for Q classes are considered to be points with coordinates given by their $Q(Q-1)$ misclassification rates, then optimal classifiers lie on the convex hull of these points. Here we describe the surface in terms of Pareto optimality and in section 3 we give an evolutionary algorithm for locating the optimal ROC surface when the classifier's parameters may be adjusted as part of the optimisation. Since multi-class ROC surfaces live in $Q(Q-1)$ dimensions visualisation is problematic, even for $Q = 3$; in section 4 we therefore consider visualisation methods for ROC surfaces for a probabilistic k-nn classifier [Holmes and Adams, 2002] and a multi-layer perceptron classifying synthetic data.

ROC analysis is frequently used for evaluating and comparing classifiers, the area under the ROC curve (AUC) or, equivalently, the Gini index. Although the straightforward analogue of the AUC is unsuitable for more than two classes, in section 5 we develop a straightforward generalisation of the Gini index which quantifies the superiority of a classifier's performance to random allocation. Hand and Till [2001] have presented an index of a classifier's performance based on the area under the 'ROC curves' for each pair of classes averaged over all pairs. In section 5 we also describe a procedure for maximising this measure over a parameterised family of classifiers; the procedure is illustrated on standard machine learning datasets.

2 ROC Analysis

Here we describe the straightforward extension of ROC analysis to more than two classes (multi-class ROC) and draw some comparisons with the two class case.

In general a classifier seeks to allocate an exemplar or measurement $\mathbf{x}$ to one of a number of classes. Allocation of $\mathbf{x}$ to the incorrect class, say $\mathcal{C}_j$, usually incurs some, often unknown, cost denoted by λ_{kj}; we count cost a correct classification as zero: $\lambda_{kk} = 0$. Denoting the probability of assigning an exemplar to $\mathcal{C}_j$ when its true class is in fact $\mathcal{C}_k$ as $p(\mathcal{C}_j \,|\, \mathcal{C}_k)$ the overall risk or expected cost is

$$R = \sum_{j,k} \lambda_{jk} p(\mathcal{C}_j \,|\, \mathcal{C}_k) \pi_k \tag{1}$$

where π_k is the prior probability of $\mathcal{C}_k$. The performance of some particular classifier may be conveniently be summarised by a confusion matrix or contingency table, $\hat{C}$, which summarises the results of classifying a set of examples. Each entry $\hat{C}_{kj}$ of the confusion matrix gives the number of examples, whose true class was $\mathcal{C}_k$, that were actually assigned to $\mathcal{C}_j$. Normalising the confusion matrix so that each column sums to unity gives the confusion rate matrix, which we denote by C, whose entries are estimates of the misclassification probabilities: $p(\mathcal{C}_j \,|\, \mathcal{C}_k) \approx C_{kj}$. Thus the expected risk is estimated as

$$R = \sum_{j,k} \lambda_{jk} C_{kj} \pi_k. \tag{2}$$

A slightly different perspective is gained by writing expected risk in terms of the posterior probabilities of classification to each class. The conditional risk or average cost of assigning $\mathbf{x}$ to $\mathcal{C}_j$ is

$$R(\mathcal{C}_j \,|\, \mathbf{x}) = \sum_{k} \lambda_{jk} p(\mathcal{C}_k \,|\, \mathbf{x}) \tag{3}$$

where $p(\mathcal{C}_k \,|\, \mathbf{x})$ is the posterior probability that $\mathbf{x}$ belongs to $\mathcal{C}_k$. The expected overall risk is

$$R = \int R(\mathcal{C}_j \,|\, \mathbf{x}) p(\mathbf{x}) \, d\mathbf{x}. \tag{4}$$

The expected risk is then minimised, being equal to the Bayes risk, by assigning $\mathbf{x}$ to the class with the minimum conditional risk [e.g. Duda and Hart, 1973]. Choosing 'zero-one costs', $\lambda_{jk} = 1 - \delta_{jk}$, means that all misclassifications are equally costly and the conditional risk is equal to the class posterior probability; one thus assigns to the class with the greatest posterior probability, which minimises the overall error rate.

When the costs are known it is therefore straightforward make assignments to achieve the Bayes risk (provided, of course, that the classifier yields accurate assessments of the posterior probabilities $p(\mathcal{C}_k \,|\, \mathbf{x})$). However, costs are frequently unknown and difficult to estimate, particularly when there are many classes; in this case it is useful to be able to compare the classification rates as the costs vary. For binary classification the conditional risk may be simply rewritten in terms of the posterior probability of assigning to $\mathcal{C}_1$, resulting in the rule: assign $\mathbf{x}$ to $\mathcal{C}_1$ if $P(\mathcal{C}_1 \,|\, \mathbf{x}) > t = \lambda_{12}/(\lambda_{12} + \lambda_{22})$. This classification rule reveals that there is, in fact, only one degree of freedom in the binary cost matrix and, as might be expected, the entire range of classification rates for each class can be swept out as the classification threshold t varies from 0 to 1. It is this variation of rates that the ROC curve exposes for binary classifiers. ROC analysis focuses on the classification of one particular class, say $\mathcal{C}_1$, and plots the true positive classification rate for $\mathcal{C}_1$ versus the false positive rate as the threshold t or, equivalently, the ratio of misclassification costs is varied.

If more than one classifier is available (often produced by altering the parameters, $\mathbf{w}$, of a particular classifier) then it can be shown that the convex hull of the ROC curves for the individual classifiers is the locus of optimum performance for that set of classifiers. [Provost and Fawcett, 1997, 1998] and Scott et al. [1998] have shown that performance at any point on the convex hull can be obtained by stochastically combining classifiers at the vertices of the convex hull.

Frequently in two class problems the focus is on a single class, for example, whether a set of medical symptoms are to be classified as benign or dangerous, so the ROC analysis practice of plotting of true and false positive rates for a single class is helpful. Also, since there is are only three degrees of freedom in the binary confusion matrix, classification rates for the other class are easily inferred. Indeed, the confusion *rate* matrix, C has only two degrees of freedom for binary problems. Focusing on one particular class is likely to be misleading when more than two classes are available for assignment. We therefore concentrate on the misclassification rates of each class to the others. In terms of the confusion rate matrix C we consider the off-diagonal elements, the diagonal elements (i.e., the true positives) being determined by the off-diagonal elements since each column sums to unity.

With Q classes there are $D = Q(Q-1)$ degrees of freedom in the confusion rate matrix and it is desirable to simultaneously minimise all the misclassification rates represented by these. For most problems, as in the binary problem, simultaneous optimisation will clearly be impossible and some compromise between the various misclassification rates will have to be found. Knowledge of the costs makes this determination simple, but if the costs are unknown we propose to use multi-objective optimisation to discover the optimal trade-offs between the misclassifications rates.

Since the units in which costs are measured are immaterial, the costs may, without loss of generality, be taken as summing to unity. We assume here, that there is zero cost for correct assignment, $\lambda_{ii} = 0$, so there are $Q(Q-1) - 1 = D - 1$ degrees of freedom for the specification of costs. Consequently, the optimal trade-off surface is, in general, of dimension $D - 1$, one fewer than the dimension of the ambient space. By considering the trivial classifiers that place all the misclassification cost on a single class $\mathcal{C}_k$ and none on any of the others it is clear that the ROC surface can be extended to each of the D corners of the hypercube with coordinates $(1, \ldots, 1, 0, 1, \ldots, 1)$ where the zero occurs in the kth position. In a similar manner to the ROC curve for binary problems which is parameterised by the ratio of misclassification costs, the ROC surface may thus be thought of as a $D - 1$ dimensional surface dividing the origin of the $[0, 1]^D$ hypercube from the $(1, \ldots, 1)$ corner and locally parameterised by the ratios of misclassification costs. To make these ideas more precise we now define the optimal ROC surface in terms of a Pareto front.

In general we will consider locating the optimal ROC surface as a function of the classifier parameters, $\mathbf{w}$, as well as the costs. For notational convenience and because they are treated as a single entity, we write the cost matrix $\boldsymbol{\lambda}$ and parameters as a single vector of generalised parameters, $\boldsymbol{\theta} = \{\boldsymbol{\lambda}, \mathbf{w}\}$; to distinguish $\boldsymbol{\theta}$ from the classifier parameters $\mathbf{w}$ we use the optimisation terminology *decision vectors* to refer to $\boldsymbol{\theta}$. We consider the D misclassification rates to be functions (depending on the particular classifier) of the decision vectors, thus $C_{jk} = C_{jk}(\boldsymbol{\theta})$. The optimal trade-off between the

Algorithm 1 Multi-objective evolution scheme for ROC surfaces.

Inputs:
T — *Number of generations*
N_λ — *Number of costs to sample*

$E :=$ initialise()
`for` $t := 1 : T$
  $\{\mathbf{w}, \boldsymbol{\lambda}\} = \boldsymbol{\theta} :=$ select(E) — *PQRS*
  $\mathbf{w}' :=$ perturb($\mathbf{w}$) — *Perturb parameters*
  `for` $i := 1 : N_\lambda$
    $\boldsymbol{\lambda}' :=$ sample() — *Sample costs*
    $C :=$ classify($\mathbf{w}', \boldsymbol{\lambda}'$) — *Evaluate classification rates*
    $\boldsymbol{\theta}' := \{\mathbf{w}', \boldsymbol{\lambda}'\}$
    `if` $\boldsymbol{\theta}' \not\preceq \boldsymbol{\phi} \; \forall \boldsymbol{\phi} \in E$
      $E := \{\boldsymbol{\phi} \in E \,|\, \boldsymbol{\phi} \not\prec \boldsymbol{\theta}'\}$ — *Remove dominated elements*
      $E := E \cup \boldsymbol{\theta}'$ — *Insert* $\boldsymbol{\theta}'$
    `end`
  `end`
`end`

misclassification rates is thus the defined by the minimisation problem:

$$\text{minimise} \quad C_{jk}(\boldsymbol{\theta}) \quad \text{for all } j, k. \tag{5}$$

If the all misclassification rates for one classifier with decision vector $\boldsymbol{\theta}$ are no worse than the classification rates for another classifier $\boldsymbol{\phi}$ and at least one rate is better, then the classifier parameterised by $\boldsymbol{\theta}$ is said to *strictly dominate* that parameterised by $\boldsymbol{\phi}$. Thus $\boldsymbol{\theta}$ strictly dominates $\boldsymbol{\phi}$ (denoted $\boldsymbol{\theta} \prec \boldsymbol{\phi}$) iff:

$$\begin{aligned} C_{jk}(\boldsymbol{\theta}) &\le C_{jk}(\boldsymbol{\phi}) \quad \forall j,k \quad \text{and} \\ C_{jk}(\boldsymbol{\theta}) &< C_{jk}(\boldsymbol{\phi}) \quad \text{for some} \;\; j,k. \end{aligned} \tag{6}$$

Less stringently, $\boldsymbol{\theta}$ *weakly dominates* $\boldsymbol{\phi}$ (denoted $\boldsymbol{\theta} \preceq \boldsymbol{\phi}$) iff

$$C_{jk}(\boldsymbol{\theta}) \le C_{jk}(\boldsymbol{\phi}) \quad \forall j, k. \tag{7}$$

A set A of decision vectors is said to be *non-dominated* if no member of the set is dominated by any other member:

$$\boldsymbol{\theta} \not\prec \boldsymbol{\phi} \quad \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in A. \tag{8}$$

A solution to the minimisation problem (5) is thus *Pareto optimal* if it is not dominated by any other feasible solution, and the non-dominated set of all Pareto optimal solutions is known as the Pareto front. Recent years have seen the development of a number of evolutionary techniques based on dominance measures for locating the Pareto front; see Coello Coello [1999]; Deb [2001] and Veldhuizen and Lamont [2000] for recent reviews. Kupinski and Anastasio [1999] and Anastasio et al. [1998] introduced the use of multi-objective evolutionary algorithms (MOEAs) to optimise ROC curves for binary problems, illustrating the method on a synthetic data set and for medical imaging problems; and we have used a similar methodology for locating optimal ROC curves for safety-related systems [Fieldsend and Everson, 2004; Everson and Fieldsend, 2005]. In the following section we describe a straightforward evolutionary algorithm for locating the Pareto front for multi-class problems. We illustrate the method on a synthetic problem for two different classification models in 4.

3 Locating multi-class ROC surfaces

Here we describe a straightforward algorithm for locating the Pareto front for multi-class ROC problems using an analogue of mutation-based evolution. The procedure is based on the Pareto

Archived Evolutionary Strategy (PAES) introduced by Knowles and Corne [2000]. In outline, the algorithm maintains a set or archive E, whose members are mutually non-dominating, which forms the current approximation to the Pareto front. As the computation progresses members of E are selected, copied and their decision vectors perturbed, and the objectives corresponding to the perturbed decision vector evaluated; if the perturbed solution is not dominated by any element of E, it is inserted into E and any members of E which are dominated by the new entrant are removed. It is clear, therefore, that the archive can only move towards the Pareto front: it is in essence a greedy search where the archive E is the current point of the search and perturbations to E that are not dominated by the current E are always accepted.

Algorithm 1 describes the procedure in more detail. The archive E is initialised by evaluating the misclassification rates for a number (here 100) of randomly chosen parameter values and costs, and discarding those which are dominated by another element of the initial set. Then at each generation a single element, $\boldsymbol{\theta}$ is selected from E (line 3 of Algorithm 1); selection may be uniformly random, but partitioned quasi-random selection (PQRS) [Fieldsend et al., 2003] was used here to promote exploration of the front. PQRS prevents clustering of solutions in a particular region of the front biasing the search because they are selected more frequently, thus increasing the efficiency and range of the search.

The selected *parent* decision vector is copied, after which the costs $\boldsymbol{\lambda}$ and classifier parameters $\mathbf{w}$ are treated separately. The parameters $\mathbf{w}$ of the classifier are perturbed or, in the nomenclature of evolutionary algorithms, mutated to form a *child*, $\mathbf{w}'$ (line 4). Here we seek to encourage wide exploration of parameter space by perturbing with a random number δ drawn from a heavy tailed distribution (such as the Laplacian density, $p(\delta) \propto e^{-|\delta|}$) to each of the parameters. The Laplacian distribution has tails that decay relatively slowly, thus ensuring that there is a high probability of exploring regions distant from the current solutions, facilitating escape from local minima [Yao et al., 1999].

With a proposed parameter set $\mathbf{w}'$ on hand the procedure then investigates the misclassification rates as the costs are varied with fixed parameters. In order to do this we generate N_λ sample costs, $\boldsymbol{\lambda}'$, and evaluate the misclassification rates for each of them. Since the misclassification costs are non-negative and sum to unity, a straightforward way of producing samples is to make a draws from a Dirichlet distribution:

$$p(\boldsymbol{\lambda}) = Dir(\boldsymbol{\lambda} \,|\, \alpha_1, \dots, \alpha_D) \tag{9}$$

$$= \frac{\Gamma(\sum_{i=1}^{D} \alpha_i)}{\prod_{i=1}^{D} \Gamma(\alpha_i)} \left(1 - \sum_{i=1}^{D-1} \lambda_i\right)^{\alpha_D - 1} \prod_{i=1}^{D-1} \lambda_i^{\alpha_i - 1} \tag{10}$$

where the index i labels the $D = Q(Q-1)$ off-diagonal entries in the cost matrix. As figure 1 illustrates, samples from a Dirichlet density lie on the simplex $\sum_{jk} \lambda_{jk} = 1$. The $\alpha_{jk} \geq 0$ determine the density of the samples; since we have no preference for particular costs here, we set all the $\alpha_{jk} = 1$ so that the simplex (that is, cost space) is sampled uniformly with respect to Lebesgue measure.

The misclassification rates for each cost sample $\boldsymbol{\lambda}'$ and classifier parameters $\mathbf{w}$ are used to make class assignments for each example in the given dataset (line 7). Usually this step consists of merely modifying the posterior probabilities $p(\mathcal{C}_k \,|\, \mathbf{x})$ to find the assignment with the minimum expected cost and is therefore computationally inexpensive as the probabilities need only be computed once for each $\mathbf{w}'$. The misclassification rates $C_{jk}(\boldsymbol{\theta}')$ $(j \neq k)$ comprise the objective values for the decision vector $\boldsymbol{\theta}' = \{\mathbf{w}', \boldsymbol{\lambda}\}$ and decision vectors that are not dominated by members of the archive E are inserted into E (line 11) and any decision vectors in E that are dominated by the new entrant are removed (line 10). We remark that this algorithm, unlike the original PAES algorithm, uses an archive whose size is unconstrained, permitting better convergence [Fieldsend et al., 2003].

A $(\mu + \lambda)$ evolutionary scheme (ES) is defined as one in which μ decision vectors are selected as parents at each generation and perturbed to generate λ offspring.[1] The set of offspring and parents are then truncated or replicated to provide the μ parents for the following generation. Although Algorithm 1 is based on a (1+1)-ES, it is interesting to note that each parent $\boldsymbol{\theta}$ is perturbed to yield

[1] We adhere to the optimisation terminology for $(\mu + \lambda)$-ES, although there is a potential for confusion with the costs λ_{jk}.

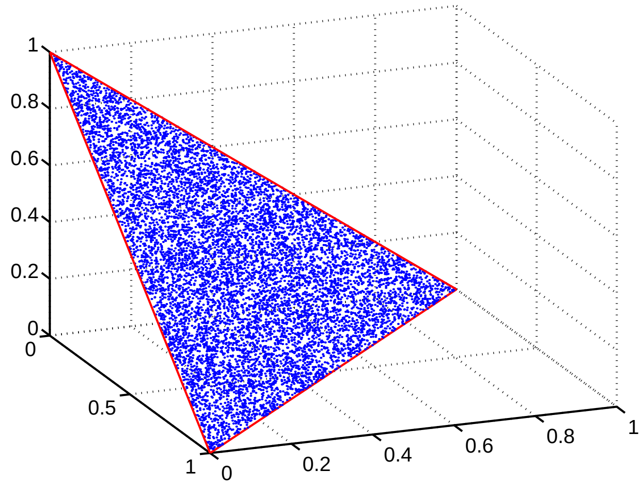

Figure 1. Samples from a 3-dimensional Dirichlet distribution, $Dir(\boldsymbol{\lambda} \mid 1, 1, 1)$.

N_λ offspring, all of whom have the classifier parameters $\mathbf{w}'$ in common. With linear costs, evaluation of the objectives for many $\boldsymbol{\lambda}'$ samples is inexpensive. Nonlinear costs could be incorporated in a straightforward manner, although it would necessitate complete reclassification for each $\boldsymbol{\lambda}'$ sample and it would therefore be more efficient to resample $\mathbf{w}$ with each $\boldsymbol{\lambda}'$.

4 Illustrations

In this section we illustrate the performance of the evolutionary algorithm on synthetic data, which is readily understood. Subsequently we give results for a number of standard multi-class problems. For simplicity we use two relatively simple classifiers, the k-nearest neighbour classifier (k-nn), which we now briefly describe in its probabilistic form [Holmes and Adams, 2002], and the multi-layer perceptron (MLP), a standard neural network.

4.1 Synthetic data

In order to gain an understanding of the Pareto optimal ROC surface for multiple class classifications we extend a two-dimensional, two-class synthetic data set devised by Ripley [1994] by adding additional Gaussian functions corresponding to an additional class. The resulting data set comprises 3 classes, the conditional density for each being a mixture of two Gaussians. Covariance matrices for all the components were isotropic: $\boldsymbol{\Sigma}_j = 0.3\mathbf{I}$. Denoting by $\boldsymbol{\mu}_{ji}$ for $i = 1, 2$ the means of the two Gaussian components generating samples for class j, the centres were located at:

$$\begin{aligned} \boldsymbol{\mu}_{11} &= (0.7, 0.3)^T & \boldsymbol{\mu}_{12} &= (0.3, 0.3)^T \\ \boldsymbol{\mu}_{21} &= (-0.7, 0.7)^T & \boldsymbol{\mu}_{22} &= (0.4, 0.7)^T \\ \boldsymbol{\mu}_{31} &= (1.0, 1.0)^T & \boldsymbol{\mu}_{32} &= (0.0, 1.0)^T \end{aligned} \tag{11}$$

Each component had equal mixing weight 1/6. The 300 samples used here, together with the equal cost Bayes optimal decision boundaries, are shown in Figure 2.

4.2 Probabilistic k-nn

One of the most popular methods of statistical classification is the k-nearest neighbour model (k-nn). The method has a ready statistical interpretation, and has been shown to have an asymptotic error rate no worse than twice the Bayes error rate [Cover and Hart, 1967]. It appears in symbolic AI under the guise of case-based reasoning. The method is essentially geometrical, assigning the class of an unknown exemplar to the class of the majority of its k nearest neighbours in some training data. More precisely, in order to assign a datum $\mathbf{x}$, given known classes and examples in the form of training data $\mathcal{D} = \{y_n, \mathbf{x}_n\}_{n=1}^N$, the k-nn method first calculates the distances $d_i = ||\mathbf{x} - \mathbf{x}_i||$. If the Q classes are a priori equally likely, the probability that $\mathbf{x}$ belongs to the j-th class is then evaluated as $p(\mathcal{C}_j \,|\, \mathbf{x}, k, \mathcal{D}) = k_j/k$, where k_j is the number of the k data points with the smallest d_n belonging to $\mathcal{C}_n$.

Holmes and Adams [2002, 2003] have extended the traditional k-nn classifier by adding a parameter β which controls the 'strength of association' between neighbours. The posterior probability of $\mathbf{x}$ belonging to each class $\mathcal{C}_j$ is given by the predictive likelihood:

$$p(\mathcal{C}_j \,|\, \mathbf{x}, k, \beta, \mathcal{D}) = \frac{\exp[\beta \sum_{\mathbf{x}_n \sim \mathbf{x}}^k u(d(\mathbf{x}, \mathbf{x}_n))\delta_{jy_n}]}{\sum_{q=1}^Q \exp[\beta \sum_{\mathbf{x}_n \sim \mathbf{x}}^k u(d(\mathbf{x}, \mathbf{x}_n))\delta_{qy_n}]}. \tag{12}$$

Here δ_{mn} is the Kronecker delta and $\sum_{\mathbf{x}_n \sim \mathbf{x}}^k$ means the sum over the k nearest neighbours of $\mathbf{x}$ (excluding $\mathbf{x}$ itself). If the non-increasing function of distance $u(\cdot) = 1/k$, then the term $\sum_{\mathbf{x}_n \sim \mathbf{x}}^k u(d(\mathbf{x}_n, \mathbf{x}))\delta_{jy_n}$ counts the fraction of the k nearest neighbours of $\mathbf{x}$ in the same class j as $\mathbf{x}$. In the work reported here we choose u to be the tricube kernel which gives decreasing weight to distant neighbours [Fan and Gijbels, 1996].

Holmes & Adams use the probabilistic formulation of the k-nn classifier as part of a Bayesian scheme in which they average over the parameters k and β. Here we regard $\mathbf{w} = \{k, \beta\}$ as parameters to be adjusted as part of Algorithm 1 as the Pareto optimal ROC surface is sought.

To discover the Pareto optimal ROC surface, the optimisation algorithm was run for $T = 10000$ proposed parameter values, with $N_\lambda = 100$, resulting in an estimated Pareto front comprising approximately 7500 mutually non-dominating parameter and cost combinations; we judge that the algorithm is very well converged and obtain very similar results by permitting the algorithm to run for only $T = 2000$ generations.

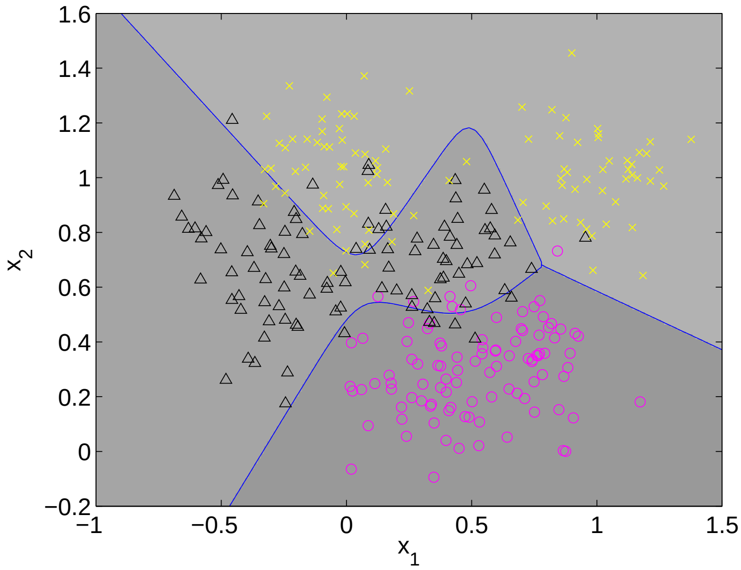

Figure 2. Synthetic 3-class data. Magenta circles mark class 1, black triangles class 2 and yellow crosses class 3. Lines mark the optimal decision boundaries for equal misclassification costs.

There are $D = Q(Q-1) = 6$ objectives to be minimised and, in common with other high-dimensional optimisation problems, visualisation of the 5-dimensional Pareto front is important for understanding the trade-offs possible. Here we use Neuroscale [Lowe and Tipping, 1996; Tipping and Lowe, 1998] to map the solutions on the Pareto front in objective space into two or three dimensional space for visualisation. Neuroscale constructs a mapping, represented by a radial basis function neural network, from the higher dimensional space into the visualisation space. The form of the mapping is determined by the requirement that distances between the representation of solutions in visualisation space are as close as possible, in a least squares sense, to those in objective space. More precisely, if d_{ij} is the distance between a pair of solutions $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ on the Pareto front and let $\hat{d}_{ij}$ be the distance between them in the visualisation space, then the Neuroscale mapping is determined by minimising the *stress* defined as

$$S = \sum_{i<j} (d_{ij} - \hat{d}_{ij})^2 \tag{13}$$

where the sum runs over all the solutions on the Pareto front.

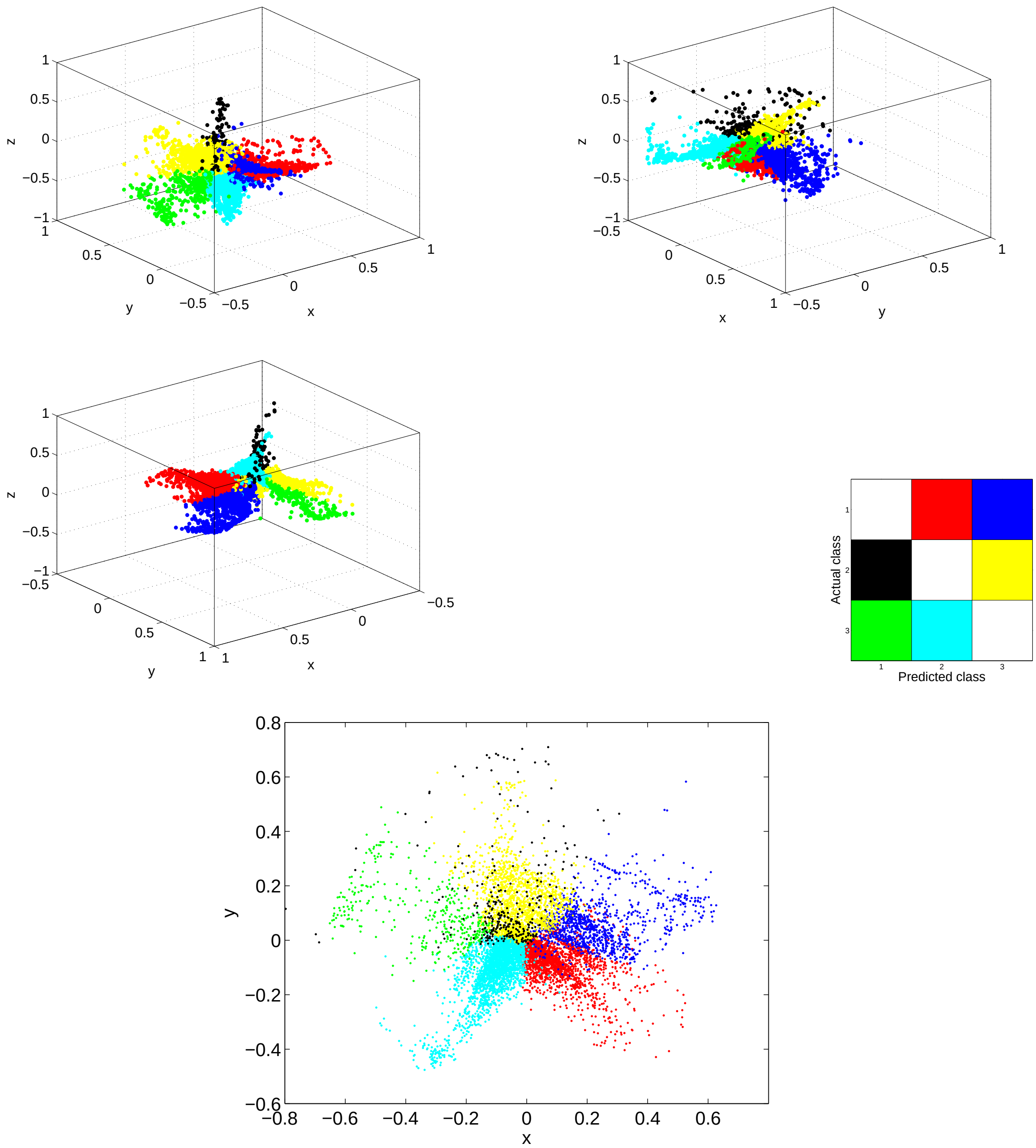

Figure 3. Two-dimensional (bottom panel) and three-dimensional (top-left panels) Neuroscale representations of the Pareto front for the synthetic data using the k-nn classifier. The top-left three panels show the front from three different angles. Solutions are coloured according to the actual and predicted classes which are most misclassified, as shown by the right panel of the middle row.

Figure 3 shows two and three-dimensional Neuroscale representations of the Pareto front. Solutions are coloured according to both the class for which most errors are made and the class into which most of those solutions are misclassified; that is, according to the largest entry in the confusion matrix for that solution. We call this the *type* of misclassification. For example, solutions coloured red correspond to classifiers that make more misclassifications of $\mathcal{C}_1$ examples as $\mathcal{C}_2$, so that $\hat{C}_{12} \geq \hat{C}_{kj}$. It is immediately apparent from the visualisations that the front is divided into regions corresponding to the misclassifications of a particular type. The three dimensional views show that these regions are broadly homogeneous with distinct boundaries between them. The two dimensional representation, however, is unable to show this homogeneity and gives the erroneous impression that solutions with different types of misclassifications are intermingled on the front (especially, for example, the $\mathcal{C}_2$ true to $\mathcal{C}_1$ predicted misclassification). We therefore prefer the three-dimensional Neuroscale representations, although we shall use the two-dimensional representation to describe locations on the front. The structure may most easily be appreciated from motion and we therefore make short movies of the fronts available from `http://www.dcs.ex.ac.uk/~reverson/research/mcroc`.

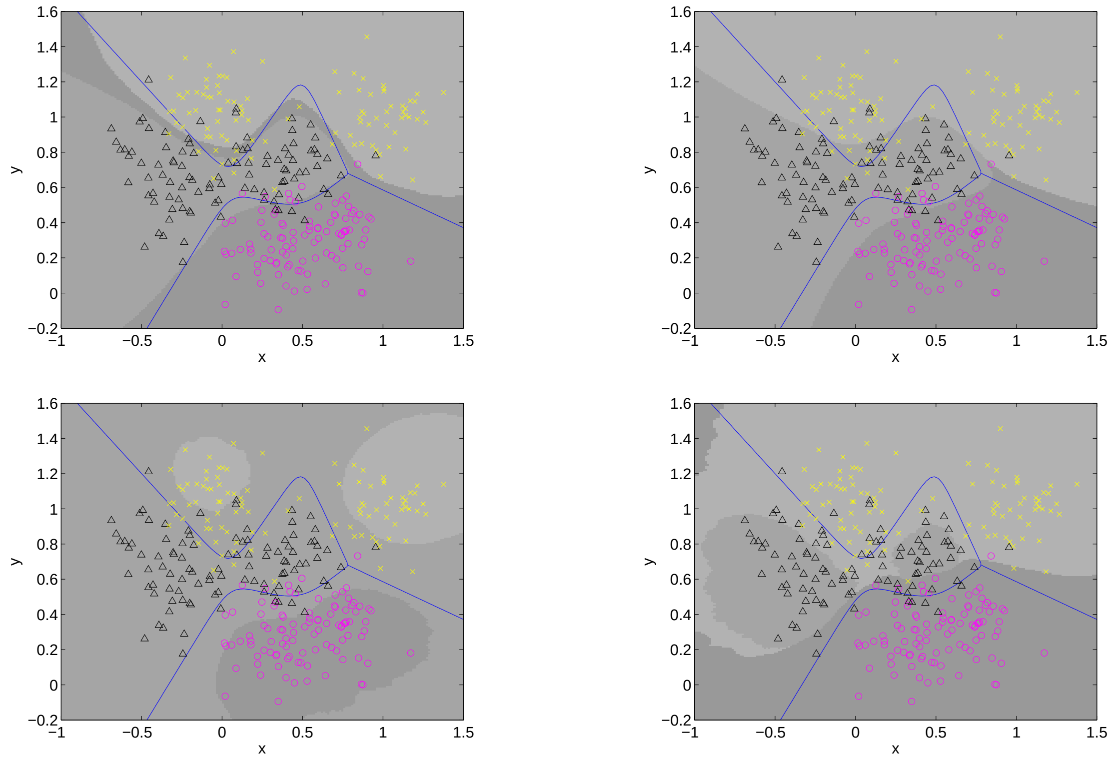

Figure 4. Decision regions for various k-nn classifiers on multi-class ROC surface. Grey scale background shows the class to which a point would be assigned. Blue lines show the ideal equal-cost decision boundary. Symbols show actual training data. *Top left:* Parameters corresponding to the middle of the 2D Neuroscale plot. *Top right:* Parameters corresponding to minimum total misclassification error on the training data. *Bottom left:* Decision regions corresponding to the minimum C_{21} and C_{23} and conditioned on this, minimum C_{31} and C_{13}. *Bottom right:* Decision regions corresponding to minimising C_{12} and C_{32}.

We emphasise that the division into distinct homogeneous regions is not an artefact of the visualisation and that information about which class was most misclassified into which was not used in constructing the Neuroscale mapping; the colours were added afterwards. In fact, it is possible to modify the objective-space distance used in the stress (13) to incorporate misclassification information as follows:

$$d'_{ij} = (1-\alpha)d_{ij} + \alpha\Delta_{ij} \tag{14}$$

where d_{ij} is the Euclidean distance in objective space between solutions $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ and Δ_{ij} is 0 if $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ make misclassifications of the same type and 1 if the make misclassifications of different types. This metric therefore tends to cluster solutions with misclassifications of the same type to a degree dependent upon α. However, we have found no additional benefit to using (14) and therefore set $\alpha = 0$.

It is clear from Figure 3 that the regions of different type meet at a single point in the centre of the Neuroscale representations. The decision regions corresponding to the classifier parameterisation close to this centre are shown in the top-left panel of Figure 4. The grey scale background shows the class to which each point would be assigned on the basis of this parameterisation. It is apparent that the decision regions quite closely approximate equal-cost Bayes rule boundaries, although there is a narrow class-1 'funnel' between classes 2 and 3. Interestingly the costs for this parameterisation are not all close to 1/6, being $\boldsymbol{\lambda} = \begin{bmatrix} 0 & 0.101 & 0.213 \\ 0.0243 & 0 & 0.285 \\ 0.0678 & 0.309 & 0 \end{bmatrix}$, although there are other nearby parameterisations with more equal costs. The top-right panel of the figure shows the decision regions that yield the smallest *total* misclassification error, 32/300. This point is also located close to the centre of the Neuroscale representation, having 2D Neuroscale coordinates $(-0.050, 0.045)$, and has very similar decision regions to the central parameterisation as might be expected since the overlap between classes is approximately comparable and there are equal numbers in each class. The principal difference from the central parameterisation is the slight shift of the decision regions away from the equal-cost Bayes optimal ones, which reflects an effective over-fitting of these particular data. Again, there are (3 in the results presented here) other parameterisations and cost combinations that achieve this minimum misclassification rate.

By contrast with the decision regions which are optimal for roughly equal costs, the bottom two panels of Figure 4 show decision regions for imbalanced costs. The bottom left panel shows a decision region corresponding to minimising C_{21} and C_{23}: this, of course, can be achieved by setting λ_{21} and λ_{23} to be large, so that every $\mathcal{C}_2$ example (black triangles) is correctly classified, no matter what the cost. For these data there are many decision regions correctly classifying every $\mathcal{C}_2$ and we display the decision regions that also minimise C_{31} and C_{13}. For these data, it is possible to make $C_{31} = C_{13} = 0$ because $\mathcal{C}_1$ and $\mathcal{C}_3$ are adjacent only along a boundary distant from $\mathcal{C}_2$ points; such complete minimisation will in general not be possible. Of course, the penalty to be paid for minimising the $\mathcal{C}_2$ rates together with C_{31} and C_{13} is that C_{32} and C_{12} are large: in fact, $C_{32} > C_{12}$ and this point lies in the region coloured cyan in the Neuroscale visualisations.

The bottom-right panel of Figure 4 shows the reverse situation: here the costs for misclassifying either $\mathcal{C}_1$ or $\mathcal{C}_3$ as $\mathcal{C}_2$ are high. With these data, although not in general, of course, it is possible to reduce C_{12} and C_{32} to zero, as shown by the decision regions which ensure that $\mathcal{C}_2$ examples (black triangles) are only classified correctly when it does not result in incorrect assignment of the other two classes to $\mathcal{C}_2$. In this case the greatest misclassification rate is C_{23} (black triangles as yellow crosses) and the parameterisation lies in the yellow region of the Neuroscale representation of the Pareto front.

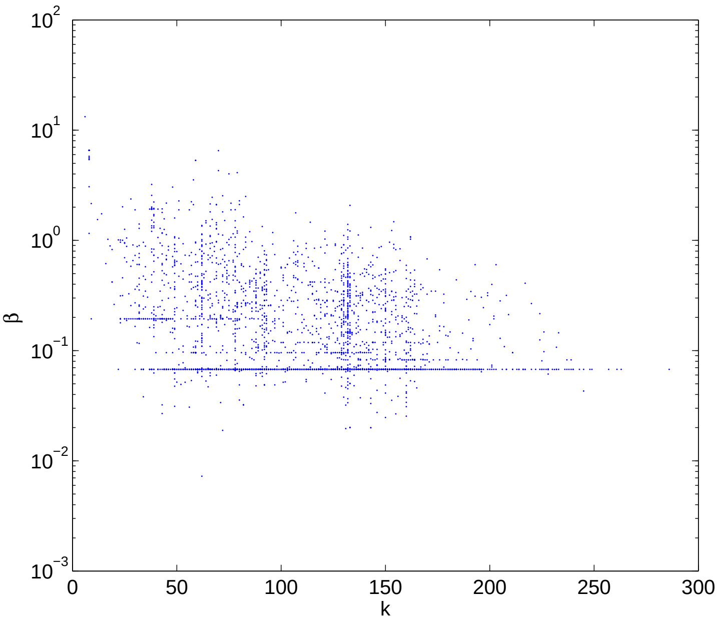

Figure 5. Parameters β versus k for members of the Pareto optimal ROC surface whose Neuroscale visualisation is shown in Figure 3.

It should be emphasised that the evolutionary algorithm has explored a wide range of cost and parameter combinations on the Pareto optimal ROC surface. Figure 5 shows the k and β pairs for the solutions on the front. The range of both these parameters is considerably larger than would be visited by a Markov chain integrating over the posterior $p(k, \beta \,|\, \mathcal{D})$ because, while the posterior probability of extreme parameter values is small, they give rise to non-dominated misclassification rates in combination with the costs and are therefore retained in the archive. Values of each λ_{jk} on the front ranges from below 10^{-3} to above 0.8, all having means of approximately 1/6, providing assurance a complete range of costs is being explored by the algorithm.

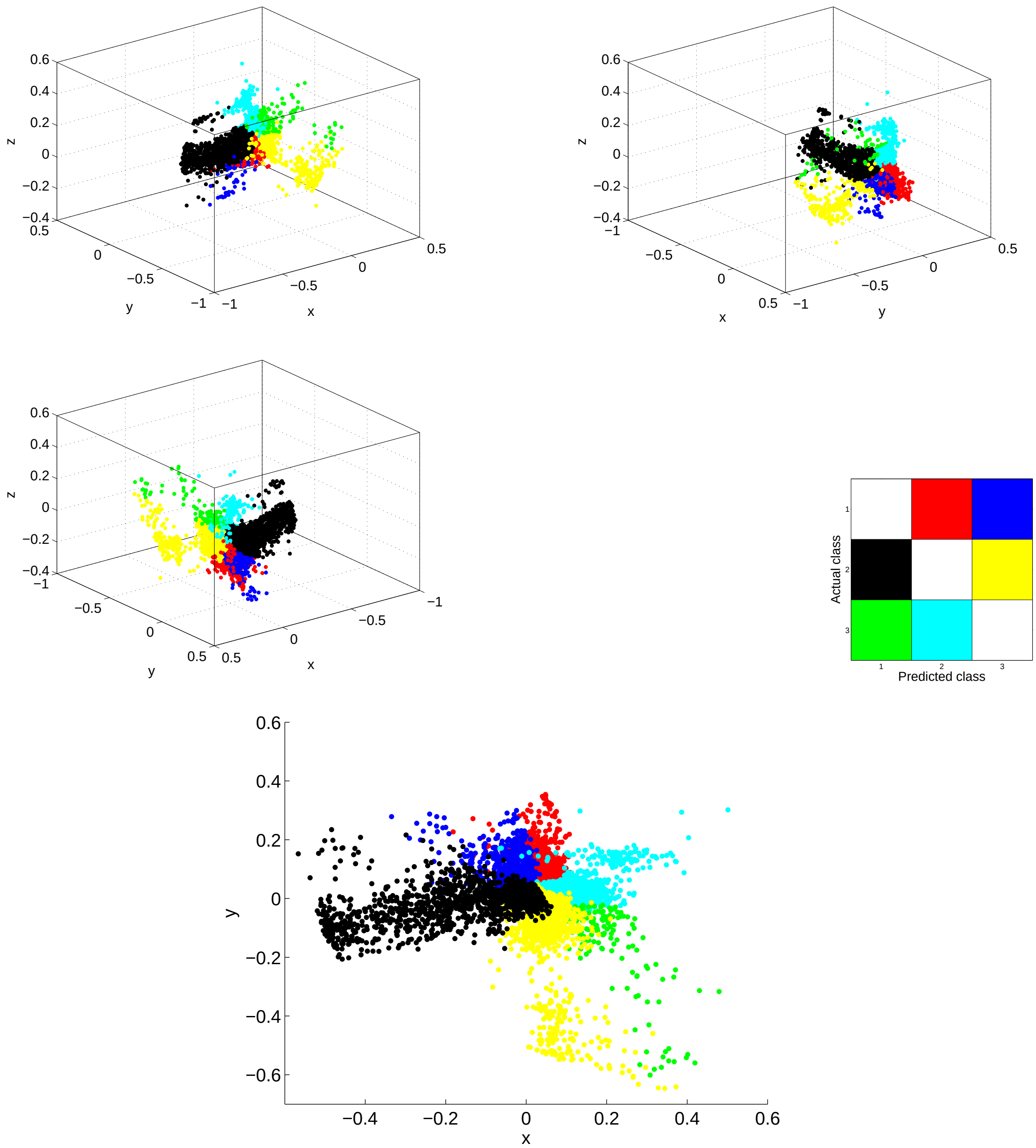

Figure 6. Two-dimensional (bottom panel) and three-dimensional (top-left panels) Neuroscale representations of the Pareto front for the synthetic data using the MLP classifier. The top-left three panels show the front from three different angles. Solutions are coloured according to the actual and predicted classes which are most misclassified, as shown by the right panel of the middle row.

4.3 Multi-layer perceptron classifiers

We also used an multi-layer perceptron (MLP) with a single hidden layer with 5 units and softmax output units as the classifier optimised by Algorithm 1. Again, the algorithm was run for $T = 10000$ evaluations of the classifier, resulting in an estimated Pareto front or ROC surface comprising approximately 4800 mutually non-dominating parameter and cost combinations. Note that for the

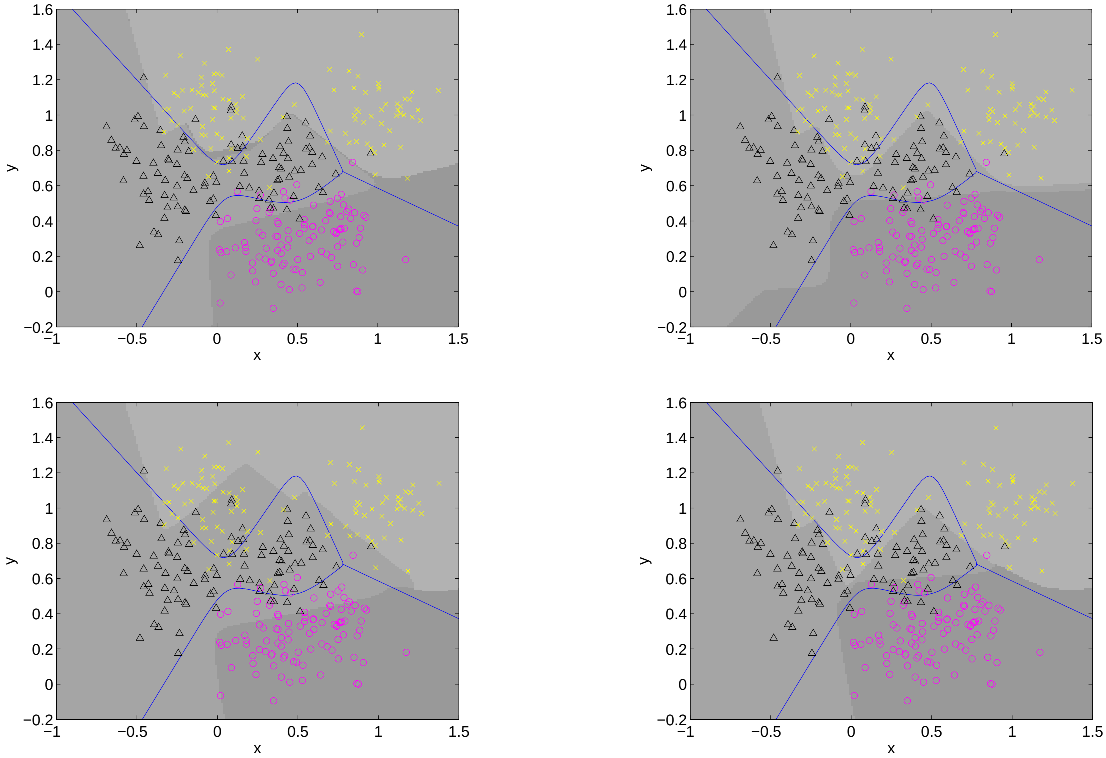

Figure 7. Decision regions for various MLP classifiers on multi-class ROC surface. Grey scale background shows the class to which a point would be assigned. Blue lines show the ideal equal-cost decision boundary. Symbols show actual training data. *Top left:* Parameters corresponding to the middle of the 2D Neuroscale plot. *Top right:* Parameters corresponding to minimum total misclassification error on the training data. *Bottom left:* Decision regions corresponding to the minimum C_{21} and C_{23} and conditioned on this, minimum C_{31} and C_{13}. *Bottom right:* Decision regions corresponding to minimising C_{12} and C_{32}.

MLP the parameter vector $\mathbf{w}$ consists of 33 weights and biases in contrast to just two parameters for the k-nn classifier. In this case the archive was initialised by training a single MLP using quasi-Newton optimisation of the data likelihood [e.g. Bishop, 1995] which finds a point on or near the Pareto front corresponding to equal misclassification costs; subsequent iterations of the evolutionary algorithm are therefore largely concerned with exploring the Pareto front rather than locating it. Neuroscale visualisations of the front are shown in Figure 6 and Figure 7 shows decision regions for points on the Pareto front corresponding to those shown for the k-nn classifier in Figure 4.

Although different in detail to the k-nn Pareto front, the visualisations of the MLP Pareto surface have similar features: the front is divided into homogeneous regions corresponding misclassifications of one type or another; and these regions all meet at a single point. Likewise the boundaries between adjacent regions on the front can be understood in terms of the configuration of the data. For example, the boundary between the cyan and yellow regions in Figure 6 corresponds to the transition between the predominance of C_{32} (cyan) and C_{23} misclassifications, which can be effected by altering the decision boundary between the black triangles and the yellow crosses in the decision region plots, Figure 7.

Decision regions corresponding to the centre of the ROC surface (Figure 7, top left) are similar to those for the k-nn classifier, as are the decision regions corresponding to minimum overall misclassification error (Figure 7, top right). The additional flexibility inherent in the MLP with 33 adjustable parameters permits the decision regions to be more finely tuned to the data: for example, the $\mathcal{C}_2$ (black triangles) $\mathcal{C}_3$ (yellow crosses) boundary in Figure 7 lies to the right of the $\mathcal{C}_2$ data point at $(-0.456, 1.21)$ so that it is correctly classified in contrast to the k-nn decision regions in Figure 4. Although no explicit measures were taken to prevent over-fitting, the decision boundaries on the front are quite smooth and do not exhibit signs of over-fitting; permitting the optimisation algorithm to run for very long times might lead to over-fitting but we have not encountered it in the work reported here.

MLP decision regions minimising C_{21}, C_{23}, C_{31} and C_{13}, shown in the bottom left panel of Figure 7 are similar to the k-nn regions (Figure 4) where the data density is high, but differ in detail where data are sparse. The same is true of the decision regions minimising misclassifications as $\mathcal{C}_2$, as can be seen by comparing the bottom right panels of Figures 4 and 7.

The decision regions illustrated in the bottom row of Figures 4 and 7 may thought of as lying on the edges of the Pareto surface because they correspond to one or more objectives being exactly minimised. These points are the analogues of the extreme ends of the usual two-class ROC curve where the false positive rate or the true positive rate is extremised. The curvature of the ROC curve in these regions is generally small, signifying that large changes in the costs yield large changes in either the true or false positive rate, but only small changes in the other. We observe a similar behaviour here: quite large changes the λ_{jk} in these regions yield quite small changes in the all the misclassification rates except the one which has been extremised, suggesting that the curvature of the Pareto surface is low in these areas.

A common use of the two-class ROC curve is to locate a 'knee', a point of high curvature. The parameters at the knee are chosen as the operational parameters because the knee signifies the transition from rapid variation of true positive rate to rapid variation of false positive rate. Methods for numerically calculating the curvature of a manifold from point sets in more than two dimensions are, however, not well developed (although see work on 3D point sets by Lange and Polthier [2005] and Alexa et al. [2003]). Initial investigations in this direction have so far yielded only very crude approximations to the curvature in the 6-dimensional objective space and we refrain from displaying them here. Although direct visual inspection of the curvature for multi-class problems is presently infeasible, we draw attention to the fact that the evolutionary algorithm yields a Pareto front of classifier parameterisations and associated costs. The Neuroscale visualisation of the front permits systematic exploration of the front and thus choice of operational parameters.

4.4 False positive rates

Humans are particularly adept at visualisation in two and three dimensions, the intrinsic dimensions of the world we inhabit, and relatively inept at visualisation of high-dimensional objects. It is tempting therefore to attempt to reduce the dimension of the ROC surface sought so as to permit visualisation. One straightforward way to achieve is to locate the trade-off surface for minimising misclassifications into each class, that is the false positive rate for each class. We minimise the Q objectives

$$F_k(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{j \neq k} C_{kj} \qquad k = 1, \ldots, Q. \tag{15}$$

The evolutionary algorithm is easily adapted to this minimisation problem and Figure 8 shows the Pareto surface obtained for the synthetic data, using the k-nn classifier and running the optimiser for $T = 10^4$ generations. We call this front the 'false positive rate front'. The figure clearly shows the tradeoff between false positive rates for each of the three classes and a 'corner' or 'knee' can be located where all three misclassification rates are small and approximately equal. Decision regions for parameterisations close to the corner are similar to the equal-costs Bayes decision regions (Figure 2) and correspond to cost matrices, $\boldsymbol{\lambda}$, with approximately equal entries. As may be expected the false positive rate for one class may be reduced, but, as the surface shows, only at the expense of raising the false positive rate for the other classes.

The false positive rate Pareto front is easily visualised (at least for three class problems), but clearly information on exactly *how* misclassifications are made is lost. However, the full D-dimensional Pareto surface may usefully be viewed in 'false positive space'. Figure 9 shows the solutions on the estimated Pareto front obtained using the full $Q(Q-1)$ objectives for the k-nn classifier, but each solution is plotted as a coloured symbol at the coordinate given by the $Q = 3$ false positive rates (15), with the symbol colour denoting the class into which the greatest number of misclassifications are made.[2] Although the solutions obtained by optimising the false positive rates directly clearly lie on the full Pareto surface (in $Q(Q-1)$ dimensions) the converse is not true and the projections

[2] A movie showing views from other angles can be found at `http://www.dcs.ex.ac.uk/~reverson/research/mcroc`.

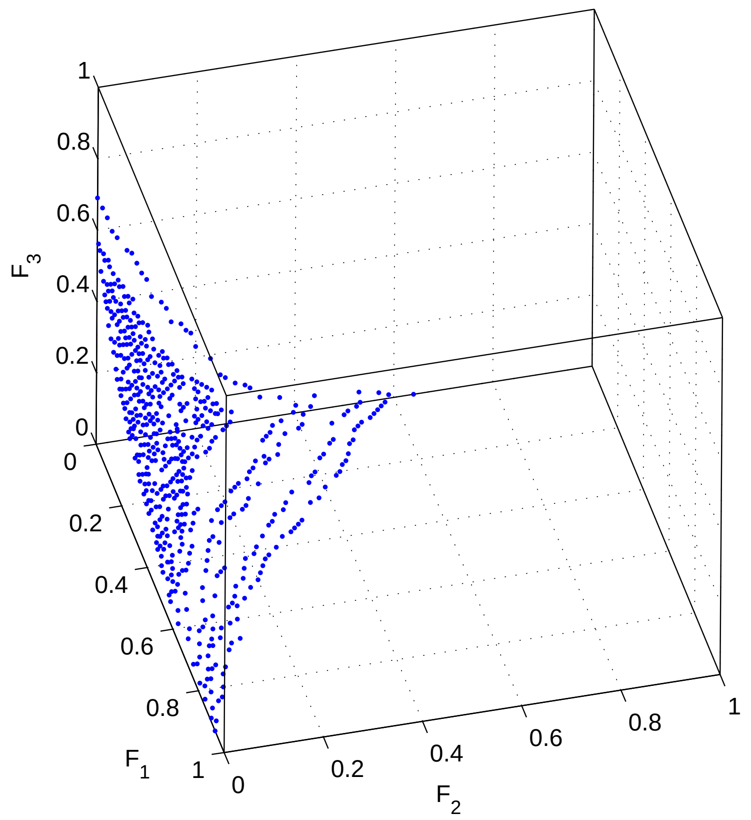

Figure 8. Estimated Pareto front where the objectives are the overall misclassification rates for each class. Synthetic data using k-nn classifiers.

into false positive space do not form a surface. Nonetheless, at least for these data, they lie close to a surface, which aids visualisation and navigation of the full Pareto front. The relation between the solutions on the full Pareto front and the false positive rate front is made more precise as follows. If A is a set of $Q(Q-1)$-dimensional solutions describing lying in the full Pareto front, let E_Q be the set of Q-dimensional vectors representing the false positive coordinates of elements of E. The extremal set of non-dominated elements of E_Q is

$$\tilde{E}_Q = \{\mathbf{f} \in E_Q \,|\, \mathbf{f} \nprec \mathbf{f}' \in E_Q\}. \tag{16}$$

Then solutions in $\tilde{E}_Q$ also lie in the false positive rate front.

5 Comparing classifiers

In two class problems the area under the ROC curve (AUC) is often used to compare classifiers. As clearly explained by Hand and Till [2001], the AUC measures a classifier's ability to separate two classes over the range of possible costs and is linearly related to the Gini index. Below we discuss the optimisation of a pairwise measure of multiple class separation introduced by Hand and Till; in this section we compare the k-nn and MLP classifiers using a measure based on the volume dominated by the Pareto optimal ROC surface.

By analogy with the AUC, we might therefore use the volume of the $Q(Q-1)$-dimensional hypercube that is dominated by elements of the ROC surface for classifier A as a measure of A's performance. In binary and multi-class problems alike its maximum value is 1 when A classifies perfectly. If the classifier allocates at random, the ROC surface is the simplex in $Q(Q-1)$-dimensional space with vertices at length $Q-1$ along each coordinate vector. The volume of the unit hypercube dominated

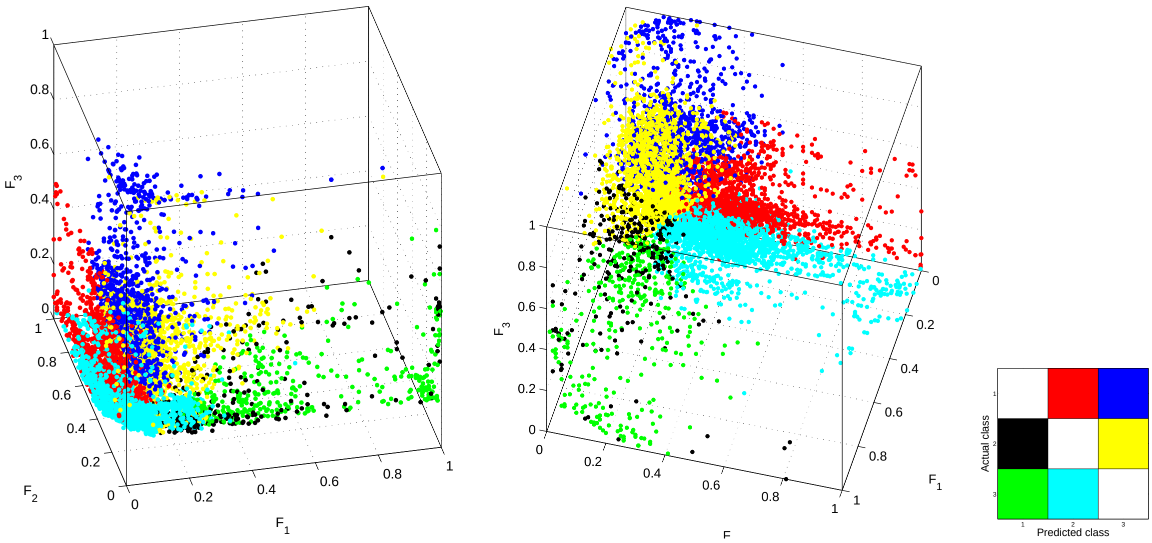

Figure 9. Two views of the estimated Pareto front for synthetic data classified with a k-nn classifier viewed in false positive space. Axes show the false positive rates for each class and points are coloured according to the class into which the greatest number of misclassifications are made. the overall misclassification rates for each class.

by this can be derived as follows: First we note that the volume of the pyramidal region between the origin and the simplex with vertices at a distance L along each coordinate vector is $\frac{L^{Q(Q-1)}}{Q(Q-1)!}$. The volume lying between the origin and the random allocation simplex is, therefore:

$$\frac{(Q-1)^{Q(Q-1)}}{Q(Q-1)!}. \tag{17}$$

Only part of this volume lies in the unit hypercube however, as the corners (excluding that at the origin) relate to infeasible regions where classification rates are > 1. Each of these $Q(Q-1)$ corner regions is also a pyramidal volume, but with sides of length $Q-2$. The total volume of the region between the origin and the random allocation simplex which *also* lies in the unit hypercube is therefore

$$\frac{(Q-1)^{Q(Q-1)}}{Q(Q-1)!} - \frac{Q(Q-1)(Q-2)^{Q(Q-1)}}{Q(Q-1)!}. \tag{18}$$

We denote this region by P. When $Q = 2$ the second term in equation 18 is zero so that the total volume (area) between the origin and the random allocation simplex is just $1/2$. This corresponds to the area under the diagonal in a conventional ROC plot (although binary ROC plots are usually made in terms of true positive rates versus false positive rates for one class, the false positive rate for the other class is just 1 minus the true positive rate for the other class). However, when $Q > 2$, the volume not dominated by the random allocation simplex is very small; even when $Q = 3$, the volume not dominated is ≈ 0.0806. We therefore define $G(A)$ to be the analogue of the Gini index in two dimensions, namely the proportion of the volume of the $Q(Q-1)$-dimensional unit hypercube that is dominated by elements of the ROC surface, but is not dominated by the simplex defined by random allocation (as illustrated by the shaded area in Figure 10 for the $Q = 2$ case). In binary classification problems this corresponds to twice the area between the ROC curve and the diagonal. In multi-class problems $G(A)$ quantifies how much better A is than random allocation. It can be simply estimated by Monte Carlo sampling of this volume in the unit hypercube.

If every point on the optimal ROC surface for classifier A is dominated by a point on the ROC surface for classifier B, then classifier B has a superior performance to classifier A. In general, however, neither ROC surface will completely dominate the other: regions of A's surface will be dominated by B and vice versa; in binary problems this corresponds to ROC curves that cross. To quantify the classifier's relative performance we therefore define $\delta(A, B)$ to be the volume of P that is dominated by elements of A and not by elements of B (marked in Figure 10 with horizontal lines). Note that $\delta(A, B)$ is not a metric; although it is non-negative, it is not symmetric. Also if A and B are subsets of the same non-dominated set W, (i.e., $A \subseteq W$ and $B \subseteq W$), then $\delta(A, B)$ and $\delta(B, A)$

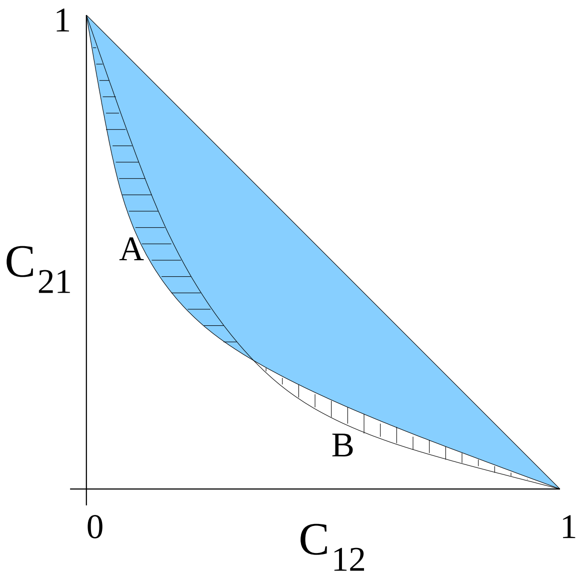

Figure 10. Illustration of the G and δ measures where $Q = 2$. Shaded area denotes $G(A)$, horizontally dashed area denotes $\delta(A, B)$, vertically dashed area denotes $\delta(B, A)$.

may have a range of values depending on their precise composition; see Fieldsend et al. [2003] for more details. Situations like this are rare in practice, however, and measures like δ have proved useful for comparing Pareto fronts.

Table 1. Generalised Gini coefficients and exclusively dominated volume comparisons of the *k*-nn and MLP classifiers.

$G(k\text{-nn})$	$G(\text{MLP})$	$\delta(k\text{-nn}, \text{MLP})$	$\delta(\text{MLP}, k\text{-nn})$
0.916	0.970	0.0001	0.054

Table 1 shows G and δ calculated from 10^5 Monte Carlo samples for the k-nn and MLP classifiers. The MLP ROC surface dominates a larger proportion of the volume and, as the δ measures show, almost every point on the k-nn ROC surface is weakly dominated by a point on the MLP surface. As noted above, the MLP has 33 adjustable parameters compared with 2 for k-nn, so it is unsurprising that the MLP front almost completely dominates the k-nn front.

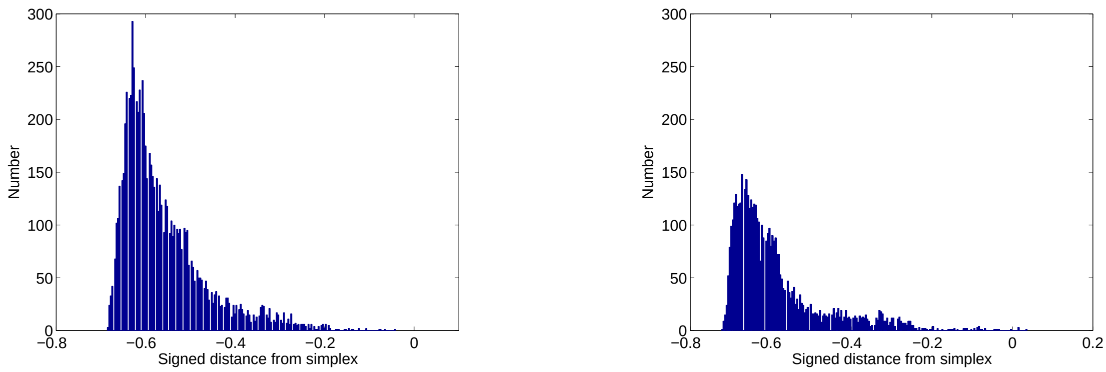

Figure 11. Distances from the random classifier simplex. Negative distances correspond to models in P. *Left:* *k*-nn; *Right:* MLP.

It should be noted that the not all of the classifiers located by the evolutionary algorithm lie in P. This occurs because in multi-class problems, performance on one misclassification rate may be sacrificed to be worse than random in order to obtain superior performance on the other rates. In fact all of the k-nn models and all but 4 of ≈ 4800 MLP models on the ROC surface lie in P. Figure 11 shows the signed distances of classifiers on the ROC surface from the random allocation simplex; negative distances correspond to classifiers in P. Clearly the majority of classifiers lie some distance closer to the origin than the random allocation simplex. In the absence of additional information or preferences (such as a maximum misclassification rate that can be tolerated for a particular class), a method of selecting the optimal classifier is to choose the one most distant from the random allocation simplex, a practice that corresponds to the selecting the classifier most distant from the diagonal on binary ROC plots.

5.1 Separating classes

Hand and Till [2001] introduced a generalisation of the AUC for comparing classifiers. In summary, their M measure is the average of the pairwise AUCs between the $Q(Q-1)/2$ pairs of classes. More precisely, Hand and Till show that the AUC is the probability, denoted $\hat{A}(j \mid k)$ that a randomly drawn member of class $\mathcal{C}_j$ will have a lower estimated probability of belonging to class $\mathcal{C}_k$ than a randomly drawn member of $\mathcal{C}_k$. Clearly a classifier which is able to separate $\mathcal{C}_k$ from $\mathcal{C}_j$ the classes has large $\hat{A}(k \mid j)$, whereas if it makes assignments no better than chance $\hat{A}(k \mid j) = 1/2$. Except in the two class problem $\hat{A}(k \mid j) \neq \hat{A}(j \mid k)$, and exchanging class labels does not alter their separability, so the classifier's ability to separate $\mathcal{C}_j$ and $\mathcal{C}_k$ is measured by $\hat{A}(j,k) = [\hat{A}(k \mid j) + \hat{A}(j \mid k)]/2$. Hand and Till then define overall performance of a classifier as:

$$M = \frac{2}{Q(Q-1)} \sum_{j<k} \hat{A}(j,k). \tag{19}$$

This measure thus measures the average ability of a classifier to separate classes, although it considers the pairwise performances of the classifier, rather than the full Pareto front. Hand and Till also describe the measure for a classifier with fixed parameters, rather than for a parameterised family of classifiers, as done in earlier sections of this article. A natural generalisation is to consider the multiobjective maximisation (for a parameterised family) of the $Q(Q-1)$ pairwise $\hat{A}(j,k)$. In fact, this leads to a simple algorithm for the maximisation of M itself, which we now describe.

The key to maximising M is that we are content to find a *set* E of parameters $\mathbf{w}$ that together maximise M. Consequently if the addition of a proposed parameter vector $\mathbf{w}'$ to E increases any one of the $\hat{A}(j,k)$ it automatically increases M; since an unrestricted set of parameters is kept, no other elements of E need be deleted so the other $\hat{A}(j,k)$ are, at worst, not decreased. This leads to the straightforward procedure outlined in Algorithm 2. As for the multi-objective evolutionary algorithm (Algorithm 1), we maintain an archive E of solutions. At each stage, a randomly selected member of E is perturbed and the M measure of the archive plus $\mathbf{w}'$ evaluated; if the addition of $\mathbf{w}'$ increases M then $\mathbf{w}'$ is retained (line 6 of Algorithm 2) and any parameters which now do not contribute to M are removed (lines 7-9).

Algorithm 2 Evolutionary optimisation of Hand and Till's M measure.

Inputs:
T — *Number of generations*

1: $E :=$ initialise()
2: `for` $t := 1 : T$
3: $\quad \mathbf{w} :=$ select(E)
4: $\quad \mathbf{w}' :=$ perturb($\mathbf{w}$) — *Perturb parameters*
5: $\quad$ `if` $M(E \cup \mathbf{w}') > M(E)$
6: $\quad\quad E := E \cup \mathbf{w}'$ — *Insert* $\mathbf{w}'$
7: $\quad\quad$ `for` $\mathbf{u} \in E$
8: $\quad\quad\quad$ `if` $M(E) = M(E \setminus \mathbf{u})$
9: $\quad\quad\quad\quad E := E \setminus \mathbf{u}$ — *Remove redundant elements*
10: $\quad\quad\quad$ `end`
11: $\quad\quad$ `end`
12: $\quad$ `end`
13: `end`

When maximising M over a family of classifiers several ROC curves for individual classifiers generally contribute to the composite ROC curve for the family. Example ROC curves for 8 classifiers resulting

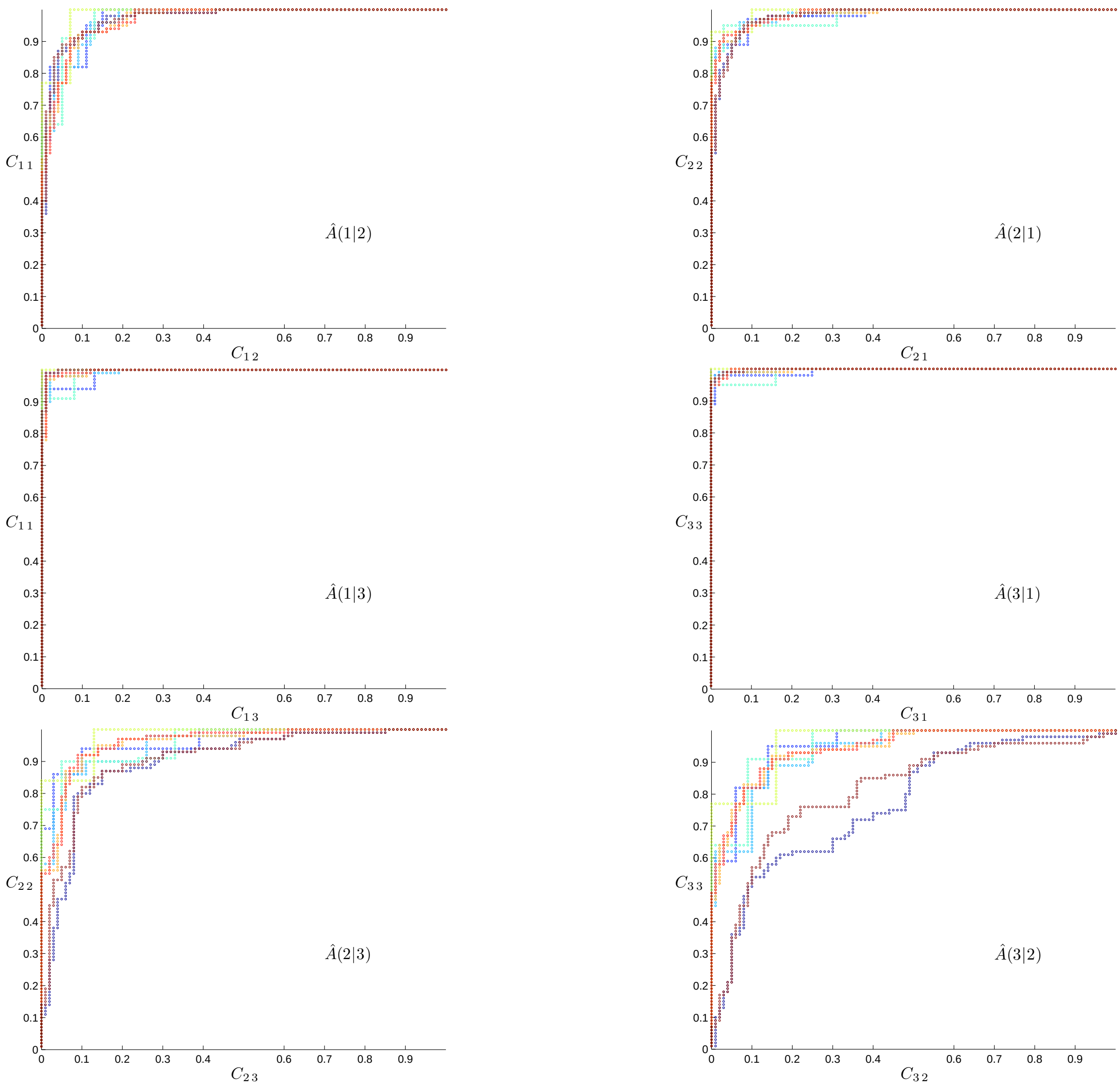

Figure 12. Pairwise ROC curves for the k-nn classification of the 3-class synthetic data set. Each row corresponds to a pair of classes. Axes correspond to the true positive rate C_{kk} and the rate at which $\mathcal{C}_k$ examples are misclassified as $\mathcal{C}_j$. Each curve corresponds to a distinct parameter combination, so that $\hat{A}(k\,|\,j)$ is the area under the envelope of the curves.

from the optimisation of M for synthetic data using the k-nn classifier are shown in Figure 12. For each pair of classes the axes of each panel are C_{kk}, the true positive rate for $\mathcal{C}_k$, and C_{kj}, the rate at which misclassifications of $\mathcal{C}_k$ examples are classified as $\mathcal{C}_j$. Each 'ROC curve' corresponds to a distinct $\mathbf{w} = \{k, \beta\}$ parameter value, and the optimised M is achieved by the envelope of these curves. Evaluation of the $\hat{A}(k\,|\,j)$ that contribute to M can be performed by applying the method described by Hanley and McNeil [1982] and Hand and Till [2001, page 174] for calculating the AUC for a single classifier to the envelope of the ROC curves.

As Figure 12 shows, after optimisation only 8 distinct (k, β) combinations contribute to the optimised $M \approx 0.991$, although during optimisation up to 20 parameter combinations were involved. These 8 models are to be contrasted with the approximately 7500 solutions on the Pareto optimal ROC surface described earlier. The Pareto optimal ROC surface, however, describes the full range of trade-offs that may be obtained between classification rates, rather than the average class separability over the range of pairwise cost ratios described by M. Selection of the operating parameters on the basis of the $\hat{A}(j, k)$ is possible, but we emphasise that the $\hat{A}(j, k)$ summarise the *overall* pairwise separability rather than permitting specific choices to be made between particular misclassification rates. Additional information is available through examination of the families of pairwise tradeoff curves such as those displayed in Figure 12. As the figure shows, separation of $\mathcal{C}_1$ and $\mathcal{C}_3$ is relatively easy, as might be expected from their separation Figure 2. On the other hand separation of $\mathcal{C}_2$ and $\mathcal{C}_3$ (black triangles and yellow crosses) is more difficult.

As the optimised M measures the ability of a particular family of classifiers to separate classes, it may be used for comparing classifiers as an alternative to the volume measures G and δ discussed earlier. Table 2 shows the optimised M and number of distinct models (distinct parameter values) contributing to M for a number of standard machine learning data sets taken from the UCI repository [Blake and Merz, 1998]. The two-class Ionosphere data is well known to be easily classified and M (actually the AUC here) is correspondingly high with only 3 distinct parameter sets for the k-nn classifier and 4 sets for the MLP. The Image data can be well separated, but only with the use of 13 parameter sets for k-nn; again better separation is achieved by the more flexible MLP, but at the expense of many more models. The DNA data with only 3 classes but 180 features requires 181 (k, β) combinations for optimal separation. In contrast, even after optimisation the Satimage data cannot be well separated with k-nn classifiers. Results are not presented for the MLP classification of the Abalone, Satimage and DNA datasets because the computation of the $\hat{A}(j \mid k)$ for envelopes of individual classifiers becomes exorbitantly expensive with many samples and models.

In summary, although M provides a global measure of a classifier's performance on a particular dataset and identifies a relatively small number of optimal parameter sets, the question of how to select an operating point remains. We are therefore inclined to use the generalised Gini coefficient and to select the operating point most distant from the random allocation simplex.

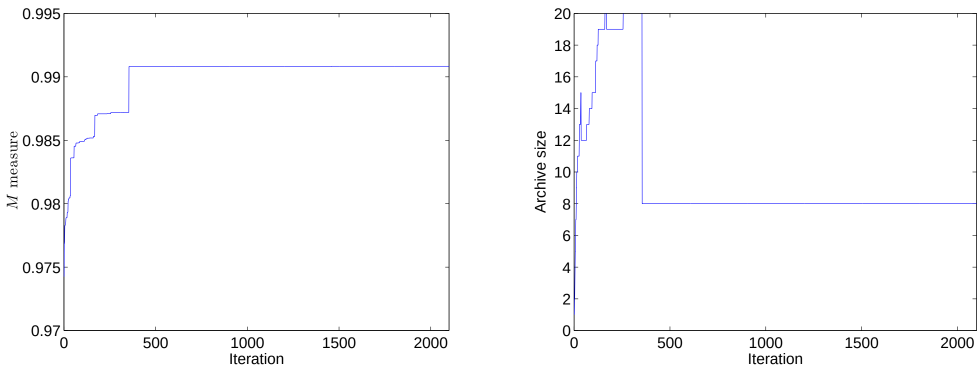

Figure 13. *Left:* Growth of M with iteration during optimisation. *Right:* Number of distinct parameter combinations contributing to M during optimisation. Results for k-nn classification of 3-class synthetic data.

Table 2. Optimised M measure for UCI data sets.

				k-nn		MLP	
Name	Examples	Features	Q	M	Models	M	Models
Abalone	3133	10	3	0.927	33		
Image	210	19	7	0.996	13	0.999	25
Ionosphere	200	33	2	0.992	3	0.996	4
Vehicle	564	18	4	0.973	11	0.966	75
Satimage	4435	36	7	0.713	20		
DNA	2000	180	3	0.989	181		

6 Conclusion

In this paper we have considered multi-class generalisations of ROC analysis from a multi-objective optimisation perspective. Consideration of the role of costs in classification leads to a multi-objective optimisation problem in which misclassification rates are simultaneously optimised. The resulting trade-off surface generalises the binary classification ROC curve because on it one misclassification rate cannot be improved without degrading at least one other. We have presented a straightforward general evolutionary algorithm which efficiently locates approximations to the Pareto optimal ROC surface.

An appealing quality of the ROC curve is that it can be plotted in two dimensions and an operating point selected from the plot. Unfortunately, the dimension of the Pareto optimal ROC surface grows as the square of the number of classes, which hampers visualisation. Projecting the surface into two or three dimensions with tools such as Neuroscale was demonstrated, although additional work is required for interpretation. Projection into 'false positive space' is an effective visualisation method for 3-class problems as the false positive rates summarise the gross overall performance, allowing further analysis of exactly which classes are misclassified into which to be focused in particular regions. Nonetheless, it is likely that problems with more than three classes will require some a priori assessment of important trade-offs because of the difficulty in interpreting 16 or more competing rates. Reliable calculation and visualisation of the curvature of the ROC surface will be important for selecting operating points; current work is focused on this area.

The Pareto optimal ROC surface yields a natural way of comparing classifiers in terms of the volume that the classifiers' ROC surfaces dominate. We defined and illustrated a generalisation of the Gini index for multi-class problems that quantifies the superiority of a classifier to random allocation. An alternative measure for comparing classifiers (and selecting an operating point) is the pairwise M measure of Hand and Till [2001]. The optimisation algorithm we describe yields a relatively small number of classifiers from which to select an operating point in terms of the $Q(Q-1)/2$ quantities $\hat{A}(k,j)$. However, these measures describe the overall trade-off between classifier models rather than permitting detailed choices about particular misclassification rates to be made.

Finally, we remark that some imprecise information about the costs of misclassification may often be available; for example the approximate bounds on the ratios of the λ_{jk} may be known. In this case the evolutionary algorithm is easily focused on the relevant region by setting the Dirichlet parameters α_{jk} appearing in (9) to be in the ratio of the expected costs, with their magnitudes setting the variance in the cost ratios.

Acknowledgment

This work was supported in part by the EPSRC, grant GR/R24357/01. We thank Trevor Bailey, Adolfo Hernandez, Wojtek Krzanowski, Derek Partridge, Vitaly Schetinin and Jufen Zhang for their helpful comments.

References

N.M. Adams and D.J. Hand. Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition*, 32:1139–1147, 1999.

M. Alexa, J. Behr, D. Cohn-Ohr, S. Heiselman, D. Levin, and C.T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9:3–15, 2003.

M. Anastasio, M. Kupinski, and R. Nishikawa:. Optimization and FROC analysis of rule-based detection schemes using a multiobjective approach. *IEEE Transactions on Medical Imaging*, 17: 1089–1093, 1998.

C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

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

C.A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, 1999.

T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley, Chichester, 2001.

R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

R.M. Everson and J.E. Fieldsend. Multi-objective optimisation of safety related systems: An application to short term conflict alert. *IEEE Transactions on Evolutionary Computation*, 2005. Under review. Draft available from `http://www.dcs.ex.ac.uk/academics/reverson`.

J.Q. Fan and I. Gijbels. *Local polynomial modelling and its applications.* Chapman & Hall, London, 1996.

J.E. Fieldsend and R.M. Everson. ROC Optimisation of Safety Related Systems. In J. Hernández-Orallo, C. Ferri, N. Lachiche, and P. Flach, editors, *Proceedings of ROCAI 2004, part of the 16th European Conference on Artificial Intelligence (ECAI)*, pages 37–44, Valencia, Spain, 2004.

J.E. Fieldsend, R.M. Everson, and S. Singh. Using Unconstrained Elite Archives for Multi-Objective Optimisation. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, 2003.

P. Flach, H. Blockeel, C. Ferri., J. Hernández-Orallo, and J. Struyf. Decision support for data mining: Introduction to ROC analysis and its applications. In D. Mladenic, N. Lavrac, M. Bohanec, and S. Moyle, editors, *Data Mining and Decision Support: Integration and Collaboration.* Kluwer, 2003.

D.J. Hand and R.J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.

J.A. Hanley and B.J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 82(143):29–36, 1982.

J. Hernández-Orallo, C. Ferri, N. Lachiche, and P. Flach, editors. *ROC Analysis in Artificial Intelligence, 1st International Workshop, ROCAI-2004, Valencia, Spain, August 22, 2004*, 2004.

C.C. Holmes and N.M. Adams. A probabilistic nearest neighbour method for statistical pattern recognition. *Journal Royal Statistical Society B*, 64:1–12, 2002. See also code at `http://www.stats.ma.ic.ac.uk/~ccholmes/Book_code/book_code.html`.

C.C. Holmes and N.M. Adams. Likelihood inference in nearest-neighbour classification models. *Biometrika*, 90(1):99–112, 2003.

J.D. Knowles and D. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

M.A. Kupinski and M.A. Anastasio. Multiobjective Genetic Optimization of Diagnostic Classifiers with Implications for Generating Receiver Operating Characterisitic Curves. *IEEE Transactions on Medical Imaging*, 18(8):675–685, 1999.

C. Lange and K. Polthier. Anisotropic fairing of point sets. *Computer Aided Geometrical Design*, 2005. To appear. Available from `http://www.zib.de/polthier/articles.html`.

D. Lowe and M. E. Tipping. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications*, 4:83–95, 1996.

F. Provost and T. Fawcett. Analysis and visualisation of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48, Menlo Park, CA, 1997. AAAI Press.

F. Provost and T. Fawcett. Robust classification systems for imprecise environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 706–7, Madison, WI, 1998. AAAI Press.

B.D. Ripley. Neural networks and related methods for classification (with discussion). *Journal of the Royal Statistical Society Series B*, 56(3):409–456, 1994.

M.J.J. Scott, M. Niranjan, and R.W. Prager. Parcel: feature subset selection in variable cost domains. Technical Report CUED/F-INFENG/TR.323, Cambridge University Engineering Department, 1998.

A. Srinivasan. Note on the location of optimal classifiers in n-dimensional ROC space. Technical Report PRG-TR-2-99, Oxford University Computing Laboratory, Oxford, 1999. URL `ftp://ftp.comlab.ox.ac.uk/pub/Packages/ILP/Papers/AS/roc.ps.gz`.

M.E. Tipping and D. Lowe. Shadow targets: a novel algorithm for topographic projections by radial basis functions. *NeuroComputing*, 19:211–222, 1998.

D. Van Veldhuizen and G. Lamont. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation*, 8(2):125–147, 2000.

X. Yao, Y. Liu, and G. Lin. Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.

M.H. Zweig and G. Campbell. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39:561–577, 1993.