# Visualisation of multi-class ROC surfaces

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

## Abstract

The Receiver Operating Characteristic (ROC) has become a standard tool for the analysis and comparison of binary classifiers when the costs of misclassification are unknown. Although there has been relatively little work in examining ROC for more than two classes – there has been growing interest in the area, and in recent studies we have formulated it in terms of misclassification rates.

Although techniques exist for the numerical comparison of the fronts generated by these new methods, the useful visualisation of these fronts to aid the selection of a final operating point are still very much in their infancy. Methods exist for the visualisation of similar surfaces, Pareto fronts, which we discuss, however the particular properties of the ROC front that the practitioner is interested in may also direct us to new and more suitable visualisation methods. This paper briefly outlines what is currently in use, and what avenues may be of interest to examine in the future.

## 1. Introduction

A significant proportion of the discussion at the end of the first Receiver Operating Characteristic Analysis in Machine Learning (ROCML) workshop, ROC Analysis in Artificial Intelligence, (Hernández-Orallo et al., 2004) was on the possible extension of ROC analysis from its traditional domain of comparing binary classifier models to classification problems with more than two classes. We have recently presented a methodology for accomplishing this by casting the ROC surface for the $Q$-class problem in terms of a multi-objective optimisation problem (Everson & Fieldsend, 2005; Fieldsend & Everson, 2005). The goal is then to simultaneously minimise the $D = Q(Q-1)$ misclassification rates. This is done using multi-objective evolutionary algorithms (MOEAs) (Coello Coello, 1999). Techniques for comparing fronts/surfaces produced using a multi-class analogue of the Gini coefficient have also been developed. When $Q \geq 3$, we refer to this as multi-class ROC.

Although we feel a reasonable methodology now exists for multi-class ROC, a difficulty is still apparent in the visualisation of the front itself. Although ROC analysis aids in the comparison of classifier families or parameterisations, eventually a single operating point needs to be chosen (or a combination of a few to operate on the convex hull). It is for the ready selection of this point that visualisation is useful – if not imperative in some situations.

## 2. Pareto dominance

Before considering the possible methods for visualising multi-class ROC surfaces, we should consider the properties of these surfaces. The optimal trade-off between the misclassification rates is defined by the minimisation problem:

$$\text{minimise} \quad C_{kj}(\boldsymbol{\theta}) \quad \text{for all } k, j. \tag{1}$$

Here $C_{kj}$ is the misclassification rate of class $k$ as class $j$. If all the misclassification rates for one classifier with parameterisation $\boldsymbol{\theta}$ are no worse than the classification rates for another classifier parameterisation $\boldsymbol{\phi}$ 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:

$$\begin{array}{c} C_{kj}(\boldsymbol{\theta}) \leq C_{kj}(\boldsymbol{\phi}) \quad \forall k, j \quad \text{and} \\ C_{kj}(\boldsymbol{\theta}) < C_{kj}(\boldsymbol{\phi}) \quad \text{for some } k, j. \end{array} \tag{2}$$

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

Appearing in *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*, Bonn, Germany, 2005.
Copyright 2005 by the author(s)/owner(s).

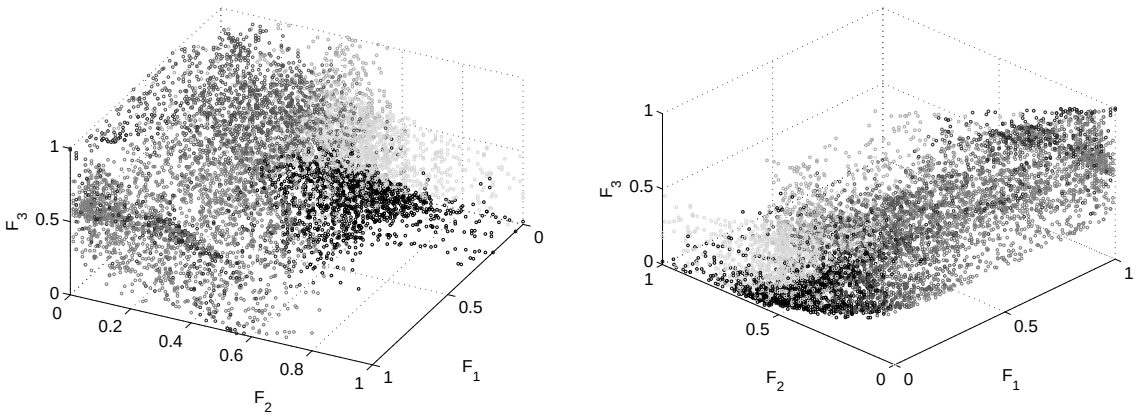


*Figure 1.* The estimated Pareto front for synthetic data classified with a multinomial logistic regression classifier viewed in false positive space. Axes show the false positive rates for each class and different greyscales represent the class into which the greatest rate of misclassifications are made. (Points better than random shown.)

member:

$$\boldsymbol{\theta} \not\prec \boldsymbol{\phi} \quad \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in A. \tag{3}$$

A solution to the minimisation problem (1) 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 – of which the optimal ROC curve for a classifier is an example. In the 3 class case (for instance) the Pareto front/ROC curve (or an estimate of it) is a 5 dimensional surface lying in a 6 dimensional space.

Evolutionary techniques based on dominance measures for locating the Pareto front for a given problem are now well developed; see (Coello Coello, 1999; Deb, 2001) and (Veldhuizen & Lamont, 2000) for recent reviews. Assuming that a good approximation to the ROC surface for a problem has been found using these types of methods, then the problem arises as to how to visualise this ROC surface to aid in the comparison and selection of the final operating point(s).

## 3. Visualisation methods

Some methods for visualising multi-dimensional Pareto fronts are already in use in the MOEA community. We discuss these below and highlight some additional properties of multi-class ROC analysis that may point to fruitful future research for their visualisation. First, however, we briefly discuss the special situation where $Q = 3$.

### 3.1. Special case, Q=3

In the special case where the number of classes $Q = 3$, we can project our $Q(Q-1)$ dimensional operating points down to $Q$ dimensions by plotting their corresponding false positive rate $F_k$ for each class; that is, by plotting:

$$F_k(\boldsymbol{\theta}) = \sum_{j \neq k} C_{kj} \qquad k = 1, \ldots, Q \tag{4}$$

where $C_{kj}$ is the proportion of class $j$ points classified as class $k$ for a model with parameterisation $\boldsymbol{\theta}$. The false positive rate front is easily visualised when $Q = 3$. Figure 1 shows the solutions on the estimated Pareto front obtained using the full $Q(Q-1)$ objectives for the multinomial logistic regression classifier, but each solution is plotted at the coordinate given by the $Q = 3$ false positive rates (4).[1] Exact information is lost on exactly *how* misclassifications are made, however the use of colour (or greyscale here) can convey additional information such as the class into with the greatest number or rate of misclassifications are made. Solutions are shaded 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.

Although the solutions obtained by directly optimising the false positive rates lie on the full Pareto surface (in $Q(Q-1)$ dimensions) the converse is not true and the projections into false positive space do not form a surface. Nonetheless, at least for these data, they lie close to a surface in false positive space, which aids visualisation and navigation of the full Pareto front.

[1] Corresponding to the front found for the multinomial logistic regression classifier on the synthetic data, from Fieldsend and Everson (2005).

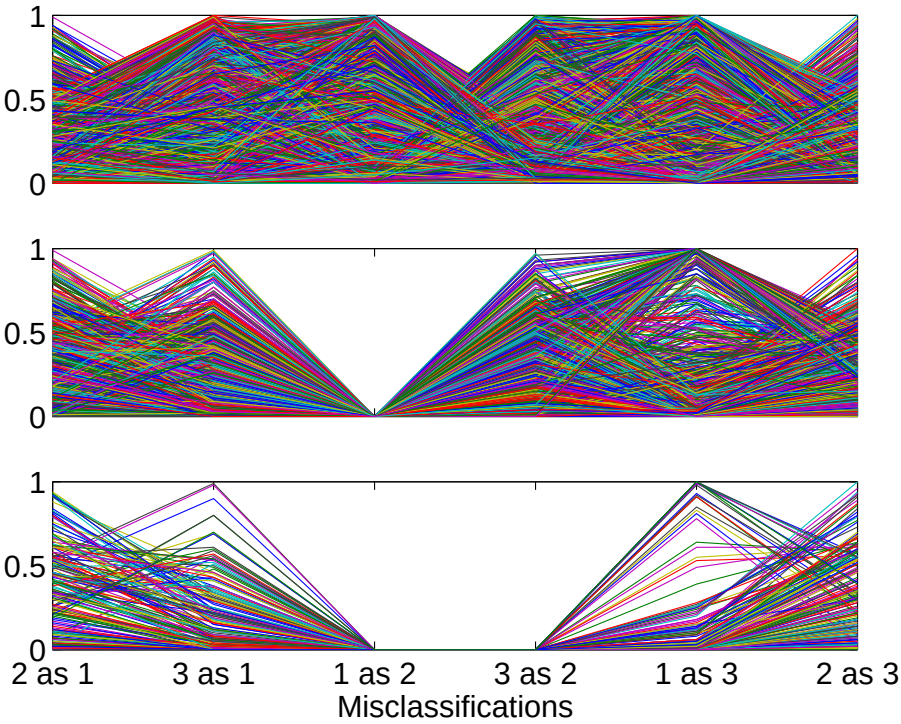


*Figure 2.* Coordinated graph for a $Q = 3$ problem. *Top:* All points on the front (over 9000). *Middle:* All 1615 points with a misclassification of class 1 to class 2 rate of zero. *Bottom:* All 205 points with a misclassification of class 1 to class 2 rate of zero and a misclassification of class 3 to class 2 rate of zero.

### 3.2. Parallel coordinated graphs

Parallel co-ordinated graphs *or* trade-off graphs (Fonseca & Fleming, 1993; Parmee & Abraham, 2004), have been used for most of the life of the MOEA community to represent the trade-offs between multiple objectives. An example is shown in Figure 2, in the situation of multi-class ROC and its $Q(Q-1)$ misclassification formulation, where $Q = 3$ (using the same data as the previous section). The top plot in the figure shows the parallel coordinates of all the points found on the multi-class ROC front; the middle plot shows those members that satisfy an additional condition, minimising the misclassification of class 1 to class 2; the bottom plot shows those members that also meet a further constraint. It is through manipulating different constraints and desires that search can be focused onto a particular area of the front—or, equivalently, a particular subset of models. A drawback of this approach is that the order in which the objectives are plotted can influence the perceived relationship between objectives (misclassification costs). This notwithstanding, parallel coordinated graphs are a commonly used and easily understood representation.

### 3.3. Pairwise plots

Another approach popular in the MOEA literature is the generation of pairwise plots of objectives.[2] That is, plotting the Pareto front in two objective space, for all different objective pairings. Figure 3 shows points on the Pareto front/ROC surface discussed in the previous section plotted in this manner.

Although this formulation allows a quick visualisation of some of the interactions between objectives, for instance what combinations of misclassification rates are not apparent in the front, the removal of all other objective information makes it difficult to ascertain the other trade-offs that are taking place when moving around this two-dimensional space (however, colouring by misclassification type, as provided in Figure 3, does give *some* indication of this). Another detraction from this method is that the number of pairwise plots it is possible to make is $\binom{Q(Q-1)}{2}$, which rapidly makes this method unwieldy as $Q$ increases.

[2]This method is also often used to view parameter space.

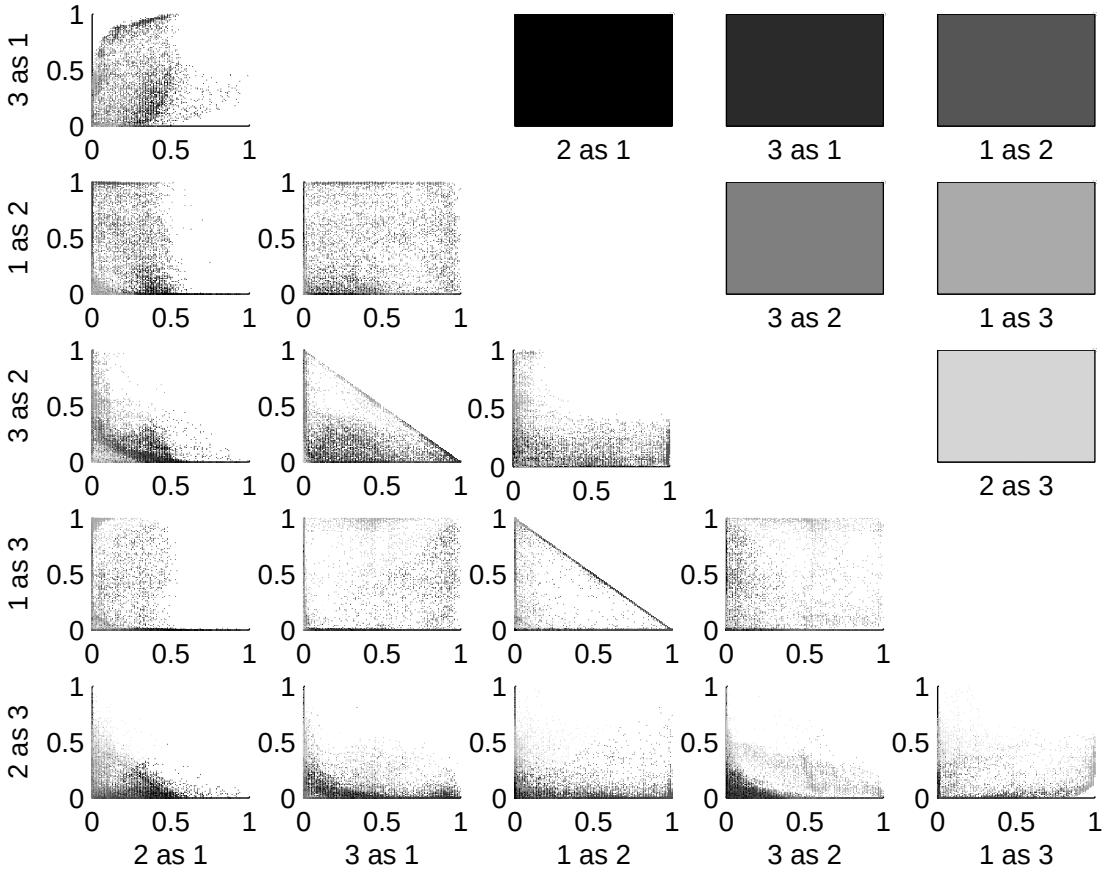


*Figure 3.* Pairwise plots of the Pareto front, with the surface mapped with regard to only two objectives at a time. Top left plots show the greyscales associated with the 6 misclassification *types* used throughout the paper.

### 3.4. Unsupervised learning methods

A variety of techniques exist for representing and visualising high-dimensional data by projecting it down into two or three dimensions. Prominent among these are principal component analysis (PCA) and factor analysis which locate a new low dimensional coordinate system which best approximates the original data. While these methods are attractive because of their simplicity and ease of calculation, they are not generally suitable for representing Pareto fronts or ROC surfaces because these surfaces are generally highly curved. PCA on the other hand is best suited to representing data clouds that are convex; indeed, the assumption underlying PCA is that the data are Gaussian distributed. This inevitably means that the PCA representation of the curved manifold forming the Pareto front will be poor. Mixtures of probabilistic principal component analysers (Tipping & Bishop, 1999) or local linear embedding (Roweis & Saul, 2000), which approximate the manifold by many *local* linear mappings, may be useful for visualisation but we investigate the use of global methods here.

Unsupervised learning methods such as Self-Organising Maps (SOMs) (Kohonen, 1995), the Generative Topographic Mapping (Bishop et al., 1998) and Neuroscale (Lowe & Tipping, 1996; Tipping & Lowe, 1998) find a low dimensional representation of data through a nonlinear mapping. SOMs have been used by Obayashi (2002) and Neuroscale has also been used by Everson and Fieldsend (2005) in order to visualise $D > 3$ Pareto surfaces. Here we demonstrate the use of two of these methods (SOMs and Neuroscale) for mapping multi-class ROC fronts to two or three dimensional space for visualisation.

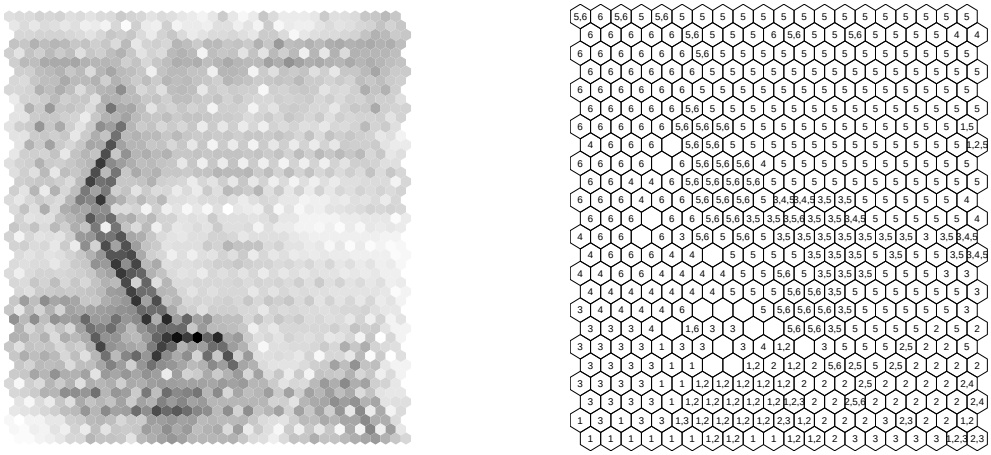

*Figure 4.* SOM for a $Q = 3$ problem. *Left:* Unified distance matrix between map units, high values (darker grey levels) indicate cluster borders. *Right:* Categorisation of units concerned with particular regions of the Pareto front. In the example here it shows which of the $Q(Q-1)$ misclassification rates it is chiefly concerned with minimising.

### 3.4.1. SOMs

The SOM is a neural network model that consists of units organised in a grid, typically in two dimensions. Each unit is connected to its neighbouring units by a relation that determines the structure of the map (here, like Obayashi (2002), we use a two dimensional hexagonal structure). Each unit is associated with a $D$ dimensional weight vector, and these parameters are adjusted during training so that the unit which is closest to an example in the training data is moved closer to the example – pulling its neighbours with it. Formally, at each iteration $t$ a point is randomly selected from the ROC curve, and its off diagonal elements of $C$ projected into a $Q(Q-1)$ length vector $\mathbf{y}(t)$. The unit closest within the SOM, $k$, based on its weight vector $\mathbf{w}(t)$, is selected based on a distance measure (typically Euclidean (Vesanto et al., 2000)),

$$||\mathbf{y}(t) - \mathbf{w}_k(t)|| = \min_i ||\mathbf{y}(t) - \mathbf{w}_k(i)|| \qquad (5)$$

After finding the unit with the closest weights, the unit's, and those of its immediate neighbours, are then altered to pull it closer to $\mathbf{y}(t)$.

We can train a SOM using the same data as used in the previous section, and view the front in the lower dimensional SOM representation, as shown in Figure 4 using a batched version of SOM (the whole ROC surface is presented to the SOM before any weights are shifted).[3] The left plot shows the unit distances between neighbours, which can indicate clusters (groups of units dealing with similar points). The right plot gives an example of how this visualisation may be presented for interpretation: it shows which of the $Q(Q-1)$ misclassification rates it is chiefly concerned with minimising (equally one could look at the rates on which they are doing worse, or other measures). This in turn shows that the SOM has identified distinct regions on the front in different distinct regions of the SOM.

### 3.4.2. Neuroscale

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 on the Pareto front between a pair of solutions $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ 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

$$S = \sum_{i<j} (d_{ij} - \hat{d}_{ij})^2 \qquad (6)$$

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

Figure 5 shows two-dimensional and three-dimensional

[3] Here we use the free Matlab SOM toolbox available from `http://www.cis.hut.fi/projects/somtoolbox`.

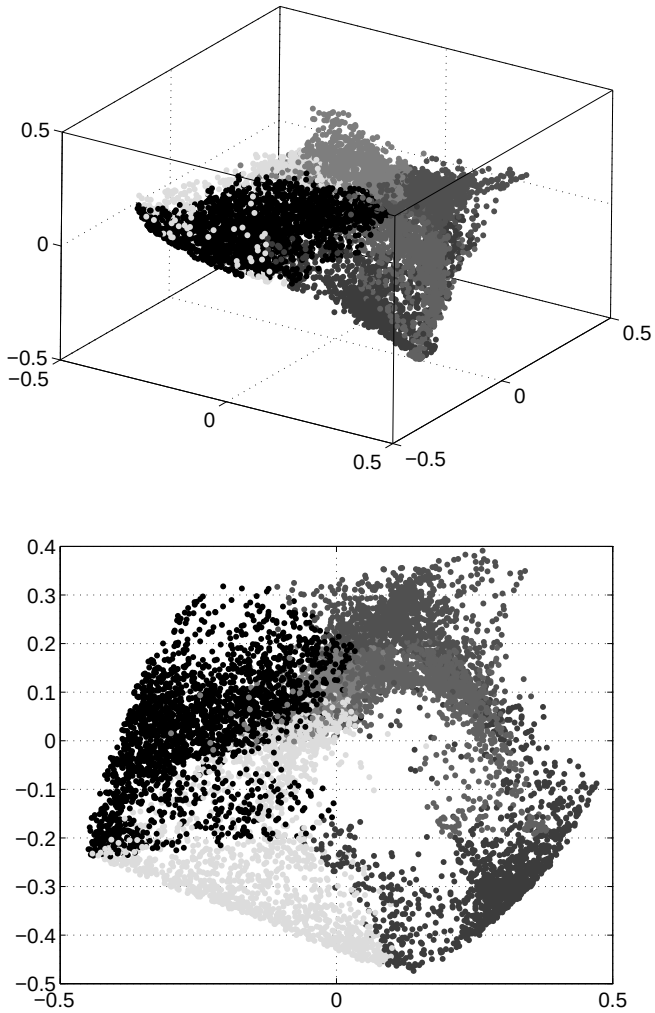


*Figure 5. Left:* Three-dimensional representation of the Pareto front for the synthetic data. *Right:* Top-down view of left plot. Solutions are shaded according to the class which is (proportionally) most misclassified.

Neuroscale representations of the Pareto front. Points are shaded according to their type of misclassification. 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 class-2 true to class-1 predicted misclassification). We therefore prefer the three-dimensional Neuroscale representations. The structure may most easily be appreciated from motion and colour and we therefore make short movies of the fronts available from `http://www.dcs.ex.ac.uk/~reverson/research/mcroc`.

### 3.5. Curvature

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. Direct visual inspection of the curvature for multi-class problems is presently infeasible and we feel this is an interesting and useful future area of research.

## 4. Conclusions

A number of different methods for visualisation of many-class ROC fronts have been briefly discussed. Some are already in use in the MOEA literature, while some are specific to the ROC case. We note that none of the projection methods employed here utilise the fact that the Pareto set is a set of non-dominated points. Although in general the mutual non-dominance of these elements must be sacrificed when projecting them into a lower dimensional space for visualisation, an area of research is projection methods that attempt to preserve approximate non-dominance. Calculating and visualising curvature has also been highlighted as a future, ROC-specific, area of interest. Other representations, like cost curves (Drummond & Holte, 2004), may also prove useful in higher dimensions.

## Acknowledgements

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, Jufen Zhang and two anonymous referees for their helpful comments.

## References

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

Bishop, C., Svensén, M., & Williams, C. (1998). GTM: The Generative Topographic Mapping. *Neural Computation*, *1*, 215–235.

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

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.

Drummond, C., & Holte, R. (2004). What ROC Curves Can't Do (and Cost Curves Can). *Proceedings of the ROC Analysis in Artificial Intelligence, 1st International Workshop. Valencia, Spain.*

Everson, R., & Fieldsend, J. (2005). *Multi-class ROC analysis from a multi-objective optimisation perspective* (Technical Report 421). Department of Computer Science, University of Exeter.

Fieldsend, J., & Everson, R. (2005). Formulation and comparison of multi-class ROC surfaces. *Proceedings of ROCML 2005, part of the 22nd International Conference on Machine Learning (ICML 2005)* (p. submitted).

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 416–423). Morgan Kauffman.

Hernández-Orallo, J., Ferri, C., Lachiche, N., & Flach, P. (Eds.). (2004). *ROC Analysis in Artificial Intelligence, 1st International Workshop, ROCAI-2004, Valencia, Spain.*

Kohonen, T. (1995). *Self-organising maps*. Springer.

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

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

Obayashi, S. (2002). Pareto Solutions of Multipoint Design of Supersonic Wings using Evolutionary Algorithms. *Adaptive Computing in Design and Manufacture V* (pp. 3–15). Springer-Verlag.

Parmee, I., & Abraham, J. (2004). Supporting Implicit Learning via the Visualisation of COGA Multiobjective Data. *Congress on Evolutionary Computation (CEC'2004)* (pp. 395–402).

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Tipping, M., & Bishop, C. (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation*, *11*, 443–482.

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

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

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (2000). *SOM Toolbox for Matlab 5* (Technical Report). SOM Toolbox Team, Helsinki University of Technology.