On the efficient use of uncertainty when performing expensive ROC optimisation

Jonathan E. Fieldsend and Richard M. Everson

Abstract— **When optimising receiver operating characteristic (ROC) curves there is an inherent degree of uncertainty associated with the operating point evaluation of a model parameterisation $\mathbf{x}$. This is due to the finite amount of training data used to evaluate the true and false positive rates of $\mathbf{x}$. The uncertainty associated with any particular $\mathbf{x}$ can be reduced, but only at the computation cost of evaluating more data. Here we explicitly represent this uncertainty through the use of *probabilistically* non-dominated archives, and show how expensive ROC optimisation problems may be tackled by only evaluating a small subset of the available data at each generation of an optimisation algorithm. Illustrative results are given on data sets from the well known UCI machine learning repository.**

I. Introduction

A growing area in the application of (evolutionary) optimisation techniques is that of the *tuning* of classifiers and other prediction systems where there are multiple measures of error to be traded-off (see e.g. [1] for a collection of recent work in the area). However, a significant problem has been encountered in transferring these methods from the academic to the industrial sphere. Whereas the data used from academic machine learning repositories (e.g. [2]) may contain a few hundred or a few thousand samples, industrial data sets can easily be of the order of hundreds of thousands (if not larger). Coupled with this the system to be optimised may also be on a specific architecture, and may therefore have limited scope for parallelisation (see e.g. [3], [4] for a problem that demonstrates both these traits). Research has already borne fruit in the development of specialised multi-objective optimisers (MOO) for general time/cost expensive problems (see for instance the recent work of Knowles [5]). However here we will focus our attention exclusively on problems whose cost (in time and/or money) is due to the use of data, and develop novel methods to trade-off this cost against the uncertainty manifest when training with fewer data (specifically in the context of receiver operating characteristic curve optimisation).

Given that the computational cost of processing data samples for the majority of widely used classifiers is linearly proportional to the sample size, the principled use of subsampling methods can be of significant use when there are constraints on time. The methods described here rely the use of probabilistic dominance to maintain a 'thick' archive which contains elements which are mutually non-dominating with a known probability. This is necessary because solutions that are only an approximation to the true (infinite data) objectives are evaluated and stored during the optimisation. The savings in evaluation time are shown to allow optimisers using these archiving approaches to find good classifier parameterisations far quicker (and to maintain them) than when using standard deterministic archiving.

The paper is organised as follows: the commonly used receiver operating characteristic is described in Section II. Dominance and probabilistic dominance are outlined in Section III and the effect of varying the number of sample data points used in an evaluation in ROC analysis is discussed in Section IV. A general multi-objective optimisation algorithm that draws on the work described in the earlier sections is introduced in Section V, and an implementation with various sample sizes is contrasted with a benchmark deterministic archiving scheme in Section VI. Conclusions are drawn in Section VII and future work in the area, based on the results described here, is also discussed.

II. Receiver Operating Characteristic curves

Depending on the classification problem at hand the cost of making the wrong classification may range from the negligible to life threatening. Some *false positives* are inevitable in most practical systems (e.g. mis-identification of someone as belonging on a "stop list" at passport control when using a recognition classifier). Attempts to reduce these false positives often leads to a decrease in the number of *true positives* the system alerts. Selecting a particular classifier and the set of operating parameters, $\mathbf{x}$, to simultaneously maximise the true positive rate, $T(\mathbf{x})$, while minimising the false positive rate, $F(\mathbf{x})$, is a multi-objective optimisation problem gaining application popularity amongst proponents of evolutionary MOO.

If one is using a classifier that gives an estimate of a data sample's probability of belonging to each of the classes, and when the relative costs of misclassification are known, it is straightforward to determine the decision rule that minimises the average cost of misclassification. However, the true costs of misclassification are frequently unknown and difficult to determine precisely (e.g. [6], [7]). In such cases those using classification systems 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 useful graphical display of the trade-off between true and false positive classification rates. Since its introduction in the medical and signal processing literatures [8], [9] ROC analysis has become a prominent method for selecting an

Jonathan E. Fieldsend and Richard M. Everson are with the School of Engineering, Computing and Mathematics, University of Exeter, Exeter, EX4 4QF, UK (email: {J.E.Fieldsend, R.M.Everson}@exeter.ac.uk).

operating point. Recent work [10], [11] reintroduced ROC analysis to the machine learning community; see [12], [13], [14] for recent collections of methodologies and applications. The fundamental trade-off between true and false positive rates permits ROC analysis to be cast as a multi-objective optimisation problem. The evolutionary optimisation point of view allows a straightforward generalisation of the two class classification methodology to multiple classes, which we describe in [15]; we shall focus on purely two-objective problems here, however the methods described within are easily applied to the multiple class ROC formulation.

III. MOO AND PROBABILISTIC DOMINATION

Formally, the generation of an optimal ROC curve can be viewed as an instance of the general multi-objective optimisation problem. In these types of problem one seeks to maximise or minimise d different objectives y_i: $y_i = f_i(\mathbf{x}), i = 1, \ldots, d$, where each objective f_i depends upon a vector of p parameters $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ (this vector is also known as a *solution*). In the case of ROC optimisation these $\mathbf{x}$ are typically the parameters of the classifier (e.g., the weights in a neural network). The parameters may also be subject to the k constraints:

$$e_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, k. \tag{1}$$

Without loss of generality we can cast the problem as strictly a minimisation one, and therefore we can express it as minimise

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_d(\mathbf{x})) \tag{2}$$

subject to

$$\mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_k(\mathbf{x})) \geq \mathbf{0}. \tag{3}$$

When multiple objectives are to be minimised as opposed to a single objective, solutions may exist for which performance on one objective cannot be improved without sacrificing a degree of performance on one or more of the other objectives. Such solutions are known as *Pareto optimal* solutions, and the set of all of these solutions is known as the *Pareto front*.

Dominance can be used to make Pareto optimality clearer. In the standard framework, a decision vector $\mathbf{x}$ is said to *strictly dominate* another $\mathbf{x}'$ ($\mathbf{x} \prec \mathbf{x}'$) iff

$$\begin{array}{c} f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \quad \forall i = 1, \ldots, d \quad \text{and} \\ f_i(\mathbf{x}) < f_i(\mathbf{x}') \quad \text{for some } i. \end{array} \tag{4}$$

A less stringent *weakly dominates* form ($\mathbf{x} \preceq \mathbf{x}'$) exists iff

$$f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \quad \forall i = 1, \ldots, d. \tag{5}$$

A set of solutions A is known as a *non-dominated set* if no member of the set is dominated by any other member:

$$\mathbf{x} \not\prec \mathbf{x}' \quad \forall \mathbf{x}, \mathbf{x}' \in A. \tag{6}$$

Most recent multi-objective optimisers maintain a non-dominated set A (known as an archive), which is its estimate of the true Pareto front, and in elitist algorithms is actively used in the search process rather than merely being a passive repository. For instance an elitist multi-objective genetic algorithm will typically have two populations which are drawn from to provide 'parents' for new potential solutions, a search population S and the elite set A. The proportion of parents that are drawn from A influencing the *degree* of elitism within a multi-objective search algorithm.

A. Probabilistic dominance

Many optimisation problems exhibit uncertainty in their function evaluation which affects the notation previously introduced. In the case of classification problems this uncertainty is due to the fact that the objective evaluations (true and false positive rates) are based on a finite set of training data, drawn from the process being modelled.[1] A repeat draw of data of the same size from the same process will often lead to a different objective evaluation, although the uncertainty (measured by the variance) in the mean reduces with the square root of the number of data samples [16]. To indicate the dependence of the objectives on the data D as well as the parameters we may denote an objective evaluation as $y_i = f_i(\mathbf{x}; D)$.

Given that the evaluation of an objective is uncertain we ought really to speak in terms of the *probability* of dominance rather than strict dominance. We use the notation, introduced in [17], $\mathbf{x} \prec^{\alpha} \mathbf{x}'$ to denote that $p(\mathbf{x} \prec \mathbf{x}') \geq \alpha$. When $\alpha = 1$ this reverts to the standard deterministic dominance.

The use of probabilistic dominance still permits us to use standard deterministic optimisation algorithms, but with an altered archiving mechanism; one in which we have a degree of *confidence*. This confidence relates to the probability that we haven't omitted solutions from the archive that may actually perform better on the *underlying* process. There are two principal ways to do this. We can take the approach that a proposal $\mathbf{x}$ is only accepted into the archive A if the *total* probability that it is dominated by other points in the archive is less that $1 - \alpha$. In this scheme a proposal $\mathbf{x}'$ is added to the archive A if

$$\sum_{\mathbf{x} \in A} p(\mathbf{x} \prec \mathbf{x}') \leq 1 - \alpha. \tag{7}$$

Once a new entrant has been accepted into A we then need to remove any solutions $\mathbf{x}$ for which

$$\sum_{\mathbf{z} \in A} p(\mathbf{z} \prec \mathbf{x}) \geq 1 - \alpha. \tag{8}$$

(Please refer to [17] for details on how this may be performed in a computationally efficient fashion.)

This approach may become problematic when the archive becomes large, as lots of small probabilities can quickly accumulate. A second scheme, which we adopt in this paper, is therefore simply to insert a proposal if no *single* archive solution dominates it with a probability greater than $1 - \alpha$. That is

$$p(\mathbf{x} \prec \mathbf{x}') \leq 1 - \alpha \quad \forall \mathbf{x} \in A. \tag{9}$$

[1] There may also be uncertainty generated by errors in the labelling of the data itself, however this is outside the scope of this particular paper.

Once a new entrant has been accepted into A in this framework, we then remove any solutions $\mathbf{x}$ for which

$$p(\mathbf{z} \prec \mathbf{x}) > 1 - \alpha \quad \text{for at least one } \mathbf{z} \in A. \tag{10}$$

It is worth noting that, while this second framework does mitigate the effect of lots of small probabilities accumulating, it does leave open the possibility of points entering the archive which are *almost* dominated by a large subset of A at the $1 - \alpha$ level.

When using a probabilistic dominance framework the crucial issue is how to appropriately calculate $p(\mathbf{x} \prec \mathbf{x}')$ for the particular problem at hand, and then deciding on an appropriate level for α. We now discuss using this for ROC optimisation.

B. Using probabilistic dominance for ROC optimisation

If the uncertainty affecting the function evaluations can be assumed to be affecting each independently then the probability of dominance can be decomposed in to a product of probabilities for each objective dimension:

$$p(\mathbf{x} \prec \mathbf{x}') = \prod_{i=1}^{d} p(f_i(\mathbf{x}) < f_i(\mathbf{x}')) \tag{11}$$

Each of the constituent probabilities $p(f_i(\mathbf{x}) < f_i(\mathbf{x}'))$ is:

$$\int_{-\infty}^{\infty} p(f_i(\mathbf{x}; D)) \int_{f_i(\mathbf{x})}^{\infty} p(f_i(\mathbf{x}'; D'))\, df_d(\mathbf{x}'; D')\, df_d(\mathbf{x}; D) \tag{12}$$

where D and D' represent the data used in the evaluation of the parameters $\mathbf{x}$ and $\mathbf{x}'$ respectively (these data may be the same or may be different as discussed in Section IV).

In [17] we discuss equation (12) where the evaluated objective is the true objective plus a Gaussian-distributed noise term. In the case of ROC optimisation the variability in the true positive rate, $T(\mathbf{x})$, and false positive rate, $F(\mathbf{x})$, can be quantified using a "stratified" bootstrap sample [4]. We concentrate on the true positive rate, but exactly analogous expressions hold for the false positive rate. Given N_1 samples from class 1 (chosen at random from the training set with replacement) and N_2 from class 2 (selected in a similar fashion), the probability of obtaining exactly y true positives in this sample is known to follow the binomial distribution:

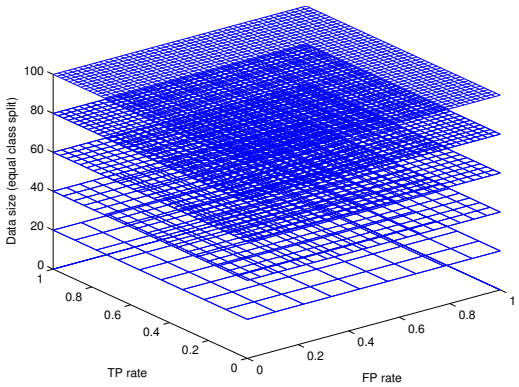
$$p(y|T(\mathbf{x})) = \binom{N_1}{y} T(\mathbf{x})^y (1 - T(\mathbf{x}))^{N_1 - y}, \tag{13}$$

The variance of the number of true positive examples in a bootstrap sample is $T(\mathbf{x})(1 - T(\mathbf{x}))N_1$, so the variance in the true positive rate is

$$\sigma_T^2 = \frac{T(\mathbf{x})(1 - T(\mathbf{x}))}{N_1} \tag{14}$$

Evaluating (12) for binomial probabilities leads to unwieldy expressions, but when the number of examples in each class is even moderately large ($N_1, N_2 \gtrapprox 20$) the binomial is well approximated by a Normal distribution:

$$p(y|T(\mathbf{x}) = \mathcal{N}(y|f(\mathbf{x}; D), \sigma_T^2) \tag{15}$$



a

Fig. 1. The possible true positive and false positive combinations (ROC elements) as data size varies (assuming identical numbers of each class in data and all objective combinations feasible).

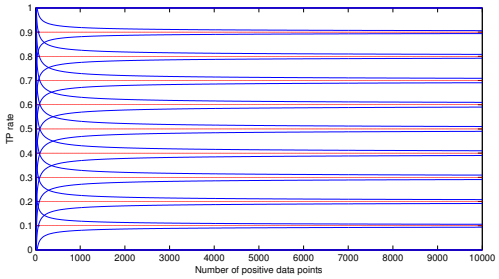


Fig. 2. Varying confidence in selected true positive rates as the number of positive samples in the training data change (illustrated using two standard deviations either side – as calculated from equation (14)).

may be used instead of (13), with σ_T^2 given by (14) and

$$\mathcal{N}(y|\mu, \sigma^2) = (2\pi\sigma)^{-\frac{1}{2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}. \tag{16}$$

is the standard normal density with mean μ and standard deviation σ^2. Using (15) in (12) gives the probability of dominance in terms of the error function [18]:

$$p(f(\mathbf{x}; D) < f(\mathbf{x}'; D')) = \frac{1}{2}\left[1 + \text{erf}\left(m/\sqrt{2}\right)\right] \tag{17}$$

where

$$m = \frac{f(\mathbf{x}'; D') - f(\mathbf{x}; D)}{\sqrt{\sigma_T^2 + \sigma_{T'}^2}}. \tag{18}$$

This expression shows that the probability of one solution dominating another can be calculated with increasing accuracy as the number of data samples increases (as the σ_T^2 terms decrease; see (14)). This reduction is proportional to the square root of the number of samples. (Note, if $f(\mathbf{x}; D) = f(\mathbf{x}'; D')$ then $m = 0$ and $p(f(\mathbf{x}; D) < f(\mathbf{x}'; D')) = 1/2$, as expected.)

IV. Confidence versus computation expense

As described above, the greater the number of data samples used in calculating the fitness evaluation (true and false positive), the greater the confidence (or lesser the uncertainty)

Algorithm 1 Elitist (1+1)–ES multi-objective evolution scheme for ROC optimisation using probabilistic dominance archives.

	α	Probability of dominance level
	D	Training data
	N	Maximum number of data samples to be processed during run
1:	$(A, tot\_num\_samp) := \text{initialise}(D, \alpha)$	Initialise archive A and record number of samples evaluated
2:	$counter := tot\_num\_samp$	Initialise sample counter
3:	`while` $counter < N$	Loop until number of data sample evaluations reaches N
4:	$\quad \mathbf{x} := \text{select}(A)$	Select from archive
5:	$\quad \mathbf{x}' := \text{vary}(A)$	Vary parameters
6:	$\quad (D', num\_samp) := \text{sample}(D, counter)$	Bootstrap sample with replacement from D
7:	$\quad (T(\mathbf{x}'), F(\mathbf{x}')) := \text{classify}(\mathbf{x}', D')$	Evaluate TP and FP rates
8:	$\quad \mathbf{x}' := \text{update}(\mathbf{x}', T(\mathbf{x}'), F(\mathbf{x}'), num\_samp)$	Associate TP and FP evaluations, and samples used, with $\mathbf{x}$
9:	$\quad$ `if` $!\text{prob\_doms}(A, \mathbf{x}', 1-\alpha)$	If $\mathbf{x}'$ is not dominated at the $1-\alpha$ level by the set A
10:	$\quad\quad A := A \cup \mathbf{x}'$	Insert $\mathbf{x}'$
11:	$\quad\quad$ `foreach` $\mathbf{x} \in A$	Check each archive element
12:	$\quad\quad\quad$ `if` $\text{prob\_doms}(A, \mathbf{x}, 1-\alpha)$	If $\mathbf{x}$ is dominated at the $1-\alpha$ level by the set A
13:	$\quad\quad\quad\quad A := A \setminus \{\mathbf{x}\}$	Remove dominated elements
14:	$\quad\quad$ `end`	
15:	$\quad$ `end`	
16:	$\quad counter := counter + num\_samp$	Update counter with the number of samples evaluated
17:	`end`	

associated with the assigned fitness. Also, in order to increase confidence in *both* the true positive rate *and* the false positive rate then both the number of class 1 samples and the number of class 2 samples must be increased. As the number of samples increases, so does the resolution in objective space (that is the number of distinct objective combinations). As illustrated in Figure 1 this growth is rapid; indeed one can quickly determine that the maximum number of distinct points in objective space is $N_1 N_2$ and, because the Pareto front is a non-decreasing curve, the maximum number of possible points in the Pareto front is $\min(N_1, N_2)$. Another effect of having a higher number of data points is the reduced uncertainty illustrated in Figure 2. Here the variability in the form of two standard deviations to either side of selected true positive rates is given, as the number of positive samples in the data set is varied – note for any particular number of positive samples in the data set, the standard deviations at different TP rates vary.

The increased confidence from evaluating a parameter set on a larger data sample does not come for free however. This extra data needs to be processed by whatever prediction model depends upon the parameters $\mathbf{x}$, and this will take time, which is typically linear in the number of data points (for any given classifier topology and parameterisation). Indeed there is a ready trade-off between the confidence one can obtain in the accuracy of the assigned ROC evaluation of an $\mathbf{x}$ and the computation time it takes assess the objective functions. There are many industrial classification problems that take a significant amount of time to evaluate on the training data provided (e.g. [4] discusses a classifier problem that takes 5 minutes per evaluation). Techniques that can reduce this cost while still leading to good solutions are therefore of significant use.

V. A General Probabilistic Multi-Objective Optimisation Algorithm

Algorithm 1 presents a general framework for implementing a probabilistic archiving approach in a $(1+1)$ multi-objective evolution strategy (ES). It is worth noting that the only parts which vary from a deterministically archiving (1+1)–ES are lines 8-16 which are concerned with the maintenance of a probabilistically non-dominated algorithm and line 6 which subsamples the data prior to evaluation. As such the archiving approach is easily inserted into any other multi-objective optimiser which otherwise uses a non-dominated archive.

Algorithm 1 commences with the initialisation of the archive with a number of seed solutions (the insertion taking the same fashion as lines 9-15), the counter which keeps track of the total number of data points processed is then updated with the archive initialisation cost in line 2. Line 6 allows the number of data points sampled to be dependent upon the point in the optimisation process (if desired) and line 8 ensures that the number of data points used to calculate the objective evaluation is maintained by the solution for later use in the archiving process (lines 9-15).

VI. Experiments

In this section we present an analysis of the effect of using a probabilistically non-dominated archive when optimising ROC problems, and of varying the amount of training data used during the optimisation process (line 6 of Algorithm 1). A simple evolutionary optimiser is used to implement the framework, but as discussed above, the archive scheme should fit into any elitist MOO framework.

The benchmark algorithm is an elitist (1+1)–ES, similar to that used in [4], [15]. At each generation a member is

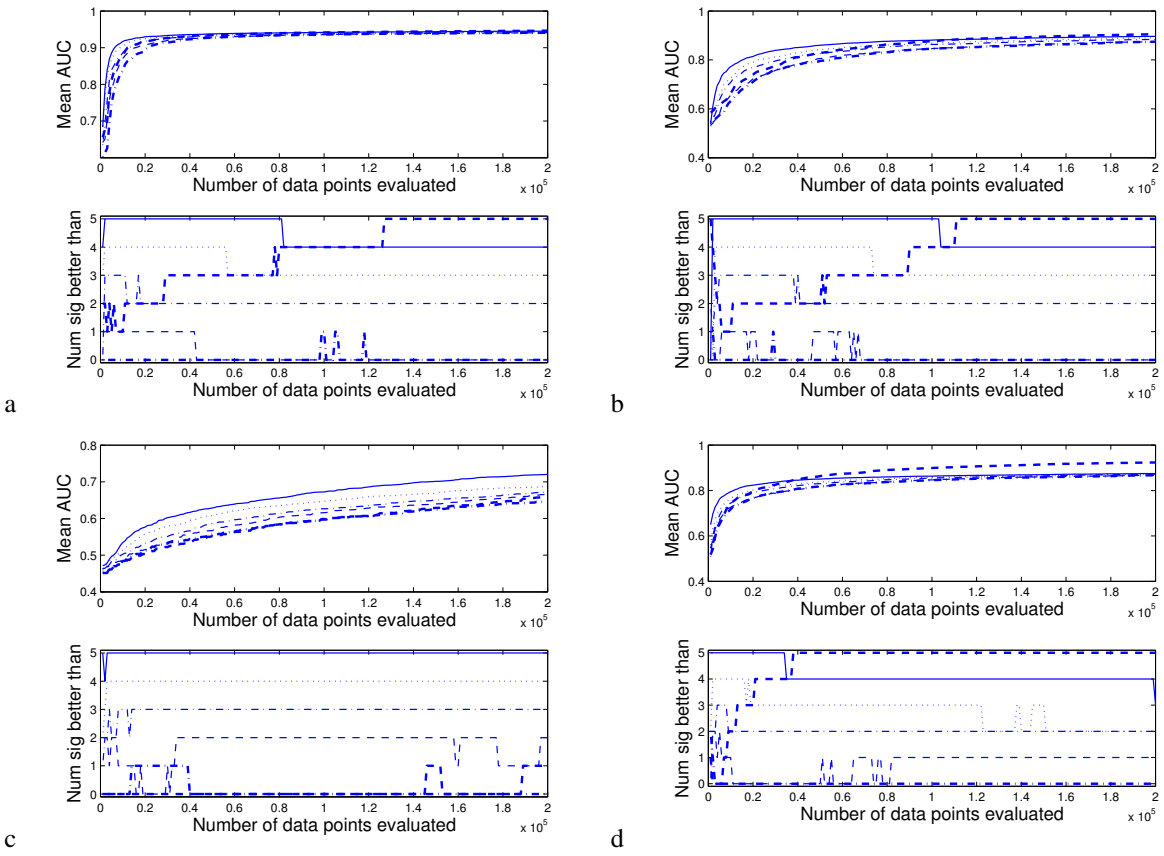


Fig. 3. Plots of mean AUC on complete training data and statistical significance versus number of data points for different optimisation regimes using (a) Ripley's synthetic data, (b) UCI Australian credit data, (c) UCI chess data and (d) UCI heart data. In each subplot; *Top:* plot of mean AUC versus number of data point evaluations for different optimisers. AUC averaged over 20 runs for each method (AUC calculated using full training data). Solid line denotes the mean when optimisation is based on a bootstrap sample (with replacement) of 20% of the training data. Dotted line 40%, dash-dotted line 60%, dashed line 80%. The thick dash-dotted line denotes 100%. The thick dashed line is the benchmark optimiser results. *Bottom* subplot of number of other optimisers each optimiser is significantly better than on the AUC measure (at the 0.05 level using the non-parametric Mann-Whitney "U" test).

selected from the archive and perturbed. It is then evaluated on all the training data, its true and false positive rates noted and archived in the standard fashion if appropriate.[2] This algorithm is modified in the comparison implementations to conform to Algorithm 1. α is fixed at 0.2 and the sample method is implemented with sample size equal to $0.2|D|$, $0.4|D|$, $0.6|D|$, $0.8|D|$ and $1.0|D|$.[3]

The vary and select schemes were identical across all optimisers. In vary, the probability of an element of $\mathbf{x}$ being mutated was 0.2, with the mutation being the addition of a Laplacian-distributed random number with a width of 0.8. select was implemented in a similar fashion to [4], [15], with an objective dimension chosen at random, followed by a random draw of a value from the range of that objective in the archive; the archive member to be copied and perturbed is that with the closest objective evaluation to this value. The initialise scheme randomly generates 10 parameterisations $\mathbf{x}$ (from draws from a Normal distribution) and evaluates each of these parameterisations on a bootstrap sample of the training data (or in the case of the benchmark optimiser, all the training data). The (probabilistically) non-dominating members of these initial 10 forming the initial archive A.

The algorithms were evaluated on four two-class classification test problems: the synthetic problem described in [19], along with three problems from the UCI machine learning repository [2] (details provided in Table I).

TABLE I
DATA SET DETAILS.

Data set	Source	$\lvert D\rvert$	Test size	Features
Synthetic	[19]	250	1000	2
Australian credit	[2]	460	230	14
Chess	[2]	2130	1066	36
Heart	[2]	180	90	13

The classifier used for all problems was the Netlab [20] implementation of an MLP with 10 hidden units and a logistic output unit activation function thresholded at 0.5. All optimisers were run for up to 200,000 data sample evaluations on

[2]For the purposes of this paper each parameterisation is treated as a hard classifier. It is worth noting if in practice one is using a soft classifier a range of true and false positive rates may typically be accessed for a single model parameterisation by varying a classification threshold.

[3]Note this last model is not the same as the benchmark as the bootstrap sampling is with replacement.

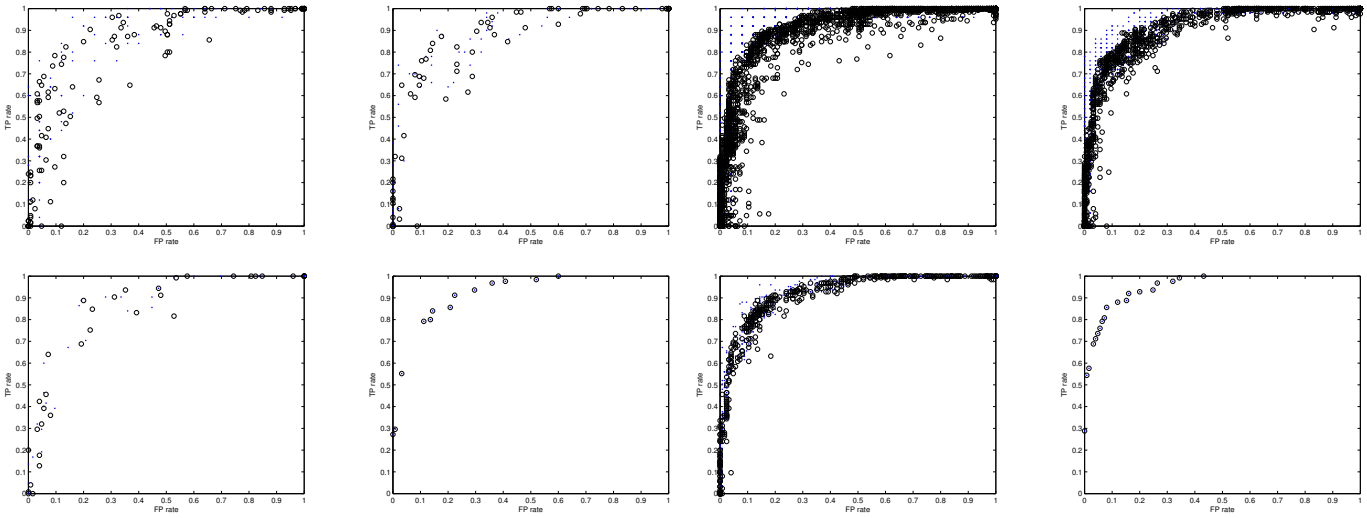


Fig. 4. Plot of example archives after 10,000 data samples evaluated on the synthetic Ripley data. Top left each objective calculation made on a bootstrap sample (with replacement) of 20% of the training data. Top right 40%, bottom left 100%. Bottom right is the benchmark optimiser results. Dots indicate the objective values held in the archive and circles indicate the classifiers evaluated on all the training data.

Fig. 5. Plot of example archives after 200,000 data samples evaluated on the synthetic Ripley data. Top left each objective calculation made on a bootstrap sample (with replacement) of 20% of the training data. Top right 40%, bottom left 100%. Bottom right is the benchmark optimiser results. Dots indicate the objective values held in the archive and circles indicate the archived classifiers evaluated on all the training data.

each training set and each run was repeated 20 times to allow statistical comparisons. During the optimisation process the true positive and false positive rates on all the training data were also recorded for the archived solutions and stored in a second set whose membership was identical to the primary archive but which played no part in the optimisation process. Clearly this incurred an additional computational cost, however in practice this computation would only be required on the final archive after an optimisation run has completed when using a probabilistic archiving optimiser in order to reassess the final solutions with greater confidence. By maintaining this secondary store we can track the effect of stopping the optimisers at any point between 0-200,000 data samples and assess their comparative performance on a fixed data set.

The area under the curve (AUC) is one of the most popular measures for assessing the quality of an ROC curve [8] (which resembles the commonly used hypervolume measure in the MOO literature, with the volume being the unit square). This was calculated on the secondary store for each optimiser every 1000 data samples.

Empirical results are presented in Figure 3, which shows the average growth in AUC for each optimiser as the number of samples processed increases, along with the number of other optimisers each optimiser is significantly better than at each sample level (significance measured using the non-parametric Mann-Whitney "U" test at the 0.05 level). As can clearly be seen the optimisers using a small subsample of the data rapidly find good solutions. On most data sets the deterministic archiving optimiser, which processes more data samples at each parameter evaluation, catches up and overhauls those optimisers subsampling the data in the later stages. This result is what we would have predicted (and the latter point to be expected). The optimisers which only subsample the data are able to evaluate a larger number of different model parameterisations for the same sample processing cost, and therefore should be expected to rapidly push forward (when archived using an accurate assessment of the uncertainty/confidence that can be attributed to their evaluations). As the optimisation process matures the benchmark model catches them up on the AUC measure as it uses the entirety of the training data in the archiving of its solutions, which the AUC is calculated on here. (Though this does leave it susceptible to potentially *overfitting* to this data, as we shall see Section VI-A.)

Indicative plots of the archives produced for the Ripley data are provided in Figures 4 and 5, with the primary (search) archive indicated with dots and the secondary store (evaluated on all the training data) indicated with circles. The probabilistic archives can be seen to be getting progressively 'narrower' as the proportion of the training data sampled is increased due to the increase in evaluation confidence. Again note that the optimiser that evaluates the objectives on a sample (drawn with replacement) whose size is the same as *all* the training data still yields a 'thick' front as the data set is still finite and there is therefore still some uncertainty in the "true" objective values.

Of crucial importance is the point at which the benchmark approach overtakes the probabilistically archiving methods in terms of AUC. For the Ripley data this can be seen in Figure 3 to occur between 80,000 and 128,000 data sample evaluations; for the Australian credit data this occurs just after 100,000 data sample evaluations; and for the heart data around 38,000 sample evaluations. For the chess data set the

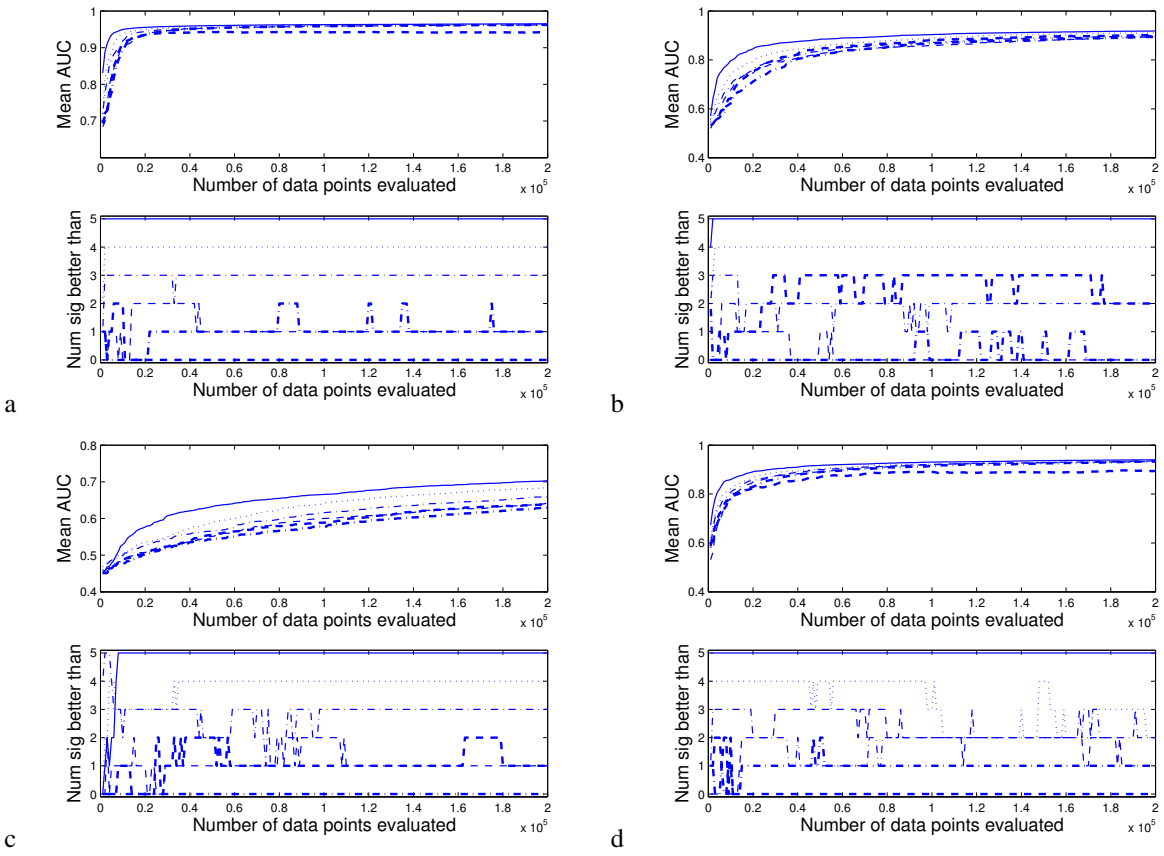


Fig. 6. Plots of mean AUC on test data and statistical significance versus number of data points for different optimisation regimes using (a) Ripley's synthetic data, (b) UCI Australian credit data, (c) UCI chess data and (d) UCI heart data. In each subplot; *Top:* plot of mean AUC versus number of data point evaluations for different optimisers. AUC averaged over 20 runs for each method (AUC calculated using full training data). Solid line denotes the mean when optimisation is based on a bootstrap sample (with replacement) of 20% of the training data. Dotted line 40%, dash-dotted line 60%, dashed line 80%. The thick dash-dotted line denotes 100%. The thick dashed line is the benchmark optimiser results. *Bottom* subplot of number of other optimisers each optimiser is significantly better than on the AUC measure (at the 0.05 level using the non-parametric Mann-Whitney "U" test).

benchmark model is still significantly worse than 4 of the 5 probabilistic archiving optimisers after 200,000 data samples evaluations. Interestingly the chess data is also the largest dataset, and there is a direct correlation between the number of data points in the complete training set and the time it takes for the standard optimiser to approach the performance of the subsampling probabilistic optimisers. This result is a useful one if it holds up in general as computationally expensive industrial classifiers in the authors' experience tend to have a large corpus of training data.

A. *Generalisation error*

Up until now we have been concerned with the performance of solutions on the training data, the data available to alter the model parameters. We end the empirical section with an analysis of the generalisation performance by evaluating the true and false positive rates on independent test data. Figure 6 replicates the results discussed in the previous section, but with the secondary archive storing performance evaluations for the true and false positive rates on the test data for each problem. When compared to the set of solutions stored by the optimisers using probabilistically dominating archives the benchmark optimiser is seen to significantly under-perform on all data sets, because the optimisation has led to classifiers over-fitted to the particular training data set. The probabilistic archive achieve a better generalisation performance because a sample used during training is statistically equivalent to the others, but, because each sample is different, and the archive maintained is probabilistically non-dominated, the classifiers cannot be optimised to any particular one. An obvious question that arises from this is how to determine the better generalising classifiers from those in the archive *prior* to evaluating them on the test data (see Figure 7 for an example of the range of solution performance). At this point in time it is unclear, although aggregating and ensemble techniques [1] may be an appropriate starting point (based on *e.g.* regions of classifiers from the archive) – and this is an area of current research by the authors.

VII. Conclusion

In this paper we have applied the probabilistic dominance framework to ROC optimisation – a problem which exhibits inherent uncertainty due to the function assessment depending on a data set of finite size. Methods for maintaining

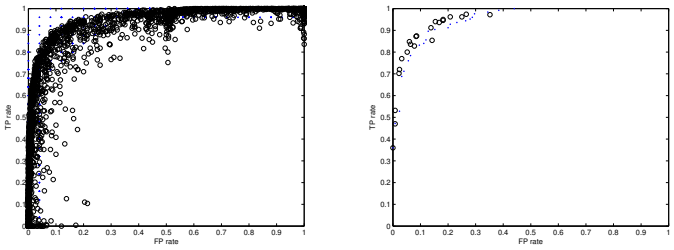


Fig. 7. Plot of example archives after 200,000 data samples evaluated on the synthetic Ripley training and test data. *Left:* each objective calculation made on a bootstrap sample (with replacement) of 20% of the training data. *Right:* standard deterministic archiving optimiser results. Dots indicate the objective values maintained in the archive and circles indicate the evaluation of $\mathbf{x}$ on the test data.

a probabilistically non-dominated archive in this situation were introduced and results shown on four different well-known data sets of varying size and difficulty. A benchmark algorithm was compared to other optimisers based on the benchmark, but maintaining probabilistically non-dominated archives using different data subsampling proportions. As the computational cost for evaluating these types of problem is proportional to the quantity of data processed by any particular parameterisation of a classifier, these optimisers were compared in terms of amount of total data processed during an optimisation run (using the widely used AUC measure). It was observed that the probabilistic archiving optimisers' performance significantly outperformed the deterministic archiving algorithm when the total amount of data processed during an optimisation run was small (around 100,000 for three of the four problems examined here), however as the amount processed grew large the deterministic algorithm overtook the probabilistic algorithms (with respect to the AUC measure calculated on the training data) – although for the problem with the largest amount of data this was not the case. One potential reason for this is that the optimisers have still yet to converge on this problem (as evinced by the AUCs for all optimisers still increasing at the 200,000 sample cut-off point). As such the probabilistic archiving optimisers may still be in their advantageous stage.

It was however also shown that the deterministic archiving algorithm was prone to overfitting, as the solutions maintained by it were less capable of generalisation than a number of those maintained by the probabilistic archiving optimisers. (The deterministic benchmark algorithm consistently performing worse on the AUC measure calculated on the test data). Although this result shows the probabilistic archiving method actively maintains better general solutions, ways to effectively select this subset *a priori* are, as yet, unclear and is an area worthy of further research. Finally α was not tuned at all in this paper, and it would be useful to evaluate the effect different α values have on the convergence speed of optimisers using probabilistic archives.

VIII. Acknowledgement

The authors would like to express their thanks to William Reckhouse and the staff from the Nation Air Traffic Service Operational Analysis and Support Group for their helpful comments during the generation of this paper.

References

[1] Y. Jin, Ed., *Multi-Objective Machine Learning*, ser. Studies in Computational Intelligence. Springer, 2006, no. 16.

[2] C. Blake and C. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[3] J. Fieldsend and R. Everson, "ROC Optimisation of Safety Related Systems," in *Proceedings of ROCAI 2004, part of the 16th European Conference on Artificial Intelligence (ECAI)*, J. Hernández-Orallo, C. Ferri, N. Lachiche, and P. Flach, Eds., 2004, pp. 37–44.

[4] R. Everson and J. Fieldsend, "Multi-objective optimisation of safety related systems: An application to short term conflict alert," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 2, pp. 187–198, 2006.

[5] J. Knowles, "ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[6] A. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145–1159, 1997.

[7] N. Adams and D. Hand, "Comparing classifiers when the misallocation costs are uncertain," *Pattern Recognition*, vol. 32, pp. 1139–1147, 1999.

[8] J. Hanley and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 82, no. 143, pp. 29–36, 1982.

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

[10] F. Provost and T. Fawcett, "Robust classification systems for imprecise environments," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. Madison, WI: AAAI Press, 1998, pp. 706–7.

[11] ——, "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*. Menlo Park, CA: AAAI Press, 1997, pp. 43–48.

[12] 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 *Data Mining and Decision Support: Integration and Collaboration*, D. Mladenic, N. Lavrac, M. Bohanec, and S. Moyle, Eds. Kluwer, 2003, pp. 81–90.

[13] J. Hernández-Orallo, C. Ferri, N. Lachiche, and P. Flach, Eds., *ROC Analysis in Artificial Intelligence, 1st International Workshop, ROCAI-2004, Valencia, Spain*, 2004.

[14] F. Tortorella, Ed., *Pattern Recognition Letters: Special Issue on ROC Analysis in Pattern Recognition*, vol. 26, 2006.

[15] R. Everson and J. Fieldsend, "Multi-class roc analysis from a multi-objective optimisation perspective," *Pattern Recognition Letters*, vol. 27, pp. 918–927, 2006.

[16] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[17] J. E. Fieldsend and R. M. Everson, "Multi-objective Optimisation in the Presence of Uncertainty," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC'05)*, 2005, pp. 476–483.

[18] E. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Evolutionary Multi-Criterion Optimization, EMO 2001*, ser. LNCS, E. Zitzler, K. Deb, L. Thiele, C. Coello Coello, and D. Corne, Eds., vol. 1993. Springer, 2001, pp. 329–342.

[19] B. Ripley, "Neural networks and related methods for classification (with discussion)," *Journal of the Royal Statistical Society Series B*, vol. 56, no. 3, pp. 409–456, 1994.

[20] I. T. Nabney, *"NETLAB: Algorithms for Pattern Recognition"*. Springer, 2002.