Parcel:
feature subset selection
in variable cost domains

M.J.J. Scott, M. Niranjan, R.W. Prager

CUED/F-INFENG/TR. 323

May 1998

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England
email: `mjjs@eng.cam.ac.uk`

Abstract

The vast majority of classification systems are designed with a single set of features, and optimised to a single specified cost. However, in examples such as medical and financial risk modelling, costs are known to vary subsequent to system design. In this paper, we present a design method for feature selection in the presence of varying costs.

Starting from the Wilcoxon nonparametric statistic for the performance of a classification system, we introduce a concept called the maximum realisable receiver operating characteristic ($MRROC$), and prove a related theorem. A novel criterion for feature selection, based on the area under the $MRROC$ curve, is then introduced. This leads to a framework which we call *Parcel*. This has the flexibility to use different combinations of features at different operating points on the resulting $MRROC$ curve. Empirical support for each stage in our approach is provided by experiments on real world problems, with *Parcel* achieving superior results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Question 1 *Why should we do feature selection?*

To construct a supervised classification system, one requires a data set of labelled examples with which to train and test the system. Each example is made up of a class label, *i.e.* what it is an example of, and a number of features, *i.e.* measurements or attributes describing the example. A classification system contains a model of how these features describe the different classes, and it is hoped that this model will allow novel examples, for which no label is available, to be correctly classified. If, for some reason, one does not wish to use all of the available features, then one must carry out feature selection: pick a subset of features from those available.

Why would we want to do this? Intuitively, the greater the number of features, the richer our classification models will become, and hence the better able to classify unknown examples. Theoretically,[1] this is indeed the case: classification accuracy increases monotonically with the number of features, or dimensions, in the data being modelled. In real world situations this is not the case: the classification accuracy can decrease as a result of having too many features. This effect is known as the *curse of dimensionality*. [2]

The *curse of dimensionality* can be caused by a number of factors. The probability distributions of the features must be modelled by the classification system. In the theoretical case, these distributions are known *a priori*. In practice this is rare, and so these distributions must be estimated from the available data. If we think of the examples as points in feature space, we can see why increasing the dimensions of the data can lead to problems in estimating probability

[1] In theory, there is no difference between *theory* and *practice*. In practice, there is.

[2] A number of definitions for the *curse of dimensionality*, exist, which differ in effect and cause; in this report, the *curse of dimensionality* will be used to refer to the decrease in performance of a classification system, observed when dimensionality increases above some point, regardless of the underlying cause.

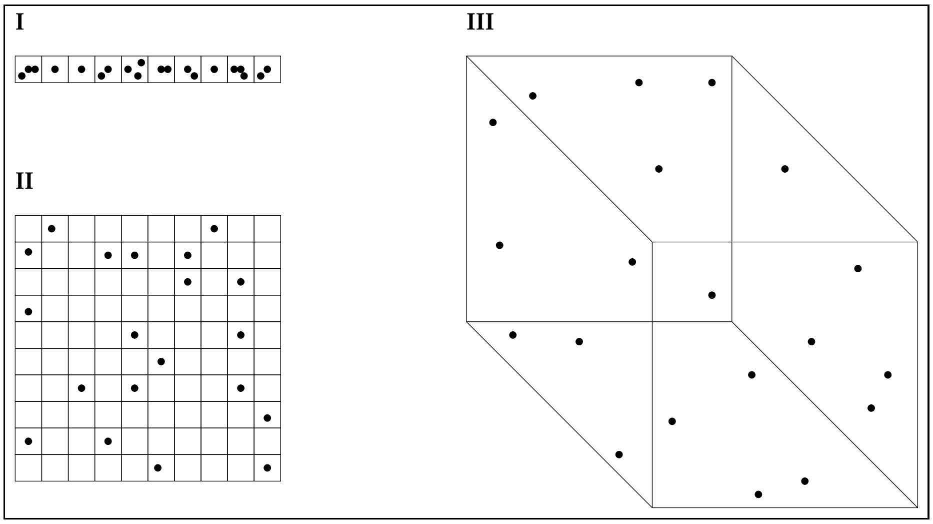


Figure 1.1: The effects of increasing dimensionality on a data set of twenty examples. **I.** With one ten-valued dimension, the data fills the ten location feature space rather well. Probability estimates made with this data, in this sized feature space, might be reasonably reliable. **II.** Two dimensions, and one hundred possible locations. The data is now more sparsely spread over the feature space. **III.**With three dimensions, and one thousand possible locations, the data fills the feature space poorly. Probability estimates made with this data, in this sized feature space, would be very unreliable.

distributions. As we increase the number of features, without increasing the number of examples, we create an ever more sparsely populated space. Take the example in Figure 1.1; if we had one feature, which could take on ten possible values, then having twenty examples might fill our feature space rather well, as there are only ten locations in it. If we were to then add another ten-valued feature, but no more examples, we see that we now have one hundred possible locations, and only twenty examples to fill it; add a third such feature, and the situation is worse again, twenty examples in one thousand locations. In fact, the requirement for data increases *exponentially* with the dimensionality of the feature space. If a feature space is sparsely populated, estimates of distributions in that space will be poor. The problems associated with estimation are compounded by the increased effects of noise; it is clear from Figure 1.1 that any noise present in the three dimensional space will have a much greater influence upon a probability estimate than noise in the one dimensional space.

The inclusion of useless, or irrelevant, features can also cause performance to decline. A number of classification algorithms are known to be adversely affected by inclusion of irrelevant features in the data. For example, Langley[64] and Aha[1] both observe that the volume of data required by nearest-neighbour algorithms increases exponentially in the presence of irrelevant features, without which performance degrades significantly; similarly, Caruana and Freitag[20] and Kohavi[57] note that decision trees also suffer degraded performance and increased complexity attributable to irrelevant features. Strongly correlated, or redundant, features can also be problematic. Once again, certain classification algorithms, such as the naive Bayes classifier, are adversely affected by redundancy in the features used to describe the data.

The factors contributing to the *curse of dimensionality*,

- sparsely populated feature spaces,
- noise,
- irrelevance, and
- redundancy,

are well known problems in the statistical pattern recognition and regression communities. These issues have been addressed for some time, and a number of solutions to the feature selection[3] problem have been developed. It is more recently that the machine learning and data mining communities have shown a strong interest in feature selection; the latter two factors, irrelevance, and redundancy, are of particular interest, as these cause pathologies in classification algorithms[4] that may not be encountered in some statistical domains.

With the widespread integration of information technology throughout industry and public sector institutions, the need for sophisticated data analysis has grown. Many businesses now hold large databases of customer information, and these businesses wish to use this data to make predictions about customer and market behaviour. Likewise, vast amounts of medical data are now stored digitally, and there exist significant potential benefits to society, if that data can be used for the accurate diagnosis of disease, and prediction of outcome.

These data sets may not, however, have been gathered with classification as an objective. Frequently, they are the product of a business initiative to simply computerise all records, the idea of using the data for classification being *a posteriori*. As such, along with many useful features, these data sets tend to have large numbers of noisy, irrelevant and redundant features. Compounding this, the data might not be interpretable by a system designer: the data might consist of infra-red energy readings from satellite images. These factors indicate the need to use some form of automated feature selection procedure when designing a classification system, so as to avoid the *curse of dimensionality*. The elimination of features has other, not insignificant, benefits: some features are costly to gather, and removal of these features without detriment to performance can make a classification system more attractive to an end user. For example, in a medical diagnosis system, the features might be measurements from blood tests and biopsies. In addition to cost and discomfort, there may be a risk associated with some of these features, such as the biopsy. If the biopsy features could be eliminated without reducing the accuracy of the diagnosis system, this would represent both a financial saving and a minimisation of patient risk. The motivation for feature selection might be summarised as so:

> "*General motivation*. We are given the task of designing a classification system, for which the available data have a large number of dimensions. We need to select an

[3] Known as variable selection in this field.
[4] Sometimes referred to as *induction* algorithms.

appropriate subset of features, from the large number available. Furthermore, we wish to select the smallest possible appropriate subset."

1.1.1 Variable costs

An issue not widely addressed in the feature selection literature is that of variable costs. The accuracy of a classification system simply reports how many of the test examples were correctly labelled. In many cases getting one class wrong is far worse that another. It is generally accepted that it is far worse to tell a sick patient he is healthy, than to tell a healthy patient he is sick: the latter might start treatment, but would soon be discovered to be healthy, the former, left untreated, might die. It is straightforward to assess the performance of a system given some specific trade-off, such as "it is ten times worse to misclassify a sick patient than a healthy one". A cost function can be computed with such a trade-off, and a system trained to minimise this cost.

This relies crucially upon an accurate assessment of this trade-off, and this can sometimes be difficult, or impossible, to obtain. In a number of environments, the users of a system would like to be able to set a cost trade-off dynamically; to have the freedom to alter the trade-off, perhaps in response to changes in operating circumstances, or in professional opinion as to the effects of misclassification.

Chapter 2

Data and methods

2.1 Introduction

In this report a number of feature selection algorithms will be described and empirically evaluated. Each empirical test will use the same methodology, employ one of two possible classification algorithms, and obtain results on one or more real world classification tasks; this chapter describes the experimental method, classification algorithms and classification tasks.

2.2 Experimental method

Feature selection algorithms are utilised during the design of a classification system. It follows that a reasonable method for comparing two feature selection algorithms is to build two classification systems, evaluate the performance of the two systems, and compare the results. In any practical implementation, only a finite amount of data is available, hence the performance of a classification system must be estimated. An unbiased estimate of performance can be obtained using a hold-out data set; that is, data that was not available during the construction of the classification system. Given two unbiased performance estimates, one can test the null hypothesis that a difference between the two occurred by chance, and is not indicative of a difference in the true system performances.

Adopting this methodology, each experiment in this report will use three data sets, as shown in Figure 2.1.

1. A *training* data set, used to train the classification system.

2. A *validation* data set, used to estimate performance during construction of the system; for example, during feature selection. This data set may be used multiple times during the training of the system.

3. A *hold-out* data set, unavailable during any part of the training of the classification system. This data is used once only, to give an unbiased estimate of performance.

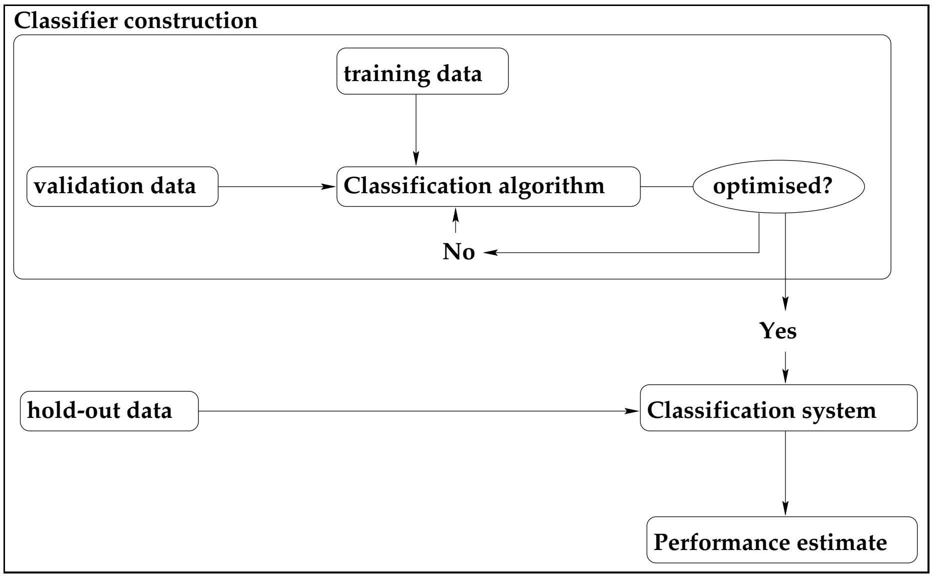


Figure 2.1: Obtaining unbiased estimates of classification system performance.

When comparing two feature selections algorithms, both use the same data sets, *i.e.* both are given the same resources for training, and tested on the same hold-out data.

2.3 Classification algorithms

Two classification algorithms were used in this report: a simple linear model, and naive Bayes. These algorithms were deliberately chosen for their simplicity. The experiments carried out in this report aim to demonstrate the effects of feature selection, and the differences in feature selection algorithms. These effects and differences are, in general, independent of the classification algorithm used[1]. Having simple classification algorithms removes a layer of complexity that might otherwise confuse the results obtained: if the algorithms behaviour is stable and well understood, then the effects of a feature selection algorithm might not be confused with some artifact of the classifier being used.

[1]However, some classification algorithms may be more robust than others to some of these effects.

2.3.1 Simple linear model

The implementation of a simple linear model by Rasmussen[95] was used in this report. The model is defined as

$$f_w(x) = \sum_{i=1}^{m+1} x_i w_i. \tag{2.1}$$

w_i, $i = 1, .., m+1$, are the model parameters, with w_{m+1} acting as the bias, x_i are the features (i.e. the model inputs), with an extra feature x_{m+1} for the bias parameter. The model is fit to the data by maximum likelihood, assuming zero mean Gaussian noise.

2.3.2 The naive Bayes classifier

The naive Bayes algorithm computes a discriminant function for each of n possible classes. Let E be an example vector, with a features $\{X_1, .., X_a\}$, and $B_i(E)$ the discriminant function corresponding to the ith class. The chosen class, C_k, is the one for which

$$B_k(E) > B_i(E) \forall i \neq k.$$

The discriminant function $B_i(E)$ is defined as

$$B_i(E) = Pr(Class = C_i) \prod_{j=1}^{a} Pr(X_j = v_j | Class = C_i),$$

where v_j is the value of feature X_j in example E.

The classification rule might be changed to reflect some desired operating conditions: in a two class problem the rule might be changed such that if one discriminant were above a given threshold, then that class would be assigned, regardless of the value of the other discriminant.

This classification algorithm is optimal for problems in which the features in the data are independent, see Michie *et al*[77] and Ripley[96], and this assumption is made during training and classification. As this assumption is rarely true, the algorithm has been given various (mildly derogatory) names such as *naive* and *stupid* Bayes. However, the interested reader is pointed to the recent article of [31], in which the regions of optimality for this algorithm are shown to be far greater than those indicated by the independence assumption. Indeed, empirical studies comparing the performance of naive Bayes to many more powerful algorithms such as $C4.5$ [93], on the real world datasets at the repository in UCI [72], indicate that naive Bayes is a much under rated algorithm, as it outperforms $C4.5$ on 70% of the datasets examined.

2.4 Data

Seven real world classification tasks were used to test the algorithms presented in this report. Some of these algorithms address two-class classification tasks, hence, for consistency and to allow all results to be comparable, two-class tasks were used throughout. As detailed in the experimental methodology, there were three data sets for each of the seven problems: training, validation and hold-out. Table 2.1 gives the number of cases in each data set; each data set was formed by randomly partitioning the original data.

Classification task	training set	validation set	hold-out set
Adult	20108	10054	15060
BrodSat	1206	603	500
Cotton	2956	1478	2000
Field	6000	3000	3392
Grey	2956	1478	2000
Thyroid	2514	1257	3413
Tree	6000	3000	3392

Table 2.1: The data sets for the seven classification tasks examined in this report; the number of cases in each data set are given.

In experiments using the naive Bayes classifier, discrete data sets are required. The `discretize` software, see Kohavi *et al*[60], was used to discretise continuous data. This implements entropy based discretisation, described in Dougherty *et al*[32]. Only default settings were used. For each classification task, the hold-out data set was discretised using parameters estimated with the training and validation data alone, thus avoiding the introduction of an experimental bias.

Adult This two-class problem originated from the US Census Bureau, 1994 Census database, and involves predicting whether a person will have a salary of greater or less than $\$50,000$. In total, it contains $45,222$ labelled examples, with fourteen features. This data is held at the UCI repository[72].

BrodSat This problem was originally a seven-class problem, with target classes: brickface, sky, foliage, cement, window, path, and grass. It was transformed to a two-class problem, man-made and natural, by combining brickface, cement, window, and path, to form man-made, and combining sky, foliage, and path, to form natural. The data was created from a satellite image classification problem by Carla Brodley at the Vision Group, University of Massachusetts. In total there are 2309 examples, with nineteen features. This data is held at the UCI repository[72].

Cotton and Grey Both of these classification problems were formed from a multi-class LandSat image classification problem. The problem was originally used by King *et al*[53] and Michie *et al*[77] in the Statlog project. There are a total of 6434 examples with thirty six features. Initially there were six classes: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil.

Combining grey soil, damp grey soil, and very damp grey soil, into the grey class, and the remainder into a class called other, gives the Grey two-class problem. Likewise, combining cotton crop and soil with vegetation stubble into the cotton class, and the remaining classes into other, gives the Cotton two-class problem. This data is held at the UCI repository[72].

Field and Tree These two problems were formed from a geological remote sensing data set, used by Bailey *et al*[7], and held at the Image Processing and Neural Networks Laboratory University of Texas at Arlington. It has four classes: urban, fields, trees, and water. The Field problem has two classes, one combining field and urban, the other combining. The Tree problem has keeps the class tree, and combines the remainder into a class named other. In total there are $12,392$ examples in the data, with eighteen features.

Thyroid The data for this three-class problem was originally used by Schiffmann *et al*[97] to train and test neural network classifiers. The three classes were: normal, hypo-thyroid, and hyper-thyroid. The two-class problem in this report has classes abnormal and normal. There are a total of 7184 examples, with twenty one features. This data is held at the UCI repository[72].

Chapter 3

Literature review

3.1 Introduction

Question 2 *What methods of feature selection have others developed?*

Currently a topic of much interest in the machine learning (see Bradley[15], Langley[64], Michalski *et al*[76] and Morik[80]) and data mining communities (see Craven and Shavlik[25], Glymour *et al*[41], Fayyad and Smyth[36] and Fayyad *et al*[35]), feature selection has been studied widely in the pattern recognition and statistics literature. Reviews of this problem domain and the solutions available are given in Devijer and Kittler[29], Kittler[56], and Wyse *et al*[109]; many of the algorithms described in these papers are currently applied in the machine learning literature, although in some cases the algorithms make assumptions that are not valid in this domain, as noted by John *et al*[52]. Recent comparative studies of these and more recent algorithms, applied to machine learning, include those carried out by Dash and Liu[26], Gordon and desJardins[42], Langley [63], Siedlecki and Sklansky[100], Jain and Zongker[50] and Kohavi and John[59], the last of which is referenced in detail in subsequent chapters of this report.

In general, feature subset selection algorithms have two components: an evaluation function $J(\cdot)$, which "scores" candidate feature sets, and a search engine for finding those sets. Given a set of features the selection algorithm will examine a series of sets of features, and choose the one that maximises $J(\cdot)$; in the event of multiple feature sets maximising $J(\cdot)$, the smallest one is chosen.

When examining the current state of the art, one finds that feature selection algorithms fall broadly into two different frameworks, *wrappers* and *filters*, this categorisation being determined by the nature of $J(\cdot)$.

Relevant work regarding wrappers and filters is described next. In section 3.3, the state of the are concerning feature selection search engines is discussed. Section 3.4 contains details of additional feature selection algorithms that could be categorised as either filters or wrappers, but are either sufficiently distinct conceptually or specific to a particular type of classification algorithm, to warrant a separate description.

3.2 Filters and wrappers

Langley[63] defines a taxonomy of two types of feature selection framework, derived from the nature of the evaluation function $J(\cdot)$ used: *filters* and *wrappers*. In a filter framework, $J(\cdot)$ measures the performance of a feature set in a manner that does not include the classification algorithm which will eventually use the features. In the classifier design process, filters carry out feature selection prior to classifier optimisation (training). In a wrapper framework, $J(\cdot)$ incorporates the classification algorithm. As such, in a wrappers framework, feature selection and optimisation are carried out in a single step during system design.

3.2.1 Filters

A *filter* is defined as a feature selection algorithm using a performance metric based entirely on the training data, without reference to the classifier for which the features are to be selected. The named is derived from the way in which the features are *filtered* before the classification system is trained and tested. Two types of filter measures will be discussed: statistical dependence and inter-class distance.

3.2.1.1 Statistical dependence

The features used to describe a data set can be thought of as random variables, with some distribution that can be estimated from the data. Dependence metrics seek to estimate the statistical dependence of an output variable on a set of features. Metrics based on the mutual information (see Cover and Thomas[24]) between the features and the target class labels have been developed by Battiti[9] and Bonnlander[13].

This is appealing theoretically, as mutual information offers a powerful measure of the statistical dependences between the data and the class labels. It is these dependencies that a classifier attempts to model when mapping from inputs to outputs. However, the accurate estimation of mutual information from continuous valued data is difficult, and requires an extremely large amount of data for even small feature dimensions (> 6), which has been noted by Bonnlander[13, 14] and, in the context of image registration, Viola[107, 108].

Bonnlander[13] evaluates candidate feature sets by estimating the mutual information between the current feature set and the target output of a multi-layer perceptron. To do so, a non-parametric, Parzen window density estimation technique (see Venables and Ripley[106]) is employed to model the joint probability density function between the feature and target variables.

It could be argued that, once the probability function has been modelled, the use of a neural network is redundant, and might even be detrimental to the design of a classification system. If the goal is to model this density, and that has already been done using the Parzen window estimation technique, why do so again with the neural network? A counter to this could be that the network might provide a more concise model of the density, and so reduce the computational overheads associated with the Parzen window, which requires the storage of all the training data; as noted, if accurate estimates are sought, this might prove to be extremely large. However, this is not addressed by Bonnlander in [13]. If the Parzen density estimate were kept and used for classification, then this technique for feature selection would belong in the wrapper framework.

3.2.1.2 Distance

A number of measures exist that measure the inter-class separation, or distance, produced by a subset of features. Logically, the larger the separation between classes, the easier it will be to define a decision boundary, and to achieve a lower error rate on novel data. Methods such as the Mahalanobis distance, see Devijer and Kittler[29] and Duda and Hart[34], make Gaussian assumptions about the data. Others, such as the Bahttacharyya distance do not make Gaussian assumptions, see Kittler[56] and Ripley[96].

These distance measurement arise in the statistics literature. Despite apparently producing large inter-class distances, some feature sets may subsequently produce poor classifiers, see Jain and Zongker[50]. This is symptomatic of the *curse of dimensionality*, as described by Bishop[12], Duda and Hart[34], Hand[45] and Jain and Zongker[50]. As the dimensionality (*i.e. number of features*) of a data set increases, the resulting of estimates of inter-class distance become less reliable. These distance estimates monotonically increase with the dimensions of the data, but only for that data set: these estimates may not generalise to the novel data used in subsequent classification tests. Despite this weakness, these measure have been widely used. A good reason for this is that the majority of distance measures are cheap to compute, especially when compared to the computational cost of training and testing a classifier. Therefore, in times when computing power was expensive, these methods held an obvious attraction; as computing power is now cheap, a major justification for using such measures has been removed.

3.2.2 Wrappers

It is a general weakness of filter frameworks that feature subsets may rate highly, even when they are inappropriate to the classification algorithm being used. This weakness provides a compelling argument for the inclusion of the classification algorithm in the performance metric $J(\cdot)$, and has lead to the development of *wrapper* algorithms. The name is derived from the notion that the feature selection algorithm is inextricable from the end classification system, and is *wrapped* around it.

Although the term wrapper was coined recently, the logical idea of using the performance of an actual classifier in feature selection is not new. Ben-Basset[10] noted that measures such as the Bahttacharyya distance do not, over arbitrary feature subsets drawn from the same original features, induce the same order of preference as that obtained by comparing the errors of the Bayes classifier. This evidence, coupled with the reasons detailed above, has lead many researchers, such as Siedlecki and Sklansky [100] to conclude:

> "it seems that the only promising and legitimate way of evaluating features must be through the error rate of the classifier being designed."

Consequently, increasing interest is falling upon feature selection algorithms that are classifier inclusive, i.e. that use the test performance of a classifier as $J(\cdot)$. In this case $J(\cdot)$ is evaluated as the expected loss, of the classifier trained and tested using the feature set in question. If zero-one loss, see Friedman[40], is used, then this is equivalent to the error rate, or $1-$accuracy, of the classifier.

A number of papers present results indicating that, on real world datasets, wrapper algorithms that carry out subset selection can significantly increase the performance of the classification systems being designed. Lovell *et al* [65, 66, 67] have extensively studied the feature selection problem in the context of a large obstetrics risk prediction problem, involving over $700,000$ patient histories, making novel use of the area under the ROC curve (receiver operating characteristic, see Chapter 5) used as a classifier inclusive metric. Lovell found that the estimates of patient risk produced by systems that had used feature selection were significantly better than those that had not. Other recent work reporting the successful use of wrapper algorithms on real world problems include Kohavi[57], Doak[30], Aha and Bankert[3] and Mladenic[79].

Kohavi and John[59] have recently published a large empirical study of wrapper algorithms, using a number of artificial and real world datasets. The classification accuracy of the $C4.5$ decision tree (see Quinlan[93]) and naive Bayes classifiers (see Chapter 2) was estimated on a wide range of problems, using both the total set of features for each problem and a subset selected by a wrapper. They conclude that feature subset selection significantly improved the accuracy of the classification systems produced.

3.3 Search engines

Regardless of the choice of evaluation function $J(\cdot)$, *i.e.* whether it is a wrapper or a filter, an algorithm will require a method of generating candidate feature sets. The solution space of all possible feature subsets can be viewed as a lattice; Figure 3.1 illustrates the lattice for a problem with four features. Each node represents a feature set, a one indicates feature inclusion, a zero exclusion. Hence, the operation of a feature selection algorithm can be seen as a search of this lattice, seeking the node, S, such that $J(S)$ is maximised; this is a common view, adopted in, among others, Siedlecki and Sklansky[100], Jain and Dubes[49], Davies and Russell[27] and Kohavi and John[59].

From a node representing a feature set, the children of the node can be reached by inclusion or exclusion of individual features from the set, and by repeated actions, any node in the lattice may be reached from any start point. The manner by which the lattice is traversed, the inclusion and exclusion operators, defines the search engine of the feature selection algorithm.

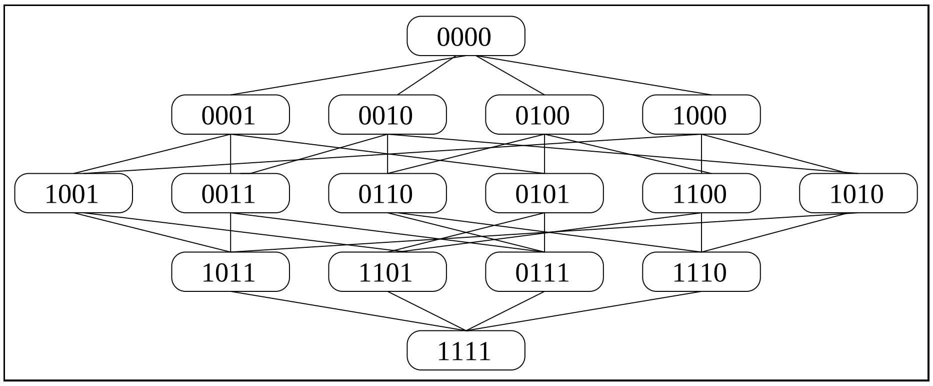


Figure 3.1: The solution space of the feature subset selection problem can be viewed as a graph or lattice.

The lattice for a problem with n features has 2^n nodes. Due to the exponentially explosive nature of this search space, exhaustive search of the lattice is not a feasible solution for real world problems. As a result, a number of sub-optimal search strategies from the artificial intelligence literature have been applied to feature selection.

The most simple searches, widely used in both filter and wrapper frame works, are forwards selection (FS) and backwards elimination (BE). FS starts with an empty feature set and has an inclusion operator, adding one feature at a time, attempting to maximise $J(\cdot)$; in contrast BE starts with a set of all the available features, and uses only an exclusion operator. These methods have been generalised to *stepwise forwards selection* (SFS) and *stepwise backwards elimination* (SBE), wherein both use inclusion and exclusion operators, see Devijer and Kittler[29]; it is the convention that a search starting with an empty set is moving "forwards", and one starting with a full set "backwards". These forms of searches appear in the statistics and regression literature as variable selection algorithms, for examples see Draper and Smith[33], Peduzzi *et al*[86], and Stark and Fitzgerald[102].

Numerous applications have used these types of search engines, including Bonnlander[13], Caruana and Freitag[20], John *et al*[52], Kittler[55, 56], Lovell *et al*[67]. Bonnlander uses FS exclusively, as the number of features used in BE render the required probability estimations computationally infeasible given the Parzen estimation method employed. Similarly, researchers using wrapper algorithms report a high complexity associated with backwards searches, which can make them less attractive than their forwards counterparts, for example Kohavi and John[59] and Langley[63]. Aha and Bankert[2, 3]applied wrapper feature selection algorithms to the problem of cloud classification, and empirically compared forwards and backwards searches, concluding that forwards search was superior.

The only possible search engine that guarantees an optimal solution is *branch and bound* . Theoretically, an exhaustive search of the lattice is implicitly carried out using the branch and bound algorithm and, therefore, this strategy should always select an optimal feature set; descriptions, applications, and comparisons of this algorithm to others are to be found in Hand[44], Jain and Zongker[50], Foroutan and Sklansky[39], Narendra and Fukunaga[83], Ripley[96], and Yu and Yuan[110].

However, branch and bound makes the assumption that $J(S)$ monotonically increases with the dimensionality of the data set; this assumption is a necessary condition for the implicit exhaustive search. Branch and bound usually employs a distance metric, such as the Mahalanobis distance, for $J(\cdot)$. As we have seen, the Mahalanobis distance and other similar distance measures, are indeed monotonic within one data set, but in practice suffer from the *curse of dimensionality*, and do not behave monotonically on novel data sets. Even if the monotonicity assumption is made, the search space can still be exponential, and for problems with more than thirty features this can prove an infeasible search. In a case particularly relevant to machine learning, Siedlecki and Sklansky[100] demonstrate that the monotonicity requirement does not hold for error rates, and note that, in general, the monotonicity requirements of *branch and bound* cannot be assumed.

Without the monotonicity requirement of branch and bound, all sequential search algorithms that do not perform an exhaustive search are sub-optimal, as shown by Cover and Campenhout[23] for hill-climbing searches. Despite the sub-optimal nature of feature selection, many algorithms exist that significantly improve the performance of classification systems by selecting an appropriate subset of features from those available, and in doing so justify the investigation and application of feature selection algorithms.

Recent advances in search strategies include the development of the *sequential forwards floating selection*($SFFS$) and *sequential backwards floating elimination* ($SBFE$) algorithms by Pudil *et al*[90]. Genetic algorithms, indicated by Siedlecki and Sklansky[100] as a direction of future research, have been explored by Punch *et al*[91]. They used a combination of a genetic algorithm and a k nearest-neighbours algorithm, hence a wrapper; the computational overhead of this approach required a parallel implementation to make it attractive, and so does not yet offer a widely applicable solution to the feature selection problem.

In the recent empirical study by Jain and Zongker [50], fifteen search algorithms were compared empirically; $SFFS$, SFS, and FS, three search algorithms examined in this report, as well $SFBE$ and a genetic algorithm, were amongst the fifteen. A filter framework was used, using the Mahalanobis distance for the evaluation function $J(\cdot)$, and both artificial data and a satellite image classification problem. It was concluded that $SFFS$ was the best search algorithm of those tested. An empirical comparison of $SFFS$ to FS was reported by Pudil *et al*[88]; a wrapper method was employed, utilising the classification accuracy of a Gaussian classifier, and two real world problems were examined. Pudil *et al* concluded that $SFFS$ was the best algorithm tested, based on the results obtained. However, it appears that the classification accuracy was estimated on data that was also used *during* feature selection. A number of papers have reported classification results in feature selection that were not calculated with hold-out data, and as such are optimistically biased and should be interpreted with caution; as a result, the empirical evidence reported in Pudil *et al* may be neither statistically significant, nor conclusive. Kohavi[57] points to results produced by Doak[30], Aha and Bankert[3] and Mladenic[79], as examples of biased results, that should be treated with similar caution.

3.4 Alternative approaches

Recently, work in the artificial intelligence community has been carried out on algorithms that attempt to select features based on relevance determination. If a feature is important, in some sense, to modelling a concept or target function correctly, it is said to be relevant.

Some interest has developed in algorithms, *Relief* and $FOCUS$, that attempt to eliminate irrelevant features prior to classifier training; as such these algorithms fall into the filter framework. In the neural network community, an attempt to detect suitable features through the effects they produce on weights during training has lead to algorithms for relevance detection; as this inherently uses the neural network classifier in evaluating the feature set, such algorithms would belong in the wrapper framework.

The use of decision trees, as either wrappers or filters, has attracted a wealth of attention in the data mining community. Seen as more interpretable and "user friendly" than other classifiers, such as neural networks, much application-oriented research has been undertaken, with the aim of creating easy-to-use, interactive data mining tools for end users; decision tree packages available in the commercial sector include $CART$, see Steinberg[103], and $MineSet$, see Bunk *et al*[18] and Kohavi[58]. The commercial success of these and other systems, such as $PREDITAS$[89], indicates a large commercial market for feature selection tools.

3.4.1 *Relief* and *FOCUS*

Kira and Rendell[54] and Kononenko[61] examine the *Relief* algorithm, in which a relevance weighting is attributed to each feature in the dataset. A drawback of this approach is that it does not discriminate between two features that may be redundant with respect to each other, i.e. if one were identical to the other, both would have the same relevance. Selection on this basis will select both, which would not be beneficial, and would in some cases be detrimental, to a classification algorithm: no benefit is gained as both carry the same information, harm is done because classifier parameter estimation is carried out in a higher dimensional feature space than necessary.

Kohavi and John[59] report problems associated with variance in relevance estimates from the original algorithm. They have implemented an adaption, *Relieved*, that does not suffer from the variance in relevance estimates, but still selects redundant features. Caruana and Freitag[21] compared the accuracy of a series of $C4.5$ decision tree classifiers that had used either wrapper feature selection algorithms or one of two relevance selection algorithms, *Relief* or $FOCUS$ (see Almuallim and Dietterich[4, 5, 6]).

Caruana and Feitag found that both algorithms selected features that, while qualifying as relevant, were not useful for classification. They found that, for the problems considered, the wrapper algorithms produced superior classifiers, and concluded

> "it is not clear that one needs to resort to proxy measures like relevance to perform attribute selection."

Davies and Russell[27] considered the $FOCUS$ algorithm, and concluded that, although good results can be obtained with $FOCUS$, the search for relevant features in some real world problems is intractable. A number of assumptions and adaptations are required to make $FOCUS$ useful in a practical setting. Even with such assumptions, Davies and Russell, with reference to John *et al*[52], note that the use of wrapper algorithms may be more appropriate than relevance algorithms such as $FOCUS$.

3.4.2 Neural networks

Automatic relevance determination (ARD) (see Mackay[68, 69] and a recent empirical review by Neal[84]) is a wrapper algorithm, applicable to multi layer perceptron[68, 69] classification systems. Relevance weightings are produced for each of the inputs to the network by examining the parameters of the network during training.

No features are ever completely excluded, so ARD is not strictly a feature selection algorithm; it always uses all the features available. However, ARD can be said to be performing a form of *soft* selection, see Neal[84], due to the application of relevance weightings. Neal concludes that for neural network architectures, *soft* feature selection (of the kind carried out by ARD) is often helpful. Unlike the other *wrapper* algorithms considered here, ARD is not generally applicable: it is specifically designed for neural network architectures.

Recently Messer *et al*[73, 74, 75] have developed a feature selection algorithm that also selects features based the effects observed in the weights of a training neural network. This has been successfully applied this to an image retrieval problem domain. An empirical comparison with a filter, using floating forwards search[90] and a statistical criterion for $J(\cdot)$, led Messer *et al* to conclude that the weight analysis method offered "an attractive alternative to the statistical method".

3.4.3 Decision trees

The use of decision trees such as CART[16] and $C4.5$[94] perform feature selection as part of the construction process. There is no constraint on such algorithms to use all of the available features, and in many cases a subset will be used. The tree algorithm selects a feature based on that feature producing the best split of the training data, in terms of some training objective function. This objective might take the form of an impurity measure, such as the Gini index employed by CART, or an information theoretic measure such as those used in $C4.5$.

There has been a recent surge of interest in the field of decision trees, from both the data-mining and machine learning communities. The inherent feature selection carried out during tree construct indicates useful or important features; hence such algorithms are used for the purposes of knowledge discovery and data-mining in large commercial databases. However, results reported by Kohavi and John[59] indicate that the results of $C4.5$ are improved by applying a wrapper feature selection algorithm during classifier design. Ongoing research in the decision tree literature seeks improved methods of selecting features to split the training data, *i.e.* feature selection; see Brodley[17], Fisher[38], Lopez de Mantaras[70], Mingers[78] and Murthy*et al*[81, 82].

Decision trees can be classed as wrappers if the constructed tree is used for classification, or as filters, if the tree is used to select features that will subsequently be used for another algorithm, for example Cardie[19] uses decision trees to select the features for a case-based learning algorithm. Cherkauer and Shavlik[22] use the $ID3$ algorithm (the public domain pre-cursor to $C4.5$, see Quinlan[92, 93] and Quinlan and Cameron-Jones[94]) to indicate features that form good representations of target concepts. These features are subsequently used as the inputs to a neural network algorithm. The construction and optimisation of one algorithm to suggest features for another is similar, conceptually, to the work of Bonnlander[13]. Cherkauer and Shavlik justify this filter method by arguing that the neural network algorithms that are eventually used are computationally complex; $ID3$ can be trained and tested more efficiently, and so is appropriate to use as a filter.

Chapter 4

Classification in variable cost domains

4.1 Introduction

In the previous chapter, the feature selection literature was reviewed. From the recent findings of Kohavi and John[59], we conclude that wrapper feature selection algorithms present a useful method by which feature selection may be carried out. Given the motivation of this report, detailed in Chapter 1, we shall pose three questions:

Question 3 *Is it likely that variable misclassification costs might be encountered in practice?*

Question 4 *What are the implications of variable costs when carrying out feature subset selection?*

Question 5 *Is it possible, using the wrapper algorithms we have seen, to find a single feature set that that performs well across multiple costs?*

Real world examples will be provided, showing that variable costs are indeed encountered in practice. For the designer of a classification system, the implication of this is that existing wrapper algorithms can actually reject a feature subset that produces the best classification system, for some set of misclassification costs. If this set of costs contains the operating costs of the system (chosen after implementation), then the best available system is *not* being provided to the end-user.

Before giving these examples and empirical results, I will discuss variable costs, and how one might assess the performance of a classification system in a variable cost domain. In the next section, I describe the receiver operating characteristic (ROC) curve, which can graphically summarise performance over the complete range of possible operating costs. Furthermore, an ROC allows a designer, or user, to use problem specific criteria to choose a suitable threshold with which to perform classification, and so tailor a system to suit individual needs.

4.2 The receiver operating characteristic

A large fraction of decision support systems, particularly those used in medical diagnostics (e.g. diagnosis of cancer with digital mamography), are two-class pattern classification systems. Once a set of features and the functional form of the classifier have been chosen, the classifier is designed to optimise some cost function. When the costs of the different types of errors can be specified exactly, the optimum classifier may be designed to minimise the expected risk [34]. The particular feature set and the functional form chosen then define how well the performance of the classifier approaches the Bayes' performance.

It is often the case, in many real world applications, that the cost of different types of errors is not known at the time of designing the classifier. One also finds applications where the costs change over time. Further, some costs cannot be specified quantitatively. In such situations we resort to specifying the classifier in the form of an adjustable threshold and a receiver operating characteristic (ROC) curve obtained by setting the threshold to various possible values. An example of such an ROC curve is shown in Figure 4.1. In the example, the classifier must

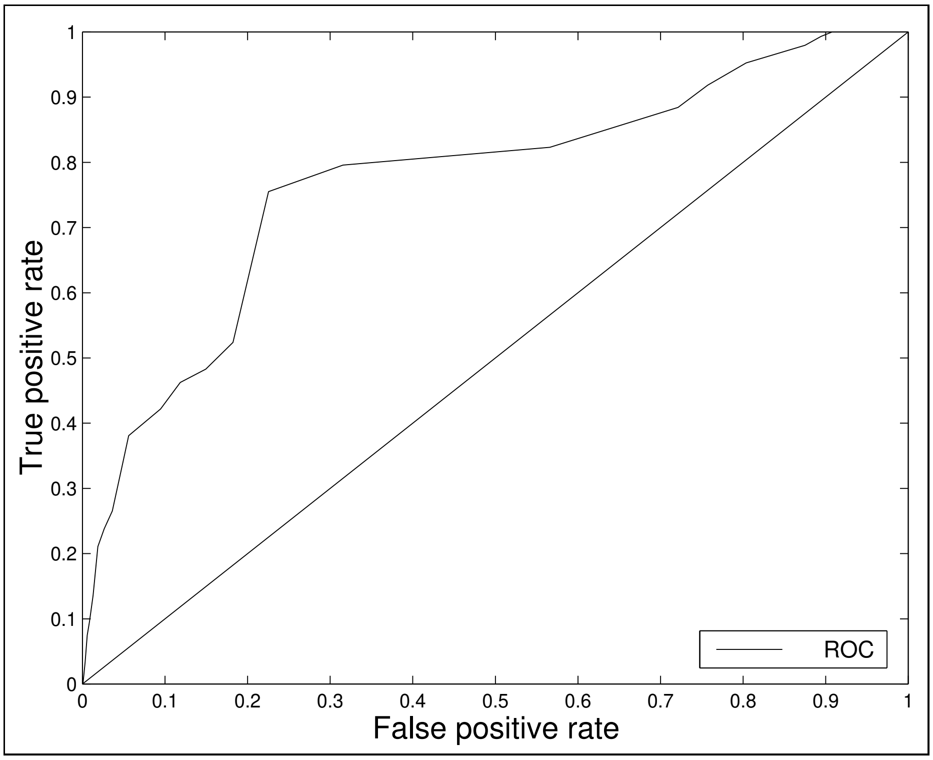


Figure 4.1: An ROC curve for a medical diagnostic test for abnormal thyroid condition. The true positive rate corresponds to the probability that a sick patient will be diagnosed as sick, the false positive to the probability that a healthy patient will be diagnosed as ill.

classify a patient's condition as either adverse or benign. The data in the example was obtained from the UCI Machine Learning repository (see Chapter 2 and [72]), and represents the results of a number of diagnostic tests for abnormal thyroid conditions. A simple linear classification algorithm (see Chapter 2) was used, producing a continuous output, and a threshold is placed upon the output to determine the final classification.

Two rates can be calculated for any series of classifications: the true-positive and false-positive rates. When an adverse case is *correctly* classified as adverse, a true-positive has occurred, and a false-positive when a benign case is *incorrectly* classified as adverse. By varying the level of the threshold, different degrees of true-positive and false-positive rates can be achieved, producing the ROC.

Each point on the ROC curve represents a classifier, formed by setting the threshold to an appropriate level. Given the ROC curve for a classification system, an end user can pick a point on the curve, that represents an operating classifier with the most desirable true- and false-positive rates. The manner in which this point is chosen may vary, but the result is the same: a point on the curve indicates which threshold should be applied during classification.

Early work on analysis of systems using the ROC was carried out on signal detection for military applications, see Swets [104], Jerison [51] and Hatfield [48]. In recent years an increasing number of researchers in the medical diagnosis/risk prediction domain have applied ROC techniques, for example see the obstetrics risk prediction work of Lovell *et al* [65, 67, 66] and the modelling of rejection risk in liver transplant patients studied by Melvin [71]. A plenary text book on the matter was written by Swets and Pickett [105], which has without doubt contributed to the propagation of ROC analysis in cost variable domains.

The correct interpretation and analysis of summary statistics such as the area under the ROC have been examined in a number of papers. This is equivalent to the Wilcoxon (or Mann-Whitney) statistic [28, 45, 46, 47, 67]. If X and Y are two sets of continuous observations, for example a diagnostic test on sick and healthy patients respectively, then the Wilcoxon statistic indicates the probability that a randomly drawn pair (x, y), $x \in X, y \in Y$, will be ordered correctly in terms of risk (i.e. $x > y$) by this classification system. As this metric is divorced from any single set of misclassification costs, it would seem a natural choice when analysing systems for which costs may be unspecified or variable. It has been proposed that the $AUROC$ could be used in a *wrapper* type feature selection algorithm. Lovell *et al*[67, 66] use this statistic as a criterion for feature selection in a large obstetrics problem involving 48 features and $700,000$ cases.

Hanley and McNeil [46, 47] offer techniques for non parametric summaries of the ROC, given certain assumptions about class distributions, and DeLong *et al* [28] have offered a fully non parametric estimate of the area under the ROC. In a real world situation, one might need to choose between candidate classification systems. This might occur if a novel classification algorithm were applied to an existing problem, for which there already existed a classification system. Given the novel system, it might fall to a designer to decide which system was superior. In many case studies reported in the literature, this decision is made on the basis of the $AUROC$, the system with the larger area being chosen.

The $AUROC$, however, is a gross simplification of the information conveyed by a ROC, as noted by Hand in [45]. The costs of different operating points need to be taken into consideration. Hand further suggests this to be an important factor when the ROC curves of the classification systems that are being considered cross. Experimental results in this chapter will show that the ROC with the largest $AUROC$ is not necessarily dominant across all operating conditions, supporting the cautionary stance adopted by Hand.

Recently, some focus has shifted to the analysis of the ROC based on realistic or probable operating costs and conditions, as suggested by Hand[45]. Halpern *et al* [43] propose the analysis of ROC curves through optimal operating points,[1] and iso-cost[2] lines tangential to the ROC. The utility of ROC analysis for cost variable classification problems is starting to be realised by the pattern recognition and machine learning communities. Provost and Fawcett [87] have proposed the use of ROC analysis in classification problems beyond medical decision making, as many real world problems exhibit variable cost. Independently of Halpern *et al* they also suggest the analysis of ROC data by iso-cost lines and optimal operating points, pointing out that the set of tangential iso-cost lines forms a convex hull on the ROC.

4.2.1 Neyman Pearson criterion

Once a classification system has been designed (and possibly selected from a range of systems), one must select an operating point from the ROC curve. There are a number of ways to do this, including methods that pick points such that a cost function is minimised, as described in the text by Hand [45]. However, such methods require a precise definition of the costs associated with misclassification; for example, one may have to specify that it is exactly twenty times worse to misclassify a sick patient than it is to misclassify a healthy one. This is not always desirable, nor is it always possible.

In many systems being designed today, it is specified that the user be able to choose operating points in a flexible, or even subjective, manner, guided by practical or contractual constraints. The user may not be able, nor wish, to express the misclassification costs in a precise fashion, but may instead wish to pick an operating point based on the specific true- and false- positive rates that this would yield.

In such a situation, a Neyman Pearson criterion might be employed: a maximum allowable false-positive rate is decided upon by the user, the selected system must have a false-positive rate less than this. Once a Neyman Pearson criterion is decided, a point on the ROC curve is then chosen, this having the highest true-positive rate with a false-positive rate less than or equal to the maximum allowable. Figure 4.2 illustrates a Neyman Pearson criterion.

[1] An optimal operating point is one that minimises the expected cost of the classifier with respect to a specific set of misclassification costs.

[2] This is a straight line, whose slope is determined by a cost function, comprised of a set of misclassification costs and prior class probabilities. All classifiers on this straight line have the same overall expected cost with respect to the cost function, despite having different true- and false-positive rates.

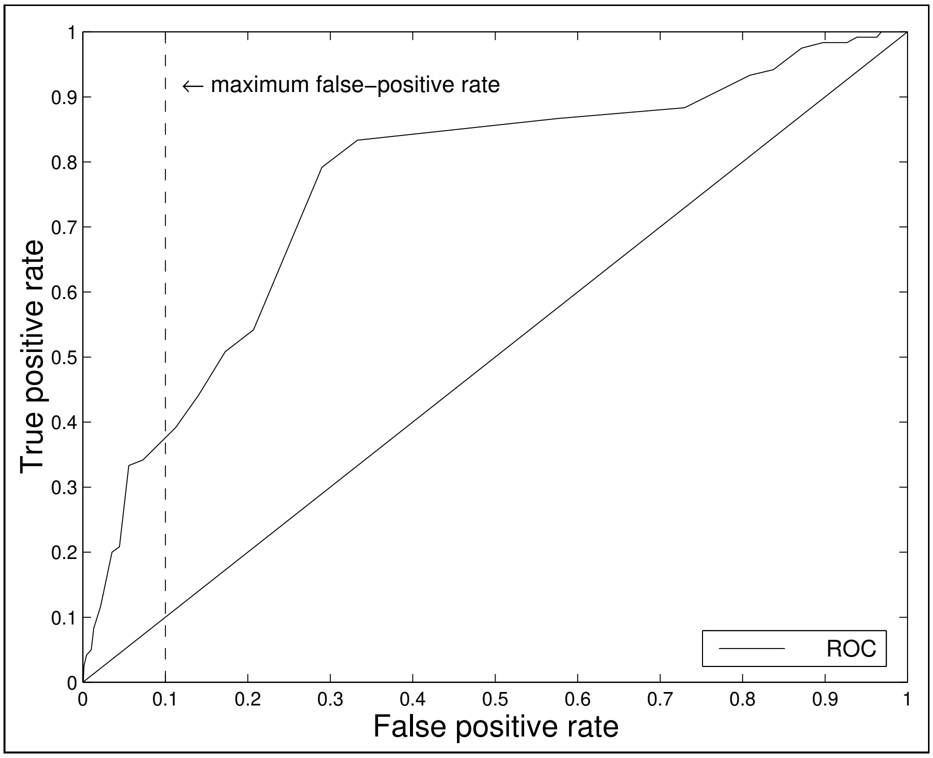


Figure 4.2: An example of a Neyman Pearson criterion. A maximum false-positive rate of 0.1 is set, describing a vertical line in ROC space. The point on the ROC curve that intersects this line indicates a threshold that should be applied to the classification system to achieve the desired true- and false-positive rates.

4.2.2 Design examples with variable operating conditions

The selection of an appropriate threshold and classifier from an ROC curve does not necessarily occur once, at the end of the design process. There are a number of problem domains within which the end user will want to be able to vary the choice of threshold, and so control the true and false-positive rates of the classifier dynamically, after it has been designed. Given the ROC curve of a classification system, and appropriate software, this can be easily achieved: the user simply enters a desired maximum false-positive rate, and the threshold used for classification can be updated dynamically. Although it is possible to envisage a number of scenarios where this might occur, it may be useful to give two real world examples of this, and therefore provide an affirmative answer to

Question 3 *Is it likely that variable misclassification costs might be encountered in practice?*

Predicting liver rejection. Melvin *et al* [71] have carried out extensive work designing a classification system for predicting whether a patient, who has recently undergone a liver transplant, is in an early phase of organ rejection. A patient classified as "early reject" is treated to prevent the onset of a full rejection episode.

The clinicians involved did not want to have to fix a specific set of misclassification costs. Rather, it was decided that they should be supplied with the ROC curve of a diagnostics system, and be given the ability to set various Neyman Pearson criteria. Furthermore it should be possible for them to alter these criteria at will, thereby varying the classification threshold. It was envisaged that variation in criteria might occur due to subjective differences in the operating conditions favoured by different clinicians, and also through changes in financial constraints placed upon the institution concerned.

Detecting oil spills in satellite images. In a recent paper, Kubat *et al* [62], describe the design and implementation of a system for detecting oil spills in satellite radar images. The problem involved early detection of oil spills at sea, thus enabling action to be taken to prevent possible ecological damage from pollution. However, there was a considerable cost involved in sending aircraft to verify that an oil spill existed, hence a trade off resulted between the benefits of maximising early detection and the cost of a false positive. A classification system was produced, and described by an ROC, from which a user could select a desired operating point.

> " We decided that in the version of the system that will be delivered to end users there will not be a preprogrammed way of condensing the ROC curve to a single performance measure. Instead, the user will be able to move along the curve and choose the point that best meets his/her current needs. This is typical of fielded systems. The user needs to be able to tune the system's behaviour so as to trade off various conflicting needs." Kubat, Holte and Matwin[62].

4.3 Wrappers and variable cost problems

We have seen specific design examples where the operating costs of a classification system are not known at the time of system design. Despite not knowing the eventual operating costs, feature selection and classifier training must still be carried out. In this situation, feature selection is normally executed with reference to a single fixed cost function, typically classification accuracy.[3] Subsequent to feature selection and training, such systems may operate in variable cost domains, such as those described in the previous section. The second of the questions posed at the start of this chapter is now considered:

Question 4 *What are the implications of variable costs when carrying out feature subset selection with the wrapper algorithms?*

A wrapper will select a single feature set to use in the construction of a classification system. I will show empirically that the ROC curve of the classification system produced using such a

[3] Also called error rate and zero-one loss.

feature set is not dominant across all operating conditions. Furthermore, I will demonstrate that a wrapper may reject feature sets that produce systems that are superior, for some operating conditions, to the final system.

The implication of this is that wrappers will produce unpredictable results in variable cost domains.[4] The designer of a classification system is therefore presented with a dilemma: feature selection is required, but the methods available were not constructed for variable cost domains, and so assurances cannot be given that the best available system is being designed. Such an assurance can only be made about a system operating at the costs for which the wrapper objective function was optimised.

The designer might try to answer this dilemma by modifying the wrapper objective function, making it more suitable for variable cost problems. In this vein, Lovell *et al*[67] attempted to maximise the $AUROC$ as the objective function of a feature selection wrapper. However, as pointed out by Hand[45], the $AUROC$ is a drastic simplification of the information conveyed by an ROC. A comparison between two curves on the basis of area gives no indication of whether the curves cross, which is precisely the situation we wish to avoid.

In the next section I will show that neither the use of accuracy nor $AUROC$ as a wrapper objective produces systems that are superior over all operating conditions.

4.4 Empirical results

Current wrapper methods, whether they employ accuracy or $AUROC$, are inappropriate in variable cost domains. It is being assumed implicitly that the feature set which produced the best classifier for the fixed costs used by the wrapper, will also produce the best classifier for other costs. While it is not clear that this is necessarily *always* untrue, intuitively the assumption does not seem reasonable. This assumption is tested here, for wrappers using accuracy and $AUROC$ objectives, and shown to be incorrect for the problems examined.

By examining two real world problems, using wrapper algorithms, I will illustrate the effects of variable operating conditions, and provide results with which we might answer the third question posed in this chapter:

Question 5 *Is it possible, using the wrapper algorithms we have seen, to find a single feature set that that performs well across multiple costs?*

4.4.1 A wrapper with an accuracy based objective

The first test was to examine accuracy based wrappers. The results of applying a *sequential forwards floating selection* ($SFFS$) algorithm to the *Grey* and *BrodSat* problems were examined.

[4]It is also clear that filter algorithms will also face these problems, as a fixed cost function is also assumed.

$Grey$	BrodSat
$\{ 16 \}$	$\{ 18 \}$
$\{ 16, 23 \}$	$\{ \mathbf{17}, \mathbf{18} \}$
$\{ 16, 23, 24 \}$	
$\{ 23, 24 \}$	
$\{ \mathbf{8}, \mathbf{23}, \mathbf{24} \}$	

Table 4.1: The feature sets considered by a $SFFS$ wrapper using an accuracy objective criterion, applied to the $Grey$ and $BrodImg$ data sets. The sets eventually selected are in bold-face type, the other sets were considered but rejected by the wrapper.

In each case, a sequence of feature subsets had been considered during the operation of $SFFS$, these are given in Table 4.1.

Each of these feature sets was used to train a classification algorithm, and an ROC curve produced for each resulting classification system. Following the experimental methodology detailed in Chapter 2, a hold-out data set was used to calculate the ROC curves. For each problem, the hold-out ROC curves were plotted together to see if overlap occurred. A problem will occur for the designer if the classification system ROC curve for the selected feature set is crossed by the curve produced using a rejected set.

In both the $Grey$ and the $BrodSat$ problems, the ROC curves of the classification systems trained with rejected subsets crossed the ROC curve of the system trained using the final subset (see Figures 4.3 4.4 and). In the case of the $BrodSat$ problem, this crossing does not appear severe, it is nevertheless problematic. At the points where the curves cross and touch, the operating performance is similar, and it might be argued that simply picking the generally more dominant curve is the correct option, in this case. However, should the system eventually operate at these points, that would mean that we had selected a classification system twice as complex as was needed: it would use two features when one would suffice.

4.4.2 A *wrapper* with an $AUROC$ objective.

Lovell *et al* [67] account for variable operating costs by attempting to maximise the $AUROC$ rather than accuracy, when carrying out feature selection. It was decided to test the results of this wrapper for crossing ROC curves. A wrapper which used a naive Bayes classification algorithm and an $AUROC$ based objective function was applied to the $BrodSat$ problem. Using a SFS search algorithm, a feature set was chosen that maximised the $AUROC$ on a validation data set. The feature sets considered by the wrapper are given in Table 4.2.

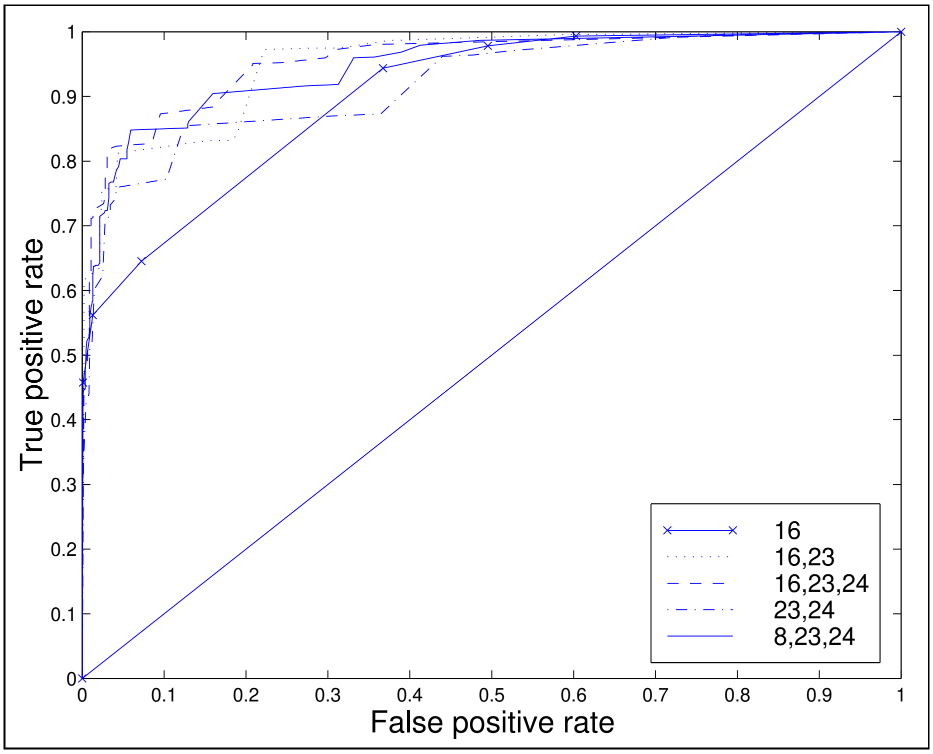


Figure 4.3: The five feature subset ROCs in the $Grey$ problem. The ROC curves cross indicating that no system is dominant over all operating conditions.

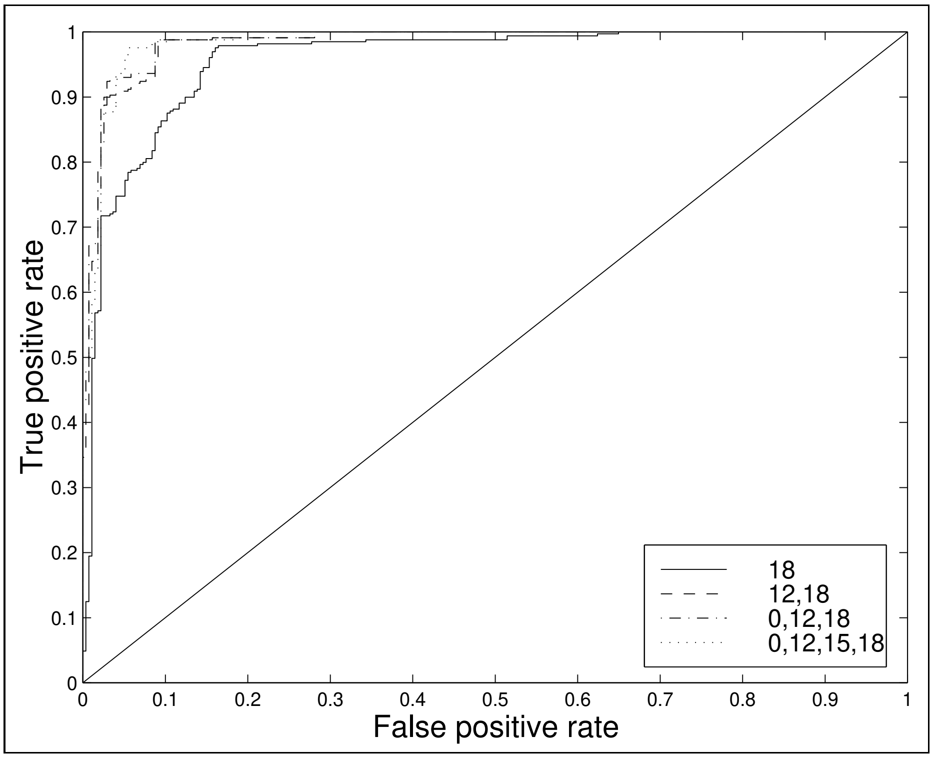


Figure 4.5: The 4 feature subset ROCs in the $BrodSat$ problem. The ROC curves cross indicating that no system is dominant over all operating conditions.

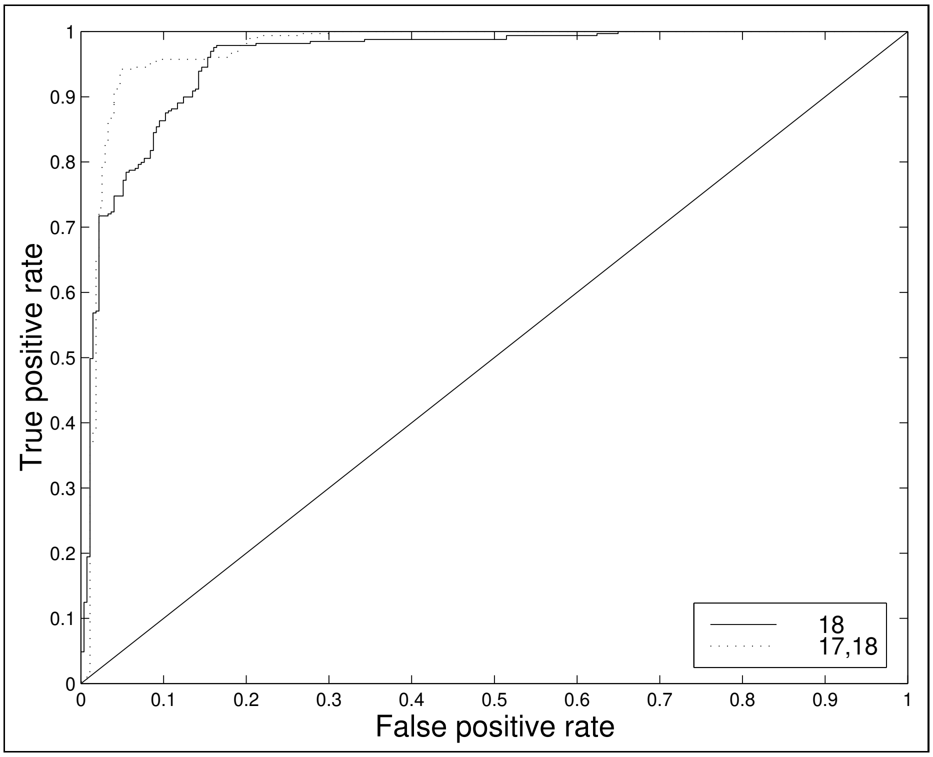


Figure 4.4: The two feature subset ROCs in the $BroadSat$ problem. The ROC curves cross indicating that no system is dominant over all operating conditions.

As in the previous tests, ROC curves were calculated using hold-out data sets, for classification systems trained with each of the feature sets in Table 4.2. The curves are plotted in Figure 4.5. It can be seen that the problem of crossing ROC curves has not been avoided by maximising the $AUROC$ rather than accuracy in the wrapper objective function.

4.5 Conclusions

This chapter answered three questions; the first by giving real world examples, the others with empirical results.

Question 3 *Is it likely that variable misclassification costs might be encountered in practice?*

Question 4 *What are the implications of variable costs when carrying out feature subset selection?*

Question 5 *Is it possible, using the wrapper algorithms we have seen, to find a single feature set that that performs well across multiple costs?*

$BrodSat$
$\{ 18 \}$
$\{ 12, 18 \}$
$\{ 0, 12, 18 \}$
$\{ \mathbf{0, 12, 15, 18} \}$

Table 4.2: The feature sets considered by a $SFFS$ wrapper using an $AUROC$ objective criterion, applied to the $BrodImg$ data set. The set in bold-face was selected by the wrapper, the remainder were considered, but rejected.

Through real world examples of classification system design, we have seen that variable operating conditions arise not only as a result of undetermined misclassification costs, but also as a requirement in the design specifications.

The results of carrying out feature selection with a wrapper algorithm were then examined. Two types of wrapper objective function were used: one maximising classification accuracy, the other maximising $AUROC$. For two real world problems, it was shown that variable costs could lead to unpredictable, and undesirable behaviour from the wrapper, regardless of the objective function used. A single feature set, producing a generally dominant ROC curve, could not be discovered for either of the classification problems considered.

Based on the examples and results presented in this chapter, we can conclude that there exist design problems for which we

1. must carry out feature selection;
2. cannot specify the operating conditions *a priori*;
3. cannot find a single feature set, producing a classification system with a dominant ROC curve, using the $SFFS$ wrapper.

These conclusions raise the following question:

Question 6 *If we cannot find a suitable single feature subset for designing a particular classification system, what might we do about this?*

In the rest of this report, I will develop a novel algorithm for carrying out feature selection; one that is robust to variable operating conditions, and that is not limited to selecting a single feature set. In the next chapter, a new technique for combining distinct classifiers is presented, allowing the utilisation of multiple feature sets. This leads, in Chapter 6, to the *Parcel* algorithm for feature selection in variable cost domains.

Chapter 5

The maximum realisable ROC

5.1 Introduction

When applied in variable cost domains, wrapper feature selection algorithms can produce unsatisfactory results. Essentially, each feature subset considered by the wrapper represents a classification system. In order to pick the "best" feature subset, the wrapper decides which classification system is the "best". In the light of the results of Chapter 4, we have posed the question:

Question 6 *If we cannot find a suitable single feature subset for designing a particular classification system, what might we do about this?*

This chapter addresses the issue arising when the ROC curves of two or more candidate classification systems cross: that of choosing the "best" system. It will be shown that, in this situation, no single system can be said to be superior to the others. Rather, a system made by combining the best aspects of the originals should be formed, this being generally superior to any of the original individual systems.

The chapter opens by describing in detail the problem of determining the "best" classification system when the ROC curves of the candidates cross. Next, a novel technique for combining classifiers is proposed, and a related theorem proved. This technique leads to a method for combining two or more classification systems. For all false positive rates, the resulting combination system is as good as, or better than, all of the originals, in terms of true positive rates. The chapter concludes with empirical results from both artificial and real world problems, and a discussion on the applicability of this method in practice.

5.2 Choosing the "best" classification system

During the design of a classification system, a number of candidate systems may be considered, and it is the goal of the design process to choose the system that is, in some sense, the "best" of those available. This section will illustrate that this goal is not, in many practical situations, achieved. It would be incorrect to interpret the "best" system as meaning "the dominant *single* system from the set of existing systems". We will show that a truly "best" system will be comprised of elements of a number of existing systems, instead of just one, and a simple method for producing such a composite system will be described. We will then indicate that it is possible to produce more sophisticated composite systems by the application of a novel technique for combining classifiers. The rest of this Chapter will be devoted to the description of these more sophisticated systems.

In Chapter 3 we reviewed the literature concerning a feature selection framework, the wrapper, that selects the candidate system (and thus a feature set) that maximises classification accuracy (or some related function of misclassification costs). Then, in Chapter 4, variable or undefined operating costs were introduced. The Neyman Pearson criterion (a maximum allowable false positive rate) was defined, and it was shown that this criterion could be imposed during operation, to choose a classifier from the ROC curve of a classification system. It was also stated that, due to changing practical constraints during operation, the Neyman Pearson criteria might change, and in this event an alternative classifier would be chosen from the ROC, using the new maximum false positive rate.

In a real world example, ROC curves were used to analyse the performance of the candidate systems being considered by a wrapper. It was shown that the curve of the chosen system and those of the rejected systems crossed at various points. Although the wrapper selected the "best" system for $0 - 1$ loss, it is clear that, for various Neyman Pearson criterion, the chosen system might not be the best available (see Figure 5.1 for an example of this). Furthermore, it was shown that using the area under the ROC ($AUROC$) as a performance criteria did not remedy the problem. The wrapper, using the $AUROC$, still selected a system whose ROC curve was crossed by those of rejected systems. The underlying problem can be stated as so:

> If two or more classification systems are being considered, the ROC curves of which cross, then this indicates that no single system is superior, in terms of true positive rates, to the others across the entire range of possible false positive rates. Instead, one system may be superior for some subset of operating conditions, and others superior for those remaining. Therefore, it is not possible, in this circumstance, to pick a single "best" classification system from those available; "best" implies that there is no false positive rate for which an alternative system would have a higher true positive rate, *i.e.* a "best" ROC curve encloses all others.

This problem makes the use of unqualified summary metrics, such as the $AUROC$ (the Wilcoxon statistic), less desirable. A single measure tends to suggest the selection of a single superior classification system. Indeed, the $AUROC$ is widely used to select classification systems in

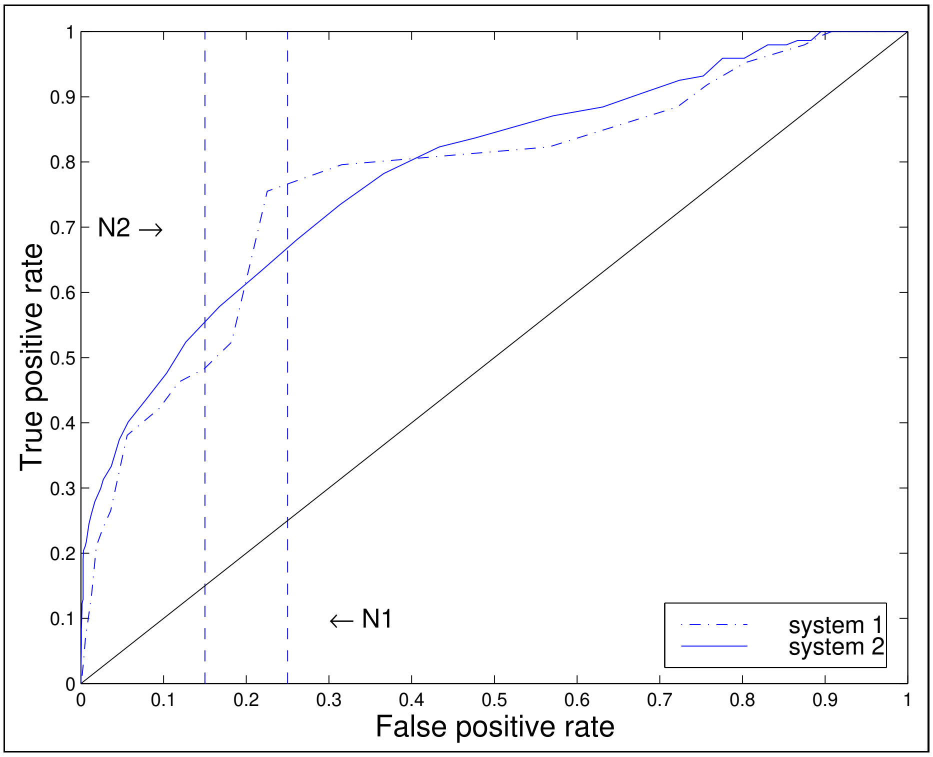


Figure 5.1: The problem of selecting a single "best" classifier. System 2 has a larger area than system 1, and might be chosen as the "best" system using the $AUROC$ as a selection criteria. The problem is illustrated by imposing two Neyman Pearson criteria, $N1$ and $N2$. $N1$ allows for a maximum false positive rate of 0.15, and $N2$ a rate of 0.25. It can be seen that under $N2$, a classifier should indeed be chosen from system 2. However, under $N1$, a classifier should be chosen from system 1, as it would significantly outperform a corresponding classifier from system 2.

practical applications. Often the implication is that the chosen system is superior, over all operating conditions, when compared to the available alternatives. However, if the ROC curves cross, a judgement regarding the overall superiority of one system should be qualified with this information. This qualification could prove crucial, as at some point a Neyman Pearson criterion might be imposed for which the chosen system is not the best available.

In the experiments of the previous chapter, we saw the ROC curves of multiple classification systems crossing. One system was chosen as superior to the others by the wrapper algorithm, but no qualification was given indicating the operating conditions (Neyman Pearson criteria) for which the chosen system was superior. In real world problem domains, such as medical diagnosis, small improvements in true positive rate can, literally, be the difference between life and death. The task then, will be to build into the design process a method by which the best true positive rates are always obtained for each false positive rate.

There exists a simple solution to the above problem: during design, locate the points at which

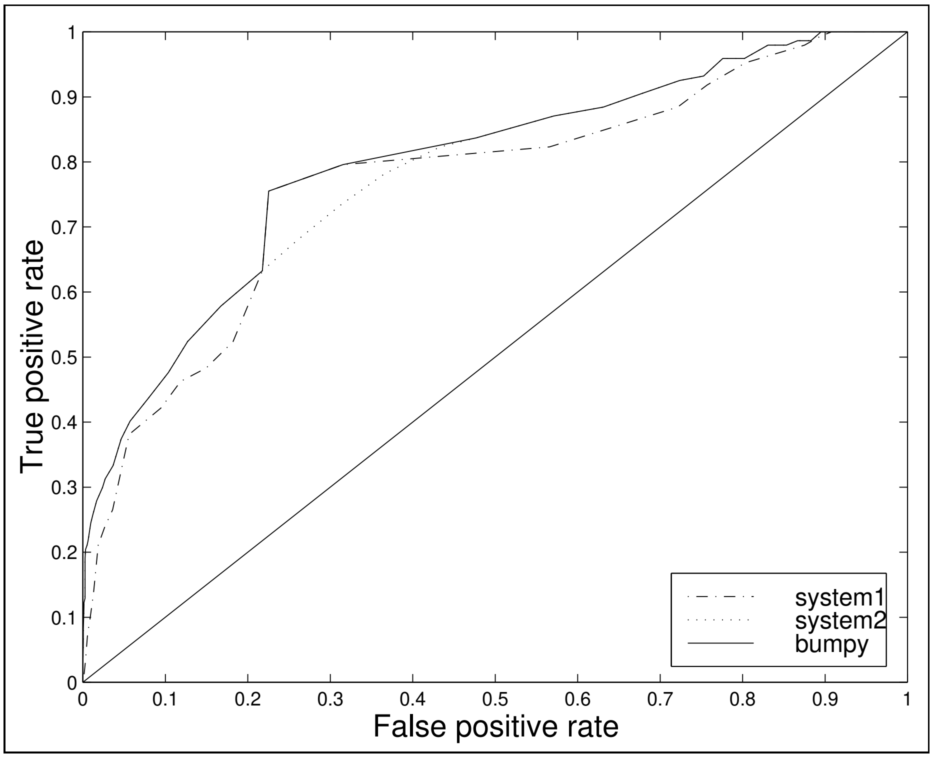


Figure 5.2: The *bumpy* ROC curve formed by taking the dominant sections of the existing curves and forming a composite system with them.

curves cross, and, when operating between two such points, select a classifier from the system that produces the uppermost (i.e. higher true positive rate) curve. This would, in effect, produce a composite classification system. This composite system would have a *bumpy* ROC curve, composed of sections of curve from different systems (see Figure 5.2).

Using Neyman Pearson criteria, classifiers could now be selected from this composite curve. In this Chapter, we propose a more sophisticated solution that will guarantee to be at least as good as, if not better than the *bumpy* composite solution. In the next section, a novel technique for combining classifiers using randomised decision rules is proposed. This will lead to a classification system whose ROC curve is the convex hull over all the existing ROC curves. Given the available systems, this curve, called the maximum realisable ROC ($MRROC$), represents a system that is equal to, or better than, all the existing systems, for all Neyman Pearson criteria.

5.3 Classifier combination

In this section a novel technique for combining classifiers is proposed, and a related theorem proved. This technique will allow the construct of composite classification systems, leading to the $MRROC$.

We think of an ROC curve as representing a set of points in a ROC space. A point (fp_c, tp_c) represents an existing classifier c, that classifier producing a false positive with a probability $Pr(\text{false} - \text{positive}) = fp_c$, and a true positive with probability $Pr(\text{true} - \text{positive}) = tp_c$.

Take two classifiers, c_a and c_b, each with distinct false positive and true positive rates. These two classifiers are the end points of a straight line in ROC space, L_{ab}. The line L_{ab} defines a set of classifiers, i.e. point $(fp_{c_r}, tp_{c_r}) \in L_{ab}$ represents the classifier that would produce those true-positive and false-positive rates.

We observe in this chapter, that, given only c_a and c_b, one may realise the output of classifier c_r by randomly choosing between the output of c_a and c_b. The probability of choosing the output of c_a over that of c_b is determined by the distance along L_{ab} between c_r and c_a.

This technique has parallels in classical statistics.[1] When estimating the power of a hypothesis test, the sample space of which has discrete probabilities, randomised decision rules could be employed. This allowed the estimation of specific power[2] values, even when an observed estimate was unavailable [37, 101] The use of randomised decision rules had been adapted from statistical game theory, where it was known that, given a number of discrete outcomes in a loss space, all the points in the convex hull over the existing loss points could be achieved via random decision strategies. To apply these methods, the hypothesis test was cast as a two player game. As power values are now easily computable to high precision, estimation techniques involving randomised decision rules are all but defunct, and are seldom even mentioned in modern statistical texts.

Theorem 1 *The realisable classifier. Two existing classifiers, c_a and c_b, produce true positive and false positive rates (tp_a, fp_a) and (tp_b, fp_b) respectively for a series of m inputs $x_1..x_m$. In a 2 dimensional plot of false positive rate against true positive (ROC space), call the straight line linking (fp_a, tp_a) and (fp_b, tp_b) L_{ab}.*

Any point (fp_r, tp_r) on L_{ab} corresponds to the point that would be produced by a classifier c_r. Call the set of classifiers corresponding to n points on L_{ab}, $\mathbf{R} = \{c_{r,1}, .., c_{r,n}\}$.

Given c_a and c_b, the output of a realisable classifier, $c_{r,i} \in \mathbf{R}$, for any input x_j, is a random variable that assumes the output of one or other of c_a and c_b with probability

$$\begin{aligned} Pr(c_{r,i}(\cdot) = c_b(\cdot)) &= \frac{fp_{c_{r,i}} - fp_a}{fp_b - fp_a} \\ Pr(c_{r,i}(\cdot) = c_a(\cdot)) &= 1 - Pr(c_{r,i}(\cdot) = c_b(\cdot)), \end{aligned}$$

where $fp_{c_{r,i}}$ is the false positive rate of $c_{r,i}$.

[1] The author would like to thank Dr. David Spiegelhalter for first indicating this parallel.

[2] The *power* of a statistical hypothesis test is the probability of rejecting the null hypothesis when it is false. In terms of a medical test, the null hypothesis states that the patient is healthy, therefore the power of a test is equivalent to the true-positive rate.

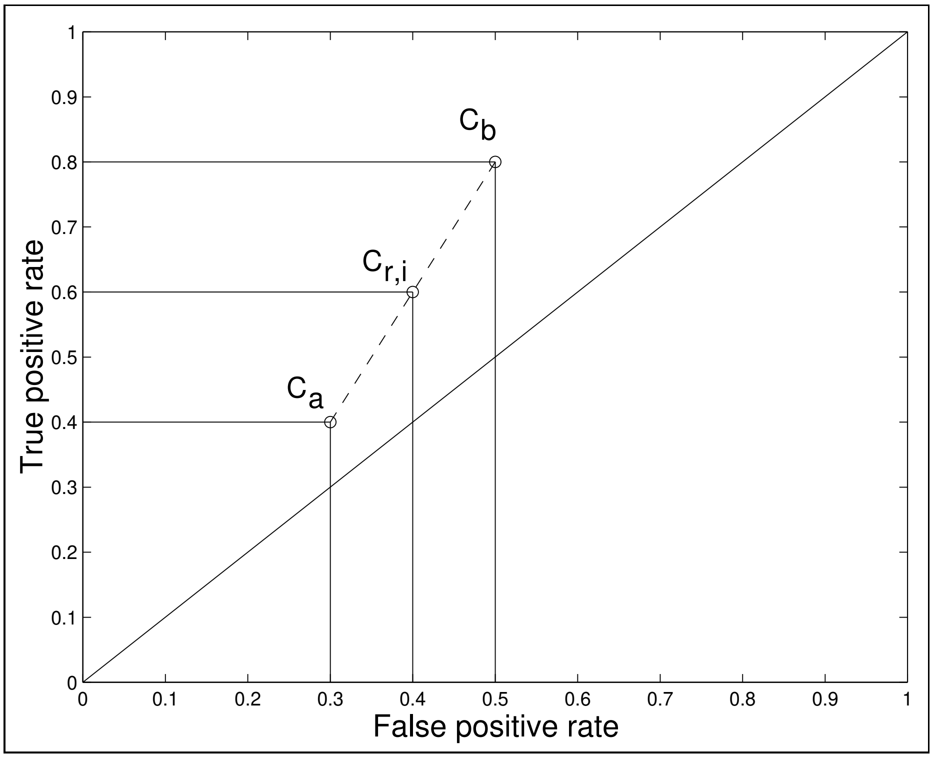


Figure 5.3: An example of a realisable classifier. The point $c_{r,i}$ on the line joining c_a and c_b may be realised by the application of Theorem 1

The proof of Theorem 1 is straightforward. To construct the output of a realisable classifier $c_{r,i}$ with false positive rate $fp_{c_{r,i}}$, randomly select between the outputs of c_a and c_b with the given probability. The expected false positive rate produced by doing so is

$$\begin{aligned} E[fp] &= Pr(c_{r,i}(\cdot) = c_b(\cdot)) \times fp_b + Pr(c_{r,i}(\cdot) = c_a(\cdot)) \times fp_a \\ &= \frac{fp_{c_{r,i}} - fp_a}{fp_b - fp_a} \times fp_b + (1 - \frac{fp_{c_{r,i}} - fp_a}{fp_b - fp_a}) \times fp_a \\ &= \frac{fp_b(fp_{c_{r,i}} - fp_a) + fp_a(fp_{r_i} - fp_a) - fp_a(fp_b - fp_a)}{fp_b - fp_a} \\ &= \frac{(fp_b - fp_a)(fp_{c_{r,i}} - fp_a) + fp_a(fp_b - fp_a)}{fp_b - fp_a} \\ &= fp_{c_{r,i}} \quad Q.E.D. \end{aligned}$$

And similarly for the true positive rate.

Figure 5.3 illustrates an example of a realisable classifier. The realisable classifier $c_{r,i}$, with false positive rate $fp_{c_{r,i}} = 0.4$, lies on the line between classifiers c_a and c_b, with false positive rates $fp_a = 0.3$, and $fp_b = 0.5$ respectively. To realise the output of $c_{r,i}$, calculate the probabilities for selecting the outputs of the existing classifiers using Theorem 1,

$$\begin{aligned}
Pr(c_{r,i}(\cdot) = c_b(\cdot)) &= \frac{fp_{c_{r,i}} - fp_a}{fp_b - fp_a} \\
&= \frac{0.4 - 0.3}{0.5 - 0.3} \\
&= 0.5 \\
Pr(c_{r,i}(\cdot) = c_a(\cdot)) &= 1 - Pr(c_{r,i}(\cdot) = c_b(\cdot)) \\
&= 0.5.
\end{aligned}$$

To obtain the classification output of $c_{r,i}$ on a set of unseen cases, $\mathbf{x} = \{x_1, .., x_n\}$, the classifications of c_a and c_b would be calculated

$$\begin{aligned}
c_a(\mathbf{x}) &\rightarrow \{(x_1 = \text{Adverse}), (x_2 = \text{Adverse}), (x_3 = \text{Benign}), .., (x_n = \text{Adverse})\} \\
c_b(\mathbf{x}) &\rightarrow \{(x_1 = \text{Benign}), (x_2 = \text{Adverse}), (x_3 = \text{Adverse}), .., (x_n = \text{Benign})\}.
\end{aligned}$$

Using the probabilities calculated above, the output of $c_{r,i}$ is then determined by randomly selecting one of the outputs, like so:

$$c_{r,i}(\mathbf{x}) \rightarrow \{(c_a(x_1) = \text{Adverse}), (c_b(x_2) = \text{Adverse}), (c_a(x_3) = \text{Benign}), .., (c_b(x_n) = \text{Benign})\}$$

5.4 The maximum realisable ROC

We can now realise all classifiers that lie on straight line segments with end points formed by existing classifiers. What advantage can be gained by this? Take the example[3] illustrated in Figure 5.4. The ROC is produced using a linear model on the 1 dimensional classification problem shown. The steps in the ROC occur because the linear model cannot capture the multi modal nature of the data.

We currently have a set of classifiers provided by the linear model. The current ROC curve is produced by this set, and can be used to select the best available classifier for a particular false positive rate. It is possible, however, to obtain a new set of classifiers that will give better performance, in terms of true positive rates, than those provided by the linear model.

Calculate a convex hull [85] such that it contains all the points on the current ROC. The vertex points of the convex hull will be points corresponding to existing classifiers generated by the linear model. We know from Theorem 1 that all the points on a facet of the hull (i.e. on a straight line between two vertices) represent realisable classifiers.

The convex hull represents a set of realisable classifiers that will at all times be either equal or superior to those of the linear model (with respect to Neyman Pearson criteria), and that are generated by a subset of the original classifiers. The convex hull is the maximum realisable ROC ($MRROC$) given the available existing classifiers.

[3] The example given in Figure 5.4 is a deliberately extreme one, designed to illustrate clearly the effects of the *realisable classifier theorem*.

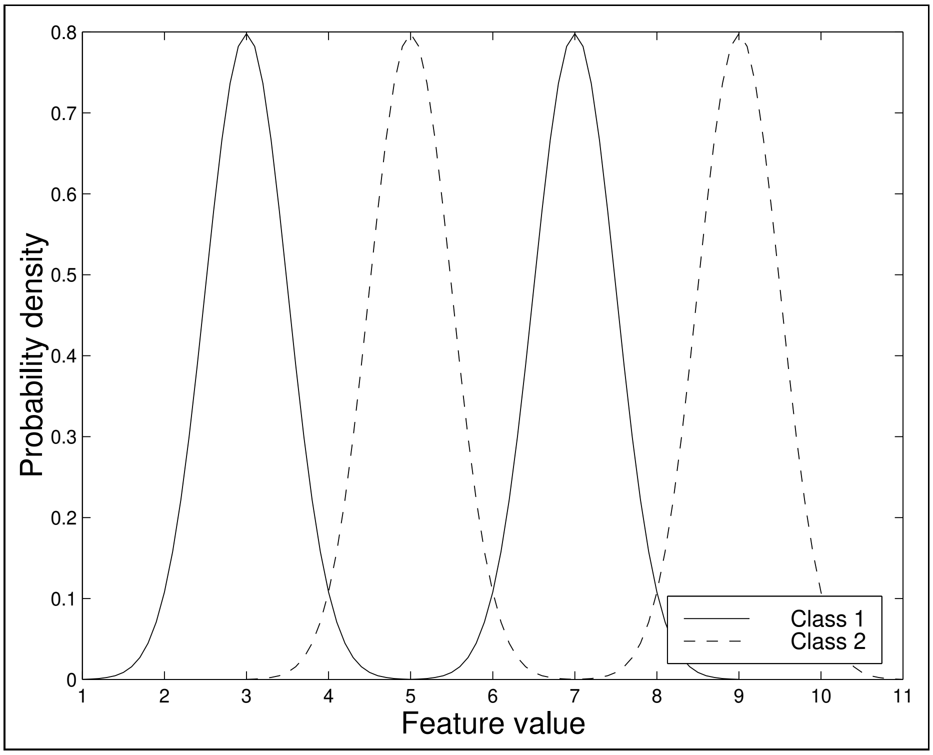


Figure 5.4: An example of 2 class multi modal data. Due to the nature of the data, accurate classification with a linear classification system will be difficult.

5.5 Empirical results

The results of two experiments are presented here. The first experiment was carried out to demonstrate estimation of the $MRROC$ in practice. The experiment had two parts; one dealing with artificial data, the other with a real world medical dataset. The results show that, not only is the estimation of the $MRROC$ a valuable tool in the design of a classification system, but that it is also easily achievable. The second experiment looks at the results of the the wrapper algorithm presented in Chapter 4. Taking the logical step of applying the *realisable classifier theorem* to these results achieves a $MRROC$ that is larger than any of the constituent curves, and reproducible on an unseen data set.

5.5.1 Experiment 1

5.5.1.1 Objectives

The objective of this experiment was to verify empirically that application of the *realisable classifier theorem* could lead to achievable $MRROC$ curves on novel data. In the first part of the

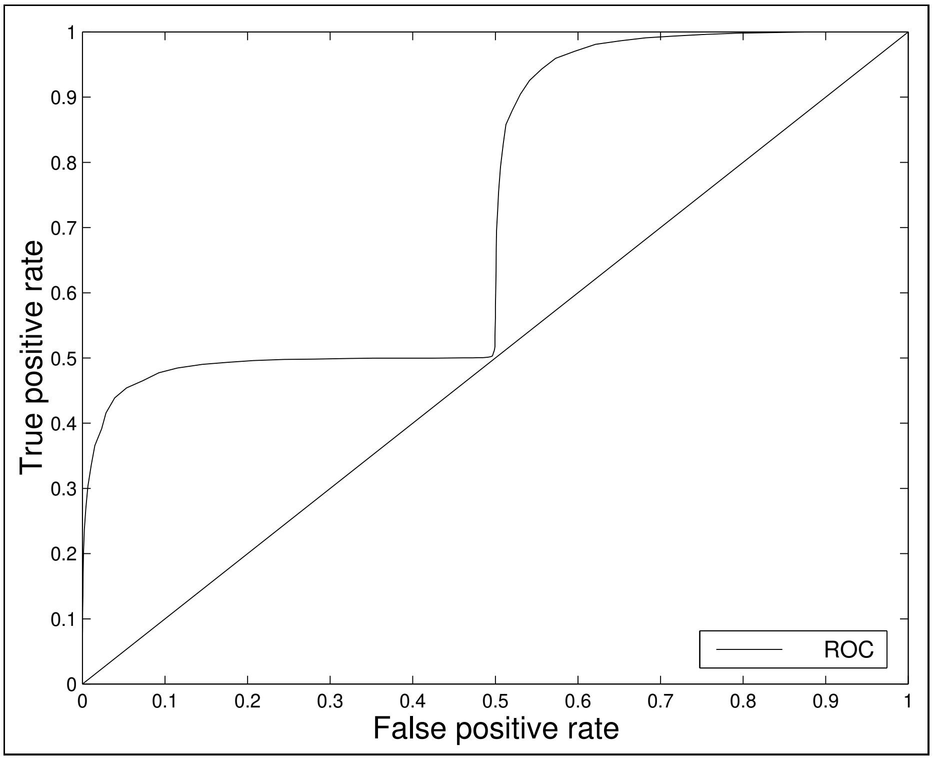


Figure 5.5: The ROC curve produced by varying the threshold on the output of a linear model of the multi modal data shown. The ROC has a step like appearance because the linear model fails to capture the nature of the data.

experiment, an artificial data set is used with a linear classifier. This data is designed to exhibit clearly the non-convexity required for the *realisable classifier theorem* to be appropriate. In the second part of the experiment, two linear classifiers are applied to a real world medical data set. The ROC curves of both systems cross, indicating that neither is superior for all costs. This data set was used to demonstrate that an $MRROC$ curve predicted on a real dataset was achievable on novel data.

5.5.1.2 Part 1: Artificial data

Data Multi modal data was generated for the one dimensional, two class classification problem of Figure 5.3. A linear model was trained using 5000 training examples. By varying the threshold used on the output of the model when presented with 5000 validation cases, the ROC curve of Figure 5.3 was obtained. The true-positive rate was the rate of correct classifications of class 1, the false-positive rate was the rate of cases of class 2 being incorrectly classified as belonging to class 1.

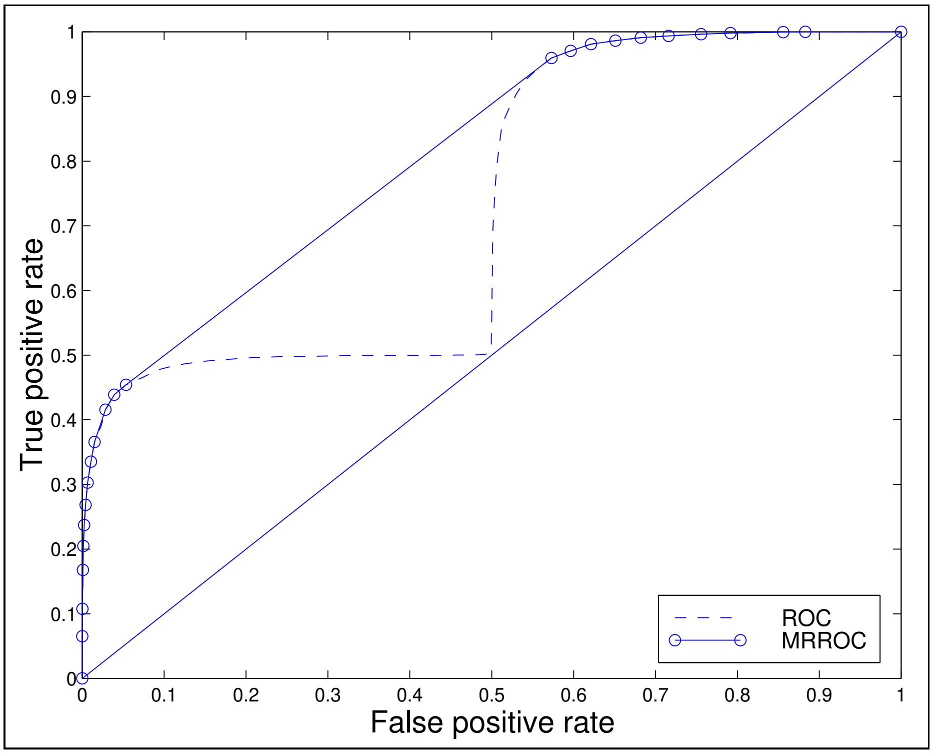


Figure 5.6: The convex hull containing the ROC of a linear model is found. This hull is the $MRROC$ of the set of realisable classifiers produced from the set of existing linear classifiers.

Results Using the quickhull algorithm [8], the convex hull containing all the points in the ROC was obtained. Each vertex in the hull represented an existing classifier. Each of these existing classifiers was defined by the threshold used, on the output of the linear model, to yield a final classification for each validation case. It was required to show that all the points on the facets of the convex hull, corresponding to classifiers that were not currently available, could be realised by application of Theorem 1 of this paper to the set of vertex classifiers. The $MRROC$ indicates the expected performance, over the complete range of false positive rates, that one might hope to achieve using this approach.

Figure 5.4 plots the $MRROC$ over the ROC of the linear model on the validation data.

To show empirically that the characteristic curve indicated by the $MRROC$ could actually be obtained, a third data set of 5000 hold-out cases was generated. This hold-out data was processed by the linear model. The thresholds corresponding to the existing classifiers in the convex hull were each applied to the outputs of the linear model, producing a number of sets of classifications. For any point on a facet of the hull, a classification for an individual hold-out case could be obtained by randomly selecting one of the classifications made by the two existing classifiers at the end points of the facet. As described above, this methodology leads to the realisation of the set of classifiers on the facets of the hull.

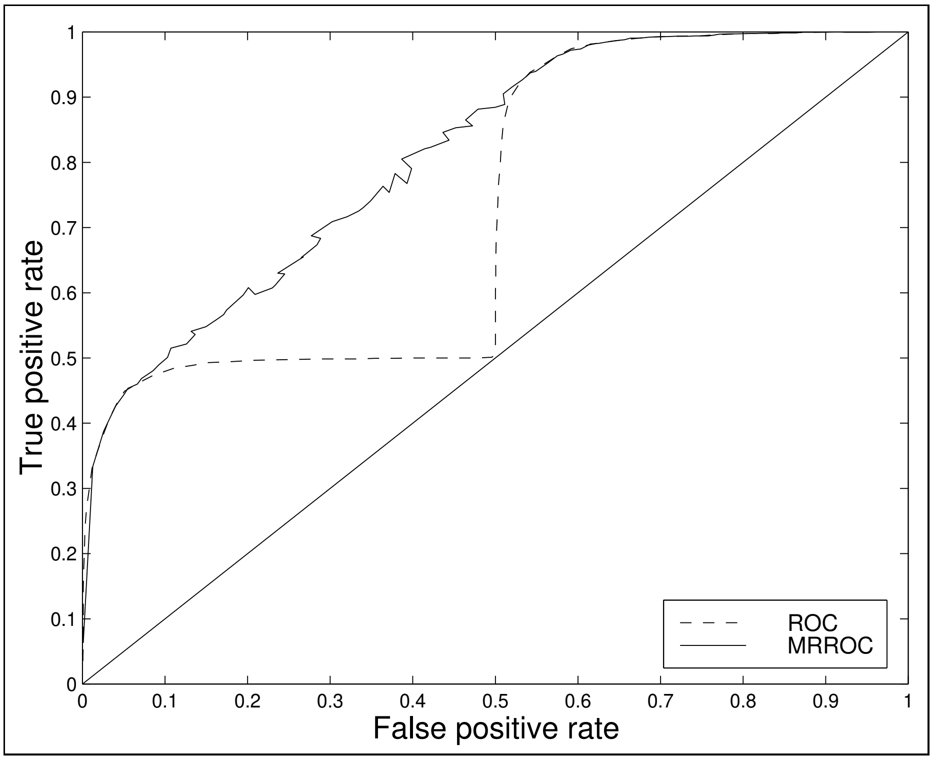


Figure 5.7: The $MRROC$ plotted for the hold-out data set. The $MRROC$ is consistent with that predicted.

Figure 5.5 plots, for the hold-out data set, the ROC of the $MRROC$ realisable classifiers against the ROC of the linear model. It can be seen that the set of realisable classifiers produce an ROC consistent with the $MRROC$, and superior to the ROC of the linear model. The $MRROC$ appears slightly jagged. This is entirely consistent with the nature of the classifiers used to form it. The classifiers are random variables, whose central tendency will be to lie on the $MRROC$.

5.5.1.3 Part 2: Thyroid data

Data A medical data set describing patients with abnormal thyroid conditions was obtained from the UCI machine learning repository [72] (see Thyroid classification problem, Chapter 2). The data originally contained 7200 instances, with 3 possible classes, *hyperthyroid, hypothyroid*, and *normal*, and 21 features. In this experiment, the classes were merged to form 2: *Adverse* and *Benign*. The data was randomly split into 3 data sets; a training set with 3800 instances, a validation set with 1700 instances, and a hold-out with 1700 instances.

Results Two classification systems were made, System 1 and System 2, using a simple linear model trained with a single feature to describe the data. The plot in Figure 5.8 shows the ROC

curves for both classification systems using the validation data set to calculate the true and false positive rates (note that the curves cross). The plot in Figure 5.9 shows the $MRROC$ predicted by the convex hull containing the Validation ROC curves. The vertex points on the hull corresponded to existing classifiers. It was required to show that all the points on the convex hull were realisable classifiers (by Theorem 1) and could be achieved in practice, resulting in the $MRROC$.

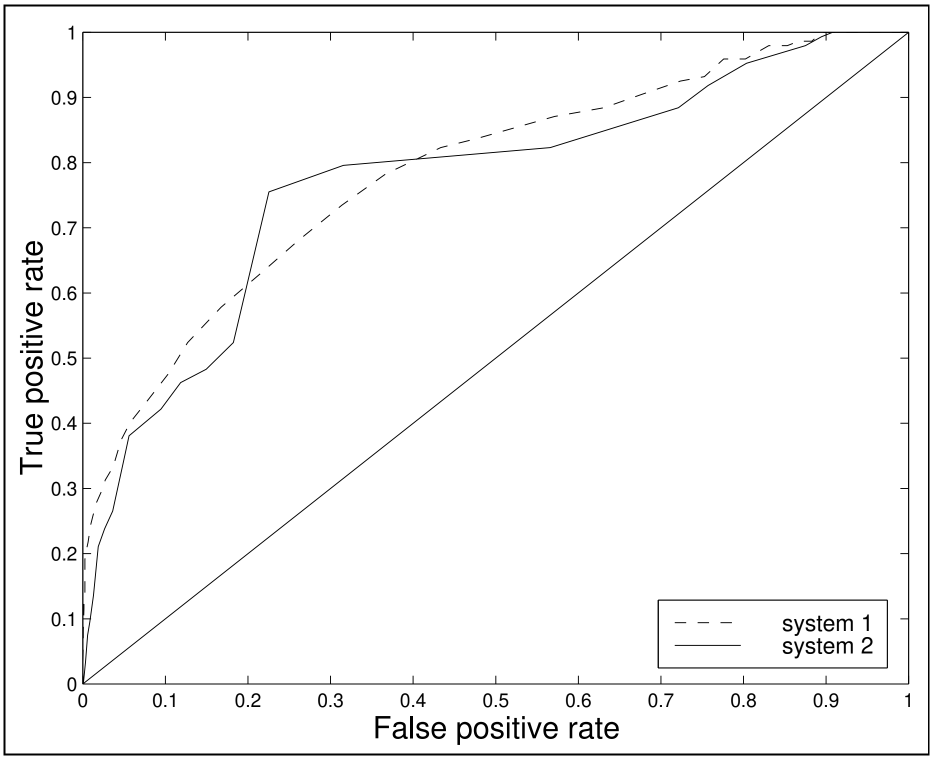


Figure 5.8: The ROC curves of two abnormal thyroid classification systems cross.

The ROC curves for both of the original systems, and for the set of realisable classifiers on the hull are plotted for the hold-out data in Figure 5.10. The $MRROC$ produced by application of Theorem 1 is consistent with that predicted, indicating that the $MRROC$ is achievable in practice.

5.5.2 Experiment 2

5.5.2.1 Objectives

The Grey classification problem is revisited. It was the objective of this experiment to show the utility of applying the *realisable classifier theorem* in a feature selection problem, using a real world dataset.

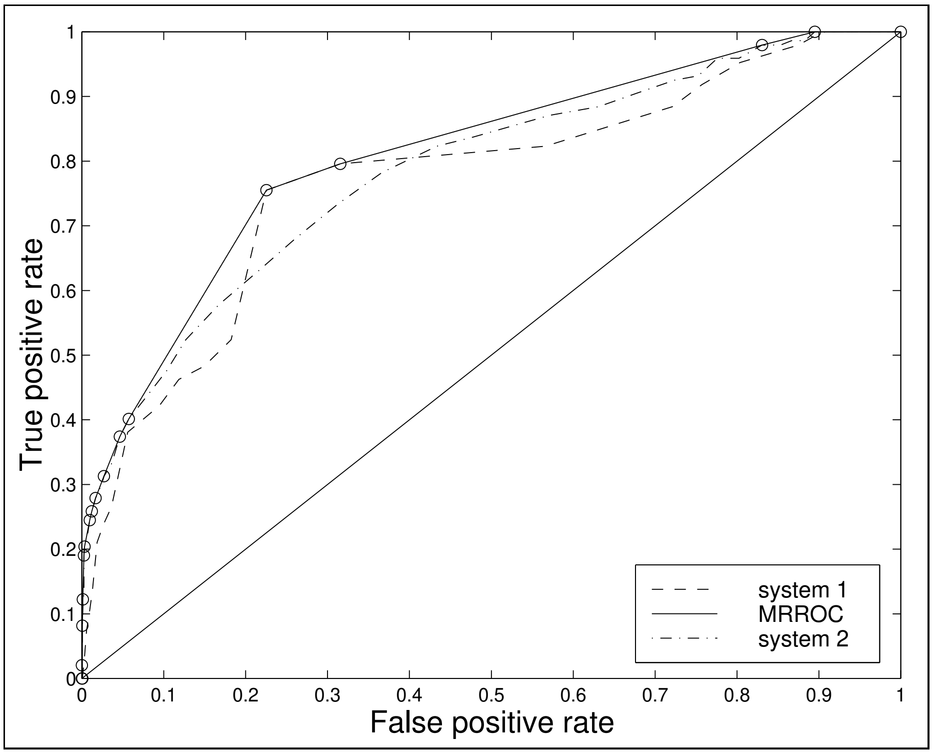


Figure 5.9: The convex hull over both original ROC curves. This is the predicted $MRROC$.

In Chapter 4 an experiment using the Grey data was carried out. It was shown that the classification system produced using a feature subset selected by a wrapper algorithm was *not* superior over all Neyman Pearson criteria to those subsets that had been rejected on the basis of performance at $0 - 1$ loss. In this experiment, the logical step of applying the *realisable classifier* theorem to the results of the $SFFS$ wrapper algorithm was taken. The goal is to keep the best operating points produced by each of the five feature sets considered by the wrapper (see Chapter 5.4), and then create a $MRROC$. It is required to show that the system produced using the $MRROC$ is superior or, at worst, equal in performance, across all Neyman Pearson criteria, to the system produced by the wrapper.

5.5.2.2 Data

The data used for this experiment was the Grey classification data, described in Chapter 2, and used in the experiments of Chapters 4.

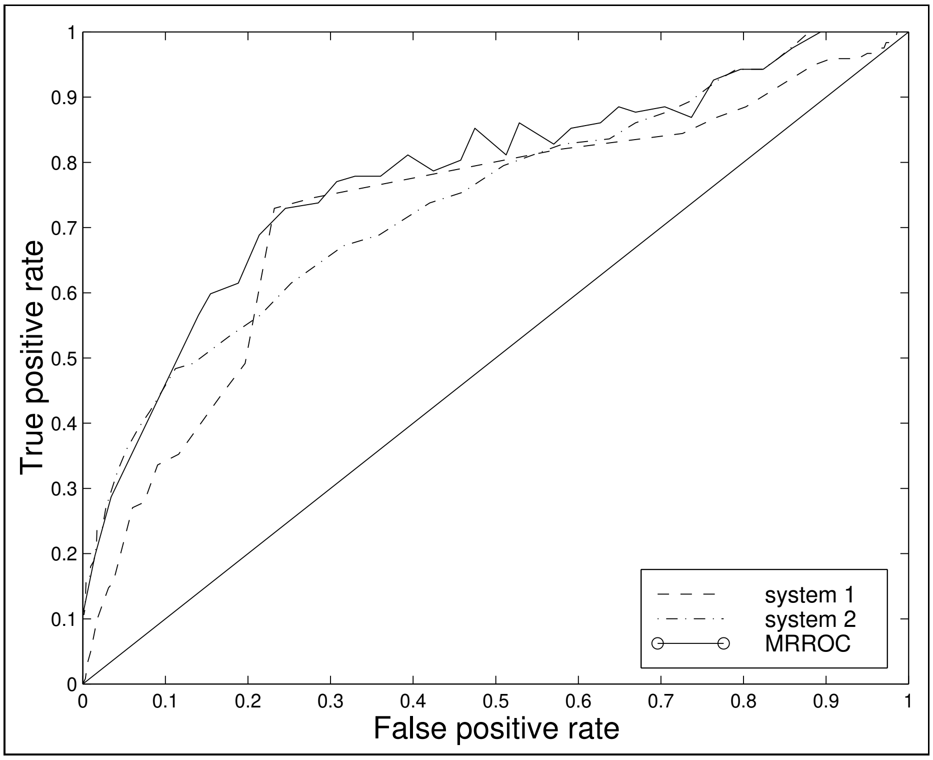


Figure 5.10: The $MRROC$ plotted for the hold-out data set. The $MRROC$ is consistent with that predicated by the convex hull over the validation data ROC curves. The ROC curves for System 1 and System 2 using the hold-out data are plotted for comparison with the $MRROC$.

5.5.2.3 Results

Using the training and validation data sets, the ROC curves for the classification systems corresponding to each of the 5 feature subsets were evaluated, Figure 5.11. It can be seen that when the costs vary from error rate, no single feature set produces a superior classification system. The $MRROC$ was predicted by fitting the convex hull over the 5 curves. The classifiers at each vertex were saved.

In Figure 5.12 the ROC curves for the five classification systems and the $MRROC$ on the hold-out data set are presented. The $MRROC$ obtained by application of the *realisable classifier theorem* on the hold-out data is consistent with that predicted by the $MRROC$ on the validation data, and is the dominant curve when compared to any of the individual systems.

5.6 Conclusions

This technique for realising novel operating points given a set of existing classifiers and the ROC formed by them has implications for designers of classification systems in domains where Neyman Pearson operating criteria may vary or not be known *a priori*. It has been shown theoretically that an enhanced ROC curve, the $MRROC$, can be achieved by application of the *realisable classifier theorem*, and empirical results provided, on both artificial and real world data, to show the effect in practice. Given two ROC curves that cross, the $MRROC$ produced using both will be superior to either alone, and may realise operating points that were previously unavailable.

In Experiment 1 (Section 5.5.1), on multi modal artificial data, the realisable classifiers lying on the facets of the convex hull represent classifiers that are not possible to obtain with the original linear classification system. The increased range of possible classifiers is not obtained at the expense of clarity or simplicity, nor does it require some degree of expert knowledge to be teased out of the system. The gain is obtained by a clear and simple analysis of the currently available system (via the ROC), and the utilisation of the information acquired by that analysis.

The *realisable classifier theorem* essentially constructs randomised decision rules based on information obtained from the ROC. Randomised decision rules have not been widely used in problem domains such as medical decision making. We are proposing the application of such rules under the operating constraints faced in many real world domains: given a maximum allowable false positive rate, what is the best possible true positive rate attainable? For randomised decision rules in general, Berger [11] notes:

> There are reasons for studying randomised decision rules other than their usefulness in situations involving intelligent opponents. If it is desired to use a classical procedure with certain fixed error probabilities (say to fulfill contractual obligations), randomised rules may be necessary.

Experiments on the Thyroid and Grey classification problems indicate that this method is both applicable and feasible in real world applications such as feature selection. We are therefore able to provide an answer to

Question 6 *If we cannot find a suitable single feature subset for designing a particular classification system, what might we do about this?*

The *realisable classifier theorem* will allow us to modify the results of existing feature selection algorithms, combining multiple feature sets to produce classification systems that are robust in problem domains for which operating constraints may vary. However, in the next chapter we will see that this is not the end of the story. We will show how the *realisable classifier theorem*, and the $MRROC$ systems it can produce, can be used to form an objective function for a novel feature subset selection algorithm *Parcel*. *Parcel* is designed to selects multiple feature subsets, rather than just one, with the goal of producing classification systems that have the highest possible true-positive rates for all Neyman Pearson criteria. Empirical results will show that *Parcel* produces systems superior to those produced by a $SFFS$ wrapper in five of the seven real world problems examined.

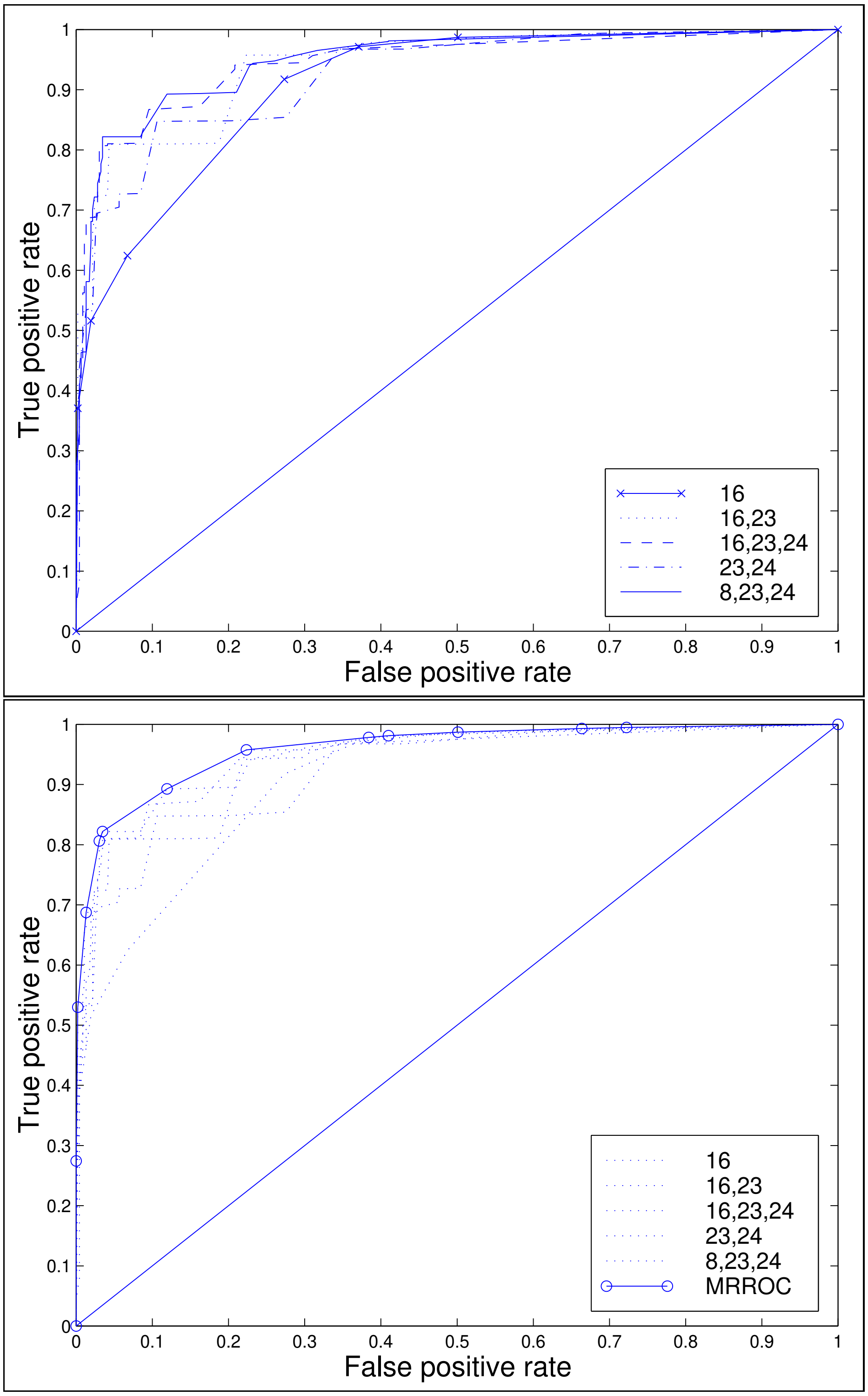
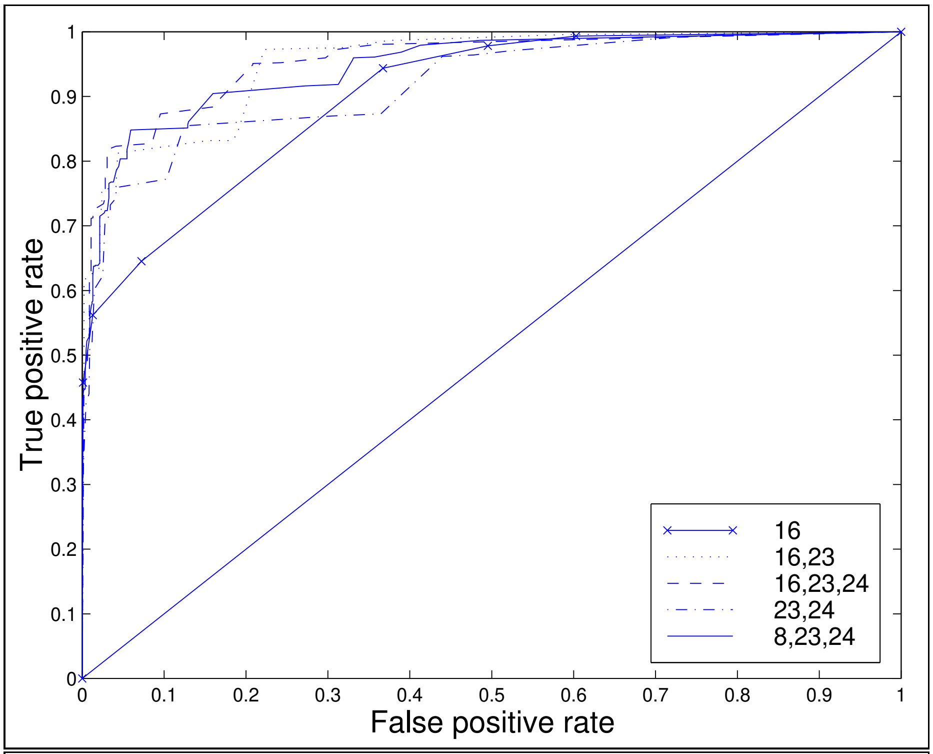


Figure 5.11: Top: the ROC curves for the five feature subsets produced using the validation data. Bottom: the $MRROC$ obtained using the validation ROCs



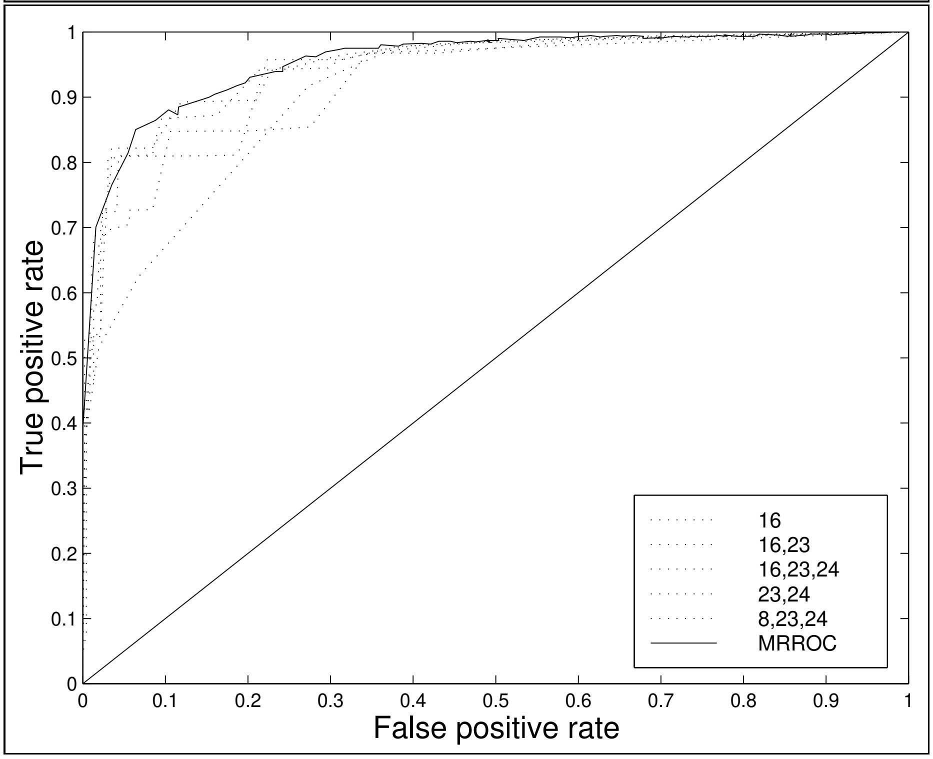


Figure 5.12: Top: the ROC curves for the five feature subsets produced using the hold-out data. Bottom: the $MRROC$ obtained on the hold-out data, by application of Theorem 1 to the existing classifiers found at the vertices of the validation data $MRROC$.

Chapter 6

Parcel

6.1 Introduction

Question 6 *If we cannot find a suitable single feature subset for designing a particular classification system, what might we do about this?*

In Chapter 4 we saw the effect on the feature selection process of varying misclassification costs: the single feature set selected by a $SFFS$ wrapper did not produce a superior system across all costs. Empirical results suggested that, in cases where costs might vary, one would require a classification system that employed multiple feature sets. Chapter 5 saw the introduction of the *realisable classifier theorem*, which allowed classifiers using distinct feature sets to be combined, hence the maximum realisable ROC ($MRROC$) was obtained. A real world example showed that the results of a $SFFS$ algorithm could be adapted with the *realisable classifier theorem*, retaining more than one feature set, and so produce a classification system that was robust to variable operating conditions. By adapting the $SFFS$ wrapper in this manner, a solution for Question 6was provided. However, this did not seem very satisfactory: intuitively it was felt that a more elegant solution could be engineered by considering variable cost criteria when designing the actual feature selection algorithm, rather than attempting an *a posteriori* adaptation of an existing algorithm.

In this chapter, we go beyond the modification of existing algorithms, rendering them suitable for variable operating conditions. A novel feature selection algorithm is presented, *Parcel*, which specifically caters for the design of classification systems in variable cost domains; *i.e.* those for which variable Neyman Pearson criteria[1] will be used to select an operating classifier. Rather than selecting one feature set, with reference to one set of costs, *Parcel* may select multiple feature sets with the aim of providing the best available true-positive rate for all possible Neyman Pearson criteria.

In the next section, an objective function is defined for feature selection in variable cost do-

[1] Where the Neyman Pearson criteria will specify the highest allowable false-positive rate.

mains. The *Parcel* algorithm, which attempts to satisfy this function, is then described. A description of an implementation of *Parcel* is given, and the results of its application to the Grey classification task are presented in detail, illustrating clearly the operation of the algorithm. Subsequently, *Parcel* is applied to the seven classification tasks described in Chapter 2, and the results produced by *Parcel* are compared to those produced by a *SFFS* wrapper. The empirical results suggest that the classification systems produced by *Parcel* are superior to those produced by $SFFS$. Given that *Parcel* selects multiple subsets, it is logical to consider what would happen were these subsets simply combined. A further set of empirical results suggest that, although the systems produced by such combinations are at times similar in performance to the *Parcel* systems, they are more complex, using larger feature sets. Therefore, in terms of feature selection, these combined systems are less desirable.

6.2 A novel feature selection objective criterion

We saw in the previous chapter that the $MRROC$ is achieved by taking the convex hull over all existing ROCs for a given classification problem. By retaining the classifiers found at each vertex and applying the *realisable classifier theorem*, all points on the hull can be obtained in practice. All the operating points on the facets of the hull, i.e. the points on the $MRROC$, are achievable by application of the *realisable classifier theorem* (see also Scott *et al*[99]).

A $MRROC$ is defined by a set of vertices, or points in ROC space. A vertex is defined by the classifiers and associated parameters producing it:

$$MRROC = \{\mathcal{C}(d; f_1; t_1), \mathcal{C}(d; f_2; t_2), .., \mathcal{C}(d; f_n; t_n)\}.$$

A classifier, $\mathcal{C}(d; f; t)$, produces a point in ROC space, i.e. a false- and true-positive rate (fp, tp). Three elements are required: a labelled dataset d, which contains both a training and validation set, a binary feature mask f, defining which features in d to use, and a threshold t, to use for making classifications. The algorithm employs the training data in d to train a classification system. Only the features in d that have a '1' in the corresponding bit of the feature mask are used.

Once a system has been trained, the examples in the validation data of d are mapped by the classification system to various real numbers in some interval. Hard classifications are made on each validation example by using the threshold t on the output of the classification system.

As d is a labelled dataset, the true and false positive rate of the classification algorithm on the validation data can be calculated. A ROC curve can be generated for a particular classification paradigm (for example, a neural network), and a given feature set, by estimating the true- and false-positive rates for all t (or some reasonable set of values, that results in both rates ranging between zero and one).

By definition, the $MRROC$ has the largest area of all ROCs considered, as it wholly contains all the ROCs considered. The *Scott-Niranjan-Prager* (SNP)[2] objective criterion can now be

[2]For want of a meaningless three letter acronym (TLA), author names were used.

stated for any system seeking to maximise true-positive rates across all false-positive rates, or Neyman Pearson criteria:

Objective 1 Scott-Niranjan-Prager objective criterion. *Seek individual classifiers that create points outside the known $MRROC$. Incorporating these points into the $MRROC$ will increase the Wilcoxon statistic, or area under the curve, for the system as a whole. It is guaranteed that, for any false-positive rate, the true-positive rate of the updated system is equal to, or greater than, that of the previous system.*

A feature selection algorithm can be driven using this objective criterion. Feature subsets that produce classifiers that lie outside the currently known $MRROC$ are saved (along with the classifier and threshold used to produce the new point). Such a system preserves the intuitive properties of using the $AUROC$, although the simplicity of selecting a single best subset of features is lost. This however, is not necessarily a failing, as in real world problems it may not be possible to discover a single subset of features that will produce the best performance over all costs. Seeking such a single subset may be an unrealistic goal, and may, as with the Grey and BrodSat classification problems in Chapter 4, lead to poor results across the entire range of operating costs. *Parcel* is described below, and provides a novel solution to Question 6

Question 6 *If we cannot find a suitable single feature subset for designing a particular classification system, what might we do about this?*

6.3 Algorithm description

It is the objective of the *Parcel* algorithm to produce a $MRROC$ that has the largest possible area underneath it, i.e. to maximise the Wilcoxon statistic associated with the classification system defined by the $MRROC$.

This is achieved by searching for, and retaining, those classifiers that extend the convex hull defined by the $MRROC$. It is not necessary to keep ROC points (classifiers) that lie within the hull, as these will at all times, and for all costs, be sub optimal to those that lie on the surface of the hull. The *Parcel* algorithm has two components with which to achieve its objective:

1. a classification algorithm, $\mathcal{C}^* \to \mathcal{C}(d; f; t)$, and
2. a search strategy, $\mathcal{S}(d; f; \mathcal{C}^*, MRROC^{old}) \to MRROC^{new}$.

The classification algorithm is the set of rules or methods that produce classifiers for a given set of parameters $\{d, f, t\}$.

The search strategy component, $\mathcal{S}(d; f; \mathcal{C}^*, MRROC^{old})$, defines the manner in which classifiers that improve the $MRROC$ are sought. As parameters it also takes the data d, and a feature mask f. In addition, it takes a classification algorithm $\mathcal{C}^*$, and a $MRROC$, which, during searching, is denoted $MRROC^{old}$. This is the best $MRROC$ currently known: it is the task of the search strategy to improve upon it.

The search strategy is initialised with the feature mask f, and generates one or more new feature masks, f_i, derived from the original f. For each new feature mask f_i, a ROC curve, ROC_i, with n points, is generated by creating n classifiers $\mathcal{C}(d; f_i; t_j)$, varying t_j through a range of thresholds, $t_j \in \{t_1, ..t_n\}$, as described above.

A point on ROC_i, (fp_i, tp_i), is defined by, and stored as, the classifier that produced it: $\mathcal{C}(d, f_i, t_j)$. All the points from each ROC_i are combined together with the points in $MRROC^{old}$, producing a large set of points, R, in ROC space.

A convex hull is formed over R, and the points corresponding to the vertices of this hull are saved. The hull is $MRROC^{new}$. If

$$MRROC^{new} \neq MRROC^{old},$$

then, by definition,

$$MRROC^{old} \subset MRROC^{new}.$$

This means that the convex hull, $MRROC^{old}$, has been improved, and $MRROC^{new}$ should be returned as a superior curve to $MRROC^{old}$. If the new $MRROC$ contains the old, then it has, by definition, a larger area beneath it. As the area beneath the curve equates to the Wilcoxon statistic, which we desire to maximise, and the curves do not cross, the new curve is superior to the old. Searching should now continue, using the feature masks contained in the updated $MRROC^{new}$. If, on the other hand,

$$MRROC^{new} = MRROC^{old},$$

then $\mathcal{S}(d; f; \mathcal{C}^*, MRROC^{old})$ has failed to find any classifiers that lie outside $MRROC^{old}$, and searching stops. Thus the *Parcel* algorithm can be summarised as so:

1. $\forall\, \mathcal{C}(d; f_{vertex}, t_{vertex}) \in MRROC^{old}$
   $\mathcal{S}(d; f_{vertex}; \mathcal{C}^*, MRROC^{old}) \to MRROC^{new}$

2. if $MRROC^{old} = MRROC^{new}$ stop *Parcel*

3. $MRROC^{old} \leftarrow MRROC^{new}$

4. goto 1.

Figure 6.1 illustrates the operation of *Parcel*.

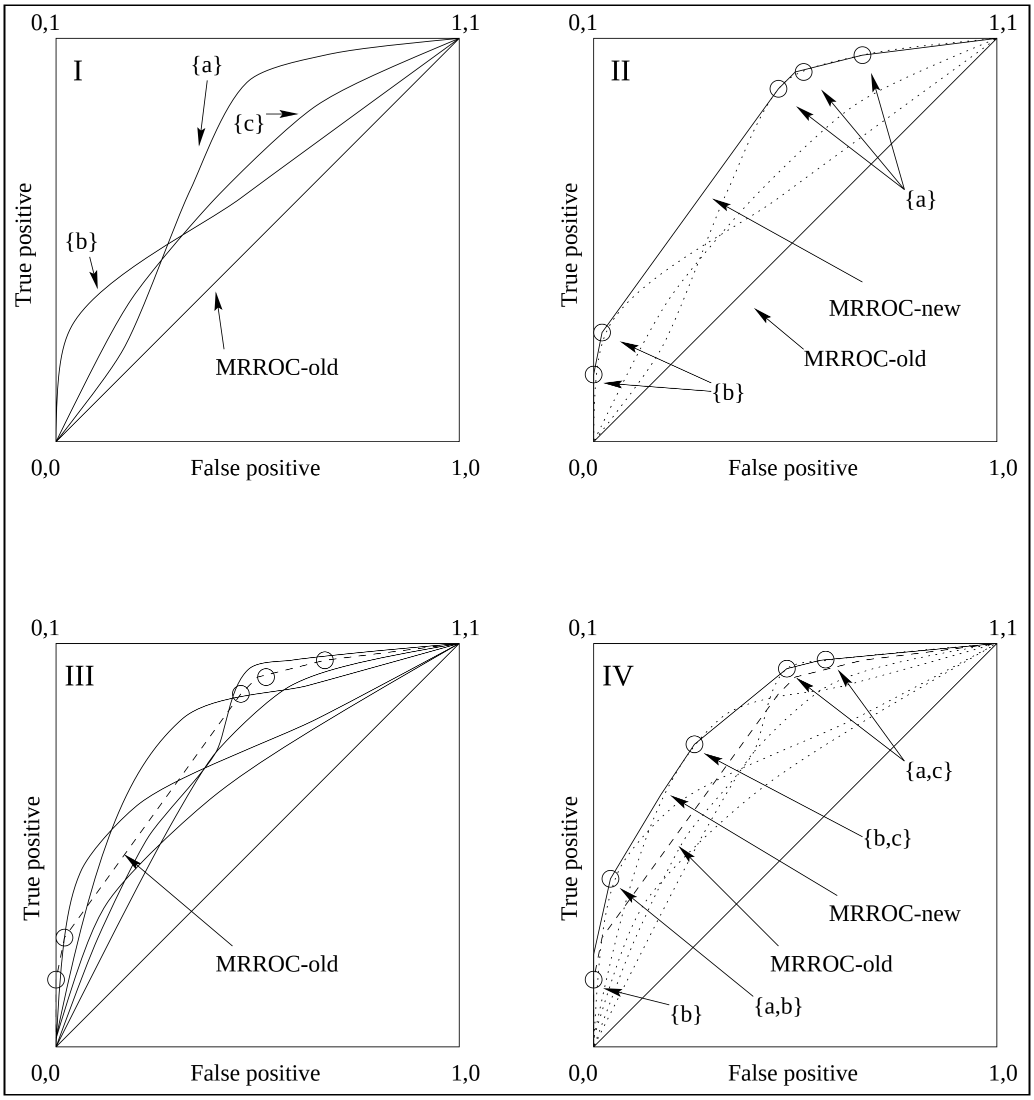


Figure 6.1: Two cycles of the operation of *Parcel*. The objective is to find a $MRROC$ for a problem with a data set described by the features $\{a, b, c\}$. **I.**The ROC curves produced using a single feature. The $MRROC^{old}$ is the diagonal. **II.** The convex hull, $MRROC^{new}$, over the curves has five vertices. Two use feature subset $\{b\}$, and three use $\{a\}$. $MRROC^{new}$ differs from $MRROC^{old}$, indicating that the *SNP* objective criterion is satisfied, so the algorithm proceeds. **III.** The search algorithm takes the feature subsets from the vertices of $MRROC^{old}$ and searches for new classifiers. Some of the ROC curves found by the search algorithm are plotted over $MRROC^{old}$. **IV.** $MRROC^{new}$ containing the new curves and $MRROC^{old}$ has five vertices. Two use feature subset $\{a, c\}$, the others using $\{b\}$, $\{a, b\}$ and $\{b, c\}$. $MRROC^{new}$ differs from $MRROC^{old}$, hence the *SNP* objective criterion is satisfied, and the algorithm continues.

6.4 An implementation of *Parcel*

The pseudo code of the implementation of *Parcel* used in this chapter is given in Figure 6.2. The naive Bayes classification algorithm is used in this chapter. In the pseudo code, the set of thresholds to be used for classification is kept in `thresholds`.

The *Parcel* algorithm commences with a $MRROC$ consisting of a single "dumb" classifier (producing the diagonal line in the ROC space). This classifier has the empty set of features, and uses an arbitrary threshold of .5. This classifier represents the performance one would obtain on the dataset `data` by guessing [45, 105].

```
MRROC^old ← {C(data; ∅; .5)}
thresholds ← {t_1, .., t_m}
WHILE( SNP )
        FORALL vertex ∈ MRROC^old
            feat-mask ← f_vertex
            FOR(i = 1 to number of features)
                feat-mask[i] ← not(feat-mask[i])
                FOR(j = 1 to m)
                    roc ← roc ∪ C(data, feat-mask, t_j)
                ENDFOR
                feat-mask[i] ← not(feat-mask[i])
            ENDFOR
        ENDFORALL
        roc ← roc ∪ MRROC^old
        MRROC^new ← ConvexHull(roc)
        IF SigDiff(MRROC^new, MRROC^old)
            THEN MRROC^old ← MRROC^new
                SNP ← TRUE
        ELSE SNP ← FALSE
        ENDIF
ENDWHILE
```

Figure 6.2: Pseudo code for the *Parcel* algorithm. The function `ConvexHull(pts)` returns the points forming the convex hull over `pts`. The function `SigDiff(a,b)` tests to see whether there exists a significant difference between `a` and `b`.

In the implementation given in Figure 6.2, the search strategy employed by $\mathcal{S}(d; f; \mathcal{C}(\cdot;\cdot;\cdot), MRROC^{old})$ is *sequential forwards selection* (SFS) [56]. Given a feature subset f, with n elements, add or delete individual features from the subset to produce n new subsets. In pseudo code, this corresponds to flipping the bit of the feature mask corresponding to the individual feature to be added or deleted. A feature set obtained by flipping one bit of the feature mask is called a *child* of f. If there are n features, there will be n children of f.

Within the `WHILE` loop of the code, the algorithm selects each feature set $f_{\texttt{vertex}}$ from the classifier corresponding to each `vertex` in $MRROC^{old}$. Using SFS, the n children of $f_{\texttt{vertex}}$ are found. For each, an ROC is computed. The points of the n ROC curves are combined in one large set, along with the points in $MRROC^{old}$.

A convex hull, $MRROC^{new}$ is formed over these points. If $MRROC^{new}$ differs significantly from $MRROC^{old}$, then the *SNP* objective has been increased. $MRROC^{old}$ is set to be equal to $MRROC^{new}$, and the loop repeats. The question of what, in practice, constitutes a significant difference will be implementation dependent. The method used in this chapter is described in Appendix A

If $MRROC^{new}$ and $MRROC^{old}$ do not significantly differ, this indicates that each vertex has been visited, SFS has been carried out using the feature subsets at each vertex, and no classifiers have been discovered that improve upon $MRROC^{old}$. In this case, the *SNP* objective has not been increased, so the loop, and thus the algorithm, terminates. At this point, the $MRROC$ given by $MRROC^{old}$ is the best possible to achieve using SFS, `data`, and the naive Bayes classification algorithm.

6.4.1 *Parcel* applied to the Grey problem

To illustrate the operation of *Parcel*, the implemented algorithm was applied to the Grey classification problem. The *Parcel* algorithm employed a naive Bayes classifier, and used a forwards selection (FS) search algorithm; i.e. candidate feature sets were generated by the addition of single features to existing feature sets.

In Figure 6.3, it can be seen that the $MRROC$ produced by *Parcel* uses thirteen different feature subsets, one at each vertex. In practice, any classifier on the $MRROC$ will require at most two of these subsets to be used, *i.e.* one subset at each vertex on either side of the classifier.

Parcel f_1	$\{0, 1, 2\}$	*Parcel* f_8	$\{1, 27, 28\}$
Parcel f_2	$\{18, 20, 35\}$	*Parcel* f_9	$\{17, 27, 28\}$
Parcel f_3	$\{7, 12, 19\}$	*Parcel* f_{10}	$\{11, 27, 28\}$
Parcel f_4	$\{11, 16, 31\}$	*Parcel* f_{11}	$\{27, 28, 31\}$
Parcel f_5	$\{24, 26, 32\}$	*Parcel* f_{12}	$\{26, 27, 28\}$
Parcel f_6	$\{13, 24, 26\}$	*Parcel* f_{13}	$\{14, 15, 16\}$
Parcel f_7	$\{27, 28, 29\}$		

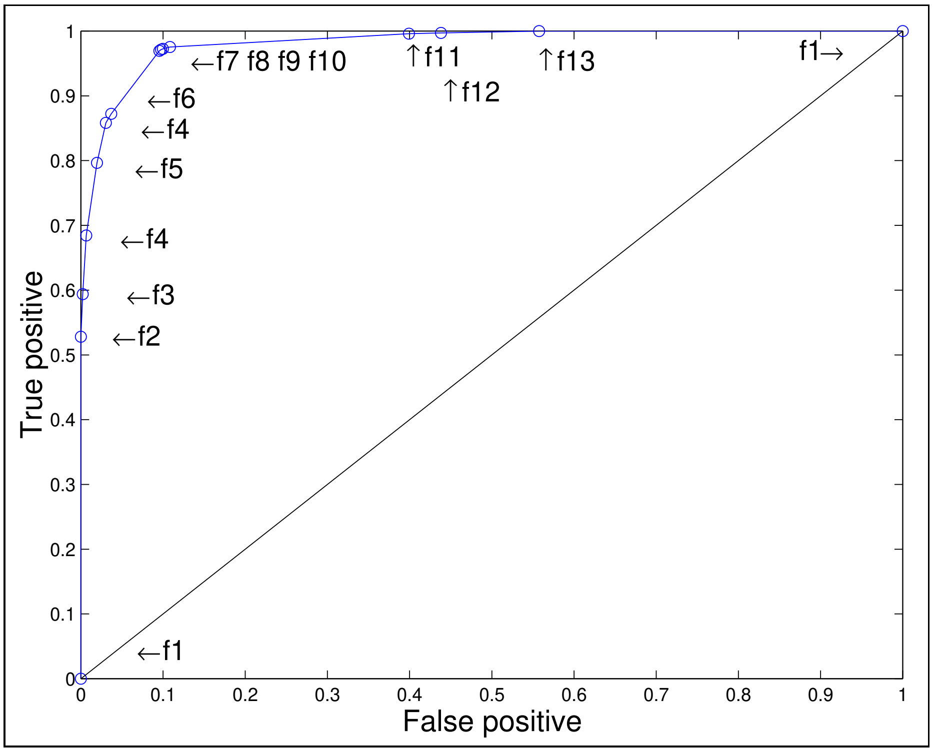


Figure 6.3: The *ROC* produced by *Parcel* on the Grey validation data set. Each vertex label indicates the feature subset used to produce that vertex.

6.5 Empirical results

We have seen in the previous chapter that classification systems employing multiple feature subsets can, potentially, be more robust to variable Neyman Pearson criteria than those systems that use just one feature subset. Furthermore, we have now developed a feature selection algorithm, *Parcel*, that is able to select multiple subsets that are appropriate for such cost-variable environments.

In this section, two questions are posed, and we seek empirical evidence upon which we might base answers. The questions are:

Question 7 *How do classification systems designed using* Parcel *compare with those designed using a* $SFFS$ *wrapper?*

Question 8 *Given that a classification system designed using* Parcel *employs a number of feature subsets, how does it compare with a system that uses a single subset, where that single subset is made up of the union of all the* Parcel *subsets?*

In presenting empirical results we required a means of comparing two systems. We have already seen that comparison on the basis of $AUROC$ alone can be misleading. However, $AUROC$ is still a useful metric, especially in the case where the curves do not cross, in that we may test for significance in the difference between two areas. We wished to achieve a balance of interpretability and accurate representation of the results. Therefore, when presenting the results of a comparison between two classification systems

- the comparison is made using hold-out data sets, and so is unbiased;
- the ROC curves for both systems are plotted together, so that any crossing or touching of the curves can be seen;
- the $AUROC$s are compared, and any difference tested for statistical significance, to a 95% confidence interval.

The seven classification problems described in Chapter 2, were used here: Adult, BrodSat, Cotton, Field, Grey, Thyroid, and Tree. Each problem had three data sets, training, validation and hold-out.

Classification problem	training set	validation set	hold-out set
Adult	20108	10054	15060
BrodSat	1206	603	500
Cotton	2956	1478	2000
Field	6000	3000	3392
Grey	2956	1478	2000
Thyroid	2514	1257	3413
Tree	6000	3000	3392

Table 6.1: The data sets used in the empirical evaluation of the *Parcel* algorithm. The number of cases in each data set is given.

Dataset	$AUROC$ $SFFS$	$AUROC$ *Parcel*	increase	z ratio	significant?
Adult	**0.9116**	**0.9007**	**+0.0109**	**3.2891**	**Yes**
BrodSat	0.9892	0.9918	+0.0025	0.4423	No
Cotton	**0.9437**	**0.9671**	**+0.0234**	**2.4295**	**Yes**
Field	**0.9022**	**0.9151**	**+0.0129**	**1.7353**	**Yes**
Grey	**0.9461**	**0.9694**	**+0.0233**	**3.2395**	**Yes**
Thyroid	0.9936	0.9938	+0.0002	0.0271	No
Tree	**0.8863**	**0.9150**	**+0.0287**	**2.8779**	**Yes**

Table 6.2: The $AUROC$ for the classification systems produced by the $SFFS$ wrapper and *Parcel*; the ROC curves were calculated on hold-out data sets, not available during training or feature selection. The increase is the increase in $AUROC$ one would get by using the *Parcel*. A z ratio value greater than 1.69 indicates that the increase is statistically significant, to a 95% confidence interval. Rows in bold-face indicate a statistically significant improvement when using *Parcel*.

6.5.1 A comparison of *Parcel* and a $SFFS$ wrapper

This experiment will compare the results of a $SFFS$ wrapper with the *Parcel* algorithm. The $SFFS$ wrapper is the same as that used in Chapter 4, with an objective function based on the accuracy of a naive Bayes classifier.

For each of the seven classification problems considered, feature selection was carried out using the training and validation sets. The $SFFS$ wrapper selects a single feature set, based on maximising the classification accuracy on the validation data set. Having trained a naive Bayes classifier with this feature set and the training data, one can calculate an unbiased ROC by presenting the hold-out data to the trained system, and varying the threshold used for classification, as described in detail in Chapter 4.

An unbiased ROC can be calculated for the system produced by *Parcel*, again by using the hold-out data set. The classification system is defined by a number of classifiers with fixed classification thresholds: these are the vertices of a convex hull, the area of which is maximised by *Parcel*. The true- and false-positive rates of each of these classifiers is evaluated using the hold-out data. This will yield a number of points in ROC space; these points are unbiased estimates of the locations of the vertices of the convex hull. Every other point in the *Parcel* system will lie on the straight lines joining these points, and can be obtained using the *realisable classifier theorem* (see Chapter 5).

For each classification problem, a single feature subset is selected by the wrapper and a series of the feature sets selected by *Parcel*; these subsets are given in Appendix B. The ROC curves of the classification systems produced by *Parcel* and the $SFFS$ wrapper are plotted together for inspection. Table 6.2 gives value of the $AUROC$ for each problem, the increase in $AUROC$ obtained by using *Parcel* rather than the wrapper, and the z value indicating whether this increase is statistically significant; $z \geq 1.69$ indicates statistical significance to a 95% confidence interval, as described in Appendix A.

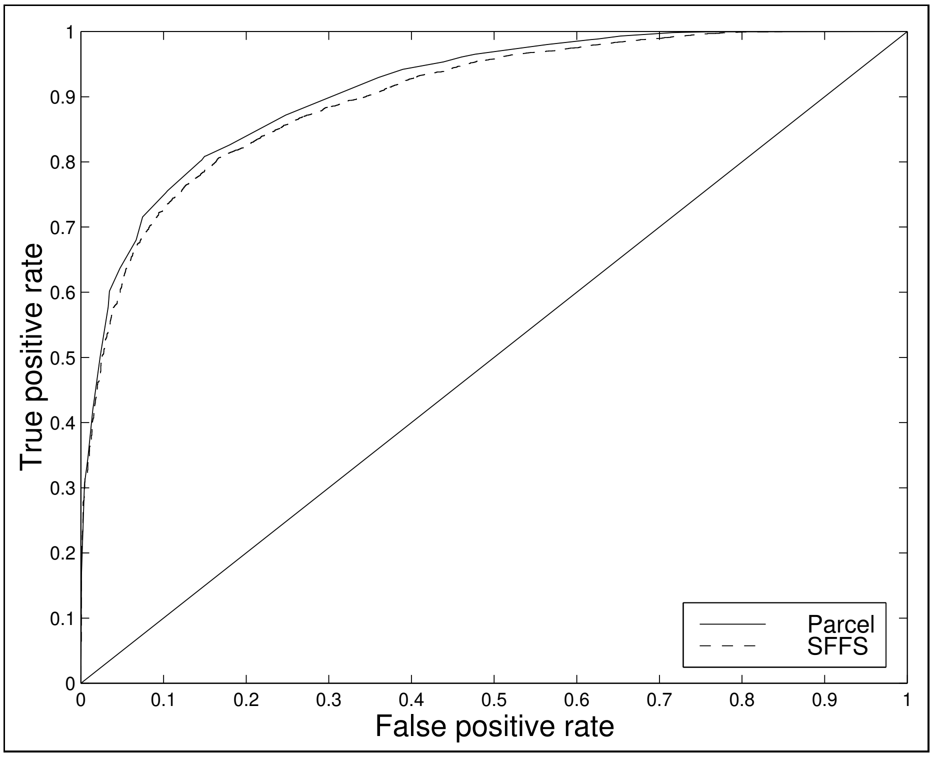


Figure 6.4: The Adult classification task. The ROC curves produced by the *Parcel* and *SFFS* wrapper algorithms on the unseen data set. As this data was not available during selection, these results will be unbiased.

The classification system produced by *Parcel* has a significantly larger $AUROC$ than the system produced by the $SFFS$ wrapper in five of the seven problems examined; on the remaining two there is no statistically significant difference between the $AUROC$s. In plotting the curves we can see that, for each of the seven problems, the ROC curve of the *Parcel* system is either dominant (uppermost, and therefore superior) or approximately co-linear with the curve of the system produced by the $SFFS$ wrapper.

Question 7 *How do classification systems designed using* Parcel *compare with those designed using a* $SFFS$ *wrapper?*

It would be incorrect to conclude that *Parcel* has produced systems that are at worst equivalent to those of the $SFFS$ wrapper. In the two cases where no significant improvement was gained in $AUROC$, we could conclude that the $SFFS$ wrapper was superior. The reason for this was that, in these two cases, a smaller feature set produced the same results; in terms of the motivation for feature selection, this qualifies as an improvement. Despite this, we can conclude that, overall, *Parcel* produced better systems with regard to the problems considered.

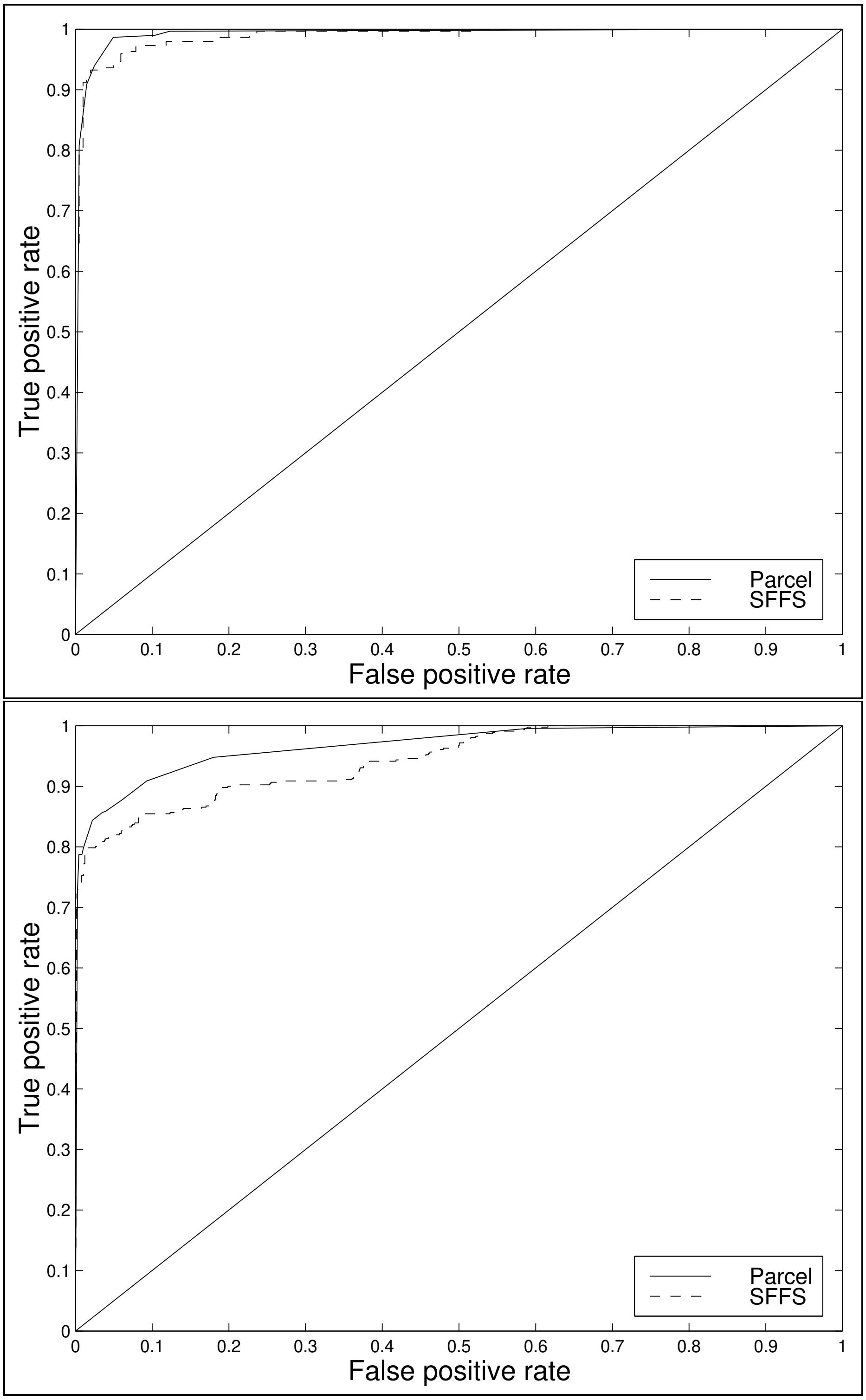


Figure 6.5: Top: The BrodSat classification task. The *ROC* curves produced by the *Parcel* and *SFFS* wrapper algorithms on the unseen data set. As this data was not available during selection, these results will be unbiased. Bottom: The Cotton classification task.

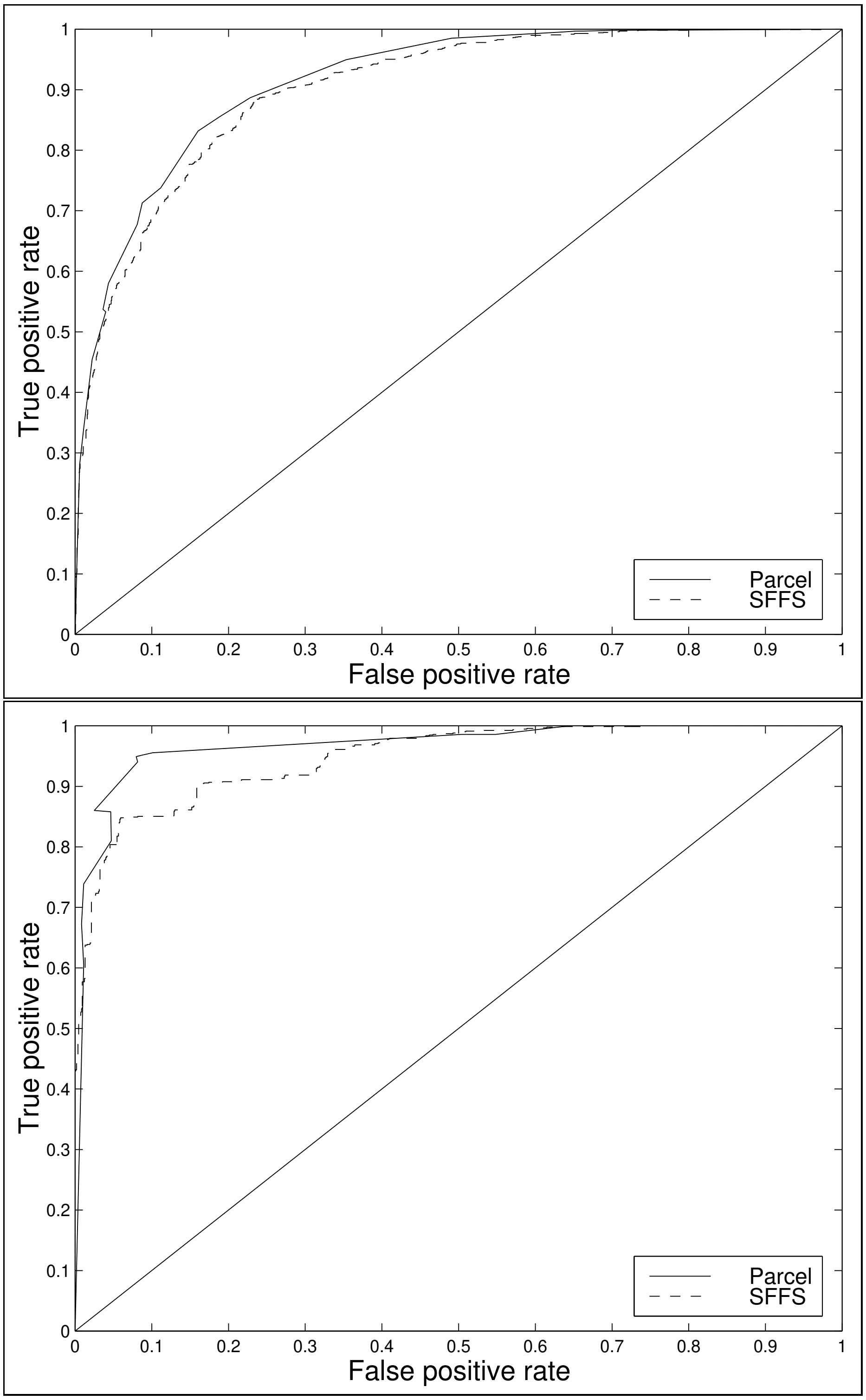


Figure 6.6: Top: The Field classification task. The *ROC* curves produced by the *Parcel* and *SFFS* wrapper algorithms on the unseen data set. As this data was not available during selection, these results will be unbiased. Bottom: The Grey classification task.

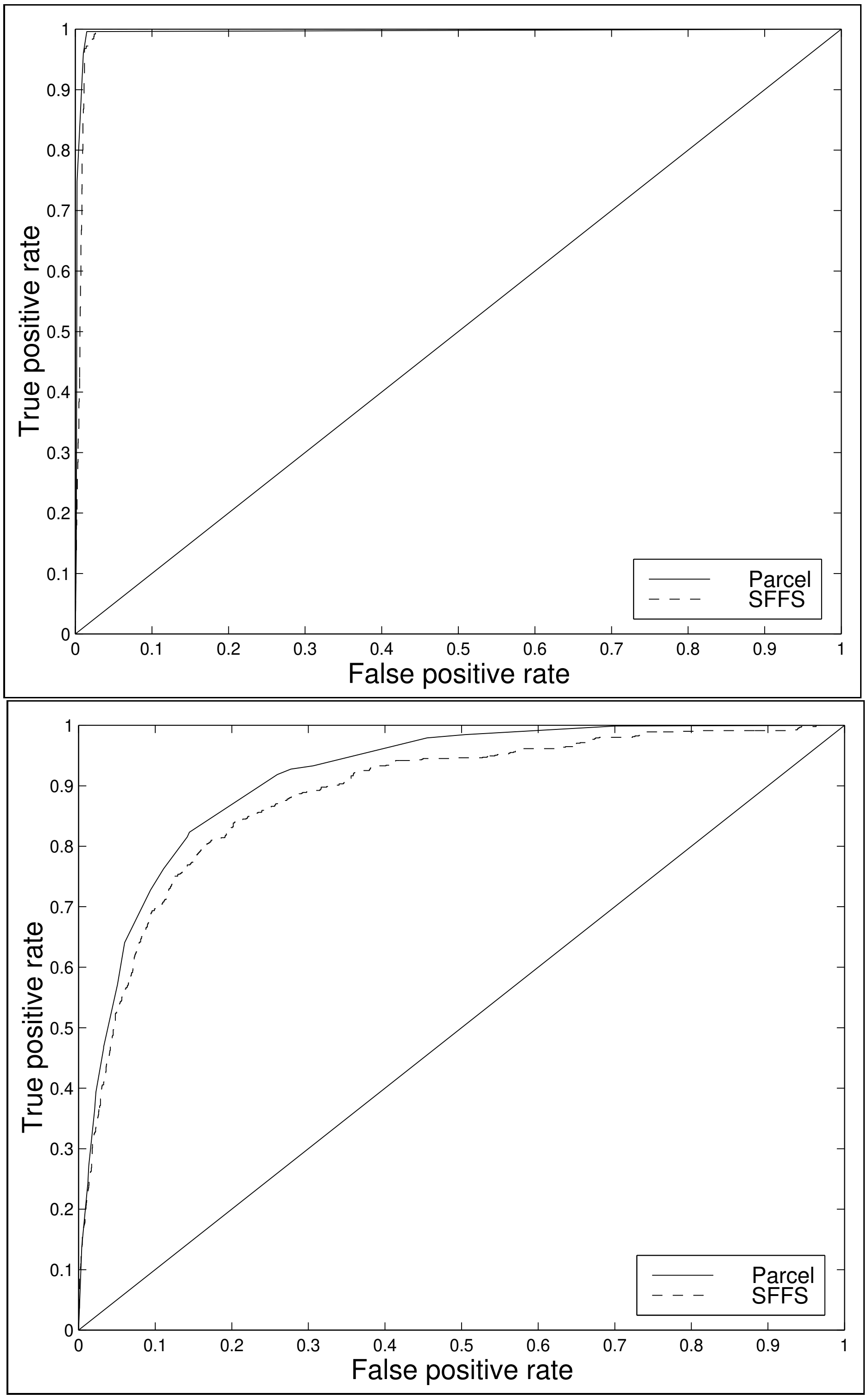


Figure 6.7: Top: The Thyroid classification task. The *ROC* curves produced by the *Parcel* and *SFFS* wrapper algorithms on the unseen data set. As this data was not available during selection, these results will be unbiased. Bottom: The Tree classification task.

Dataset	Combined *Parcel* feature sets	size
Adult	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12\}$	12
BrodSat	$\{0, 1, 2, 3, 5, 9, 10, 11, 12, 14, 18\}$	11
Cotton	$\{1, 2, 5, 9, 13, 17, 21, 25, 32, 33\}$	10
Field	$\{0, 1, 3, 4, 5, 8, 9, 10, 11, 12, 14, 15, 16, 17\}$	14
Grey	$\{0, 1, 2, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 26, 27, 28, 29, 31, 32, 35\}$	22
Thyroid	$\{0, 1, 2, 3, 7, 16, 17, 18\}$	8
Tree	$\{0, 1, 2, 3, 4, 5, 8, 9, 10, 11, 13, 14, 17\}$	13

Table 6.3: For each classification problem, a single feature set is formed by combining the multiple sets found by *Parcel*.

6.5.2 Combination results

Examining the results of the *Parcel* algorithm, a logical question[3] to ask is:

Question 8 *Given that a classification system designed using* Parcel *employs a number of feature subsets, how does it compare with a system that uses a single subset, where that single subset is made up of the union of all the* Parcel *subsets?*

To provide an answer for this question, the feature subsets, selected by *Parcel* for each of the seven classification problems examined, were combined into seven single subsets, see Table 6.3. For each classification problem, a naive Bayes algorithm was trained using the single combination feature set. An ROC curve was then obtained using the hold-out data set for that problem. These seven ROC curves were then plotted and compared with the respective ROC curves from the *Parcel* systems. The $AUROC$ for the combined feature set systems and the *Parcel* systems are given in Table 6.4. The increase obtained by using the *Parcel* system is also given, and the z value indicating whether this is statistically significant.

On five of the seven problems, there was no significant difference between using the *Parcel* system or the combined system. On the remaining two, the *Parcel* system was significantly better. Although the $AUROC$ was improved on only two of the seven problems, there is another motivation for using the *Parcel* systems. Recall our initial motivation for carrying out feature selection

> "*General motivation*. We are given the task of designing a classification system, for which the available data have a large number of dimensions. We need to select an appropriate subset of features, from the large number available. Furthermore, we wish to select the smallest possible appropriate subset."

As the combined system is made up of the various subsets used in the *Parcel* system, it will, by definition, use at least the same number features, if not more. Any classifier chosen from the

[3]The author would like to thank Prof. David Hand for suggesting this question.

Dataset	*AUROC Combined*	*AUROC Parcel*	increase	z ratio	significant?
Adult	0.9159	0.9116	-0.0043	1.3459	No
BrodSat	**0.9506**	**0.9918**	**$+0.0411$**	**4.0650**	**Yes**
Cotton	0.9681	0.9671	-0.0011	0.1292	No
Field	**0.8980**	**0.9151**	**$+0.0170$**	**2.2732**	**Yes**
Grey	0.9751	0.9694	-0.0057	1.0357	No
Thyroid	0.9941	0.9938	-0.0003	0.0599	No
Tree	0.9016	0.9150	$+0.0134$	1.3958	No

Table 6.4: The $AUROC$ for the classification system produced by combining all the subsets found by *Parcel*, and for the *Parcel* system itself; the ROC curves were calculated on hold-out data sets, not available during training or feature selection. The increase is the increase in $AUROC$ one would get by using the *Parcel*. A z ratio value greater than 1.69 indicates that the increase is statistically significant, to a 95% confidence interval. Rows in bold-face indicate a statistically significant improvement when using *Parcel*.

ROC curve of the *Parcel* system will require at most two of the subsets to be used. Therefore, in terms of lowering the cost of gathering features during operation, the *Parcel* system will be superior to the combined system. This fact, combined with the $AUROC$ results, lead to the conclusion that using the *Parcel* system is preferable to using the combined system.

6.6 Conclusions

This chapter opened with the question

Question 6 *If we cannot find a suitable single feature subset for designing a particular classification system, what might we do about this?*

A priori, a solution to this question was provided by the results obtained in Chapter 5: adapting a $SFFS$ wrapper using the *realisable classifier theorem* produced a multi-feature set classification system, robust to variable operating criteria. However, it was felt that a better solution might be found. It had been determined that a good solution to Question 6would probably involve the use of multiple feature sets, and a new technique for combining classifiers built with distinct feature sets had been developed. It seemed logical to use this technique to engineer an algorithm that specifically catered for problems with variable Neyman Pearson criteria, rather than simply attempting to "fix" the results of an algorithm that did not. The algorithm developed in this way was named *Parcel*.

The SNP objective criterion provides the base upon which the *Parcel* algorithm is constructed: maximisation of the $MRROC$, or convex hull over all currently known operating points. *Parcel* was designed to allow for the selection of multiple feature sets, although not constrained to do

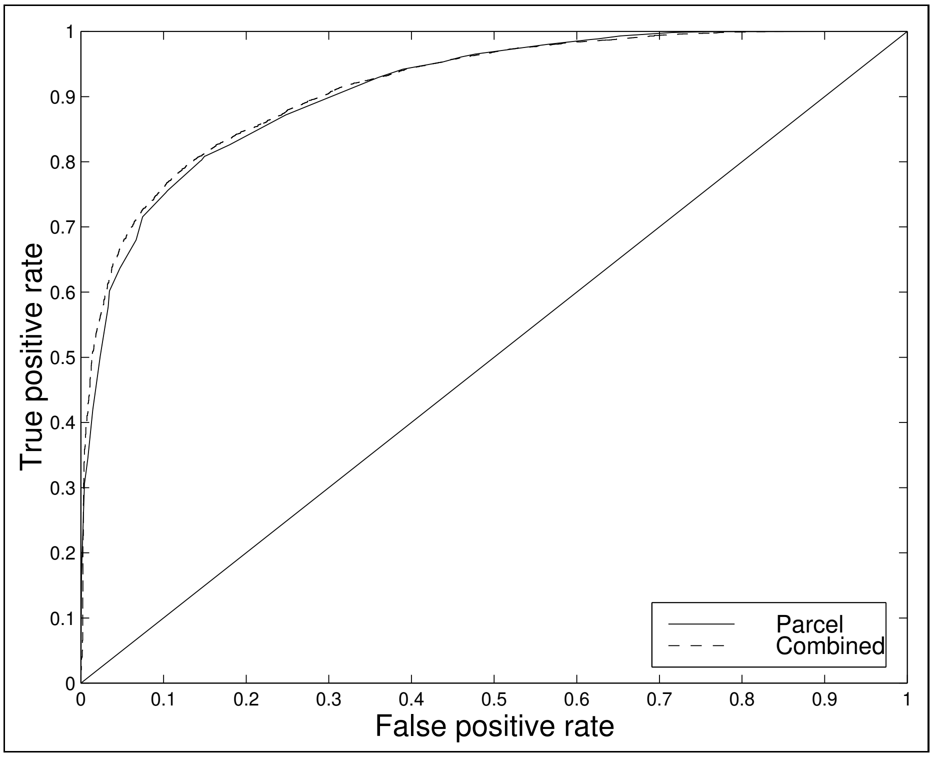


Figure 6.8: The Adult classification task. The ROC curves produced by the *Parcel* algorithm and the system produced using all of the *Parcel* subsets combined. The unseen data used to produce was not available during feature selection, hence these results will be unbiased.

this; if only a single subset was required to satisfy the SNP objective, then only that subset would be used.

Rather than constructing artificial data sets, the seven real world sets examined in Chapter 4 were used to test *Parcel*. In Chapter 4, a $SFFS$ wrapper was identified, on theoretical and empirical grounds, as a good feature selection algorithm. It was decided to compare the results of *Parcel* to the $SFFS$ wrapper, on the basis of an unbiased estimate of $AUROC$, and by plotting the ROC curves of each classification system together, to highlight any crossing points. On five of the seven data sets examined, *Parcel* produced statistically significant improvements over the wrapper in terms of $AUROC$; on the remaining two they were equivalent in terms of performance, but the $SFFS$ wrapper produced simpler systems. On all of the data sets, the ROC curve of the *Parcel* system was either dominant (uppermost) or approximately co-linear with that of the wrapper.

It was also decided to compare the systems produced by *Parcel* to the system one would obtain by combining all of the subsets that *Parcel* found. In this instance, using the *Parcel* system provided an improvement in two of the seven problems; and was equivalent in the remaining five. When in operation, however, at any given time, the *Parcel* systems would be using fewer features, and so, given the motivation for feature selection, they were considered superior.

In this chapter, *Parcel* was implemented using a naive Bayes classification algorithm, and a forwards selection (FS) search heuristic. Theoretically, *Parcel* can be used with any type of classification algorithm, and a logical progression of the work presented here would be to implement and test *Parcel* with other algorithms, such as the $C4.5$ decision tree [93] or a neural network algorithm, such as those detailed by Bishop[12] and Ripley[96]. It would be expected that implementations of *Parcel* that use such classification algorithms could incur heavy computational overheads: a naive Bayes classifier can be trained very quickly, but a back propagation neural network might take far longer.

The strategy or heuristic used to search for new subsets of features could also be investigated. Examination of the results of different search heuristics might well indicate a heuristic superior to the one used in this chapter, FS. FS starts with an empty set, and performs a sub-optimal search, so one might at least expect different, if not superior, results given a stepwise or floating search. Perhaps the most interesting avenue of future research is to use not one classification algorithm, but many. The *realisable classifier theorem* treats the classifiers to be combined as black boxes, hence classifiers based on distinct algorithms can be combined. For example, a decision tree using feature set S_1 and a neural network using feature set S_2 might be combined.

I believe that the results in this chapter indicate a novel direction for feature subset selection. Seeking a single best subset is appropriate for problems with a fixed set of operating costs. In many real world problems operating criteria can vary, and a classification system that makes use of multiple feature sets may be more appropriate, as it might not be possible to find a single best feature set. In such situations, the *Parcel* algorithm can produce useful results, that are not only easily implemented, but will allow an end user to set dynamically the operating criteria for the system.

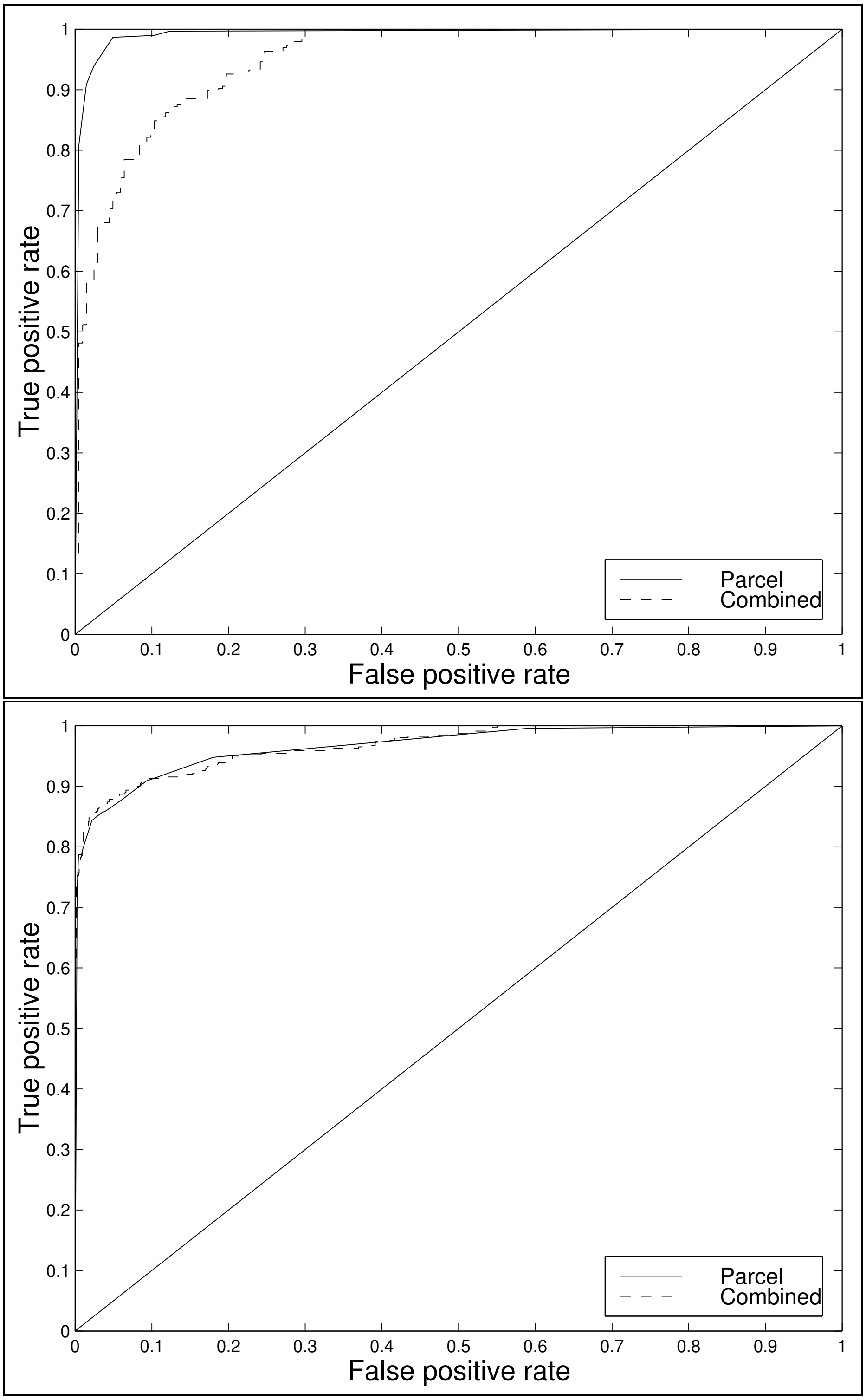


Figure 6.9: Top: The BrodSat classification task. The *ROC* curves produced by the *Parcel* algorithm and the system produced using all of the *Parcel* subsets combined. The unseen data used to produce was not available during feature selection, hence these results will be unbiased. Bottom: The Cotton classification task.

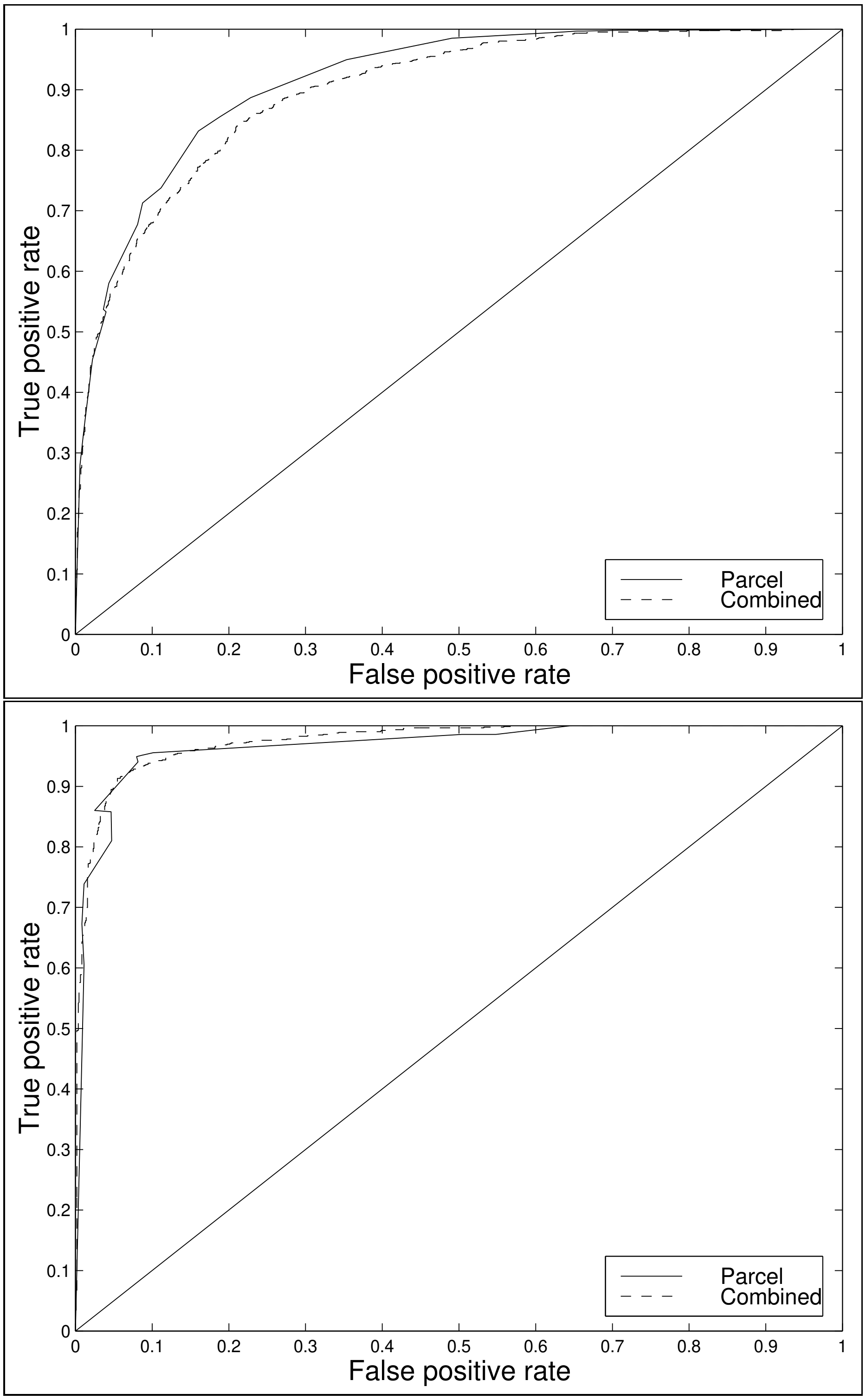


Figure 6.10: Top: The Field classification task. The *ROC* curves produced by the *Parcel* algorithm and the system produced using all of the *Parcel* subsets combined. The unseen data used to produce was not available during feature selection, hence these results will be unbiased. Bottom: The Grey classification task.

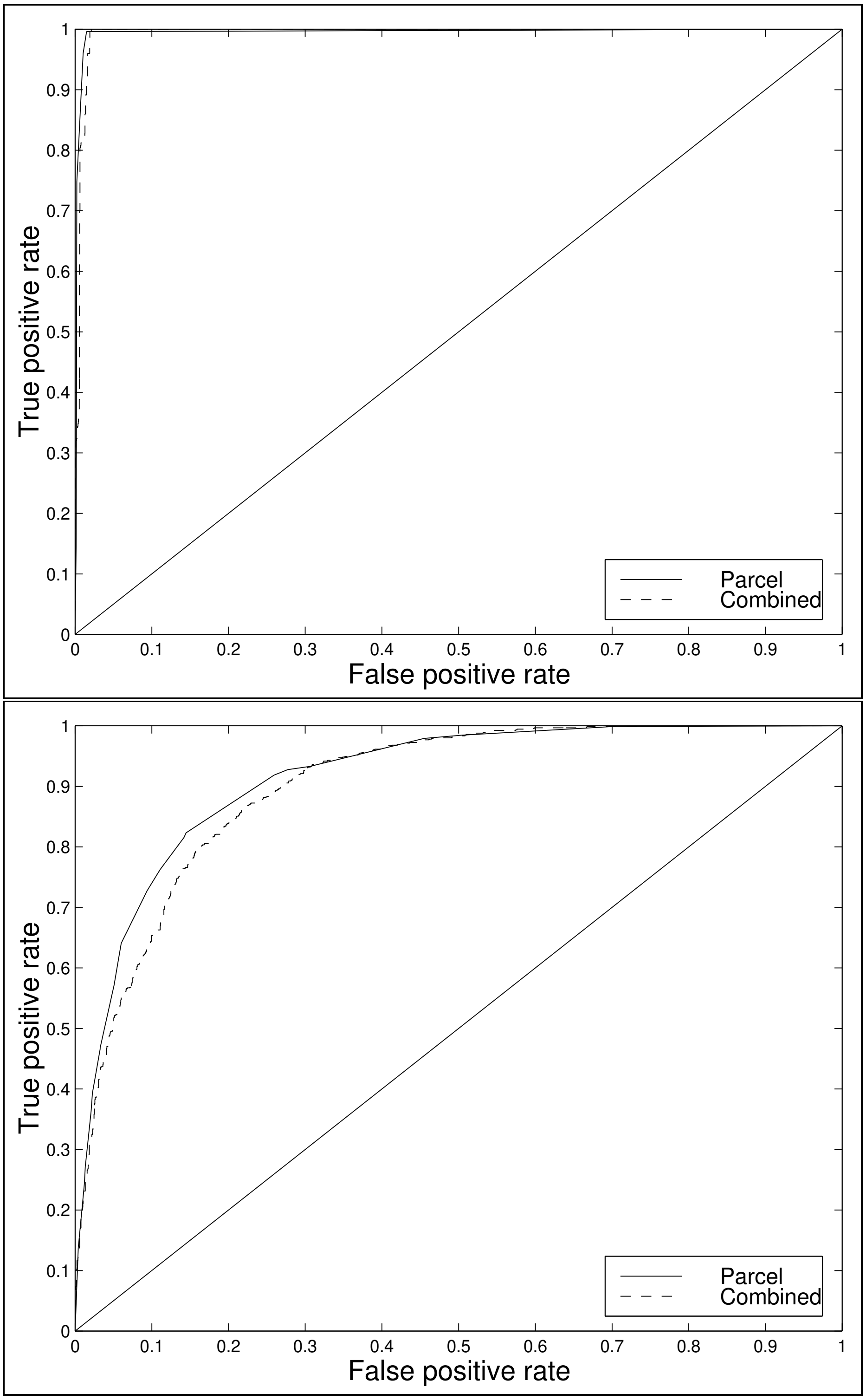


Figure 6.11: Top: The Thyroid classification task. The *ROC* curves produced by the *Parcel* algorithm and the system produced using all of the *Parcel* subsets combined. The unseen data used to produce was not available during feature selection, hence these results will be unbiased. Bottom: The Tree classification task.

Chapter 7

Conclusions

The *Parcel* algorithm is a departure from the classical feature subset selection genre, in which a single feature subset is sought to maximise the performance of a classification system. Although theoretically it is possible to obtain such a single subset, in practice it has been shown that the subset chosen will be highly dependent upon the classifier used [59].

It has been shown in this paper that, in addition to a dependence upon the classifier used, the subset selected will also be effected by the misclassification costs assigned to the problem. A subset that is picked to optimise the combined performance of a particular classifier for a particular set of costs, may produce poor results if the costs are varied.

Many real world problems, such as medical diagnosis tasks, have variable or undefined (at the time of system design) costs. Given this, and the subsequent effect of cost variation upon the system produced, algorithms that produce systems which are designed for one set of costs may be vulnerable to degradation in performance over a range of operating costs, when compared to other available systems. In Chapter 4 a classification systems for two real world problems were produced by selecting a single feature subset with a *wrapper* algorithm (optimising for $0/1$ loss), for use with a naive Bayes classifier. In both cases, the *wrapper* rejected a number of candidate subsets during optimisation, and it was shown that some of these rejected subsets produced superior naive Bayes classifiers given a subsequent change of misclassification costs.

The *Parcel* algorithm seeks not to select a single best feature subset, but rather to select as many different subsets as are necessary to produce satisfactory performance across *all* costs. Making novel use of ROC analysis, and the *realisable classifier theorem* [98], the SNP objective criterion of *Parcel* maximises the $AUROC$.[1] Satisfying the SNP criterion also justifies the use of the $AUROC$ as a metric of system performance and comparison, which would otherwise require qualification regarding crossing or touching ROC curves. Empirical results have been presented to show that the classification systems produced using an implementation of *Parcel* will be at least as good as, if not better, than those produced using the state of the art *wrapper* algorithm, in terms of true-positive rates for all Neyman Pearson operating criteria.

[1] Also referred to as the Wilcoxon statistic

The results presented here concern a single classification algorithm, naive Bayes, used in a number of problem domains. The *Parcel* algorithm, however, requires no fixed classification algorithm to be used, nor indeed does it require a *single* classification algorithm to be used. As the *realisable classifier theorem* does not require the two existing classifiers to be of the same type, it is possible to use multiple classification algorithms, say naive Bayes and a multi layered perceptron, and to carry out the search for suitable classifiers to form the $MRROC$ by not only varying the feature subset, but also the classification algorithm.

While there is no guarantee that a *Parcel* algorithm will produce a classification system superior to that produced by a *wrapper*, it will be at least equivalent to that produced by a *wrapper*, in terms of true-positive rate. In other words, there is no guarantee that the *wrapper* will be adversely effected by a change of costs. If it is, however, then there is a strong possibility that a *Parcel* system will be superior, as its objective function is sensitive to all costs. Empirical results suggest that real world classification systems are adversely effected by shifting costs, due to having been designed with reference to only one cost point, and the constraint of selecting a single feature set.

Parcel represents a novel and interesting departure from the norm in feature selection algorithms, allowing for systems, robust to cost variation, to be developed using combinations of multiple feature subsets and classifiers. Current research includes incorporating *Parcel* as part of a real world trial of classification systems for the prediction of organ rejection in liver transplant patients, applying *Parcel* to a classification problem involving *in vivo* ultrasound scans of abnormal breast masses, and examining novel search algorithms for use with *Parcel*.

7.1 Acknowledgements

The authors would like to thank Dr. David Spiegelhalter, for his suggestions about the *realisable classifier theorem*, indicating the parallels with classical statistical hypothesis testing, and Prof. David Hand for reading an early version of this report, and for suggesting the investigation of the combination of *Parcel* feature subsets. Further thanks are due to Dr. David Lovell, Dr. David Melvin, Mr. Antranig Basmann, and Ms. Heather Gilderdale for their useful comments and suggestions regarding early drafts of this report.

Appendix A

Significance tests

A.1 McNemars Test

Let two classifiers, trained with data d, be denoted $c_1(\cdot)$ and $c_2(\cdot)$, respectively, and the target function that they are approximating be $g(\cdot)$.

McNemars Test examines the null hypothesis that, for a randomly drawn test set d', the accuracy of $c_1(d')$ is the same as $c_2(d')$, i.e.

$$Pr[c_1(d') = g(d')] = Pr[c_2(d') = g(d')].$$

Using all the examples in the test set d', a contingency table like that in Table A.1is calculated. Under the null hypothesis, the error rate should be the same, hence $n_{01} = n_{10}$ and the expected contingency table should be that of Table A.2.

McNemars Test is a statistic based on these contingency tables, and is approximately χ^2 distributed, with one degree of freedom:

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}. \tag{A.1}$$

(The -1 is included as a continuity correction to account for the discrete nature of the statistic, while χ^2 is continuous.). If the null hypothesis is true, then the probability p of this quantity

n_{00} = number of examples misclassified by both $c_1(\cdot)$ and $c_2(\cdot)$	n_{01} = number of examples misclassified by $c_1(\cdot)$ but not by $c_2(\cdot)$
n_{10} = number of examples misclassified by $c_2(\cdot)$ but not by $c_1(\cdot)$	n_{11} = number of examples misclassified by neither $c_1(\cdot)$ nor $c_2(\cdot)$

Table A.1: A contingency table based on the errors produced by both classifiers. $n = (n_{00} + n_{00} + n_{00} + n_{00})$, the number of examples in the test data set.

n_{00}	$\frac{(n_{01}+n_{10})}{2}$
$\frac{(n_{01}+n_{10})}{2}$	n_{11}

Table A.2: The expected contingency table based on the null hypothesis

being greater than 3.84 is less than 0.05, greater than 6.63, $p < 0.01$, and greater than 7.88, $p < 0.005$.

A.2 Critical ratio for the difference in two $AUROC$s

need to say that test is pessemistically biased - used for tests between regular ROCs and MR-ROCs as well as between MRROCs in Parcel

To define a statistically significant difference between two $MRROC$ curves, one containing the other, the areas beneath the curves can be compared. As previously stated, the area of the $MRROC$ is equivalent to an estimate of the Wilcoxon statistic. To compare two Wilcoxon statistics, the standard errors of both need to be calculated. As both statistics are estimated from the same dataset, there will be correlation between the standard errors. The task of estimating, and making statistical significance judgements on the difference between, the areas of multiple correlated ROC curves has been examined in [28, 47, 46, 105].

A method for detecting a significant difference between two curves is to calculate the critical ratio z,

$$z = \frac{Area_1 - Area_2}{\sqrt{SE_1^2 + SE_2^2 - 2rSE_1SE_2}}, \tag{A.2}$$

where SE_1 refers to the standard error of the estimate of $Area_1$, and r represents the estimated correlation between $Area_1$ and $Area_1$. A value of $z \geq 1.96$ indicates, with a 95% confidence interval, that a difference exists.

When the areas are sufficiently large, say $> .7$, the standard error of the area $Area$ can be estimated as [46]

$$SE(Area) = \sqrt{\frac{Area(1 - Area) + (|C_1| - 1)(Q_1 - Area^2) + (|C_2| - 1)(Q_2 - Area^2)}{|C_1||C_2|}}$$

where $|C_1|$ and $|C_2|$ are the number of examples of class 1 and class 2 respectively, and

$$Q_1 = \frac{Area}{(2 - Area)}$$
$$Q_2 = \frac{2Area^2}{(1 + Area)}.$$

As the $MRROC$ hull is formed by combining the outputs of different classifiers at its vertices, and not by varying a threshold over a continuous (or discrete ordinal) output function, existing generalised non parametric methods [28, 47] for calculating the correlation between standard errors are not applicable. Failure to subtract out the $2rSE_1SE_2$ statement from the denominator in Equation A.2will result in conservative estimates of z, thereby making significant changes more difficult to detect.

The motivation in this paper for significance testing is to terminate the *Parcel* algorithm when the difference between $MRROC^{old}$ and $MRROC^{new}$ is very small. As convex hulls, iteratively formed one over the other, are being dealt with, there is no issue of hulls crossing, and what should be discouraged is progressively refining vertices on the hull by tiny amounts.

This is a justification for allowing the use of a conservatively biased significance test. The refinement of a vertex frequently means the replacement of a classifier with a small feature subset by one with a larger feature subset. The performance of the classifier with the smaller subset will tend to be more stable than that of the larger. Both have the same amount of data with which to model the classification task. Hence there should be less variance associated with estimated parameters from the classifier with a lower dimensioned feature space (as this space will be more densely populated by the available data). If increasing the dimension of the feature space results in a small estimated improvement in performance, this should be discouraged. A conservatively biased significance test will be biased against detecting small changes in the area of the $MRROC$, as desired.

Appendix B

Subsets selected by *Parcel*

Algorithm	**Adult feature subset**	*Algorithm*	**Adult feature subset**
wrapper	$\{0,3,7,10\}$		
Parcel f_1	$\{0,2,8,10,11\}$	*Parcel* f_9	$\{0,3,5,6,10\}$
Parcel f_2	$\{0,1,8,10,11\}$	*Parcel* f_{10}	$\{0,5,6,7,10\}$
Parcel f_3	$\{0,5,6,7,10\}$	*Parcel* f_{11}	$\{3,6,7,10,11\}$
Parcel f_4	$\{0,5,7,10,12\}$	*Parcel* f_{12}	$\{1,3,7,10,11\}$
Parcel f_5	$\{0,4,7,10,12\}$	*Parcel* f_{13}	$\{0,4,5,10,11\}$
Parcel f_6	$\{0,5,6,10,11\}$	*Parcel* f_{14}	$\{0,3,7,10,11\}$
Parcel f_7	$\{0,3,5,10,12\}$	*Parcel* f_{15}	$\{4,5,6,10,11\}$
Parcel f_8	$\{3,5,10,11,12\}$	*Parcel* f_{16}	$\{0,3,5,10,11\}$
Algorithm	**BrodSat feature subset**	*Algorithm*	**Cotton feature subset**
wrapper	$\{17,18\}$	*wrapper*	$\{5,15,17\}$
Parcel f_1	$\{0,1,14\}$	*Parcel* f_1	$\{9,13,33\}$
Parcel f_2	$\{2,11,18\}$	*Parcel* f_2	$\{2,9,17\}$
Parcel f_3	$\{0,10,18\}$	*Parcel* f_3	$\{9,13,25\}$
Parcel f_4	$\{0,9,18\}$	*Parcel* f_4	$\{9,17,25\}$
Parcel f_5	$\{0,2,3\}$	*Parcel* f_5	$\{5,13,17\}$
Parcel f_6	$\{0,11,18\}$	*Parcel* f_6	$\{13,17,21\}$
Parcel f_7	$\{9,12,18\}$	*Parcel* f_7	$\{13,21,32\}$
Parcel f_8	$\{11,12,18\}$	*Parcel* f_8	$\{1,21,25\}$
Parcel f_9	$\{5,11,18\}$	*Parcel* f_9	$\{9,21,25\}$

Table B.1: The feature subsets selected by both the $SFFS$ wrapper and *Parcel* algorithms for the Adult, BrodSat and Cotton classification problems.

Algorithm	Field feature subset	Grey feature subset
wrapper	$\{3, 8, 17\}$	$\{8, 23, 24\}$
Parcel f_1	$\{0, 1, 3, 11\}$	$\{0, 1, 2\}$
Parcel f_2	$\{1, 9, 11, 14\}$	$\{18, 20, 35\}$
Parcel f_3	$\{0, 3, 11, 14\}$	$\{7, 12, 19\}$
Parcel f_4	$\{1, 3, 17\}$	$\{11, 16, 31\}$
Parcel f_5	$\{1, 3, 11\}$	$\{24, 26, 32\}$
Parcel f_6	$\{3, 5, 16, 17\}$	$\{13, 24, 26\}$
Parcel f_7	$\{3, 11\}$	$\{27, 28, 29\}$
Parcel f_8	$\{3, 9, 11, 14\}$	$\{1, 27, 28\}$
Parcel f_9	$\{0, 3, 11, 17\}$	$\{17, 27, 28\}$
Parcel f_{10}	$\{0, 3, 11\}$	$\{11, 27, 28\}$
Parcel f_{12}	$\{0, 3, 5, 17\}$	$\{27, 28, 31\}$
Parcel f_{13}	$\{9, 10, 14, 17\}$	$\{26, 27, 28\}$
Parcel f_{14}	$\{3, 11, 12\}$	$\{14, 15, 16\}$
Parcel f_{15}	$\{5, 10, 14, 15\}$	
Parcel f_{16}	$\{4, 5, 14, 15\}$	
Parcel f_{17}	$\{3, 14, 17\}$	
Parcel f_{18}	$\{8, 9, 11, 14\}$	
Parcel f_{19}	$\{3, 9, 11, 14\}$	
Parcel f_{20}	$\{9, 14, 17\}$	
Parcel f_{21}	$\{3, 14, 15, 17\}$	
Parcel f_{22}	$\{11, 15, 16\}$	
Parcel f_{23}	$\{1, 3\}$	
Parcel f_{24}	$\{11, 15\}$	
Parcel f_{25}	$\{3, 11, 12, 14\}$	
Parcel f_{26}	$\{11, 12, 14, 15\}$	

Table B.2: The feature subsets selected by both algorithms for the Field and Grey classification problems.

Algorithm	Thyroid feature subset	Tree feature subset
wrapper	$\{2, 16\}$	$\{1, 14\}$
Parcel f_1	$\{2, 16, 18\}$	$\{0, 3, 5\}$
Parcel f_2	$\{0, 1, 3\}$	$\{3, 4, 14\}$
Parcel f_3	$\{2, 16, 17\}$	$\{3, 14\}$
Parcel f_4	$\{2, 7, 16\}$	$\{3, 10, 14\}$
Parcel f_5	$\{0, 2, 16\}$	$\{3, 5, 14\}$
Parcel f_6		$\{3, 11, 14\}$
Parcel f_7		$\{0, 1, 2\}$
Parcel f_8		$\{3, 10, 14\}$
Parcel f_9		$\{3, 4, 11\}$
Parcel f_{10}		$\{3, 17\}$
Parcel f_{11}		$\{3, 11\}$
Parcel f_{12}		$\{3, 11, 13\}$
Parcel f_{13}		$\{3, 5, 8\}$
Parcel f_{14}		$\{3, 5, 11\}$
Parcel f_{15}		$\{1, 9\}$

Table B.3: The feature subsets selected by both algorithms for the Thyroid and Tree classification problems. *Parcel* found five subset for the Thyroid problem, and fifteen for the Tree problem.

Bibliography

[1] D.W. Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, 36(1):267–287, 1992.

[2] D.W. Aha and R.L. Bankert. Feature selection for case-based classification of cloud types: an empirical comparison. In *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*, pages 106–112, 1994.

[3] D.W. Aha and R.L. Bankert. A comparative evaluation of sequential feature selection algorithms. In D. Fisher and H. Lenz, editors, *Fifth International Workshop on Artificial Intelligence and Statistics*, pages 1–7. Morgan Kaufmann, 1995.

[4] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *Ninth International Conference on Artificial Intelligence*, pages 547–552. MIT Press, 1991.

[5] H. Almuallim and T.G. Dietterich. Efficient algorithms for identifying relevant features. In *Ninth Canadian Conference on Artificial Intelligence*, pages 38–45. Morgan Kaufmann, 1992.

[6] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–306, 1994.

[7] R.R. Bailey, E.J. Pettit, R.T. Borochoff, M.T. Manry, and X. Jiang. Automatic recognition of usgs land use/cover categories using statistical and neural network classifiers. In *Proceedings of SPIE OE/Aerospace and Remote Sensing*. SPIE, 1993.

[8] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls,. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[9] B. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[10] M. Ben-Basset. f-entropies, probability of error, and feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:227–242, 1978.

[11] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis (Second Edition)*. Springer Verlag, 1985.

[12] C.M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[13] B. Bonnlander. *PhD Thesis : Nonparametric selection of input variables for connectionist learning*. PhD thesis, University of Colorado Department of Computer Science, 1996.

[14] B. Bonnlander. Mutual information. Private communication, Dec 1997.

[15] A. Bradley. *PhD Thesis : Machine learning for medical diagnostics*. PhD thesis, Dept Electrical and Computer Engineering, University of Queensland, Australia, 1996.

[16] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

[17] C.E. Brodley and P.E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45 – 77, 1995.

[18] C. Brunk, J. Kelly, and R. Kohavi. Mineset: an integrated system for data mining. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997.

[19] C. Cardie. Using decison trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann, 1993.

[20] R. Caruana and D. Freitag. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 180–189. Morgan Kaufmann, 1994.

[21] R. Caruana and D. Freitag. Technical report: How useful is relevance? In *Proceedings of the Fall'94 AAAI Symposium on Relevance, New Orleans*, Nov 1994.

[22] K.J. Cherkauer and J.W. Shavlik. Rapidly estimating the quality of input representations for neural networks. In *Proceedings of the IJACI-95 Workshop on Data Engineering for Inductive Learning*. Morgan Kaufmann, 1995.

[23] T.M. Cover and J.M.V. Campenhout. On the possible orderings in the measurment selection problem. *IEEE Transactions Systems, Man and Cybernetics*, 7(9):657–661, 1977.

[24] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[25] M.W. Craven and J.W. Shavlik. Using neural networks for data mining (submitted). *Future Generation Computer Systems, special issue on data mining*, 1, 1998.

[26] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3):http://www.elsevier.com/locate/ida, 1997.

[27] S. Davies and S. Russell. Np-completeness of searches for smallest possible feature sets. In *Proceedings of the AAAI Fall'94 Symposium on Relevance, New Orleans*, Nov 1994.

[28] E.R. DeLong, D.M. DeLong, and D.L. Clarke-Pearson. Comparing the areas under two or more correlated reciever operating characteristic curves: a non parametric approach. *Biometrics*, 44(11):837–845, 1988.

[29] P.A. Devijer and J. Kittler. *Pattern Recognition: a Statistical Approach*. Prentice Hall, 1982.

[30] J. Doak. An evaluation of feature selection methods and their application to computer security, technical report cse-92-18. Technical report, Department of Computer Science, University of California at Davis, 1992.

[31] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Leanring*, 12(29):103–130, 1997.

[32] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretisation of continuous features. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 194–202. Morgan Kaufmann, 1995.

[33] N. Draper and H. Smith. *Applied Regression Analysis*. Wiley, 1981.

[34] R.H. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley, 1973.

[35] U. Fayyad, D. Haussler, and P. Stolorz. Kdd for science data analysis: issues and examples. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[36] U. Fayyad and P. Smyth. Kdd for science data analysis: issues and examples. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[37] T.S. Ferguson. *Mathematical Statistics, a Decision Theoretic Approach*. Academic Press, 1967.

[38] D. Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research*, 4:147–179, 1996.

[39] I. Foroutan and J. Sklansky. Feature selection for automatic classification of non-gaussian data. *IEEE Transactions Systems, Man and Cybernetics*, 17(2):187–198, 1987.

[40] J.H. Friedman. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[41] C. Glymour, D. Madigan, D. Pregibon, and P. Smyth. Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery*, 1:25–42, 1996.

[42] D.F. Gordon and M. desJardins. Evaluation and selection of biases in machine learning. *MAchine Learning*, 20:1 – 17, 1995.

[43] E.J. Halpern, M.A. Albert, A.M. Krieger, C.E. Metz, and A.D. Maidment. Comparison of receiver operating characteristic curves on the basis of optimal operating points. *Academic Radiology*, 3(3):245–253, 1996.

[44] D.J. Hand. Branch and bound in statistical data analysis. *The Statistician*, 30:1–13, 1981.

[45] D.J. Hand. *Construction and Assessment of Classification Rules*. Wiley, 1997.

[46] 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.

[47] J.A. Hanley and B.J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 83(148):839–843, 1983.

[48] J.L. Hatfield and D.R. Soderquist. Practice effects and signal detection indicies in an auditory vigilance task. *Journal of the American Acoustical Society*, 46:1458–1463, 1969.

[49] A. K. Jain and Richard Dubes. Feature definition in pattern recognition with small sample size. *Pattern Recognition*, 10:85–97, 1978.

[50] A. K. Jain and D. Zongker. Feature selection: evaluation, application and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[51] H.J. Jerison, R.M. Pickett, and H.H. Stevenson. The elicited observing rate and decision processes in vigilance. *Human Factors*, 7:107–128, 1965.

[52] G.H. John, R. Kohavi, and K Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of Machine Learning '94*. Morgan Kaufmann, 1994.

[53] R.D. King, C. Feng, and A. Sutherland. Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9:189–333, 1995.

[54] K. Kira and L.A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*. Morgan Kaufmann, 1992.

[55] J. Kittler. Mathematical methods of feature selection in pattern recognition. *International Journal of Man-Machine Studies*, 7:609–637, 1975.

[56] J. Kittler. Feature search algorithms. In CH Chen, editor, *Pattern Recognition and Signal Processing*. Sijthoff and Noordhoff, The Netherlands, 1978.

[57] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Department of Computer Science, Stanford University, 1995.

[58] R. Kohavi. Data mining with mineset: what worked, what did not, and what might. In *Proceeding of the KDD-98 workshop on the Commercial Success of Data Mining*, 1998.

[59] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[60] R. Kohavi, D. Sommerfield, and J. Dougherty. Data mining using $mlc++$, a machine learning library in $c++$. *International Journal of Artificial Intelligence Tools,*, 6(4):537–566, 1997.

[61] I. Kononenko. Estimating attributes: analysis and extensions of relief. In *Proceedings of the European Conference on Machine Learning*. ECML, 1994.

[62] M. Kubat, R. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2/3):195–215, 1998.

[63] P. Langley. Selection of relevant features in machine learning. In *Proceedings of AAAI Fall Symposium on Relevance*. AAAI, September 1994.

[64] P. Langley. *Elements of Machine Learning*. Morgan Kaufmann, 1996.

[65] D.R Lovell, C.R. Dance, M. Niranjan, R.W. Prager, and K.J. Dalton. Using upper bounds on discrimination to select discrete valued features. In S. Usui, Y. Tohkura, S. Katagiri, and E. Wilson, editors, *Neural Networks for Signal Processing VI*, pages 233–242. Keihanna, Japan, 1996.

[66] D.R. Lovell, B. Rosario, M. Niranjan, R.W. Prager, K.J. Dalton, R. Derom, and J. Chalmers. Design, construction and evaluation of systems to predict risk in obstetrics. *Internationl Journal of Medical Informatics*, 46(3):159–173, 1997.

[67] D.R. Lovell, M.J.J. Scott, M. Niranjan, R.W. Prager, and K.J. Dalton. On the use of expected attainable discrimination for feature selection in large scale medical risk predication problems. Technical Report CUED/F-INFENG/TR.299, Department of Engineering, University of Cambridge, England, August 1997.

[68] D.J.C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

[69] D.J.C. MacKay. Bayesian non-linear modelling for the energy prediction competition. *ASHRAE Transactions*, 100(2):1053–1062, 1994.

[70] R. Lopez De Mantaras. A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6:81–92, 1991.

[71] D.G. Melvin. A comparison of statistical and connectionist techniques for liver transplant monitoring. Technical Report CUED/F-INFENG/TR.282, Department of Engineering, University of Cambridge, England, December 1996.

[72] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases, 1998.

[73] K. Messer and J. Kittler. A comparison of colour texture attributes selected by statistical feature selection and neural network methods. In *Pattern Recognition Letters*, pages 1241–1246, 1997.

[74] K. Messer and J. Kittler. Using feature selection to aid an iconic search through an image database. In *International Conference on Acoustics, Speech and Signal Processing, Munich, Germany, (April 21-24)*, volume 4, pages 1141–1146, 1997.

[75] K. Messer, J. Kittler, and M. Kraaijveld. Selecting features for neural networks to aid an iconic search through an image database. In IEE, editor, *IEE 6th International Conference on Image Processing and Its Applications*, pages 428–432, 1997.

[76] R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors. *Machine Learning, an Artificial Intelligence approach*. Morgan Kaufmann, 1986.

[77] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine learning, neural and statistical classification*. Ellis Horwood, 1994.

[78] J. Mingers. An empirical comparison of selection measures for decision tree induction. *Machine Learning*, 3:319–342, 1989.

[79] D. Mladenic. Automated model selection. In *Proceedings of the ECML Workshop on Knowledge Level Modeling and Machine Learning*, pages 52–65, 1995.

[80] K. Morik. Applications of machine learning. In *Proceedings of the 6th European Knowledge Acquisition Workshop*, pages 9–13. Springer-Verlag, Berlin, 1992.

[81] S. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[82] S. Murthy and S. Salzberg. Lookahead and pathology in decision tree induction. In *Proceedings Fourteenth International Joint Conference on Artificial Intelligence*, pages 1025–1031. AAAI, 1995.

[83] P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computing*, 77(26):917–922, 1977.

[84] R.M. Neal. Assessing relevance determination methods using delve. In C.M. Bishop, editor, *(To appear) Generalisation in Neural Networks and Machine Learning*. Springer Verlag, 1998.

[85] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1995.

[86] P.N. Peduzzi, R.J. Hardy, and T.R. Holford. A stepwise variable selection procedure for nonlinear regression models. *Biometrics*, 36:511–516, 1980.

[87] 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*. American Association for Artificial Intelligence, 1997.

[88] P. Pudil, F.J. Ferri, J. Novovicova, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceedings of the Twelveth International Conference on Pattern Recognition, IAPR*, Oct 1994.

[89] P. Pudil, J. Novovicova, and S. Blaha. A statistical approach to pattern recognition: theory and practical solution by means of preditas system. *Kybernetika*, 27:1–78, 1991.

[90] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[91] W.F. Punch, E.D. Goodman, Min Pei, Lai Chia-Shun, P. Hovland, and R. Enbody. Further reserch on feature selection and classification using genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*, pages 557–564, 1993.

[92] R.J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[93] R.J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[94] R.J. Quinlan and R. Cameron-Jones. Oversearching and layered search in empirical learning. In *Proceedings Fourteenth International Joint Conference on Artificial Intelligence*, pages 1019–1024. AAAI, 1995.

[95] C.E. Rasmussen. The linear model lin-1, delve technical report. Technical report, University of Toronto, Department of Computer Science, 1996.

[96] B.D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.

[97] W. Schiffmann, M. Joost, and R. Werner. Synthesis and performance analysis of multilayer neural network architectures, technical report 16/1992. Technical report, University of Koblenz, Institute fur Physics, 1992.

[98] M.J.J. Scott, M. Niranjan, D.G. Melvin, and R.W. Prager. Maximum realisable performance: a principled method for enhancing performance by using multiple classifiers in variable cost problem domains. Technical Report CUED/F-INFENG/TR.320, Department of Engineering, University of Cambridge, England, April 1998.

[99] M.J.J. Scott, M. Niranjan, and R.W. Prager. Realisable classifiers: improving operating performance on variable cost problems *(to appear)*. In *British Machine Vision Conference*. BMVC, September 1998.

[100] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[101] D. Silvey. *Statistical Inference*. Chapman and Hall, 1975.

[102] J.A. Stark and W.J. Fitzgerald. Searching for the optimal data model: two strategies for statistical variable selection. Technical Report CUED/F-INFENG/TR.259, Department of Engineering, University of Cambridge, England, June 1996.

[103] D. Steinberg. Hybrid cart logit models in classification and data mining. In *Advanced Research Techniques (ART) Forum of the American Marketing Association Meetings*, 1998.

[104] J.A. Swets. Information-retrieval systems. *Science*, 141:245–250, 1963.

[105] J.A. Swets and R.M. Pickett. *Evaluation of Diagnostic Systems*. Academic Press, 1982.

[106] W.N. Venables and B.D. Ripley. *Modern Applied Statistics with S-Plus*. Springer Verlag, 1996.

[107] P. Viola. *PhD Thesis : Alignment by maximization of mutual information*. PhD thesis, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1995.

[108] P. Viola. Density estimation for mutual information. Private communication, Nov 1997.

[109] N. Wyse, R. Dubes, and A. Jain. A critical evaluation of intrinsic dimensionality algorithms. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pages 415–425. Morgan Kaufmann, 1980.

[110] B. Yu and B. Yuan. A more efficient branch and bound algorithm for feature selection. *Pattern Recognition*, 26:883–889, 1993.