Pergamon

Pattern Recognition, Vol. 30, No. 7, pp. 1145–1159, 1997
© 1997 Pattern Recognition Society. Published by Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0031-3203/97 $17.00+.00

PII: S0031-3203(96)00142-2

THE USE OF THE AREA UNDER THE ROC CURVE IN THE EVALUATION OF MACHINE LEARNING ALGORITHMS

ANDREW P. BRADLEY[*]

Cooperative Research Centre for Sensor Signal and Information Processing, Department of Electrical and Computer Engineering, The University of Queensland, QLD 4072, Australia

(*Received* 15 *April* 1996; *in revised form* 29 *July* 1996; *received for publication* 10 *September* 1996)

Abstract—In this paper we investigate the use of the area under the receiver operating characteristic (ROC) curve (AUC) as a performance measure for machine learning algorithms. As a case study we evaluate six machine learning algorithms (C4.5, Multiscale Classifier, Perceptron, Multi-layer Perceptron, *k*-Nearest Neighbours, and a Quadratic Discriminant Function) on six "real world" medical diagnostics data sets. We compare and discuss the use of AUC to the more conventional overall accuracy and find that AUC exhibits a number of desirable properties when compared to overall accuracy: increased sensitivity in Analysis of Variance (ANOVA) tests; a standard error that decreased as both AUC and the number of test samples increased; decision threshold independent; and it is invariant to *a priori* class probabilities. The paper concludes with the recommendation that AUC be used in preference to overall accuracy for "single number" evaluation of machine learning algorithms. © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

The ROC curve   The area under the ROC curve (AUC)   Accuracy measures
Cross-validation   Wilcoxon statistic   Standard error

1. INTRODUCTION

The Receiver Operating Characteristic (ROC) curve has long been used, in conjunction with the Neyman–Pearson method, in signal detection theory.(1,2) As such, it is a good way of visualising a classifier's performance in order to select a suitable operating point, or decision threshold. However, when comparing a number of different classification schemes it is often desirable to obtain a single figure as a measure of the classifier's performance. Often this figure is a cross-validated estimate of the classifier's overall accuracy [probability of a correct response, $P(C)$]. In this paper we discuss the use of the area under the ROC curve (AUC) as a measure of a classifier's performance.

This paper addresses the generic problem of how to accurately evaluate the performance of a system that learns by being shown labelled examples. As a case study, we look at the performance of six different classification schemes on six "real world" medical data sets. These data sets are chosen to characterize those typically found in medical diagnostics, they have primarily continuous input attributes and have overlapping output classes. When comparing the performance of the classification schemes, Analysis of Variance (ANOVA) is used to test the statistical significance of any differences in the accuracy and AUC measures. Duncan's multiple range(3) test is then used to obtain the significant subgroups for both these performance measures. Results are presented in the form of ROC curves and ranked estimates of each classification scheme's overall accuracy and AUC. Discussion is then focused both on the performance of the different classification schemes *and* on the methodology used to compare them.

The paper is structured in the following way: Section 2 details some commonly used performance measures and describes the use of the ROC curve and, in particular, AUC as a performance measure; Section 3 briefly describes the six data sets to be used in the experimental study; Section 4 details the implementations of the six learning algorithms used and describes how they are modified so that the decision threshold can be varied and a ROC curve produced; Section 5 describes the experimental methodology used, outlines three types of experimental bias, and describes how this bias can be avoided; Section 6 gives specific details of the performance measures and Section 7 the statistical techniques used to compare these measures. Section 8 presents a summary of the results, which are then discussed in detail in the remaining sections of the paper.

2. AUC AS A PERFORMANCE MEASURE

The "raw data" produced by a classification scheme during testing are counts of the correct and incorrect classifications from each class. This information is then normally displayed in a *confusion matrix*. A confusion matrix is a form of contingency table showing the differences between the true and predicted classes for a set of labelled examples, as shown in Table 1.

In Table 1, T_p and T_n are the number of true positives and true negatives respectively, F_p and F_n are the numbers of false positives and false negatives respectively.

---

[*] Present address: Department of Computing Science, 615 General Services Building, University of Alberta, Edmonton, Canada T6G 2H1. E-mail: abradley@cs.ualberta.ca.

Table 1. A confusion matrix

True class	Predicted class		
	−ve	+ve	
−ve	T_n	F_p	C_n
+ve	F_n	T_p	C_p
	R_n	R_p	N

The row totals, C_n and C_p, are the number of *truly* negative and positive examples, and the column totals, R_n and R_p, are the number of *predicted* negative and positive examples, N being the total number of examples ($N = C_n + C_p = R_n + R_p$). Although the confusion matrix shows *all* of the information about the classifier's performance, more meaningful measures can be extracted from it to illustrate certain performance criteria, for example:

$$\text{Accuracy } (1 - \text{Error}) = \frac{(T_p + T_n)}{(C_p + C_n)} = P(C), \quad (1)$$

$$\text{Sensitivity } (1 - \beta) = \frac{T_p}{C_p} = P(T_p), \quad (2)$$

$$\text{Specificity } (1 - \alpha) = \frac{T_n}{C_n} = P(T_n), \quad (3)$$

$$\text{Positive predictive value} = \frac{T_p}{R_p}, \quad (4)$$

$$\text{Negative predictive value} = \frac{T_n}{R_n}. \quad (5)$$

All of these measures of performance are valid only for one particular *operating point*, an operating point normally being chosen so as to minimise the *probability of error*. However, in general it is not misclassification rate we want to minimise, but rather *misclassification cost*. Misclassification cost is normally defined as follows:

$$\text{Cost} = F_p \cdot C_{F_p} + F_n \cdot C_{F_n}. \quad (6)$$

Unfortunately, we rarely know what the individual misclassification costs actually are (here, the cost of a false positive, C_{F_p} and the cost of a false negative, C_{F_n}) and so system performance is often specified in terms of the required false positive and false negative rates, $P(F_p)$ and $P(F_n)$. This then is equivalent to the Neyman–Pearson method,$^{(1,2)}$ where $P(F_n)$ is specified and $P(F_p)$ is minimised with that constraint, or *vice versa*. Often, the only way of doing the constrained minimisation required for the Neyman–Pearson method is to plot $P(T_p)$ against $P(F_p)$ as the *decision threshold* is varied. Selecting the operating point (decision threshold) that most closely meets the requirements for $P(F_n)$ and $P(F_p)$. The plotted values of $P(T_p)$ and $P(F_p)$ as the decision threshold is varied is called a Receiver Operating Characteristic (ROC) curve.

There is still, however, a problem with specifying performance in terms of a single operating point [usually a $P(T_p)$, $P(T_n)$ pair], in that there is no indication as to how these two measures vary as the decision threshold is varied. They may represent an operating point where sensitivity [$P(T_p)$] can be increased with little loss in specificity [$P(T_n)$], or they may not. This means that the comparison of two systems can become ambiguous. Therefore, there is a need for a *single* measure of classifier performance [often termed accuracy, but not to be confused with $P(C)$] that is invariant to the decision criterion selected, prior probabilities, and is easily extended to include cost/benefit analysis. This paper describes the results of an experimental study to investigate the use of the area under the ROC curve (AUC) as such as a measure of classifier performance.

When the decision threshold is varied and a number of points on the ROC curve [$P(F_p) = \alpha$, $P(T_p) = 1 - \beta$] have been obtained the simplest way to calculate the area under the ROC curve is to use trapezoidal integration,

$$\text{AUC} = \sum_i \left\{ (1 - \beta_i \cdot \Delta\alpha) + \frac{1}{2}[\Delta(1 - \beta) \cdot \Delta\alpha] \right\}, \quad (7)$$

where

$$\Delta(1 - \beta) = (1 - \beta_i) - (1 - \beta_{i-1}), \quad (8)$$

$$\Delta\alpha = \alpha_i - \alpha_{i-1}. \quad (9)$$

It is also possible to calculate the AUC by assuming that the underlying probabilities of predicting negative or positive are Gaussian. The ROC curve will then have an exponential form and can be fitted either: directly using an iterative Maximum Likelihood (ML) estimation,$^{(4)}$ giving the difference in means and the ratio of the variances of the positive and negative distributions; or, if the ROC curve is plotted on double probability paper, a straight line can be fitted to the points on the ROC curve.$^{(5)}$ The slope and intercept of this fitted line are then used to obtain an estimate of the AUC.

As noted in reference (6), the trapezoidal approach systematically underestimates the AUC. This is because of the way all of the points on the ROC curve are connected with straight lines rather than smooth concave curves. However, providing there are a reasonable number of points on the ROC curve the underestimation of the area should not be too severe. In this experiment we obtain *at least* seven points from which to estimate the AUC and in most cases there are 15 points. The trapezoidal approach also does not rely on any assumptions as to the underlying distributions of the positive and negative examples and, as will be elaborated on in Section 9.3, is exactly the same quantity measured using the Wilcoxon test of ranks.

The Standard Error of the AUC(SE($\hat{\theta}$))$^{(6)}$ is of importance if we wish to test the significance of one classification scheme producing a higher AUC than another. Conventionally there have been three ways of calculating this variability associated with the AUC:$^{(7)}$

1. from the confidence intervals associated with the maximum likelihood estimate of AUC, ($\hat{\theta}$);
2. from the standard error of the Wilcoxon statistic, SE(W); and
3. from an approximation to the Wilcoxon statistic that assumes that the underlying positive and negative

distributions are exponential in type.(6) This assumption has been shown to be conservative; it slightly overestimates the standard error, when compared to assuming a Gaussian based ROC curve (as in the ML method).

The standard error, SE(W), is given by

$$\mathrm{SE}(W) = \sqrt{\frac{\theta(1-\theta)+(C_p-1)(Q_1-\theta^2)+(C_n-1)(Q_2-\theta^2)}{C_pC_n}}, \tag{10}$$

where, C_n and C_p are the number of negative and positive examples respectively and

$$Q_1 = \frac{\theta}{(2-\theta)}, \tag{11}$$

$$Q_2 = \frac{2\theta^2}{(1+\theta)}. \tag{12}$$

In this paper we shall calculate AUC using trapezoidal integration and estimate the standard deviation, SD($\hat{\theta}$), using both SE(W) and cross-validation, details of which are given in Sections 5 and 6. Next, we shall present the details of the data sets, classification algorithms, and methodology chosen for this experimental study.

3. THE DATA

The data sets used in this experiment all have two output classes and have between four and 13, primarily continuous, input variables. Except for the algorithms C4.5 and the Multiscale Classifier which automatically handle categorical inputs, any categorical input variables were made continuous by producing *dummy variables*.(8)

The six data sets chosen for use in this experiment were:

1. Cervical cell nuclear texture analysis (*Texture*);(9)
2. Post-operative bleeding after cardiopulmonary bypass surgery (*Heart*);(10)
3. Breast cancer diagnosis (*Breast*);(11)
4. Pima Indian's diabetes prediction (*Pima*);(12)
5. Heart disease diagnosis:(13,14)
    (a) Hungarian data set (*Hungarian*);
    (b) Cleveland data set (*Cleveland*).

All input variables were scaled to the range [0,1] using a linear transformation making the minimum value zero and the maximum value 1. This is a requirement for the Multiscale Classifier,(15)[1] but was done for all of the learning algorithms for consistency (with no loss of generality). Also, all examples in the data sets that had any missing input variables were removed; this was less than 1% of the available data in most of the data sets.

3.1. Cervical cell nuclear texture

These data were gathered by Ross Walker as part of a study into the use of nuclear texture analysis for the diagnosis of cervical cancer.(9) The data set consisted of 117 segmented images of normal and abnormal cervical cell nuclei. Using Grey Levels Co-occurrence Matrix (GLCM) techniques, 56 texture features were extracted from each of these images. The six most discriminatory features were then selected using sequential forward selection (SFS) with the Bhattacharyya distance measure,(16,17) giving 117 examples (58 normal, 59 abnormal) each with six features:

1. Inertia at distance one;
2. Correlation at distance one;
3. Cluster prominence at distance one;
4. Entropy at distance 15;
5. Inverse Difference Moment (IDM) at distance 11;
6. Cluster prominence at distance three.

3.2. Post-operative bleeding

The data were gathered independently as part of a study into post-operative bleeding undertaken at the Prince Charles Hospital in Brisbane.(10) Over 200 parameters have been recorded for each of 134 patients. However, due to the limited size of the data set, only the four routinely measured parameters with the highest statistical correlation to blood loss were used.[2] The four parameters were

1. WBAGCOL: Aggregation with collagen (pre-operative);
2. POAGCOL: Aggregation with collagen (post-operative);
3. POSTPLT: Platelet count (post-operative);
4. DILNPLAS: Plasma dilution (post-operative).

Of the original data set of 134 patient records only 113 contained all four of the required input parameters. All of the input parameters are continuous-valued with a lowest possible value of zero. These parameters are then used to predict the total blood loss, in the three hours post-operative, expressed as a ratio of body surface area. The blood loss is then quantised into two classes, normal and excessive bleeding. Here, a prediction of excessive bleeding is defined as a total blood loss, in the 3 h post-operative, of greater than 16.4 ml/m^2. This defines 25% of all patients to have bled excessively and is an arbitrary definition that includes patients not clinically assessed as bleeding excessively. It was necessary to associate this absolute binary classification to the blood loss to make the data set consistent with the others used in this paper,

---

[1] It is also recommended for methods such as k nearest neighbours.(16)

[2] They were not highly correlated to the other features selected.

and as part of this preliminary study, this simplistic model was thought to be sufficient. However, most of the classification algorithms detailed in Section 4 have been used for regression, where the actual amount of blood loss would be predicted quantitatively.

The remaining data sets were obtained from the University of Southern California, machine learning repository, ftp://ics.uci.edu:pub/machine-learning-databases.

3.3. *Breast cancer diagnosis*

Collected by Wolberg(11) at the University of Wisconsin, this domain contains some noise and residual variation in its 683 data points, the 16 examples with missing attributes being removed. There are nine integer inputs, each with a value between 1 and 10. The two output classes, benign and malignant, are non-evenly distributed (65.5% and 34.5% respectively).

3.4. *Pima Indian's diabetes*

The diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e. if the 2 h post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). The population lives near Phoenix, Arizona, U.S.A. There are eight continuously valued inputs with some noise and residual variation.(12) The two non-uniformly distributed output classes (65.1% and 34.9%) are tested negative or positive for diabetes. There is a total of 768 data points.

3.5. *Heart disease diagnosis*

The goal of this data set is to predict the presence of coronary artery disease from a number of demographic, observed, and measured patient features. Here, we used two of the available data sets (the ones with the most instances); both data sets have the same instance format but were collected at different hospitals.

3.5.1. Cleveland data. These data were collected by Robert Detrano, M.D., Ph.D. at V. A. Medical centre, The Cleveland Clinic Foundation. The data originally were collected with 76 raw attributes; however, in previous studies(13,14) only 14 attributes were actually used. The data set contains 297 examples, there being six examples removed because they had missing values. Class distributions are 54% heart disease absent, 46% heart disease present.

3.5.2. Hungarian data. These data were collected by Andras Janosi, M.D. at the Hungarian Institute of Cardiology, Budapest. The data are in exactly the same format as the Cleveland data, except three attributes were removed due to a large percentage of missing values. There are 261 examples, 34 examples being removed because they had missing values. Class distributions are 62.5% heart disease absent, 37.5% heart disease present.

4. THE LEARNING ALGORITHMS

The learning algorithms chosen for this experimental comparison were:

- Quadratic Discriminant Function(18) (*Bayes*);[3]
- k-Nearest Neighbours(19) (*KNN*);
- C4.5(20) (C4.5);
- Multiscale Classifier(15) (*MSC*);
- Perceptron(21) (*PTRON*);
- and Multi-layer Perceptron(22) (*MLP*).

We chose a cross-section of popular machine learning techniques together with one algorithm developed in association with the author. There were two statistical methods (KNN and Bayes), two neural networks (PTRON, and MLP), and two decision trees (C4.5 and MSC).

The following should be noted about the implementations of each of the methods, *Quadratic discriminant function* (Bayes). The training data are used to estimate the prior probabilities, $P(w_j)$, mean, m_j, and covariance, C_j of the two class distributions. The Bayes decision function for class w_j of an example x is then given by

$$d_j(x) = \ln P(w_j) - \frac{1}{2}\ln|C_j| - \frac{1}{2}\left[(x-m_j)^{\mathrm{T}} C_j^{-1}(x-m_j)\right]. \tag{13}$$

This decision function is then a hyper-quadric, the class of an example being selected as the minimum distance class. Misclassification costs, c_j, are then applied to these distances, d_j, so as to weight the decision function and minimise the Bayes risk of misclassification. For these experiments misclassification costs were used in the range [0,1] in steps of 1/14.

k-Nearest Neighbours. For each test example, the five nearest neighbours (calculated in terms of the sum of the squared difference of each input attribute) in the training set are calculated. Then, if greater than L, where L=[0, 1, 2, 3, 4, 5], if the nearest neighbours are of class 1, the test sample is assigned to class 1; if not, it is assigned to class 0.

Release 5 of the C4.5 decision tree generator(20) was used with the following modification: when pruning a decision tree (in file prune.c) weight the local class distributions with the misclassification costs for each class. The default values for all parameters were used on all the data sets. Relative misclassification costs of [0.0:1.0, 0.015625:1.0, 0.03125:1.0, 0.0625:1.0, 0.125:1.0, 0.25:1.0, 0.5:1.0] were used for both classes on all the data sets.

---

[3] We shall refer to this method as "Bayes" even though it is not a truly Bayesian method. It would only be a Bayesian method, i.e. optimal, if the true distributions of the input variables were Gaussian.

The *Multiscale Classifier.* Version 1.2b1 of the Multiscale Classifier was used on each data set. The MSC was first trained for 10 epochs, or until 100% classification was achieved on the training set, then both pessimistic (MSCP) and minimum error (MSCM) pruning were used on the decision trees produced on each training set. The default pruning parameters of cf=1% and of m=8 were used on all data sets for pessimistic and minimum error pruning respectively. Relative misclassification costs of [1.0:1.0, 1.25:1.0, 1.5:1.0, 2.0:1.0, 4.0:1.0, 8.0:1.0, 16.0-:1.0, 32.0:1.0] were used for both of the classes on all data sets.

The *Perceptron.* Consisting of one neuron with a threshold activation function. The number of inputs (and weights) to the neuron is equal the number of input attributes for the problem, plus a bias. The network was trained, using the Perceptron learning algorithm(23) for 1000 epochs. The weights learnt were then tested using a neuron with a linear activation function, scaled to give an output in the range [0,1]. The output of this linear neuron was then thresholded at values [0, 0.1, 0.2, 0.3, ..., 1.0] to simulate different misclassification costs.(24)

The *Multi-layer Perceptron.* Three network architectures were implemented, each with different numbers of hidden units. Their network architecture was as follows: an input layer consisting of a number of units equal to the number of input attributes for the problem domain; a hidden layer consisting of 2, 4 and then 8 units; and finally one output unit (MLP2, MLP4, and MLP8 respectively). All of the neurons were fully connected, with log-sigmoid activation functions, i.e. their outputs were in the range [0,1]. All three networks were trained using back-propagation with a learning rate of 0.01, and a momentum of 0.2. Initial values for the weights in the networks were set using the Nguyen–Widrow method,(25) and the networks were trained for 20,000 epochs. Again, during the testing phase the output neuron was thresholded at values [0, 0.1, 0.2, 0.3, ..., 1.0] to simulate different misclassification costs.(24)

5. THE TRAINING METHODOLOGY

It is known that single train and test partitions are not reliable estimators of the true error rate of a classification scheme on a limited data set.(26,27) Therefore, it was decided that a random sub-sampling scheme should be used in this experiment to minimise any *estimation bias.* A leave-one-out classification scheme was thought computationally too expensive[4] and so, in accordance with the recommendations in reference (26), 10-fold cross-validation was used on all of the data sets. For consistency, exactly the same data were used to train and test all of the nine classification schemes, this is often called a *paired* experimental design.(7) The 10-fold cross-validation scheme has been extensively tested and has been shown to provide an adequate and accurate estimate of the true error rate.(27) The cross-validation sampling technique used was random but ensured that the approximate proportions of examples of each class remain 90% in the training set and 10% in the test set. This slight adjustment to maintain the prevalence of each class does not bias the error estimates and is supported in the research literature.(26)

As pointed out by Friedman,(28) no classification method is universally better than any other, each method having a class of target functions for which it is best suited. These experiments then, are an attempt to investigate which learning algorithms should be used on a particular subset of problems. This subset of "medical diagnostic" problems is characterized by the six data sets presented. Our conclusions are therefore targeted towards this subset of problems and should not be extrapolated beyond the scope of this class of problem. We have tried to minimise any bias in the selection of the problem domains, whilst tightly defining the subset of problems (*selection bias*). We have selected problems with a wide range of inputs (4–13) which would represent a typical number of features measured, or feature subset selected for medical diagnostic problems. The binary output classes are, as would be typically expected, overlapping. This is due to varying amounts of noise and residual variation in the measured features, and so a 100% correct classification would not, in general, be possible.

We have tried to minimise the effect of any *expert bias* by not attempting to tune any of the algorithms to the specific problem domains. Wherever possible, default values of learning parameters were used. These parameters include the pruning parameters for the decision trees, the value of k for the nearest neighbour algorithm, and the learning parameters (learning rate, momentum, and initial conditions) for the neural networks. This naïve approach undoubtedly results in lower estimates of the true error rate, but it is a bias that affects all of the learning algorithms equally. If we had attempted to tune the performance of each algorithm on each data set, then our different expertise with each method would of advantaged some algorithms but disadvantaged others. The experimentation time would also have increased dramatically as we evaluated different input representations, input transformations, network architectures, learning parameters, pruning parameters, or identified *outlying* examples in the training set. Also, in domains with a limited availability of data the introduction of an evaluation set (extracted from the training set) could actually reduce the overall accuracy of the algorithms.

6. THE PERFORMANCE MEASURES

For each learning algorithm (9 off) on each data set (6 off), 10 sets of results (one for each of the 10-fold cross-validation partitions) were stored. The raw data were stored in the form of a confusion matrix and for each of the 10 test partitions the decision thresholds were varied (to produce the ROC curves), giving between 7 and 15

---

[4] Particularly for the Multi-layer Perceptron.

sets of results for each test partition. In order to evaluate the performance of the different learning algorithms on each of the data sets, a number of measures have to be extracted from this raw data (over 6000 sets of results).

Overall accuracy, $P(C)$. For the default (conventional) decision thresholds, with equal misclassification costs, the value of the estimate of the true error rate [equation (1)] was calculated for the 10 cross-validation partitions.

The ROC curve. On each test partition the decision thresholds were effectively varied (by varying misclassification costs, as described in Section 4) to give a set of values for $P(T_p)$ and $P(F_p)$. The "average" ROC curves for each classification scheme are shown in Section 8.

The area under the ROC curve (AUC). As the misclassification costs were varied, as described in Section 4, each successive point on the ROC curve was used in the trapezoidal integration to calculate AUC. The AUC was calculated for each learning algorithm on each of the 10 test partitions. This is in effect using a *jackknife* estimate to calculate the standard error of the AUC(29) and will be discussed in more detail shortly.

Remark. It should be noted that there are two distinct possibilities when it comes to combining the ROC curves from the different test partitions,(30)

1. *Pooling.* In pooling, the raw data (the frequencies of true positives and false positives) are averaged. In this way one average, or *group* ROC curve is produced from the pooled estimates of each point on the curve. In this case we have 10 estimates of $P(T_p)$ and $P(F_p)$ for each point on the ROC curve. The assumption made when pooling the raw data is that each of the classifiers produced on each of the training partitions comes from the same population. Although the assumption that they come from the same population may be true in terms of their overall discrimination capacity (accuracy), the assumption that for each partition they are all estimating the same points on the ROC curve is less palatable. This can be seen from the fact that pooling the data in this way depresses the combined index of accuracy, AUC.(30)
2. *Averaging.* This alternative approach is to average the accuracy index extracted from each of the ROC curves on the 10 train and test partitions. So, AUC is calculated for the 10 ROC curves and then averaged, giving an estimate of the true area and an estimate of its standard error, calculated from the standard deviation of the 10 areas. The only problem with this approach is that it does not result in an average ROC curve, only an average AUC. For this reason the *pooled* responses are used when actually visually showing the whole ROC curves, as in Section 8.

The standard deviation of AUC, $\mathrm{SD}(\hat{\theta})$. In order to validate our estimate of the standard deviation of AUC obtained using *averaging*, $\mathrm{SD}(\hat{\theta})$, $\mathrm{SE}(W)$ was also calculated using the approximation to the Wilcoxon method, given in equation (10).

7. THE COMPARATIVE TECHNIQUES

7.1. *Analysis of variance*

In this paper we will use Analysis of Variance (ANOVA) techniques to test the hypothesis of equal means over a number of learning algorithms (populations) simultaneously.(3) The experimental design allows us to compare, on each data set, the mean performance for each learning algorithm *and* for each train and test partition. This is called *two-way classification* and effectively tests *two* hypotheses simultaneously:

1. H_0', that all the means are equal due to the different train and test partitions;
2. H_0'', that all the means are equal due to the different learning algorithms.

Of these two hypotheses we are only really interested in the second, H_0'', and we could have used a one-way ANOVA to test this hypothesis alone. However, a one-way ANOVA assumes that all the populations are independent, and can often be a less sensitive test than a two-way ANOVA, which uses the train and test partitions as a blocking factor.(31) The f ratio calculated from this ANOVA table is insensitive to departures from the assumption of equal variances when the sample sizes are equal, as in this case.(3) For this reason a test for the equality of the variances was not done.

7.2. *Duncan's multiple range test*

When the analysis of variance test on an accuracy measure produces evidence to reject the *null* hypotheses, H_0' and H_0'', we can accept the alternative hypothesis—that all of the mean accuracies are not equal. However, we still do not know which of the means are significantly different from which other means, so we will use Duncan's multiple range test to separate significantly different means into subsets of homogeneous means.

For the difference between two subsets of means to be significant it must exceed a certain value. This value is called the *least significant range* for the p means, R_p, and is given by

$$R_p = r_p\sqrt{s^2/r}, \qquad (14)$$

where the sample variance, s^2, is estimated from the error mean square from the analysis of variance, s_3^2, r the number of observations (rows), and r_p the *least significant studentized-range* for a given level of significance (we chose α=0.05), and the degrees of freedom $[(r-1)(c-1)=72]$. Tables 2–7 show the subsets of adjacent means that are not significantly different, this being indicated by drawing a line under the subset.

8. RESULTS

In this section we give the summary of the results.

- *Nuclear Texture*: See Table 2 and Figs 1 and 2.
- *Post-operative Heart Bleeding*: See Table 3 and Figs 3 and 4.
- *Breast Cancer*: See Table 4 and Figs 5 and 6.
- *Pima Indians Diabetes*: See Table 5 and Figs 7 and 8.
- *Cleveland Heart Disease*: See Table 6 and Figs 9 adn 10.
- *Hungarian Heart Disease*: See Table 7 and Figs 11 and 12.

Table 2. Rank ordered significant subgroups from Duncan's multiple range test on the nuclear texture data

Classifier:	PTRON	MSCM	MSCP	C4.5	KNN	BAYES	MLP8	MLP4	MLP2
Accuracy:	85.0	85.0	85.0	89.2	89.2	89.2	90.0	90.0	91.7
Classifier:	MSCP	MSCM	C4.5	KNN	BAYES	PTRON	MLP4	MLP8	MLP2
AUC:	88.1	88.7	92.1	96.2	96.7	97.8	98.3	98.5	98.6

Table 3. Rank ordered significant subgroups from Duncan's multiple range test on the heart bleeding data

Classifier:	MSCM	MSCP	C4.5	PTRON	KNN	MLP8	MLP4	MLP2	BAYES
Accuracy:	69.2	70.8	71.7	72.5	74.2	75.0	76.7	78.3	79.1
Classifier:	C4.5	KNN	MLP4	MLP8	MLP2	PTRON	MSCM	MSCP	BAYES
AUC:	48.7	60.9	65.5	65.7	66.1	69.8	70.0	70.5	73.3

Table 4. Rank ordered significant subgroups from Duncan's multiple range test on the breast cancer data

Classifier:	PTRON	C4.5	MSCM	MSCP	MLP8	MLP4	MLP2	KNN	BAYES
Accuracy:	72.2	90.7	90.9	91.2	92.7	93.3	93.5	93.6	94.2
Classifier:	C4.5	MSCM	MSCP	PTRON	MLP4	MLP8	MLP2	KNN	BAYES
AUC:	93.7	94.4	94.4	94.5	95.2	96.2	96.5	97.0	98.2

Table 5. Rank ordered significant subgroups from Duncan's multiple range test on the Pima diabetes data

Classifier:	MSCM	MSCP	C4.5	PTRON	KNN	BAYES	MLP8	MLP4	MLP2
Accuracy:	68.1	68.2	71.7	73.6	74.8	75.9	77.0	77.1	78.4
Classifier:	MSCM	MSCP	BAYES	KNN	C4.5	MLP8	MLP4	PTRON	MLP2
AUC:	74.1	74.4	76.3	79.4	80.2	82.3	83.4	84.7	85.3

Table 6. Rank ordered significant subgroups from Duncan's multiple range test on the Cleveland heart disease data

Classifier:	MSCM	MSCP	PTRON	C4.5	MLP8	MLP4	MLP2	KNN	BAYES
Accuracy:	68.7	68.7	75.0	77.7	81.0	81.0	81.3	82.7	86.3

Classifier:	MSCP	MSCM	C4.5	MLP8	MLP2	MLP4	KNN	BAYES	PTRON
AUC:	73.7	73.8	84.2	84.4	85.9	86.1	86.9	90.8	91.2

Table 7. Rank ordered significant subgroups from Duncan's multiple range test on the Hungarian heart disease data

Classifier:	MSCM	MSCP	C4.5	KNN	MLP4	PTRON	MLP8	BAYES	MLP2
Accuracy:	71.5	71.5	73.0	74.1	75.5	76.7	77.4	78.9	79.3

Classifier:	MSCM	MSCP	C4.5	KNN	MLP8	MLP4	BAYES	MLP2	PTRON
AUC:	70.1	70.2	79.2	82.0	82.1	82.3	83.8	84.7	87.8

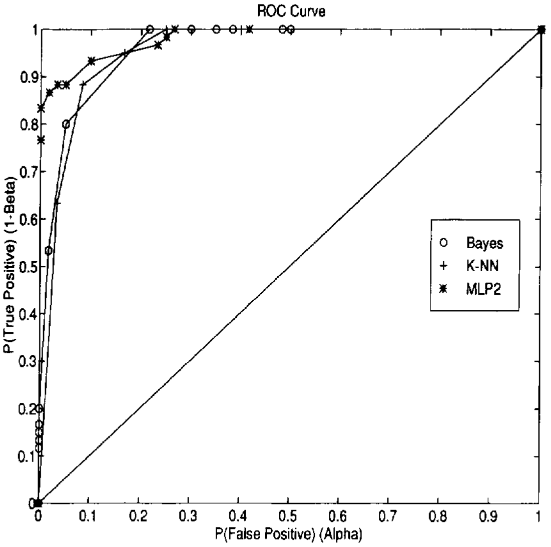

Fig. 1. ROC curve for Bayes, KNN, and MLP on the nuclear texture data.

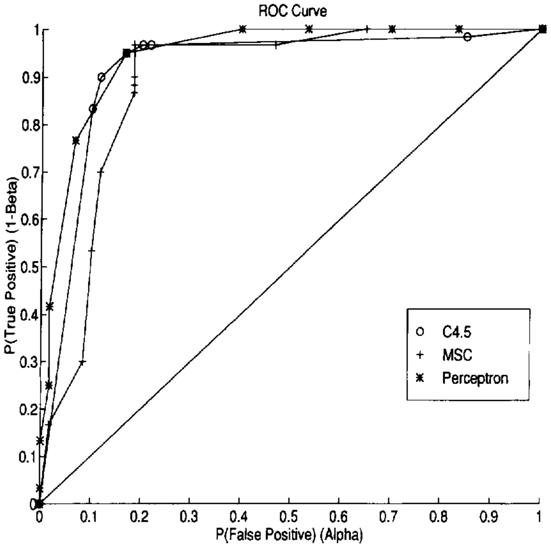

Fig. 2. ROC curve for C4.5, MSC, and Perceptron on the nuclear texture data.

9. DISCUSSION

In this section we discuss only the second hypothesis tested by the two-way analysis of variance (ANOVA), H_0''. This is the variance due to the different learning algorithms (column effects). The reason for this is that the train and test partitions are being used as what is called a *blocking factor*. We would hope for a significant effect due to the train and test partitions,[5] not because this variance is of any scientific interest, but because it is necessary for the two-way ANOVA to be more efficient than the one-way ANOVA.

9.1. Overall accuracy

All of the data sets showed some difference in average accuracy for each of the learning algorithms. However, the ANOVA showed that on one of these data sets (Nuclear Texture) there was *no* significant evidence ($p < 0.05$) for the mean accuracies to be actually differ-

---

[5] So that we can reject H_0'.

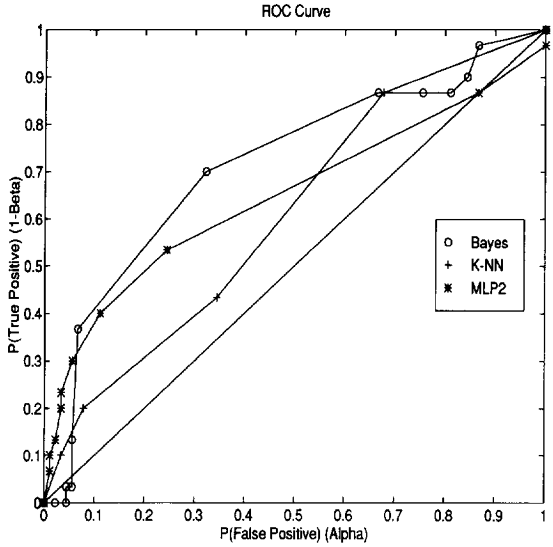

Fig. 3. ROC curve for Bayes, KNN, and MLP on the heart bleeding data.

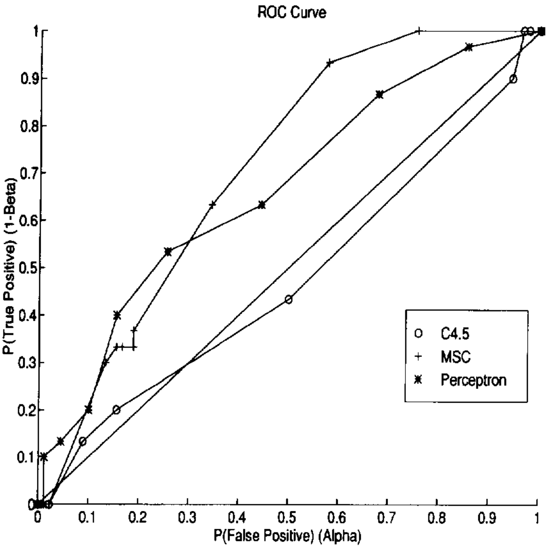

Fig. 4. ROC curve for C4.5, MSC, and Perceptron on the heart bleeding data.

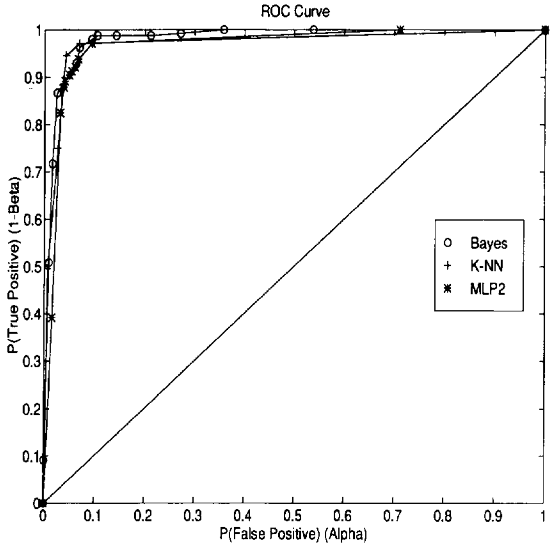

Fig. 5. ROC curve for Bayes, KNN, and MLP on the breast cancer data.

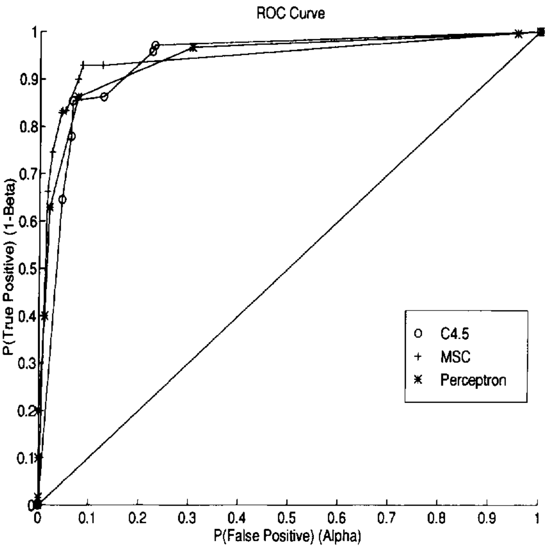

Fig. 6. ROC curve for C4.5, MSC, and Perceptron on the breast cancer data.

ent. On the other five data sets (where there was significant evidence to reject the null hypothesis, H_0'') Duncan's multiple range test was used to find the significant subgroups.

The Post-operative heart bleeding data set shows only two significant subgroups. Table 3 also shows that there is only a significant difference between the two decision trees methods (MSC and C4.5) and the MLP with two and four hidden units and Bayes.

Table 4 shows that for the Breast Cancer data set there are three significant subgroups: one subgroup containing only the Perceptron; one containing the two decision trees (MSC and C4.5); and the other learning algorithms in the third. There is also an overlap between the last two groups as the number of hidden units in the MLP is increased above 2. The fact that the Perceptron is in the lowest subgroup on its own would indicate that this problem is not linearly separable and so the Perceptron lacks the representation power to achieve a high overall accuracy. In addition, the lower performance observed using the decision tree methods may indicate that the optimal decision surface is smooth in nature.

The Pima Indians diabetes data set (Table 5) shows three significant subgroups under overall accuracy. The lowest accuracy group contains the decision trees (MSC and C4.5) though only Bayes and the Multi-layer Perceptrons (MLP) have a significantly ($p < 0.05$) higher overall accuracy. The poor performance of the decision

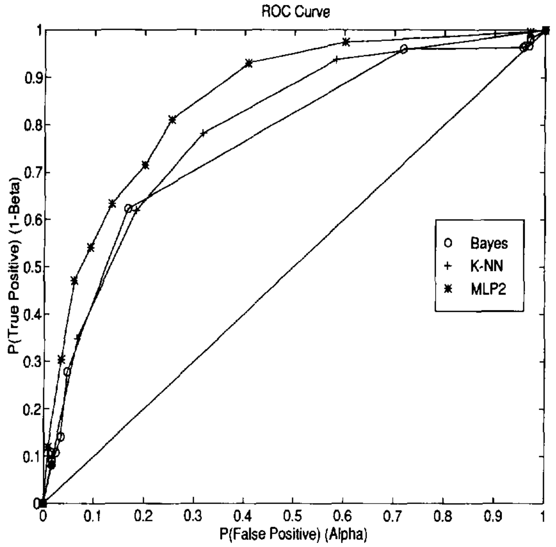

Fig. 7. ROC curve for Bayes, KNN, and MLP on the Pima diabetes data.

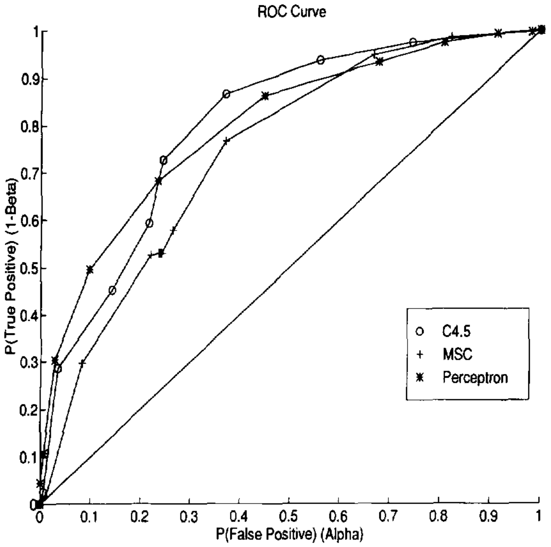

Fig. 8. ROC curve for C4.5, MSC, and Perceptron on the Pima diabetes data.

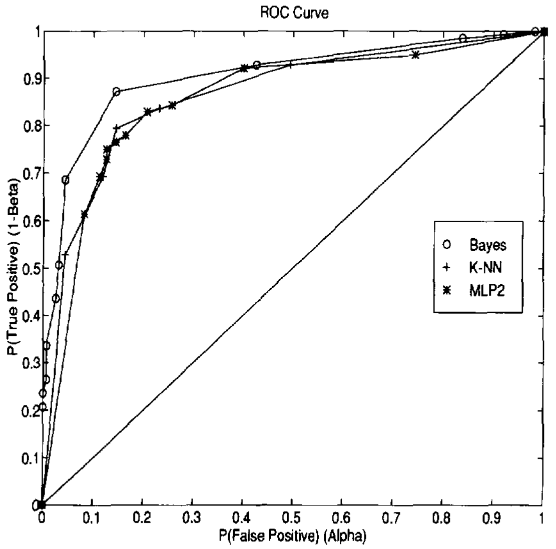

Fig. 9. ROC curve for Bayes, KNN, and MLP on the Cleveland heart disease data.

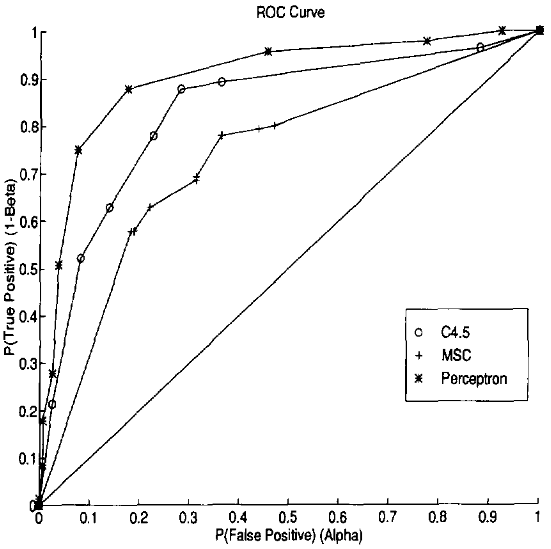

Fig. 10. ROC curve for C4.5, MSC, and Perceptron on the Cleveland heart disease data.

trees may indicate that the smooth decision hyperplanes are perhaps better suited to this problem, especially with the limited training data available. The relative success of the MLPs over the Bayesian method would indicate that the input features are not Normally distributed and so the covariance matrix is not being reliably calculated.

From Table 6, it can be seen that the Cleveland heart disease data set has four significant subgroups under overall accuracy. However, due to the large amount of subgroup overlap, there seems to be little discrimination due to the classification method. Perhaps of note, though, is the fact that on this problem the Bayes and KNN methods obtained the highest overall accuracies. This was surprising because the number of input features is 13, it being considered that when you have more than 10 input features the curse of dimensionality will start having a major effect.(8) Of all the learning algorithms used in this experiment, one would expect the Bayes and KNN to be the most severely affected by the curse of dimensionality. However, on this domain, this was obviously not the case.

Table 7 shows two significant subgroups for overall accuracy on the Hungarian heart disease data set. However, both of these subgroups are widely overlapping, the only significant differences being between the MSC and both the Bayes and the MLP (with two hidden units).

In general, when performance is measured in terms of overall accuracy, the hyper-plane (Bayes and MLP) and

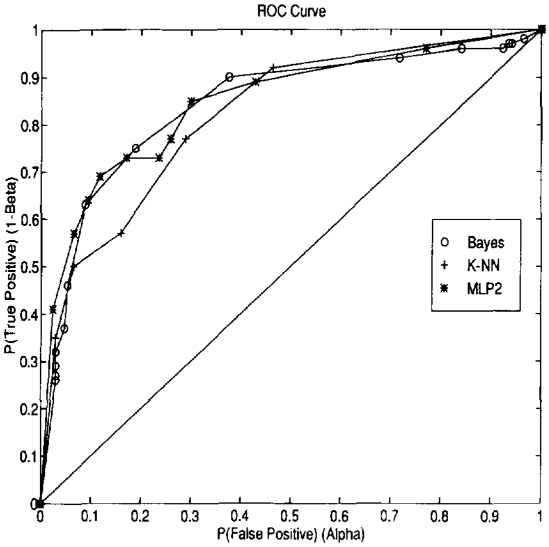

Fig. 11. ROC curve for Bayes, KNN, and MLP on the Hungarian heart disease data.

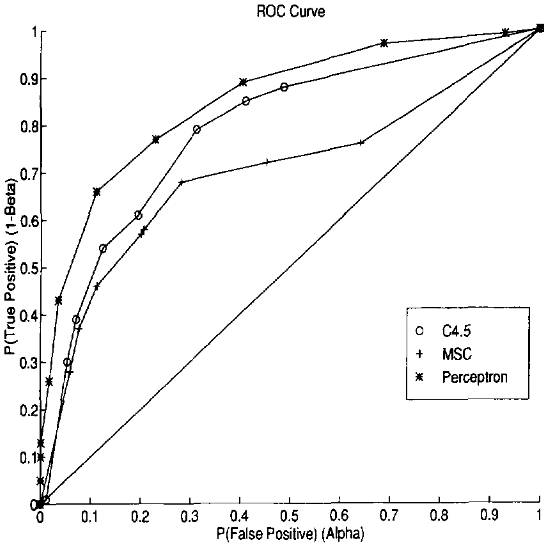

Fig. 12. ROC curve for C4.5, MSC, and Perceptron on the Hungarian heart disease data.

exemplar (KNN) based methods seemed to have a better performance when compared to the decision trees (MSC and C4.5). This result confirms what, from previous discussion, might be expected for data sets of this type, where the optimal decision boundary is a smooth hyperplane. For the decision tree methods to accurately estimate this type of decision boundary they would require *a lot* more training data to adequately populate decision nodes deep in the tree.

9.2. *The ROC curve*

The ROC curves for each learning algorithm on each data set are shown in Figs 1–12. These curves are the *pooled* ROC curves over the 10 train and test partitions. Curves for the MLPs with four and eight hidden units are not shown because of their similarity to the MLP with two hidden units (MLP2); also, for the same reason, only the curves for MSC with minimum error pruning are shown. It is perhaps worth making a couple of general comments as to the visual shape of the ROC curves presented in Figs 1–12.

- Decision trees (MSC and C4.5) do not appear to be producing ROC curves that conform to any Gaussian underlying distributions for the negative and positive classes, i.e. they do not form smooth exponential curves. This confirms our choice of trapezoidal integration over Maximum Likelihood estimation to calculate AUC. The dips and peaks seen in the ROC curves for the decision trees are probably due to the discrete way in which the decision trees are pruned, i.e. when the decision tree is pruned, a sub-tree is reduced to either a single leaf of the class with the minimum error, this single leaf can then subsequently lead to a number of misclassifications and so, the error rises in a discrete step.
- Though the ROC curves often appear to be producing a similar AUC, one curve may be preferable because it may have a lower $P(F_p)$ or $P(F_n)$ at a specific operating point. This reiterates the fact that for one particular application, the best way to select a classifier, and its operational point, is to use the Neyman–Pearson method.(1,2) Here, we select the required sensitivity and then maximise the specificity with this constraint (or *vice versa*).

The ROC curve is mainly of use when visualizing the performance of a classification algorithm as the decision threshold is varied. Any one point on the curve is a possible operational point for the classifier and so can be evaluated in the same manner as accuracy, $P(C)$, as above. However, in order to evaluate the whole curve we need to extract some distinguishing feature from it. The feature we have chosen to measure and evaluate is the area under the ROC curve (AUC).

9.3. *The area under the ROC curve*

As was the case for overall accuracy, all of the data sets showed some difference in average AUC for each of the learning algorithms. However, for the AUC the analysis of variance showed that on *all* of the data sets there were significant ($p < 0.01$) differences in mean AUCs. In addition, on all but one data set (Breast Cancer) the computed f values were greater for the AUC ANOVA test than for overall accuracy ANOVA. These larger f values led to a higher level of significance ($p < 0.01$ rather than $p < 0.05$) on two of the data sets (Post-operative bleeding and Hungarian heart disease). This indicates that the AUC is a more sensitive test than overall accuracy. The variance associated with the AUC, especially on the data sets with either high accuracy or ample test data, was less than that associated with $P(C)$. Again, Duncan's multiple range test was carried out on all six data sets to determine the significant subgroups.

On the nuclear texture data set, three significant subgroups were obtained, as shown in Table 2. The decision trees (MSC and C4.5) are in a lower performance subgroup of their own, with C4.5 in a second subgroup with KNN, and Bayes, the third, highest performance group, now includes the Perceptron and Multi-layer Perceptrons but excludes the decision trees (C4.5 and MSC). The poor performance obtained using the decision tree methods can be attributed to the fact there are limited data with which to construct and prune the trees and that smooth decision hyper-planes are probably more suitable than hyper-rectangles in this problem domain. Of note is the fact that the Perceptron and MSC obtained the same accuracy, $P(C)$, but the Perceptron now has a significantly higher ($p < 0.05$) AUC. With that exception there is an extremely good correlation between the rankings given from $P(C)$ and that given from AUC. However, AUC produced significant differences between the mean performance, whereas $P(C)$ did not.

There are two significant subgroups for the post-operative bleeding data set, as shown in Table 3. The lowest performance subgroup contains C4.5 only, the other subgroup containing all of the other methods. The low performance of C4.5 when measured using AUC can also be visually seen in the ROC curves of Figs 3 and 4. In this data set there are patients who have bled excessively due to a surgically related complication (a technical error). Some of the training data have therefore effectively been misclassified because the excessive bleeding was not related to any of the features measured, but was a consequence of the technical error. These cases should randomly affect the data and therefore become isolated examples in feature space. We would hope that they would have little effect on the classifier during training, but this will be dependent on the classification algorithm used. The effect of these points on the MLP, Perceptron, and Bayes methods is to bias the position of the decision boundary(s); however, if, as is thought for this case, the number of misclassified points is not too large, this effect should be minimal. KNN will be affected dependent upon the amount of smoothing built into the classification, i.e. upon the choice of K. For the decision tree methods (C4.5 and MSC) these points will cause the formation of erroneous decision nodes in the tree. However, it should then be possible to prune these examples from the tree to eliminate their effect, as they will be nodes that have seen very few training points, i.e. they have a low confidence level. However, because of the lack of data in this domain it is very difficult to determine with certainty which data points are due to a technical error and therefore should be pruned and which data points are due to the underlying problem. This can be seen in Fig. 4 particularly in the cases of the decision tree C4.5 where the pruning has reduced the structure of the tree too much and hence reduced the sensitivity. This means that C4.5 is very rarely predicting any cases as being positive; this "over caution" leads to what appears to be a acceptable accuracy, but a very low AUC. This means that the decision tree is actually doing very little work. In previous experiments[32] we found that the MSC obtained a higher accuracy (76%) when no pruning was done on the tree. This is an example of a problem domain where the algorithm has been biased by the decision tree pruning.[33]

There are three significant subgroups shown for the Breast Cancer data set in Table 4. There is a large amount of overlap in these subgroups and so no real identifiable groups seem to exist. However, there is an indication of a general increase in performance from the decision trees through the Perceptron on to the MLPs and then up to the KNN and Bayes methods. Again, with the exception of the Perceptron, which again obtained a higher ranking of performance under AUC than it did under $P(C)$, there is good agreement in the ranking between the two performance measures.

Table 5 shows that for the Pima Indians Diabetes data set there are four significant subgroups (as compared to three for overall accuracy). This again would indicate the increased sensitivity of AUC over $P(C)$ as a measure of classifier performance. In fact, it may well be worth going to a higher level of significance (say $p=0.01$) to reduce the number of subgroups and reveal a more general underlying trend. In addition, it can be seen from the ROC curve for the Bayes classifier (Fig. 7) that there are only really three points from which to estimate the AUC. This means that the AUC calculated for the Bayes classifier on this data set will be pessimistically biased. To avoid this effect it may be possible to implement a systematic way of varying the decision threshold when producing the ROC curves, rather than using linear steps.[34]

The Cleveland heart disease data set has three significant subgroups of performance under AUC (see Table 6). The MSC is in a subgroup of its own, the other two groups being fairly overlapping and so no meaningful subgroups can be identified. Again, the Perceptron obtained a higher ranking under AUC than it did under overall accuracy. With this exception, there is a good level of agreement in the ranking of the performance of the classification algorithms under accuracy and AUC.

Where accuracy found two broad significant subgroups, Table 7 shows that AUC has produced three subgroups on the Hungarian Heart Disease data set. The MSC is in the lowest performance subgroup (on its own) while the remaining two subgroups are broadly overlapping with only a significant difference between the AUC for C4.5 (lowest) and the Perceptron (highest). As was the case for the Cleveland heart disease data set, the Perceptron performed better under AUC than it did under overall accuracy, but otherwise accuracy and AUC produced similar rankings of performance.

9.3.1. The meaning of AUC.

It may seem that extracting the area under the ROC curve is an arbitrary feature to extract. However, it has been known for some time that this area actually represents the probability that a randomly chosen positive example is correctly rated (ranked) with greater suspicion than a

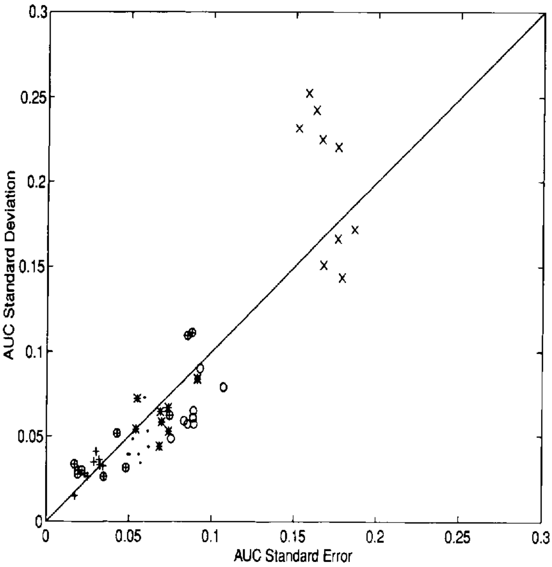

Fig. 13. Scatter plot of the standard error of the Wilcoxon statistic versus the standard deviation of the AUC. There are nine learning algorithms, each data set being shown with a different tick mark.

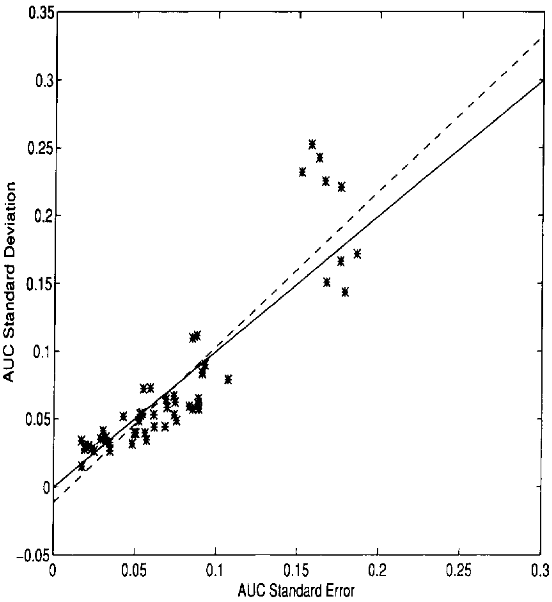

Fig. 14. Linear relationship between the standard error of the Wilcoxon statistic and the standard deviation of the AUC.

randomly chosen negative example.[(6)] Moreover, this probability of correct ranking is the same quantity estimated by the Wilcoxon statistic.[(6,35)]

The Wilcoxon statistic, W, is usually used to test the hypothesis that the distribution of some variable, x, from one population (p) is equal to that of a second population (n), $H_0: x_p = x_n$.[(3)] If this (*null*) hypothesis is rejected then we can calculate the probability, p, that $x_p > x_n$, $x_p < x_n$, or $x_p \neq x_n$. In our case, where we want good discrimination between the populations p and n, we require $P(x_p > x_n)$ to be as close to unity as possible. The Wilcoxon test makes no assumptions about the distributions of the underlying populations and can work on continuous, quantitative, or qualitative variables.

As already discussed AUC effectively measures $P(x_p > x_n)$. In the same situation, given one normal example and one positive example,[6] a classifier with decision threshold t will get both examples correct with a probability,

$$P(C) = P(x_p > t)P(x_n < t). \qquad (15)$$

$P(C)$ is dependent on the location of the decision threshold t and is therefore not a general measure of classifier performance.

9.3.2. The standard error of AUC. The AUC, θ, is an excellent way to measure $P(x_p > x_n)$ for binary classifiers and the direct relationship between, W, and θ can be used to estimate the standard error of the AUC, using SE(W) in equation (10).

Figures 13 and 14 show how the standard error of the Wilcoxon statistic, SE(W), is related to the standard deviation of the *averaged* AUC, SD(θ), calculated using 10-fold cross-validation. The *correlation coefficient* between SE(W) and SD(θ) is 0.9608 which indicates that there is a very strong linear relationship between SE(W) and SD(θ). Over all six data sets, SE(W) has a mean value of 0.0770 and a standard deviation of 0.0482, whilst SD(θ) has mean 0.0771 and standard deviation 0.0614. This again would indicate that although SD(θ) has a higher variance it is a very good estimator of SE(W). The straight line fitted (in a least squared sense) to SE(W) and SD(θ) in Fig. 14 again reiterates the quality of SD(θ) as an estimate of SE(W).

The larger variance observed for SD(θ) can be explained when you consider the fact that SD(θ) has *two* sources of variance. The first source of variance, which is also the variance estimated by SE(W), is due to the variation of the *test data*. That is, in each of the 10 iterations of cross-validation there is a different 10% of the data in each test partition. These different sets of test data therefore produce different ROC curves, and AUC varies accordingly. The second source of variance is due to variation of the *training data* in each cross-validation partition. The variation in the training data used in each cross-validation partition also affect the ROC curves produced and this causes AUC to vary. However, because only one-ninth of the training data vary with each subsequent training partition, this second source of variance is small and therefore, as was shown, SD(θ) is a good estimator of SE(W).

Figure 15 shows how the standard error of the Wilcoxon statistic varies with the number of test samples and the actual value of the AUC. The two trends to notice are:

1. As the number of test samples increase the standard error decreases, SE(W) being inversely pro-

---

[6] Often referred to as a two alternative forced choice experiment (2AFC).

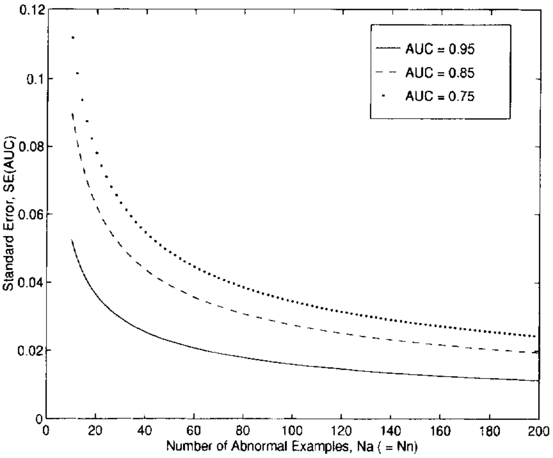

Fig. 15. Variation of the standard error of the Wilcoxon statistic with AUC and the number of test examples, assuming $C_n = C_p$.

portional to $\sqrt{N}$, where N is the number of test samples.

2. SE(W) is inversely proportional to AUC. There is a high variance associated with small values of AUC (< 0.8) and the variance becomes very small as the AUC gets close to 1. This effect can also be seen in Fig. 13; the "x" points represent the standard error and deviation estimated for the heart bleeding domain. On this domain the AUC was quite low ($\approx$0.66) and so the variation is noticeably higher.

There are also methods to reduce the standard error estimate for classifiers tested on the same data,(7) with its own significance test (to compare two AUCs). There are other measures of performance such as output signal-to-noise ratio, or deflection criterion,(36) but the AUC seems to be the only one that is independent of the decision threshold and not biased by prior probabilities.

10. CONCLUSIONS

In general there was not a great difference in the accuracies obtained from each of the learning algorithms over all the data sets. Generally, the hyperplane (Bayes, MLP) and exemplar (KNN) based methods performed better than the decision trees (C4.5, MSC) in terms of overall accuracy and AUC. However, this is due, in part, to the type of problems we have analysed, being primarily continuous inputs with overlapping classes; the models used by these methods are particularly well suited to this type of problem.

The area under the ROC curve (AUC) has been shown to exhibit a number of desirable properties as a classification performance measure when compared to overall accuracy:

- Increased sensitivity in the Analysis of Variance (ANOVA) tests;
- It is not dependent on a decision threshold chosen;
- It gives an indication of how well separated the negative and positive classes are for the decision index, $P(x_p > x_n)$;
- It is invariant to prior class probabilities.
- It gives an indication of the amount of "work done" by a classification scheme, giving low scores to the random or "one class only" classifiers.

However, there was good agreement between accuracy and AUC as to the ranking of the performance of the classification algorithms. It was also found that the standard deviation of the averaged AUCs from the 10-fold cross-validation can be used as an estimate of the standard error of the AUC calculated using the approximation to the Wilcoxon statistic.

The results quoted for the all the algorithms are only valid for the particular architecture or parameter settings tested, there may be other architectures that offer better performance. However, care should be taken when choosing parameters so as not to optimistically bias the results. Using a training, evaluation, and test set methodology should prevent this. Finally, for one particular application, the best way to select a classifier and its operational point is to use the Neyman–Pearson method, of selecting the required sensitivity and then maximising the specificity with this constraint (or *vice versa*). The AUC however, appears to be one of the best ways to evaluate a classifier's performance on a data set when a "single number" evaluation is required or an operational point has not yet been determined.

Acknowledgements—The Author is grateful to Geoffrey Hawson and Michael Ray of the Prince Charles in Brisbane for allowing access to the post-operative heart bleeding data set used in this study. The work of Michael Ray and Geoffrey Hawson is kindly supported by the Prince Charles Hospital Private Practice Study, Education, and Research Trust Fund. Thanks are also due to Gary Glonek, Brian Lovell, Dennis Longstaff, and the anonymous referees for helpful comments on earlier drafts of this paper.

REFERENCES

1. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Edn. Academic Press, San Diego, California (1990).
2. C. W. Therrien, in *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. Wiley, New York (1989).
3. R. E. Walpole and R. H. Myers, *Probability and Statistics for Engineers and Scientists*. Macmillan, New York (1990).
4. D. D. Dorfmann and E. Alf, Maximum likelihood estimation of parameters of signal detection theory and determination of confidence intervals—rating method data, *J. Math. Psychology* **6**, 487–496 (1969).
5. J. A. Swets, ROC analysis applied to the evaluation of medical imaging techniques, *Invest. Radiol.* **14**, 109–121 (1979).
6. J. A. Hanley and B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology* **143**, 29–36 (1982).
7. J. A. Hanley and B. J. McNeil, A method of comparing the areas under receiver operating characteristic curves derived from the same cases, *Radiology* **148**, 839–843 (1983).

8. J. H. Friedman, An overview of predictive learning and function approximation, *From statistics to neural networks: Theory and pattern recognition applications*, V. Cherkassky, J. H. Friedman and H. Wechsler, eds, *NATO ASI Series: Computer and Systems Sciences*, Vol. 136, pp. 1–61. Springer, Berlin (1993).
9. R. Walker, P. Jackway, B. Lovell and D. Longstaff, Classification of cervical cell nuclei using morphological segmentation and textural feature extraction, *Proc. 2nd Australian and New Zealand Conf. on Intelligent Information Systems*, pp. 297–301 (1994).
10. M. J. Ray, G. A. T. Hawson, S. J. E. Just, G. McLachlan and M. O'Brien, Relationship of platelet aggregation to bleeding after cardiopulmonary bypass, *Ann. Thoracic Surgery* **57**, 981–986 (1994).
11. W. H. Wolberg and O. L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, *Proc. Nat. Acad. Sci.*, U.S.A. **#87#12**, 9193–9196 (1990).
12. J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler and R. S. Johannes, Using the (ADAP) learning algorithm to forecast the onset of diabetes mellitus, *Proc. Symp. on Computer Applications and Medical Care*, pp.261–265. IEEE Computer Society Press, Silver Spring, Maryland (1988).
13. R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandu, K. Guppy, S. Lee and V. Froelicher, International application of a new probability algorithm for the diagnosis of coronary artery disease, *Am. J. Cardiol.* **64**, 304–310 (1989).
14. J. H. Gennari, P. Langley and D. Fisher, Models of incremental conceot formation, *Artif. Intell.* **40**, 11–61 (1989).
15. B. C. Lovell and A. P. Bradley, The multiscale classifier, *IEEE Trans. Pattern Analysis Mach. Intell.* **18**, 124–137 (1996).
16. P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Prentice-Hall, London (1982).
17. T. W. Rauber, M. M. Barata and A. S. Steiger-Garcao, A Toolbox for the Analysis and Visualisation of Sensor Data in Supervision, *Intelligent Robots Group Technical report*, Universidade Nova de Lisboa, Portugal (1993).
18. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Addison-Wesley, Reading, Massachusetts (1981).
19. D. J. Hand, *Discrimination and Classification*. Wiley, New York (1981).
20. J. R. Quinlan, *C4.5:Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1993).
21. M. L. Minsky and S. A. Papert, *Perceptrons*. MIT Press, Cambridge, Massachusetts (1969).
22. D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning Internal Representations by Error Propagation, *Parallel Distributed Computing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, eds. MIT Press, Cambridge, Massachusetts (1986).
23. F. Rosenblatt, *Principles of Neurodynamics*. Spartan Press, Washington D.C. (1961).
24. J. M. Twomey and A. E. Smith, Power curves for pattern classification networks, *Proc. IEEE Int. Conf. on Neural Networks*, San Francisco, California, pp. 950–955 (1993).
25. D. Nguyen and B. Widrow, Improving the Learning Speed of 2-Layer Neural networks by Choosing Initial Values of Adaptive Weights, *Proc. Int. Joint Conf. on Neural Networks* **3**, 21–26 (1990).
26. S. M. Weiss and C. A. Kulikowski, *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning and Expert Systems*. Morgan Kaufmann, San Mateo (1991).
27. L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*. Wadsworth, Belmont (1984).
28. J. H. Friedman, Introduction to computational learning and statistical prediction, Tutorial, *Twelth Int. Conf. on Machine Learning*, Lake Tahoe, California (1995).
29. B. Efron, Bootstrap methods: Another look at the jackknife, *Ann. Statist.* **7**, 1–26 (1979).
30. J. A. Swets and R. M. Pickets, *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. Academic Press, New York (1982).
31. C. R. Hicks, *Fundamental Concepts in the Design of Experiments*. Saunders College Publishing, London (1993).
32. A. P. Bradley, B. C. Lovell, M. Ray and G. Hawson, On the methodology for comparing learning algorithms: A case study, in *Proc. Second Australian and New Zealand Conf. on Intelligent Information Systems*, pp. 37–41. IEEE Publications, Brisbane, Australia (1994).
33. C. Schaffer, Overfitting avoidance as bias, *Machine Learning* **10**, 153–178 (1993).
34. R. F. Raubertas, L. E. Rodewald, S. G. Humiston and P. G. Szilagyi, ROC Curves for Classification Trees, *Med. Decision Making* **14**, 169–174 (1994).
35. D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*. Wiley, New York (1966, 1974).
36. B. PicinBono, On deflection as a performance criterion in dection, *IEEE Trans. Aerospace Electronic Systems* **31**, 1072–1081 (1995).

About the Author—ANDREW BRADLEY received his B.Eng. Honours degree in Electrical and Electronic Engineering in 1989 from the University of Plymouth, U.K. After working in industry for a number of years he has recently completed a Ph.D. at the Department of Electrical and Computer Engineering at The University of Queensland, Australia. His research interests include machine learning and pattern recognition for medical diagnostics, image analysis and scale-space methods.