Department of Statistics
University of Wisconsin, Madison
Technical Report 979
June 30, 1997
(revised January 27, 1998)

An Empirical Comparison of Decision Trees and Other Classification Methods

Tjen-Sien Lim, Wei-Yin Loh*
and Yu-Shan Shih†
limt@stat.wisc.edu, loh@stat.wisc.edu
and yshih@math.ccu.edu.tw
University of Wisconsin, Madison
and National Chung Cheng University, Taiwan

Abstract

Twenty two decision tree, nine statistical, and two neural network classifiers are compared on thirty-two datasets in terms of classification error rate, computational time, and (in the case of trees) number of terminal nodes. It is found that the average error rates for a majority of the classifiers are not statistically significant but the computational times of the classifiers differ over a wide range. The statistical POLYCLASS classifier based on a logistic regression spline algorithm has the lowest average error rate. However, it is also one of the most computationally intensive. The classifier based on standard polytomous logistic regression and a decision tree classifier using the QUEST algorithm with linear splits have the second lowest average error rates and are about 50 times faster than POLYCLASS. Among decision tree classifiers with univariate splits, the classifiers based on the C4.5, IND-CART, and QUEST algorithms have the best combination of error rate and speed, although the C4.5 trees tend to have about twice as many leaves as those from the other two algorithms. The C4.5 classifier based on rules also has good accuracy, but it does not scale as well as the other methods.

1 Introduction

There is much current research in the machine learning and statistics communities on the design of algorithms for decision tree classifiers. The emphasis has mainly focussed on the accuracy of the classifiers. One study, called the StatLog project (Michie, Spiegelhalter and Taylor, 1994), compared the accuracy of several decision tree classifiers against some non-decision tree classifiers on a large number of datasets. Other studies that are smaller in scale include Brodley and Utgoff (1992), Brown, Corruble and Pittard (1993), Curram and Mingers (1994), and Shavlik, Mooney and Towell (1991).

*Supported in part by grants from the U. S. Army Research Office and Pfizer, Inc. and a University of Wisconsin Vilas Associateship

†Supported in part by Republic of China National Science Council grant 86-2115-M-194-020

Recently, comprehensibility of the tree structures has received some attention. Since comprehensibility decreases with increase in tree size and complexity, if two trees employ the same kind of tests and have the same prediction accuracy, the one with fewer leaves is usually preferred. Breslow and Aha (1997) survey methods of simplifying trees to improve their comprehensibility.

A third criterion that has been largely ignored is the relative computational speed of the classifiers. The StatLog project indicates that no classifier is uniformly most accurate over the datasets studied. Instead, many classifiers possess comparable accuracy. For such classifiers, computational speed may be an important criterion (Hand, 1997).

The purpose of our paper is to extend the results of the StatLog project in the following ways:

1. In addition to classification accuracy and size of trees, we compare the relative computational speed of the algorithms studied. Although computational speed is affected by the implementation technique, it turns out that there are such large differences in computational times among the classification methods that these differences cannot be attributed to implementation alone.

2. We evaluate some decision tree classifiers that were not included in the StatLog project, such as LMDT (Brodley and Utgoff, 1995), OC1 (Murthy, Kasif and Salzberg, 1994), T1 (Holte, 1993; Auer, Holte and Maass, 1995), and QUEST (Loh and Shih, 1997). QUEST is unique among decision trees in that it has negligible selection bias in its splits.

3. We include some of the newest and highly accurate spline-based statistical classifiers. The classification accuracy of these classifiers serve as useful benchmarks for comparison of accuracy of decision tree classifiers.

4. We study the effect of the addition of independent noise variables on the classification accuracy and tree size of each classifier. It turns out that except possibly for three classifiers, all the other classifiers adapt to noise variables quite well.

5. We also examine the scalability of some of the classifiers as the sample size is increased.

Our experiment compared twenty-two decision tree classifiers, nine classical and modern statistical classifiers, and two neural network classifiers. Many of the datasets were taken from the University of California, Irvine, Repository of Machine Learning Databases (Merz and Murphy, 1996). Fourteen of the datasets were from real-life domains and two were artificially constructed. Five of the datasets were also used in the StatLog project. We doubled the number of datasets by adding noise variables to each set, making a total of 32 datasets.

Section 2 briefly describes each of the classification algorithms and Section 3 gives some background to the datasets. Section 4 explains the experimental setup used in this study and Section 5 reports the results. The scalability of some classifiers is examined in Section 6. Conclusions and recommendations are given in Section 7.

2 Descriptions of algorithms

Only a short description of each algorithm is given. Details may be found in the cited references.

2.1 Trees and rules

CART: We use the version of CART implemented in the `cart` style of the IND package (Buntine and Caruana, 1992) with the Gini index of diversity as the splitting criterion. The trees based on the 0-SE and 1-SE pruning rules are denoted by `IC0` and `IC1` respectively.

S-Plus tree: This is a variant of the CART algorithm written in the S language (Becker, Chambers and Wilks, 1988). It is described in Clark and Pregibon (1993). It treats the tree as a probability model and employs deviance as the splitting criterion. The best tree is chosen by 10-fold cross validation. Pruning is performed with the `p.tree()` function in the `treefix` library (Venables and Ripley, 1997)

from the StatLib S Archive at http://lib.stat.cmu.edu/S/. The 0-SE and 1-SE trees are denoted by ST0 and ST1 respectively.

C4.5: We use Release 8 (Quinlan, 1996) (http://www.cs.su.oz.au/~quinlan/) with the default settings, which include pruning. The algorithm is described in Quinlan (1993). After a tree is constructed, the C4.5 rules induction program is used to produce a set of rules. The trees are denoted by C4T and the rules by C4R.

FACT: This fast classification tree algorithm is described in Loh and Vanichsetakul (1988). It employs statistical hypothesis tests to select a variable for splitting each node and then uses discriminant analysis to find the split point. The size of the tree is determined by a set of stopping rules. The trees based on univariate splits (splits on a single variable) are denoted by FTU and those based on linear combination splits (splits on linear functions of the variables) are denoted by FTL. The Fortran 77 program may be obtained from http://www.stat.wisc.edu/~loh/.

QUEST: This is a new classification tree algorithm derived from the FACT method. QUEST can be used with univariate splits or linear combination splits. Unlike FACT, QUEST uses cross-validation pruning.

A feature that distinguishes QUEST from other decision tree classifiers is that, when used with univariate splits, the classifier performs approximately unbiased variable selection. Specifically, if all the variables are independent and are uninformative with respect to the class variable, then each variable has approximately the same chance of being selected to split a node. This is achieved by first employing statistical F- and chi-squared tests to select the variable and then employing cluster and discriminant analyses on the selected variable to find the split point. Each unordered variable is converted to an ordered discriminant coordinate prior to variable selection at every node. The trees pruned with the 0-SE and 1-SE rules are denoted by QU0 and QU1, respectively.

When used with linear combination splits, QUEST projects the data onto the largest discriminant coordinate and then finds the split point along this coordinate. The trees are denoted by QL0 and QL1 for the 0-SE and 1-SE rules, respectively.

The results in this paper are based on version 1.6 of QUEST. Details of the method are given in Loh and Shih (1997). The QUEST code may be obtained from http://www.stat.wisc.edu/~loh/.

IND: This is due to Buntine (1992). We use version 2.1 with the default settings. It is available from W. Taylor at taylor@ptolemy.arc.nasa.gov. IND comes with several standard predefined styles. We compare four of the styles in this paper: bayes, bayes opt, mml, and mml opt (denoted by IB, IBO, IM, and IMO, respectively). The styles bayes and mml are Bayesian. The opt methods extend the Bayes trees by growing many different trees and storing them in a compact and/or graph structure. Although time and memory intensive, they can provide significant improvements in prediction accuracy.

OC1: We use version 3 of the algorithm in Murthy et al. (1994) (http://www.cs.jhu.edu/~salzberg/announce-oc1.html). Three styles are studied. The first one (denoted by OCM) is the default that uses a mixture of linear combination and univariate splits. The second one (option -a; denoted by OCU) uses only univariate splits. The third one (option -o; denoted by OCL) uses only linear combination splits. Other options are set at their default values.

LMDT: The algorithm is described in Brodley and Utgoff (1995). It constructs a decision tree based on multivariate tests that are linear combinations of the variables. The tree is denoted by LMT. We use the default values in the software obtained from
http://yake.ecn.purdue.edu/~brodley/software/lmdt.html.

CAL5: This is developed by the Fraunhofer Society, Institute for Information and Data Processing, Germany (Müller and Wysotzki, 1994; Müller and Wysotzki, 1997). We use version 2 due to W. Mueller (wmueller@epo.iitb.fhg.de). CAL5 is designed specifically for numerical valued attributes. However, it has a procedure to handle categorical attributes so that mixed attributes (numerical and categorical) can be included. In this study we optimize the two parameters which control tree construction. They are the predefined threshold S and significance level α. We randomly split the training

set into two parts, stratified by the classes: 2/3 are used to construct the tree and 1/3 are used as a validation set to choose the optimal parameter configuration. We employ the `c-shell` program that comes with the CAL5 package to automatically choose the best parameters by varying α between 0.10 and 0.90 and S between 0.20 and 0.95 in steps of 0.05. The best combination of values that minimize the error rate on the validation set is chosen. The tree is then constructed on all the cases in the training set using the chosen parameter values. It is denoted by `CAL`.

T1: This is a decision tree that classifies examples on the basis of only one split on a single attribute (Holte, 1993; Auer et al., 1995). It is therefore a 1-level decision tree. The tree is denoted by `T1`. The software is obtained from
`http://www.csi.uottawa.ca/~holte/Learning/other-sites.html`.

2.2 Statistical classifiers

When a classifier in this section requires class prior probabilities, they are taken to be proportional to the training sample sizes.

LDA: Linear discriminant analysis is a classical statistical method for classification. It models the class distributions as normal with the same covariance matrix. This yields linear discriminant functions.

QDA: This classifier also models class distributions as normal, but estimates each covariance matrix by the corresponding sample covariance matrix. As a result, the discriminant functions are quadratic. Details on `LDA` and `QDA` can be found in many statistics textbooks, e.g., Johnson and Wichern (1992). We use the SAS PROC DISCRIM (SAS Institute, Inc., 1990) implementation of `LDA` and `QDA` with the default settings.

NN: This is the SAS PROC DISCRIM implementation of the nearest neighbor method. The pooled covariance matrix is used to compute Mahalanobis distances.

LOG: This is logistic discriminant analysis. The results here were obtained using a polytomous logistic regression (see, e.g., Agresti (1990)) Fortran90 routine written by the first author (`http://www.stat.wisc.edu/~limt/logdiscr/logdiscr.html`).

FDA: Flexible discriminant analysis is a generalization of `LDA`. Hastie, Tibshirani and Buja (1994) show that `LDA` is equivalent to multi-response linear regression using optimal scoring to represent the groups. They obtain a nonparametric version of discriminant analysis by replacing linear regression with a nonparametric regression method. Therefore any multi-response regression technique can be postprocessed to improve classification performance. Only one adaptive nonparametric regression procedure is compared in this study: MARS (Friedman, 1991). We use the S-Plus (`http://www.mathsoft.com/splus.html`) function `fda` from the `mda` library of the StatLib S Archive. Two models are used: the additive model (`degree=1`, denoted by `FM1`) and the model containing first-order interactions (`degree=2` with `penalty=3`, denoted by `FM2`).

PDA: This is a form of penalized `LDA` (Hastie, Buja and Tibshirani, 1995). It is designed for situations in which there are many highly correlated attributes, such as those obtained by discretizing a function, or the grey-scale values of the pixels in a series of images. The classification problem is cast into a penalized regression framework via optimal scoring. `PDA` is implemented in S-Plus using the function `fda` with `method=gen.ridge`.

MDA: This stands for mixture discriminant analysis (Hastie and Tibshirani, 1996). It fits Gaussian mixture density functions to each class to effect classification. `MDA` is implemented in S-Plus using the library `mda`.

POL: This is the POLYCLASS algorithm due to Kooperberg, Bose and Stone (1997). It fits a polytomous logistic regression model using linear splines and their tensor products. It provides estimates for conditional class probabilities which can then be used to predict class labels. `POL` is implemented in S-Plus using the function `poly.fit` from the `polyclass` library of the StatLib S Archive. Model selection is done with 10-fold cross validation.

2.3 Neural networks

LVQ: We use the learning vector quantization algorithms in the S-Plus `class` library (Venables and Ripley, 1997) of the StatLib S Archive. Details of the `LVQ` algorithms may be found in Kohonen (1995). Codebook initialization is done using the function `lvqinit`. The size of the codebook is set to be 10% of the size of the training set. The codebook is trained by the optimized-learning-rate `olvq1`, the fastest and most robust `LVQ` algorithm. Additional fine-tuning in learning is performed using the function `lvq1`. The number of iterations is 10 times the size of the training set in both `olvq1` and `lvq1`. The default values of α, the learning rate parameter, are 0.3 for `olvq1` and 0.03 for `lvq1`.

RBF: The radial basis function network is fitted using the SAS `tnn3.sas` macro for feedforward neural networks written by W. S. Sarle. The macro can be obtained from SAS at `http://www.sas.com`. The network is trained by specifying the architecture in the macro via the `ARCH=RBF` argument. In this study, we construct a network with only one hidden layer. The number of hidden units is chosen to be 20% of the total number of input and output units [2.5% (5 hidden units) only for the `dna` and `dna+` datasets and 10% (5 hidden units) for the `tae` and `tae+` datasets because of memory and space limitations]. Although the macro can perform model selection to adaptively choose the optimal number of hidden units, we did not utilize this capability because it would have taken too long for some of the datasets (see Table 5 below). Therefore the results reported here for this classifier should be regarded as lower bounds on its performance. The hidden layer is fully connected to the input and output layers but there is no direct connection between the input and output layers. At the output layer, each of the classes is represented by one unit taking the value of 1 for that particular category and 0 otherwise, except for the last one which is the reference category. To avoid local optima, 10 preliminary trainings were conducted and the best estimates used for subsequent training. More details on the radial basis function network can be found in Bishop (1995), Ripley (1996), and Sarle (1994).

3 Descriptions of datasets

In this section, we briefly describe the 16 datasets used in the study as well as any modifications that were made for our experiment. Fourteen of them are from real domains while two are artificially created. Thirteen datasets are obtained from the UCI Repository of Machine Learning Databases. Some of the following descriptions are adapted from the `readme` files that accompany the data.

Wisconsin breast cancer (`bcw`). The breast cancer database was collected at the University of Wisconsin Hospitals, Madison by W. H. Wolberg. It is part of the collection of databases at the University of California, Irvine. The problem is to predict whether a tissue sample taken from a patient's breast is malignant or benign. The tissue samples consist of visually assessed nuclear features of fine needle aspirates taken from patients' breasts. Each sample was assigned a 9-dimensional vector by Dr. Wolberg. Each component is in the interval 1 to 10, with value 1 referring to a normal state and 10 to a most abnormal state. Malignancy is determined by taking a tissue sample from the patient's breast and performing a biopsy on it. A benign diagnosis is confirmed by biopsy or by periodic examination, depending on the patient's choice. There are 2 classes, 9 numerical attributes, and 699 observations. Sixteen instances contain a single missing attribute value and are discarded from the analysis. Our results are therefore based on 683 observations only. The error rates are estimated using 10-fold cross validation. A decision tree analysis of an early subset of these data using the FACT classifier is reported in Wolberg, Tanner, Loh and Vanichsetakul (1987), Wolberg, Tanner and Loh (1988), and Wolberg, Tanner and Loh (1989). The data has also been analyzed with linear programming methods (Mangasarian and Wolberg, 1990).

Contraceptive method choice (`cmc`). The dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were pregnant at the time of the interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics. There are 3 classes, 2 numerical attributes, 7 categorical

attributes, and 1473 observations. The error rates are estimated using 10-fold cross validation. A more detailed analysis can be found in Lerman, Molyneaux, Pangemanan and Iswarati (1991). The data may be obtained from `http://www.stat.wisc.edu/~loh/`.

StatLog DNA (`dna`). This dataset from molecular biology was used in the StatLog project. It can be obtained from the UCI repository. Splice junctions are points on a DNA sequence at which "superfluous" DNA is removed during the process of protein creation in higher organisms. The problem is to recognize, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out). There are 3 classes and 60 categorical attributes each having 4 categories. The 60 categorical attributes represent a window of 60 nucleotides, each having one of four symbolic values (A, C, G, T). The middle point in the window is classified as one of exon/intron boundaries (referred to as EI sites), intron/exon boundaries (IE sites), or neither of these. The 3186 examples in the database were divided randomly into a training set of size 2000 and a test set of size 1186. The error rates are estimated from the test set.

StatLog heart disease (`hea`). This UCI dataset originally came from the Cleveland Clinic Foundation and was provided by R. Detrano of the V.A. Medical Center, Long Beach, CA. The problem concerns the prediction of the presence or absence of heart disease given the results of various medical tests carried out on a patient. There are 2 classes, 7 numerical attributes, 6 categorical attributes, and 270 observations. The 13 attributes are extracted from a larger set of 75. The analyses conducted in the StatLog project employ misclassification costs for different possible misclassifications. We did not incorporate the cost matrix in our analyses. The error rates are estimated using 10-fold cross validation.

Boston housing (`bos`). This dataset gives housing values in Boston suburbs (Harrison and Rubinfeld, 1978). There are 3 classes, 12 numerical attributes, 1 binary attribute, and 506 observations. The dataset can also be obtained from the UCI repository. Following Loh and Vanichsetakul (1988), the classes were created from the attribute median value of owner-occupied homes as follows: class = 1 if log(median value) $\leq$ 9.84, class = 2 if 9.84 $<$ log(median value) $\leq$ 10.075, class = 3 otherwise. The error rates are estimated using 10-fold cross validation.

LED display (`led`). This domain is described in Breiman, Friedman, Olshen and Stone (1984). It contains 7 Boolean attributes, representing 7 light-emitting diodes, and 10 classes, the set of decimal digits. All attribute values are either 0 or 1, according to whether the corresponding light is off or on for the decimal digit. Each attribute value has 10% probability of having its value inverted. The class attribute is an integer ranging between 0 and 9 inclusive. A C program to generate the data can be obtained from the UCI repository. We constructed 2000 observations for the training set and 4000 observations for the test set. The error rates are estimated from the test set.

BUPA liver disorders (`bld`). This dataset is also part of the UCI repository and was contributed by R. S. Forsyth. The problem is to predict whether or not a male patient has liver disorders based on various blood tests and the amount of alcohol consumption. There are 2 classes, 6 numerical attributes, and 345 observations. The error rates are estimated using 10-fold cross validation.

PIMA Indian diabetes (`pid`). This dataset is a subset of a larger database maintained by the National Institute of Diabetes and Digestive and Kidney Diseases. It was donated by V. Sigillito, Applied Physics Laboratory, Johns Hopkins University and is available at the UCI repository. All the patients in this particular dataset are females at least 21 years old of Pima Indian heritage living near Phoenix, Arizona, USA. The problem is to predict whether a patient would test positive for diabetes according to World Health Organization criteria (i.e., if the 2-hour post-load plasma glucose is at least 200 mg/dl at any survey examination or if found during routine medical care) given a number of physiological measurements and medical test results. There are 2 classes, 7 numerical attributes, and 532 observations. The original dataset consists of 768 observations with 8 numerical attributes. However, many of the attributes, notably the serum insulin, contain zero values which are physically impossible. Following Ripley (1996), we treated those zero values as missing and removed the cases from the data. We also omitted the attribute serum insulin. The error rates are estimated using 10-fold cross validation.

StatLog satellite image (sat). The StatLog version of the satellite image dataset consists of the multi-spectral values (4 spectral bands) of pixels in 3×3 neighborhoods in a satellite image, and the classification associated with the central pixel in each neighborhood. The aim is to predict this classification, given the multi-spectral values. The dataset is from the UCI repository. In the sample database, the class of a pixel is coded as a number. There are 6 classes and 36 numerical attributes. The training set consists of 4435 observations while the test set consists of 2000 observations. The error rates are estimated from the test set.

StatLog image segmentation (seg). The observations were randomly drawn from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel as one of brickface, sky, foliage, cement, window, path, or grass. The dataset is part of the UCI database collection. There are 7 classes, 19 numerical attributes and 2310 observations in the dataset. The error rates are estimated using 10-fold cross validation.

In the case of `T1`, because it requires a large amount of memory, we discretized the values of each attribute, except for attributes 3, 4, and 5 into 100 categories.

Attitude towards smoking restrictions (smo). This dataset came from the Attitudes Toward Smoking Legislation Survey–Metropolitan Toronto 1988, which was funded by NHRDP (Health and Welfare Canada). It was collected by L. Pederson and S. Bull at the Institute for Social Research at York University (Bull, 1994). It was obtained from
`http://lib.stat.cmu.edu/datasets/csb/`. The problem is to predict attitude toward restrictions on smoking in the workplace (prohibited, restricted, or unrestricted) based on bylaw-related, smoking-related, and sociodemographic covariates. There are 3 classes, 3 numerical attributes, and 5 categorical attributes. We divided the original dataset into a training set of size 1855 and a test set of size 1000. The error rates are estimated from the test set.

Thyroid disease (thy). The thyroid disease dataset was contributed by R. Werner of Daimler-Benz (the ANN version). It is available from the UCI repository. The problem is to determine whether or not a patient referred to the clinic is hyperthyroid. There are 3 classes (normal, hyperfunction, and subnormal functioning), 6 numerical attributes, and 15 binary attributes. The training set consists of 3772 observations and the test set has 3428 observations. The error rates are estimated from the test set.

StatLog vehicle silhouette (veh). This is another UCI dataset that originated from the Turing Institute, Glasgow, Scotland. The problem is to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different angles. The original purpose was to find a method of distinguishing 3-D objects within 2-D images by application of an ensemble of shape feature extractors to the 2-D silhouette of the objects. Four "Corgi" model vehicles were used for the experiment: a double decker bus, Chevrolet van, Saab 9000, and an Opel Manta 400. There were 4 classes, 18 numerical attributes, and 846 observations. The error rates are estimated using 10-fold cross validation.

1984 U. S. Congressional voting records (vot). The dataset originated from the Congressional Quarterly Almanac (CQA), 98th Congress, 2nd session, 1984, Volume XL: Congressional Quarterly Inc., Washington, D.C., 1985 and was contributed by J. C. Schlimmer to the UCI repository. It includes votes for each of the U. S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists 9 different types of votes: voted for, paired for, and announced for (these three simplified to "yea"); voted against, paired against, and announced against (simplified to "nay"); voted present, voted present to avoid conflict of interest, and did not vote or make a position unknown (simplified to an unknown disposition). The problem is to classify a Congressman as a Democrat or a Republican based on the 16 key votes. There are 2 classes, 16 categorical attributes with 3 categories each, and 435 observations. The database contains missing values. Since a missing value denoted by "?" in the original database means that the value is neither "yea" nor "nay", we converted it to form another category of the attribute. The error rates are estimated using 10-fold cross validation.

Waveform (wav). This artificial domain is described in Breiman et al. (1984). It is a 3-class problem based on 3 different waveforms. Each class consists of a random convex combination of 2 of these waveforms sampled at the integers with noise added. A description for generating the data is given in Breiman et al. (1984) and a C program is available from the UCI repository. There are 3 classes, 21 numerical attributes, and 600 observations in the training set. The test set consists of 3000 observations. The error rates are estimated from the test set.

TA evaluation (tae). The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin–Madison. The scores were divided into 3 roughly equal-sized categories ("low", "medium", and "high") to form the class variable. The predictor variables are (i) whether or not the TA is a native English speaker (binary), (ii) course instructor (25 categories), (iii) course (26 categories), (iv) summer or regular semester (binary), and (v) class size (numerical). This dataset was first analyzed in Loh and Shih (1997). It is unique among the datasets used in this experiment in that there are two categorical variables with large numbers of categories. As a result, decision tree algorithms such as CART that employ exhaustive search usually take much longer to compute than other classifiers (CART has to evaluate $2^{c-1}-1$ splits for each categorical variable with c values). The error rates are estimated using 10-fold cross validation. The data may be obtained from `http://www.stat.wisc.edu/~loh/`.

A summary of the features in each dataset is given in Table 1.

Table 1: Characteristics of the datasets. The last three columns give the number and type of added noise variables for each dataset. The notation "N(0,1)" denotes standard normal distribution, "UI(m,n)" denotes uniform distribution over the integers m through n inclusive, and "U(0,1)" denotes uniform distribution over the unit interval.

			No. of original attributes								No. and type of noise attributes		
Data	No. of	No. of	Num.	Categorical						Total	Numerical	Categorical	Total
set	cases	classes		2	3	4	5	25	26				
bcw	683	2	9							9	9 UI(1,10)		9
cmc	1473	3	2	3		4				9	6 N(0,1)		6
dna	2000	3				60				60		20 UI(1,4)	20
hea	270	2	7	3	2	1				13	7 N(0,1)		7
bos	506	3	12	1						13	12 N(0,1)		12
led	2000	10		7						7		17 UI(0,1)	17
bld	345	2	6							6	9 N(0,1)		9
pid	532	2	7							7	8 N(0,1)		8
sat	4435	6	36							36	24 UI(20,160)		24
seg	2310	7	19							19	9 N(0,1)		9
smo	1855	3	3	3	1		1			8	7 N(0,1)		7
thy	3772	3	6	15						21	4 U(0,1)	10 UI(0,1)	14
veh	846	4	18							18	12 N(0,1)		12
vot	435	2			16					16		14 UI(1,3)	14
wav	600	3	21							21	19 N(0,1)		19
tae	151	3	1	2				1	1	5	5 N(0,1)		5

4 Experimental setup

Some classifiers are not designed for categorical variables. In these cases, the categorical variables are converted to ordered variables by coding into 0-1 dummy vectors. The affected classifiers are all the statistical and neural network classifiers as well as the tree classifiers `QL0`, `QL1`, `FTL`, `OCU`, `OCL`, `OCM`, and `LMT`.

In order to increase the number of datasets and to study the effect of noise variables on each classifier, we created 16 new datasets by adding independent noise variables to each of the above datasets. The numbers and types of noise variables added are given in the right panel of Table 1. The name of each new dataset is the same as the original dataset except for the addition of a '+' symbol. For example, the `bcw` dataset with noise added is denoted by `bcw+`.

For each dataset, we use one of two different ways to estimate the error rate of each classifier. For large datasets (size much larger than 1000 and test set of size at least 1000), we use a test set to estimate the error rate. The classifier is constructed using the observations in the training set and then it is tested on the test set. Twelve of the thirty-two datasets were analyzed this way.

For the remaining twenty datasets, we use a 10-fold cross validation experiment to estimate the error rate. The 10-fold cross validation experiment is conducted as follows:

1. The dataset is randomly divided into 10 disjoint subsets, with each containing approximately the same number of cases. The sampling is stratified by the class labels to ensure that the subset class proportions are roughly the same as those in the whole dataset.

2. For each subset, a classifier is constructed using the observations outside that particular subset. The classifier is then tested on the withheld subset to estimate its error rate.

3. The 10 cross validation error estimates are averaged to provide an estimate for the classifier constructed from all the data.

Because the classifiers were implemented in different programming languages and some languages were available to us only on certain platforms, three types of UNIX workstations were used in our study. The workstation type and implementation language for each classifier is given in Table 2. The relative performance of the workstations according to SPEC marks is given in Table 3. Using these numbers as a guide, we estimate that the ratios of the amounts of time it takes each workstation to complete a given task is on average about

$$\text{SS5} : \text{DEC} : \text{SS20} = 1.4 : 1 : 0.8.$$

For comparison purposes, all CPU times are reported here in terms of DEC-equivalent seconds. Thus CPU times recorded on a SS5 and a SS20 are divided by 1.4 and 0.8, respectively.

5 Results

The error rate and CPU time for each classifier are given in separate tables for each dataset in the Appendix. The tables also report the error rate of the "plurality" or "naive" rule, which ignores the information in the covariates and classifies every case to the majority class in the training sample.

5.1 Error rates

Table 4 summarizes the error rates of the classifiers relative to that of the best classifier for each dataset. The second row of the table gives the average error rate for each classifier, averaged over the 32 datasets. The minimum and maximum error rates and that of the plurality rule are in the last three columns. It is interesting to note that some of the classifiers can be less accurate than the plurality rule.

In addition to average error rates, we can rank the classifiers according to the number of times each is close to the best as well as the number of times it is the worst. Let the smallest observed error rate for a given dataset be p. If p is obtained from an independent test-sample, let n denote the size of the test-sample. Otherwise, if p is obtained from cross-validation, let n denote the size of the learning sample. The standard error of p is then estimated as $\sqrt{p(1-p)/n}$. If a classifier has an error rate less than $p + \sqrt{p(1-p)/n}$, it is considered to be close to the best and is indicated by a $\surd$-mark. The total number of $\surd$-marks for each classifier is given in the third row of the table. The classifier with the worst error rate for each dataset is indicated by a `X`-mark. The total number of `X`-marks for each classifier is given in the fourth row of the table.

The following observations may be drawn from the table:

Table 2: Workstation type for each classifier. The types are DEC 3000 Alpha Model 300 (DEC), Sun SPARCstation 20 Model 61 (SS20), and Sun SPARCstation 5 (SS5).

Classifier	Machine	Language	Classifier	Machine	Language
Tree & Rules			ST1	DEC	S
QU0	DEC	F77	LMT	DEC	C
QU1	DEC	F77	CAL	SS5	C++
QL0	DEC	F77	T1	DEC	C
QL1	DEC	F77			
FTU	DEC	F77	Statistical		
FTL	DEC	F77	LDA	DEC	SAS
C4T	DEC	C	QDA	DEC	SAS
C4R	DEC	C	NN	DEC	SAS
IB	SS5	C	LOG	DEC	F90
IBO	SS5	C	FM1	SS20	S
IM	SS5	C	FM2	SS20	S
IMO	SS5	C	PDA	SS20	S
IC0	SS5	C	MDA	SS20	S
IC1	SS5	C	POL	SS20	S
OCU	SS5	C			
OCL	SS5	C	Neural Network		
OCM	SS5	C	LVQ	SS20	S
ST0	DEC	S	RBF	DEC	SAS

1. A ranking of the classifiers in terms of average error rates (given in parentheses) yields the ordering

   POL(.195), QL0(.202), LOG(.204), QL1(.205), MDA(.207), LDA(.208), (1)
   PDA(.213), IC0(.215), FM2(.217), IBO(.219), IMO(.219), C4R(.220),
   LMT(.220), IM(.220), C4T(.220), QU0(.221), QU1(.224), OCU(.227),
   IC1(.227), IB(.229), OCM(.230), ST0(.232), ST1(.233), FTL(.234),
   FTU(.238), FM1(.242), RBF(.257), OCL(.260), LVQ(.269), CAL(.270),
   NN(.281), QDA(.301), T1(.354).

   POL has the lowest average error rate and T1 the highest.

2. Another way to rank the classifiers is in terms of the total number of $\surd$- and X-marks. According to this criterion, the most accurate classifier is POL, because it has 15 $\surd$-marks and no X-marks. There

Table 3: SPEC benchmark summary

	Workstation	SPECfp92	SPECint92	Source
DEC	DEC 3000 Model 300 (150MHz)	91.5	66.2	SPEC Newsletter Vol. 5, Issue 2, June 1993
SS20	Sun SPARCstation 20 Model 61 (60MHz)	102.8	88.9	SPEC Newsletter Vol. 6, Issue 2, June 1994
SS5	Sun SPARCstation 5 (70MHz)	47.3	57.0	SPEC Newsletter Vol. 6, Issue 2, June 1994

are two classifiers (FTL and FM1) with the next highest number of $\surd$-marks, but they also have at least one X-mark. If we exclude the classifiers with one or more X-marks, the 10 most accurate classifiers, in order of decreasing number of $\surd$-marks (given in parentheses), are:

$$\begin{aligned} &\texttt{POL}(15), \texttt{QLO}(12), \texttt{QL1}(12), \texttt{LOG}(11), \texttt{LDA}(10), \texttt{PDA}(10), \\ &\texttt{QUO}(9), \texttt{OCU}(9), \texttt{C4R}(8), \texttt{RBF}(8). \end{aligned} \tag{2}$$

The top four classifiers in (3) are the same as the top four in (2), except that the positions of QL1 and LOG are reversed. Four of the classifiers in (3) are decision trees, four are statistical classifiers, one is a decision rule, and one a neural network. The most inaccurate classifier in terms of number of X-marks is T1 (it has 11 out of 32 X-marks). The classifiers with the next most number of X-marks are NN and LVQ (four X-marks each) and OCL and QDA (three X-marks each). These five classifiers also rank among the bottom six according to the ordering in (2).

3. The easiest datasets to classify are bcw, bcw+, vot, and vot+; the error rates all lie between 0.03 and 0.09.

4. The most difficult datasets are cmc, cmc+, and tae+, where the minimum error rates are greater than 0.4.

5. Two other difficult datasets are smo and smo+. In the case of smo, only T1 has a (marginally) lower error rate than that of the plurality rule. No classifier has a lower error rate than the plurality rule for smo+.

6. The datasets with the largest range of error rates are thy and thy+, where the rates range from 0.005 to 0.890. However, the maximum of 0.890 is due to QDA. If QDA is ignored, the maximum error rate drops to 0.0961.

7. There are six datasets with only one $\surd$-mark each. They are bld+ (POL), sat (LVQ), sat+ (FM2), seg+ (IBO), veh and veh+ (QDA both times).

8. Overall, the addition of noise variables does not appear to increase significantly the error rates of the classifiers.

5.2 CPU times

Table 5 gives the median DEC-equivalent CPU times for the classifiers and the relative CPU time for each classifier per dataset. Owing to the large range of CPU times, only the order relative to the fastest classifier for each dataset is reported. The fastest classifier is indicated by a '0'. A classifier that is between 10^{x-1} to 10^x times as slow is indicated by the value 'x'. For example, in the case of the dna+ dataset, the fastest classifiers are C4T and T1, each requiring 2 CPU seconds. The slowest classifier is FM2, which took more than 3 million seconds (almost 40 days) and hence is between 10^6 to 10^7 times as slow. The fastest and slowest times for each dataset are given in the last two columns.

A ranking from fastest to slowest in terms of median CPU time (given in parentheses) is:

$$\begin{aligned} &\texttt{C4T}(5\text{s}), \texttt{FTU}(7\text{s}), \texttt{FTL}(8\text{s}), \texttt{LDA}(10\text{s}), \texttt{QDA}(15\text{s}), \texttt{C4R}(20\text{s}), \\ &\texttt{NN}(20\text{s}), \texttt{IB}(34\text{s}), \texttt{IM}(34\text{s}), \texttt{T1}(36\text{s}), \texttt{QU1}(40\text{s}), \texttt{QUO}(41\text{s}), \\ &\texttt{OCU}(46\text{s}), \texttt{IC1}(47\text{s}), \texttt{ICO}(52\text{s}), \texttt{PDA}(56\text{s}), \texttt{LVQ}(1.1\text{m}), \texttt{MDA}(3\text{m}), \\ &\texttt{QLO}(3.7\text{m}), \texttt{QL1}(3.7\text{m}), \texttt{LOG}(4\text{m}), \texttt{LMT}(5.7\text{m}), \texttt{OCM}(13.7\text{m}), \texttt{ST1}(14.4\text{m}), \\ &\texttt{OCL}(14.9\text{m}), \texttt{STO}(15.1\text{m}), \texttt{FM1}(15.6\text{m}), \texttt{IBO}(27.5\text{m}), \texttt{IMO}(33.9\text{m}), \\ &\texttt{CAL}(1.3\text{h}), \texttt{POL}(3.2\text{h}), \texttt{FM2}(3.8\text{h}), \texttt{RBF}(11.3\text{h}) \end{aligned} \tag{3}$$

The fastest overall classifier is C4T, followed closely by FTU, FTL, and LDA. The classical statistical classifiers QDA and NN are also quite fast. As expected, decision tree classifiers that employ univariate splits are faster

than those that use linear combination splits. Of the three slowest classifiers, two are spline-based (POL and FM2) and one is the neural network RBF.

Although IC0, IC1, ST0 and ST1 all claim to implement the CART algorithm, the IND versions are faster than the S-Plus versions. One reason may be because IC0 and IC1 are written in C whereas ST0 and ST1 are written in the S language. A more important reason is that the IND versions use heuristics (Buntine, personal communication) instead of greedy search when the number of categories in a categorical variable is large. This is most apparent in the tae+ dataset where there are categorical variables with up to 26 categories. In this case IC0 and IC1 took about 40 seconds compared to about 2.5 hours for ST0 and ST1. The results in Table 4 suggest that IND's classification accuracy is not adversely affected by such heuristics. The technique in Aronis and Provost (1997) may also be useful in increasing the efficieny of some algorithms when there are categorical variables with many categories.

5.3 Size of trees

Table 7 gives the number of leaves for each tree classifier and dataset before noise variables are added. In the case that an error rate is obtained by 10-fold cross-validation, the entry is the average number of leaves over the 10 cross-validation trees. Table 9 shows how the number of leaves changes with addition of the noise variables. The mean and median of the number of leaves for each classifier are given in the last columns of the two tables. IBO and IMO clearly yield the largest trees by far. Apart from T1, which is necessarily short by design, the classifier with the shortest trees on average is QL1, followed closely by OCL and FTL. Among classifiers with univariate splits, the shortest trees are given (in increasing order) by T1, IC1, ST1, QU1, IC0, ST0, FTU, and QU0.

Addition of noise variables typically decreases the size of the trees, except for C4T and CAL which tend to grow larger trees, and IMO which seems to fluctuate rather wildly. Our study serves as a complement to that of Oates and Jensen (1997) who looked at the effect of sample size on the number of leaves of decision tree classifiers. They found a significant relationship between tree size and training sample size for C4T. They also observed that tree classifiers that employ cost-complexity pruning seem to be able to control tree growth.

5.4 Statistical significance

So far, we have simply ranked the classifiers according to various criteria, without paying attention to the statistical significance of the rankings. The statistical procedure called *two-way analysis of variance* can be used to test the simultaneous statistical significance of differences between average error rates of the classifiers (called "treatments") while controlling for differences between datasets (called "blocks") (Box, Hunter and Hunter, 1978, p. 209). Simultaneous confidence intervals for differences between average error rates can then be obtained using the Tukey method (Miller, 1981, p. 71). According to this procedure, a difference between the average error rates of two classifiers is statistically significant at the 10% level if they differ by more than 0.0584.

To visualize this result, Figure 1(a) plots the average error rate of each classifier versus its median ln(CPU time). The solid vertical line in the plot is 0.0584 units to the right of the average error rate for the best classifier POL. Therefore any classifier lying to the left of the line has an average error rate that is not statistically significantly different from that of POL. Of the top ten classifiers identified as most accurate in (3), only RBF lies to the right of the vertical line. Therefore the average error rate of RBF is significantly higher than that of POL while the other eight are not. This somewhat surprising result may be due to lack of adaptive optimization of the number of hidden units in RBF.

The horizontal dotted line in Figure 1(a) simply divides the classifiers into two equal halves according to their CPU times. Those below the horizontal line are on average quicker to execute than those above the line. The rectangle in the lower left corner of the plot delineated by the two lines thus identifies a subset of the classifiers with good *average* accuracy and computational speed. A magnified plot for this group is shown in Figure 1(b). Only five of the top ten classifiers listed in (3) appear in this group, namely (in increasing average error rates):

LDA, PDA, C4R, QU0, OCU. (4)

These classifiers therefore provide the best trade-offs between accuracy and speed. According to the Tukey procedure, pairwise differences between the average error rates of these five classifiers are not statistically significant.

6 Scalability of classifiers

Although the differences in average error rates between POL and many other classifiers are not statistically significant, it is clear that if error rate is the sole criterion, POL would be the method of choice. Unfortunately, POL is one of the most compute-intensive classifiers. To see how CPU times increase with sample size, a small scalability study was carried out with the classifiers QU0, QL0, FTL, C4T, C4R, IC0, LDA, LOG, FM1, POL.

The data for the study were constructed by bootstrapping from the datasets sat, smo+, and tae+ as follows. Given a dataset and a desired sample size N, N cases were randomly drawn *with replacement* from the dataset. To avoid getting many replicate cases, the value of the class attribute for each sampled case was randomly changed to another value with probability 0.1. (The new value was selected from the pool of alternatives with equal probability.) This procedure was repeated for values of N from 1,000, 2,000, ..., 8,000. The resulting CPU times required to train the classifiers are plotted (in log-log scales) in Figure 2.

The logarithms of the CPU times seem to increase linearly with $\log(N)$ for the majority of the classifiers, except for POL. One reason for the non-monotonic behavior of POL may be due to its model selection algorithm which is based on cross-validation. Some of the cross-validation samples it generated may be easy to fit and hence produce lower computation times.

Except for C4R, the lines in Figure 2 are roughly parallel. This suggests that the relative computational speed of the classifiers is fairly constant over the range of sample sizes considered. The steeper slope for C4R confirms the observation in Cohen (1995) that it does not scale as well as the other classifiers.

7 Conclusions

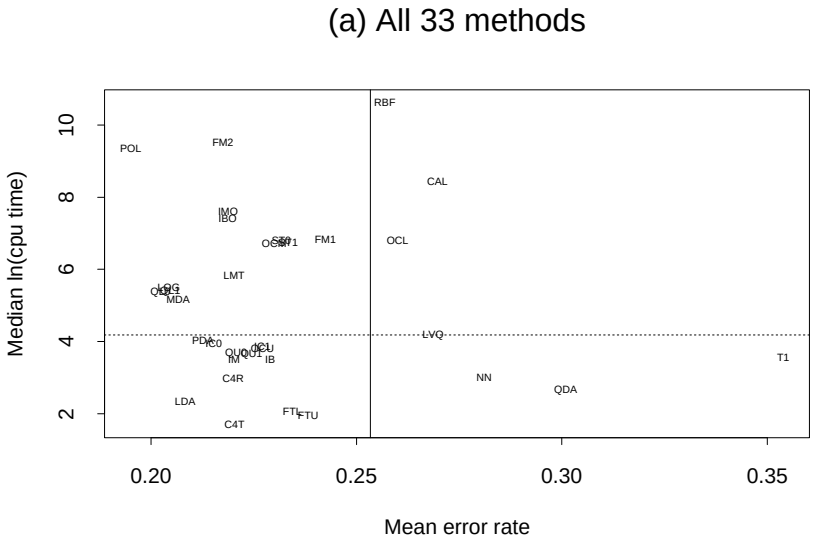
Our results show that the average error rates of many classifiers are sufficiently similar that their differences are statistically insignificant. For example, the four lowest average error rates (POL, QL0, LOG, QL1) lie between 0.195 and 0.205, a range of 0.01. If such small differences are not important in practical applications, the user may wish to select a classifier based on other criteria such as computational speed or interpretability.

Unlike error rates, there are huge differences between the CPU times of the classifiers. POL, the classifier with the lowest average error rate, takes about 50 times as long as the next best classifiers QL0, LOG and QL1. The ratio of times is roughly equivalent to hours versus minutes, and Figure 2 shows that it is maintained over a wide range of sample sizes. For large applications where time is a factor, it may be advantageous to use one of the latter classifiers instead of POL.

It is interesting that the old statistical classifier LDA has an average error rate close to the minimum. This is surprising because (i) it was not designed for binary-valued attributes (all categorical attributes were transformed to 0-1 dummy vectors prior to application of LDA), and (ii) it can be ineffective in situations where the class populations are multi-modal. The low error rate of LDA for the datasets used in the study suggests that multi-modality is not a problem here. We may therefore expect LDA to perform similarly well in many other real-life applications. Because it is fast, easy to implement, and readily available in statistical packages, we recommend that it be always used as a means for comparison against other classifiers, if nothing else.

The low error rates of LOG and LDA probably account for much of the performance of the better classifiers. For example, POL is basically a modern version of LOG. It increases the flexibility of LOG by employing spline-based functions and automatic model selection. Although this strategy is computationally costly, it does yield a slightly reduced average error rate—enough to bring it to the top of the pack.

The good performance of QL0 and QL1 may be similarly attributable to LDA. The QUEST linear-split algorithm was designed to overcome the difficulties encountered by LDA in multi-modal situations. It does this by applying a modified form of LDA to partitions of the data, where each partition is represented by a leaf of the decision tree. This strategy alone, however, is not enough, as the higher average error rate of FTL shows. FTL is based on the FACT algorithm which is a precursor to QUEST. One major difference between



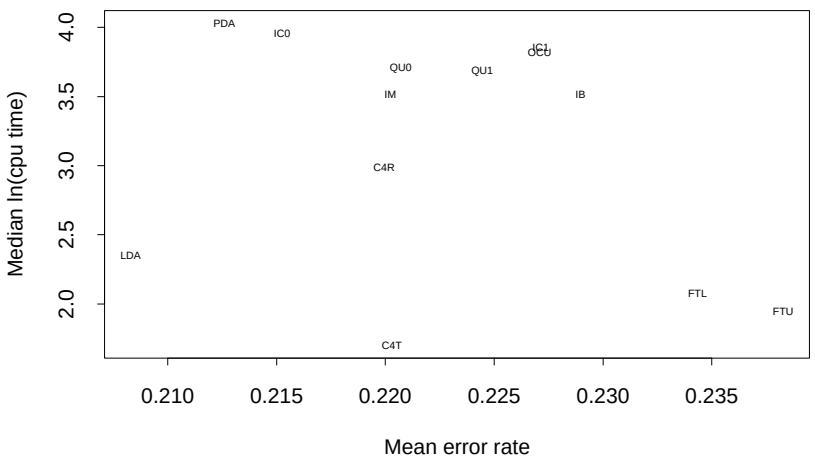


Figure 1: Plots of average error rates versus log-CPU times. The solid vertical line in plot (a) divides the classifiers into two groups: the average error rates of the classifiers in the left group do not differ significantly (at the 90% simultaneous confidence level) from that of POL, which has the minimum average error rate. The dotted horizontal line splits the classifiers into two groups of equal sizes.

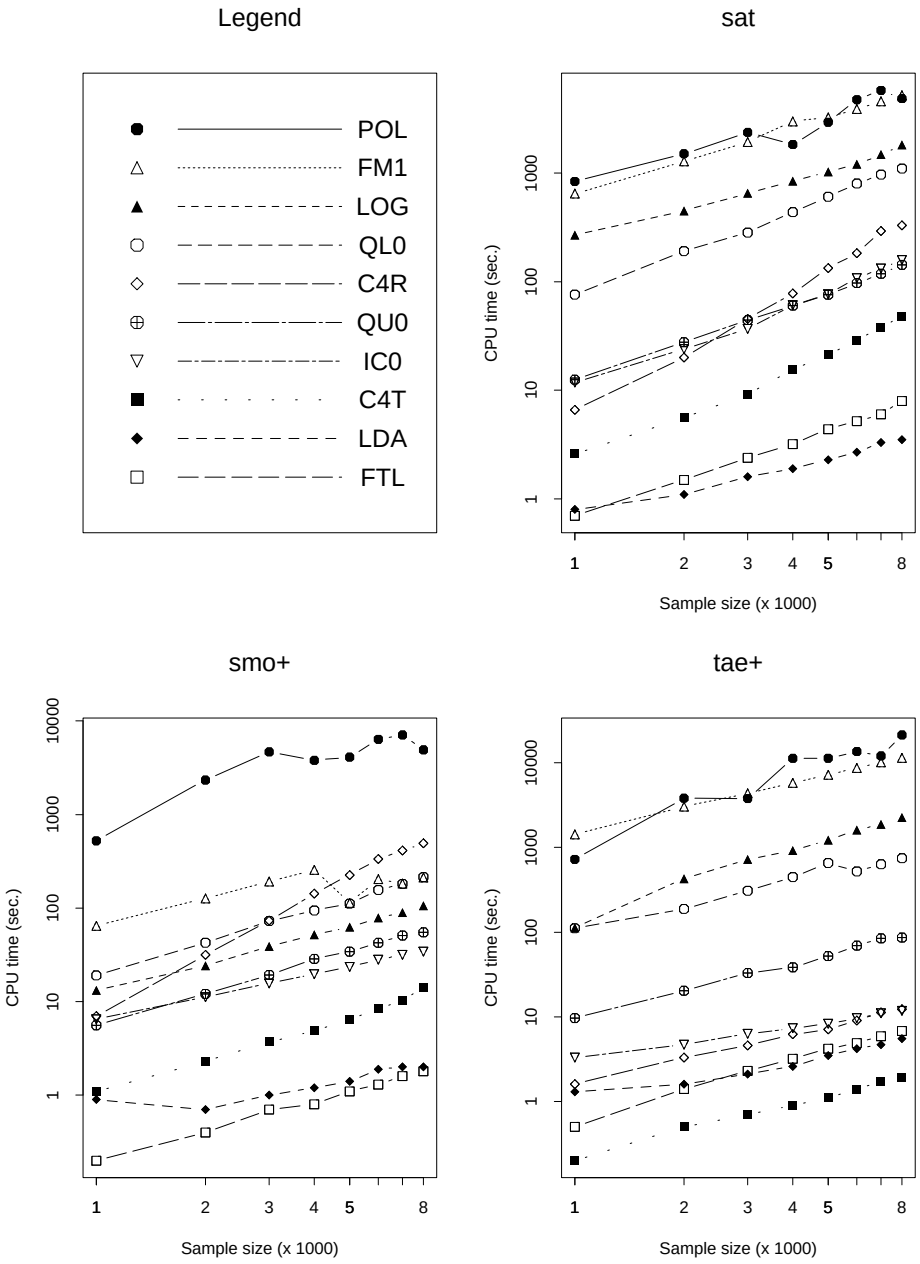


Figure 2: Plots of ln(CPU time) versus ln(N) for selected classifiers. The data in each plot were obtained by perturbing bootstrap samples drawn from the named database.

the QUEST and FACT algorithms is that the former employs the cost-complexity pruning method of CART whereas the latter does not. Our results suggest that some form of bottom-up pruning is essential for low error rates.

If the purpose of constructing a classifier is for data interpretation, then perhaps only decision rules or trees with univariate splits can suffice. Figure 1(a) shows that differences between the average error rates of the decision rule and tree classifiers are not statistically significant. The one with the lowest average error rate is `IC0`. `C4R`, `C4T` and `QU0`, with almost identical average rates, are next best. Any of these four classifiers should provide good classification accuracy. `C4T` is the fastest by far, although it tends to yield trees with twice as many leaves as `IC0` and `QU0`. (One reason for the faster speed of `C4T` is that its pruning algorithm does not require cross-validation.) `C4R` is the next fastest, but Figure 2 shows that it does not scale well with larger sample sizes. `IC0` and `QU0` are very similar in terms of speed, error rate and size of trees. However, `QU0` is much less likely to produce trees with spurious splits (Loh and Shih, 1997).

Acknowledgments

We are grateful to many individuals for helping us with this project. J. R. Quinlan answered our questions on C4.5. W. Buntine and S. K. Murthy gave advice on the installation of the IND and OC1 programs, respectively. B. Schulmeister and C. Kooperberg provided the CAL5 program and the `polyclass` library, respectively. R. C. Holte pointed us to T1 and P. Auer explained its memory requirements. J. W. Molyneaux kindly provided the 1987 National Indonesia Contraceptive Prevalence Survey data.

References

Agresti, A. (1990). *Categorical Data Analysis*, John Wiley & Sons, New York, NY.

Aronis, J. M. and Provost, F. J. (1997). Increasing the efficiency of data mining algorithms with breadth-first marker propagation, *in* D. Heckerman, H. Mannila, D. Pregibon and R. Uthurusamy (eds), *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, pp. 119–122.

Auer, P., Holte, R. C. and Maass, W. (1995). Theory and applications of agnostic PAC-learning with small decision trees, *in* A. Prieditis and S. Russell (eds), *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kaufmann, San Mateo, CA, pp. 21–29.

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988). *The New S Language*, Wadsworth.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, New York, NY.

Box, G. E. P., Hunter, W. G. and Hunter, J. S. (1978). *Statistics for Experimenters*, Wiley, New York.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*, Chapman and Hall, New York, NY.

Breslow, L. A. and Aha, D. W. (1997). Simplifying decision trees: A survey, *Knowledge Engineering Review* **12**: 1–40.

Brodley, C. E. and Utgoff, P. E. (1992). Multivariate versus univariate decision trees, *Technical Report 92-8*, Department of Computer Science, University of Massachusetts, Amherst, MA.

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

Brown, D. E., Corruble, V. and Pittard, C. L. (1993). A comparison of decision tree classifiers with back-propagation neural networks for multimodal classification problems, *Pattern Recognition* **26**: 953–961.

Bull, S. (1994). Analysis of attitudes toward workplace smoking restrictions, *in* N. Lange, L. Ryan, L. Billard, D. Brillinger, L. Conquest and J. Greenhouse (eds), *Case Studies in Biometry*, John Wiley & Sons, New York, NY, pp. 249–271.

Buntine, W. (1992). Learning classification trees, *Statistics and Computing* **2**: 63–73.

Buntine, W. and Caruana, R. (1992). *Introduction to IND Version 2.1 and Recursive Partitioning*, NASA Ames Research Center, Moffet Field, CA.

Clark, L. A. and Pregibon, D. (1993). Tree-based models, *in* J. M. Chambers and T. J. Hastie (eds), *Statistical Models in S*, Chapman & Hall, New York, NY, pp. 377–419.

Cohen, W. W. (1995). Fast effective rule induction, *in* A. Prieditis and S. Russell (eds), *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, pp. 115–123.

Curram, S. P. and Mingers, J. (1994). Neural networks, decision tree induction and discriminant analysis: an empirical comparison, *Journal of the Operational Research Society* **45**: 440–450.

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion), *Annals of Statistics* **19**: 1–141.

Hand, D. J. (1997). *Construction and Assessment of Classification Rules*, John Wiley & Sons, Chichester, England.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic prices and the demand for clean air, *Journal of Environmental Economics and Management* **5**: 81–102.

Hastie, T. and Tibshirani, R. (1996). Discriminant analysis by Gaussian mixtures, *Journal of the Royal Statistical Society, Series B* **58**: 155–176.

Hastie, T., Buja, A. and Tibshirani, R. (1995). Penalized discriminant analysis, *Annals of Statistics* **23**: 73–102.

Hastie, T., Tibshirani, R. and Buja, A. (1994). Flexible discriminant analysis by optimal scoring, *Journal of the American Statistical Association* **89**: 1255–1270.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets, *Machine Learning* **11**: 63–90.

Johnson, R. A. and Wichern, D. W. (1992). *Applied Multivariate Statistical Analysis*, 3 edn, Prentice Hall, Englewood Cliffs, NJ.

Kohonen, T. (1995). *Self-Organizing Maps*, Springer-Verlag, Heidelberg.

Kooperberg, C., Bose, S. and Stone, C. J. (1997). Polychotomous regression, *Journal of the American Statistical Association* **92**: 117–127.

Lerman, C., Molyneaux, J. W., Pangemanan, S. and Iswarati (1991). The determinants of contraceptive method and service point choice, *Secondary Analysis of the 1987 National Indonesia Contraceptive Prevalence Survey*, Vol. 1: Fertility and Family Planning, East-West Population Institute, Honolulu, HI.

Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees, *Statistica Sinica* **7**: 815–840.

Loh, W.-Y. and Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion), *Journal of the American Statistical Association* **83**: 715–728.

Mangasarian, O. L. and Wolberg, W. H. (1990). Cancer diagnosis via linear programming, *Siam News* **23**: 1–18.

Merz, C. J. and Murphy, P. M. (1996). *UCI Repository of Machine Learning Databases*, Department of Information and Computer Science, University of California, Irvine, CA. (`http://www.ics.uci.edu/~mlearn/MLRepository.html`).

Michie, D., Spiegelhalter, D. J. and Taylor, C. C. (eds) (1994). *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, London.

Miller, Jr., R. G. (1981). *Simultaneous Statistical Inference*, second edn, Springer-Verlag, New York.

Müller, W. and Wysotzki, F. (1994). Automatic construction of decision trees for classification, *Annals of Operations Research* **52**: 231–247.

Müller, W. and Wysotzki, F. (1997). The decision-tree algorithm CAL5 based on a statistical approach to its splitting algorithm, *in* G. Nakhaeizadeh and C. C. Taylor (eds), *Machine Learning and Statistics: The Interface*, John Wiley & Sons, New York, NY, pp. 45–65.

Murthy, S. K., Kasif, S. and Salzberg, S. (1994). A system for induction of oblique decision trees, *Journal of Artificial Intelligence Research* **2**: 1–33.

Oates, T. and Jensen, D. (1997). The effects of training set size on decision tree complexity, *in* J. D. H. Fisher (ed.), *Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, pp. 254–262.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5, *Journal of Artificial Intelligence Research* **4**: 77–90.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Sarle, W. S. (1994). Neural networks and statistical models, *Proceedings of the Nineteenth Annual SAS Users Groups International Conference*, SAS Institute, Inc., Cary, NC, pp. 1538–1550. (`ftp://ftp.sas.com/pub/neural/neural1.ps`).

SAS Institute, Inc. (1990). *SAS/STAT User's Guide, Version 6*, Vol. 1 & 2, SAS Institute, Inc., Cary, NC.

Shavlik, J. W., Mooney, R. J. and Towell, G. G. (1991). Symbolic and neural learning algorithms: an empirical comparison, *Machine Learning* **6**: 111–144.

Venables, W. N. and Ripley, B. D. (1997). *Modern Applied Statistics with S-Plus*, 2 edn, Springer, New York, NY.

Wolberg, W. H., Tanner, M. A. and Loh, W.-Y. (1988). Diagnostic schemes for fine needle aspirates of breast masses, *Analytical and Quantitative Cytology and Histology* **10**: 225–228.

Wolberg, W. H., Tanner, M. A. and Loh, W.-Y. (1989). Fine needle aspiration for breast mass diagnosis, *Archives of Surgery* **124**: 814–818.

Wolberg, W. H., Tanner, M. A., Loh, W.-Y. and Vanichsetakul, N. (1987). Statistical approach to fine needle aspiration diagnosis of breast masses, *Acta Cytologica* **31**: 737–741.

Appendix

Table 4: Minimum, maximum, and naive (plurality rule) error rates for each dataset. A '√'-mark indicates that the classifier has an error rate within one standard error of the minimum for the dataset. A 'X'-mark indicates that the classifier has the worst error rate for the dataset. The average error rate for each classifier is given in the second row.

	Decision trees and rules																					Statistical classifiers									Nets		Error rates		
	QUO	QU1	QLO	QL1	FTU	FTL	C4T	C4R	IB	IBQ	IM	IMQ	ICO	IC1	OCU	OCL	OCM	STO	ST1	LMT	CAL	T1	LDA	QDA	NN	LOG	FM1	FM2	PDA	MDA	POL	LVQ	RBF	Min	Max	Naive
Mean	.221	.224	.202	.205	.238	.234	.220	.220	.229	.219	.220	.219	.215	.227	.227	.260	.230	.232	.233	.220	.270	.354	.208	.301	.281	.204	.242	.217	.213	.207	.195	.269	.257			
#√	9	7	12	12	4	13	7	8	3	7	5	6	4	5	9	4	4	5	5	2	5	4	10	4	3	11	13	12	10	9	15	4	8			
#X	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3	1	0	1	0	0	11	0	3	4	0	2	1	0	1	0	4	0			
bcw			√	√						√						X										√						√	√	.028	.085	.350
bcw+			√	√																		X				√	√	√		√		√	√	.029	.076	
cmc	√	√																							X						√			.434	.601	.573
cmc+	√	√																							X						√			.432	.577	
dna						√																X					√	√			√			.047	.379	.492
dna+						√																X						√			√			.044	.379	
hea			√	√		√																	√						√			X		.137	.341	.444
hea+			√	√		√																	√						√			X		.148	.311	
bos			√	√			√	√				√					√					√			√		√				√	X	√	.221	.314	.657
bos+							√								√												√	X			√			.225	.422	
led	√		√			√	√	√		√	√	√			√	√						X	√	√		√	√	√	√	√	√			.268	.816	.890
led+						√									√							X	√			√	√	√	√	√	√			.265	.813	
bld								√								√	√					X					√	√			√			.279	.432	.419
bld+																						X									√			.286	.441	
pid	√		√	√		√		√			√		√		√	X		√			√		√	√		√	√		√	√	√		√	.221	.310	.333
pid+	√	√	√	√		√		√				√				X							√			√	√	√	√	√	√			.217	.318	
sat																						X										√		.098	.400	.765
sat+																												√				X		.116	.410	
seg										√															√		X							.022	.515	.857
seg+										√																	X							.026	.574	
smo	√	√	√	√	√	√	√				√	√		√	√	√	√	√	√		√	√	√	X		√	√	√	√	√	√	√		√	.304	.454	.305
smo+	√	√	√	√	√	√					√		√	√	√	√	√	√	√	√	√	√	√		X	√		√	√	√	√		√	.305	.445	
thy							√	√	√				√	√	√			√	√					X										.006	.890	.073
thy+							√	√	√		√	√												X										.005	.875	
veh																						X		√										.145	.487	.739
veh+																						X		√										.155	.487	
vot	√	√		√	√	√				√		√		√	√				√		√	√					√	√	√	X				.037	.062	.386
vot+	√	√	√	√	√	√	√	√		√		√		√	√		X	√	√	√	√	√	√		√	√	√	√	√		√		√	.043	.066	
wav																						X				√							√	.151	.477	.667
wav+						√																			X	√	√			√		√		.160	.446	
tae						X				√															√									.325	.693	.656
tae+			√	√					√										X				√							√				.411	.696	

Table 5: DEC Alpha-equivalent CPU and relative CPU times of classifiers. The first and second rows give the median CPU time and rank for each classifier. An entry of 'x' in the each of the subsequent rows indicates that a classifier is between 10^{x-1}–10^{x} times slower than the fastest classifier for the dataset. The fastest classifier is denoted by an entry of '0'. The minimum and maximum CPU times are given in the last two columns. 's', 'm', 'h', 'd' denote seconds, minutes, hours and days, respectively.

	Decision trees and rules																										Statistical classifiers										Nets		CPU time	
	QU0	QU1	QL0	QL1	FTU	FTL	C4T	C4R	IB	IBO	IM	IMO	IC0	IC1	OCU	OCL	OCM	ST0	ST1	LMT	CAL	T1	LDA	QDA	NN	LOG	FM1	FM2	PDA	MDA	POL	LVQ	RBF	Min	Max
Median CPU	41s	40s	3.7m	3.7m	7s	8s	5s	20s	34s	27.5m	34s	33.9m	52s	47s	46s	14.9m	13.7m	15.1m	14.4m	5.7m	1.3h	36s	10s	15s	20s	4m	15.6m	3.8h	56s	3m	3.2h	1.1m	11.3h	5s	11.3h
Rank	12	11	19	20	2	3	1	6	8	28	9	29	15	14	13	25	23	26	24	22	30	10	4	5	7	21	27	32	16	18	31	17	33		
bcw+	1	1	2	2	1	1	0	1	1	3	1	3	2	2	1	3	3	3	3	2	3	1	1	1	1	2	3	3	2	2	4	2	4	4s	2.7h
cmc+	2	2	2	2	0	1	1	2	1	3	1	3	2	2	2	3	3	3	3	2	4	2	1	1	1	2	3	4	2	2	4	2	4	12s	23.9h
dna+	2	2	4	4	1	2	0	2	1	2	1	2	1	1	3	3	3	3	3	4	5	0	2	2	3	3	5	7	3	3	5	4	6	2s	39.6d
hea+	1	1	2	2	1	1	0	1	1	4	1	3	2	2	1	2	2	3	3	2	4	1	1	1	1	2	3	3	1	2	3	1	4	4s	3.3h
bos+	1	1	2	2	0	0	1	1	1	3	1	3	2	2	1	3	3	3	3	2	4	2	1	1	1	2	3	2	1	2	4	1	4	9s	5.5h
led+	2	2	3	3	1	1	1	2	1	3	1	3	2	1	2	3	3	3	3	3	4	0	1	2	2	3	3	4	2	3	4	2	5	1s	12.4h
bld+	1	1	2	2	1	0	1	1	1	4	1	4	2	2	1	3	3	3	3	2	3	1	1	1	1	1	2	3	1	2	3	1	3	5s	1.5h
pid+	1	1	2	2	0	0	1	1	1	3	1	3	2	2	1	3	3	3	3	2	3	1	1	1	1	1	2	3	1	2	4	1	3	7s	2.5h
sat+	2	2	3	3	0	1	1	2	2	3	2	3	2	2	2	4	3	3	3	3	4	2	1	1	2	3	4	5	2	2	4	3	5	8s	6.1d
seg+	2	2	2	2	1	1	1	1	2	2	2	2	2	2	1	3	3	2	2	2	4	2	0	1	1	3	3	4	2	2	4	2	5	28s	6.3d
smo+	2	2	2	2	0	0	1	2	2	3	2	3	2	2	2	3	3	3	3	3	4	2	1	1	1	2	3	4	2	2	5	2	4	1s	3.8h
thy+	2	2	2	2	1	1	0	1	1	2	1	2	1	1	2	3	2	3	3	3	3	2	1	1	2	2	3	4	2	2	5	3	5	3s	16.1h
veh+	1	1	2	2	1	1	1	1	2	3	2	3	2	2	1	3	3	3	3	2	4	3	0	1	1	2	4	4	1	2	4	2	4	14s	1.2d
vot+	2	2	3	3	1	1	0	1	2	4	2	3	2	2	2	3	3	3	3	3	4	0	1	1	2	3	4	5	2	3	4	2	5	2s	2.1d
wav+	1	1	2	2	1	1	1	1	1	3	1	3	2	2	1	2	2	2	2	2	3	2	0	1	1	2	3	4	1	2	4	1	4	4s	4.3h
tae+	1	1	2	2	0	0	1	1	1	2	1	3	1	1	1	1	2	4	4	2	3	1	1	1	1	2	3	4	1	2	3	1	4	6s	10.2h

Table 7: Number of terminal nodes (leaves) of decision tree algorithms. The numbers for bcw, cmc, hea, bos, bld, pid, seg, veh, vot, and tae are averages from 10-fold cross-validation experiments.

	Data																	
Classifier	bcw	cmc	dna	hea	bos	led	bld	pid	sat	seg	sec	thy	veh	vot	wav	tae	Average	Median
QU0	9	11	13	14	24	31	29	3	140	66	1	20	68	3	34	21	30.4	20.5
QU1	7	10	13	6	7	24	10	2	112	44	1	18	31	2	16	15	19.9	11.5
QL0	3	24	7	2	6	31	6	5	30	39	1	13	16	2	5	10	12.5	6.5
QL1	2	11	5	2	3	15	4	2	11	21	1	6	8	2	5	6	6.5	5.0
FTU	5	23	12	6	14	21	5	6	102	53	1	19	38	2	26	9	21.4	13.0
FTL	3	3	5	3	4	12	2	3	49	18	1	13	22	2	3	1	9.0	3.0
C4T	11	143	97	23	36	29	26	18	216	42	1	12	65	10	54	79	53.9	32.5
IB	31	894	325	48	79	103	73	81	348	58	817	17	123	49	66	262	210.9	80.0
IB0	27829	26153	7513	35238	26278	415	26862	23889	3174	13759	9884	129	9683	46695	24295	5548	17959.0	18824.0
IM	29	965	316	48	79	102	73	79	342	80	824	15	123	48	66	392	223.8	79.5
IM0	23939	12325	3582	34839	25500	207	31247	30375	1381	2964	4849	290	1390	19838	18641	11653	13938.8	11989.0
IC0	13	12	13	7	18	22	12	9	63	69	6	11	71	5	14	43	24.2	13.0
IC1	7	8	12	5	6	21	7	5	29	28	1	6	27	2	9	14	11.7	7.5
OCU	19	48	15	6	9	30	4	7	70	39	10	6	51	2	15	39	23.1	15.0
OCL	4	12	10	3	10	14	5	8	22	16	4	11	14	2	4	2	8.8	9.0
OCM	5	8	16	3	7	14	5	5	33	26	2	5	41	2	15	20	12.9	7.5
ST0	7	18	13	9	13	22	16	9	33	16	1	6	30	3	26	13	14.7	13.0
ST1	5	8	12	3	8	19	7	4	19	15	1	6	24	2	17	7	9.8	7.5
LMT	3	46	3	3	11	38	12	11	20	6	61	6	13	2	4	5	15.2	8.5
CAL	7	15	37	7	25	40	22	11	178	72	34	16	79	2	62	34	40.1	29.5
T1	3	5	5	4	5	3	3	3	8	9	5	5	6	4	5	27	6.2	5.0
Average	2473.4	1940.1	572.6	3346.6	2483.0	57.8	2782.4	2596.9	303.8	830.5	786.0	30.0	567.8	3175.2	2065.8	866.7		
Median	7	15	13	6	11	24	10	7	63	39	2	12	38	2	16	20		

Table 9: Differences in the number of terminal nodes (leaves) of decision tree algorithms for datasets with and without noise. A negative difference means that the tree with noise is shorter than the one without noise.

	Data																	
Classifier	bcw	cmc	dna	hea	bos	led	bld	pid	sat	seg	smo	thy	veh	vot	wav	tae	Average	Median
QU0	-1	3	0	3	-14	38	-17	-1	-5	-10	0	0	-14	0	-19	-8	-2.8	-1.0
QU1	-1	0	-1	2	-2	-8	-5	0	-43	-8	0	0	-1	0	-6	-9	-5.1	-1.0
QL0	-1	-4	-2	0	-2	1	0	-2	-12	-24	0	-7	-3	0	0	-6	-3.9	-2.0
QL1	0	-4	0	0	0	0	-2	0	1	-9	0	0	-2	0	-2	-3	-1.3	0.0
FTU	0	3	0	1	1	-1	1	0	-1	0	0	0	1	0	-1	1	0.3	0.0
FTL	0	0	1	0	-1	5	0	-1	0	0	0	-1	0	0	0	0	0.2	0.0
C4T	0	116	0	1	1	91	7	13	8	0	222	0	-1	1	-9	30	30.0	1.0
IB	-5	-399	-3	-9	-15	583	-17	-19	-37	-2	-343	0	-19	-12	-6	-56	-22.4	-13.5
IB0	-7113	-11878	-1304	-2369	-3897	13441	-685	3147	2603	405	-166	-18	-5845	-14858	-21228	6782	-2686.4	-994.5
IM	-5	-422	0	-10	-15	589	-19	-19	-30	-25	-390	0	-20	-12	-6	-111	-30.9	-17.0
IM0	-8181	3806	-598	-15113	4600	4921	-8466	-1126	4726	4234	1001	357	1261	-17810	-8795	10399	-1549.0	679.0
IC0	-1	-1	0	1	-10	31	-7	0	-7	-7	-5	0	-26	1	-6	-19	-3.5	-3.0
IC1	-2	0	0	0	-2	7	-3	-1	-3	1	0	0	-8	0	-2	-9	-1.4	-0.5
OCU	-2	-15	0	0	2	3	1	12	-5	-1	-8	0	-6	0	1	-24	-2.6	0.0
OCL	-1	2	-7	-1	-6	2	0	0	-4	-3	-2	-6	-3	0	8	0	-1.3	-1.0
OCM	-1	0	-2	1	4	0	1	-1	17	0	0	-1	-16	0	5	-9	-0.1	0.0
ST0	2	-9	0	-3	-3	4	-6	2	-6	0	0	0	-2	0	8	-3	-1.0	0.0
ST1	0	-1	0	1	-3	-1	-4	0	4	0	0	0	-1	0	-1	-1	-0.4	0.0
LMT	-1	-20	0	-1	-7	-28	-7	-5	-2	-3	-51	-1	-6	0	-1	-3	-8.5	-3.0
CAL	4	5	-1	0	1	-27	-6	2	250	12	-26	7	3	0	-47	-10	10.4	0.5
T1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
Average	-729.0	-419.9	-91.3	-833.1	30.1	935.8	-439.7	95.3	355.0	217.1	11.0	15.7	-224.2	-1556.7	-1433.7	806.7		
Median	-1	0	0	0	-2	3	-4	0	-2	0	0	0	-3	0	-2	-3		

Table 10: Breast cancer Wisconsin (bcw, 2 classes, 9 numerical attributes, 683 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.350. The noise attributes are 9 independent uniform integer variates between 1 and 10.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.0439	22.5	0.0439	16	0:00:39
QU1	0.0410	16	0.0469	19	0:00:37
QL0	0.0337	5	0.0293	1.5	0:01:23
QL1	0.0308	2	0.0293	1.5	0:01:20
FTU	0.0498	28.5	0.0498	22	0:00:09
FTL	0.0439	22.5	0.0424	14	0:00:10
C4T	0.0425	21	0.0513	24	0:00:04
C4R	0.0395	12.5	0.0395	12	0:00:17
IB	0.0424	19	0.0600	31	0:00:36
IBO	0.0336	3	0.0425	15	0:21:56
IM	0.0424	19	0.0453	17	0:00:35
IMO	0.0380	9.5	0.0409	13	0:28:30
IC0	0.0453	24	0.0542	27	0:00:59
IC1	0.0468	25	0.0541	26	0:00:53
OCU	0.0423	17	0.0584	29	0:00:21
OCL	0.0848	33	0.0526	25	0:15:59
OCM	0.0408	14	0.0585	30	0:13:28
ST0	0.0512	30	0.0483	20	0:11:53
ST1	0.0498	28.5	0.0512	23	0:11:53
LMT	0.0351	7	0.0468	18	0:03:27
CAL	0.0706	31	0.0658	32	0:43:46
T1	0.0760	32	0.0760	33	0:00:05
Statistical					
LDA	0.0394	11	0.0379	9	0:00:11
QDA	0.0481	26	0.0496	21	0:00:11
NN	0.0482	27	0.0555	28	0:00:15
LOG	0.0337	5	0.0352	8	0:01:10
FM1	0.0366	8	0.0350	6.5	0:08:18
FM2	0.0380	9.5	0.0321	3	1:02:26
PDA	0.0395	12.5	0.0380	10	0:00:48
MDA	0.0409	15	0.0350	6.5	0:02:38
POL	0.0424	19	0.0383	11	2:41:04
Neural Network					
LVQ	0.0278	1	0.0336	5	0:00:58
RBF (2 hidden units)	0.0337	5	0.0322	4	1:56:54

Table 11: Contraceptive method choice (cmc, 3 classes, 2 numerical attributes, 7 categorical attributes, 1473 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.573. Error rates greater than that of the plurality rule are printed in *italics*. The noise attributes are 6 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.439	2.5	0.436	2	0:02:15
QU1	0.439	2.5	0.442	3	0:02:07
QL0	0.520	31	0.499	20.5	0:13:19
QL1	0.513	28	0.507	26	0:13:53
FTU	0.498	23	0.496	19	0:00:12
FTL	0.496	22	0.501	24	0:00:13
C4T	0.483	16	0.516	27	0:00:27
C4R	0.450	6	0.469	10	0:07:15
IB	0.505	26	0.523	30	0:01:47
IBO	0.509	27	0.500	22.5	0:54:18
IM	0.447	4	0.457	8	0:01:48
IMO	0.463	12	0.480	13	1:05:26
IC0	0.451	7.5	0.446	5	0:03:31
IC1	0.449	5	0.445	4	0:03:03
OCU	0.475	14	0.489	15.5	0:03:16
OCL	0.504	25	0.502	25	0:42:21
OCM	0.478	15	0.478	12	0:43:34
ST0	0.457	9	0.462	9	1:36:46
ST1	0.451	7.5	0.456	7	1:26:02
LMT	0.514	29	0.544	32	0:19:53
CAL	0.503	24	0.485	14	21:15:04
T1	0.518	30	0.518	29	0:03:58
Statistical					
LDA	0.492	20.5	0.500	22.5	0:00:15
QDA	0.541	32	0.541	31	0:00:20
NN	*0.601*	33	*0.577*	33	0:00:31
LOG	0.489	18	0.494	17.5	0:07:34
FM1	0.466	13	0.472	11	0:43:54
FM2	0.460	11	0.455	6	6:27:28
PDA	0.492	20.5	0.499	20.5	0:02:03
MDA	0.486	17	0.494	17.5	0:06:33
POL	0.434	1	0.432	1	17:31:14
Neural Network					
LVQ	0.491	19	0.517	28	0:04:40
RBF (4 hidden units)	0.458	10	0.489	15.5	23:53:45

Table 12: StatLog DNA (`dna`, 3 classes, 60 categorical attributes, 2000 observations). Error rates are estimated from a test set of 1186 observations. Standard errors range from 0.0059 to 0.014. Plurality rule error rate is 0.492. The noise attributes are 20 independent uniform integer variates between 1 and 4.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QUO	0.0582	10.5	0.0582	6	0:00:50
QU1	0.0582	10.5	0.0616	9	0:00:46
QLO	0.0683	18.5	0.0742	18.5	2:10:05
QL1	0.0649	17	0.0742	18.5	2:08:01
FTU	0.115	25	0.115	25	0:00:05
FTL	0.0472	2	0.0489	2	0:00:23
C4T	0.0759	21	0.0759	20	0:00:02
C4R	0.0683	18.5	0.0683	17	0:00:22
IB	0.0793	22	0.0801	21	0:00:06
IBO	0.0573	7	0.0632	11.5	0:00:22
IM	0.0835	23	0.0835	22	0:00:06
IMO	0.0565	5	0.0641	13.5	0:00:21
ICO	0.0582	10.5	0.0582	6	0:00:17
IC1	0.0616	14.5	0.0616	9	0:00:13
OCU	0.0919	24	0.0927	24	0:12:25
OCL	0.196	29	0.167	28	0:21:21
OCM	0.175	28	0.158	27	0:30:47
STO	0.0582	10.5	0.0582	6	0:13:12
ST1	0.0616	14.5	0.0616	9	0:12:50
LMT	0.0742	20	0.0843	23	0:33:52
CAL	0.278	30	0.277	29	6:43:22
T1	0.379	33	0.379	33	0:00:02
Statistical					
LDA	0.0600	13	0.0641	13.5	0:00:33
QDA	0.137	27	0.298	30	0:02:05
NN	0.317	32	0.375	32	0:05:06
LOG	0.0641	16	0.0675	16	0:11:40
FM1	0.0523	4	0.0548	4	17:50:48
FM2	0.0472	2	0.0438	1	950:46:10
PDA	0.0573	7	0.0632	11.5	0:10:26
MDA	0.0573	7	0.0658	15	0:09:10
POL	0.0472	2	0.0497	3	9:59:15
Neural Network					
LVQ	0.132	26	0.154	26	0:36:08
RBF (5 hidden units)	0.293	31	0.303	31	122:45:41

Table 13: StatLog heart disease (hea, 2 classes, 7 numerical attributes, 6 categorical attributes, 270 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.444. The noise attributes are 7 independet standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.241	28	0.233	23.5	0:00:25
QU1	0.226	23.5	0.256	29	0:00:25
QL0	0.137	1.5	0.159	3	0:01:31
QL1	0.137	1.5	0.163	4.5	0:01:29
FTU	0.233	26.5	0.230	22	0:00:06
FTL	0.148	4.5	0.148	1	0:00:07
C4T	0.196	13.5	0.193	11	0:00:04
C4R	0.200	15.5	0.200	14	0:00:08
IB	0.222	21.5	0.226	20.5	0:00:31
IB0	0.196	13.5	0.189	10	1:11:24
IM	0.200	15.5	0.196	12.5	0:00:32
IM0	0.204	17	0.211	18	0:52:00
IC0	0.207	18	0.207	16	0:00:46
IC1	0.219	19.5	0.244	27	0:00:40
OCU	0.230	25	0.241	26	0:00:15
OCL	0.189	10	0.226	20.5	0:03:53
OCM	0.222	21.5	0.207	16	0:04:21
ST0	0.256	30	0.233	23.5	0:09:49
ST1	0.233	26.5	0.215	19	0:09:49
LMT	0.163	7.5	0.170	6.5	0:02:13
CAL	0.270	31.5	0.256	29	1:13:32
T1	0.270	31.5	0.270	32	0:00:18
Statistical					
LDA	0.141	3	0.156	2	0:00:09
QDA	0.248	29	0.259	31	0:00:09
NN	0.226	23.5	0.256	29	0:00:10
LOG	0.159	6	0.185	9	0:01:09
FM1	0.193	11.5	0.196	12.5	0:07:40
FM2	0.219	19.5	0.237	25	1:05:54
PDA	0.148	4.5	0.163	4.5	0:00:36
MDA	0.163	7.5	0.170	6.5	0:01:55
POL	0.174	9	0.207	16	0:27:43
Neural Network					
LVQ	0.341	33	0.311	33	0:00:21
RBF (4 hidden units)	0.193	11.5	0.174	8	3:18:58

Table 14: Boston housing (bos, 3 classes, 12 numerical attributes, 1 binary attribute, 506 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.657. The noise attributes are 12 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.261	26.5	0.267	14.5	0:01:02
QU1	0.286	31	0.286	24	0:01:02
QL0	0.233	6.5	0.248	5	0:04:18
QL1	0.233	6.5	0.252	6	0:04:19
FTU	0.261	26.5	0.263	12	0:00:09
FTL	0.254	20.5	0.254	7	0:00:09
C4T	0.221	1	0.225	1	0:00:17
C4R	0.236	8	0.261	10.5	0:00:30
IB	0.251	18.5	0.274	19	0:01:15
IB0	0.249	16.5	0.259	9	1:25:32
IM	0.241	11.5	0.258	8	0:01:14
IM0	0.231	5	0.273	17.5	1:50:50
IC0	0.254	20.5	0.264	13	0:02:13
IC1	0.266	28.5	0.280	23	0:01:54
OCU	0.241	11.5	0.233	2	0:00:41
OCL	0.272	30	0.358	32	0:16:01
OCM	0.239	9.5	0.267	14.5	0:16:52
ST0	0.259	24.5	0.275	20	0:23:34
ST1	0.255	22	0.261	10.5	0:23:17
LMT	0.251	18.5	0.312	29	0:07:28
CAL	0.259	24.5	0.293	26	2:53:49
T1	0.287	32	0.287	25	0:03:53
Statistical					
LDA	0.249	16.5	0.279	22	0:00:10
QDA	0.266	28.5	0.273	17.5	0:00:12
NN	0.227	4	0.335	31	0:00:15
LOG	0.243	13.5	0.268	16	0:04:05
FM1	0.239	9.5	0.243	4	0:21:36
FM2	0.243	13.5	0.422	33	0:04:44
PDA	0.247	15	0.278	21	0:01:05
MDA	0.257	23	0.294	27	0:03:26
POL	0.225	2.5	0.239	3	5:28:53
Neural Network					
LVQ	0.314	33	0.322	30	0:01:00
RBF (3 hidden units)	0.225	2.5	0.298	28	5:18:29

Table 15: LED display (`led`, 10 classes, 7 binary attributes, 2000 observations). Error rates are estimated from a test set of 4000 observations. Standard errors range from 0.0061 to 0.0079. Plurality rule error rate is 0.890. The noise attributes are 17 independent binary variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.273	8	0.278	12	0:00:35
QU1	0.280	21.5	0.293	21	0:00:33
QL0	0.275	17	0.290	17.5	0:02:33
QL1	0.285	25	0.289	15	0:02:27
FTU	0.285	25	0.288	14	0:00:07
FTL	0.271	4	0.266	2.5	0:00:07
C4T	0.271	4	0.290	17.5	0:00:02
C4R	0.275	17	0.309	23.5	0:00:33
IB	0.276	19	0.342	26	0:00:08
IBO	0.274	12.5	0.363	27	0:04:53
IM	0.274	12.5	0.290	17.5	0:00:07
IMO	0.274	12.5	0.296	22	0:02:16
IC0	0.279	20	0.277	11	0:00:11
IC1	0.286	27	0.274	10	0:00:07
OCU	0.272	6	0.269	8	0:00:51
OCL	0.275	17	0.369	28	0:13:57
OCM	0.280	21.5	0.329	25	0:13:55
ST0	0.285	25	0.280	13	0:08:32
ST1	0.289	28	0.290	17.5	0:08:52
LMT	0.284	23	0.309	23.5	0:04:53
CAL	0.301	30	0.374	29	1:22:16
T1	0.816	33	0.813	33	0:00:01
Statistical					
LDA	0.271	4	0.265	1	0:00:05
QDA	0.273	8	0.292	20	0:00:11
NN	0.294	29	0.403	30	0:00:25
LOG	0.269	2	0.266	2.5	0:04:17
FM1	0.274	12.5	0.271	9	0:04:21
FM2	0.268	1	0.268	6.5	1:46:35
PDA	0.274	12.5	0.267	4.5	0:00:38
MDA	0.273	8	0.267	4.5	0:04:34
POL	0.274	12.5	0.268	6.5	0:54:13
Neural Network					
LVQ	0.313	31	0.413	31	0:01:13
RBF (3 hidden units)	0.446	32	0.430	32	12:25:14

Table 16: BUPA liver disorders (bld, 2 classes, 6 numerical attributes, 345 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.419. Error rates greater than that of the plurality rule are printed in *italics*. The noise attributes are 9 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QUO	0.373	27	0.417	29	0:00:30
QU1	0.376	28	0.382	26	0:00:29
QLO	0.309	8.5	0.381	25	0:01:10
QL1	0.323	17	0.357	21	0:01:11
FTU	0.401	29.5	0.402	28	0:00:06
FTL	0.413	31	0.419	30	0:00:05
C4T	0.308	7	0.329	9	0:00:06
C4R	0.292	5	0.320	4.5	0:00:12
IB	0.328	21	0.340	12.5	0:00:41
IBO	0.322	15.5	0.333	10	1:28:47
IM	0.320	13.5	0.340	12.5	0:00:40
IMO	0.312	11	0.336	11	1:26:18
ICO	0.327	20	0.327	7	0:00:58
IC1	0.319	12	0.313	2	0:00:52
OCU	0.350	25	0.347	19	0:00:13
OCL	0.296	6	0.328	8	0:08:26
OCM	0.279	1	0.367	23	0:08:30
STO	0.311	10	0.326	6	0:15:50
ST1	0.326	18.5	0.351	20	0:14:13
LMT	0.322	15.5	0.362	22	0:02:45
CAL	*0.420*	32	0.398	27	1:06:51
T1	*0.432*	33	*0.441*	33	0:00:26
Statistical					
LDA	0.326	18.5	0.343	15	0:00:08
QDA	0.401	29.5	*0.435*	32	0:00:09
NN	0.370	26	*0.423*	31	0:00:10
LOG	0.309	8.5	0.344	17	0:00:39
FM1	0.289	4	0.314	3	0:03:05
FM2	0.280	2	0.320	4.5	0:20:35
PDA	0.329	22.5	0.343	15	0:00:31
MDA	0.320	13.5	0.374	24	0:01:54
POL	0.286	3	0.286	1	1:12:40
Neural Network					
LVQ	0.329	22.5	0.343	15	0:00:23
RBF (1 hidden units)	0.330	24	0.346	18	0:37:23

Table 17: Pima Indians diabetes (pid, 2 classes, 7 numerical attributes, 532 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.333. The noise attributes are 8 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.233	12.5	0.234	14	0:00:43
QU1	0.243	21.5	0.230	10.5	0:00:43
QL0	0.225	5	0.230	10.5	0:01:49
QL1	0.223	4	0.221	2	0:01:48
FTU	0.247	24.5	0.258	25	0:00:07
FTL	0.221	1.5	0.225	5	0:00:07
C4T	0.242	20	0.252	21	0:00:08
C4R	0.227	8	0.233	12.5	0:00:16
IB	0.252	30	0.278	31	0:00:51
IB0	0.258	31	0.259	26	1:29:08
IM	0.233	12.5	0.246	17	0:00:49
IM0	0.246	23	0.265	28	1:44:21
IC0	0.237	15.5	0.233	12.5	0:01:21
IC1	0.239	19	0.248	18	0:01:13
OCU	0.237	15.5	0.256	23.5	0:00:21
OCL	0.310	33	0.318	33	0:16:00
OCM	0.247	24.5	0.261	27	0:17:19
ST0	0.237	15.5	0.269	30	0:19:06
ST1	0.250	28.5	0.244	16	0:18:30
LMT	0.249	27	0.253	22	0:03:53
CAL	0.226	6.5	0.250	19.5	1:30:23
T1	0.250	28.5	0.250	19.5	0:00:40
Statistical					
LDA	0.221	1.5	0.223	3	0:00:10
QDA	0.238	18	0.256	23.5	0:00:10
NN	0.295	32	0.313	32	0:00:12
LOG	0.230	9.5	0.226	6	0:00:57
FM1	0.222	3	0.224	4	0:04:39
FM2	0.248	26	0.228	8	0:32:16
PDA	0.226	6.5	0.228	8	0:00:40
MDA	0.231	11	0.228	8	0:02:18
POL	0.237	15.5	0.217	1	2:27:44
Neural Network					
LVQ	0.243	21.5	0.267	29	0:00:38
RBF (2 hidden units)	0.230	9.5	0.240	15	1:20:23

Table 18: StatLog satellite image (`sat`, 6 classes, 36 numerical attributes, 4435 observations). Error rates are estimated from a test set of 2000 observations. Standard errors range from 0.0066 to 0.011. Plurality rule error rate is 0.765. The noise attributes are 24 independent uniform integer variates between 20 and 160.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QUO	0.138	6.5	0.137	4	0:02:51
QU1	0.137	5	0.141	5	0:02:48
QL0	0.139	8	0.151	14	0:26:59
QL1	0.150	18	0.157	20	0:26:56
FTU	0.151	19.5	0.152	15	0:00:08
FTL	0.157	23	0.154	16	0:00:15
C4T	0.146	14.5	0.158	21.5	0:00:44
C4R	0.148	17	0.150	13	0:02:01
IB	0.146	14.5	0.144	8	0:01:40
IBO	0.153	21	0.149	11.5	0:15:34
IM	0.144	12.5	0.146	10	0:01:38
IMO	0.177	29	0.143	6.5	0:17:04
ICO	0.138	6.5	0.143	6.5	0:04:40
IC1	0.154	22	0.155	17	0:03:54
OCU	0.158	24	0.158	21.5	0:02:59
OCL	0.250	32	0.268	30	2:19:22
OCM	0.142	10	0.164	26.5	0:22:02
ST0	0.151	19.5	0.156	18.5	0:36:08
ST1	0.168	27	0.159	23	0:35:44
LMT	0.147	16	0.166	28	1:39:42
CAL	0.210	30	0.226	29	4:45:06
T1	0.400	33	0.400	32	0:07:08
Statistical					
LDA	0.160	25	0.163	25	0:00:12
QDA	0.141	9	0.145	9	0:00:31
NN	0.217	31	0.304	31	0:01:43
LOG	0.163	26	0.164	26.5	1:58:16
FM1	0.143	11	0.135	3	7:00:18
FM2	0.129	3	0.116	1	147:06:36
PDA	0.172	28	0.160	24	0:06:46
MDA	0.144	12.5	0.156	18.5	0:06:49
POL	0.136	4	0.124	2	6:03:55
Neural Network					
LVQ	0.0980	1	0.410	33	0:45:26
RBF (8 hidden units)	0.121	2	0.149	11.5	53:00:10

Table 19: StatLog image segmentation (**seg**, 7 classes, 19 numerical attributes, 2310 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.857. The noise attributes are 9 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QUO	0.0459	15	0.0494	14	0:05:17
QU1	0.0494	17	0.0528	16	0:05:38
QLO	0.0429	13.5	0.0481	12	0:21:42
QL1	0.0506	18	0.0502	15	0:21:30
FTU	0.0576	22	0.0576	21	0:00:34
FTL	0.0593	23	0.0593	22	0:00:36
C4T	0.0320	3	0.0359	4.5	0:01:26
C4R	0.0364	7	0.0403	9	0:02:31
IB	0.0342	6	0.0351	3	0:04:41
IBO	0.0247	2	0.0264	1	0:34:34
IM	0.0377	11	0.0359	4.5	0:04:43
IMO	0.0325	4.5	0.0338	2	0:40:56
ICO	0.0372	10	0.0476	11	0:07:30
IC1	0.0532	20	0.0532	17	0:08:34
OCU	0.0368	8	0.0407	10	0:03:43
OCL	0.0770	24	0.0823	25.5	2:10:38
OCM	0.0571	21	0.0571	20	2:04:52
STO	0.0369	9	0.0363	6	0:23:38
ST1	0.0528	19	0.0489	13	0:15:23
LMT	0.0403	12	0.0550	19	0:36:43
CAL	0.124	30	0.137	30	13:28:46
T1	0.360	32	0.360	32	0:42:58
Statistical					
LDA	0.0831	26	0.0844	27	0:00:28
QDA	0.123	29	0.118	29	0:00:48
NN	0.0221	1	0.0823	25.5	0:01:07
LOG	0.0429	13.5	0.0545	18	1:18:28
FM1	0.515	33	0.574	33	4:54:33
FM2	0.0468	16	0.0398	8	32:52:03
PDA	0.0835	27	0.0853	28	0:06:35
MDA	0.0788	25	0.0792	24	0:27:53
POL	0.0325	4.5	0.0364	7	22:56:23
Neural Network					
LVQ	0.0844	28	0.0771	23	0:15:59
RBF (5 hidden units)	0.126	31	0.172	31	150:15:17

Table 20: Attitudes towards workplace smoking restrictions (smo, 3 classes, 5 categorical attributes, 3 numerical attributes, 1855 observations). Error rates are estimated from a test set of 1000 observations. Standard errors range from 0.015 to 0.016. Plurality rule error rate is 0.305. Error rates greater than that of the plurality rule are printed in *italics*. The noise attributes are 7 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QUO	0.305	9.5	0.305	7.5	0:00:18
QU1	0.305	9.5	0.305	7.5	0:00:19
QLO	0.305	9.5	0.305	7.5	0:01:30
QL1	0.305	9.5	0.305	7.5	0:01:31
FTU	0.305	9.5	0.305	7.5	0:00:01
FTL	0.305	9.5	0.305	7.5	0:00:01
C4T	0.305	9.5	*0.405*	30	0:00:04
C4R	*0.353*	28	*0.349*	27	0:01:04
IB	*0.424*	32	*0.411*	31	0:00:18
IBO	*0.393*	30	*0.387*	29	0:04:59
IM	0.305	9.5	*0.308*	16.5	0:00:16
IMO	*0.311*	22	*0.339*	26	0:04:08
ICO	*0.319*	26	0.305	7.5	0:00:31
IC1	0.305	9.5	0.305	7.5	0:00:28
OCU	*0.312*	23	0.305	7.5	0:00:26
OCL	*0.317*	25	0.305	7.5	0:07:54
OCM	0.305	9.5	0.305	7.5	0:08:06
STO	0.305	9.5	0.305	7.5	0:11:12
ST1	0.305	9.5	0.305	7.5	0:11:06
LMT	*0.350*	27	*0.306*	15	0:02:08
CAL	*0.316*	24	*0.310*	19	0:56:18
T1	0.304	1	*0.317*	23.5	0:01:10
Statistical					
LDA	0.305	9.5	*0.311*	21	0:00:04
QDA	*0.454*	33	*0.442*	32	0:00:04
NN	*0.410*	31	*0.445*	33	0:00:09
LOG	0.305	9.5	*0.308*	16.5	0:00:37
FM1	*0.310*	20.5	*0.323*	25	0:04:06
FM2	*0.307*	18.5	*0.311*	21	0:19:21
PDA	0.305	9.5	*0.311*	21	0:00:29
MDA	*0.310*	20.5	*0.309*	18	0:01:05
POL	0.305	9.5	0.305	7.5	3:45:44
Neural Network					
LVQ	*0.366*	29	*0.358*	28	0:00:49
RBF (3 hidden units)	*0.307*	18.5	*0.317*	23.5	1:31:31

Table 21: Thyroid disease (thy, 3 classes, 6 numerical attributes, 15 binary attributes, 3772 observations). Error rates are estimated from a test set of 3428 observations. Standard errors range from 0.0012 to 0.0056. Plurality rule error rate is 0.0729. Error rates greater than that of the plurality rule are printed in *italics*. The noise attributes are 10 independent binary variates and 4 independet uniform variates between 0 and 1.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.00875	12	0.00904	12	0:01:00
QU1	0.00904	13	0.00933	13	0:00:56
QL0	0.0181	17	0.0207	18.5	0:03:37
QL1	0.0204	20	0.0207	18.5	0:03:46
FTU	0.0149	16	0.0155	16	0:00:05
FTL	0.0198	19	0.0190	17	0:00:06
C4T	0.00613	3	0.00467	1	0:00:03
C4R	0.00554	1	0.00496	2.5	0:00:07
IB	0.00583	2	0.00496	2.5	0:00:15
IBO	0.00759	11	0.00759	11	0:02:04
IM	0.00700	9.5	0.00525	4	0:00:14
IMO	0.00700	9.5	0.00583	5	0:02:01
IC0	0.00671	8	0.00671	10	0:00:16
IC1	0.00642	5.5	0.00642	7.5	0:00:16
OCU	0.00642	5.5	0.00642	7.5	0:00:56
OCL	0.0268	22	0.0368	22	0:11:47
OCM	0.0102	14.5	0.0108	14	0:04:43
ST0	0.00642	5.5	0.00642	7.5	0:06:25
ST1	0.00642	5.5	0.00642	7.5	0:06:16
LMT	0.0195	18	0.0228	20	0:06:34
CAL	0.0726	32	*0.0750*	30	0:43:11
T1	0.0400	25	0.0400	24	0:00:32
Statistical					
LDA	0.0619	28	0.0619	27	0:00:06
QDA	*0.890*	33	*0.875*	33	0:00:08
NN	0.0645	30	*0.0961*	32	0:01:21
LOG	0.0405	26	0.0403	25	0:03:59
FM1	0.0347	24	0.0374	23	0:09:51
FM2	0.0228	21	0.0242	21	5:49:25
PDA	0.0621	29	0.0621	28	0:00:56
MDA	0.0598	27	0.0586	26	0:02:18
POL	0.0102	14.5	0.0137	15	8:48:33
Neural Network					
LVQ	0.0712	31	*0.0890*	31	0:06:25
RBF (5 hidden units)	0.0318	23	0.0633	29	16:06:50

Table 22: StatLog vehicle silhouette (`veh`, 4 classes, 18 numerical attributes, 846 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.739. The noise attributes are 12 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QUO	0.299	25	0.284	19	0:02:02
QU1	0.282	21	0.306	24	0:01:59
QLO	0.196	4.5	0.239	10	0:09:41
QL1	0.220	9	0.247	11	0:09:41
FTU	0.310	26	0.318	25	0:00:20
FTL	0.410	32	0.406	31	0:00:24
C4T	0.277	18	0.272	15	0:00:31
C4R	0.260	14.5	0.279	17.5	0:01:04
IB	0.260	14.5	0.296	22	0:02:21
IBO	0.274	17	0.267	13	1:03:24
IM	0.289	22	0.290	21	0:02:22
IMO	0.278	19	0.271	14	0:56:36
ICO	0.265	16	0.304	23	0:04:03
IC1	0.298	23.5	0.326	26	0:03:32
OCU	0.298	23.5	0.286	20	0:01:19
OCL	0.355	29	0.384	30	0:45:59
OCM	0.316	27	0.334	27	0:45:06
STO	0.251	13	0.279	17.5	0:49:02
ST1	0.279	20	0.278	16	0:48:34
LMT	0.215	8	0.234	9	0:11:22
CAL	0.350	28	0.360	28	7:44:05
T1	0.487	33	0.487	33	0:25:58
Statistical					
LDA	0.224	11.5	0.229	8	0:00:14
QDA	0.145	1	0.155	1	0:00:19
NN	0.224	11.5	0.252	12	0:00:24
LOG	0.196	4.5	0.223	6	0:19:19
FM1	0.207	7	0.211	5	4:11:03
FM2	0.204	6	0.172	3	18:22:20
PDA	0.221	10	0.228	7	0:01:43
MDA	0.194	3	0.178	4	0:06:10
POL	0.162	2	0.169	2	22:48:03
Neural Network					
LVQ	0.374	31	0.376	29	0:02:28
RBF (4 hidden units)	0.372	30	0.429	32	28:59:24

Table 23: 1984 United States Congressional voting records (vot, 2 classes, 16 categorical attributes, 435 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.386. The noise attributes are 14 independent uniform integer variates between 1 and 3.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.0435	7.5	0.0458	16.5	0:00:32
QU1	0.0435	7.5	0.0435	5	0:00:31
QL0	0.0503	23.5	0.0435	5	0:08:09
QL1	0.0457	14	0.0435	5	0:08:01
FTU	0.0435	7.5	0.0435	5	0:00:08
FTL	0.0457	14	0.0457	12.5	0:00:12
C4T	0.0480	17.5	0.0503	21.5	0:00:02
C4R	0.0526	28	0.0457	12.5	0:00:12
IB	0.0526	28	0.0554	27	0:00:22
IB0	0.0386	2	0.0506	23	1:07:06
IM	0.0503	23.5	0.0594	29	0:00:22
IM0	0.0367	1	0.0457	12.5	0:07:00
IC0	0.0480	17.5	0.0548	26	0:00:28
IC1	0.0435	7.5	0.0457	12.5	0:00:24
OCU	0.0435	7.5	0.0435	5	0:01:08
OCL	0.0574	31	0.0651	32	0:03:40
OCM	0.0580	32	0.0662	33	0:04:23
ST0	0.0500	21	0.0500	20	0:14:21
ST1	0.0432	3.5	0.0432	1	0:14:31
LMT	0.0483	19	0.0529	24	0:04:51
CAL	0.0457	14	0.0457	12.5	0:46:59
T1	0.0435	7.5	0.0435	5	0:00:02
Statistical					
LDA	0.0458	16	0.0458	16.5	0:00:13
QDA	0.0549	30	0.0617	31	0:00:18
NN	0.0526	28	0.0577	28	0:00:26
LOG	0.0500	21	0.0435	5	0:08:18
FM1	0.0455	11.5	0.0457	12.5	0:57:29
FM2	0.0432	3.5	0.0480	19	49:12:23
PDA	0.0455	11.5	0.0455	9	0:01:59
MDA	0.0617	33	0.0545	25	0:03:23
POL	0.0523	25.5	0.0477	18	0:33:46
Neural Network					
LVQ	0.0500	21	0.0614	30	0:01:18
RBF (7 hidden units)	0.0523	25.5	0.0503	21.5	35:26:12

Table 24: Waveform (wav, 3 classes, 21 numerical attributes, 600 observations). Error rates are estimated from a test set of 3000 observations. Standard errors range from 0.0065 to 0.0091. Plurality rule error rate is 0.667. The noise attributes are 19 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.255	18	0.261	19.5	0:00:14
QU1	0.280	23	0.278	28	0:00:14
QL0	0.162	3.5	0.201	11	0:01:14
QL1	0.162	3.5	0.202	12	0:01:15
FTU	0.291	27	0.276	26.5	0:00:07
FTL	0.179	12.5	0.162	2	0:00:06
C4T	0.261	20	0.272	24	0:00:05
C4R	0.261	20	0.253	17.5	0:00:08
IB	0.285	25	0.270	23	0:00:31
IBO	0.261	20	0.253	17.5	0:07:04
IM	0.284	24	0.261	19.5	0:00:31
IMO	0.243	17	0.243	15	0:08:43
IC0	0.297	29	0.276	26.5	0:00:47
IC1	0.313	30	0.279	29	0:00:43
OCU	0.278	22	0.268	22	0:00:23
OCL	0.224	16	0.239	14	0:05:23
OCM	0.218	15	0.248	16	0:05:20
ST0	0.290	26	0.274	25	0:05:35
ST1	0.293	28	0.265	21	0:05:23
LMT	0.176	9	0.198	10	0:01:13
CAL	0.382	31	0.303	30	0:48:26
T1	0.477	33	0.429	32	0:00:43
Statistical					
LDA	0.178	10.5	0.184	7.5	0:00:04
QDA	0.179	12.5	0.237	13	0:00:07
NN	0.396	32	0.446	33	0:00:17
LOG	0.154	2	0.163	3.5	0:01:43
FM1	0.165	6	0.166	5	0:11:16
FM2	0.180	14	0.196	9	3:14:15
PDA	0.178	10.5	0.184	7.5	0:00:35
MDA	0.163	5	0.160	1	0:00:54
POL	0.169	7	0.173	6	1:19:30
Neural Network					
LVQ	0.170	8	0.163	3.5	0:00:35
RBF (5 hidden units)	0.151	1	0.392	31	4:17:58

Table 25: Teaching assistant evaluation (`tae`, 3 classes, 1 numerical attribute, 4 categorical attributes, 151 observations). Error rates are estimated using 10-fold cross-validation experiment. Plurality rule error rate is 0.656. Error rates greater than that of the plurality rule are printed in *italics*. The noise attributes are 5 independent standard normal variates.

	Original		+ Noise		CPU Time
Methods	Error Rate	Rank	Error Rate	Rank	(hh:mm:ss)
Tree & Rules					
QU0	0.470	14.5	0.517	15	0:00:24
QU1	0.484	16	0.510	12.5	0:00:25
QL0	0.430	8	0.411	1	0:03:46
QL1	0.437	9	0.431	2	0:03:45
FTU	0.538	22.5	0.555	22	0:00:06
FTL	*0.693*	33	*0.681*	31	0:00:06
C4T	0.503	19	0.502	10	0:00:08
C4R	0.583	26	0.529	17	0:00:18
IB	0.373	4	0.451	5	0:00:31
IBO	0.325	1	0.470	8	0:06:10
IM	0.538	22.5	0.530	18	0:00:31
IMO	0.492	18	0.549	21	0:40:21
IC0	0.372	3	0.543	20	0:00:41
IC1	0.537	21	0.582	25	0:00:38
OCU	0.451	12	0.650	29	0:00:34
OCL	0.590	27	0.596	26	0:00:59
OCM	0.418	7	0.563	23	0:01:29
ST0	*0.669*	30.5	*0.682*	32	2:19:50
ST1	*0.675*	32	*0.696*	33	2:18:38
LMT	0.470	14.5	0.573	24	0:06:53
CAL	0.439	10	0.510	12.5	0:35:50
T1	0.540	24	0.536	19	0:00:26
Statistical					
LDA	0.411	6	0.450	4	0:00:12
QDA	0.543	25	0.511	14	0:00:18
NN	0.349	2	0.460	7	0:00:17
LOG	0.450	11	0.458	6	0:02:26
FM1	0.636	29	0.628	27	0:45:56
FM2	*0.669*	30.5	0.642	28	4:23:09
PDA	0.490	17	0.478	9	0:00:56
MDA	0.404	5	0.445	3	0:02:25
POL	0.530	20	0.518	16	0:16:20
Neural Network					
LVQ	0.628	28	*0.669*	30	0:00:18
RBF (5 hidden units)	0.464	13	0.504	11	10:13:24