13

A Comparative Study of Classification Algorithms: Statistical, Machine Learning and Neural Network

R. D. King

R. Henery

Department of Statistics and Modelling Science,
University of Strathclyde, Glasgow.

C. Feng

Turing Institute, Glasgow.

A. Sutherland

Department of Statistics and Modelling Science,
University of Strathclyde, Glasgow.

Abstract

The aim of the StatLog project is to compare the performance of statistical, machine learning, and neural network algorithms, on large real world problems. This paper describes the completed work on classification in the StatLog project. Classification is here defined to be the problem, given a set of multivariate data with assigned classes, of estimating the probability from a set of attributes describing a new example sampled from the same source that it has a pre-defined class. We gathered together a representative collection of algorithms from statistics (Naive Bayes, K-nearest Neighbour, Kernel density, Linear discriminant, Quadratic discriminant, Logistic regression, Projection pursuit, Bayesian networks), machine learning (CART, C4.5, NewID, AC2, CAL5, CN2, ITrule – only propositional symbolic algorithms were considered), and neural networks (Backpropagation, Radial basis functions, Kohonen). We then applied these algorithms to eight large real world classification problems: four

from image analysis, two from medicine, and one each from engineering and finance. Our results are still provisional, but we can draw a number of tentative conclusions about the applicability of particular algorithms to particular database types. For example: we found that K-nearest Neighbour can perform well on complex image analysis problems if the attributes are properly scaled, but it is very slow; machine learning algorithms are very fast and robust to non-Normal features of databases, but may be out-performed if particular distribution assumptions hold. We additionally found that many classification algorithms need to be extended to deal better with cost functions (problems where the classes have an ordered relationship are a special case of this).

1 INTRODUCTION

StatLog is an ESPRIT project with ten academic and industrial partners (Appendix A). Its aim is to evaluate the performance of Statistical, Machine Learning, and Neural Network Algorithms on large-scale, complex commercial and industrial problems. The problems are in the areas of classification, forecasting, control, and unsupervized learning. The objectives of the project are threefold:

1. to provide critical performance measurements, and criteria for measurement on available Learning Algorithms which improve confidence for full exploitation;
2. to indicate the nature and scope of the next-stage development which particular algorithms require to meet commercial performance expectations;
3. to indicate the most promising avenues of development for the commercially immature approaches.

This chapter describes the completed work on classification in the StatLog project. Classification is here defined to be the problem, given a set of multivariate data with assigned classes, of estimating the probability from a set of attributes describing a new example sampled from the same source that it has a pre-defined class (this problem is often known as discrimination in statistics, and supervised learning in machine learning - it

fused with clustering which is also sometimes termed classification). We gathered together a representative collection of algorithms from statistics (Naive Bayes, K-nearest Neighbour, Kernel density, Linear discriminant, Quadratic discriminant, Logistic regression, Projection pursuit, Bayesian networks), machine learning (CART, C4.5, NewID, AC2, CAL5, CN2, Itrule – only propositional symbolic algorithms were considered), and neural networks (Backpropagation, Radial basis functions, Kohonen). We then applied these algorithms to eight large real world classification problems: four from image analysis, two from medicine, and one each from engineering and finance. This basic methodology can be thought of as a table: with the algorithms along one axis, the datasets along the other, and a performance measure at each position in the matrix. The objective measures of performance we use include processing time (for training and test data), and error rate (or cost if there is a cost function available). Subjective measures are much more difficult to use and we have done relatively little work on them, but they include: understandability of the decision rule, ease of use of the algorithm (particularly as perceived by a naive user), and robustness to required parameter input. Some results have already been presented by Henery and Taylor 1992, Sutherland *et al.* 1992. Tables of results obtained are at the end of this chapter.

1.1 Previous Comparative Studies

Several authors have recently compared the performance of neural algorithms to other machine learning methods such as ID3 (Quinlan 1986). Some of the tests indicated that neural algorithms worked better than other methods. Other tests have shown that neural algorithms performed worse.

Particular methods may do well in some domains, but not in others. For example, k-nearest neighbour methods usually do fairly well in handwritten character recognition, although backpropagation and/or radial basis function methods may be preferred for reasons of speed and memory.

Fisher and McKusick (1989), for example, compared ID3 with backpropagation (a neural net method) on two natural domains and found that backpropagation was a few percentage

points more accurate. Shavlik *et al.* (1991) compared ID3 with backpropagation on five natural domains. The performance of both systems was rather similar, but on some datasets backpropagation worked better.

Weiss, Galen, and Tadepalli (1987) compared the PVM algorithm (that produces classification rules) with backpropagation. In this series of tests, the symbolic method performed better in three out of four domains. Weiss and Kapouleas (1989) showed that the CART system of Breiman *et al.* (1984) (similar to ID3) usually worked better than backpropagation. These comparative studies are further described in Weiss and Kulikowski (1991).

Quinlan (1990) compared the neural network approach as reported in Hinton with FOIL, a system that is capable of learning relational descriptions. Quinlan showed that FOIL can learn the given task as well as Hinton's neural network.

Difficulties in interpreting recent comparative studies arise from a number of problems, as the following extract from the StatLog Technical Annex makes clear:

- They have not always compared like with like; some methods are based on assumptions about the domain that can give the method an unfair advantage.
- Some learning methods are not complete, and require parameters to be 'tweaked' to tune the system to a particular domain. Sammut (1988) also reported that this tweaking was considerably important to achieve reasonable performance with some algorithms. Parameter tweaking, common, for instance, with neural net software, needs to be taken into account in the comparative evaluation.
- Some comparative studies use variant but not identical data sets and algorithms. A related problem is that some researchers preprocess their data sets in a manner that prevents direct comparison of results, for instance by treating unknown or unspecified values in some manner, or by partitioning a real-valued attribute into discrete attributes.
- When comparing different learning methods, the studies need to be carried out by experts both in the differ-

ent methods and in the problems tackled. Three studies were presented at the International Joint Conference on A.I. '89 that compared pattern recognition, neural networks and AI machine learning techniques (Fisher and McKusick 1989, Mooney *et al.* 1989, Weiss *et al.* 1989). Questions raised at this conference indicated that there was widespread concern that these comparisons were unfair. Two of the studies used decision-tree induction methods that were discarded by the applied statistics community in the early seventies. All three used a neural net method (back-propagation) that is several years out of date in a rapidly developing field. One study used a Bayesian classifier that would not have been applied by a trained Bayesian to the problems concerned. Because relative novices may well misuse techniques or apply outdated techniques, it is important that comparative trials be advised by experts in the areas concerned.

- Very frequently, studies use simulated data. These have the advantage of investigating the behaviour of algorithms under known conditions. For example, Cherkaoui and Cléroux (1991) investigated the performance of six procedures applied to data with a mixture of binary, nominal, ordinal, and continuous attributes.

2 TESTING METHODOLOGY

To ensure the fairness of comparison, a number of measures were taken in StatLog to reduce bias towards one category of algorithm or another.

- Firstly, all the data sets were collected at one centre for pre-processing. Each data set was issued to all testing sites at the same time with the same format and pre-processing. This was designed to eliminate the possible pre-knowledge the users have about the data sets.
- Part of the data was kept back from the testing sites in case the results were disputed. Missing values were replaced by a constant method.

- It was aimed for all algorithms to be tested by experts. For example, many of the statistical algorithms were tested at the Department of Statistics in Strathclyde University, and many machine learning algorithms were tested in The Turing Institute Ltd. Many results were validated by another naive partner. The validating sites were supplied with and used the *log* files from the testing sites, which kept records of the procedure followed.

3 STATLOG ALGORITHMS

This section describes a collection of algorithms that reflect the state of the art in statistical and logical learning. In the context of classification problems, they divide into three broad groups:

3.1 Statistical Algorithms

3.1.1 *Naive Bayes algorithm (Bayes)*

This algorithm directly applies the Bayes rule: $P(c|e) = P(e|c) \cdot P(c)/P(e)$, where c is the class and e is a given example. The aim is to obtain the most probable class given the data. This is c_i, if $P(e|c_i)P(c_i) > P(e|c_j)P(c_j)$ for all j $(i \neq j)$. Because the Bayes method requires complete and accurate probability data, for real problems it is not directly applicable. Some simplifying assumptions are made, i.e. all attributes are independent conditional on the classes, which entails $P(e|c) = P(a_1|c) \cdot P(a_2|c) \cdot ... \cdot P(a_n|c)$, where a_i $(i = 1, ..., n)$ are the n attributes. Despite the unrealistic nature of this assumption, it is found to perform well in many simple tasks in practice. In principle, it is possible to include prior information into the Bayesian analysis, but this is rarely done in reality. Naive Bayes is very simple to apply, it can cope with missing values and often it can produce reasonable results even when its assumptions are violated.

3.1.2 *K-nearest neighbour (K-N-N)*

This is a very simple algorithm, it assigns each new object to the class of the majority of its k nearest neighbours in attribute space (normally Euclidean).

3.1.3 *Kernel density estimation (ALLOC80)*

The program performs multigroup discriminant analysis; within each group the variability is modelled using a non-parametric density estimator, based on kernel functions. Suppose that we have to estimate the $p-$dimensional density function $f(x)$ of an unknown distribution. Information about f is given by n independent observations from this distribution, i.e. $Y_i = (Y_{i1}, \ldots, Y_{im}, \ldots, Y_{ip})$ with $i = 1, 2, \ldots, n$. Let $K^{(p)}(X; Y_i, \lambda)$ be a kernel function centred at Y_i and let λ denote the window width of the kernel. The estimate of $f(x)$ is given by

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K^{(p)}(X; Y_i, \lambda)$$

Observations in the test data are then allocated to classes based on a calculation of the posterior odds by a standard Bayesian calculation. The smoothness of the kernel density estimate is determined in a data-based manner by a pseudo maximum likelihood method. The program can handle continuous as well as mixed (discrete) data. This program is computationally expensive, both in storage and CPU terms. The methods to choose the smoothing parameter automatically are not always successful, and the version in StatLog used is rather unwieldy . However, it has performed consistently well in the trials. It is expected to do better than standard methods where the data are highly non-Normal. For more bizarre datasets, such as two interlocking spirals, this method performs very well. In general, any situation where the boundaries between classes is not easily modelled by a straight line or quadratic may lend themselves well to this approach. The main difficulty, as with most nonparametric density estimators, is to ensure a good choice of smoothing parameter.

3.1.4 *Linear discriminants (Discrim)*

This is an implementation in Splus of Fisher's linear discriminant analysis (1936). The algorithm calculates a linear combination of the attribute values for each class and assigns a new observation to the class with the largest value. The algorithm is optimal when the data are multivariate normal with a common

covariance matrix. The boundaries between classes are hyperplanes in attribute space. This algorithm like Quadra and LogReg were implemented in Splus/Fortran by the Department of Statistics and Modelling Science, University of Strathclyde, Glasgow, Scotland.

3.1.5 *Quadratic discriminants (Quadra)*

This a variant of the above linear algorithm for the unequal covariance case. The algorithm calculates a quadratic combination of the attribute values for each class and assigns a new observation to the class with the largest value. The boundaries between classes are conic sections in attribute space. The algorithm requires to estimate many more co-efficients than linear discriminants and so will only perform well when the training set is sufficiently large (and close to multivariate normal).

3.1.6 *Logistic regression (LogReg)*

Discriminant algorithms above cannot cope with combinations of continuous, categorical and qualitative attributes. So models of generalised linear models (GLIM) are proposed, which includes the logistic class (Cox 1966, Day and Kerridge 1967). In the logistic class of generalized linear models, the probability of an example e given the class of $c_e = k$ $(k = 1, ..., m)$ can be calculated from the relative probability $R(e|c_e = k)$ to a fixed class (the last one, say) which is a logistic function of the parametric linear combination of attributes:

$$P(e|c_e = k) = \frac{R(e|c_e = k)}{\sum_{i=1}^{m} R(e|c_e = i)},$$

$$R(e|c_e = k) = \frac{P(e|c_e = k)}{P(e|c_e = m)} = e^{-(\alpha_{k1}a_1^e + ... + \alpha_{kn}a_n^e)},$$

where a_i^e $(i = 1, ..., n)$ are the attribute values of the example e and $R(e|c_e = m) = 1$. The coefficients α_{kj} $(k = 1, ..., m, j = 1, ..., n)$ are estimated for each class and it must maximize the total likelihood:

$p(\{e|e \text{ is in the sample}\})$. Assume that the sample is randomly chosen so the examples are independent:

$$p(\{e|e \text{ in sample}\}) = \prod_{\{e|c_e=1\}} p(e|c_e = 1) \prod_{\{e|c_e=2\}} p(e|c_e = 2)\ldots$$

$p(e|c_e = k)$ has close links with the binomial distribution, thus its arguments can be categorical.

The logistic regression method also produces a linear separation of classes although it appears to be different from other linear discriminants. It is identical, in theory, to Discrim for many restricted (e.g. normal or binomial) distributions with equal covariances. In fact, logistic regression may start the search for actual coefficients from the coefficients estimated by Discrim. So, the only differences between the two are in the way that the coefficients $\alpha_{kj}(k = 1,...,m, j = 1,...,n)$ (the parameters for the separation hyperplanes) are estimated. Fisher's linear discriminants optimize a quadratic cost function whereas logistic regression optimises on the total likelihood. There may well be occasions when a quadratic cost function is appropriate, in which case Fisher's linear discriminants are justified without appealing to the assumption of multivariate normality.

The logistic model can also be extended to include prior probability, but the number of parameters required in the function grows exponentially as the complexity of the model increases. In the training phase it is considerably more expensive computationally than linear discriminants, although in the testing phase the two methods are indistinguishable. While the model underlying logistic regression is more general, its success depends more critically on the correctness of the underlying assumptions.

3.1.7 *Projection pursuit (SMART)*

SMART (Smooth Multiple Additive Regression Technique) is a collection of FORTRAN subroutines written by Friedman. It is a generalization of projection pursuit regression PPR (Friedman and Stutzle 1981). The regression models take the form

$$E[Y_i|x_1, x_2, \ldots, x_p] = \overline{Y}_i + \sum_{m=1}^{M} \beta_{im} f_m(\sum_{j=1}^{p} \alpha_{jm} x_j)$$

with $\overline{Y}_i = EY_i$, $Ef_m = 0$, $Ef_m^2 = 1$ and $\sum_{j=1}^{p} \alpha_{jm}^2 = 1$. The coefficients β_{im}, α_{jm} and the functions f_m are parameters of the model and are estimated by least squares. The criterion

$$L_2 = \sum_{i=1}^{q} E[Y_i - \overline{Y}_i - \sum_{m=1}^{M} \beta_{im} f_m(\alpha_m^T x)]^2$$

is minimised wrt to the parameters β_{im}, $\alpha_m^T = (\alpha_{1m}, \ldots, \alpha_{pm})$ and the functions f_m.

Classification is closely related. The objective here is to minimise the misclassification risk

$$R = E[\min_{1 \le j \le q} \sum_{i=1}^{q} l_{ij} p(i|x_1, x_2, \ldots, x_p)]$$

where l_{ij} is the user specified loss for predicting $Y = c_j$ when its true value is c_i ($l_{ii} = 0$). The conditional probability is reformulated using a conditional expectation which is then modelled by the regression model above.

This algorithm would be expected to perform very well whenever a cost matrix is applicable, because it (unusually) uses the cost matrix in the **training** phase as well as in the classification stage. Although the training time is not competitive, the algorithm does generally produce good misclassification rates.

3.1.8 *Bayesian networks (CASTLE)*

CASTLE (CAusal STructures From Inductive Learning (Acid *et al.* 1991) is an implementation of the polytree algorithm defined by Pearl (1988). Causal networks are directed acyclic graphs (DAGs) in which the nodes represent propositions (or variables), the arcs signify the existence of direct causal dependencies between the linked propositions, and the strengths of these dependencies are quantified by conditional probabilities.

The structure of a causal network can be determined in the following way: each variable in the domain is identified with a node in the graph. We then draw arrows to each node X_i from a set of nodes $C(X_i)$ considered as direct causes of X_i. The strengths of these direct influences are quantified by assigning to each variable X_i a matrix $P(X_i|C(X_i))$ of conditional probabilities of the events $X_i = x_i$ given any combination of values

of the parent set $C(X_i)$. The conjunction of these local probabilities defines a consistent global model, i.e., a joint probability distribution. Once the network is constructed it constitutes an efficient device to perform probabilistic inferences. The problem of building such a network remains. The structure and conditional probabilities necessary for characterizing the network could be provided either externally by experts or from direct empirical observations.

Under the Bayesian approach, the learning task in causal networks separates into two highly related subtasks, *structure learning*, that is, to identify the topology of the network, and *parameter learning*, the numerical parameters (conditional probabilities) for a given network topology.

CASTLE, currently being developed by members of the Department of Computer Science and Artificial Intelligence at the University of Granada, focuses on learning a particular kind of causal structure: polytrees (singly connected networks), networks where no more than one path exists between any two nodes. As a consequence, a polytree with n nodes has no more than $n-1$ links. It is in polytrees (and specially in trees) where the ability of networks to decompose and modularize the knowledge attains its ultimate realization. Polytrees do not contain loops, that is, undirected cycles in the underlying network (the network without the arrows or skeleton), and this fact allows a locally efficient propagation procedure (Pearl 1988).

3.2 Machine Learning

From the field of machine learning only propositional symbolic algorithms were considered by StatLog. No algorithms from the emerging field of Inductive Logic Programming (ILP) were included, nor were Genetic algorithms examined.

3.2.1 *CART*

CART, *Classification and Regression Tree*, is a binary decision tree algorithm. The acronym CART comes from *Classification And Regression Tree* (Breiman *et al.* 1984). CART is not really a single algorithm, but a collection of algorithms and analysis methods for classification and regression trees (or Discrimina-

tion And Forecasting Trees). The algorithm described in this section is the most commonly used version. CART is a binary decision tree algorithm A binary decision tree consists of nodes and each node has two branches. There is a single test (or decision) on each node, splitting the node into two subtrees. Depending on whether the result of a test is true or false, the tree will branch to left or right, at which this splitting process recursively continues. At each leaf node a decision is made on the class assignment.

The advantages of using a decision tree methodology are that it is non-parametric and produces classifications which can be easily understood. This latter feature has made them popular in machine learning.

The fundamental idea in CART's tree construction is to select each split so that the data in each of the descendant subsets are 'purer' than the data in the parent subset. The splitting evaluation function developed for CART is different from that used in the ID3 family of decision tree algorithms. Consider the case of a problem with two classes, and a node has 50 examples from each class, the node has maximum impurity. If a split could be found that divides the data into one subgroup of 40:5 and another of 10:45, then intuitively the impurity has been reduced. The impurity would be completely removed if a split could be found that produced sub-groups 50:0 and 0:50. In CART this intuitive idea of impurity is formalized in the *GINI* index for the current node c:

$$Gini(c) = 1 - \sum_j p_j^2$$

where p_j is the probability of class j in c. For each possible split the impurity of the subgroups is summed and the split with the minimum impurity chosen.

The GINI criteria is local: there is no guarantee that the overall tree is optimal. As a result, decision trees produced can potentially contain many nodes. This may result in the number of examples available to test on becoming very small for some branches. The original CART authors explored related impurity measures such as entropy.

In CART each split depends only on a single attribute. For ordered and numeric attributes, CART considers all possible splits in the sequence. For n values of the attribute, there are n splits. For categorical attributes CART examines all possible binary splits, which is similar to the attribute subsetting method used for C4.5. At each node CART searches through the attributes one by one. For each attribute it finds the best split. Then it compares the best single splits and selects the best attribute of the best splits.

CART uses a sophisticated form of pruning to try and avoid over-fitting the data. If no pruning is used then the trees generated by CART will be too specialized and biased towards the training data. CART uses cost-complexity pruning to decide the order of branches to prune, and cross-validation to decide on the pruning parameters.

Two versions of CART were investigated: the commercial version of CART (used by the University of Granada), and INDCART (a free version of CART supplied by Wray Buntine at NASA Ames, who also supplied the Naive Bayes program). INDCART differs from the standard version of CART (Breiman *et al.* 1984) by using a different (probably better) way of handling missing values, in a different default setting for pruning, not fully using costs, and in not implementing the regression part of CART.

3.2.2 *NewID*

NewID from the Turing Institute Ltd in Glasgow, Scotland, is a direct descendant of the decision tree algorithm ID3 (Quinlan 1986). It differs from the original ID3 in being designed to cope with continuous variables and noise. Its basic structure is very similar to that of CART. It differs from CART in using a different splitting criterion and a different pruning method. NewID uses entropy gain instead of GINI. The entropy of the example set at a node is $-\sum_j p_j \log p_j$, where p_j is the probability estimate of the jth class in that set. The entropy *gain* is the difference of entropy between the current set and the subsets created by the split. The attribute with the highest gain is selected, giving the most informative split at that node.

3.2.3 *C4.5*

C4.5 (Quinlan 1987) is also a direct descendant of the decision tree algorithm ID3 (Quinlan 1986). It also is designed to cope with continous variables and noise. It differs from NewID in using a slightly different splitting criterion and pruning method. C4.5 uses the entropy gain ratio as a splitting criteria. This takes the information provided by the attribute a into account. The attribute to split on should maximise the information gain relative to the information needed to determine the attribute value, i.e. the ratio:

$$gain\_ratio(c, a) = \frac{gain(c, a)}{I(a)} = \frac{I(c \wedge a) - I(c)}{I(a)}.$$

3.2.4 *AC2*

AC2 from Isoft in Paris is an extension of ID3 to deal with hierarchical data. It can learn structures from a predefined hierarchy of attributes. A hierarchy may be imposed to create trees that are more meaningful to the end-user: there is the additional advantage of savings in learning and testing time since certain tests may be ruled out by the hierarchical structure.

3.2.5 *Cal5*

Cal5 is a decision tree algorithm based on statistical methods and designed for continuous variables. It is contributed by the Institute of Automation, Berlin (Unger and Wysotzki 1981). Interestingly, it was developed by the Institute of Automation in East Berlin independently of the work on decision trees in the West. In Cal5 trees are constructed sequentially starting with one attribute and branching with other attributes recursively, if no sufficient discrimination of classes can be achieved. That is, if at a node no decision for a class c_i according to the above formula can be made, a branch formed with a new attribute is appended. If this attribute is continuous, a discretization, i.e. intervals corresponding to qualitative values, has to be used.

Let N be a certain non-leaf node in the tree construction process. At first the attribute with the best local discrimination measure at this node has to be determined. For that a method

working without any knowledge about the result of the desired discretization is used. For continuous attributes the quotient

$$quotient(N) = \frac{A^2}{A^2 + D^2}$$

is a discrimination measure for a single attribute, where A is the standard deviation of examples in N from the centroid of the attribute value and D is the mean value of the square of distances between the classes. This measure has to be computed for each attribute. The attribute with the least value of $quotient(N)$ is chosen as the best one for splitting at this node. Note that at each current node N all available attributes $a_1, a_2, \ldots, a_n$ will be considered again. If a_i is selected and occurs already in the path to N, than the discretization procedure leads to a refinement of an already existing interval.

3.2.6 *CN2*

CN2 is a decision rule algorithm (Clark and Niblett 1988, Clark and Boswell 1991). It learns decision rules for each class in turn. Initially it starts with a 'universal rule': '**If** all conditions **Then** Current class'. This rule ought to cover at least one of the examples in the current class. Specializations of this rule are then repeatedly generated and explored until a rule has been found. This rule ought to cover examples of the 'right' mixture belonging to the current class and other classes. The mixture is usually determined by heuristics or is user-specified. Intuitively, as few as possible *negative* examples, i.e. examples in other classes, should be covered. Each specialization is obtained by adding a condition to the left-hand side of the rule, i.e. requiring one particular attribute to have value within a range. Similar to decision tree algorithms such algorithms are better suited to deal with logical attributes and classes.

CN2 is an extension of an earlier algorithm AQ (Michalski 1983) that can deal with noise in data. It can also accept continuous numeric values in attributes though not classes. The main technique for reducing error is use of Laplace's Law of Succession. If there are $m1, m2, ..., mk$ examples of classes $1, 2, .., k$, where the total no. of examples is n, then the probability that

a new data item will fall into class i is:

$$(m_i + 1)/(n + k).$$

3.2.7 *ITrule*

ITrule (Goodman and Smyth 1989) from CalTech University produces rules of the form **'If ... Then ... with probability ...'**. This algorithm contains probabilistic inference in its evaluation function of rule candidates through the J-measure and varies from AQ and CN2 in its method of constructing rules by incorporating generalization as well (i.e. dropping conditions). The J measure is a product of conditional prior probabilities and the cross-entropy of class values conditional on the attributes values. ITrule cannot deal with continuous numeric values.

3.3 Neural Networks

3.3.1 *Back-propagation Multi-Layer Perceptron (Backprop)*

Back-propagation Multi-Layer Perceptron is a neural network algorithm (McClelland *et al.* 1986) which consists of a network of 'neurons' arranged in a number of layers, where each neuron is connected to every neuron in the adjacent layers. Each neuron sends a signal along the connections to the neurons in the layer above. The signals are multiplied by weights corresponding to each connection.

We used a version with three layers: an input layer, a hidden layer and an output layer. *strictly layered, 3-layer MLP*, which is the mapping

$$y_i^{(H)} = f^{(H)}\left(\sum_j w_{ij}^{(HI)} y_j^{(I)}\right)$$

$$y_i^{(T)} = f^{(T)}\left(\sum_j w_{ij}^{(TH)} y_j^{(H)}\right)$$

from the inputs $y^{(I)}$ to the targets $y^{(T)}$, via the *hidden nodes* $y^{(H)}$. The parameters are the *weights* $w^{(HI)}$ and $w^{(TH)}$. The univariate functions $f^{(\cdot)}$ are usually each set to be logistic which varies smoothly from 0 at $-\infty$ to 1 at ∞.

This is also called a '2-layer' MLP by authors who prefer to count layers of weights rather than layers of nodes. It can be shown that, given enough hidden nodes, this can approximate any mapping to an arbitrary accuracy.

This mapping has a biological interpretation: Node (model neuron) i produces a strong output if its *activation potential* $\sum_j w_{ij} y_j$ is positive, and a weak output if it is negative. The activation is increased if node i is connected to an active node j via an *excitatory* synapse ($w_{ij} > 0$), and is decreased by active nodes connected via *inhibitory* synapses ($w_{ij} < 0$).

One input to each layer, say node 0, is traditionally assigned the constant value 1.0 so that w_{i0} provides a constant offset or *bias* for the activation of i. This device allows the *threshold* activation (due to the remaining nodes), at which $y_i = 0.5$, to be adjusted away from 0.

The code was written by R. Rohwer whilst at the Centre for Speech Technology Research, Edinburgh, Scotland.

3.3.2 *Radial basis functions (Radial)*

Radial basis function methods are closely related to the kernel estimator methods discussed under nonparametric discriminant analysis Poggio and Girosi, 1990). The radial basis function network mapping is given by

$$y_i^{(H)} = f^{(H)}\left(\frac{\sqrt{\sum_j (y_j^{(I)} - c_{ij})^2}}{r_i}\right)$$

$$y_i^{(T)} = \sum_j w_{ij}^{(TH)} y_j^{(H)}$$

It has a linear output layer like an MLP with such an option, but the hidden layer is different. Hidden node i computes a function of the Euclidian distance of an input from its *centre* $c_{i\cdot}$, on a scale determined by its *radius* r_i. Usually the chosen function is the Gaussian.

As in the MLP, a bias node is introduced into the linear layer. Sometimes non-Euclidian distance measures are used. The loosely-defined region for which a radial basis function has a significant output is its *receptive field.* The functions computed at the hidden nodes are the 'radial basis functions' *per*

se. They are 'radial' in that their receptive fields are spherically symmetric; there is an r_i for each centre i but not an r_{ij} for each centre and input coordinate. Such a generalization is often made, however, and if it is not, then it is desirable to prescale the input data to give it equal variance in each dimension. The code was supplied by Richard Rohwer then at the Department of Statistics and Modelling Science, Strathclyde University, Scotland.

3.3.3 *Kohonen net (Kohonen)*

Kohonen net is a self-organizing mapping algorithm (Kohonen 1989). Our implementation comes from J. Paul, Institut fuer Kybernetick und Systemtheorie, Am Hlsenbusch 54, W-4630 Bochum 1, Germany. It is capable of learning a mapping between an input space and an output space by establishing a topology-conserving map on a usually planar array of 'neuronal' nodes. A feature map is a data structure where the interrelationships of the data are captured in the spatial arrangement of the corresponding nodes. The map defines the ordering of the nodes allowing the algorithm to freely develop within this structure.

4 STATLOG DATASETS

The datasets studied by StatLog were all the large classification datasets of commercial and industrial interest that could be found.

4.1 Satellite Image (Satellite)

The database consists of the multi-spectral values of pixels in 3×3 overlapping neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values. In the sample database, the class of a pixel is coded as a number.

This sample database from LandSat Multi-Spectral Scanner image data was provided by: Ashwin Srinivasan, Department of Statistics and Modelling Science, University of Strathclyde. The original LandSat data for this database was generated from

data purchased from NASA by the Australian Centre for Remote Sensing, and used for research at The Centre for Remote Sensing, University of New South Wales.

The sample database was generated taking a small section (82 rows and 100 columns) from the original data. The binary values were converted to their present ASCII form by Ashwin Srinivasan. Each line of data corresponds to a 3×3 square neighbourhood of pixels completely contained within the 82×100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the nine pixels in the 3×3 neighbourhood and a number indicating the classification label of the central pixel. The number is a code for the following classes:

The data has 36 numerical attributes and six classes. The data was divided into a training set with 4435 examples and a test set with 2000 examples. A one-shot train-and-test was used to calculate the accuracy.

4.2 Handwritten Digits (Digits)

The purpose of the hand-written digit dataset is to classify 4×4 pixel images as the digits 0–9. Each member of a set of 18 000 handwritten digits was digitized onto a 16×16 pixel array with greylevels 0(white)-255(black). The pixel values were then averaged over 4×4 neighbourhoods to produce the 4×4 images. Each line of the dataset consists of the 16 pixel-values read out from left-to-right and top-to-bottom across the image, followed by the value of the digit appropriate to that image. The dataset has been divided into equal test and training sets with 9000 examples in each set. There are 900 examples of each digit in either set. The digits were gathered from postcodes on letters passing through the German Federal Post. A very small number of mistaken images have been allowed to appear, e.g. one of the ones is actually an '!' and one of the eights is actually a capital letter 'B'.

This dataset has already been studied by Kressel *et al* (1990), and Kressel (1991) of AEG, Ulm, who have published a comparative study of backpropagation and their own 'polynomial

classifier'. They used the full 256 attribute version and achieved results of the order of 98% accuracy.

The data has 16 numerical attributes and 10 classes. The data was equally divided into a training set with 9000 examples and a test set with 9000 examples. A one-shot train-and-test was used to calculate the error rate.

4.3 Karhunen–Loeve Digits (KL)

The Karhunen–Loeve (kl) digits dataset is very closely related to the other handwritten digits dataset. Whereas the other dataset was produced by averaging over 4×4 neighbourhoods in the original 16×16 images, the kl dataset is produced by a linear transformation of the original 16×16 images. The eigenvectors of the covariance matrix of the original 16×16 images were computed. The scalar products of the top 40 eigenvectors with the original images were calculated. It was found that the original images could be reconstructed from these 40 eigenvectors with minimal loss of information. Therefore, the original 256 attributes had been compressed down to 40. It is these 40 scalar products which are the attributes of the kl dataset. In statistical terminology, the 256 attributes were replaced by the first 40 principal components.

The data has 40 numerical attributes and 10 classes. The data was equally divided into a training set with 9000 examples and a test set with 9000 examples. A one-shot train-and-test was used to calculate the error rate.

4.4 Vehicle Silhouettes (Vehicle)

The purpose of this dataset 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.

This data was originally gathered at the Turing Institute in 1986-87 by J.P. Siebert (1987). It was partially financed by Barr and Stroud Ltd. The original purpose was to find a method of distinguishing 3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects. Measures of shape features extracted from example

silhouettes of objects to be discriminated were used to generate a classification rule tree by means of computer induction. This object recognition strategy was successfully used to discriminate between silhouettes of model cars, vans, and buses viewed from constrained elevation but all angles of rotation.

The features were extracted from the silhouettes by the HIPS (Hierarchical Image Processing System) extension BINATTS, which extracts a combination of scale independent features utilizing both classical moments based measures such as scaled variance, skewness, and kurtosis about the major/minor axes and heuristic measures such as hollows, circularity, rectangularity, and compactness. Four 'Corgi' model vehicles were used for the experiment: a double decker bus, Chevrolet van, Saab 9000, and an Opel Manta 400. This particular combination of vehicles was chosen with the expectation that the bus, van and either one of the cars would be readily distinguishable, but it would be more difficult to distinguish between the cars. The attributes were all real and the range for each attribute was different.

The data has 18 numerical attributes and four classes. There are 846 examples and nine-fold cross-validation was used to estimate the error rate.

4.5 Head Injury (Head)

The data set is a series of 1000 patients with severe head injury collected prospectively by neurosurgeons between 1968 and 1976. This head injury study was initiated in the Institute of Neurological Sciences, Glasgow. After four years two Netherlands centres (Rotterdam and Groningen) joined the study, and late data came also from Los Angeles.

The original purpose of the head injury study was to investigate the feasibility of predicting the degree of recovery which individual patients would attain, using data collected shortly after injury. Severely head injured patients require intensive and expensive treatment; even with such care almost half of them die and some survivors remain seriously disabled for life. Clinicians are concerned to recognize which patients have potential for recovery, so as to concentrate their endeavours on them. Outcome was categorized according to the Glasgow Outcome Scale, but

the five categories described therein were reduced to three for the purpose of prediction. Titterington *et al* (1981) compared several discrimination procedures on this dataset. Our dataset differs by replacing all missing values with the class median. All attribute values are integers.

There are five numerical attributes and one binary (categorical), and there are three classes. There are 900 examples in the dataset. Nine-fold cross-validation was used to estimate the average cost.

This is one of the two datasets with a cost matrix. The matrix below gives the different cost of various possible misclassifications (d/v = dead or vegetative, sev = severe disability, and m/g = moderate disability or good recovery).

	d/v	*sev*	*m/g*
d/v	0	10	75
sev	10	0	90
m/g	750	100	0

4.6 Heart Disease (Heart)

This database comes from the Cleveland Clinic Foundation and was supplied by Robert Detrano, M.D., Ph.D. of the V.A. Medical Center, Long Beach, CA. It is part of the collection of databases at the University of California, Irvine collated by David Aha.

The purpose of the dataset is to predict the presence or absence of heart disease given the results of various medical tests carried out on a patient. This database contains 13 attributes, which have been extracted from a larger set of 75. The database originally contained 303 examples but six of these contained missing values and so were discarded leaving 297. 27 of these were left out as a validation set, leaving a final total of 270. There are two classes: presence and absence (of heart-disease). This is a reduction of the number of classes in the original dataset where there were four different degrees of heart-disease.

This data has been studied before, but without taking the cost matrix into account. In an unpublished study Detrano *et al.* got approximately a 77% correct classification accuracy

with a logistic-regression-derived discriminant function. Aha and Kilber (1988) used instance-based prediction and got 77.0% accuracy with NTgrowth and 74.8% with C4. John Gennari got 78.9% with the CLASSIT conceptual clustering system.

There are eight numerical attributes and five binary/categorical ones, there are two classes. There are 270 examples in the dataset. Nine-fold cross-validation was used to estimate the average cost.

This is one of the two datasets with a cost matrix. The matrix below gives the different costs of various possible misclassifications. It was supplied by doctors in Leeds to Dr. C.C. Taylor, Statistics Dept, Leeds University.

	absent	*present*
absent	0	1
present	5	0

4.7 Credit Risk (Credit)

The purpose of this dataset is to evaluate various customers of the credit industry as good or bad credit risks. The dataset was supplied by Attar Software Ltd of Leigh, Lancashire. For commercial reasons the meaning of the attributes is secret. The original dataset had 39 attributes some of which were numerical and others categorical. Some of the categorical attributes had very large numbers of categories.

This original dataset presented structural problems for many of the StatLog algorithms (statistical, machine learning, and neural netwok). For example CART cannot deal with categorical attributes with large numbers of categories.

Therefore a new dataset with 16 attributes was produced by J. Mitchell of Strathclyde University. Many of the less informative attributes were thrown away, others were binarzsed and some of the numeric attributes were log-transformed to make them closer to normal. Missing values were replaced.

The ratio of good to bad customers in the dataset is almost 1:1. This is not representative of the population as a whole, where the ratio is more like 10:1. However, we assumed that the effect of this would be cancelled out by the different costs

of misclassifying a good or bad customer; it is about ten times more costly to misclassify a bad customer as good compared to a good customer as bad.

The data has eight numerical and eight binary/categorical attributes, with two classes. The data was divided into a training set with 6230 examples and a test set with 2670 examples. A one-shot train-and-test was used to calculate the error rate.

4.8 Shuttle control (Shuttle)

The dataset was provided by Jason Catlett who was then at the Basser Department of Computer Science, University of Sydney, N.S.W., Australia. The data originated from NASA and concern the position of radiators within the Space Shuttle. The problem appears to be noise-free in the sense that arbitrarily small error rates are possible given sufficient data.

The data was divided into a train set and a test set with 43500 examples in the train set and 14500 in the test set. A one-shot train-and-test was used to calculate the accuracy. With samples of this size, it should be possible to obtain an accuracy of 99 – 99.9%. Approximately 80% of the data belong to class 1. At the other extreme, there are only six examples of class 6 in the learning set, so no rule could be constructed to have uniform accuracy for all classes.

The data has seven numerical attributes with seven classes. The data was divided into a training set with 43500 examples and 14500 examples test set. A one-shot train-and-test was used to calculate the error rate.

4.9 Characterization of Datasets

An important objective in StatLog is to investigate why certain algorithms do better on some datasets and other algorithms do better on different datasets Table 13.1. Appendix B describes a list of measures which will help to explain our findings. At present these measures are mostly very simple or statistically based, and none of these measures is really appropriate to decision trees as such.

There is a need for a measure which indicates when decision trees will do well. Bearing in mind the success of decision trees

in the Tsetse Fly data described by Ripley (1992), it seems that some measure of multimodality might be useful in this connection.

Some algorithms have built-in measures which are given as part of the output. For example, CASTLE measures the Kullback–Leibler information in a dataset. Such measures are useful in establishing the validity of specific assumptions underlying the algorithm, and although they do not always suggest what to do if the assumptions do not hold, at least they give an indication of internal consistency.

5 RESULTS

The results fall naturally into three groups based on the databases. The first group contains the four image analysis datasets: satellite image, handwritten digits, Karhunen–Loeve digits, and vehicle recognition. The second group includes the two medical datasets (both involving cost matrices): head injury, heart disease. The last group includes the two datasets most difficult to characterize: credit risk, and shuttle.

5.1 Image Analysis

Within the group of image analysis datasets all the attributes are numerical. The attributes for satellite image, and handwritten digits come directly from the images with only minimum processing, i.e. are brightness levels. The attributes for the vehicle dataset were generated using an image analysis package.

5.1.1 *Satellite*

In the satellite dataset K-nearest Neighbour performs best, Table 13.2. Not surprisingly, radial basis functions and Alloc80 also do fairly well as these three algorithms are closely related. Their success suggests that all the attributes are equally scaled and equally important. There appears to be little to choose between any of the other algorithms, except that Naive Bayes does badly (and its close relative CASTLE does relatively badly also). This dataset has the highest correlation between attributes ($corr\_abs = 0.5977$). This may partly explain the failure of Naive Bayes (assumes attributes are conditionally in-

dependent), and CASTLE (confused if several attributes contain equal amounts of information). Note that only three linear discriminants are sufficient to separate all six class means ($fract\_3 = 0.9691$). This may be interpreted as evidence of seriation, with the three classes 'grey soil', 'damp grey soil' and 'very damp grey soil' forming a continuum. Equally, this result can be interpreted as indicating that the original four attributes may be successfully reduced to three with no loss of information. Here 'information' should be interpreted as mean square distance between classes, or equivalently, as the cross entropies of the normal distributions.

5.1.2 *Digits*

The results for the digits dataset and the KL-digits dataset are very similar so are treated together, Table 13.3 and Table 13.4. Most algorithms perform a few percent better on the KL-digits data. The Karhunen–Loeve version of digits is the closest to being normal. This could be predicted beforehand, as it is a linear transformation of the attributes that, by the Central Limit Theorem, would be closer to normal than the original. Because there are very many attributes in each linear combination, the KL-digits dataset is very close to normal ($skewness = 0.1802, kurtosis = 2.9200$) as against the exact normal values of ($skewness = 0, kurtosis = 3.0$).

In both Digits datasets dataset K-nearest Neighbour comes top and Radial basis functions and Alloc80 also do fairly well. These three algorithms are all closely related. Kohonen also does well in the Digits dataset (it has not yet been applied to KL-digits); Kohonen has some similarities with nearest neighbour type algorithms. The success of such algorithms suggests that the attributes are equally scaled and equally important. Quadratic discriminant also does well, coming second in both datasets. The KL version of digits appears to be well suited to quadratic discriminants: there is a substantial difference in variances ($SD\_ratio = 1.9657$), while at the same time the distributions are not too far from multivariate normality with kurtosis of order 3.

Backpropagation does quite well on the Digits dataset. This

might be expected from the literature (McClelland *et al.* 1986). Neural networks are widely considered to do well at character recognition.

5.1.3 *Vehicle*

The attributes for the vehicle dataset, unlike the other image analysis, were generated using image analysis tools and were not simply based on brightness levels. This suggests that the attributes are less likely to be equally scaled and equally important. This is confirmed by the lower performances of K-nearest Neighbour and Radial Basis functions Table 13.5. Alloc80 still does very well and appears to be more robust than the other two algorithms. The original Siebert (1987) paper showed machine learning performing better than K-nearest Neighbour. Quadratic discriminant does best. The high value of $fract\_2 = 0.8189$ might indicate that linear discrimination could be based on just two discriminants. This may relate to the fact that the two cars are not easily distinguishable, so might be treated as one (reducing dimensionality of the mean vectors to 3D). However, although the fraction of discriminating power for the third discriminant is low (1 minus 0.8189), it is still statistically significant, so cannot be discarded without a small loss of discrimination. Backpropagation also does well on this dataset.

5.2 Medical Datasets with Costs

Both medical datasets have cost matrices associated with them. In the results for both datasets the top ten algorithms (algorithms with the lowest costs) are all capable of utilizing costs in the testing phase. SMART performed best on the Head injury dataset, Table 13.6. It is the only algorithm that as standard can utilize costs directly in the training phase (we used in our results a modified version of Backpropagation that could utilise costs, but this is very experimental). Naive Bayes performed best on the heart dataset, Table 13.7. This may reflect the careful selection of attributes by the doctors. Logistic regression does very well and so do Linear and Quadratic discriminants.

It appears that in the head dataset a single linear discrimi-

nant is sufficient to discriminate between the classes (more precisely: a second linear discriminant does not improve discrimination). Therefore the head injury dataset is very close to linearity. This may also be observed from the value of $fract\_1 = 0.9787$, which is very close to unity. In turn, this suggests that the class values reflect some underlying continuum of severity, so this is not a true discrimination problem. Note the similarity with Fisher's original use of discrimination as a means of ordering populations. Perhaps this dataset would best be dealt with by a pure regression technique, either linear or logistic. If so, manova indicates that the middle group is slightly nearer to category 3 than 1, but not significantly nearer. It appears that there is not much difference between the covariance matrices for the three populations in the head dataset ($SD\_ratio = 1.1231$), so the procedure quadratic discrimination is not expected to do much better than linear discrimination.

In the heart dataset the leading correlation coefficient $cancor1 = 0.7384$, this is not very high (bear that in mind it is *correlation* that gives a measure of predictability). Therefore the discriminating power of the linear discriminant is only moderate.

5.3 Other Datasets

5.3.1 *Credit*

In the credit dataset most of the attributes are symbolic. Many of them are also irrelevant; it is possible to get 87% accuracy by using just one attribute (N.B. many algorithms do worse than this), Table 13.8. Most machine learning algorithms agree that this simple rule is indeed the best rule (NewID, AC2, CART, CN2), and this is also the conclusion of CASTLE. Due credit should be given to these procedures for finding a rule that is not only simple but efficient.

On the other hand, some algorithms (SMART and backprop) achieve about the same accuracy but the simple structure of the problem is completely masked. The statistical procedures (discrim, logdiscr and quadisc) also fail to find the simple rule: although there are procedures for dropping attributes that do not contribute usefully to the discrimination, these have not

been implemented.

The poor performance of k-nearest neighbour on this dataset is thought to be due to the presence of symbolic attributes (which are difficult to scale): irrelevant attributes are also known to decrease its performance.

It is interesting to note that the characterization of this dataset is quite similar to that of the heart dataset. This would suggest that it would be an interesting experiment to see how well the algorithms do on the heart dataset without a cost matrix. If the dataset characterization is useful then algorithms that did well on credit should also do well on head without a cost matrix.

5.3.2 *Shuttle*

The shuttle dataset also departs widely from typical distribution assumptions. One important feature of the data is that there is very little 'noise', i.e. it is possible to get arbitrarily close to 100% accuracy Table 13.9. The attributes are numerical and appear to exhibit multimodality (we do not have a good statistical test to measure this).

In this dataset the data seem to consist of isolated islands or clusters of points, each of which is pure (belongs to only one class), with one class comprising several such islands. However, neighbouring islands may be very close and yet come from different populations. The boundaries of the islands seem to be parallel with the coordinate axes. If this picture is correct, and the present data do not contradict it, as it is possible to classify the combined dataset with 100% accuracy using a decision tree, then it is of interest to ask which of our algorithms are *guaranteed* to arrive at the correct classification given an arbitrarily large learning dataset. In the following, we ignore practical matters such as training times, storage requirements etc., and concentrate on the limiting behaviour for an infinitely large training set.

Procedures guaranteed to give the perfect rule for this dataset would seem to be: k-nearest neighbour, Bayes rule, CASTLE, backprop and Alloc80. Radial basis functions should also be capable of perfect accuracy, but some changes would be required

in the particular implementation used in the project (to avoid singularities).

Decision trees will also find the perfect rule provided that the pruning parameter is properly set, but may not do so under all circumstances as it is occasionally necessary to override the splitting criterion (Gordon and Olshen 1978).

The statistical procedures Discrim, Quadisc and LogReg would not improve their accuracy much beyond that given in Table 13.9 (97.1% for Quadisc).

6 CONCLUSIONS

It is not easy to compare the performance of 19 algorithms tested on eight datasets, with three different criteria (accuracy/cost, time), especially as, at this stage in the project, some of the trials are not yet performed. However a number of tentative conclusions can be made.

The most important of these is: that there appears to be no one algorithm that is best for all types of dataset. What algorithm is best depends on features of the dataset. Much work needs to be done, both theoretical and empirical, to determine what features of datasets suit what types of algorithms.

If the distribution assumptions of Linear discriminant are met then this algorithm is provably optimal in terms of maximizing accuracy (not necesarily in terms of speed or human understandability. If the attributes are equally important and equally scaled then nearest neighbour algorithms can do very well; this appears to be true for some image analysis tasks where the attributes have not been overly transformed.

Machine Learning algorithms (symbolic propositional ones) can perform relatively well when the attributes are symbolic or far from assumptions of normality. They appear to be very robust, but may lose efficiency if certain distribution assumptions are met. There is a confusing profusion of Machine Learning algorithms, but they all seem to perform at about the same level.

With care, neural networks perform well on some problems. It is not clear how to characterize these problems. In terms

of computational burden, and the level of expertise required, they are much more complex than, say, the machine learning procedures.

There were great differences in the time taken by the various algorithms. This may be important in some applications. The fastest algorithms were the simplest statistical ones Linear discriminant and Naive Bayes. The machine learning algorithms were also fast (AC2 is an exception, probably because of its complex interface and it is written in LISP). Backpropagation was very slow, but there are however some variations which make improvements to this. The nearest neighbour algorithms were extremely slow to classify new examples, however it is known (Hart 1968) that substantial time saving can be effected, at the expense of some slight loss of accuracy, by using a condensed version of the training data.

7 FURTHER WORK

Much more attention needs to be paid to the use of cost matrices in algorithms. Very few algorithms use costs in their learning phase, and many algorithms do not even use costs in their testing phase (this is easy to implement if a class probability estimate can be made). This use of costs has been virtually ignored by machine learning and neural network workers. Many problems are a combination of discrimination and regression, i.e. the classes are linearly ordered (e.g. Head injury). This problem can be considered as a special case of the use of cost matrices.

It can fairly be said that the performance of linear and quadratic discriminants was exactly as might be predicted on the basis of theory. Several practical problems remain however: (i) the problem of deleting attributes if they do not contribute usefully to the discrimination between classes; (ii) the desirability of transforming the data; and the possibility of including some quadratic terms in the linear discriminant as a compromise between pure linear and quadratic discrimination. Much work needs to be done in this area.

The performance of CASTLE should be related to how 'tree-

like' the dataset is. A major criticism of CASTLE is that there is no internal measure that tells us how closely the empirical data are fitted by the chosen polytree. We recommend that any future implementation of CASTLE incorporates such a 'polytree' measure. Although it would seem that this measure could be based on the Kullback–Leibler information for the fitted polytree, it is not clear exactly how this should be done. The main reason for using CASTLE is that the polytree models the whole structure of the data, and no special role is given to the variable being predicted, viz. the class of the object. However instructive this may be, it is not the principal task in StatLog (which is to produce a classification procedure). So maybe there should be an option in CASTLE to produce a polytree which *classifies* rather than fits all the variables. To emphasize the point, it is easy to deflect the polytree algorithm by making it fit irrelevant bits of the tree (that are strongly related to each other but are irrelevant to classification).

In decision trees there are no indications in our results that any splitting criterion is best, but the case for using some kind of pruning is overwhelming, although, again, our results are too limited to say exactly how much pruning to use (it appears to be some function of the amount of noise in the data). Work needs to be done to relate the performance of a decision tree to some measures of complexity and pruning, specifically the average depth of the tree and the number of terminal nodes (leaves).

One major weakness of neural nets is the lack of diagnostic help. If something goes wrong, it is difficult to pinpoint the difficulty from the mass of inter-related weights and connectivities in the net. For example, in the prediction problems, the two neural nets performed rather badly. This is primarily due to their inability, as black boxes, to react to the several components of time series – trend, seasonality and correlated random effects. Any neural net predictor would need an architecture capable not only of incorporating these features but also of telling the operator when and how to adjust for changes in the features.

It may be possible to combine the best features of algorithms to produce a hybrid system than does better than any one single

system.

Acknowledgments

This work has been supported by the Commission of the European Community under ESPRIT project no. 5170. We thank all the partners in StatLog, in particular Charles Taylor of Leeds University. We would also like to acknowledge the help of G. Nakhaeizadeh of Daimler-Benz, R. Molina of Granada University and J. Stender of Brainware, Germany. We also thank Pavel Brazdil of Porto University for work on software tools that were used in algorithm testing. A special acknowledgement is due to Ashwin Srinivasan of Strathclyde University, who contributed the satellite image database. The heart disease data came from the University of California in Irvine. It was given to us by David Aha of the Applied Physics Laboratory, John Hopkins University. Some algorithms came from Wray Buntine of the NASA Ames laboratory. Finally we would like to thank Donald Michie of the Turing Institute and Brian Ripley of Oxford University for initiating StatLog.

APPENDIX A – STATLOG KEY PERSONNEL: PARTNERS

Dr. Pavel B. Brazdil (Principal Investigator),
University of Porto,
Laboratory of AI and Computer Science (LIACC),
R. Campo Alegro 823,
4100 Porto,
Portugal.

Prof. E. von Goldammer (Principal Investigator),
Institut fuer Kybernetick und Systemtheorie,
Am Hlsenbusch 54,
W-4630 Bochum 1,
Germany.

Dr. R.J. Henery (Technical Director),
Department of Statistics and Modelling Science,
University of Strathclyde,
Glasgow G1 1XH,
UK.

Dr. R. Molina (Principal Investigator),

University of Granada,
Department of Computer Science and AI,
Facultad de Ciencias,
18071 Granada,
Spain.

Dr. G. Nakhaeizadeh (Project Director),
Daimler-Benz AG,
Forschungzentrum Ulm,
Ebeerhard-Finckh Str. 11,
D-7900 Ulm,
Germany.

Mr. H. Hendrix (Principal Investigator),
Isoft,
Chem de Moulon,
91190 Gif sur Yvette,
France.

Dr. S. Muggleton (Project Coordinator Algorithms),
Turing Institute,
Glasgow G1 2AD,
UK.

Dr. R. Rohwer,
Department of Computer Science and Applied Mathematics,
Aston University,
Birmingham B4 7ET,
UK.

Herr J. Stender (Project Coordinator Prediction and Control),
Brainware GmbH,
Gustav-Meye-Allee 25,
1000 Berlin,
Germany.

Dr. C.C. Taylor (Project Coordinator Data),
Department of Statistics,
School of Mathematics,
Leeds University,
Leeds LS2 9JT,
UK.

Prof. Wysotzki,
Fraunhofer-Gesellschaft IITB-EPO,

Kurstrasse 33,
O-1086 Berlin,
Germany.

Dr. Alejandro Moya (Project Officer),
Breydei 9/179,
45 Ave. d'Auderghem,
B-1049 Brussels,
Belgium.

APPENDIX B – MEASURES FOR DATASETS

Number of observations

This is the total number of observations in the whole dataset.

Number of attributes

The total number of attributes.

Number of classes

The total number of classes represented in the entire dataset.

Number of bin_cat

The total number of number of attributes that are binary or categorical.

Cost_matrix

If the dataset has a cost matrix (1 = yes).

Homogeneity of covariances

Homogeneity of covariances is the geometric mean ratio of standard deviations of the populations of individual classes to the standard deviations of the sample, and is tabulated as $SD_$ratio. The $SD_$ratio is strictly greater than unity if the covariances differ, and is equal to unity if and only if all individual covariances are equal to the covariances of the whole sample.

Mean absolute correlation coefficient

The correlations ρ_{ij} between all pairs of attributes indicate the dependence between the attributes. They are calculated for

each class separately. The absolute values of these correlations are averaged over all pairs of attributes and over all populations to give the measure *corr_abs*, which is a measure of interdependence between attributes. If *corr_abs* is near unity, there is much redundant information in the attributes, and some procedures such as logistic discriminants may have technical problems associated with this. Also, CASTLE may be misled substantially by fitting relationships to the attributes instead of concentrating on relationship between the classes and the attributes.

Canonical discriminant correlations

Examples of n attributes from a sample are points in a n-dimensional space, where they form clusters of roughly elliptical shape. The sample points from one population of class form a cluster around its population mean. In general, if there are k populations, the k means lie in a $k-1$ dimensional subspace. On the other hand, it happens frequently that the populations form some kind of sequence so that the population means are strung out along some curve that lies in a m – –dimensional space ($m < k-1$). For example, the simplest case occurs when $m = 1$ and the population means lie along a straight line.

Canonical discriminants are a way of systematically projecting the mean vectors in an optimal way to maximize the ratio of between-mean distances to within-cluster distances, successive discriminants being orthogonal to earlier discriminants. Thus the first canonical discriminant gives the best single linear combination of attributes that discriminates between the populations. The second canonical discriminant is the best single linear combination orthogonal to the first, and so on. The success of these discriminants is measured by the *canonical correlations.* If the first canonical correlation is close to unity, the k means lie along a straight line nearly. If the $q+1$th canonical correlation is near zero, the means lie in q – –dimensional space.

Variation explained by first four canonical discriminants

The sum of the first q eigenvalues of the canonical discriminant matrix divided by the sum of all the eigenvalues represents the 'proportion of total variation' explained by the first q

canonical discriminants. We tabulate, as $fract\_q$, the values of $(\lambda_1 + ... + \lambda_q)/(\lambda_1 + \lambda_2 + \ldots + \lambda_p)$ for $q = 1, 2, 3, 4$. This gives a measure of collinearity of the class means. When the classes form an ordered sequence, for example soil types might be ordered by wetness, the class means typically lie along a curve in low dimensional space. The λs are the squares of the canonical correlations. The significance of the λs can be judged from the χ^2 statistics produced by 'manova'.

Univariate skewness and kurtosis

These are univariate measures of the non-Normality of the attributes when considered separately. The univariate skewness $\beta_1(i, j)$ is a measure of asymmetry of the distribution of the ith attribute in the jth class, essentially describing how the left and right tails differ. As a single measure of skewness for the whole data set, we quote the mean absolute value of $\beta_1(i, j)$, averaged over all attributes and over all classes. This gives the measure *skew_abs*. For a normal population *skew_abs* should be zero: for uniform and exponential variables, the theoretical values of *skew_abs* are zero and 16 respectively.

To compare the thickness of the tails of the distributions compared to that of the Gaussian or normal, we use the kurtosis $\beta_2(i, j)$ of the ith attribute in the jth class. As an overall measure, we use the average of the univariate kurtosis $\beta_2(i, j)$, averaged over all attributes and populations. This gives the measure *kurtosis*. For a normal population, $kurtosis = 3$ exactly, and the corresponding figures for a uniform and an exponential are 1.8 and 9 respectively.

REFERENCES

Acid, S., de Campos, L., González, A., Molina, R. and Pérez de la Blanca (1991) CASTLE : A Tool for Bayesian Learning. In *Esprit Conference 1991.*

Aha, D. and Kibler, D. (1988) Detecting and Removing Noisy Instances From Concept Descriptions (Technical Report 88-12). Department of Information and Computer Science. University of California, Irvine, CA, 92717.

Breiman, L., Friedman, J.H., Olshen, R. and Stone, C. (1984) *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, Ca.

Cherkaoui, O. and Cléroux, R. (1991) Comparative study of six classification methods for mixtures of variables. In *Computing Science and Statistics. Proceedings of the 23rd Symposium on the Interface, Fairfax, Virginia.* (ed. E. M. Keramidas), pp. 233–236.

Clark, P. and Niblett, T. (1988) The CN2 induction algorithm. *Machine Learning*, **3**(4), 261–283.

Clark, P. and Boswell, R. (1991) Rule induction with cn2: some recent improvements. In *EWSL '91: Machine Learning: Proceedings of the European Working Session on Learning*, pp. 151–163, Springer-Verlag, Berlin.

Cox, D.R. (1966) Some procedures associated with the logistic qualitative response curve. *Research Papers in Statistics: Frestschrift for J. Neyman, vol 45* (ed. F.N. David), Wiley, New York.

Day, N. and Kerridge, D. (1967) A general maximum likelihood discriminant. *Biometrics*, **23**, 313–323.

Fisher, D.H. and McKusick, K.B. (1989) An empirical comparison of ID3 and back-propagation and machine learning classification methods. In *International Joint Conference on Artificial Intelligence*, pp. 788–793. Morgan Kaufmann, Detroit.

Fisher, R.A. (1936) The use of multiple measurements in taxonomic problems. *Ann. Eugen.*, **7**, 179.

Friedman, J.H. and Stutzle, W. (1981) Projection pursuit regression. *J. Amer. Statist. Assoc.*, **76**, 817-823.

Gennari, J. *Models of Incremental Concept Formation* (to appear in the AI journal)

Goodman, R.M. and Symth, P. (1989) The induction of probabilistic rule sets – the itrule algorithm. In *Proceedings of the Sixth International Workshop on Machine Learning* (ed. B. Spatz), pp. 129–132, Morgan Kaufmann, San Mateo, CA.

Gordon, L. and Olshen, R.A. (1978) Asymptotically efficient solutions to the classification problem. *Annals of Statistics*, **6**, 515–544.

Hart, P.M. (1968) The condensed nearest neighbour rule. *IEEE Trans. Comput.*, **24**, 515–516.

Henery, R.J. and Taylor, C.C. (1992) StatLog: An evaluation of machine learning and statistical algorithms. *Compstat 1992*, Neuchatel.

Kohonen, T. (1989) *Self-Organisation and Associative Memory.* Springer-Verlag.

Kressel U., Franke J. and Schuermann J. (1990) Polynomial Classifier versus Multilayer Perceptron, *DAGM.*

Kressel U., (1991) The impact of the learning set size in handwritten digit recognition. In *ICANN 91, Helsinki.*

McClelland, J.L., Rumelhart, D.E. and Hinton, G.E. (1986) *Parallel Distributed Processing: explorations in the microstructure of cognition. Volumes I, II and III.* MIT Press, Cambridge, MA.

Michalski, R.S. (1983) A theory and methodology of inductive learning. *Artificial Intelligence*, **20**(2) 111–161.

Mooney, R., Shavlik, J., Towell, G. and Gove, A. (1989) An experimental comparison of symbolic and connectionist learning algorithms. In *International Joint Conference on Artificial Intelligence,* pp. 775–780 Morgan Kaufmann.

Pearl, J. (1988) *Networks of Belief: Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufman.

Poggio, T. and Girosi, F. (1990) Networks for approximation and learning. *Proceedings of the IEEE*, **78**(9), 1481–1497.

Quinlan, J.R. (1986) Induction of Decision Trees. *Machine Learning,* **1**, 81–106.

Quinlan, J.R. (1987) Simplifying decision trees. *International Journal of Man-Machine studies*, **27**(3), 221–234.

Quinlan, J.R. (1990) Learning logical definitions from relations. *Machine Learning,* **5**, 239–266.

Ripley, B.D. (1992) Statistical aspects of neural networks *SemStat*, Sanbjerg, Denmark.

Sammut, C. (1988) Experimental results from an evaluation of algorithms that learn to control dynamic systems. In *Proceedings of the Fifth International Conference on Machine Learning*, pp. 437–443, Morgan Kaufmann

Shavlik, J.W., Mooney, R.J. and Towell, G. (1991) Symbolic and neural learning algorithms: an experimental comparison. *Machine Learning*, **6**, 2, 111–143.

Siebert, J.P. (1987) Vehicle recognition using rule based methods.

Turing Institute Technical Report TIRM-87-018.

Sutherland, A., Henery, R., Molina, R., Taylor, C. and King, R. (1992) Statistical Methods in Learning In *Conference on Information Processing and Management of Uncertainty. (IPMU '92)*, Palma de Mallorca.

StatLog Technical Annexe (1990) *Esprit project no 5170.*

StatLog Report on phase II (1992) *Esprit project no 5170.*

Titterington, D.M., Murray, G.D., Murray, L.S., Spiegelhalter, D.J., Skene, A.M.,Habbema, J.D.F. and Gelpke, G.J. (1981) Comparison of Discrimination Techniques Applied to a Complex Data Set of Head Injured Patients (with discussion). *J. Royal Statist. Soc.*, **144**, 145–175.

Unger, S. and Wysotzki, F. (1981) *Lernfaehige Klassifizierungssystems, Academieverlag*, Berlin.

Weiss, S.M., Galen, R.S. and Tadepalli P.V. (1987) Optimizing the predictive value of diagnostic decision rules. In *Proceedings AAAI-87: Sixth National Conference on Artificial Intelligence*, pp. 521–526

Weiss, S.M. and Kapouleas, I. (1989) An empirical comparison of pattern recognition, neural nets and machine learning classification methods. In *IJCAI 89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 781–787

Weiss, S.M. and Kapouleas, I. (1991) *Computer Systems that Learn.* Morgan Kaufmann, San Mateo.

Table 13.1. The characterization of the eight datasets: Satellite, Digits, KL-Digits, Vehicle, Head, Heart, Credit, and Shuttle (see Appendix A for details). The measures for KL-digits are based on the training examples (except no. of examples). For Shuttle a reduced dataset was used with all members of classes 2,3,6,7 and only 2000 examples each from classes 1,4,5. Formally speaking, the skewness and kurtosis figures for classes 2 and 7 are undefined as there are variables here with attributes whose values are constant (attribute 4 for class 2 and attribute 1 for class 7).

	Satellite	Digits	KL-digits	Vehicle	Head	Heart	Credit	Shuttle
N_examples	6435	18000	18000	846	900	270	8900	58000
N_attributes	36	16	40	18	6	13	16	9
N_classes	6	10	10	4	3	2	2	7
N_bin_cat	0	0	0	0	1	5	8	0
cost_matrix	0	0	0	0	1	1	0	0
SD_ratio	1.2970	1.5673	1.9657	1.5392	1.1231	1.0612	1.0273	1.6067
corr_abs	0.5977	0.2119	0.1093	0.4828	0.1217	0.1236	0.0825	0.3558
cancor_1	0.9366	0.8929	0.9207	0.8420	0.7176	0.7384	0.7618	0.9668
cancor_2	0.9332	0.8902	0.9056	0.8189	0.1057	0.0000	0.0000	0.6968
cancor_3	0.7890	0.7855	0.8440	0.3605	0.0000	NA	NA	0.2172
cancor_4	0.2385	0.6982	0.7761	0.0000	NA	NA	NA	0.1458
fract_1	0.3586	0.2031	0.1720	0.4696	0.9787	1.0000	1.0000	0.6252
fract_2	0.7146	0.4049	0.3385	0.9139	1.0000	1.0000	1.0000	0.9499
fract_3	0.9691	0.5621	0.4830	1.0000	1.0000	NA	NA	0.9814
fract_4	0.9923	0.6862	0.6053	1.0000	NA	NA	NA	0.9957
skew_abs	0.7316	0.8562	0.1802	0.8282	1.0071	0.9560	1.2082	4.4371
kurtosis	4.1737	5.1256	2.9200	5.1800	5.0408	3.6494	4.4046	160.3108

Table 13.2. Satellite image results – '@' times based on transputer; '#' no time given for classifying training examples; '!' uses new version of Cal5, so real time should be higher; examples were cycled 40 times and 1600 nodes are used in this Kohonen feature map.

Algorithm	*Source*	*Accuracy(%)*		*Time(sec.)*	
		Train	Test	Train	Test
k-N-N	leeds	91.1	90.6	2105	944
Radial	strath	88.9	87.9	723	74
Alloc80	leeds	96.4	86.8	63840	28757
INDCART	strath	98.9	86.3	2109	9
CART	granada	NA	86.3	348	14
Backprop	strath	88.8	86.1	54371	39
NewID	turing	93.3	85.0	296	53
C4.5	turing	95.7	84.9	449	11
CN2	daimler	98.6	84.8	1718	16
Quadra	strath	89.4	84.7	276	93
Cal5!	fraunh	87.8	84.6	1345	13
AC2	isoft	NA	84.3	8244#	17403
SMART	leeds	87.7	84.1	83068	20
LogReg	strath	88.1	83.1	4414	41
Kohonen@	luebeck	NA	82.1	12627	129
Discrim	strath	85.1	82.9	68	12
CASTLE	granada	81.4	80.6	NA	NA
Bayes	strath	71.3	69.3	56	12

Table 13.3. Digits results – '@' times based on transputer

Algorithm	*Source*	*Accuracy(%)*		*Time(sec.)*	
		Train	Test	Train	Test
k-N-N	leeds	98.4	95.3	2231	2039
Quadra	strath	94.8	94.6	194	152
Alloc80	leeds	93.4	93.2	3250	134370
Kohonen@	luebeck	NA	92.5	67176	2075
Backprop	strath	92.8	92.0	28910	110
Radial	strath	92.0	91.7	1150	250
LogReg	strath	92.1	91.4	5110	138
SMART	leeds	90.4	89.6	51435	33
Discrim	strath	89.0	88.6	65	30
CN2	turing	99.9	86.6	2229	78
NewID	turing	91.9	85.5	516	80
C4.5	turing	95.9	85.1	543	39
INDCART	strath	98.9	84.6	3615	51
AC2	isoft	NA	84.5	32965	22384
CASTLE	granada	82.5	82.1	4341	4090
CART	granada	84.1	81.9	291	40
ITrule	brainwr	NA	77.8	8283	NA
Bayes	strath	78.0	76.7	104	62
Cal5	fraunh	78.5	71.5	570	55

Table 13.4. KL digits results

Algorithm	*Source*	*Accuracy(%)*		*Time(sec.)*	
		Train	Test	Train	Test
k-N-N	leeds	100.0	98.0	0	13881
Alloc80	leeds	100.0	97.6	48106	48188
Quadra	strath	98.4	97.5	1990	1648
Backprop	strath	95.9	95.1	129840	240
LogReg	strath	96.8	94.9	3538	1713
Radial	strath	95.2	94.5	2280	580
SMART	leeds	95.7	94.3	389448	58
Discrim	strath	93.0	92.5	141	54
C4.5	daimler	NA	82.2	1434	35
CASTLE	granada	87.4	86.5	49162	45403
NewID	isoft	100.0	83.8	785	109
AC2	isoft	100.0	83.2	27382	24791
INDCART	granada	99.7	83.2	3550	53
CN2	turing	96.4	82.0	9183	103
ITrule	brainwr	NA	78.4	NA	8175
Bayes	isoft	79.5	77.7	141	76
Cal5	fraunh	75.2	66.9	3049	64.0

Table 13.5. Vehicle data results – '@'times based on transputer; '*' indicates that the time includes training and testing

Algorithm	*Source*	*Accuracy(%)*		*Time(sec.)*	
		Train	Test	Train	Test
Quadra	strath	91.5	85.0	251	29
Alloc80	leeds	100.0	82.7	30	10
LogReg	strath	83.3	80.9	758	8
Backprop	strath	83.2	79.3	14411	4
Discrim	strath	79.8	78.4	16	3
SMART	leeds	93.8	78.3	3017	1
C4.5	turing	93.5	73.4	153	1
k-N-N	leeds	100.0	72.5	164	23
CART	granada	NA	71.6	29	1
AC2	isoft	NA	70.3	595	23
NewID	turing	97.0	70.2	18	1
INDCART	strath	95.3	70.2	85	1
Radial	strath	90.2	69.3	1736	12
CN2	turing	98.2	68.6	100	1
ITrule	brainwr	NA	67.6	985*	NA
Kohonen@	luebeck	88.5	66.0	5962	50
Cal5	fraunh	70.3	64.9	41	1
CASTLE	granada	49.5	45.5	23	3
Bayes	strath	48.1	44.2	4	1

Table 13.6. Head injury results – '*' indicates that the time includes both training and testing; '!' present version does not utilize costs fully.

Algorithm	*Source*	*Avg. Cost*		*Time(sec.)*	
		Train	Test	Train	Test
LogReg	strath	16.6	18.0	736	7
Discrim	strath	19.8	19.9	28	3
Quadra	strath	17.8	20.1	253	32
CASTLE	granada	18.9	20.9	30	3
CART	granada	19.8	20.4	20	1
Backprop	strath	18.2	21.5	656	32
SMART	leeds	13.6	21.8	420	4
Bayes	strath	23.6	25.0	2	1
INDCART!	strath	21.9	29.3	56	1
k-N-N	leeds	9.2	35.3	9	11
ITrule	brainwr	NA	37.6	7*	NA
Cal5	fraunh	32.3	38.4	5	1
Alloc80	leeds	45.3	46.1	322	276
NewID	isoft	18.9	53.6	16	2
Radial	strath	53.4	63.1	17	5
C4.5	daimler	59.8	82.0	49	1

Table 13.7. Heart disease results – '*' indicates that the time includes both training and testing; '!' present version does not utilize costs fully; '@' times based on transputer.

Algorithm	*Source*	*Avg. Cost*		*Time(sec.)*	
		Train	Test	Train	Test
Bayes	strath	0.351	0.374	6	3
Discrim	strath	0.315	0.393	14	3
LogReg	strath	0.271	0.396	128	7
Alloc80	leeds	0.394	0.407	31	5
Quadra	strath	0.274	0.422	60	16
CASTLE	granada	0.374	0.441	16	3
CART	granada	0.463	0.452	7	1
k-N-N	leeds	0	0.478	0	1
SMART	leeds	0.264	0.478	725	1
ITrule	brainwr	NA	0.515	5*	NA
Cal5	fraunh	0.517	0.559	8*	NA
Backprop	strath	0.381	0.574	128	13
INDCART!	strath	0.261	0.630	8	1
Kohonen@	luebeck	0.429	0.693	227.1	1.9
AC2	isoft	0	0.744	250*	NA
CN2	turing	0.206	0.767	25	5
C4.5	turing	0.439	0.781	34	1
Radial	strath	0.303	0.781	26	4
NewID	isoft	0	0.844	12*	NA

Table 13.8. Credit data results – '!' These algorithms may have classified the test set exactly the same way (perhaps based only on attribute 14; '*' indicates times based on training and testing; '@' times based on transputer.)

Algorithm	*Source*	*Accuracy(%)*		*Time(sec.)*	
		Train	Test	Train	Test
INDCART	isoft	92.0	92.0	206	193
SMART	leeds	89.5	89.1	2151	5
CASTLE	granada	88.3	88.1	81	33
Cal5	fraunh	89.5	87.4	76	12
CART!	granada	87.2	87.0	19	19
NewID!	isoft	100	87.0	380	4
Discrim!	strath	87.2	87.0	71	16
AC2	isoft	100	87.0	7970	410
Radial	strath	87.5	87.0	837	54
LogReg	strath	87.3	86.9	251	30
Backprop	strath	88.2	86.9	28819	19
CN2	daimler	100	86.7	2309	13
Quadra	strath	86.3	86.0	78	20
Bayes	isoft	85.0	84.4	44	8
ITrule	strath	83.9	83.3	773*	NA
Alloc80	leeds	97.7	83.0	24	738
Kohonen@	luebeck	83.3	81.0	30704	71
k-N-N	leeds	100.0	80.6	0	1851

Table 13.9. Shuttle control data results – '!' indicates that only a sample of the training data could be used (C4.5 32760; AC2 4351; Bayes, INDCART 32625; Discrim, Quadra, LogReg 20000); '*' indicates times based on training and testing.

Algorithm	*Source*	*Accuracy(%)*		*Time(sec.)*	
		Train	Test	Train	Test
NewID	daimler	100	99.99	6180*	NA
CN2	daimler	100	99.97	11160*	NA
C4.5!	turing	99.90	99.96	11131	11
SS1 0.81 INDCART!	strath	99.96	99.92	1152	16
AC2!	isoft	100.0	99.68	4493	3397
Cal5	fraunh	NA	99.60	552	18
k-N-N	leeds	99.61	99.56	65270	21698
SMART	leeds	99.39	99.41	110010	93
Alloc80	leeds	99.05	99.17	55215	18333
CASTLE	granada	96.34	96.23	819	263
LogReg	strath	96.06	96.17	6946	106
Bayes!	strath	95.42	95.45	1030	22
Discrim!	strath	95.02	95.17	508	102
Backprop	strath	95.1	95.1	28800	75
Quadra!	strath	93.65	93.28	709	177