Inductive, Evolutionary and Neural Computing Techniques for Discrimination: A Comparative Study[*]

Siddhartha Bhattacharyya
Department of Information and Decision Sciences
College of Business Administration
University of Illinois at Chicago
601 South Morgan Street
Chicago, IL 60607-7124
sidb@uic.edu

Parag C. Pendharkar,
Capital College, Penn State University,
777 W. Harrisburg Pike,
Middletown, PA.
e-mail: pxp19@psu.edu

[*] The authors extend sincere thanks to two anonymous referees and the associate editor for their invaluable suggestions in improving the manuscript.

Abstract

This paper provides a comparative study of machine learning techniques for two-group discrimination. Simulated data is used to examine how the different learning techniques perform with respect to certain data distribution characteristics. Both linear and non-linear discrimination methods are considered. The data considered has been previously used in the comparative evaluation of a number of techniques, and helps relate our findings across a range of discrimination techniques.

Subject Areas:, *Discriminant Analysis, Genetic Algorithms, Genetic Programming, Inductive Learning, Machine Learning, and Neural Networks.*

Preview

The classification problem of assigning observations into one of several groups plays a key role in decision making. The binary classification problem, where the data are restricted to one of two groups, has wide applicability in problems ranging from credit scoring, default prediction and direct marketing to applications in biology and medical domains. It has been studied extensively by statisticians, and in more recent years a number of machine learning approaches have been proposed. The latter group of techniques can broadly be categorized as so-called "soft computing" methods, noted to tradeoff optimality and precision for advantages of representational power and applicability under wider conditions. The problem is often referred to as discriminant analysis in statistics and supervised learning in the machine learning literature. The form

of the solution varies with the technique employed, and is expressed in some form of discrimination rule or function based on the multivariate data defining each observation.

Techniques from the statistics realm beginning with Fisher's seminal work (Fisher, 1936) include linear, quadratic and logistic discriminant models, and are amongst the most commonly used. They differ with respect to assumptions on group distributions and functional form of the discriminant. In linear discriminant analysis, the model is expressed in terms of vector of weights **w** together with scalars c1 and c2 such that given a observation defined by a vector **x** of attribute values, it is classified into group 1 if **w** $\cdot$ **x** $\leq$ c1 and into group 2 when **w** $\cdot$ **x** > c2 (c1 and c2 are generally considered equal). Linear models are generally preferred for decision making (Hand, 1981), given the ease of interpretation of results and higher reliability in predicting unseen cases; nonlinear models, though more accurate on the training data, tend to show sharp declines in performance on unseen test samples (Altman, 1981). The major drawback with these methods arise from the fact that real-world data often do not satisfy their underlying assumptions. Nonparametric methods are less restrictive, and popular amongst them are the k-nearest neighbor and linear programming methods (Freed & Glover, 1981; Hand, 1981).

Machine learning techniques used for discrimination fall into two categories: the connectionist models employing some form of neural network learning algorithm, and the inductive learning models where the discriminant is expressed in symbolic form using rules, decision trees, etc. The backpropagation neural network (Rumelhart, Hinton, & Williams, 1986) is most commonly used of connectionist schemes. Various induction algorithms have been suggested for classification, popular amongst them being CART

(Brieman, Friedman, Olshen &Stone, 1984), ID3 (Quinlan, 1986), and CN2 (Clark & Niblett, 1989). A third set of techniques that has received less attention for classification problems are based on the evolutionary computing paradigm, and include genetic algorithms (Holland, 1975). With demonstrated robustness in a wide variety of applications, genetic algorithms have been noted to be useful for discrimination too (Koehler, 1991).

Recent years have witnessed an increasing use of such machine learning techniques for classification in decision making. Goonatilake (1995) argues that an ability to learn decision processes and tasks directly from data forms the most important feature of intelligent business systems and notes: "some professionals such as financial traders and insurance assessors have a very high premium on their available time and therefore the capability to learn directly from data without human intervention becomes very important" (p.5). Improved performance of learning techniques over traditional methods have been reported across a range of applications, including credit evaluation (Walker, Haasdijk, & Gerrets, 1995), bankruptcy prediction (Tam and Kiang, 1992; Raghupati, Schkade, & Raju, 1991), investment management (Refenes, Zapranis, Connor, & Bunn, 1995), analysis of accounting reports (Berry and Triueiros, 1993), financial market prediction (Yoon and Swales, 1991; Allen and Karjalainen, 1993) and direct marketing (Furness, 1995; Bhattacharyya, 1996; David Shepard Associates, 1990). Organizations implementing such learning based systems have reported significant cost savings; for example, Visa International has realized savings to the tune of $40 million over a six-month period through its neural network based system used for credit card fraud detection

(Goonatilake, 1995). Classification also forms a key task in the emerging field of data mining where machine learning techniques have found widespread use (Piatetsky-Shapiro and Frawley, 1991). A number of commercial tools are available today and provide decision makers a range of learning techniques – for example Clementine provides for neural networks and decision-tree induction based on Quinlan's (1993) C4.5 system; Darwin, from Thinking Machines Corporation includes a range of algorithms including genetic algorithms and neural networks; Mineset from Silicon Grapics, seeking broad applicability, provides a range of machine learning as well as statistical techniques and aims at providing a variety of tools for decision makers to choose from for different problem tasks.

Given this variety of approaches available, a key and difficult task for decision makers becomes the selection of a particular technique that "best" matches a given problem. With little theoretical guidance on the relative practical utility of different machine learning approaches -- most formal analyses focus on worst case scenarios (Ehrenfeucht, Haussler, Kearns, & Valiant, 1989; Haussler, 1988) -- empirical studies provide the sole means for comparative analyses. The literature reports a number of studies comparing the performance of machine learning versus statistical approaches (Atlas, Cole, Connor, El-Sharkawi, Marks, Muthusamy, & Barnard, 1990; Chung and Silver, 1992; Fisher and McKusick, 1989; Hansen, McDonald, & Stice, 1992; Shavlik, Mooney, & Towell, 1991; Weiss and Kapuoleas, 1989). Compiled interpretation of results from these various studies is, however, difficult, given variations in the data and algorithms used, data pre-processing steps employed that could possibly favor one

technique over another, optimization of technique-related parameters in some studies, etc. Arguing thus, the StatLog project (King, Henry, Feng, & Sutherland, 1994) undertakes a comparative evaluation of a large number of techniques on an assortment of real-world data. Reviewing a number of comparative studies on symbolic and neural network methods, Quinlan (1993) emphasizes that no single method uniformly demonstrates superior performance.

Such findings are in line with recent theoretical results on the "No Free Lunch" (NFL) theorems on search (Wolpert and McReady, 1995), pointing out that "positive performance in some learning situations must be balanced by negative performance in others" (Schaffer, 1994), that is, search algorithms perform the same when performance is considered averaged over a sufficiently large group of problems. Radcliffe and Surry (1995), focusing on the representations of the search space, present a more accessible form of these results, and elaborate on the role of domain knowledge in the problem representation and search operators used. The NFL results emphasize that conclusions regarding a technique's performance can be made only with respect to the specific problem type being examined. Use of different techniques needs to be guided by an understanding of their respective strengths and limitations. It thus emphasizes empirical analyses using problem generators where problem characteristics can be systematically controlled and studied, as opposed to considering performance over an arbitrary assortment of problems.

Most comparisons of machine learning techniques are based on real-world data sets from diverse domains. As mentioned, use of simulated data allows greater control

and evaluation under strictly known conditions, thus helping relate defining data characteristics with technique. This research focuses on the performance of learning algorithms with respect to a range of group distribution characteristics. While much examined for statistical approaches, data distribution characteristics have received far less attention in the comparative learning techniques literature, given the nonparametric nature of learning techniques. We use data previously reported in the literature in comparing statistical and linear programming approaches for the discrimination problem (Abad & Banks, 1993; Joachimsthaler & Stam, 1988; Koehler & Eregnuc, 1990).

It has been noted that data conditions of multivariate normality and variance-covariance homogeneity represent critical considerations in seeking to apply traditional statistical methods (Joachimsthaler & Stam, 1988). Machine learning techniques have, however, not been analyzed with respect to these key data attributes. We examine the performance of learning techniques with respect to these aforementioned data characteristics, to aid in the selection of machine learning versus traditional methods for a discriminant task. Joachimsthaler and Stam (1988) generate data with different levels of kurtosis and dispersion to compare a linear programming method and linear, logistic and quadratic discriminant functions. A number of other linear programming based discriminant methods have also been compared using this data (Abad & Banks, 1993; Koehler & Eregnuc, 1990), and Koehler (1991) also uses it to evaluate a genetic algorithm based discriminant approach. The wide earlier use of this same data allows for an equitable comparison of results across a broad range of discriminant techniques.

Four learning techniques are selected to provide coverage of different search paradigms -- the decision-tree learning program C4.5 (Quinlan,1993), implementing Quinlan's widely cited ID3 (Quinlan, 1986) inductive learning algorithm; a three-layer neural network using the backpropagation learning rule (Rumelhart et al., 1986); a genetic algorithm implementation that seeks to learn the weight vector of a linear discrimination model, similar to that reported in Koehler (1991); a public domain implementation of genetic programming (Koza, 1992), a variant of genetic search that uses a hierarchical representation to obtain general form (not restricted to linear) mathematical discrimination models. Fisher's linear discrimination procedure is used as a baseline for comparison.

The next section provides an overview of the learning techniques considered. A brief description of the data sets used and experimental method is then presented, followed by the results and analyses sections. The managerial significance of these findings is then discussed, and the concluding section addresses future research issues.

Overview of Techniques

Back-propagation Neural Networks

Based on neuronal computations in the brain, neural networks have been applied to numerous classification and discrimination problems (Shavlik et al., 1991). Each neuron is an elemental processing unit, and forms part of a larger network, the architecture for which is specified to fit the problem under consideration. Neurons (or nodes in the net) are interconnected through a set of weighted, directed arcs. The net configuration,

together with the set of arc weights, defines the model for a classification problem. A learning algorithm specifies a procedure for updating the arc weights. A variety of neural net paradigms exist, and differ in manner of node interconnections and learning procedure used.

It is useful to differentiate network architectures by the decision boundaries they form in classification (Lippmann, 1987). A single-layer network, using the perceptron convergence procedure (Rosenblatt, 1962) delineates two groups by a hyperplane, and the procedure is guaranteed to converge if the two classes are linearly separable. A modification of the perceptron convergence procedure can be used to minimize the least-mean-square error between the actual and desired outputs in a two-layer network (Duda & Hart, 1973), and this yields convex regions separating the two groups. A multi-layer network is the most general, and can form arbitrary complex regions separating the groups (Lippmann, 1987).

The backpropagation learning algorithm (Rumelhart et al., 1986), most commonly used to train multi layer networks, implements a gradient search to minimize the squared error between realized and desired outputs. Multi layer networks have been shown to approximate the Bayes optimal classifier, with the node outputs (one output node per class) approximating the class posterior probabilities (Ruck, Rogers, Kabrisky, & Oxley, 1990).

Figure 1 shows a typical three-layer network used for discrimination. The number of input layer nodes corresponds to the number of independent variables describing the data. The number of nodes in the hidden layer determines the complexity of the decision

surface generated, and needs to be empirically determined to best suit the data being considered. While larger networks tend to overfit the data, too few hidden layer nodes can hamper learning of an adequate separating region. Though having more than one hidden layer provides no advantage in terms of nature of decision surface generated, it can in cases provide for faster learning (Rumelhart et al., 1986).

<Insert Figure 1 about here>

For any neuron n_k, its output is determined by:

$$o_k = f\left(\sum_i w_{ik} o_k + q_k\right),$$

where w_{ik} is the weight on the arc connecting neuron n_k with the neuron n_i from the previous layer, $f(.)$ represents an activation function, usually the logistic function:

$$f(x) = \frac{1}{1+e^{-x}},$$

and q_k is a bias associated with each neuron effecting a translation of the logistic function to allow for a better fit to the data. At the output layer node, an output value less than 0.5 is considered a categorization into Group 0 and values greater than 0.5 imply Group 1. The network is initialized with small random values for the weights, and the backpropagation learning procedure is used to update the weights as the data is iteratively presented to the input-layer neurons. At each iteration, the weights are updated by backpropagating the classification error, e, as follows:

$$\Delta_t w_{ik} = h\, d_k o_i + a \Delta_{t-1} w_{ik}$$

where

$$d_k = \begin{cases} o_k(1-o_k)e, & \text{if } n_k \text{ is an output neuron} \\ o_k(1-o_k)\sum_j w_{kj}\, d_j, & \text{if } n_k \text{ is a hidden layer neuron.} \end{cases}$$

Here, η is the learning rate and α is the momentum term.

For the experiments in this study, a three-layer network is used, with three input nodes corresponding to the data attributes and a single output node. The number of hidden layer neurons is chosen as twice the number of data inputs, a commonly used heuristic in the literature (Patuwo, Hu, & Hung, 1993).

Decision Trees using ID3

Quinlan's (1986) ID3 has been widely reported in the literature and is amongst the more commonly used induction algorithms available in commercial implementations. We use the C4.5 programs (Quinlan, 1993) that incorporate enhancements to the basic algorithm to consider continuous variables and also includes pruning.

A decision tree is constructed top-down, with each new node implementing a partition on the data based on the attribute (independent variable) tested at that node. The attribute selection at a node is based on an information-theoretic goodness-of-split criterion. The entropy-based criterion selects an attribute A that minimizes

$$E(A) = -\sum_{i=1}^{V} \frac{n_i}{n} \sum_{j=1}^{K} \frac{n_{ij}}{n_i} \log_2 \frac{n_{ij}}{n_i},$$

and thereby maximizes the gain function:

$$\text{Gain(A)} = \sum_{j=1}^{K} \frac{n_j}{n} \log_2 \frac{n_j}{n} - E(A).$$

Here, V is the number of values for attribute A; K is the number of classes; n is the total number of observations; n_i represent the number of observations having the i^{th} value for attribute A or number of observations in the i^{th} class, based on the context; and n_{ij} is the number of observations in the j^{th} class having the i^{th} value for attribute A. This gain criterion originally used, is noted to be biased towards attributes with greater number of values. The gain-ratio criteria helps normalize for this (Quinlan, 1993):

$$\text{Gain-Ratio(A)} = \frac{\text{Gain(A)}}{-\sum_{i=1}^{v} \frac{n_i}{n} \log_2 \frac{n_i}{n}}.$$

An enhanced attribute selection measure is given in Lopez de Mantraras (1991); though prediction accuracy of trees is, in general, not sensitive to selection criteria (Lopez de Mantraras, 1991; Mingers, 1989), the new measure is reported to yield smaller trees in certain cases.

For continuous attributes two-way splits based on threshold values are considered. All possible splits, given the values for the attribute present in the data, are considered. The recursive approach to partitioning the data can yield overly complex trees that very often overfit the data. Obtained decision trees are thus pruned (Quinlan, 1993).

Genetic Algorithms and Genetic Programming

Genetic algorithms (GAs) provide a stochastic search procedure based on principles of natural genetics and survival of the fittest. They operate through a simulated evolution process on a population of string structures that represent candidate solutions in

the search space. Each population member can specify to a symbolic classification rule as in Greene and Smith (1987), or can represents the (**w,c**) vector defining a discriminant function (Koehler, 1991). In this paper, we consider the latter approach, considering linear discriminants to allow direct comparisons with traditional Fisher's disciminant analysis. For non linear models determined through genetic search, we consider genetic programming (Koza, 1992).

In the genetic algorithm models, each population member defines the linear model (**w, c**) and is used to classify an observation **x** as:

$$\mathbf{wx} + \mathbf{c} \begin{cases} \leq 0 \text{ implies Group 1} \\ > 0 \text{ implies Group 2.} \end{cases}$$

We restrict each element $(\mathbf{w}, \mathbf{c})_i \in [0,1]$, since (**w, c**) classifies an observation similarly to λ (**w, c**) for any $\lambda \in \Re$, where $\Re$ denotes real numbers. Population members are represented directly as real numbers. Though much of the GA literature considers binary encodings, practical applications have shown improved performance with more direct higher level representations (Davis, 1991; Michalewicz, 1994). Standard fitness-proportionate selection (Goldberg, 1989) is used, together with the following search operators:

Arithmetic crossover (Michalewicz, 1994): Given two parents ($\mathbf{w_1}$, c_1) and ($\mathbf{w_2}$, c_2) an offspring ($\mathbf{w_o}$, c_o) is generated through a linear combination of the two parent strings: ($\mathbf{w_o}$, c_o) = λ ($\mathbf{w_1}$, c_1) + (1 - λ) ($\mathbf{w_2}$, c_2), where $\lambda \in [0,1]$. We consider λ a uniform random number in [0,1].

Exchange crossover: Two offspring are created by exchanging uniformly chosen elements of two parent strings as in uniform crossover (Syswerda, 1989). For example, given two parent strings $<s_1, s_2, s_3, s_4>$ and $<t_1, t_2, t_3, t_4>$, a set of exchange positions P = $\{p \mid p \in \{1,2,3,4\}\}$ are uniformly randomly chosen. Two offspring are then generated by exchanging the elements of the two parent strings indicated by the P positions. For instance, P= {1, 3} would yield the following two offspring: $<t_1, s_2, t_3, s_4>$ and $<s_1, t_2, s_3, t_4>$.

Mutation: A randomly chosen element of a string is replaced by a random value.

The fitness function seeks to minimize the number of misclassifications. The genetic learning procedure begins with a population of random strings, and can be summarized as:

```
While (not terminating-condition){
        evaluate-fitness of population members
        while next-generation population is not full {
                select two parents for next generation
                With probability pcross
                  perform crossover on two parents to get two new offspring
                With probability pmutate
                  perform mutation on each offspring
                Insert offspring into next generation
        }
}
```

The search is terminated after a fixed number of iterations.

Genetic programming (Koza, 1992) is a GA variant that uses a hierarchical instead of the flat-string representation. Each population member represents a function f (**x**) of the dependent variables that can be depicted as a parse tree, thus allowing arbitrarily complex discriminant functions (Figure 2a). The function f(**x**) is interpreted as a discriminant in the usual way:

$$f(\mathbf{x}) \begin{cases} \leq 0 \text{ implies Group 1} \\ > 0 \text{ implies Group 2.} \end{cases}$$

The functional form is determined by (1) a function set F defining all permissible arithmetic operators (+, -, *, /), together with any allowed functions (log(.), exp(.), sin(.), etc.) or logical operators (=, <, IF, OR, etc.), and (2) a terminal set T defining the variables and values allowed at the leaf nodes of the tree.

Crossover is defined as a operator that exchanges randomly chosen subtrees of two parents to create two new offspring, mutation randomly changes a subtree (Figure 2b). The function-set F = {+, -, *, /} is used, and the terminal-set is T = {$\Re$, x_1, x_2, ..., x_n}, where $\Re$ denotes the set of real numbers and x_i the independent variables. Fitness proportionate selection is used as in the GA, and the same fitness function is used. The basic learning procedure follows that described above for the GA case.

<Figures 2a and 2b about here>

Experiments

This section first provides a description of the data used for our comparative study, and then presents details of the experiments and results.

Data Sets

The experiments are based on data previously used for comparing a number of statistical and linear programming techniques for discrimination. Joachimsthaler and Stam (1988) examined Fisher's linear discriminant function, the quadratic discriminant function, the logistic discriminant function and a linear programming approach under varying group

distribution characteristics. Koehler and Erenguc (1990) and Abad and Banks (1993) use the same data generator to establish identical experimental conditions to evaluate a number other linear programming approaches. Koehler (1991) also uses this data to determine the effectiveness of a genetic search approach for discrimination.

Each data sample consists of three attributes, and has 100 observations equally split between two groups. The data varies with respect to type of the distribution, determined through the kurtosis, and variance-covariance homogeneity (dispersion). Four kurtosis values of -1, 0, 1, 3 correspond approximately to samples drawn from uniform, normal, logistic and Laplace population distributions. For dispersion variations across the data, if $\sum_i$ denotes the 3x3 dispersion matrix for Group i (i=1, 2), $\sum_1$ is always I (the identity matrix) while three different values are considered for the second group: $\sum_1$, $2\sum_1$, and $4\sum_1$. In order to minimize the effect of group overlap, the group means are set as follows: the group 1 mean is m'=(0,0,0) throughout, and the group 2 mean was m'=(.5, .5, .5) when $\sum_2=\sum_1$, m'=(.6,.6,.6) when $\sum_2=2\sum_1$ and m'=(.8,.8,.8) when $\sum_2=4\sum_1$.

There are thus 12 kurtosis-dispersion factor combinations leading to 12 data set groups. For each group, 100 random samples were taken, yielding a total of 1200 data sets. A more detailed description of the data can be found in Joachimsthaler and Stam (1988).

Most of the studies mentioned above have considered the misclassification rates on the training data. Since our study includes non linear techniques (Neural Networks and Genetic Programming) that generally tend to show sharper decline in performance on unseen cases, we also examine classification accuracy on separate holdout samples. For

this, the data was evenly split into training and testing samples. In each group of 100 samples, 50 were used for training and the resulting discriminants were tested one of the remaining 50 samples.

Experimental Results

The four kurtosis and three dispersion levels, together with the four comparison techniques yields a 3 way factorial design --- the three factors are the distribution (kurtosis), the variance heterogeneity, and technique. A 3-way ANOVA is used to test for fixed and interaction effects. Pairwise comparisons of the different techniques are also examined.

Table 1 lists the classification accuracy of the different techniques on the training data. Each row here indicates the average accuracy over 100 data sets. Table 2 lists the classification accuracy on holdout samples (prediction). As mentioned above, each value here is an average obtained over 50 different train-and-test sets. The relative performance of the techniques is graphically represented in Figures 3a and 3b. Table 3 shows the reliability of the various techniques, defined as the ratio of classification accuracy on prediction to that observed on the training data. Reliability indicates the extent of overfitting, and is useful in determining the impact of data distribution and variance heterogeneity on overfitting problems with the different techniques.

<Table 1, Table 2, Table 3 about here>

The analysis of variance of classification accuracy on the training data is presented in Table 4a. The technique factor main effect explains a majority of the total variation,

followed by variance heterogeneity, and distribution alone is not significant. As seen from the two-way interactions, technique and variance together explain most of the variation.

Pairwise comparisons are given in Tables 4b, 4c and 4d. All overall pairwise comparisons are significant, with highest F-values for comparisons between linear and nonlinear techniques. Though all pairwise comparisons based on variance are significant, lower F-values are noticed for Neural Network vs. Genetic Programming and Discriminant Analysis vs. Genetic Analysis. This is also noticed in Figure 3 -- the Neural Network and Genetic Programming lines display similar patterns in the three different variance- heterogeneity regions (points 1-4, 5-8, 9-12); the Genetic Algorithm and Discriminant Analysis lines are similarly alike. Pairwise comparisons based on kurtosis (Table 4d) reveal that differences between Neural Network
and C4.5, and between Genetic Programming and C4.5 are not significant at p=.01 level of significance; comparison of Genetic Programming vs. Neural Network also gives a p-value of .0046. The three nonlinear techniques thus exhibit similar performance with respect to distribution type, as do the two linear techniques (Genetic Algorithm vs. Discriminant Analysis). Interestingly, the difference between Genetic Algorithm and Genetic Programming is also not significant with respect to distribution.

<Table 4a, Table 4b, Table 4c, Table 4d about here >

Marked differences are noticed between the training and prediction results. Table 5a gives the overall analysis of variance results for prediction accuracy. All three main factors are significant, and in contrast with the training results, variance heterogeneity here is the dominant factor, followed by technique. The two linear techniques perform better

than the nonlinear techniques when the two group variances are homogeneous. Discriminant analysis performs marginally better than Genetic Algorithms for homogeneous variances, but performance of Discriminant analysis falls below that of Genetic Algorithms as variance heterogeneity increases. All the nonlinear techniques display a marked fall in performance from training to prediction, particularly in the homogenous variance case. Their prediction accuracies increase with increasing variance heterogeneity amongst the two groups.

On pairwise comparisons (Tables 5b, 5c and 5d), the prediction accuracies of Genetic Programming and Neural Networks are not significantly different with respect to variance. Also, in keeping with the training results, the differences between Neural Network and Genetic Programming and between Discriminant Analysis and Genetic Algorithm, based on distribution, are not significant. The difference between Neural Network and C4.5 based on distribution is, however, significant in the prediction case. In another difference with the training results, the predictions of Genetic Programming and Genetic Algorithm, based on distribution, are significantly different.

<Table 5a, Table 5b, Table 5c, Table 5d about here>

The graphs of Figures 3a and 3b illustrate the differences in classification accuracies between training and prediction. While nonlinear techniques perform better across experimental conditions in training, all display a sharp fall in prediction accuracies when the two group variances are homogenous. The three nonlinear techniques are seen to perform better (both in training and prediction) with increasing variance heterogeneity. All the nonlinear techniques are also seen to perform best on uniformly distributed data.

Tables 6a, 6b, 6c and 6d analyze the reliability of the different techniques. The analysis of variance shows that all the three main factors are significant, with the variance and technique explaining the major part of the total variation. On overall pairwise comparisons, Neural Network, Genetic Programming and C4.5 are noted to exhibit similar reliability. The difference between Genetic Algorithm and Discriminant Analysis is, however, significant. Based on variance, the difference between Neural Network and Genetic Programming is not significant, and Genetic Algorithm vs. Discriminant Analysis is also less significant than the rest of the paired comparisons. Comparisons based on variation with distribution both reveal no significant difference in the reliabilities of Neural Network, Genetic Programming and Genetic Algorithm. Figure 3c plots the reliabilities of the different techniques.

<Figure 3a, Figure 3b about here>

<Table 6a, Table 6b, Table 6c, Table 6d about here>

<Figure 3c about here>

Discussion and Significance

The utility of empirical analyses of the kind presented in this paper lies in the identification of strengths and limitations of different techniques and in being able to provide guidance to decision makers in the selection of technique for a given problem task. This section elaborates on our findings with a view to setting up certain guidelines on choice of technique to suit problem data characteristics. For this, we draw upon results from other related. The various techniques are examined with respect to the following

criteria: underlying representation, amount of training data available, data quality and possible data contamination, problem complexity, reliability and interpretability of results and complexity of learned models, and error distributions and potential incorporation of misclassification costs. These provide a base set of issues for decision-makers to consider in selecting an appropriate technique for a specific problem task. Our analysis reveals issues needing further investigation in this regard.

As expected, the representation underlying the techniques is a crucial factor differentiating performance. Where reliability is of utmost concern, the choice is clearly for the linear discriminant analysis and genetic algorithm models. Many applications, however, emphasize prediction accuracy, and reliability can be traded in favor of improved prediction performance. In such cases, while our results demonstrate that linear models be preferred when the two groups have common variance, nonlinear learning techniques can be advantageous, in spite of overfitting, when the variances differ.

Emerging evidence indicates that observed poor reliabilities with inductive learning techniques often arise from the small training samples used. Recent theoretical results (Ehrenfeucht et al., 1989; Haussler, 1988; Tsai & Koehler, 1993) emphasize that large training samples are necessary to adequately train learning techniques, and though based on worst case analyses, draw a note of caution on a number of excellent results reported in the literature with learning techniques. Empirical results also reveal improved learning performance with larger training samples. Patuwo et al. (1993), comparing backpropagation neural networks with linear and quadratic discriminant functions, a linear programming and a nearest neighbor approach, report that neural networks perform better

on training but not on the test data, but that the prediction performance of neural networks improves as sample size increases. Liang, Chandler, Han, and Roan (1992), comparing the performance of probit, neural networks and ID3, also find training sample size to be an important factor in classification performance. The performance of learning techniques, particularly of genetic algorithms and genetic programming, with varying training samples sizes needs further investigation.

All our three learning techniques with nonlinear representations exhibit a higher sensitivity to distribution kurtosis with moderate and large variance heterogeneity. Their performance decreases as the data distributions become increasingly kurtotic. Similar behavior has been reported for quadratic discriminant functions (Joachimsthaler & Stam, 1988). This behavior is thus attributable more to the nonlinear representations of the discriminants rather than being a function of the underlying algorithms. More from a practice standpoint, however, distributions with increasing kurtosis values are noted to correspond to data contaminated with unusual response values or where the data contains outliers (Joachimsthaler & Stam, 1988). The prediction accuracies of the considered nonlinear learning techniques, with moderate variance heterogeneity and high kurtosis values, is seen to drop almost to the level of the linear techniques. Decision makers thus need to exercise caution in using non-linear learning techniques where the data is likely to contain outliers or unusual response values. Where such data contamination is suspected, a preprocessing step for detecting and eliminating outliers is thus recommended if nonlinear learning techniques are to be used. Joachimsthaler and Stam (1988) report contrasting behavior with logistic discriminant functions and a linear programming

discrimination technique where performance improves with increasing kurtosis; these may be preferred techniques for such contaminated data.

Where linear models are preferred, due to their readier interpretability (Altman et al., 1981) or higher reliability, genetic algorithms are seen to provide a robust alternative to traditional discriminant analysis. Both these linear techniques exhibit largely uniform performance across the data conditions considered, with the genetic algorithm model outperforming discriminant analysis on the training data. In agreement with earlier studies discriminant analysis is observed to perform well when its assumptions of multivariate normality and group variance-covariance homogeneity are satisfied ; its prediction performance is noticed to be more sensitive to variance heterogeneity. While discriminant analysis performed marginally better on prediction when the two group variances were homogeneous, genetic algorithms showed higher prediction accuracy when group variances differed. Both techniques also perform similarly with respect to data distribution type in both training and prediction. Further, as described in Koehler (1991), the genetic algorithm approach is amenable to learning reduced dimension discriminants, a distinct advantage where models with a minimum number of variables are sought.

Genetic programming, implementing evolutionary search for an unspecified nonlinear function, is largely unexplored as a discriminant tool, and our results indicate that genetic programming can provide an attractive discriminant method. On the training data, the differences between the performances of the neural network and genetic programming models, though significant, is small ; on prediction, too, though the overall accuracy of the neural networks is higher than that of genetic programming, their behavior

is similar with respect to both variance heterogeneity and distribution type. Genetic programming thus models non- linearities comparably with neural network. Further, genetic programming models provide the added advantage of interpretability -- the output is a function of the independent variables rather than a network of neurons. While certain application areas, for example, character recognition, have prediction accuracy as a sole concern and can thereby use a black box approach, model interpretability becomes important in other domains like medicine and risk management. Genetic programming, here, can thus provide decision makers a useful alternative to neural networks. It is also to be noted that the genetic programming based functions here were limited by the basic arithmetic operators in the function set considered ; performance can be significantly improved by the inclusion of other functions like log( ), exp( ), etc. in the function set.

Amongst the non-linear lea rning techniques examined, C4.5 shows similar overall reliability with neural networks and genetic programming but superior accuracy in both training and prediction, and makes a good candidate for application to discrimination tasks. Its high performance on the training data can, however, be deceptive, specially in the homogeneous variance case where its reliability is the lowest of all techniques. C4.5 proves advantageous with increasing variance heterogeneity.

A potential factor in the high performance of C4.5 here is the presence of only two classification groups. Shavlik et al. (1991) report that ID3 performs better than backpropagation neural networks when the data contains fewer classes or groups. With a large number of groups, there are fewer examples per group and ID3 tends to build decision trees with small disjuncts (leaf nodes with fewer examples), resulting in high

prediction error rates. The number of classification groups, together with the number of attributes or prediction variables, indicates a measure of task complexity (Subramanian, Hung, & Hu, 1993), and presents another consideration in the choice of discriminant technique. Subramanian et al. (1993) report superior performance of neural networks over linear and quadratic discriminant analysis with increasing problem complexity, in terms of both the number of classes and attributes. The performance of genetic learning (GA and GP) with varying problem complexity has not been addressed in the literature and calls for further research.

An issue of contention with C4.5 has been whether its performance deteriorates with numeric-valued attributes, and stems from a number of experimental results comparing ID3 and neural networks. While Atlas et al. (1990) and Shavlik et al. (1991) find backpropagation better with numeric attributes, Weiss and Kapouleas (1989) find that this is not generally true. Our results support the latter finding and indicate that numerical attributes do not place ID3 in any disadvantage. The association of ID3 with categorical variables stems from the original algorithm (Quinlan, 1986) not having explicit numeric-value handing means ; C4.5's incorporated methods for handling numeric-valued attributes seem to have overcome this seeming limitation.

The distribution of error classifications between the groups is an issue of concern in many real-world applications. While such misclassification costs and prior probabilities have been incorporated with neural network classifiers (Tam & Kiang, 1992), similar methods are not obvious for C4.5. Genetic algorithm and genetic programming, being

driven by task-defined fitness functions, also in general allow consideration of differential costs associated with the two groups, resulting in discriminants tuned to the task at hand.

As indicated earlier, a further desirable feature in a technique is minimizing the number of attributes required for discrimination. Koehler (1991) adds a minimum-attribute criteria to the genetic algorithm fitness function and shows that genetic algorithms can learn reduced-dimension discriminants without sacrificing classification accuracy. Genetic programming, given its similar fitness driven search, is potentially amenable to such an approach, and its effectiveness at learning discriminant functions of lower dimensionality needs further investigation. The entropy selection criteria used in ID3 also implicitly seeks to reduce the number of nodes to a leaf decision node. For statistical and neural network techniques, the data may need to be preprocessed for dimensionality reduction.

Finally, we note that the experimental data does not fully reflect real data, and the observed behavior of the techniques should thus be interpreted only within the context of simulated conditions of the study. This study considers only two of a set of data characteristics of concern; other potential attributes of interest are discussed in the following section. Also, all the learning techniques used here can be refined and their parameters tuned to the data being classified. No such fine-tuning was performed for our experiments, and conventionally accepted or default parameter settings were used so that all techniques are considered on an equitable basis.

Future Research

This paper has presented a comparison of learning techniques for two-group discrimination based on data distribution characteristics. While a number of empirical studies using various discrimination and classification techniques have been reported in the literature, most comparisons of machine learning techniques are based on real data. Our study examines how learning techniques perform on simulated data that varies with respect to distribution kurtosis and variance heterogeneity amongst the two groups. The use of simulated data allows for stricter control and experimentation under known conditions. Such problem generators, allowing for sensitivity analyses, are thus better suited to examine how varying data characteristics relate to different techniques. Using data that has been widely reported in comparing traditional discrimination techniques, this study provides a useful basis for comparing machine learning with traditional methods. It also helps establish certain guidelines on choice of technique to suit problem data characteristics.

As mentioned, only two of a set of data attributes of potential consideration have been examined. In line with future research directions described in (Joachimstheler & Stam, 1988), the performance of learning techniques with data from nonsymmetric populations, and with data having intercorrelated variables should be investigated. Problem complexity (Subramanian, et al., 1993) represents another factor for future research consideration, and the effectiveness of the learning techniques with larger number of predictor variables and classification categories needs systematic evaluation. The training sample size is an important consideration in inductive methods, and has not been fully addressed in the literature, especially for the evolutionary techniques. The sensitivity

of the different techniques to small available training samples, and their performance with increasing training data needs investigation. As noted earlier, parsimony of solutions is often a desirable feature, and Koehler (1991) describes how a GA fitness function can incorporate such considerations; along similar lines, the effectiveness of genetic programming for learning reduced dimensionality discriminants needs further investigation. Further research is also required to examine the efficacy of incorporating classification costs in different learning procedures, along the lines of Tam and Kiang (1992).

References

Abad, P.L. & W.J. Banks (1993). New LP Based Heuristics for the Classification Problem. *European Journal of Operations Research.* 67, 88-100.

Allen. F. & R. Karjalainen (1993). Using Genetic Algorithms to find Technical Trading Rules. The Rodney White Center for Financial Research. The Wharton School, University of Pennsylvania.

Altman, E.L., R.A. Eisenbeis & J. Sinkey (1981). *Application of Classification Techniques in Business, Banking and Finance*. Greenwich, CT: JAI Press.

Atlas, L., R. Cole, J. Connor, M. El-Sharkawi, R.J. Marks II, Y. Muthusamy & E. Barnard (1990). Performance Comparisons between Backpropagation Networks and Classification Trees on Three Real-World Applications. *Advances in Neural Information Processing Systems*, 2, Denver, CO.

Berry, R. & D. Trigueros (1993). Applying Neural Networks to the Extraction of Knowledge from Accounting Reports: A Classification Study. In *Neural Networks in Finance and Investing*, R.R. Trippi & E. Turban (Eds.), Chicago,IL: Probus.

Bhattacharyya, S. (1996). Genetic Algorithms for Direct Marketing. *Proceedings of the 18th National Conference, National Center for Database Marketing*. Orlando, FL.

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

Chung, H.M. & M.S. Silver (1992). Rule-Based Expert Systems and Linear Models: An Empirical Comparison of Learning-By-Examples Methods. *Decision Sciences*, 23, 687-707.

Clark, P. & T. Niblett (1989). The CN2 Induction Algorithm. *Machine Learning*, 3(4), 261-283.

Cronan T.P., L.W. Gorefield & L.G. Perry (1991). Production System Development for Expert Systems Using a Recursive Partitioning Inductive Approach: An Approach to Mortgage, Commercial, and Consumer Lending. *Decision Sciences*, 22, 812-845.

David Shepard Associates (1995). *The New Direct Marketing*. Irwin.

Davis, L. (1991). *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold.

Duda, R.O. & P.E. Hart (1973). *Pattern Classification and Scene Analysis*. New York: John Wiley and Sons.

Ehrenfeucht, A., D. Haussler, M. Kearns & L. Valiant (1989). A General Lower Bound on the Number of Examples needed for Learning. *Information and Computation*, 247-261.

Freed, N. & F. Glover (1981). A Linear Programming Approach to the Discriminant Problem. *Decision Sciences*, 12, 68-74.

Fisher, R.A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7, 179-188.

Fisher, D.H. & K.B. McKusick (1989). An Empirical Comparison of ID3 and Back-propagation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 788-793, Detroit, MI, San Mateo, CA: Morgan Kaufmann.

Furness, P. (1995). Neural Networks for Data-driven Marketing. In *Intelligent Systems for Finance and Business*, S. Goonatilake & P. Treleaven (Eds.), New York: John Wiley and Sons.

D.E. Goldberg (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Greene, D.P. & S.F. Smith (1987). A Genetic System for Learning Models of Consumer Choice. *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms,* J.J. Grefenstette (Ed.), Hillsdale, NJ: L. Erlbaum Associates

Goonatilke, S. (1995). Intelligent Systems for Finance and Business: An Overview. In *Intelligent Systems for Finance and Business,* S. Goonatilake & P. Treleaven (Eds.), New York: John Wiley and Sons.

Hand, D.J. (1981). *Discrimination and Classification*. New York: John Wiley and Sons.

Hansen, J.V, J.B. McDOnald & J.D. Stice (1992). Artificial Intelligence and Generalized Qualitative-Response Models: An Empirical Test on Two Audit Decision-Making Domains. *Decision Sciences*, 23, 708-723.

Haussler, D. (1988). Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. *Artificial Intelligence*, 36(2), 177-222.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor,MI: University of Michigan Press,.

Joachimsthaler, E.A. & A. Stam (1988). Four Approaches to the Classification Problem in Discriminant Analysis: An Experimental Study. *Decision Sciences*, 19, 322-333.

King, R.D., R. Henry, C. Feng & A. Sutherland (1994). A Comparative Study of Classification Algorithms: Statistical, Machine Learning and Neural Network.

Machine Intelligence, 13: Machine Intelligence and Inductive Learning, K. Furukawa, D. Michie and S. Muggleton (Eds.), Oxford, UK: Clarendon Press.

Koehler, G.J. (1991). Linear Discriminant Functions Determined through Genetic Search. *ORSA Journal on Computing*, 3(4), 345-357.

Koehler, G.J. & S.S. Erenguc (1990). Minimizing Misclassifications in Linear Discriminant Analysis. *Decision Sciences*, 21, 63-85.

Koza, J.R. (1992). *Genetic Programming:On the Programming of Computers by Means of Natural Selection*. MIT Press.

Liang, T.P., J.S. Chandler, I. Han & J. Roan (1992). An Empirical Investigation of Some Data Effects on the Classification Accuracy of Probit, ID3 and Neural Networks. *Contemporary Accounting Research*, 9(1), 306-328.

Lippmann, R.P.(1987). An Introduction to Computing with Neural Networks. *IEEE ASSP Magazine*, April, 4-22.

Lopez de Mantraras, R (1991). A Distance-Based Attribute Selection Measure for Decision Tree Induction. Technical Note, *Machine Learning*, (6), 81-92.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd Edition. Springer-Verlag.

Mingers, J. (1989). An Empirical Comparison of Selection Measures for Decision-Tree Induction. *Machine Learning*, (3), 319-342.

Patuwo, E., M.Y. Hu, & M.S. Hung (1993). Two-Group Classification Using Neural Networks. *Decision Sciences*, 24 (4), 825-845.

Piatetsky-Shapiro, G. & W. Frawley (Eds.) (1991). *Knowledge Discovery in Databases*. Menlo Park, CA: AAAI Press/MIT Press.

Quinlan, J.R. (1993). Comparing Connectionist and Symbolic Learning Methods. In , S. Hanson, G. Drastal & R.Rivest (Eds.), *Computational Learning Theory and Symbolic Learning Methods*. Cambridge, MA: MIT Press.

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

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

Radcliffe, N.J. & P.D. Surry (1995). Fundamental Limitations on Seach Algorithms: Evolutionary Computing in Perspective. In Jan van Leeuwen (Ed.), *Computer Science Today: Recent Trends and Developments: Lecture Notes in Computer Science, Volume 1000*. Springer-Verlag.

Raghupathi, W., L.L.Schkade & B.S. Raju (1991). A Neural Network Approach to Bankruptcy Prediction. In *Proceedings of the IEEE 24th Annual Hawaii International Conference on System Sciences*.

Refenes, A.N., A.D. Zapranis, J.T. Connor & D.W. Bunn (1995). Neural Networks in Investment Management. In *Intelligent Systems for Finance and Business*, S. Goonatilake & P. Treleaven (Eds.). New York: John Wiley and Sons.

Rosenblatt, F. (1962). *Principle of Neurodynamics*. New York: Spartan.

Rumelhart, D.E., G.E. Hinton & R.J. Williams (1986). Learning Internal Representations by Error Propagation. In D.E. Rumelhart & J.L. McClelland (Ed.s.), *Parallel*

Distributed Processing: Exploration in the Microstructure of Cognition, Volume 1: Foundations, Cambridge. MA: MIT Press.

Ruck, D.W, S.K. Rogers, M. Kabrisky, & M.E. Oxley (1990). The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function. *IEEE Transactions on Neural Networks*, 1(4), 296-298.

Schaffer, C. (1994). A Conservation Law for Generalization Performance. *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, 259-265.

Syswerda G. (1989). Uniform Crossover in Genetic Algorithms. *Proceedings of the third International Conference on Genetic Algorithms*, 2-9.

Shavlik, J.W., R.J. Mooney & G.G. Towell (1991). Symbolic and Neural Learning Algorithms: An Experimental Comparison. *Machine Learning*, 6, 111-143.

Subramanian, V., M.S. Hung & M.Y Hu (1993). An Experimental Evaluation of Neural Networks for Classification. *Computers and Operations Research*, 20(7), 769-782.

Tam, K.Y & M.L. Kiang (1992). Managerial Applications of Neural Networks: The Case of Bank Failure Predictions. *Management Science*, 38 (7), 926-946.

Tsai, L. & G.J. Koehler (1993). The Accuracy of Concepts Learned from Induction. *Decision Support Systems*.

Walker, R.F., E.W. Haasdijk & M.C. Gerrets (1995). Credit Evaluation Using a Genetic Algorithm. In *Intelligent Systems for Finance and Business*, S. Goonatilake & P. Treleaven (Eds.). New York: John Wiley and Sons.

Weiss, S.M. & I. Kapouleas (1989). An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. *Proceedings of the*

Eleventh International Joint Conference on Artificial Intelligence, 688-693, Detroit, MI. San Mateo, CA: Morgan Kaufmann.

Yoon, Y. & G. Swales (1991). Predicting Stock Price Performance: A Neural Network Approach. In *Proceedings of the IEEE 24th Annual Hawaii International Conference of System Sciences*, 156-162.

Wolpert, D.H. & W.G. Macready (1995). No Free Lunch Theorems for Search. Santa Fe Institute Technical Report No. SFI-TR-95-02-010.

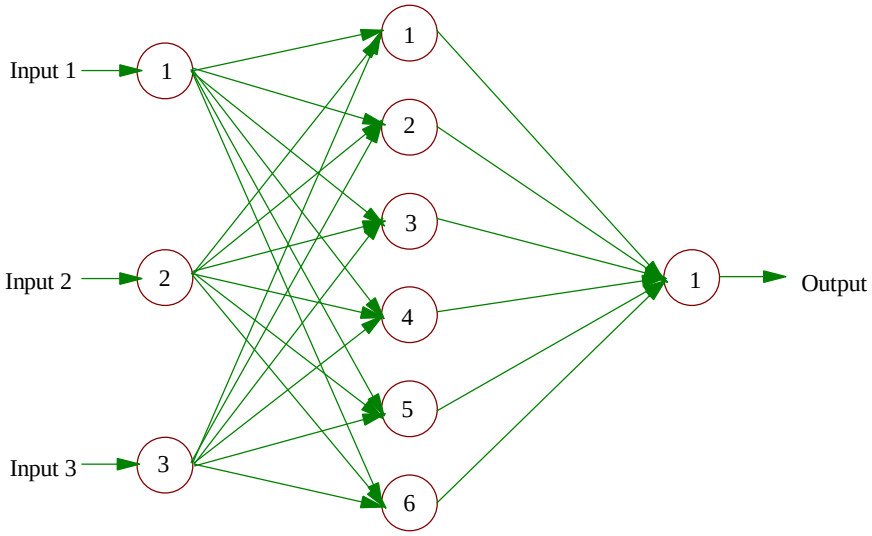

Figure 1: The Three Layer Network used for Discrimination

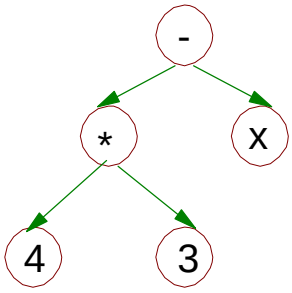

Figure 2a: The Parse Tree Representation of Function f(x)= 4*3 - x

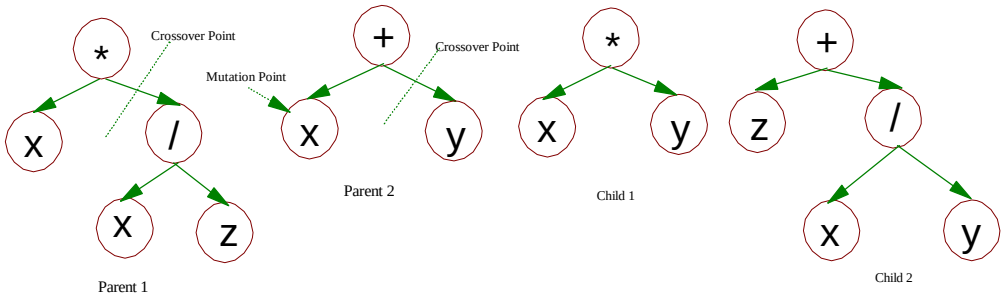

Figure 2b: A Simple Crossover and Mutation in Genetic Programming

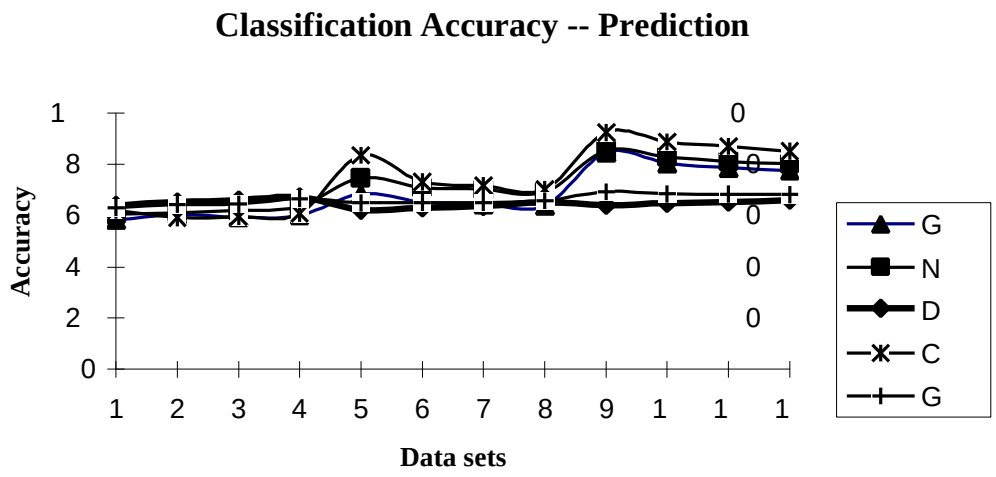

Figure 3a

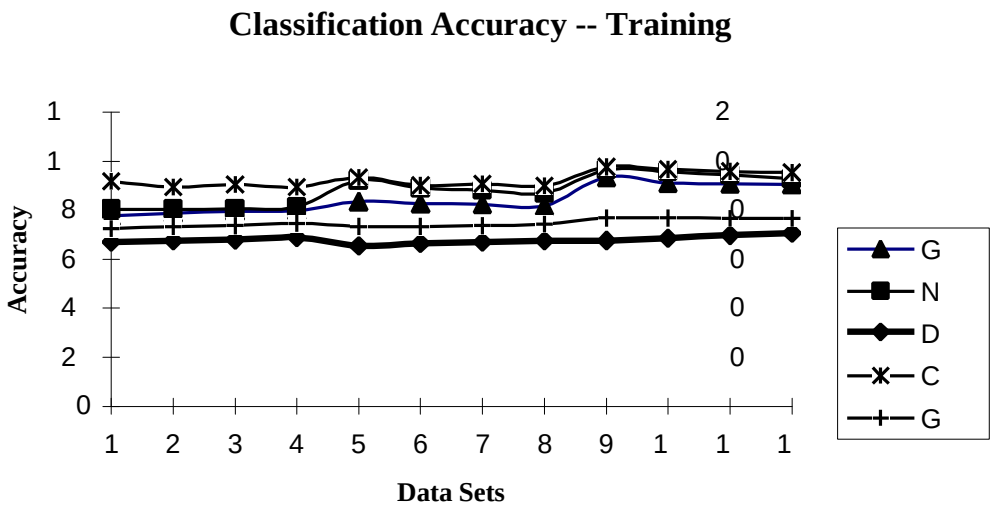

Figure 3b

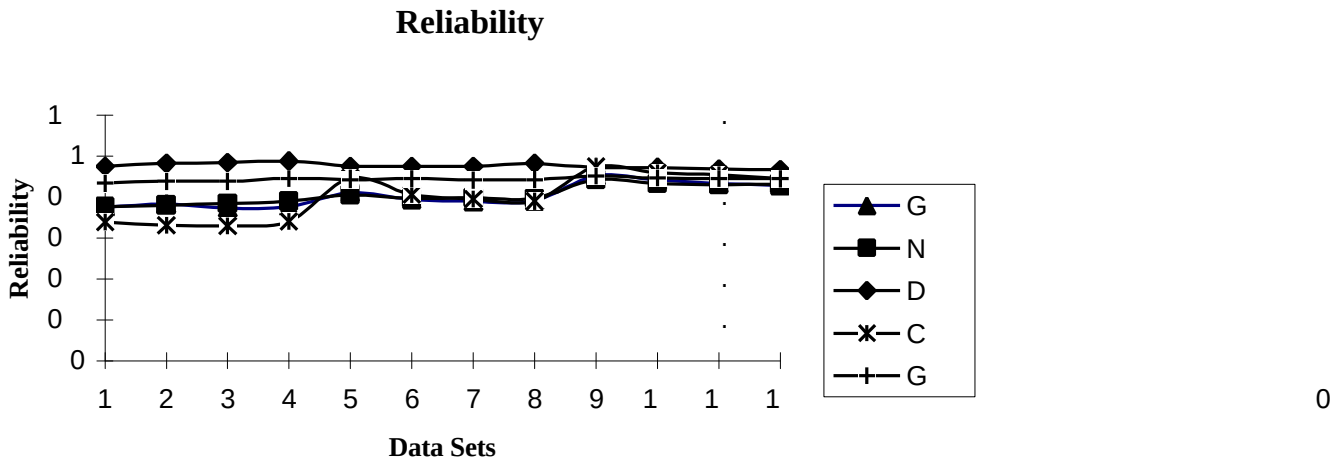
Reliability
Reliability
1
1
0
0
0
0
1 2 3 4 5 6 7 8 9 1 1 1
0
Data Sets
G
N
D
C
G

Figure 3c

Table 1:The Results of Training of the Different Techniques Across Data Sets

Variance Heterogeneity 1	2	Group Mean 1	2	Distribution Kurtosis	Genetic Programming	Neural Network	Discriminant Analysis	C4.5	Genetic Algorithm
1	1	0	0.5	-1	77.71	80.50	67.19	91.80	72.56
				0	78.73	80.41	67.69	89.50	73.32
				1	79.57	80.60	68.04	90.40	73.70
				3	79.86	81.48	68.87	89.42	74.62
1	2	0	0.6	-1	83.58	92.30	65.57	93.46	73.30
				0	82.79	89.20	66.47	90.12	73.32
				1	82.39	88.18	66.92	90.72	73.68
				3	81.67	87.08	67.54	90.04	74.30
1	4	0	0.8	-1	93.57	96.57	67.67	97.62	76.82
				0	91.13	95.73	68.59	96.68	76.72
				1	90.84	94.46	69.81	95.92	76.66
				3	90.61	92.95	70.74	95.36	76.68

Table 2: The Results of Prediction of the Different Techniques Across Data Sets

Variance Heterogeneity 1	2	Group Mean 1	2	Distribution Kurtosis	Genetic Programming	Neural Network	Discriminant Analysis	C4.5	Genetic Algorithm
1	1	0	0.5	-1	58.34	60.60	63.90	62.14	63.02
				0	60.42	61.20	65.34	59.20	64.36
				1	59.40	61.94	65.92	59.64	64.60
				3	60.14	63.64	67.24	60.90	66.58
1	2	0	0.6	-1	68.60	74.76	62.30	83.56	65.00
				0	65.30	70.88	63.02	73.12	65.20
				1	64.38	70.32	63.66	71.74	65.18
				3	63.96	69.18	65.14	70.34	65.78
1	4	0	0.8	-1	84.72	85.54	63.98	92.54	69.40
				0	80.50	82.86	64.78	88.78	68.48
				1	78.68	81.14	65.40	87.06	68.40
				3	77.66	80.26	66.16	85.08	68.22

Table 3: The Results of Reliability of the Different Techniques Across Data Sets

Variance Heterogeneity 1	2	Group Mean 1	2	Distribution Kurtosis	Genetic Programming	Neural Network	Discriminant Analysis	C4.5	Genetic Algorithm
1	1	0	0.5	-1	0.751	0.753	0.951	0.677	0.868
				0	0.767	0.761	0.965	0.661	0.878
				1	0.747	0.768	0.968	0.660	0.877
				3	0.753	0.781	0.976	0.681	0.892
1	2	0	0.6	-1	0.821	0.810	0.950	0.894	0.886
				0	0.789	0.795	0.948	0.811	0.889
				1	0.781	0.797	0.951	0.791	0.884
				3	0.783	0.794	0.964	0.781	0.885
1	4	0	0.8	-1	0.905	0.886	0.945	0.948	0.903
				0	0.883	0.866	0.944	0.918	0.893
				1	0.866	0.859	0.937	0.908	0.892
				3	0.857	0.863	0.935	0.892	0.889

Tables 4: ANOVA and Pairwise Comparisons for Accuracy onTraining Data

Table 4a: The Correct Classification ANOVA Summary Table for Training Data

Source	Sum of Squares	DF	Mean Sq.	F Ratio	P>F
Main Effect					
Distribution (D)	84.39	3	28.13	0.95	0.4174
Variance (V)	50602.97	2	25301.49	850.83	0.0001
Technique (T)	474015.18	4	118503.80	3985.01	0.0001
2 way interaction					
D x T	2323.43	12	193.62	6.51	0.0001
D x V	775.82	6	129.30	4.35	0.0002
T x V	29551.19	8	3693.89	124.22	0.0001
3 way interaction					
D x T x V	2520.77	24	105.03	3.53	0.0001

Table 4b: The Overall Pairwise Comparisons on Training Data Sets

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	222296.06	222296.06	9156.68	0.0001
NN vs. GP	1	9484.35	9484.35	318.94	0.0001
NN vs. DA	1	248799.21	248799.21	8366.54	0.0001
GP vs. DA	1	161130.10	161130.10	5418.43	0.0001
GA vs. DA	1	26334.38	26334.38	885.56	0.0001
NN vs. C4.5	1	7895.25	7895.25	265.50	0.0001
GP vs. C4.5	1	34686.41	34686.41	1166.42	0.0001
NN vs. GA	1	113245.10	113245.10	3808.17	0.0001
GP vs. GA	1	57183.84	57183.84	1922.96	0.0001
Nonlin. vs. Lin.	1	412970.13	412970.13	13887.23	0.0001

Table 4c: The Pairwise Comparisons based on Variance for Training Data Sets

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	7339.85	7339.85	246.82	0.0001
NN vs. GP	1	1513.13	1513.13	50.88	0.0001
NN vs. DA	1	17161.20	17161.20	577.09	0.0001
GP vs. DA	1	8482.74	8482.74	285.25	0.0001
GA vs. DA	1	470.63	470.63	15.83	0.0001

NN vs. C4.5	1	10998.90	10998.90	369.87	
0.0001					
GP vs. C4.5	1	4352.93	4352.93	146.38	
0.0001					
NN vs. GA	1	11947.98	11947.98	401.78	
0.0001					
GP vs. GA	1	4957.27	4957.27	166.70	
0.0001					
Nonlin. vs. Lin.	1	12520.84	12520.84	421.05	
0.0001					

Table 4d: The Pairwise Comparisons based on Kurtosis on Training Data Sets

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	819.20	819.20	27.55	
0.0001					
NN vs. GP	1	239.44	239.44	8.05	
0.0046					
NN vs. DA	1	1369.39	1369.39	46.05	
0.0001					
GP vs. DA	1	463.60	463.60	15.59	
0.0001					
GA vs. DA	1	78.96	78.96	2.66	
0.1033					
NN vs. C4.5	1	193.71	193.71	6.51	
0.0107					
GP vs. C4.5	1	2.42	2.42	0.08	
0.7754					
NN vs. GA	1	790.69	790.69	26.59	
0.0001					
GP vs. GA	1	159.91	159.91	5.38	
0.0204					
Nonlin. vs. Lin.	1	1243.78	1243.78	41.83	
0.0001					

Tables 5: ANOVA and Pairwise Comparisons for Accuracy on Holdout Data

Table 5a: The Correct Classification ANOVA Summary Table for Holdout Data

Source	Sum of Squares	DF	Mean Sq.	F Ratio	P>F
Main Effect					
Distribution (D)	2026.91	3	675.64	25.17	
0.0001					
Variance (V)	107282.70	2	53641.35	1998.30	
0.0001					

Technique (T)	38923.73	4	9730.93	362.51
0.0001				
2 way interaction				
D x T	5366.31	12	447.19	16.66
0.0001				
D x V	3201.33	6	533.55	19.88
0.0001				
T x V	60666.40	8	7583.30	282.50
0.0001				
3 way interaction				
D x T x V	1819.77	24	75.82	2.82
0.0001				

Table 5b: The Overall Pairwise Comparisons on Holdout Data Sets

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	14647.88	14647.88	545.68	
0.0001					
NN vs. GP	1	3474.80	3474.80	129.45	
0.0001					
NN vs. DA	1	15293.88	15293.88	569.74	
0.0001					
GP vs. DA	1	4188.80	4188.80	156.05	
0.0001					
GA vs. DA	1	629.30	629.30	23.44	
0.0001					
NN vs. C4.5	1	2077.70	2077.70	77.40	
0.0001					
GP vs. C4.5	1	10926.37	10926.37	407.04	
0.0001					
NN vs. GA	1	9718.52	9718.52	362.04	
0.0001					
GP vs. GA	1	1570.94	1570.94	58.52	
0.0001					
Nonlin. vs. Lin.	1	27308.51	27308.51	1017.32	
0.0001					

Table 5c: The Pairwise Comparisons based on Variance for Holdout Data Sets

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P >
F					
Mach. learn. vs. Stat.	1	21570.79	21570.79	803.58	
0.0001					
NN vs. GP	1	174.42	174.42	6.50	
0.0109					
NN vs. DA	1	17642.10	17642.10	657.22	
0.0001					
GP vs. DA	1	14308.17	14308.17	533.02	

0.0001					
GA vs. DA	1	870.01	870.01	32.41	
0.0001					
NN vs. C4.5	1	2468.48	2468.48	91.96	
0.0001					
GP vs. C4.5	1	3955.23	3955.43	147.34	
0.0001					
NN vs. GA	1	10676.60	10676.60	397.74	
0.0001					
GP vs. GA	1	8121.76	8121.76	302.56	
0.0001					
Nonlin. vs. Lin.	1	40706.53	40706.53	1516.44	
0.0001					

Table 5d: The Pairwise Comparisons based on Kurtosis on Training Data Sets

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	1886.96	1886.96	70.29	
0.0001					
NN vs. GP	1	10.45	10.45	0.39	
0.5326					
NN vs. DA	1	961.00	961.00	35.80	
0.0001					
GP vs. DA	1	1171.92	1171.92	43.66	
0.0001					
GA vs. DA	1	93.12	93.12	3.47	
0.0626					
NN vs. C4.5	1	991.20	991.20	36.93	
0.0001					
GP vs. C4.5	1	798.06	798.06	29.73	
0.0001					
NN vs. GA	1	455.82	455.82	16.98	
0.0001					
GP vs. GA	1	604.34	604.34	22.51	
0.0001					
Nonlin. vs. Lin.	1	3419.65	3419.65	127.39	
0.0001					

Tables 6: ANOVA and Pairwise Comparisons for Reliability

Table 6a: The ANOVA Summary Table for Reliability

Source	Sum of Squares	DF	Mean Sq.	F Ratio	P>F
Main Effect					
Distribution (D)	0.09	3	0.03	4.85	0.0023
Variance (V)	4.33	2	2.16	354.35	
0.0001					
Technique (T)	10.96	4	2.73	448.11	
0.0001					
2 way interaction					
D x T	0.28	12	0.02	3.87	
0.0001					

D x V	0.20	6	0.03	5.35	
0.0001					
T x V	4.43	8	0.55	90.66	
0.0001					
3 way interaction					
D x T x V	0.12	24	0.01	0.83	
0.7025					

Table 6b: The Overall Pairwise Comparisons for Reliability

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	8.08	8.08	1322.33	0.0001
NN vs. GP	1	0.00	0.00	0.00	0.9718
NN vs. DA	1	6.45	6.45	1055.66	0.0001
GP vs. DA	1	6.43	6.43	1053.37	0.0001
GA vs. DA	1	1.48	1.48	241.75	0.0001
NN vs. C4.5	1	0.02	0.02	4.03	0.0448
GP vs. C4.5	1	0.02	0.03	4.17	0.0412
NN vs. GA	1	1.75	1.75	287.05	0.0001
GP vs. GA	1	1.74	1.74	285.85	0.0001
Nonlin. vs. Lin.	1	9.45	9.45	1545.21	0.0001

Table 6c: The Pairwise Comparisons based on Variance for Reliability

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	1.17	1.17	191.24	
0.0001					
NN vs. GP	1	0.03	0.03	4.61	
0.0318					
NN vs. DA	1	0.42	0.42	68.62	
0.0001					
GP vs. DA	1	0.67	0.67	108.81	
0.0001					
GA vs. DA	1	0.05	0.05	7.53	
0.0061					
NN vs. C4.5	1	1.19	1.19	195.71	
0.0001					
GP vs. C4.5	1	0.86	0.86	140.23	
0.0001					
NN vs. GA	1	0.19	0.19	30.69	
0.0001					
GP vs. GA	1	0.36	0.36	59.10	
0.0001					
Nonlin. vs. Lin.	1	2.21	2.21	362.56	
0.0001					

Table 6d: The Pairwise Comparisons based on Kurtosis for Reliability

Contrast	DF	Contrast Sum of Sq.	Mean Sq.	F-Ratio	P > F
Mach. learn. vs. Stat.	1	0.08	0.08	13.92	0.0002
NN vs. GP	1	0.01	0.01	1.11	0.2923
NN vs. DA	1	0.02	0.02	3.39	0.0657
GP vs. DA	1	0.05	0.05	8.38	0.0038
GA vs. DA	1	0.01	0.01	1.27	0.2608
NN vs. C4.5	1	0.10	0.10	16.78	0.0001
GP vs. C4.5	1	0.06	0.06	9.26	0.0024
NN vs. GA	1	0.00	0.00	0.51	0.4738
GP vs. GA	1	0.02	0.02	3.13	0.0769
Nonlin. vs. Lin.	1	0.13	0.13	21.53	0.0001