

**COMPUTER SYSTEMS THAT LEARN:
AN EMPIRICAL STUDY OF THE EFFECT OF NOISE ON THE
PERFORMANCE OF THREE CLASSIFICATION METHODS**

James R. Nolan

Department of Quantitative Business and Computer Science

Siena College

515 Loudon Road

Loudonville, NY 12211

USA

Phone: (518) 783-2503

FAX: (518) 783-2528

Email: jnolan@siena.edu

**COMPUTER SYSTEMS THAT LEARN:
AN EMPIRICAL STUDY OF THE EFFECT OF NOISE ON THE
PERFORMANCE OF THREE CLASSIFICATION METHODS.**

ABSTRACT

Classification learning systems are useful in many domain areas. One problem with the development of these systems is feature noise. Learning from examples classification methods from statistical pattern recognition, machine learning, and connectionist theory are applied to synthetic data sets possessing a known percentage of feature noise. Linear discriminant analysis, the C5.0 tree classification algorithm, and a backpropagation neural network tool are used as representative techniques from these three categories. K-fold cross validation is used to estimate the sensitivity of the true classification accuracy to level of feature noise present in the data sets. Results indicate that the backpropagation neural network outperforms both linear discriminant analysis and C5.0 tree classification when appreciable (10% or more of the cases) feature noise is present. These results are confirmed when the same type of empirical analysis is applied to a real-world data set previously analyzed and reported in the statistical and machine learning literature.

1. Introduction

Many decision-making tasks fall into the general category of classification (Weiss & Kapouleas, 1989). Examples abound in several areas of expertise (Weiss & Kulikowski, 1991). Physicians, for instance, are always searching for the best test to make a particular diagnosis. Investors must decide to buy or sell a stock based on information about the company and its economic prospects. Banks and other credit institutions must approve or disapprove credit applications based on personal finance information.

A primary focus of study in both the statistical and machine learning communities has been on methods of *learning from examples*: a system accepts case descriptions (e.g., patient care histories) that are preclassified (e.g., breast cancer) (Fisher & McKusick, 1989). These case descriptions contain features (sometimes referred to as attributes or factors) which are thought to be important in classifying a case. Based on training, the system forms a knowledge-base that can accurately classify new cases.

Empirical learning techniques for classification span roughly three categories: statistical pattern recognition (Duda & Hart, 1973), machine learning techniques for induction of decision trees or production rules (Quinlan, 1993; 1986), and connectionist (McClelland & Rumelhart, 1988). While a method for any one of these categories is usually applicable to the same problem, the categories of procedures can differ radically in their underlying models, ability to generalize, and performance when feature noise is present in the example data.

This paper focuses on the sensitivity of prediction accuracy to the level of feature noise (description errors). Specifically, three pattern classification methods, one from each category of empirical learning techniques, will be compared as to their ability to generalize, given the presence of varying levels of noise in the set of training and test cases. This is consistent with

work on the theoretical basis of learning which assumes that the distribution of examples is the same during training and testing (Valiant, 1984). While the effect of feature noise is always to degrade performance of the classification rule, all methods will not exhibit the same amount of degradation. The objective of this study is to determine which of these methods is best able to generalize given the presence of feature noise in the domain.

1.1 Previous Work

Several researchers have done comparative studies of learning from examples algorithms. Shavlik et al. studied the difference between the ID3 inductive learning algorithm and perceptron and backpropagation neural learning (Shavlik et al., 1991). They used five real-world data sets and examined performance when noise was added to the data sets. Their results suggested that backpropagation slightly outperforms ID3. The work described in the present study differs from Shavlik et al.'s work in several important ways. First of all, Shavlik et al. used ID3 as their inductive learning algorithm. This paper describes results using the C5.0 algorithm, which, while descended from ID3, is said to perform better because of new tree pruning algorithms and the addition of a x -trial adaptive boosting option (Quinlan, 1997, 1991). Secondly, this work uses both synthetic and real-world data sets as opposed to Shavlik et al.'s use of real-world data sets only. The purpose of the synthetic data sets is to more carefully control the measurement and introduction of noise. Verifying the results of the synthetic data set analysis on a real world data set with known level of noise is also a point of this study. Finally, the introduction of noise is done differently. Shavlik et al. change a given percentage of *all* feature values, while this study changes a random number of feature values in a given percentage of cases.

Weiss and Kapouleas' work compares classification methods from statistical pattern recognition, machine learning, and neural nets, as does this study (Weiss & Kapouleas, 1989). As

with Shavlik et al., Weiss & Kapouleas look at several real-world data sets. However, they do not introduce noise in any systematic fashion other than the noise already present in the data sets.

Weiss and Kapouleas concluded that the tree induction algorithm outperformed neural nets based not only on error rates but also amount of CPU time spent developing the model.

Fisher & McKusick studied the differences between ID3 and backpropagation neural nets using both real-world and synthetic data sets (Fisher & McKusick, 1989). The effect of noise was also explored by Fisher & McKusick. Their experiments indicated that backpropagation neural nets achieve higher accuracy levels under noisy conditions but required considerably more case presentations (training time). The differences in accuracy were not significant. As previously cited, the present paper uses the C5.0 tree induction algorithm which allows for a better method for tree pruning and supports adaptive boosting, a procedure designed to improve its classification accuracy.

Finally, Datta & Kibler compare the C4.5 tree classification algorithm to various minimum distance statistical classifiers using several real-world data sets (Data & Kibler, 1997). They conclude that an improved minimum distance classifier algorithm achieves higher average generalization accuracy than C4.5.

Each of the cited works has made a contribution to our knowledge about the comparative classification ability of learning from examples algorithms. This paper hopes to do the same by updating and adding to the knowledge gained from past studies.

The next section describes the three classification methodologies to be compared. The design of the empirical study and the experimental results are presented next. These results are discussed and conclusions drawn.

2. Review of Classification Methodologies

Classification or pattern recognition algorithms are mathematical tools for detecting similarities between members of a collection of cases. The output of a classification algorithm is used to classify new cases into subsets. Classification algorithms may be constructed manually or automatically. While the first approach reflects intuitive knowledge about underlying structures and relations or influences of features, the latter approach is applied to large data sets where unknown structures and a variety of interdependencies are to be considered. Because it is a difficult task to exhaustively try all combinations of features and determine their individual influence on the classification algorithm, techniques have been developed that solve these problems automatically. Such construction techniques are based on concepts of learning from examples (Michalski & Chilausky, 1980; Fisher & McKusick, 1989).

Learning from examples tasks have to deal with problems related to feature selection criteria, such as mixtures of discrete, continuous, and fuzzy features. They also must cope with feature noise in the form of description errors. Many classification methodologies have been developed to deal with these problems. Widely used, classification methodologies include linear discriminant analysis (from the statistical pattern recognition community), classification trees and production rules (from the machine learning community), and backpropagation neural networks (from the connectionist community).

A brief description of each technique follows.

2.1 Linear Discriminant Analysis

Linear discriminant analysis is a technique used to identify structures and possible organizations of the data into meaningful groups (Dillon & Goldstein 1984; Ebert, 1992). Any given set of features related to a case can be considered as a multidimensional space where each case is

represented as a point with distinct coordinates. We identify as a group any subset of the points which is internally well connected and externally poorly connected. The underlying assumption is that cases under investigation may be grouped such that the cases residing in a particular group are in some sense more similar to each other than to cases belonging to other groups.

Typically, the classification consists of two steps. First, factor analysis or a principal-components procedure is used for reducing the dimensionality of the space. Discriminant analysis is then used to separate groups of cases according to a selected feature (Ebert, 1996). Fisher was the first to suggest that classification should be based on a linear combination of the discriminating variables (Fisher, 1936). He proposed using a linear combination which maximizes group differences while minimizing variation within the groups. An adaptation of his proposal leads to a separate linear combination, called a "classification function," for each group (Klecka, 1980). These have the following form:

$$h_k = b_{k0} + b_{k1}X_1 + b_{k2}X_2 + \dots + b_{kp}X_p$$

where h_k is the score for group k , the X 's are feature values, and the b 's are coefficients that need to be derived. A case is classified into the group with the highest score (largest h). The coefficients for these classification functions are derived by the following computation:

$$b_{ki} = (n. - g) \sum_{j=1}^p a_{ij} X_{jk}$$

where b_{ki} is the coefficient for feature i in the equation corresponding to group k , " $n.$ " is the total number of cases over all groups, " g " is the number of groups, and a_{ij} is an element from the

inverse of the within-groups sum of crossproducts matrix. A constant term is also required and

$$b_{k0} = -.5 \sum_{j=1}^p b_{kj} X_{jk}$$

defined as:

We do not usually interpret these classification function coefficients because they are not standardized and there is a different function for each group. The scores also lack intrinsic value because they are arbitrary numbers which have the property that the case resembles most closely that group on which it has the highest score. A more intuitive means of classification is to measure the distances from the individual case to each of the group centroids and classify the case into the closest group. However, when the features are correlated and do not have the same measurement units and standard deviations, the concept of "distance" is not well defined. Mahalanobis proposed a generalized distance measure which solves this problem (Mahalanobis, 1963). It is used in the following form:

$$D^2(X|G_k) = (n - g) \sum_{i=1}^p \sum_{j=1}^p a_{ij} (X_i - X_{ik})(X_j - X_{jk})$$

where $D^2(X | G_k)$ is the squared distance from point X (a specific case) to the centroid of group k. After calculating D^2 for each group, we would classify the case into the group with the smallest D^2 . That group is the one in which the typical profile on the discriminating variables most closely resembles the profile of this case. By classifying a case into the closest group according to D^2 , we are implicitly assigning it to the group for which it has the highest

probability of belonging. If the distance to the closest group is large, the profiles may match rather poorly, but they are a better match than for any other group.

If we assume that every case must belong to one of the groups, we can compute a probability of group membership for each group. The probability that object X is a member of group k is:

$$\Pr(G_k|X) = \frac{\Pr(X|G_k)}{\sum_{i=1}^g \Pr(X|G_i)}$$

Generally, a stepwise procedure is used to select features to be included in the classification equation. A forward stepwise procedure begins by selecting the individual feature which provides the greatest univariate discrimination. The procedure then pairs this first feature with each of the remaining features, one at a time, to locate the combination which produces the greatest discrimination. The feature which contributed to the best pair is selected. The procedure goes on to combine the first two with each of the remaining features to form triplets, etc. This procedure continues until all possible features have been selected or the remaining features do not contribute a sufficient increment in discriminating power.

2.2 Classification Trees

Classification trees have been widely used in many areas including image recognition, speech recognition, and natural language processing. With the development of knowledge acquisition tools and machine learning theory, many inductive machine learning systems have been developed for constructing classification trees from a training set of examples. Several methods for automatic tree generation have been described and used for real projects (Quinlan, 1983; Breiman et al., 1984). They are all based on automatic learning from examples with distinct

approaches for optimizing, controlling, and supervising the learning process (e.g., pattern recognition).

Quinlan reported on a system called ID3 that constructs a decision tree using a top-down, divide-and-conquer approach: for the two group problem, select a feature, divide the training set into subsets characterized by the possible values of the feature, and follow the same procedure recursively with each subset until no subset contains cases from both groups (Quinlan, 1983). These single group subsets correspond to the leaves of the decision tree and can be labeled with that group. The extension to more than two groups of cases is straightforward.

The method depends for its practicality on making a good choice of feature to test at each stage. ID3 and its successors, C4.5 and C5.0, take an information-theoretic approach as follows: to create a decision tree such that each node corresponds to a feature and each arc corresponds to a possible value of that feature. A leaf of the tree specifies the expected value of the feature for the examples described by the path from the root to that leaf. Each node in the decision tree should be associated with the feature which is most *informative* among the features not yet considered in the path from the root. *Entropy* is used to measure how informative is a node.

In general, if we are given a probability distribution $P = (p_1, p_2, \dots, p_n)$ then the *information* conveyed by the distribution, also called the *entropy* of P , is:

$$I(P) = - [p_1 \log(p_1) + p_2 \log(p_2) + \dots + p_n \log(p_n)]$$

For example, if P is (0.5, 0.5) then $I(P)$ is 1, if P is (0.67, 0.33) then $I(P)$ is 0.92, if P is (1, 0) then $I(P)$ is 0.

If a set T of cases is partitioned into disjoint exhaustive groups G_1, G_2, \dots, G_k on the basis of the value of the categorical feature, then the information needed to identify the group of an

element of T is $\text{Info}(T) = I(P)$, where P is the probability distribution of the partition (G_1, G_2, \dots, G_k) :

$$P = (|G_1|/|T|, |G_2|/|T|, \dots, |G_k|/|T|)$$

If we first partition T on the basis of the value of a non-categorical feature X into sets T_1, T_2, \dots, T_n then the information needed to identify the group of an element of T_i , i.e., the weighted average of $\text{Info}(T_i)$ would be:

$$\text{Info}(X, T) = \text{Sum for } i \text{ from } 1 \text{ to } n \text{ of } (|T_i|/|T|) * \text{Info}(T_i)$$

Consider the quantity $\text{Gain}(X, T)$ defined as

$$\text{Gain}(X, T) = \text{Info}(T) - \text{Info}(X, T)$$

This represents the difference between the information needed to identify an element of T and the information needed to identify an element of T after the value of feature X has been obtained. In other words, this is the *gain* in information due to feature X. We can use this notion of gain to rank features and to build decision trees where at each node is located the feature with greatest gain among the features not yet considered in the path from the root.

The intent of this ordering is two fold: to create small decision trees so that cases can be identified after only a few questions; and to match a hoped for minimality of the process represented by the cases being considered (Quinlan, 1993).

The decision tree built using the training set, because of the way it was built, deals correctly with most of the cases in the training set. In fact, in order to do so, it may become quite complex, with long and very uneven paths. Pruning techniques are generally used to reduce the size of the tree. *Pruning* of the decision tree is done by replacing a whole subtree by a leaf node. The replacement takes place if a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf.

2.3 Neural Network Approaches

Neural networks are adaptive systems that learn from examples. They are networks of simple processing nodes that are usually interconnected. Artificial neural networks serve as general purpose mechanisms for training a machine by example. Many neural network paradigms have been developed and shown to be effective in tasks involving pattern recognition, feature extraction, and machine learning. They can extract regularities and generalize concepts based on self-organizing or training algorithms, and are well suited for problem domains where ample training examples are available.

A popular neural network architecture for pattern recognition and classification is the so called backpropagation model (McClelland & Rumelhart, 1988; Hagan et al., 1996; Zurada, 1992). It has a three-layer, feed-forward structure. In this network, the first layer of neurons is the input layer and the third layer is the output layer. The middle layer is called the hidden layer, because it is the only layer that does not communicate with the external environment either by taking in the external input or sending out the system output. The backpropagation model could have more than one hidden layer, but the hidden layers have a hierarchical structure - a lower level communicates only to its immediate upper level.

Processing in a neural network takes place at the neuron level. The net input to any neuron i is equal to:

$$input_i = \sum_{\substack{\text{all } j \text{ neurons} \\ \text{connected to } i}} w_{ji} output_j + extinput_i + bias_i$$

where w_i is the interconnection weight between neurons i and j , $output_j$ is the output from neuron j being input to neuron i , $extinput_i$ represents any input the neuron receives from the external

environment, and bias_i represents a bias value used in some networks to control the activation of the neuron.

For neuron *i* to send out output, the action potential or the net input should go through a filter or transformation. There are several popular transfer functions including step, signum, sigmoid, hyperbolic tangent, linear, and threshold-linear functions. These functions allow a neuron to fire only after it has reached the threshold value.

Connection weights are the single most important factor in the input and output processes of a neuron. In almost all neural networks, it is the system itself that computes the interconnection weights among the neurons. The process by which the system arrives at the connecting weights is called learning. For a system to learn, it should be given a learning method to change the weights to the ideal values and a domain data set that represents the domain knowledge. The learning method is the most important distinguishing characteristic in various neural networks.

There are two distinct types of learning in neural networks: supervised and unsupervised. Unsupervised learning occurs when the system receives only the input, and no information on the expected classification. In supervised learning, the system is given the classifications and the model determines weights in such a way that once given the inputs, it would produce the desired output. The main idea in supervised neural network learning is that the system is repeatedly given features about various cases, along with the expected classifications. The system uses the learning method to adjust the weights in order to produce classifications close to what are expected. Supervised learning is the preferred method for pattern recognition and classification tasks when examples including features and classifications are available.

3. Complications Caused by Description Errors

All learning from examples techniques are highly sensitive to feature noise. In real-world pattern recognition and classification tasks, the description of a case will often contain errors. Therefore, the ability to tolerate noise is a necessity for robust, practical learning methods. Algorithms should demonstrate graceful degradation in performance when presented with noisy data (Quinlan, 1986; Niblett & Bratko, 1986). Two sources of feature noise in many real-world applications are mistakes made when recording feature values (random noise) and missing feature values (incompletely described examples). Regardless of the source, however, these errors can be expected to affect the formation and use of pattern recognition and classification rules in two ways.

First, learning from examples systems must employ some form of generalization to anticipate unseen cases. As detailed earlier, this is usually accomplished by determining which features best discriminate between groups of cases. These features are then used to form a classification rule. Errors in the description of these cases will tend to confuse any generalization mechanisms of this type. Suppose, for example, that an insurance company wishes to base their insurance risk assessment classification of potential clients on examples from the current insured of the company. These classifications may be difficult to discover if client features are missing or have been erroneously described.

Second, problems will arise independently of the formation of the classification rule when this rule formed from the given cases is used to classify another case. The classification rule is couched in terms of the description of a case, and if this description contains errors, the result given by the classification rule for the case in question might well be incorrect too.

Since all learning from examples techniques are affected by noise, it would be useful to understand more fully the relationship between classification accuracy and level of noise present in the examples. In particular, choosing *a priori* which particular technique holds the most promise when employed in a given classification and pattern matching domain might be possible if we better understood this relationship between classification accuracy and noise.

The experimental design chosen to learn more about the relationship between classification accuracy and noise, including generation of the data sets, statistical analysis, and results, is presented next.

4. Experiments and Results

As explained above, the presence of description errors complicates the pattern recognition and classification task. In real-world tasks of this nature, the description of a case will often contain errors, which we are referring to as noise. This noise can affect the generalization ability of a pattern recognition or classification model. While the model may do well on the training data, it will breakdown when used with new data.

Learning from examples algorithms are commonly tested against real-world data sets. However, another important way to test learning from examples algorithms is to use well understood synthetic data sets (Quinlan, 1986; Niblett & Bratko, 1986). With synthetic data sets, classification accuracy can be determined under controlled circumstances. In particular, one can introduce in a very precise way factors such as noise.

In order to investigate the effect of description errors on both the formation of classification rules and their use in classifying cases, synthetic data sets were generated.

4.1 The Generation of the Synthetic Data Sets

For this study, the task chosen as a measure of a classification model's ability to generalize is pattern recognition on a subset of the alphabet. More specifically, information about the presence or absence of eight geometric figures will be used as features needed to classify a case as one of four letters of the alphabet. While this is a relatively simple task, it will serve to illustrate each of the three classification model's responses to a predefined level of noise. It is important to realize that the chosen application, letter recognition, is used only to give the generated synthetic data sets a real-world meaning. The example could just as well have been classification of mushroom types or wine grape origins.

The data sets were created in the following manner:

1. The capital letters "A", "R", "S", and "Q" are chosen to represent classes. The geometric figures to be used as features are "/", "\", "|", "-", "O", "D", "C, and "U".
2. A noise-free data set was generated. The data set contains 320 cases, 80 each for the four letters of the alphabet used in the letter recognition study. Each of the 80 cases for each letter contains a "1" denoting the presence of a figure needed to form the letter and a "0" for all other figures. For example, "1" 's are used to denote the presence of "/", "\", and "-", respectively, for the letter "A". The positions corresponding to the other figures would be occupied by "0" 's. Table 1 displays the structure of each case for the letters "A", "R", "S", and "Q". The 320 cases in this data file are randomly distributed, i.e., there is no pattern to the occurrence of the cases in the data file.

3. Noise was systematically introduced to the data set created in step 2. The noise was simulated so as to represent noise arising from random mistakes in recording feature values as well as noise arising from imperfections caused by missing feature values. Specifically, 10% of each letter's cases had feature values randomly changed to another legal value for that feature. Legal value is used in the sense that it is changed to a "0" or "1". A uniform distribution bias was used to generate the noise. This process was also used to generate data sets with an additional 10% noise added. Thus, there are nine noisy data sets generated ($i = 1$ to 9) with percentage of noise equal to $10i$.

Table 1

Synthetic Data Set Case Structure

Letter	Geometric Figures							
	/	\		-	O	⊃	⊂	∪
A	1	1	0	1	0	0	0	0
R	0	1	1	0	0	1	0	0
S	0	0	0	0	0	1	1	0
Q	0	1	0	0	1	0	0	0

Note that introducing noise to 10% of each letter's cases does not mean that 10% of all feature values for that letter were changed. It simply means that one or more feature values were changed in 10% of the cases.

4.2 Experimental Design

The experimental design used in the study is based on the k-fold cross validation model (Stone, 1974). In k-fold cross validation, the cases are randomly divided into k mutually exclusive test partitions of approximately equal size. The cases not found in each test partition are independently used for training, and the resulting classification is tested on the corresponding test partition. The average correct classification rate over all k partitions is the cross validated correct classification rate. Previous research has shown that 10-fold cross validation is adequate and accurate (Breiman et al., 1984; Kohavi, 1995).

The result of using the 10-fold cross validation design on the synthetic data sets will be a group of ten average correct classification rates for each of the three learning from examples algorithms. Each number in the lists of ten will correspond to a data set with a predefined level of noise.

The Wilcoxon signed-rank test is used to evaluate information about both the sign of the differences and the magnitude of the differences between pairs of numbers (Levine et al., 1997). In order to use this test to identify any significant differences between the classification algorithms with respect to classification accuracy over the varying levels of noise data sets, signed ranks are generated. Specifically, for each level of noise, the classification accuracy's for the three classification algorithms are ranked from 1 to 3. In the case of ties, the ranks are split, e.g., 1.5 for each of the two algorithms if they tie for the best classification accuracy.

4.3 Experimental Results

Classification models are created using linear discriminant analysis, the C5.0 tree classification generator, and a backpropagation neural network. The canned SPSS for Windows 7.5 program was used to produce the linear discriminant results. The PC version of Quinlan's C5.0 tree

induction program was used to generate the classification results. C5.0's 10-classifier adaptive boosting option was employed.

A standard, three layer, feed forward, hierarchical architecture was used for the neural network model. This backpropagation model has an input layer with eight neurons, one for each geometric figure. The output layer contains four neurons representing the capital letters "A", "R", "S", and "Q", respectively. The activation function used was the sigmoid function. Preliminary training of the net indicated that a total of 20 neurons worked best in the hidden layer. Additionally, the number of epochs was limited to 5,000. This was found to be adequate in similar work performed by Shavlik et al. (Shavlik et al., 1991).

Table 2 contains the classification accuracy results. The table displays the average classification accuracy and the rank of the classification algorithm for that level of noise; the algorithm with the highest accuracy has rank 1. The average accuracy and average rank of each of the algorithms are shown in the last column. The average accuracy gives relative ratings of the algorithms taking into consideration the magnitude of the differences in accuracy while the average rank disregards the magnitude. The backpropagation neural network model has the highest average accuracy over all levels of noise. In fact, it ranks number 1 at all noise levels 10% and above.

Applying the Wilcoxon test to the ranks in Table 2 shows that there are significant differences at the .05 level between the backpropagation neural net and the linear discriminant and tree classification models (see Table 3). There is not, however, a statistically significant difference at the .05 level between the linear discriminant and C5.0 tree classification models.

Table 2

Classification Accuracy and Rankings¹

<u>Model</u>	<u>Noise (%)</u>										Ave.
	0	10	20	30	40	50	60	70	80	90	
Linear Discr.	.98	.96	.96	.96	.96	.94	.94	.92	.92	.91	.95
	(3)	(2.5)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2.15)
C5.0Tree	1.00	.96	.95	.93	.92	.92	.91	.90	.90	.89	.93
	(1.5)	(2.5)	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(2.80)
Backprop.	1.00	.98	.98	.98	.97	.96	.96	.95	.94	.93	.97
	(1.5)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1.05)

¹Rankings appear in parenthesis

Table 3

Statistical Significance Results

Wilcoxon Signed Rank Test¹

	Linear Discr.	C5.0 Tree Cl.	Backprop.
Linear Discriminant	_____		
CS.0 Tree Classification	.083	_____	
Backpropagation	.003	.004	_____

¹Values below .05 indicate statistically significant classification accuracies

4.4 Iris Data Set Analysis

These experiments on synthetic data sets indicate that backpropagation neural network models achieve higher classification accuracy levels under noisy conditions than either linear discriminant or tree classification models. As further validation of this conclusion the three classification algorithms were used to classify cases using a previously published real-world data set.

The real-world data set selected for this study is the iris data set. The data set is available from the University of California, Irvine repository of machine learning databases (Merz & Murphy, 1996). The iris data set was used by Fisher in his derivation of the linear discriminant function (Fisher, 1936). Three classes of iris are discriminated using four numeric attributes. The data set consists of 150 cases, 50 for each group.

Similar to the synthetic data sets, feature values for selected cases were randomly replaced according to a probability (e.g., 10%) that reflected the noise level. Results of the analysis of the iris data sets are shown in Tables 4 and 5.

Two sets of numbers are contained in Table 5. The first set of numbers (those not in parenthesis) are the significance levels when comparing the three classification algorithms' accuracy ranks for the iris data sets. These results indicate a significant difference between the backpropagation neural network and the C5.0 tree classification model as well as significant differences between the linear discriminant model and the C5.0 tree classification model. More specifically, both the backpropagation neural network and the linear discriminant model significantly outperform the C5.0 tree classification model. This conclusion is valid at the 0.5

Table 4
Classification Accuracy and Rankings¹

Iris Data Set											
<u>Noise (%)</u>											
<u>Model</u>	0	10	20	30	40	50	60	70	80	90	Ave.
LinearDiscr.	.98	.95	.91	.87	.83	.79	.73	.65	.55	.51	.77
	(1)	(1)	(2)	(2.5)	(2)	(2)	(2)	(2)	(2)	(2)	(1.85)
C5.0 Tree	.96	.94	.91	.87	.82	.78	.72	.63	.53	.49	.76
	(2.5)	(2)	(2)	(2.5)	(3)	(3)	(3)	(3)	(3)	(3)	(2.70)
Backprop.	.96	.93	.91	.88	.85	.81	.76	.68	.58	.53	.79
	(2.5)	(3)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1.55)

¹Rankings appear in parenthesis

level of significance. However, the difference in ranks between the backpropagation neural network and the linear discriminant model are not significant.

The second set of numbers contained in Table 5 (those enclosed in parenthesis) are the significance levels for the difference in the 20% to 90% noise ranks. Previous research has shown that linear discriminant analysis does very well with the iris data set (Weiss & Kapouleas, 1989). Results for the 0% and 10% noise data sets confirm this. The linear discriminant model outperforms both the C5.0 tree classification and backpropagation neural network models. However, when 20 percent or more of the cases have noise introduced to them things change.

The backpropagation neural network model significantly outperforms both linear discriminant analysis and C5.0 tree classification.

Table 5

Statistical Significance Results

Wilcoxon Signed Rank Test^{1,2}

Iris Data Set

	Linear Discr.	C5.0 Tree Cl.	Backprop.
Linear Discriminant	—		
CS.0 Tree Classification	.007 (.014)	—	
Backpropagation	.463 (.011)	.013 (.011)	—

¹Values below .05 indicate statistically significant classification accuracy's

²Values in parenthesis are for ranks in the 20% and above noise levels

5. Discussion

While it is impossible to generalize these results to all real-world data sets, we can discuss the principles of each classification system that may account for these performance differences when noise is present in both synthetic and real-world data sets. To the extent possible, the discussion will be tied to distinctions between symbolic and connectionist

paradigms more generally. What follows will qualify the results of this paper as they may apply to paradigm-wide comparisons.

Differences between the three algorithms may be attributable to the search space explored by each. The primitive evidence combination function of backpropagation generalizes C5.0's primitive logical combinators (conjunction and inclusive disjunction) (Fisher & MeKusick, 1989). Finer granularity enables backpropagation to converge on logical concepts and others with less hardware, but each primitive must be specialized, which requires greater training. The course primitives of C5.0 also suggests that it takes bigger steps in a uniform search space - it approximates the final solution more quickly, but it may accept less than optimal solutions because it "oversteps" or "understeps" the optimum. While post-pruning methods reduce this problem to a certain extent, the presence of noise exacerbates it.

Another weakness of C5.0 is that it is not purely *polythetic*: C5.0 learning considers the utility of a single feature at a time. The predictive merits of feature value combinations are not explicitly considered, presumably to the detriment of prediction accuracy, until after the decision tree is produced. The polythetic "extension" converts the tree to a set of production rules. Each value is a polythetic concept that is "massaged" in order to improve its accuracy. In contrast, backpropagation is purely polythetic, in that the values of multiple features are simultaneously considered from the beginning. The linear discriminant method has drawbacks similar to C5.0 in that it employs a stepwise procedure for the selection of the most useful discriminating features (Klecka, 1980).

Another difference between the paradigms is that C5.0 and linear discriminant analysis assume that all the cases are available for processing, while backpropagation neural networks process cases as they become available. The finer granularity of backpropagation allows more

conservative steps through a search space. This conservatism is necessary; early in training, many cases are inconsistent with the evolving concept description (Fisher & McKusick, 1989). Once again, this inconsistency is exacerbated by noise. Ideally, no cases should irrevocably impact the incomplete concept description.

A final, and perhaps most overt, distinction between C5.0 and connectionist systems is the manner in which they explore their respective search spaces. C5.0 and the tree classification systems typically reconstruct the search space on demand. In contrast, most connectionist systems, including backpropagation, pre-enumerate a subset of the space, which is implicit in the number and interconnections between nodes. However, problems can arise if too much (e.g., slow convergence) or too little (e.g., the concept can't be learned) of the space is pre-enumerated (Fisher & McKusick, 1989).

6. Conclusions

Empirical comparisons have uncovered differences in the predictive accuracy of three learning paradigms when feature noise is present in the training and test data. Results indicate that backpropagation connectionist models outperform improved tree classification models like C5.0 as well as traditional statistical pattern classification models like linear discriminant analysis. While results can't be generalized to all models in these three major categories of learning from examples techniques, important distinctions between the different paradigms can be made:

1. The finer granularity of connectionist backpropagation algorithms enables them to converge on logical concepts and others with significantly greater accuracy than linear discriminant or tree classification algorithms, especially in the presence of

noise. This finer granularity greatly improves the chances that no cases will irrevocably impact the forming concept description.

2. The completely polythetic nature of connectionist backpropagation algorithms enables them to consider the value of multiple features from the very beginning of training. This may "dissipate" the effect of noise on any individual features.

This study sheds more light on the degradation in predictive accuracy of learning from examples algorithms when feature noise is present. Tasks such as speech recognition, character recognition, and natural language processing are domain areas where the ability to handle feature noise is very important. The results of the present study help to explain why researchers using connectionist models are making significant progress in these areas.

Perhaps the most important use of these findings would be to provide new comparative information to those seeking to build multistrategy learning systems that integrate multiple computational mechanisms in one learning system. Finding complementary strategies to combine learning performance in real-world tasks as well as achieving learning synergy in the interaction of strategies requires more knowledge about the performance of mono strategy systems (Michalski & Wnek, 1997).

References

- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth.
- Clark, R. & Niblett, T. (1987). The CN2 induction algorithm. *Machine Learning*, 3, 261-285.

- Datta, P. & Kibler, D. (1997). Symbolic nearest mean classifiers. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 82-87). San Mateo, CA: Morgan Kaufmann.
- Dillon, W. R. & Goldstein, M. (1984). *Multivariate analysis: Methods and applications*. New York: Wiley.
- Duda, R. & Hart, P. (1973). *Pattern classification and scene analysis*. New York: Wiley.
- Ebert, C. (1992). Visualization techniques for analyzing and evaluating software measures. *IEEE transactions on software engineering*, 18, 11, 1029-1034.
- Ebert, C. (1996). Fuzzy classification for software criticality analysis. *Expert systems with applications*, 11, 3, 323-342.
- Fisher, D. H. & McKusick, K. B. (1989). An empirical comparison of 1D3 and back-propagation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 788-793). San Mateo, CA: Morgan Kaufmann.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 79-88.
- Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural network design*. Boston, MA: PWS.
- Klecka, W. R. (1980). *Discriminant analysis*. London: Sage Publications.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann.
- Levine, D. M., Berenson, M. L. & Stephan, D. (1997). *Statistics for managers*. Upper Saddle River, NJ: Prentice Hall.

- Mahalanobis, P. C. (1963). On the generalized distance in statistics. *Proceedings of the national institute of science, India*, 12, 49-55.
- McClelland, J. & Rumelhart, D. (1988). *Explorations in parallel distributed processing*.
Cambridge, MA: MIT Press.
- Merz, C. J. & Murphy, P. M. (1996). UCI Repository of machine learning databases
[<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California,
Department of Information and Computer Science.
- Michalski, R. S. & Chilausky, R. L. (1980). Learning by being told and learning from examples.
International Journal of Policy Analysis and Information Systems 4, 2,
125-160.
- Michalski, R. S. & Wnek, J. (1997). Guest editor's introduction. *Machine Learning*, 27, 3, 205-
208.
- Niblett, T. & Bratko, I. (1986). Learning decision rules in noisy domains. In M. A. Bramer (ed.),
Research and development in expert systems III. Brighton, England:
Cambridge University Press.
- Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess
end games. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning:
An artificial intelligence approach* (Vol.1). Palo Alto, CA: Tioga.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 81-86.
- Quinlan, J. R. (1991). Improved estimates for the accuracy of small disjuncts. *Machine Learning*.
6, 1, 93-98.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan
Kaufmann.

Quinlan, J. R. (1997). *C5. 0 and See 5: Illustrative examples*. RuleQuest Research:

<http://www.rulequest.com>.

Shavlik, J. W., Mooney, R. J., & Towell, G. G. (1991). Symbolic and neural learning algorithms:

An experimental comparison. *Machine Learning*, 6, 2, 111-144.

Stone, M. (1974). Cross-validators choice and assessment of statistical predictions. *Journal of*

the Royal Statistical Society, 36, 111-147.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27, 1134- 1142.

Weiss, S. M. & Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural

nets, and machine learning classification methods. *Proceedings of the Eleventh International*

Joint Conference on Artificial Intelligence (pp. 781-787). San Mateo, CA: Morgan

Kaufmann.

Weiss, S. M. & Kulikowski, C. A. (1991). *Computer systems that learn*. San Mateo, CA:

Morgan Kaufmann.

Zurada, J. M. (1992). *Introduction to artificial neural systems*. St. Paul, MN: West.