ELSEVIER

Decision Support Systems 35 (2003) 441–454

Decision Support Systems

www.elsevier.com/locate/dsw

A comparative assessment of classification methods

Melody Y. Kiang *

Information Systems Department, College of Business Administration, California State University, 1250 Bellflower Blvd., Long Beach, CA 90840, USA

Accepted 1 May 2002

Abstract

Classification systems play an important role in business decision-making tasks by classifying the available information based on some criteria. The objective of this research is to assess the relative performance of some well-known classification methods. We consider classification techniques that are based on statistical and AI techniques. We use synthetic data to perform a controlled experiment in which the data characteristics are systematically altered to introduce imperfections such as nonlinearity, multicollinearity, unequal covariance, etc. Our experiments suggest that data characteristics considerably impact the classification performance of the methods. The results of the study can aid in the design of classification systems in which several classification methods can be employed to increase the reliability and consistency of the classification.

Keywords: Classification methods; Neural networks; Machine learning

1. Introduction

Classification of information is an important component of business decision-making tasks. Many decision-making tasks are instances of classification problem or can be easily formulated into a classification problem, e.g., prediction and forecasting tasks, diagnosis tasks, and pattern recognition. Classification tasks have assumed even more significance with the advent of the Internet. Internet as a communication and transaction channel provides a means to implement many new enabling technologies such as collaborative filtering and recommender systems [26] that enable one-to-one marketing and mass customization. Recommender systems are intended to assist customers by making suggestions to consumers online about available products and information. Recommender systems base their decisions by analyzing past behavior patterns of the individual customer as well as the behavior of other customers. Additionally, customer relationship management (CRM) systems are intended to aid decision makers in building and implementing marketing and promotion strategies. A primary objective of these systems in decision-making tasks is to classify the available information based on some criteria.

A variety of statistical methods and heuristics from AI literature have been used in the classification tasks. Many of these methods have also been applied to other decision-making scenarios such as business failure prediction [32], portfolio management [16], and debt risk assessment [33]. More recently, the problem of

* Tel.: +1-562-985-8944; fax: +1-562-985-4080.
E-mail address: mkiang@csulb.edu (M.Y. Kiang).

0167-9236/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0167-9236(02)00110-0

performing sensitivity analysis in classification systems using inverse classification methods have also been studied [17]. However, few studies have performed systematic tests to measure the comparative performance of the algorithms used in classification tasks [4]. It is clear from past studies that there is wide variance in the performance of classification algorithms under different scenarios [12,18,32].

As classification systems become an integral part of organizational decision support systems, adaptability to variations in data characteristics and dynamics of business scenarios becomes increasingly important. It is therefore imperative to move towards adaptive classification systems that selectively employ appropriate classification method(s) by first analyzing the available data. For such adaptive behavior, decision support systems should support different classification methods and apply the most appropriate method(s) that suits the data characteristics of the problem at hand. However, in order to build adaptive classification systems, one must first understand the performance characteristics of the classification methods in a systematic manner.

One main drawback of previous research on classification algorithm is that they mainly rely on uncontrolled data characteristics (biases) in their samples. The objective of this study is to understand the strengths and limitations of different classification methods and the effects of data characteristics on their performance in a controlled setting. We utilize a synthetic data set with carefully controlled biases for this purpose. In this research, the main focus is on the investigation of two AI techniques—neural networks and a decision tree method (C4.5), and three statistical methods—linear discriminant analysis (LDA), logistic regression analysis, and *k*th-nearest-neighbor (kNN) models. While the origins of these approaches are distinct and the underlying algorithms differ substantially, the fundamental process is the same; they are all inductive methods. The intention here is to investigate how the different classification methods perform when certain assumptions about the data characteristics are violated. The findings from this study will enable a better understanding of the classification methods. The findings should also help lay the foundations for the design of adaptive classification systems.

The rest of the paper is organized as follows: A brief review of relevant classification methods is presented in Section 2. The five classification methods (neural networks, C4.5, discriminant analysis, logistic regression, and kNN) studied in this paper are discussed in Section 3. Section 4 discusses the model assumptions related to the eight data characteristics. The experimental design and simulation results are presented in Section 5. Section 6 concludes the paper and suggests directions for future research.

2. Review of classification literature

Given that each classification method has its strengths and limitations and that real world problems do not always satisfy the assumptions of a particular method, one approach is to apply all appropriate methods and select the one that provides the best solution. This approach works well if, for a given problem situation, there is always one method (i.e., method A) that dominates all the other methods. That is, the misclassified example set of method A is subsumed by the misclassified example sets of all other approaches (see Fig. 1a). It is commonly observed that the misclassification set of the methods intersect each other (see Fig. 1b). Thus, most cases that are misclassified by one method can be correctly predicted by other approaches [32]. A recent study on the comparative analysis of ID3 and neural networks conducted by Dietterich, Hild, and Bakiri [12] also has similar observations. It can be seen in Fig. 1b that the misclassified examples in region 1 of method A can be correctly predicted by both methods B and C, and regions 2 and 3 can be correctly predicted by methods B and C, respectively. Therefore, one aim of this research is to understand the factors that affect the

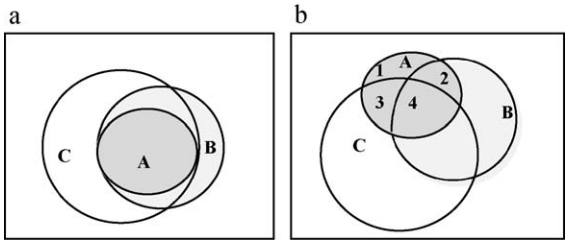

Fig. 1. (a) The misclassified example set of method A is subsumed by other approaches. (b) The misclassified example sets intersect each other.

performance of each method in order to assist in the development of a systematic approach to properly combine multiple classifiers to reduce the overall misclassification rate to region 4 of Fig. 1b.

Recent studies in comparing the performance of different classification techniques have been based mainly on experimental approaches [1,12,35]. Empirical comparisons among different algorithms suggest that no single method is best for all learning tasks [30,31]. In other words, each method is best for some, but not all tasks.

Several approaches have been proposed to utilize multiple learning algorithms. Generally speaking, there are two distinct schools of thought: the first is to combine the output from different learning methods; the other is to integrate several learning algorithms to form a hybrid classifier [3,38]. A hybrid classifier, the model class selection (MCS) system proposed by Brodley [7], performs recursive automatic bias selection based on a set of heuristic rules. Brodley suggested that different attributes may have distinct data characteristics and can be best explained by different models. Therefore, the key step in learning is to partition the data into meaningful subspaces and to choose the best model for each subspace. The MCS system contains three types of models: linear discriminant functions, decision trees, and instance-based classifiers. The result of the learning is a tree structured hybrid classifier.

An alternative approach is to combine the outputs of different classification methods. Wolpert [36] introduced stacked generalization, a way to combine the outputs from multiple generalizers trained with multiple partitionings of the original learning set. However, there are no systematic rules that can be used to generate an accurate combination. Breiman [5] followed Wolpert's idea of combining predicators instead of selecting the single best method, and proposed stacked regressions method. Stacked regression is a method for forming a linear combination of different predicators to give improved prediction accuracy. In general, improvement occurs when stacking together more dissimilar predictors. Bagging predictors, proposed by Breiman [6], is a method for generating multiple versions of a predictor, then obtaining an aggregated predictor by either taking the average over the versions (for numerical output) or using a plurality vote (for classification tasks). Breiman demonstrated that the stability of a procedure has great impact on the improvement achieved through bagging. Breiman [6] studied the instability of different predictors and concluded that neural networks, classification trees, and subset selection in linear regression were unstable, while the *k*th-nearest-neighbor method was stable.

There are a few studies in machine learning that attempted to look into the relationships of sample biases and the classification accuracy. For example, the study conducted by Shavlik, Mooney, and Towell [31] empirically analyzed the effects of three factors on the performance of two AI methods, neural networks and ID3. The three factors considered are (1) the size of training data, (2) imperfect training examples, and (3) the encoding of the desired outputs. Rendell and Cho [25] examined the effects of six data characteristics on the performance of two classification methods, ID3 and PLS1 (probabilistic learning system). The factors considered in their study include (1) size of the training set, (2) number of attributes, (3) scales of attributes, (4) error or noise, (5) class distribution, and (6) sampling distribution. The present research is different from theirs in that we approach the problem from the understanding of the underlying algorithm of each method so as to build a foundation that will facilitate the model integration using any of the above hybrid systems. In addition, a more comprehensive list of data characteristics that are pertinent to all five methods is analyzed.

A case study by Utgoff [34] suggested a hybrid algorithm called perceptron trees that combines decision trees with linear threshold units. The study focused on examining the representation biases of the two algorithms on various aspects and proposed a way to combine the two formalisms. The rational is that the two algorithms complement each other in certain ways, and by properly integrating them into one method, one can draw on the particular strengths of each individual algorithm. In this study, Utgoff's work is extended to include more algorithms. Moreover, the relationships between the sample bias and the representational (algorithm) bias are examined. Building of hybrid classifiers is beyond the scope of this research and a specific approach for integrating multiple systems is not proposed. The findings from this research can be used to suggest ways to integrate multiple algorithms into one that will have all the strengths of the algorithms without the weaknesses.

3. Classification methods

In this section, the five classification methods used in this paper are discussed. The review provides us the basis for forming hypotheses regarding the possible link between data characteristics and method performances. Although many new enhancements have been developed for AI methods aimed at solving specific types of problems, in this research, only the basic models are implemented to maintain the genuine characteristics of the original algorithms. Both AI and statistical methods can be fine-tuned for a particular problem situation. However, the more calibrated the model is, the more difficult it is for it to be generalized for new problem situations. Table 1 summarizes the important findings from the following review. Only factors that are pertinent to this study are presented.

Table 1
Summary of the five inductive methods

Method	Output format	Premises
Neural nets (back propagation) [28]	Network with weight connections	• Input/output (activation) functions are continuous and differentiable • Adequate size of training sample
C4.5 (ID3) [21,23,24]	Decision tree	• All regions are hyperrectangles • Density of regions • Need more training data for fragmented regions
LDA [15]	Math function	• Normality • Identical covariance matrices • Known prior probabilities and misclassification costs • Low correlation • Linearity • No multimodal distribution
Logistic [10]	Math function	• Low correlation • No multimodal distribution
kNN [37]	Class distribution	• Density of coverage

3.1. Neural networks

In this research, a feedforward network with back propagation [27], the most widely used learning algorithm, is implemented. A feedforward network model with no hidden layers works very much like a standard logistic regression model. One major criticism of neural networks is the difficulty in the selection of parameters needed to build a model. In the present study, the same network architecture is used throughout to minimize the need for an extensive trial-and-error process. In classification problems, the most popular network architecture used is the multilayer feedforward network (perceptron) [9]. Specifically, a network of three layers and two nodes in the hidden layer is implemented for investigation in this paper. An input preprocessor that normalizes the input values to the mean and standard deviation (S.D.) is applied, and a dot product function is used to aggregate input values. Learning rate is set at 0.01; momentum and weight decay are 0.0001. A sigmoid function is used as the output function to normalize the output to a value between zero and one that can then be interpreted as the probability of a class outcome. The employment of a sigmoid function can also attenuate the effect of outliner values and improve the overall performance of the network. A neural network needs the same training data to be fed over several iterations till it converges. In this study, it is observed that the network requires between 1000 and 2000 iterations to converge. Therefore, we set the number of iterations for training at 2000 in our experiments.

3.2. Decision tree (C4.5)

C4.5 [24] is an improved version of ID3, an inductive learning method developed by Quinlan [21–23]. C4.5 accepts both symbolic and numeric values as input, and generates a classification tree as output. It employs a splitting procedure which recursively partitions a set of examples into disjointed subsets. The division of the instance space is orthogonal to the axis of one variable and parallel to all other axes. Therefore, the resulting regions are all hyperrectangles. In other words, C4.5 will not perform well with problems that require diagonal partitioning. C4.5 also will not work well when the density of points in some regions is low or when the classification task is essentially probabilistic [24]. Moreover, the more

fragmented regions there are, the more data are needed to generate good results.

Brodley and Utgoff [8] proposed a multivariate decision tree method that does not limit the selection of a single variable at each splitting point. Therefore, diagonal partitioning is possible. However, in this research, the C4.5 (ID3) method is used due to the popularity of the algorithm. The output is a classification tree where the leaves contain class assignments determined by majority rule.

3.3. Multivariate discriminant analysis (MDA)

MDA methods accept a random sample of observations defined by a set of variables and generate a discriminant function that classifies observations into two or more groups by minimizing the expected misclassification cost. MDA assumes that all variables are normally distributed. In the case of the linear classifier, it also requires identical covariance matrices. In this research, Fisher's [15] discriminant analysis (DA) procedure, a widely used DA function, is implemented. The procedure constructs a discriminant function by maximizing the ratio of between groups' and within groups' variances. This method yields a linear function that divides the variable space into two partitions. For each example, the discriminant score, a value between 1 and -1, indicates the predicted group. The posterior probability of membership in the predicted group, given the discriminant score, can be obtained using Bayes' theorem.

Due to problems with quadratic DA functions reported in previous research [2], only linear discriminant analysis is investigated. Whether the function is linear or quadratic, a fundamental condition that must be satisfied is that the two groups are discrete and identifiable. Situations deviating from this condition can be found where observations of each group form disjoint regions in the variable space (see Fig. 2). Depending on the number of disjoint regions in each group, the discriminant functions may incur a high error rate for both the training and holdout sample.

3.4. Logistic models

An alternative to the linear DA model is logistic regression, a method that has fewer assumptions than linear discriminant models (i.e., no multivariate normality and equal dispersion assumptions) [10]. A logistic function having the following form is used:

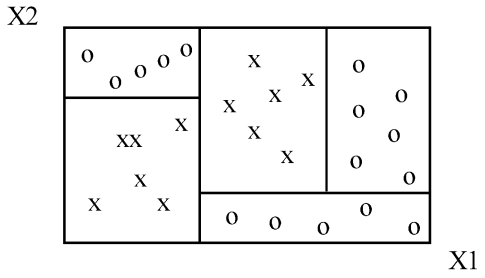

Fig. 2. Fragmented regions.

$$Y = \frac{1}{1+e^{y}}, \qquad y = a + \sum_{i=1}^{n} b_i X_i,$$

where X_i represents the set of individual variables, b_i is the coefficient of the ith variable, and Y is the probability of a favorable outcome. The outcome Y is a Bernoulli random variable. Earlier research supported the claim that when normality and identical covariance matrices assumptions hold, discriminant analysis estimators are preferred over those generated by logistic regression. However, in most applications, there is usually at least one variable that is qualitative (ruling out multivariate normality assumption), hence, the logistic regression model is preferred [13,20].

3.5. kth-Nearest neighbor (kNN)

The nonparametric (or distribution-free) method, kNN [37], is used for classifying observations into groups based on a set of quantitative variables. It relaxes the normality assumption and does not require a functional form as required in DA and logistic regression. The distance, $d(x,y)$, between any two observations x and y is usually defined by the Mahalanobis distance between x and y. Using the nearest neighbor decision rule, an observation is assigned to the group to which a majority of its kth-nearest neighbors belong. The sample distribution approximation is accomplished by dividing the variable space into an arbitrary number of decision regions, with the maximum bounded by the total number of observations. A recent study [35] shows that when applied to learning, kNN is a fairly robust and effective classifier

compared with the nearest hyperrectangle algorithm, an inductive method based on the nested generalized exemplar (NGE) theory [30].

4. Model assumptions

One focus of this research is to examine the data characteristics that may affect the performance of different inductive methods. The data characteristics were selected based on the identified strengths and weaknesses of each method. A list of the data characteristics to be investigated is summarized in Table 2. The following provides detailed description for each data characteristic.

4.1. The multivariate normal distribution of independent variables

A potential problem with linear discriminant analysis (LDA) is the appropriateness of the normality assumption. "In practice, deviations from the normality assumption, at least in economics and finance, appear more likely to be the rule than the exception" [14]. A study by Deakin [11] suggests that financial ratios are not normally distributed, rather, they are positively skewed. Violations of the normality assumptions may lead to a biased and overly optimistic prediction of the performance of rules in the population, and thus limit the usefulness of the model. The Kolmogorov–Smirnov test statistics are applied to each of the independent variables in the data sets to test for the normality.

Table 2
Summary of the five inductive methods with respect to the eight data characteristics

Method	Hypotheses
Neural nets	Both static and dynamic scenarios. Affected by sample size.
C4.5 (ID3)	Static scenario. Affected by multimodal distribution and sample size.
DA	Static scenario. Affected by normality and linearity violations, low correlation, multimodal, and identical covariances.
Logistic	Static scenario. Affected by low correlation and multimodal distributions.
kNN	Static scenario. Affected by sample size.

4.2. The linearity between dependent and independent variables

The performance of a linear model depends a great deal on the multivariate relationship between independent and dependent variables. The F-test is often used to test this joint relationship.

4.3. Multicollinearity

A high degree of correlation among independent variables (multicollinearity) will have adverse effects on the parameter estimates of LDA and logistic procedures. A simple procedure for testing collinearity is the use of the correlation matrix, while a more reliable method, variance inflation factor [19], will assist in the identification of correlated variables. Salchenberger, Cinar, and Lash [29] reported that the neural network model performs well when multicollinearity is present and a nonlinear relationship exists between the input and the output variables.

4.4. The covariance equality of two classes

The LDA method requires the presence of homoscedascity. A test for this equality of variances can be conducted through the Cochran's test [19].

4.5. The multimodal distribution of the sample

The power of the analysis of LDA and logistic models is affected when the sample is multimodal. Graphical checks for modes can be conducted through histograms, box plots, and other similar plots.

4.6. Dynamic versus static nature of the problem

Most of the methods examined assume that the population distribution will not change with time. Thus, the models based on historical data are not time-dependent and may be violated at times. Time series analysis is one approach to this type of problem. A time series model tries to account for as much as possible of the regular movement (wavelike functions, trend, etc.) in the time series, leaving out only the random error. The method can be applied when there is a time series variable in the problem to be modeled. However, a more complex dynamic system could

affect the distributional characteristics of the model over time. Neural network models have been found to handle both types of problems well. The neural network method is unique, in that it allows adaptive model adjustment. It responds swiftly to changes in the real world. This dynamic feature of the model can be tested through cross-validation. A portion of the data sample is withheld to test the behavior of the model for the portion of the sample used.

4.7. Sample proportion

Earlier research in bankruptcy prediction using DA models show that when the sample proportion differs from the true population, the prediction accuracy becomes very poor. Therefore, an option was added to the DA model to allow model builders to specify the population proportion when it is different from the sample proportion. In business applications, due to factors such as the availability of positive and negative examples (e.g., bankruptcy cases), and the cost and time involved to collect the data, an equal proportion of positive and negative examples are used for model building (training). Therefore, sample proportion bias is a common problem in real world applications. The prediction accuracy of logit models is not affected by biased sample proportion due to its nonparametric nature and it is not necessary to handle such situations differently. However, the effects of biased sample proportion on C4.5 (ID3) and neural networks are still unknown, and there is no mechanism implemented in either algorithm to adjust for the bias. One way to tackle this problem is to selectively duplicate the training examples so as to arrive at the same proportion as the population. For example, if the sample has an equal proportion while the population proportion has a 20/80 distribution for positive and negative examples, the number of negative examples in the training set can be duplicated to make them the same proportion as the population. In this study, the equal sample proportion is used to train the models. If significant deterioration in prediction accuracy is noticed for a certain method, the adjusted training sample will then be used.

4.8. Sample size

Previous research in machine learning suggests that the size of training sample not only affects the speed of training, but also has an impact on the performance of different classifiers. In other words, the reliability of the estimates may depend on the sample size used [7,25,31]. For some methods, large sample size is required in order to achieve its maximum prediction accuracy whereas others may need a relatively small data set. Similar to the problem of biased sample proportion, the size of a training set is usually constrained by resources and availability of the data and could impose an artificial constraint on the selection of the best fit model. Table 2 summarizes our hypotheses regarding the classification methods based on a review of the classification literature. These results will be tested and validated in this paper using synthetic (simulated data). The details of the simulation study are presented in the next section.

5. Experimental design and simulation results

Each of the first seven data characteristics has two states, present and absent. Extreme cases are used to contrast the impact of biases on model performance. Two independent variables (X_1 and X_2) and one dependent variable (Y) are generated for each case, as this is the minimum number of variables needed to simulate those factors. In addition, data sets that satisfy all eight characteristics (normality, little or no correlation, linearity, identical covariance matrices, static and no multimodal distribution, equal distribution, and a fair sample size) are generated as the basis for comparison. The functional form of the base case is given by the following equation:

$$Y = A_X X_1 + A_2 X_2 + \varepsilon$$

where $X_1 \sim N(\mu_1, V_1)$, $X_2 \sim N(\mu_2, V_2)$, and $\varepsilon \sim N(0,1)$. A_1, A_2, V_1, V_2, μ_1, and μ_2 are constants. The mean values and standard deviations for x_1 and x_2 are chosen so that the two groups are easily differentiable. The mean value of Y is used as the cutoff point for the two groups in order to derive an equal number of positive and negative examples in the population. The bias is inserted in the data set by systematically altering the state of data characteristic from absent to present one at a time. The performance difference for each method before and after the change is used to test the hypotheses (as stated in Table 2). Multiple t

statistics are used to measure the significance of the performance difference in the test data. If any discrepancies are revealed, the values in Table 2 will be adjusted to reflect the new findings.

To test the effect of each data characteristic, a population of 50,000 cases is generated each time. An equal number of cases (25,000 each) are generated for $Y=0$ and $Y=1$ groups. To form the training and test data sets, 50 cases are randomly drawn from the $Y=0$ group and another 50 cases from the $Y=1$ group for a total of 100 examples in each data set. The process is repeated 100 times to form 100 training and test data sets, respectively, in order to average out the possible bias in any single sample run. The results presented below are the average performances of the 100 runs, both for training and test.

The following data characteristics describe the biases inserted at each step during the test.

(1) *Nonnormal distribution*: A data set with exponential distribution is generated to compare with normally distributed sample. The random variable that measures the time between two occurrences that have a Poisson distribution is an exponential random variable. The density function for exponential distribution is $f(x)=\lambda e^{-\lambda x}$ for $x \geq 0$, $\lambda > 0$. The mean of the exponential distribution is $\mu = 1/\lambda$ and the variance is also $1/\lambda$. The same functional form as the base case is used with parameter $\lambda = \mu_1$, μ_2 for x_1, x_2, respectively.

(2) *Nonlinearity*: A quadratic function is used in this test:

$$Y = A_X X_1^2 + A_2 X_2^2 + \varepsilon,$$

where $X_1 \sim N(\mu_1, V_1)$, $X_2 \sim N(\mu_2, V_2)$, and $\varepsilon \sim N(0,1)$. Again, A_1, A_2, V_1, V_2, μ_1, and μ_2 are constants and were chosen to make two distinct groups.

(3) *High correlation*: To generate data sets with high correlation between X_1 and X_2, make $X_2 = X_1 + \varepsilon'$ in the base class function where $\varepsilon' \sim N(0,1)$.

(4) *Unequal covariance*: Data sets with different covariance matrix for the two groups $Y=0$ and $Y=1$ were generated. The base case functional form is used to generate examples for the $Y=1$ group, and the high correlation function for $Y=0$ group.

(5) *Multimodal distribution*: The same functional form as the base case is used. However, group $Y=0$ is distributed in two disjoint regions separated by points from group $Y=1$. Fig. 3 depicts an example of the sample distribution by groups.

(6) *Dynamic environment*: Again, the same functional form as the base cases is used. Instead of using a constant A_1 as the coefficient of X_1, it is assumed that the coefficient of X_1 changes over time. A sine function is used as part of the coefficient value and the sine function changes its value from 0 to 1 to 0 to -1, then back to 0. Each time, a complete cycle is used to generate 200 examples and then chronologically divided into two sets. The first 100 examples are used for training and the rest are used as test sample.

(7) *Unequal sample proportion*: The sample cases are randomly drawn from the same population used in the base case. Equal sample proportion is used for training while the true population has a 10/90 distribution and is reflected in the test cases.

(8) *Sample size*: Sample sizes of 30, 50, 100, 300, and 1000 are randomly selected from the multimodal distribution data set each time for both training and test.

For each data set generated, necessary tests were performed (i.e., plotting, normality test, etc.) to verify the existence of bias in the data. Performance is assessed with respect to the ability of the methods to accurately predict the appropriate class for the test (holdout) sample.

Each experiment includes 200 sample runs (100 training runs and 100 test runs), and the results presented are the average of the 100 runs. Therefore, there are a total of 200 (sample runs/cell) × 5 (models) × 8 (data characteristics) = 8000 runs. To test the effect of sample size on model performance, 20 train-

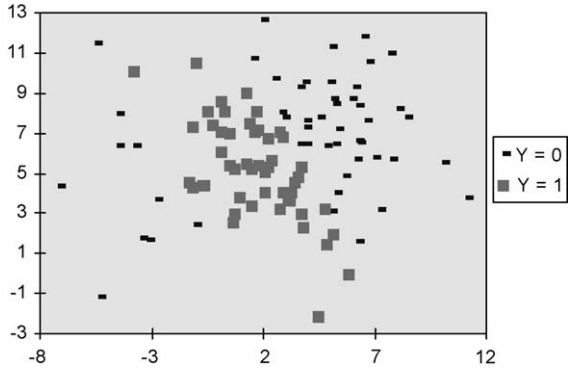

Fig. 3. Multimodal distribution.

Table 3
Misclassification rate of the training data

Method	Base	Normality assumption	Linearity assumption	Static/ dynamic	Low correlation	Multimodal	Sample proportion	Identical covariance
Neural nets	1.09	0.96	3.00	6.44	3.68	6.96	1.12	4.69
C4.5 (ID3)	1.35	1.18	2.04	2.90	2.18	2.8	1.51	2.43
DA	2.53	7.92	8.64	6.75	4.60	12.32	2.60	7.13
Logistic	0.84	0.84	3.12	6.42	3.96	11.95	0.90	5.13
kNN	2.68	2.46	3.71	5.90	3.80	4.80	2.77	4.43

Method	Sample size (30)	Sample size (50)	Sample size (100)	Sample size (300)	Sample size (1000)
Neural nets	5.50	5.80	6.75	5.65	6.83
C4.5 (ID3)	5.43	3.30	3.15	1.49	1.13
DA	12.50	10.20	11.90	9.53	11.01
Logistic	11.67	9.60	11.55	9.43	10.94
kNN	10.67	6.00	4.50	2.68	1.98

ings and 20 test runs were performed for each sample size. Therefore, to test five different sample sizes, a total of 100 trainings and 100 test runs are executed. The results of the training and test data are shown in Tables 3 and 4, respectively. Fig. 4 plots the misclassification rates of the test results for the first seven data characteristics and groups them by method. Fig. 5 shows the classification performance versus the sample size. Due to the complexity of the problem in this study, the possible interaction among factors and the varying degree of biases in each data characteristic is not tested. In order to test all the possible interactions among biases, the experiment design necessary to test these hypotheses will be greater than 2^8 factorial.

5.1. Analysis of the results

For each method, *t*-test is used to test the significance of the performance difference between the base case and each biased sample (see Table 4). The results show that for all methods except C4.5 (ID3), the bias factors have either a nonsignificant or an adverse effect on the performance of a method. For the decision tree method (C4.5), only the multimodal

Table 4
Misclassification rate of the test data

Method	Base	Normality assumption	Linearity assumption	Static/ dynamic	Low correlation	Multimodal	Sample proportion	Identical covariance
Neural nets	2.22	2.05	4.31 *	16.41 *	4.98 *	7.88 *	2.27	6.59 *
C4.5 (ID3)	9.17	7.09**	7.90**	20.16 *	7.67**	12.58 *	7.77**	9.05
DA	3.52	8.53 *	9.24 *	18.44 *	5.52 *	12.99 *	3.72	8.26 *
Logistic	2.25	2.02	4.30 *	18.11 *	5.13 *	12.48 *	2.35	6.39 *
kNN	5.26	4.64	5.72	19.18 *	5.89	8.52 *	5.01	6.74 *

Method	Sample size (30)	Sample size (50)	Sample size (100)	Sample size (300)	Sample size (1000)
Neural nets	8.95	7.80	8.55	7.07	6.85
C4.5 (ID3)	18.50	15.80	14.7	7.49	4.64
DA	13.67	13.00	13.05	10.57	11.06
Logistic	13.83	12.90	12.75	10.48	10.92
kNN	13.00	10.80	9.75	5.62	3.36

* Tests of significance, $p < 0.05$, significantly higher than base.
** Tests of significance, $p < 0.05$, significantly lower than base.

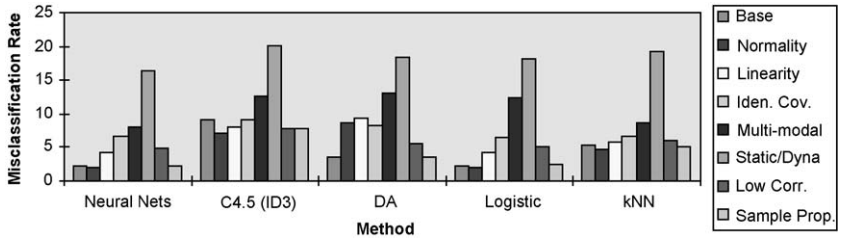

Fig. 4. Effect of data characteristics on performance (grouped by method).

and dynamic factors have an adverse impact on its performance. However, the method performed poorly in the base case. The reason for the poor performance is as follows. In the simulated data sets, there are only two independent variables. A typical distribution as shown in Fig. 6a requires diagonal partitioning. Since C4.5 does not allow diagonal partition, it will mimic it by performing a series of orthogonal partitioning along the diagonal line (see Fig. 6b). As discussed by Brodley and Utgoff [8], this will result in a large tree and poor generalization to the new instances even after substantial pruning of the tree. The argument can be further supported by the superior performance of C4.5 in all the training cases (see Table 3). When looking at the effect of sample size to the performance of a method, the performance of C4.5 seems to suggest that the problem can be alleviated when a large sample size is available. Interestingly, some of the added bias factors have significantly improved the performance of C4.5. Although the added biases have increased the complexity of the problem, they also have improved the generalizability of the resulting trees. This can be seen from the narrowed gaps between the performance of training and test data sets in those cases.

For each data characteristic, the mean differences among different methods were also compared. Table 5 shows the best performing methods based on multiple t statistics.

In some cases, the performances of all methods are much worse than their performance in the base case. This is mainly due to the increased complexity in the problem situation. Therefore, more attention should be paid to the relative performance change among the methods. The following discussion summarizes the observations from the results derived in this study.

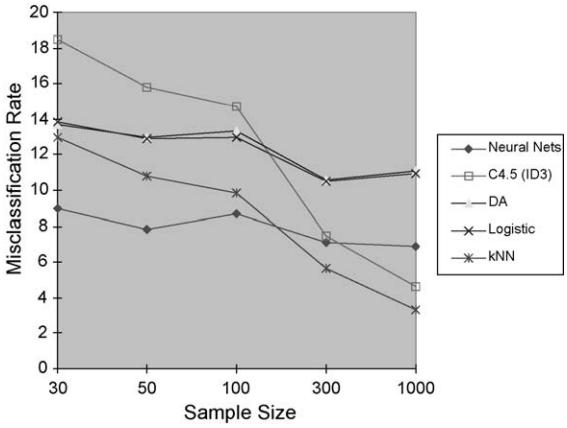

Fig. 5. The classification performance versus the sample size.

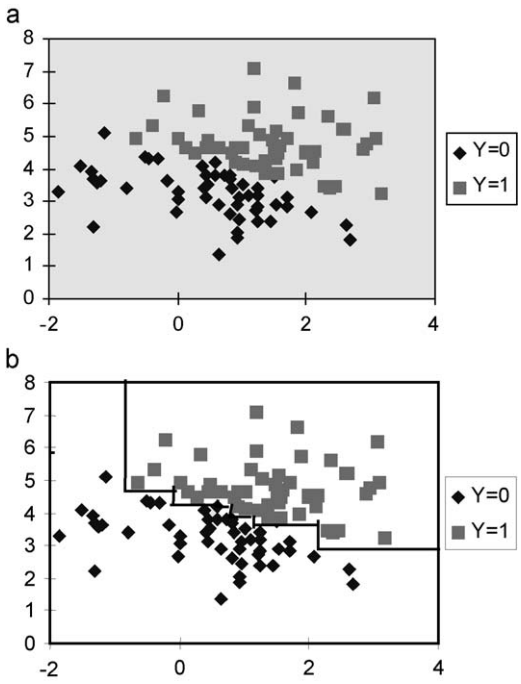

Fig. 6. (a) Base case test data set #1. (b) C4.5 partitioning.

Table 5
Best performing methods under different data characteristics

Data characteristic	Best performing methods
Base case	Logistic regression, neural net
Nonnormality	Logistic regression, neural net
Nonlinearity	Logistic regression, neural net
Dynamic scenario	Neural net
High correlation	Logistic regression, neural net
Multimodal distribution	kNN, neural net
Unequal sample proportion	Logistic regression, neural net
Unequal covariance	kNN, logistic regression, neural net
Sample size	Smaller sample: neural net Larger sample: kNN

5.2. Analysis by method

(1) Generally speaking, neural networks and the logistic model are superior to the other methods. Between the two methods, neural networks significantly outperformed the logistic model in the multimodal and dynamic cases.

(2) The poor performance of C4.5 was due to the limitation discussed earlier. Therefore, we only look at the relative performance change of C4.5 across different bias factors. Based on Table 4, only dynamic and multimodal factors have an adverse impact on the method. However, the impacts are rather minor compared to the impacts of those factors to the other methods. Moreover, for the multimodal case, when the sample size increases, the performance of C4.5 improves substantially.

(3) The logistic model is superior to DA in all cases, especially when the normality, linearity, and identical covariance assumptions do not hold.

(4) kNN performs well in the multimodal case, especially when the sample size is large.

(5) kNN significantly outperformed DA when the normality, linearity, and identical covariance assumptions are not in place. kNN is also superior to DA in the multimodal case. However, DA did better in the base and unequal sample proportion cases. As mentioned earlier, the examples in the unequal sample proportion case were randomly drawn from the base case population. An unequal sample proportion in the data did not cause the inferior performance of kNN. Moreover, the performance of kNN may improve when the sample size is increased.

5.3. Analysis by data characteristics

(6) Only the normality assumption has an impact on DA, which concurs with our hypothesis in Table 2.

(7) The results show that the linearity assumption has a moderate effect on the performance of neural nets and the logistic model, which seems to differ with our hypothesis. For neural nets, the cause of the discrepancy is probably due to using fixed network architecture throughout all experiments. The network architecture selected may not be the most suitable architecture for all problem situations. If the network is fine-tuned for each case, the performance should improve.

Table 6
Revised version of Table 2

Method	Hypotheses	Experiment results
Neural nets	Both static and dynamic scenarios. Affected by sample size.	Identical network architecture is affected in dynamic scenario. Nonlinearity, high correlation, unequal covariance, and multimodal distribution affect performance. The method remains superior to other methods in relative performance.
C4.5 (ID3)	Static scenario. Affected by multimodal distribution and sample size.	Static scenario. Generally, inferior to other methods. Affected by multimodal distribution. Increase in sample size improves performance.
DA	Static scenario. Affected by normality and linearity violations, high correlation, multimodal, and unequal covariances.	Static scenario. Unaffected only by unequal sample proportion.
Logistic	Static scenario. Affected by high correlation, and multimodal distributions.	Static scenario. Affected by nonlinearity, multimodal distribution, unequal covariance, and high correlation. Second best performing method overall.
kNN	Static scenario. Affected by sample size.	Static scenario. Affected by multimodal distribution, sample size, and unequal covariance.

(8) The results suggest that nonnormality, nonlinearity, and unequal sample proportions do not change the relative performance of the other methods.

(9) For the dynamic case (see Table 5a and d), it is clear that the neural networks model is the only method that has not changed its relative grouping and significantly outperformed all the other methods.

(10) The low correlation bias has a moderate effect on all methods except kNN.

(11) In the multimodal case, the relative performance of C4.5 (ID3) and kNN have improved from their base case while DA and logistic models performed relatively worse. Also, as shown in Fig. 5, the performance ranking of the methods changes drastically at higher sample sizes.

(12) Sample size has significant effect on kNN and C4.5 methods, and then on neural networks (see Fig. 5). It has less impact on DA and logistic models, and in our test case, the improvement stabilized after the 300-sample size for those two models.

Now the entries in Table 2 can be adjusted according to the findings. The revised version of Table 2 is shown in Table 6.

6. Conclusion and future research

The controlled experiments conducted with simulated data show that the classification algorithms are sensitive to changes in data characteristics. The misclassification rates due to biases can be substantially high in the presence of even a single bias. In general, more than one method seems to be appropriate candidates based on the type of bias in the data. Neural net and logistic regression methods provide the best relative performance under most scenarios. Another important concern brought forth by our results is the impact of dynamic variations in data on classification performance. The results indicate that all the classification methods studied here are adversely affected when the underlying phenomenon is nonstatic. Since most business phenomena exhibit dynamic behavior, care should be exercised in calibrating classification systems to such scenarios.

The study has shown that there is no single method that clearly outperforms all methods in all problem situations. Therefore, one recommendation from this study is to build classification systems that employ a number of different classification algorithms. The systems should be designed to select the right method or to properly combine different methods to form a hybrid classifier in response to the presence of different biases in data. The results of this study can be applied to the design of classification systems. Based on the results, one can design classification systems that can employ a group of methods for a given problem situation based on data characteristics. Systematic approaches can be developed to find consensus of results among various classifiers through a proper combination of them. This will enhance the consistency and adaptability of classification systems.

As packaged software solutions for classification tasks becomes more commonplace, adaptability in classification systems necessarily assumes a greater role. This requires us to study performance characteristics of classification methods in greater detail. Further research is needed for a better understanding of the performance characteristics of classification methods. While different data characteristics may affect performance of the classification methods to different degrees, the level of bias in individual data characteristics may also impact performance to different degrees. More elaborate experiments are required to examine the possible interactions among factors, and the effect of varying levels of biases on the outcome. Table 2 needs to be expanded substantially in order to include the possible interactions among all factors. For each data characteristic, several versions of biases should be used to test the models. For example, to examine the normality assumption, we have used samples with exponential distribution in the preliminary test. Other types of distributions and factors should be generated and tested to gain a full understanding of the possible impact of each factor. Similarly, to analyze the impact of multimodal distribution, different forms of multimodal distribution should be simulated. After a better understanding of the strengths and limitations of each method is obtained, the possibility of integrating two or more algorithms together to solve a problem should be investigated. The objective is to utilize the strength of one method to complement the weakness of another. For example, Dietterich et al. [12] proposed combining ID3 and neural networks by using ID3 as a preprocessor to identify the important features from a given sample. The preprocessor helps reduce the

training time substantially and improves the overall performance of the network. Future research effort should focus on investigating the possibility of combining statistical methods with the AI algorithms. Different methods of combining these methods should be explored.

Acknowledgements

The research was supported in part by a grant from the Information Technology and Organizations Program of the National Science Foundation (NSF), IRI-9505715. The author greatly appreciates the help from Dr. Raghu T. Santanam in revising the paper.

References

[1] H. Almuallim, T.G. Dietterich, Learning Boolean concepts in the presence of many irrelevant features, Artificial Intelligence 69 (1994) 279–305.
[2] E.L. Altman, R.A. Eisenbeis, J. Sinkey, Applications of Classification Techniques in Business, Banking, and Finance, JAI Press, Greenwich, CT, 1981.
[3] V. Biou, J.F. Gibrat, J.M. Levin, B. Robson, J. Garnier, Secondary structure prediction: combination of three different methods, Protein Engineering 2 (1988) 185–191.
[4] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (1998), Madison, WI.
[5] L. Breiman, Stacked regression, Machine Learning 24 (1996) 49–64.
[6] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[7] C.E. Brodley, Recursive automatic bias selection for classifier construction, Machine Learning 20 (1995) 63–94.
[8] C.E. Brodley, P.E. Utgoff, Multivariate decision trees, Machine Learning 19 (1995) 45–77.
[9] Y. Chtioui, D. Bertrand, M. Devaux, D. Barba, Comparison of multilayer perceptron and probabilistic neural networks in artificial vision application to the discrimination of seeds, Journal of Chemometrics 11 (1997) 111–129.
[10] D.R. Cox, The Analysis of Binary Data, Chapman & Hall, London, 1970.
[11] E.B. Deakin, Distributions of financial accounting ratios: some empirical evidence, Accounting Review (Jan. 1976) 90–96.
[12] T.G. Dietterich, H. Hild, G. Bakiri, A comparison of ID3 and backpropagation for english text-to-speech mapping, Machine Learning 18 (1995) 51–80.
[13] B. Efron, The efficiency of logistic regression compared to normal discriminant analysis, Journal of the American Statistical Association 70 (Dec. 1975) 892–898.
[14] R.A. Eisenbeis, Pitfalls in the application of discriminant analysis in business, finance, and economics, Journal of Finance 32 (3) (June 1977) 875.
[15] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.
[16] M.Y. Kiang, R.T. Chi, K.Y. Tam, DKAS: a distributed knowledge acquisition system in a DSS, Journal of Management Information Systems 9 (4) (Spring 1993) 59–82.
[17] M.V. Mannino, M.V. Kaushik, The cost-minimizing inverse classification problem: a genetic algorithm approach, Decision Support Systems 29 (2000) 83–300.
[18] M. Meila, D. Heckerman, An experimental comparison of model-based clustering methods, Machine Learning 42 (2001) 9–29.
[19] J. Neter, W. Wasserman, M.H. Kutner, Applied Linear Statistical Models, 3rd edn., Irwin, Homewood, IL, 1990.
[20] S.J. Press, S. Wilson, Choosing between logistic regression and discriminant analysis, Journal of the American Statistical Association 73 (1978) 699–705.
[21] J.R. Quinlan, Discovering rules by induction from large collection of examples, in: D. Michie (Ed.), Expert Systems in the Micro Electronic Age, Edinburgh University Press, Edinburgh, 1979.
[22] J.R. Quinlan, Learning efficient classification procedures and their applications to chess end games, in: R.S. Michalski, J. Carbonell, T. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach 1, Tioga Publishing, Palo Alto, CA, 1983, pp. 463–482.
[23] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[24] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[25] L. Rendell, H. Cho, Empirical learning as a function of concept character, Machine Learning 5 (1990) 267–298.
[26] P. Resnick, H. Varian, Recommender systems, Communications of the ACM 40 (3) (1997) 56–58.
[27] D.E. Rumelhart, G. Hinton, R. Williams, Learning representation by back-propagating errors, Nature 323 (9) (1986) 533–536.
[28] D.E. Rumelhart, J. McClelland, PDP Research Group (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Bradford Books, MA 1986, pp. 328–330.
[29] L.M. Salchenberger, E.M. Cinar, N.A. Lash, Neural networks: a new tool for predicting thrift failures, Decision Sciences 23 (1992) 899–915.
[30] S. Salzberg, A nearest hyperrectangle learning method, Machine Learning 6 (1991) 277–309.
[31] J.W. Shavlik, R.J. Mooney, G.G. Towell, Symbolic and neural learning algorithms: an experimental comparison, Machine Learning 6 (1991) 111–144.
[32] K.Y. Tam, M.Y. Kiang, Managerial applications of neural networks: the case of bank failure predictions, Management Science 38 (7) (July 1992) 926–947.

[33] R.R. Trippi, E. Turban (Eds.), Neural Networks in Finance and Investing—Using Artificial Intelligence to Improve Real-World Performance, Probus Publishing, Chicago, 1993.
[34] P.E. Utgoff, Perceptron trees: a case study in hybrid concept representations, Connection Science 1 (4) (1989) 377–391.
[35] D. Wettschereck, T.G. Dietterich, An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, Machine Learning 19 (1995) 5–27.
[36] D.H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.
[37] M.A. Wong, T. Lane, A *k*th nearest neighbor clustering procedure, Journal of the Royal Statistical Society, Series B 45 (1983) 362–368.
[38] X. Zhang, J.P. Mesirov, D.L. Waltz, Hybrid system for protein secondary structure prediction, Journal of Molecular Biology 225 (1992) 1049–1063.

Melody Y. Kiang is an Associate Professor of Computer Information Systems at California State University, Long Beach. She received her MS in MIS from the University of Wisconsin, Madison, and PhD in MSIS from the University of Texas at Austin. Prior to join CSULB, she was an Associate Professor at Arizona State University. Her research interests include the development and applications of artificial intelligence techniques to a variety of business problems. Her research has appeared in *Information Systems Research (ISR)*, *Management Science*, *Journal of Management Information Systems*, *Decision Support Systems*, *IEEE Transactions on SMC*, *EJOR*, and other professional journals. She is an Associate Editor of *Decision Support Systems* and the Editor-in-Chief of *Journal of Electronic Commerce Research*.