Journal of Global Optimization (2006) 36: 33–50
DOI 10.1007/s10898-005-6152-y

An Experimental Evaluation of Some Classification Methods

M. DOUMPOS, E. CHATZI and C. ZOPOUNIDIS
Department of Production Engineering and Management, Financial Engineering Laboratory, Technical University of Crete, University Campus, 73100 Chania, Greece (e-mail: kostas@dpem.tuc.gr)

(Received 15 December 2005; accepted in final form 22 December 2005)

Abstract. The classification problem is of major importance to a plethora of research fields. The outgrowth in the development of classification methods has led to the development of several techniques. The objective of this research is to provide some insight on the relative performance of some well-known classification methods, through an experimental analysis covering data sets with different characteristics. The methods used in the analysis include statistical techniques, machine learning methods and multicriteria decision aid. The results of the study can be used to support the design of classification systems and the identification of the proper methods that could be used given the data characteristics.

Key words: classification, machine learning, Monte Carlo simulation, multicriteria decision aid, Statistical techniques.

1. Introduction

The classification problem involves the assignment of some objects to a set C of predefined classes. Each object is a multivariate vector in $\mathbb{R}^n$, i.e., $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^{\cdot}$, where n is the number of attributes (independent variables) and x_{ij} is the description of object $\mathbf{x}_i$ on attribute x_j. Given a training sample with m observations $(\mathbf{x}_i, c_i)$, where $c_i \in C$ denotes the class label for the training object $\mathbf{x}_i$, the objective is to identify a classification model that will assign all objects into one of the classes, as accurately as possible.

In the literature, several methods have been proposed to develop classification models. Traditionally, statistical techniques such as discriminant analysis and logistic regression, have been dominating this field. Alternative non-parametric methods include, operations research methods (mathematical programming, multicriteria decision aid; Stam, 1997; Doumpos and Zopounidis, 2002; Zopounidis and Doumpos, 2002), rule induction algorithms and decision trees (Breiman et al., 1984), neural networks (Ripley, 1996), nearest–neighbor algorithms (NN) (Duda et al., 2001), kernel methods (Vapnik, 1998; Schölkopf and Smola, 2002), rough sets (Pawlak, 1982), etc.

During the past two decades several experimental studies have been presented considering mainly statistical methods and mathematical programming techniques. Freed and Glover (1986) compared thee linear programming (LP) models to linear discriminant analysis (LDA) through a Monte Carlo analysis. The results showed that LP models outperformed LDA, but some of them were found sensitive to outliers. Joachimsthaler and Stam (1988) also considered quadratic discriminant analysis (QDA) and logistic regression (LOG) and found that QDA provided better results, especially when there were significant differences in the class variance–covariance matrices. Similar results on the performance of LP models relative to QDA were also obtained by Rubin (1990). Östermark and Höglund (1998) extended the previous results to multi-class problems and compared LP models to statistical methods and the recursive partitioning algorithm (RPA), observing that when there are significant differences in the misclassification costs, then some LP approaches and RPA provide the best results. In a more recent study, Sueyoshi (2006) compared two mixed integer programming (MIP) models to statistical methods, neural networks and a decision tree algorithm, using both real-world data as well as a Monte Carlo simulation. The results showed that a two-stage MIP model outperformed all the other approaches in almost all situations. Sandri and Marzocchi (2004) performed an extensive analysis of several machine learning algorithms and statistical techniques, including decision trees, LDA, and NN; they found that NN performed poorly compared to the other methods, mainly for non-normal data and in the presence of irrelevant attributes.

The outgrowth in the classification research over the recent years led to the development of many other popular methods which have not been considered in prior experimental studies. This finding highlights the importance of extending the prior experimental results. Based on this motivation, this study considers a variety of parametric and non-parametric techniques in a Monte Carlo experimental analysis. Three statistical techniques (linear discriminant analysis, quadratic discriminant analysis, logistic regression), and five non-parametric techniques (NN, probabilistic neural networks, support vector machines, multicriteria decision aid), are used in the analysis. Except for the variety of the methods used in this study, another important feature of this research involves the analysis of both continuous and discrete data.

The rest of the paper is organized as follows: the next section briefly outlines the seven classification methods used in analysis, followed by the description of the experimental design and the presentation of the obtained results. The final section concludes the paper and suggests directions for future research.

2. Classification Methods

2.1. LINEAR AND QUADRATIC DISCRIMINANT ANALYSIS

Linear discriminant analysis (LDA) is a multivariate statistical classification method. The objective of LDA is to obtain a linear combination of the independent variables (attributes) that maximizes the variance between the classes relative to within-class variance. In the simplest two-class case, this leads to a linear discriminant function $f(\mathbf{x}) = \gamma + \mathbf{a}^\top\mathbf{x}$, where γ is a constant term and $\mathbf{a}$ is a column vector consisting of the discriminant coefficients of the decision attributes. Given the discriminant scores of the objects their classification is performed in a straightforward way through the introduction of a discriminant cut-off point, which is estimated according to the *a priori* probabilities of class membership and the misclassification costs. The parameter estimation process is based on two major assumptions: (a) the independent variables are multivariate normal, and (b) the class variance–covariance matrices are equal.

Quadratic discriminant analysis (QDA) extends LDA through the use of a quadratic discriminant function $f(\mathbf{x}) = \gamma + \mathbf{a}^\top\mathbf{x} + \mathbf{x}^\top\mathbf{B}\mathbf{x}$, where $\mathbf{B}$ is a $n \times n$ symmetric matrix with the discriminant coefficients b_{ij} for each product $x_i x_j$. The parameters of the quadratic discriminant function (constant term γ, discriminant coefficients $\mathbf{a}$ and $\mathbf{B}$) are estimated under the assumptions that the decision attributes are multivariate normal and the class variance–covariance matrices are unequal.

2.2. LOGISTIC REGRESSION

Logistic regression (LOG) has become increasingly popular as an alternative to LDA and QDA. The main advantage of LOG over LDA and QDA is that it does not impose assumptions of the statistical distribution of the data or the structure of the class dispersion matrices. LOG uses the logistic function to model the posterior probability of class membership given the attribute vector $\mathbf{x}$ as $f(\mathbf{x}) = \left[1 + \exp(-\gamma - \mathbf{a}^\top\mathbf{x})\right]^{-1}$. Based on the estimated posterior class membership probability $f(\mathbf{x})$ an object is classified, using a cut-off probability point estimated so as to minimize the classification error. The model's coefficients are obtained through maximum likelihood estimation techniques.

2.3. NEAREST–NEIGHBOR ALGORITHMS

Nearest–neighbor algorithms (NN) have been extensively used as non-parametric density estimation techniques (Wong and Lane, 1983). In NN the classification of any object $\mathbf{x}_i$ is based on the assessment of its similarity to the training objects. The objective is to identify a set K consisting of the k most similar to $\mathbf{x}_i$ training objects and then to take the classification decision through a simple majority vote (i.e., assign $\mathbf{x}_i$ to the class that appears most frequently within the set K). The Euclidian distance is used to assess the similarity between any two objects. The number k of nearest–neighbors defines the level of smoothing for the decision region. In this study, several experiments were performed with different values for k, and finally $k=5$ (five nearest–neighbors) was selected as the one providing good results. The advantages of NN involve their simple implementation and robust behavior for large data sets, whereas the main disadvantage involves the large storage and computational requirements (for large data sets).

2.4. PROBABILISTIC NEURAL NETWORKS

Probabilistic neural networks (PNN) can be realized as a network of three layers (Specht, 1990). The input layer includes n nodes, each corresponding to one attribute. The inputs of the network are fully connected with the m nodes of the pattern layer. Each node of the pattern layer corresponds to one training object. The input $\mathbf{x}_i$ to a pattern node j is passed to an exponential activation function that produces the output of the pattern node j(σ is a user defined smoothing parameter):

$$\text{Output}_j = \exp\left(-\|\mathbf{x}_j - \mathbf{x}_i\|^2 \Big/ 2\sigma^2\right)$$

The outputs of the pattern nodes are passed to the summation layer. In the case of dichotomous classification, the summation layer consists of two nodes each corresponding to one class. The pattern nodes corresponding to training objects from class 1 are connected only to the summation node corresponding to this class. The summation nodes simply sum the output of the pattern nodes to which they are connected with. This summation provides the outputs $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ for an input vector $\mathbf{x}$. An object $\mathbf{x}_i$ is classified in class 1 if $f_1(\mathbf{x}) > f_2(\mathbf{x})$; otherwise it is classified in class 2.

2.5. SUPPORT VECTOR MACHINES

Support vector machines (SVM) have become a popular classification method since the mid-1990s (Vapnik, 1998). The SVM are based on the structural risk minimization principle, which characterizes the performance of a model in terms of the trade-off between the training error rate and the class separating margin that is related to the complexity of the model.

For a dichotomous problem, with a linear separating function, the objective of SVM is to develop an optimal hyperplane $f(\mathbf{x}) = \gamma + \mathbf{a}^\top \mathbf{x}$ such that $f(\mathbf{x}) > 0$ iff an object belongs in class 1. The class separating margin defined for such a hyperplane is $2/\|\mathbf{a}\|$. Thus, a quadratic programming problem is formulated to estimate the parameters of the model (vector **a** and constant γ). In the general non-linear case, the input data are non-linearly mapped to a higher dimensional space F (feature space) and the above linear analysis is then performed in F. The non-linear mapping is performed using an appropriate kernel function K. With the introduction of the kernel function, the model takes the form $f(\mathbf{x}) = \gamma + K(\mathbf{x}^\top, \mathbf{A}^\top)\mathbf{a}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an $m \times n$ matrix with the training data, K is the kernel function that defines a non-linear mapping of the column vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ into $\mathbb{R}^{1 \times m}$, and **a** is a $m \times 1$ column vector. Popular kernel functions include the polynomial kernel, the radial basis kernel and the sigmoid kernel (Schölkopf and Smola, 2002). In this study the radial basis kernel is used.

2.6. THE UTADIS METHOD

The UTADIS method is a multicriteria decision aid classification method (Doumpos and Zopounidis, 2002). The method develops an additive utility classification model $U(\mathbf{x}) = p_1 u_1(x_1) + p_2 u_2(x_2) + \cdots + p_n u_n(x_n)$, where p_i is an non-negative weight for attribute x_i $(p_1 + p_2 + \cdots + p_n = 1)$, and $u_i(x_i)$ is the corresponding marginal utility function defined in a piece-wise linear form. On the basis of this functional representation form, in a dichotomous case, an object $\mathbf{x}_i$ is classified in class 1 iff $U(\mathbf{x}_i) \geqslant t$, where t is a cut-off point. The parameters of the classification model (marginal utilities, attributes' weights, cut-off point) are estimated through a linear programming approach that minimizes the classification error for the training objects. A post-optimality analysis is also employed to investigate the robustness of the obtained optimal solution (Doumpos and Zopounidis, 2002).

3. Experimental Design

The main objective of this research is to examine the relative performance of the considered methods in terms of different data characteristics. Throughout the experiment two-class data sets are considered in $\mathbb{R}^5$ (five decision attributes). The restriction of the analysis to the two-class case is due to the fundamental character of dichotomous problems in classification research. Actually, any multi-class problem can be decomposed into a series of dichotomous problems using appropriate techniques (e.g., error-correcting output coding; Dietterich and Bakiri, 1995). On the other hand, the consideration of a small number of attributes (five attributes) is based on the finding that in real-world situations the objective is to develop a reliable classification model based on limited information. Even, when a large set of attributes is available, attribute selection techniques (John et al., 1994) are often employed to reduce the dimensionality of the

Table 1. Factors describing the data characteristics in the experimental analysis

Continuous data		Discrete data	
Factors	Scenarios	Factors	Scenarios
Distribution	Normal Log-normal Mixture	Discrete levels	Two Three Mixture
Class separation	Linear Non-linear	Class separation	Linear Non-linear
Training objects	200 500 1000	Training objects	200 500 1000
Correlation	Low High	Correlation	Low High

problem and to eliminate redundant attributes. Therefore, analyzing the performance of the methods given a small set of attributes is well-suited to real-world situations.

Given the above two settings (two classes, five attributes) different factors are selected to describe other characteristics of the data sets such as: the nature of the data (continuous, discrete), the statistical distribution of the data, the number of training objects, the correlation between the decision attributes, and the separation of the classes. Table 1 summarizes the factors considered in the analysis, whereas the following two sub-sections describe in more details the design of the experiment for both the continuous and the discrete data.

3.1. CONTINUOUS DATA

3.1.1. *Data Distribution*

The data are generated from two distributions. In the first case, the standard normal distribution is used. Normality is a common assumption to statistical classification methods (LDA, QDA) and its consideration in the comparison provides a reference (benchmark) point. The alternative scenario involves the generation of log-normal variables with unit variance and mean of 1.87. The consideration of the log-normal distribution in this analysis is based on its popularity in modeling many real-world situations (for instance, a common assumption in finance is that returns are log-normally distributed).

Of course in a real-world situation it is unlikely that all attributes will follow a common distribution. Therefore, it was decided to consider in the analysis a situation where the attributes come from different distributions. This was performed considering two of the five decision attributes as standard normal random variables and the remaining three attributes as log-normal variables with the aforementioned parameters.

3.1.2. *Type of Class Separation*

The way that the classes are separated is an important factor affecting the performance of a classification model. Two scenarios are considered for this factor. In the first case it is assumed that the classes are nearly linearly separable. Some popular classification methods such as LDA and logistic regression develop linear classification models. Given, that in a real-world situation one cannot preclude the case of near linear separability, it is important to investigate the performance of the methods in this case, mainly for the methods that develop non-linear models (QDA, NN, PNN, SVM, UTADIS). To model the linearly separable case, a linear classification rule is imposed:

$$\begin{array}{lcl} f(\mathbf{x}_i) > 0 & \Leftrightarrow & \mathbf{x}_i \in \text{Class } 1, \\ f(\mathbf{x}_i) < 0 & \Leftrightarrow & \mathbf{x}_i \in \text{Class } 2, \end{array} \tag{1}$$

where $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ is a linear function of the decision attributes where $\mathbf{a}$ is taken as a uniformly distributed random vector in [1, 10].

Of course, perfect linear separation is rarely observable in practice, whereas near-linear separation is possible. Therefore, a 10% level of noise is imposed on the above classification rule, through the perturbation of the class assignment of the objects as defined by the rules (1). The level of noise introduced in the data was selected so as to ensure a reasonable amount of error rate.

The second scenario considered in the analysis involves the case where the classes are not linearly separable. Non-linear separability is considered using the classification rule (1) with a quadratic discriminant function $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{B}\mathbf{x}$, where the elements of $\mathbf{a}$ and $\mathbf{B}$ are modeled as uniform random variables, with $a_i \in [1,\ 10], b_{ii} \in [0,\ 2]$, and $b_{ij} \in [-2, 2]$ for all $i, j = 1, 2, \ldots, 5$ $(i \neq j)$. These specifications were selected to ensure a reasonable separability of the classes. Similarly to the linear case, a 10% level of noise is imposed to the quadratic classification rule.

It should be noted that the use of linear and quadratic rules for the separation of the classes does not necessarily favor LDA and QDA, since both methods are based on specific statistical assumptions (multivariate normality with equal/unequal class dispersion matrices). The use of the two classification rules in this study does not imply any of these assumptions.

3.1.3. *Number of Training Objects*

The size of the training sample as defined by the number of training objects is an important factor for the performance of a classification system. Generally, as additional information is added to the training sample (new training objects), the development of a classification model is expected to provide more reliable and robust results. However, the impact of this factor is not the same for all methods (e.g., some methods may perform well even with small training sets, whereas other may be more suitable for larger training sets). To model this factor in the experiment, three scenarios are considered each corresponding to different number of training objects: 200, 500, and 1000 objects.

3.1.4. *Correlation*

The correlation between the decision attributes is another factor which has important implications in the development of classification models. Low-correlation corresponds to cases where each individual attribute provides different information compared to other attributes, whereas high correlation indicates possible redundancy in the set of attributes. Multicollinearity issues also arise when considering correlated attributes, which may lead to model instability. The consideration of both cases (low and high correlation) in this experimental analysis enables the investigation of the performance of the methods in both situations. In modeling the low-correlation case, the attributes are generated as independent random variables. To generate data with high correlation, initially, the description x_{i1} of each object $\mathbf{x}_i$ on the first attribute x_1 is generated according to the selected distribution. Then, the description x_{ij} of each object $\mathbf{x}_i$ on the remaining attributes x_j $(j=2,\ldots,5)$ is generated as follows:

$$x_{ij}=\frac{1}{j-1}\sum_{k=1}^{j-1}x_{ik}+\gamma_{ij},$$

where γ_{ij} is a random variable with the same distribution with one used for the decision attributes (either standard normal distribution or log-normal distribution).

3.2. DISCRETE DATA

For the generation of discrete data, the factor involving the statistical distribution used for the continuous case, is replaced with the form of the discrete data. In particular, three settings (scenarios) are considered. In the first setting, binary $\{-1,1\}$ data are randomly generated. The second scenario involves the generation of discrete random data with three levels $\{-1,0,1\}$, and finally a mixture of the two scenarios is also examined [two attributes modeled as binary $\{-1,1\}$ and the remaining three attributes modeled with three levels as $\{-1,0,1\}$].

The approach used to generate correlated discrete data is similar to the one employed for the continuous case. Initially, the description x_{i1} of each object $\mathbf{x}_i$ on the first attribute x_1 is generated according to the selected distribution. Then, the description x_{ij} of each object $\mathbf{x}_i$ on the remaining attributes $x_j (j=2,\ldots,5)$ is generated as follows:

$$x_{ij} = \delta_{ij} \operatorname{sgn}\left(\frac{1}{j-1} \sum_{k=1}^{j-1} x_{ik} + \rho \gamma_{ij} \right),$$

where δ_{ij} is a discrete binary $\{-1, 1\}$ random variable such that $\Pr(\delta_{ij}=1) = 0.8$ and $\Pr(\delta_{ij}=-1)=0.2$, γ_{ij} is a random variable uniformly distributed in $[-1, 1]$, and ρ is a constant set as $\rho=0.001$ for binary $\{-1, 1\}$ data, and $\rho=0$ for the three-level data $\{-1, 0, 1\}$. All discrete attributes are finally decoded into appropriate dummy variables.

The remaining two design factors (number of training objects and type of class separation) are used in the same way as the case of continuous data.

3.3. DATA GENERATION

On the basis of the methodology described in the preceding sub-sections to model continuous and discrete data, the data generation process was implemented as follows. For each combination of the factors 30 replications are performed (overall there are 72 combinations of the design factors). At each replication, initially, 5000 training and 5000 testing objects are generated in $\mathbb{R}^5$. All objects are then classified with the classification rules of Section 3.1.2. Depending on the selected number of training objects ($m=200$, $m=500$ or $m=1000$, cf. Section 3.1.3), a random selection is performed from the 5000 objects generated for training. The random selection is performed such that the classes are balanced in the training sample. The testing samples are compiled in a similar way. Throughout the experiment, all testing samples consist of 500 cases (250 from each class).

It should be noted that the selection of balanced class sizes for both training and validation does not pose a significant limitation on the results. Of course, in many real world situations there is a considerable unbalance (asymmetry) between the classes. Developing a model without taking such an asymmetry into consideration is highly likely to lead to biased results towards the larger class. In such a situation, even if the overall accuracy rate is high, the resulting model cannot be accepted. The problem, however, can be easily addressed in various ways. For instance, the classification rule can be adjusted taking into consideration the prior probabilities or misclassification costs. Alternatively, for some algorithms it is possible to weight the training cases. The aim of such techniques is to overcome the unbalance of the classes. Thus, taking equal class sizes in this experimental analysis is not an unrealistic setting.

Table 2. Overall average test error rates (in %)

LDA	15.20 (3)
QDA	16.53 (4)
LOG	14.63 (2)
NN	22.40 (5)
PNN	14.67 (2)
SVM	13.70 (1)
UTADIS	14.25 (2)

Overall, the experimental analysis involves a $2 \times 3 \times 2 \times 3 \times 2 \times 7$ full-level factorial design with 2160 training samples (30 replications for each of the 72 combinations of the design factors) matched with the same number of testing samples.

4. Analysis of Results

The results are analyzed in terms of the test error rates of the methods using the transformation $2 \arcsin \sqrt{\text{error rate}}$ for variance stabilization (Joachimsthaler and Stam, 1988). Table 2 illustrates the overall average test error rate for all methods along with the Tukey's grouping (Yandell, 1977) at the 5% level (in parentheses). The overall results show that SVM provide the most accurate classification followed by UTADIS, LOG, and PNN. On the other hand, the worst results are obtained with the NN algorithm.

A further analysis of variance (ANOVA) of the results is employed to perform a further investigation of the performance of the methods in terms of the factors used in the experiment. The ANOVA results showed several effects involving interactions between the methods with other factors to be significant at the 5% level. The effects with the highest explanatory power (measured with the Hays ω^2 statistic; Cohen, 1988) are shown in Table 3 and they are analyzed in the subsequent sub-sections.

Table 3. Major explanatory effects of the classification performance of the methods (ANOVA results)

Effects	Degrees of freedom	Mean squares	F	ω^2(%)
Methods	6	1.936	2495.12	23.72
Data type×methods	6	1.184	1526.19	14.50
Data type×correlation×methods	6	0.583	751.37	7.14
Class separation	1	2.212	2851.58	4.52
Correlation×methods	6	0.275	354.97	3.37
Data type	1	1.103	1422.47	2.25
Training objects×methods	12	0.087	111.92	2.11
Data type×training objects×methods	12	0.086	111.20	2.10
Class separation×methods	6	0.132	170.72	1.61

Table 4. Test error rates (in %) with respect to the type of data

	Continuous	Discrete	F (p-value)
LDA	16.66 (3)	13.74 (3)	312.87 (<0.01)
QDA	19.27 (5)	13.79 (3)	1381.29 (<0.01)
LOG	16.06 (2)	13.20 (2–3)	299.47 (<0.01)
NN	18.15 (4)	26.64 (4)	593.98 (<0.01)
PNN	17.02 (3)	12.32 (1)	996.29 (<0.01)
SVM	14.46 (1)	12.94 (2)	178.57 (<0.01)
UTADIS	15.74 (2)	12.76 (1–2)	406.47 (<0.01)

4.1. THE EFFECT OF THE DATA TYPE

The impact of the type of data on the test error rates of the methods is presented in Table 4. For each method the ANOVA results (F statistic and the associated p-value) are also reported to compare the statistical significance of the differences between the cases of continuous and discrete data. The results clearly show that the performance of all methods (except NN) is significantly improved when discrete data are considered. The relative improvement is higher for QDA and PNN, whereas for SVM the improvement is limited compared to the other methods. In the case of continuous data SVM provide the best results followed by UTADIS and logistic regression. For the case of discrete data, the best results are obtained with PNN, followed by UTADIS and SVM. These results indicate that SVM and UTADIS are the more robust approaches providing good results for both discrete and continuous data. On the other hand, methods such as QDA, NN, and PNN are quite sensitive to the type of data used.

4.2. THE EFFECT OF THE TYPE OF CLASS SEPARATION

As expected the type of class separation has a significant impact on the performance of the methods. The corresponding results of Table 5 clearly demonstrate the deterioration in the performance of most methods when a non-linear separation of the classes is evident. Actually, NN is the only method for which no significant differences are observed between linear and non-linear separation. For the methods that lead to non-linear models (QDA, NN, PNN, and SVM), the increase in the error rates in the case of non-linear separation is smaller compared to the other methods. On the other hand, the increase in the error rate of LDA, LOG, and UTADIS is higher. LDA and LOG lead to the development of linear models, whereas the piece-wise linear model of UTADIS does not consider interactions between the attributes. Overall, in the linear case UTADIS and LOG provide the lowest error rates followed by SVM and LDA. Non-linear methods such as QDA, NN, and PNN perform poorly in this case. SVM also lead to highly non-linear models, but their emphasis on complexity

Table 5. Test error rates (in %) with respect to the type of class separation

	Linear	Non-linear	F(p-value)
LDA	13.12 (2)	17.28 (3)	734.11 (<0.01)
QDA	15.87 (4)	17.19 (3)	46.63 (<0.01)
LOG	12.46 (1)	16.79 (3)	836.73 (<0.01)
NN	22.13 (5)	22.66 (4)	2.43 (0.12)
PNN	13.79 (3)	15.55 (2)	96.13 (<0.01)
SVM	13.08 (2)	14.31 (1)	103.51 (<0.01)
UTADIS	12.45 (1)	16.06 (2)	625.64 (<0.01)

control seems to enable the avoidance of overfitting. In the non-linear case, SVM provide the best results followed by PNN and UTADIS. Overall, the results show that despite the flexibility of non-linear models, overfitting can be an issue when linear or near-linear class separation is evident. The complexity control approach implemented in SVM seems to be an efficient approach to address this issue.

4.3. THE EFFECT OF THE NUMBER OF TRAINING OBJECTS

Generally, as the training sample size increases, one should expect that the error rate will decrease (except for cases where the additional information incorporated in additional training examples is noisy). As demonstrated in Table 6, this expectation is supported by the results of this analysis. Except for NN the error rates for all the other methods decrease as the number of training objects increases (as it will be shown later, for increase for the NN is due to the poor performance on discrete data). The rate of improvement is higher when 500 training objects are used instead of 200, whereas the differences between 1000 and 500 objects are (in most methods) limited. In particular, the rate of decrease in the error rate when 500 objects are used instead of 200 ranges between 9.07 (LOG) and 10.97% (SVM) with an average of 9.42%. On the other hand, when comparing the test error rates for 1000 training objects as opposed to the use of 500 objects, the rate of decrease ranges between 0.44 (UTADIS) and 4.2% (QDA) with an average of 2.27%. Overall, the impact of the training set size is more significant for SVM, QDA, and NN. SVM's error rate for a training set size of 1000 objects is 14.11% lower compared to the error rate for a training set size of 200 objects. Similarly QDA's error rate decreases by 13.15%, whereas NN's error rate is increased by 19%. In terms of the relative performance of the methods, SVM provide good results in all cases, followed by UTADIS, whereas LOG and PNN provide similar results.

The above results are extended considering the interaction between the number of training objects and the type of data. The corresponding results are presented in Table 7. The results show that when continuous data are considered, the increase of the training objects reduces the error rate of all

Table 6. Test error rates (in %) with respect to the number of training objects

	200	500	1000	F(p-value)
LDA	16.17 (3)	14.83 (3)	14.59 (3)	31.26 (<0.01)
QDA	17.87 (4)	16.20 (4)	15.52 (4)	59.81 (<0.01)
LOG	15.66 (2–3)	14.24 (2–3)	13.99 (2–3)	36.12 (<0.01)
NN	20.48 (5)	22.33 (5)	24.37 (5)	30.93 (<0.01)
PNN	15.73 (2–3)	14.29 (2–3)	13.99 (2–3)	35.99 (<0.01)
SVM	14.95 (1)	13.31 (1)	12.84 (1)	118.66 (<0.01)
UTADIS	15.26 (1–2)	13.78 (1–2)	13.72 (2)	39.33 (<0.01)

Table 7. Test error rates (in %) with respect to the number of training objects and the type of data

Data type		Training objects			F(p-value)
		200	500	1000	
Continuous	LDA	18.02 (3)	16.06 (3–4)	15.89 (3–4)	28.57 (<0.01)
	QDA	20.38 (5)	18.91 (6)	18.51 (6)	25.43 (<0.01)
	LOGIT	17.49 (2–3)	15.45 (2–3)	15.23 (2–3)	30.44 (<0.01)
	NN	19.47 (4)	17.64 (5)	17.35 (5)	41.07 (<0.01)
	PNN	18.28 (3)	16.52 (4)	16.25 (4)	27.07 (<0.01)
	SVM	16.09 (1)	13.93 (1)	13.37 (1)	110.19 (<0.01)
	UTADIS	17.07 (2)	15.14 (2)	15.02 (2)	32.11 (<0.01)
Discrete	LDA	14.31 (2)	13.62 (2)	13.28 (2)	9.30 (<0.01)
	QDA	15.36 (3)	13.49 (2)	12.53 (1–2)	94.89 (<0.01)
	LOGIT	13.82 (1–2)	13.03 (2–3)	12.74 (1–2)	11.77 (<0.01)
	NN	21.49 (4)	27.03 (4)	31.40 (3)	88.44 (<0.01)
	PNN	13.19 (1)	12.06 (1)	11.72 (1)	27.46 (<0.01)
	SVM	13.81 (1–2)	12.68 (1–3)	12.32 (1–2)	32.82 (<0.01)
	UTADIS	13.45 (1–2)	12.41 (1–2)	12.43 (1–2)	15.40 (<0.01)

methods (including NN). As noted earlier the reduction is higher when 500 objects are used instead of 200. The overall improvement of the error rate (between 1000 and 200 objects) is higher for SVM (approximately 17%), whereas the improvement for the other methods range between 9.2 (QDA) and 12.9% (LOG). SVM always provide the best results, followed by UTADIS. In the case of discrete data, PNN provide the best results followed by UTADIS, and SVM. Once again the increase in the number of training objects leads to reduced error rates (except for NN), with the most significant improvement observed for QDA (18.4%). The improvements for the other methods are lower compared to the case of continuous data, ranging between 7.2 (LDA) and 11.1% (PNN).

4.4. THE EFFECT OF THE DEGREE OF CORRELATION

As mentioned earlier, building classification models using correlated data may have diverse effects on the expected error rates. The results of Table 8 show, that overall, the error rates are improved with correlated data (except for NN). The only method that seems unaffected by the existing correlations is QDA, whereas the performance of the other methods shows sig-

Table 8. Test error rates (in %) with respect to the degree of correlation

	Low correlation	High correlation	F(p-value)
LDA	15.80 (3)	14.59 (3)	48.75 (<0.01)
QDA	16.54 (4)	16.52 (4)	0.09 (0.76)
LOG	15.26 (2)	14.00 (2–3)	53.27 (<0.01)
NN	20.40 (5)	24.39 (5)	90.29 (<0.01)
PNN	16.34 (3–4)	13.00 (1)	384.86 (<0.01)
SVM	14.26 (1)	13.14 (1)	88.13 (<0.01)
UTADIS	14.97 (2)	13.54 (1–2)	80.53 (<0.01)

nificant improvement. The most significant improvement is observed for PNN. Overall, it is apparent that the existing correlations in the data can be an important factor when deciding which method to use. In the low-correlation case, SVM provide the best results followed by UTADIS and LOG, whereas in the high-correlation case PNN and SVM are the best classifiers followed by UTADIS.

The interaction of the degree of correlation with the type of data was also found to be an important issue in explaining the performance of the methods. The corresponding results presented in Table 9, extend the above analysis on the effect of the correlation on the error rates of the methods. The results show that in the case of continuous data SVM and UTADIS are the best methods for uncorrelated data, whereas for correlated data PNN also provides good results. Also, considerable improvements (more than 20%) are observed in the error rates of NN and PNN when correlated data are used. The improvements for the other methods are smaller ranging between 4.17 (LDA) and 7.84% (SVM). QDA is the only method for which no significant differences are observed between the two cases (uncorrelated vs. correlated data). QDA's performance is also robust in the discrete data case with slightly reduced error rate for high correlation. For the other methods, significant improvements are observed ranging between 7.87 (SVM) and 12.9% (UTADIS). PNN provide the best results with both uncorrelated and correlated data followed by SVM and UTADIS when uncorrelated data are used and by UTADIS in the case of high correlations.

4.5. SYNOPSIS OF THE RESULTS

Overall, the results of the experiment verify the effectiveness of most of the non-parametric methods in developing classification models as opposed to parametric techniques. In most cases, SVM outperformed the other methods with the exception of correlated discrete data. The PNN provided good results for discrete data, but for continuous data they showed medium performance compared to other techniques. UTADIS provided robust results in most cases both in continuous and discrete data. On the

Table 9. Test error rates (in %) with respect to the degree of correlation and the type of data

	Continuous data			Discrete data		
	Low correlation	High correlation	$F(p$-value)	Low correlation	High correlation	$F(p$-value)
LDA	17.01 (3)	16.30 (4)	7.46 (<0.01)	14.60 (3)	12.88 (3)	72.98 (<0.01)
QDA	19.16 (3)	19.37 (5)	0.54 (0.46)	13.91 (2–3)	13.67 (4)	1.98 (0.16)
LOG	16.56 (2–3)	15.55 (2–3)	14.79 (<0.01)	13.95 (2–3)	12.45 (2–3)	62.06 (<0.01)
NN	20.43 (5)	15.87 (3–4)	877.41 (<0.01)	20.38 (4)	32.90 (5)	566.32 (<0.01)
PNN	19.53 (4)	14.50 (1)	660.71 (<0.01)	13.15 (1)	11.49 (1)	98.31 (<0.01)
SVM	15.05 (1)	13.87 (1)	50.82 (<0.01)	13.47 (1–2)	12.41 (2–3)	44.86 (<0.01)
UTADIS	16.29 (2)	15.19 (2)	20.99 (<0.01)	13.64 (1–2)	11.88 (1–2)	102.79 (<0.01)

Table 10. Cumulative distribution of each method's rankings

	Ranking						
	1st	2nd	3rd	4th	5th	6th	7th
LDA	0.00	1.39	13.89	44.44	63.89	94.44	100.00
QDA	0.00	9.72	15.28	22.22	38.89	73.61	100.00
LOG	19.44	34.72	59.72	72.22	93.06	100.00	100.00
NN	0.00	1.39	11.11	19.44	29.17	37.50	100.00
PNN	29.17	51.39	62.50	68.06	80.56	94.44	100.00
SVM	36.11	51.39	69.44	83.33	94.44	100.00	100.00
UTADIS	19.44	51.39	68.06	91.67	100.00	100.00	100.00

other hand, NN have been found inappropriate for discrete data. From the statistical techniques, logistic regression provided the best results which, in some cases, were comparable to those of the non-parametric techniques.

Tables 10 and 11 provide a synopsis of the results. Table 10 is based on the ranking of the methods from the one with the lowest error rate (1st) to the one with the highest error rate (7th). The presented results involve the cumulative distribution of the rankings for each method considering all the experimental scenarios (factors combinations). In particular, each entry (i, j) of Table 10 shows the percentage of the scenarios where method i was ranked as j in terms of its test error rate. For instance, the entry (LDA, 3rd) shows that LDA was ranked as the third best method in 10 out of 72 scenarios (13.89%) considered in the experimental analysis. The results demonstrate the high performance of SVM which is ranked as the best method in 26 scenarios (36.11%). The UTADIS was never ranked in the two worst positions, where as in 91.67% of the total number of scenarios it is ranked within the best four methods. On the other hand, PNN outperformed the other methods in several cases, but there were also cases in which PNN provided the worst performance (four scenarios, all for continuous data).

Table 11. Pairwise comparison of the methods

	LDA	QDA	LOG	NN	PNN	SVM	UTADIS
LDA	–	76.39	1.39	77.78	31.94	27.78	2.78
QDA	23.61	–	20.83	65.28	23.61	9.72	16.67
LOG	98.61	79.17	–	83.33	41.67	41.67	31.34
NN	22.22	34.72	16.67	–	11.11	0.00	12.50
PNN	68.06	76.39	58.33	88.89	–	40.28	52.78
SVM	59.72	90.28	56.94	100.00	58.33	–	54.17
UTADIS	97.22	83.33	66.67	87.50	47.22	45.83	–

Table 11 performs a pairwise comparison of the methods. Each entry (i, j) of this table shows the percentage of the scenarios where method i had a lower error rate than method j. This comparison, clearly demonstrates the potentials of non-parametric techniques. PNN, SVM, and UTADIS outperform all parametric methods (LDA, QDA, LOG) in the majority of the cases.

5. Conclusions and Future Research

The experimental study presented in this paper analyzed the relative performance of different methods in terms of specific data characteristics. Both statistical techniques as well as non-parametric methods were analyzed. The analysis was based on an experimental setup considering continuous and discrete data.

The results suggest that, in most cases, non-parametric techniques (except for NN) outperform the statistical approaches, with SVM, PNN, and UTADIS providing the best results. SVM were found to provide good results in cases of non-linear separation and mainly for continuous data. PNN worked better for discrete data as well as for data with highly correlated attributes. Finally, UTADIS showed good performance in cases of linear class separation and managed to remain competitive to the other methods in most cases. From the three statistical methods, LOG provided the best results, which, in some cases (mainly for linear class separation), were found to be close (or even better) to the ones of the non-parametric techniques. Finally, LDA and QDA were not found to be competitive to the above methods despite the fact that the classification rules modeled in the experiment (linear, quadratic) matched the modeling forms of the two methods.

Such an experimental analysis could be extended in several aspects. For instance, other non-linear class separation forms could be investigated expect for the quadratic rule used in this analysis. The effect of class overlap and the impact of irrelevant attributes are also issues for future research. The introduction of additional classification methods, such as rule induction techniques, classification trees, and artificial neural networks, would also be useful in generalizing the findings of the analysis. Furthermore, the analysis can be extended to consider the computational aspects of the methods, as well as the similarities and dissimilarities in the predictions of different methods with respect to data characteristics. This is an important issue when considering the combination of different methods in an ensemble model framework (Dietterich, 2000).

References

1. Breiman, L., Friedman, J., Olshen, R. and Stone, C.J. (1984), *Classification and Regression Trees*, Chapman and Hall, New York.
2. Cohen, J. (1988), *Statistical Power Analysis for the Behavioral Sciences*, second edition, Academic Press, New York.
3. Dietterich, T.G. (2000), Ensemble methods in machine learning. In: *First International Workshop on Multiple Classifier Systems (Lecture Notes in Computer Science)*. Kittler, J. and Roli, F. (eds.), Springer Verlag, New York, pp. 1–15.
4. Dietterich, T.G. and Bakiri, G. (1995), Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research* 2, 263–286.
5. Doumpos, M. and Zopounidis, C. (2002), *Multicriteria Decision Aid Classification Methods*, Kluwer Academic Plublishers, Dordrecht.
6. Duda, R.O., Hart, P.E. and Stork, D.G. (2001), *Pattern Classification*, second edition, John Wiley, New York.
7. Freed, N. and Glover, F. (1986), Evaluating alternative linear programming models to solve the two-group discriminant problem, *Decision Sciences* 17, 151–162.
8. Joachimsthaler, E.A. and Stam, A. (1988), Four approaches to the classification problem in discriminant analysis: An experimental study, *Decision Sciences* 19, 322–333.
9. John, G.E., Kohavi, R. and Pfleger, K. (1994), Irrelevant attributes and the subset selection problem. In: *Machine Learning: Proceedings of the 11th International Conference*. Cohen, W.W. and Hirsh, H. (eds.), Morgan Kaufmann, San Francisco, 121–129.
10. Lam, K.F. and Moy, J.W. (1997), An experimental comparison of some recently developed linear programming approaches to the discriminant problem, *Computer and Operations Research* 24, 593–599.
11. Östermark, R. and Höglund, R. (1998), Addressing the multigroup discriminant problem using multivariate statistics and mathematical programming, *European Journal of Operational Research* 108, 224–237.
12. Pawlak, Z. (1982), Rough sets, *International Journal of Information and Computer Sciences* 11, 341–356.
13. Ripley, B.D. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.
14. Rubin, P.A. (1990), A comparison of linear programming and parametric approaches to the two-group discriminant problem, *Decision Sciences* 21, 373–386.
15. Sandri, L. and Marzocchi, W. (2004), Testing the performance of some nonparametric pattern recognition algorithms in realistic cases, *Pattern Recognition* 37, 447–461.
16. Schölkopf, B. and Smola, A.J. (2002), *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge.
17. Stam, A. (1997), Nontraditional approaches to statistical classification: Some perspectives on L_p-norm methods, *Annals of Operations Research* 74, 1–36.
18. Specht, D. (1990), Probabilistic neural networks, *Neural Networks* 3, 109–118.

19. Sueyoshi, T. (2006), DEA discriminant analysis: methodological comparison among eight discriminant, *European Journal of Operational Research* 169, 247–272.
20. Vapnik, V.N. (1998), *Statistical Learning Theory*, John Wiley, New York.
21. Wong, M.A. and Lane, T. (1983), A *k*th nearest neighbor clustering procedure, *Journal of the Royal Statistical Society B* 45, 362–368.
22. Yandell, B.S. (1977), *Practical Data Analysis for Designed Experiments*, Chapman and Hall, London.
23. Zopounidis, C. and Doumpos, M. (2002), Multicriteria classification and sorting methods: a literature review, *European Journal of Operational Research* 138, 229–246.