Credit Cards Scoring with Quadratic Utility Functions

VLADIMIR BUGERA[a], HIROSHI KONNO[b] and STANISLAV URYASEV[a,*]

[a] *ISE, Risk Management and Financial Engineering Lab, University of Florida, PO Box 116595, 303 Weil Hall, Gainesville, FL 32611-6595, USA*

[b] *Department of Industrial and Systems Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan*

ABSTRACT

The paper considers a general approach for classifying objects using mathematical programming algorithms. The approach is based on optimizing a utility function, which is quadratic in indicator parameters and is linear in control parameters (which need to be identified). Qualitative characteristics of the utility function, such as monotonicity in some variables, are included using additional constraints. The methodology was tested with a 'credit cards scoring' problem. Credit scoring is a way of separating specific subgroups in a population of objects (such as applications for credit), which have significantly different credit risk characteristics. A new feature of our approach is incorporating expert judgments in the model. For instance, the following preference was included with an additional constraint: 'give more preference to customers with higher incomes.' Numerical experiments showed that including constraints based on expert judgments improves the performance of the algorithm. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: credit cards scoring; utility functions; linear programming; classification

1. INTRODUCTION

In this paper, we consider a general approach for classifying objects and explain it with credit cards scoring problem. Classification can be defined by a classification function assigning to each object some categorical value called the class number. However, this classification function has a very inconvenient property — it is discontinuous (impossible to use it for classifying new objects). We reduce the classification problem to evaluating a continuous utility function from some general class of functions. This function is used for separating objects belonging to different sets. Values of the utility function for objects from one class should be in the same range. 'The best' utility function in some class is found by minimizing the error of misclassification. Depending upon the class of utility function, it may be a quite difficult problem from optimization point of view. However, if one is looking for a utility function, which is a linear combination of some other functions (possibly non-linear in indicator variables), it can

be formulated as a linear programming problem. Mangasarian *et al.* (1995) used this approach for failure discriminant analysis with linear utility functions (applications to breast cancer diagnosis). This function is linear in control parameters and indicator variables. Zopounidis *et al.* (1997, 1998), Pardalos *et al.* (1997) used a linear utility function for trichotomous classifications of credit card applications. Konno and Kobayashi (2000) and Konno *et al.* (2000) considered utility functions which are quadratic in indicator prameters and linear in control variables. The approach was tested with the classification of enterprises and breast cancer diagnosis. Konno and Kobayashi (2000) and Konno *et al.* (2000) imposed convexity constraints on utility functions in order to avoid disconnectness of the discriminant regions. Similar to Konno and Kobayashi (2000) and Konno *et al.* (2000), we also use the quadratic function in indicator parameters. But instead of convexity constraints, we consider monotonicity constraints reflecting experts' opinions. Extending some ideas by Zopounidis *et al.* (1997, 1998) and Pardalos *et al.* (1997) we consider the multi-class classification with many levelsets of the utility function. Although, with our approach we can classify objects to an arbitrary number of classes, in this paper, we considered the trichotomous (i.e. three classes) classification.

*Correspondence to: Dr. S. Uryasev, ISE, Risk Management and Financial Engineering Lab, University of Florida, PO Box 116595, 303 Weil Hall, Gainesville, FL 32611-6595, USA. E-mail: uryasev@ufl.edu

This paper is focused on a numerical validation of the proposed algorithm. We classified a dataset of credit card applications submitted to a bank in Greece. This dataset was earlier considered in Damaskos (1997) and Zopounidis et al. (1998). We investigated the impact of model flexibility on classification characteristics of the algorithm. We compared the performance of several classes of quadratic and linear utility functions with various constraints. Experiments showed the importance of imposing constraints adjusting 'flexibility' of the model to the size of the dataset. We studied 'in-sample' and 'out-of-sample' characteristics of the suggested algorithms. With our classification approach, we minimize the empirical risk, that is, the error of misclassification on the training set (in-sample error). Nevertheless, the objective of the procedure is to classify objects outside of the training set with minimal error (out-of-sample error). The in-sample error is always no greater than the out-of-sample error. Similar issues are studied in the Statistical Learning Theory, see Vapnik (1998).

Broadly speaking, the classification problem can be referred to problems of Data Mining or Knowledge Data Discovery. During the last 50 years, a wide set of different methodologies was proposed for data discovery. Data mining techniques can be divided into five classes, Bradley et al. (1999): predictive modeling (predicting a specific attribute based on the other attributes in the data), clustering (grouping similar data records into subsets), dependency modeling (modeling a joint probability function of the process), data summarization (finding summaries of parts of the data) and change/deviation detection (accounting for sequence information in data records).

Considered in this paper the credit cards scoring problem is a particular case of a consumer lending problem utilizing financial risk forecasting techniques, see, Thomas (2000). Scoring models are divided into two types: (1) models (or techniques) helping creditors to decide whether or not to grant credit to consumers who applied for credit; (2) behavior scoring models helping to decide how to deal with existing customers. We focus on the first type of scoring models.

In credit scoring, decision on issuing credit for a client is based on application for credit and a report obtained from a credit report agency. Also, information on previous applications and their performance is available; we call this information *in-sample information*. Creditor uses in-sample information together with applicant information to make a decision.

Credit scoring is essentially a way of separating (recognizing) specific subgroups in a population of objects (such as applications for credit), which have significantly different credit risk characteristics. Starting with ideas on discriminating between groups, which were introduced by Fisher (1936), many different approaches were developed using statistical and operational research methods. The statistical tools include discriminant analysis (linear and logistic regressions) and recursive partitioning algorithms (classification and decision trees). The operation research techniques primarily include mathematical programming methods, such as linear programming. Also, several new non-parametric and artificial intelligence approaches were recently developed. They include ubiquitous neural networks, expert systems, genetic algorithms, and the nearest neighborhood methods, see for instance, Thomas (2000).

A common weakness of many credit-scoring approaches is that they do not provide clear explanations of 'reasons' for preferring some objects versus others. Capon (1982) considered this as the main drawback of many credit-scoring algorithms. Also, there are many implementation issues, which need to be addressed before using any credit-scoring model, such as: (1) How to select a sample of previous applicants; (2) How long should be the period of time for the sample set? (3) What proportion of 'goods' and 'bads' should be had in the sample. Henley (1995) discussed some of these issues in his thesis.

A credit scoring classification problem can be defined as a decision process, which has the *input:* answers to the application form questions and various information obtained from the credit reference bureau, and *the output:* separation of applications into 'goods' and 'bads', see Thomas, (2000). The objective of credit scoring is to find a rule that separates 'goods' and 'bads' with the smallest percentage of misclassifications. Note that perfect classification is impossible due to several reasons. For instance, there could be errors in the sample data. Moreover, it is possible that some 'good' applications and 'bad' applications have the exactly the same information in all data fields (i.e. not enough information is available to make a correct decision). The statistical learning theory, see, Vapnik (1998), states that for a model the optimal prediction (i.e. out-of-sample classification with minimal misclassification) is achieved

when the in-sample error is close to the out-of-sample error.

Relatively simple statistical approaches using linear scoring functions (Bayesian decision rule, discriminant analysis, and linear regression) became the most popular for classification problems. The Bayesian decision rule works especially well in the case when the distribution of 'goods' and 'bads' can be described by multi-variate normal distributions with a common covariance matrix; it reduces the problem to the linear decision rule. If covariances of these populations are different, then, it leads to a quadratic decision rule. However, the paper by Titterington (1992) pointed out that in many cases the quadratic scoring function appears to be less robust than the linear one. Fisher (1936) used discriminant analysis to find a linear combination of variables that separates the groups in the best way. His approach does not require an assumption of normality. The other way leading to linear discriminating functions is linear regression. Myers and Forgy (1963) compared regression and discriminant analysis in credit scoring applications. Although, there were many critics of the discriminant and regression analysis (see, Eisenbeis (1978), Hand *et al.* (1996), Capon (1982)), empirical experience shows that these linear scoring techniques are very competitive with other more sophisticated approaches.

The other important method of linear discrimination is logistic regression. Earlier, this approach was associated with computational difficulties of maximum likelihood estimation, but nowadays this is not a problem due to readily available high computing power. Wiginton (1980) was one of the first who applied logistic regression to credit scoring; currently this approach is widely accepted.

Classification trees and expert systems represent another class of approaches. Classification trees typically are used in statistical, artificial intelligence, and machine learning applications. Makowski (1985) was one of the first who suggested using classification trees in credit scoring. Coffman (1986) showed that classification trees perform better than the discriminant analysis when there are interactions between variables.

The paper is organized as follows. Section 2 provides the background for the considered methodology. Section 3 formally describes our classification approach. In this section we show how it can be implemented using linear programming. Section 4 applies the developed methodology to credit cards scoring problems. It describes a

dataset of credit card applications and demonstrates computational results. Section 5 analyses the results of computational experiments. We finalize the paper with the concluding remarks in Section 6.

2. BACKGROUND

This section describes several results that stimulated development of the considered approach.

2.1. Separation by linear surfaces: two classes

Mangasarian *et al.* (1995) considered the failure discriminant approach with the linear utility function (applications to breast cancer diagnosis). Let objects $A_i, i = 1, \ldots, m$, belong to the first class, and objects B_l, $l = 1, \ldots, h$, belong to the second class. Also, let $a_i \in R^n$, $b_1 \in R^n$, be vectors of indicators (characteristics) corresponding to A_i, B_l. If there exists a vector $(c, c_0) \in R^{n+1}$, such that

$$c^{\mathrm{T}} a_i > c_0, \quad i = 1, \ldots, m, \tag{1}$$

$$c^{\mathrm{T}} b_l < c_0, \quad l = 1, \ldots, n, \tag{2}$$

then, we call $H(c, c_0) = \{x \in R^n | c^{\mathrm{T}} x = c_0\}$ a *discriminant hyperplane* (see, Figure 1a).

A discriminant hyperplane may not exist (see, Figure 1b).

If we introduce a boundary with thickness 5 between classes, conditions (1)–(2) can be equivalently rewritten as

$$c^{\mathrm{T}} a_i \geq c_0 + \delta, \quad i = 1, \ldots, m, \tag{3}$$

$$c^{\mathrm{T}} b_l \leq c_0 - \delta, \quad l = 1, \ldots, n. \tag{4}$$

Let us define subspaces

$$H_+(c, c_0) = \{x \in R^n | c^{\mathrm{T}} x > c_0\}, \tag{5}$$

$$H_-(c, c_0) = \{x \in R^n | c^{\mathrm{T}} x < c_0\}. \tag{6}$$

An object A_i, such that $a_i \in H_-(c, c_0)$, is a misclassified object of the first type. Also, an object B_l, such that $b_l \in H_+(c, c_0)$, is a misclassified object of the second type.

Let y_i be the distance of $a_i \in H_-(c, c_0)$ from hyperplane $H(c, c_0 + \delta)$, and z_l, be the distances of $b_l \in H_+(c, c_0 - \delta)$ (see Figure 2). Classification is performed by minimizing a weighted sum of misclassification distances y_i and z_l. This is a
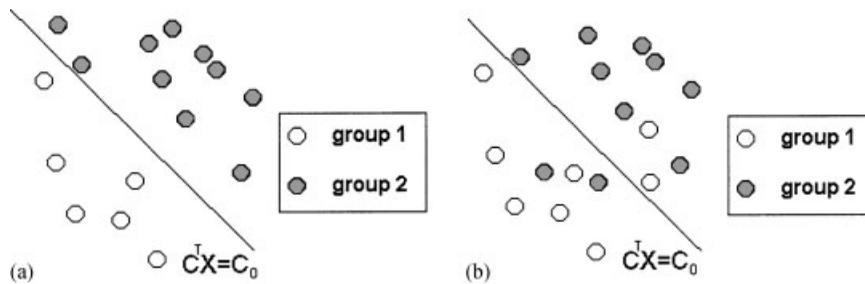
Figure 1. (a) A *discriminant hyperplane* perfectly separates objects belonging to two different classes. (b) A *discriminant hyperplane* may not exist.
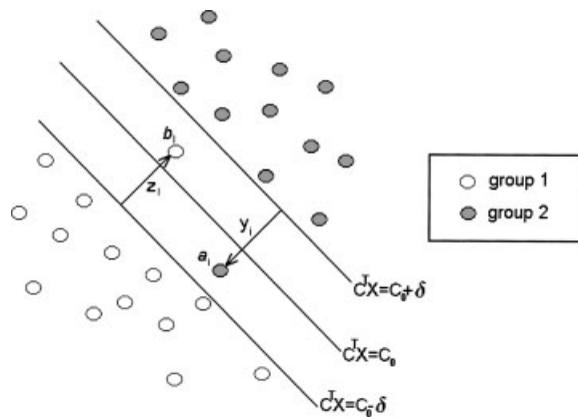

Figure 2. The objects z_l and y_i are misclassified.

linear programming problem

$$\min_{y,z} \quad a\sum_{i=1}^{m} y_i + \beta \sum_{l=1}^{n} z_l$$

$$\text{s.t.} \quad c^{\mathrm{T}}a_i + y_i \geq c_0 + \delta, \quad i = 1,\ldots,m,$$
$$c^{\mathrm{T}}b_l - z_i \leq c_0 - \delta, \quad l = 1,\ldots,n, \qquad (7)$$
$$y_i \geq 0, \qquad i = 1,\ldots,m,$$
$$z_l \geq 0, \qquad l = 1,\ldots,n,$$

where $a > 0$ and $\beta > 0$ are weights associated with the misclassiflcation of the first and the second types of objects.

Problem (7) has an optimal solution $(c^*, c_0^*, y_1^*,\ldots, y_m^*, z_1^*,\ldots, z_n^*)$ and the hyperplane $H(c^*, c_0^*)$ is the discriminant hyperplane with minimal error (for the pair of weights (α, β)). An out-of-sample object g is classified by verifying if it belongs to the set $H_+(c, c_0)$ or to the set $H_-(c, c_0)$. If $g \in H_+(c, c_0)$, we say that g belongs to the first class and if

$g \in H_-(c, c_0)$, it belongs to the second class. In the case when g is on the separating hyperplane, $g \in H(c^*, c_0^*)$, it not clear to which class the object belongs. We can assume, for simplicity, that object g, in this case, belongs to the first class. Equivalently, we can say that the object g belongs to the first class, if the utility function $U(c, c_0, x) = c^{\mathrm{T}}x - c_0$ is non-negative on the object g, i.e. $U(c, c_0, g) \geq 0$. If $U(c, c_0, g) < 0$, the object g belongs to the second class.

2.2. Separation by quadratic surfaces: two classes

To improve the precision of discrimination, Konno and Kobayashi (2000) and Konno *et al.* (2000) proposed using quadratic surfaces instead of hyperplanes. Quadratic separation surfaces are more flexible than linear surfaces, see, Figure 3.

Let $Q(C, c, c_0)$ be a quadratic surface,

$$Q(D, c, c_0) = \{x \in R^n | x^{\mathrm{T}}Dx + c^{\mathrm{T}}x = c_0\}, \ D = (d_{ij}), \tag{8}$$

where D is a $n \times n$ matrix.

The following linear programming problem minimizes the classification error:

$$\min_{y,z} \quad \alpha\sum_{i=1}^{m} y_i + \beta \sum_{l=1}^{n} z_l$$

$$\text{s.t.} \quad a_i^{\mathrm{T}}Da_i + c^{\mathrm{T}}a_i + y_i \geq c_0 + \delta,$$
$$\qquad i = 1,\ldots,m,$$
$$b_l^{\mathrm{T}}Db_l + c^{\mathrm{T}}b_l - z_l \leq c_0 - \delta, \qquad (9)$$
$$\qquad l = 1,\ldots,n,$$
$$y_i \geq 0, \quad i = 1,\ldots,m,$$
$$z_l \geq 0, \quad l = 1,\ldots,n,$$

where $D \in R^{n \times n}$, $c \in R^n$, $c_0 \in R^1$, $y_i \in R^1$, $i = 1,\ldots,m$; $z_l \in R^1$, $l = 1,\ldots,n$ are variables to be determined. Variables y_i and z_l represent
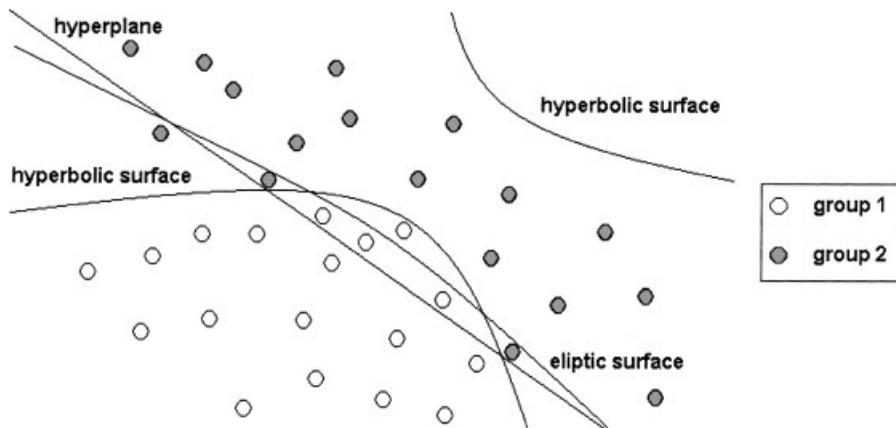
Figure 3. Quadratic separation surfaces are more flexible compared to linear ones.

distances of the misclassified objects from the quadratic surface $Q(D, c, c_0)$. The quadratic utility function has a significantly larger number of variables than linear utility function. Also, in the case of quadratic utility, the separating surface $Q(D, c, c_0)$ may be non-convex (hyperbolic surface), which leads to disconnected discriminant regions. This may poorly represent the actual structure of the financial data, since there usually are monotonic tendencies/preferences with respect to (w.r.t.) variables. To circumvent this difficulty, Konno and Kobayashi (2000) and Konno et al. (2000) imposed a positive semi-defmiteness constraint on matrix D. This results in the following semi-definite programming problem:

$$\min_{y,z} \quad \alpha \sum_{i=1}^{m} y_i + \beta \sum_{l=1}^{n} z_l$$

$$\text{s.t.} \quad a_i^{\mathrm{T}} D a_i + c^{\mathrm{T}} a_i + y_i \geq c_0 + \delta,$$
$$i = 1, \ldots, m,$$
$$b_l^{\mathrm{T}} D b_l + c^{\mathrm{T}} b_l - z_l \leq c_0 - \delta, \quad (10)$$
$$l = 1, \ldots, n,$$
$$y_i \geq 0, \quad i = 1, \ldots, m,$$
$$z_l \geq 0, \quad l = 1, \ldots, n,$$
$$\forall r: \quad r^{\mathrm{T}} D r \geq 0.$$

The last problem is solved by a cutting plane algorithm (Konno et al., 2000). Similar to the linear case, the utility function $U(D, c, c_0, x) = x^{\mathrm{T}} D x + c^{\mathrm{T}} x - c_0$ is used to classify out-of-sample objects. We say that an object g belongs to the first class, if $U(c, c_0, g) \geq 0$; otherwise, g belongs to the second class.

2.3. Separation by linear surfaces: three classes

Zopounidis et al. (1997, 1998) and Pardalos et al. (1997) used linear utility functions in the credit cards scoring problem (classification of credit card applications). Compared to approaches described in Sections 2.1 and 2.2, objects are classified into three rather than two classes. With the utility function, credit card applications are assigned to three classes (trichotomous model): the class of acceptable applications; the class of uncertain applications, for which more investigations are required; and the class of rejected applications (Figure 4). Zopounidis et al. (1997, 1998) considered three models: UTADIS I, UTADIS II and UTADIS III. These models differ only in the form of their penalty function. Coefficients of the separating hyperplanes are calculated with a linear optimization problem, similar to (7).

3. DESCRIPTION OF METHODOLOGY

3.1. Approach

Let us consider a classification problem with J-classes. Suppose that there is a set of objects $I = \{1, \ldots, m\}$ with known classifications. Each object is represented by a point in an n-dimensional Euclidian space R^n. This set of points, $X = \{x^i | i \in I\}$, is called a *training set*. Suppose that decomposition $\{I_j\}_{j=1}^{J}$, $(I_{j_1} \cap I_{j_2} = \varnothing$, $j_1 \neq j_2;$
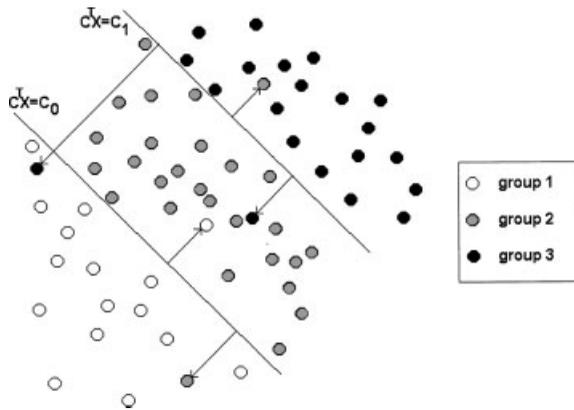
Figure 4. Points are separated by two linear surfaces in three classes.

$I = \bigcup_{j=1}^{J} I_j$) defines the classification of m objects. A point x_i belongs to the class k if $i \in I_k$.

Let us consider a utility function $U(x)$ defined on the set $X = \{x^i | i \in I\}$ and a set of thresholds $U_{TH} = \{u_0, u_1, \ldots, u_{J-1}, u_J\}$ such that $u_{j-1} < u_j$ and $u_0 = -\infty$, $u_J = +\infty$. If the following statement is valid

$$I_j = \{i | u_{j-1} < U(x^i) \leq u_j, \quad i \in I\}, \qquad (11)$$

then, the function $U(x)$ and set U_{TH} define the decomposition $\{I_j\}_{j=1}^{J}$, and, consequently, classify the training set. Let us consider the following problem: for a given classification $\{I_j\}_{j=1}^{J}$ find a utility function and a set of thresholds that define this classification. If there are no identical points in the set X, we can easily construct a classification utility function for any decomposition. For example, let us consider the discrete-valued function $U(x^i) = j : i \in I_j$ and the thresholds $u_j = j + 0.5, j = 0, \ldots, J$. Although this function perfectly classifies the training dataset, it does not estimate the class number of objects outside of the training set.

A search of a classification utility function in a predefined class, such as the earlier introduced linear or quadratic, is a more complicated problem (compared to the discrete utility function considered above). Let a class of utility functions $U^\gamma(x)$ be parameterized by a vector parameter $\gamma \in \Xi$. Suppose that the training set $X = \{x^i | i \in I\}$ is split into classes $\{I_j\}_{j=1}^{J}$ Let us denote by $F(\sigma^+, \sigma^-)$ the total penalty for all misclassified data points. For the training set X, the following optimization

problem finds 'the best' utility function $U^\gamma(x)$ in the prespecified class of functions:

$$\min_{\gamma, u, \sigma^-, \sigma^+} F(\sigma^-, \sigma^+)$$
$$u_{j-1} - \sigma_i^- + \delta \leq U^\gamma(x^i) \leq u_j + \sigma_i^+, \ i \in I_j,$$
$$j = 1, \ldots, J;$$
$$u_j - u_{j-1} \leq s, \ j = 2, \ldots, J-1;$$
$$\sigma_i^+ \geq 0, \ \sigma_i^- \geq 0 \ i = 1, \ldots, I;$$
$$\gamma \in \Xi. \qquad (12)$$

This function is nondecreasing w.r.t. classification errors σ_i^-, σ_I^+, $i \in I$. Large deviations from a perfect classification imply large penalties. Various penalty functions are considered in literature. Here, we consider the linear penalty function

$$F(\sigma^-, \sigma^+) = \sum_{i=1}^{l} (\alpha_i^- \sigma_i^- + \alpha_i^+ \sigma_i^+), \ \alpha_i^-, \alpha_i^+ \geq 0,$$
$$i = 1, \ldots, I, \qquad (13)$$

where α_i^-, α_i^+ are penalty coefficients for datapoint i. In this paper we assume $\alpha_i^- = \alpha_i^+ = 1$ for $i = 1, \ldots, I$.

Let an object x^i belong to the class I_j. The mathematical programming problem (12) implies that $\sigma_i^+ = \max\{0, U^\gamma(x^i) - u_j\}$ and $\sigma_i^- = \max\{0, u_{j-1} + \delta - U^\gamma(x^i)\}$. If the utility function $U^\gamma(x)$ on x^i from class I_j exceeds the upper threshold u_j, then the error σ_i^+ equals the difference between $U^\gamma(x^i)$ and the upper threshold u_j; otherwise, σ_i^+ equals zero. Similar, if the value of the utility $U^\gamma(x^i)$ on x^i is below $u_{j-1} + \delta$, then σ_i^- equals the difference between $u_{j-1} + \delta$ and $U^\gamma(x^i)$. With a small parameter δ, we introduced a separation area between class layers in order to uniquely identify the class of an object.

In problem (12), we optimize the penalty function w.r.t. the parameter $\gamma \in \Xi$, which parameterizes the class of utility functions. Further, we consider two types of utility functions:

(a) *Linear function* w.r.t. indicator vector parameter x and linear w.r.t. control vector variable c

$$U^c(x) = c^T x = \sum_{k=1}^{n} c_k x_k, \quad c \in R^n \qquad (14)$$

(b) *Quadratic function* w.r.t. indicator vector parameter x and linear w.r.t. control

variables D, c

$$U^{D,c}(x) = Dx + c^{\mathrm{T}}x = \sum_{k=1}^{n}\sum_{l=1}^{n}d_{kl}x_k x_l$$
$$+ \sum_{k=1}^{n}c_k x_k, \qquad (15)$$

where D is a matrix $n \times n$ and $c \in R^n$.

3.2. Classification with a linear utility function

With linear penalty (13) and linear utility function (14), problem (12) can be rewritten in the following form:

$$\min_{c,u,\sigma^-,\sigma^+}\sum_{i=1}^{l}(\alpha_{i-1}^-\sigma_{i-1}^- + \alpha_i^+\sigma_i^+)$$

$$\sum_{k=1}^{n}c_k x_k^i - u_{j-1} + \sigma_i^- \geq \delta,$$
$$\quad i \in I_j, j = 1, \ldots, J;$$

$$\sum_{k=1}^{n}c_k x_k^i - u_j - \sigma_i^+ \leq 0, \qquad (16)$$
$$\quad i \in I_j, j = 1, \ldots, J;$$

$$u_j - u_{j-1} \leq s, \quad j = 2, \ldots, J-1$$
$$\sigma_i^+ \geq 0 \quad i = 1, \ldots, I;$$
$$\sigma_i^- \geq 0 \quad i = 0, \ldots, I-1;$$
$$c \in C.$$

where C is a feasible set of coefficients of linear utility function $U^c(x) = \sum_{k=1}^{n}c_k x_k$. When C is a polyhedron (a set specified by a finite number of linear inequalities), (16) is a linear programming problem (LP).

3.3. Classification with quadratic utility function

Let us consider a quadratic utility function

$$U^{D',c}(x) = D'x + c^{\mathrm{T}}x = \sum_{k=1}^{n}\sum_{k=1}^{n}d_{kl}'x_k x_l + \sum_{k=1}^{n}c_k x_k.$$

This function has $n^2 + n$ parameters (the matrix D' has n^2 coefficients and the vector c has n coefficients). However, the quadratic function can be equivalently redefined by only $(n^2 + n)/2$ terms. Indeed,

$$U^{D',c}(x) = \sum_{k=1}^{n}d_{kk}'x_k^2 + \sum_{k=1}^{n-1}\sum_{l=k+1}^{n}(d_{kl}' + d_{lk}')x_k x_l$$
$$+ \sum_{k=1}^{n}c_k x_k.$$

With the introduction of the new notations

$$d_k = d_{kk}';$$
$$d_{kl} = d_{kl}' + d_{lk}', \quad k < l$$

the quadratic utility function can be rewritten as follows:

$$U^{D',c}(x) = \sum_{k=1}^{n}d_k'x_k^2 + \sum_{k=1}^{n-1}\sum_{l=k+1}^{n}d_{kl}x_k x_l$$
$$+ \sum_{k=1}^{n}c_k x_k. \qquad (17)$$

Further, we will use this representation of a quadratic function.

With linear penalty (13) and quadratic utility function (17), problem (12) can be rewritten in the following form:

$$\min_{D,c,u,\sigma^-,\sigma^+} F = \sum_{i=1}^{l}(\alpha_i^-\sigma_i^- + \alpha_i^+\sigma_i^+)$$

$$\sum_{k=1}^{n}d_k x_k^2 + \sum_{k=1}^{n-1}\sum_{l=k+1}^{n}d_{kl}x_k x_l \qquad i \in I_j, j = 1, \ldots, J;$$
$$+ \sum_{k=1}^{n}c_k x_k - u_{j-1} + \sigma_i^- \geq \delta$$

$$\sum_{k=1}^{n}d_k x_k^2 + \sum_{k=1}^{n-1}\sum_{l=k+1}^{n}d_{kl}x_k x_l \qquad (18)$$
$$\qquad i \in I_j, j = 1, \ldots, J;$$
$$+ \sum_{k=1}^{n}c_k x_k - u_j - \sigma_i^+ \leq 0$$

$$u_j - u_{j-1} \leq s, \qquad j = 2, \ldots, J-1$$
$$\sigma_i^+ \geq 0 \qquad i = 1, \ldots, I;$$
$$\sigma_i^- \geq 0 \qquad i = 1, \ldots, I;$$
$$(d, c) \in C.$$

If C is a polyhedron, then all constraints in problem (18) are linear w.r.t. variables D, c, u, σ^- and σ^+. Problem (18) is a linear programming problem which can be solved using standard techniques.

3.4. Data format

With our approach, objects are points in the space of indicator (characteristic) variables. For example, in the credit scoring problem considered in this paper, a credit card application is an object and 'marital status' filed in the application is an indicator variable. For example, we can code this field as follows: 0 = divorced, 1 = single, and 2 = married/widowed (we make no distinction between married and widowed persons).

Suppose we have a set of m objects, and each object is described by a vector $x = (x_1, \ldots, x_n)$ of indicator variables (or indicators by short). The set of objects is a set of points $X = \{x_1 | i \in I\}$ in n-dimension space, where $I = \{1, \ldots, m\}$. Let Ω_i be a set of all possible values of ith indicator. Each indicator can take integer or continuous values. Supposing that the marital status is the third indicator, the set of possible values of this indicator is $\Omega_3 = \{0,1,2\}$ and for widowed person $x_3 = 2$. In the case of discrete indicators, we suppose that the codification begins with 0 and takes successive integers. In some cases, a continuous variable may be replaced by a set of discrete indicators. In particular, if the utility function 'is not sufficiently flexible' in some variable, a continuous indicator corresponding to this variable may be replaced by a set of discrete indicators. Suppose that we want to discretize the ith indicator variable of an object $x = (x_1, \ldots, x_n)$. Let us suppose that after rescaling the ith indicator takes values in the interval $(0,k)$, where k is a positive integer variable. We can replace the ith component of the vector x_i by a subvector $\vec{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k-1})$, such that

$$x_{i,j} = \begin{cases} 1 & \text{if} \quad x_i \geq j, \\ 0 & \text{if} \quad x_i < j. \end{cases} \tag{19}$$

In other words, we split the continuous indicator into a set of discrete indicators. There are many alternative ways to discretize a continuous indicator. For example, for a continuous indicator, such as age, we can introduce the following approximation:

constraints may reduce excessive model flexibility. Konno and Kobayashi (2000) and Konno *et al.* (2000) imposed convexity constraints on the quadratic function to make it convex. Here, we consider a different type of constraints, which is driven by expert judgments. We consider the utility function, which is *monotonic* w.r.t. some indicator variables. In finance applications, monotonicity w.r.t. some indicators follows from 'engineering' considerations. For instance, in the credit scoring problem, we expect that the utility function is monotonic w.r.t. income level of an applicant. Our numerical experiments showed that constraints, such as monotonicity constraints, may significantly improve the out-of-sample performance of the algorithm.

For a linear utility function, the condition on monotonicity w.r.t. an indicator variable is quite simple. The linear function $U^c(x) = c^{\mathrm{T}} x = \sum_{i=1}^{n} c_i x_i$ increases w.r.t. x_i if and only if $c_i > 0$.

For example, for a credit cards scoring problem, we can impose a monotonicity constraint on the variable corresponding to the marital status indicator. In order to incorporate our preference

divorced < single < married/widowed,

we can impose a monotonicity constraint on the linear utility function. Suppose that the marital status is the third indicator. The utility function should be increasing w.r.t. x_3, i.e. $c_3 > 0$. This monotonicity condition can be easily incorporated into problem (16) by adding the non-negativity linear constraint, $c_3 \geq 0$.

Compared to a linear function, it is more difficult to impose monotonicity constraints on

Age	< 18	18–20	21–24	25–29	30–55	56–59	60–75	76–84	> 84
Code	0	1	2	3	4	5	6	7	8

Further, using formula (19), we can convert the discrete representation presented in the table to eight discrete indicators.

3.5. Monotonicity constraints

The considered utility functions, especially the quadratic utility function, may be too flexible (have too many degrees of freedom) for datasets with a small number of datapoints. Imposing additional

quadratic functions (w.r.t. indicator variables). We consider a subclass of monotonic quadratic functions with non-negative elements of matrix D and vector C. This function is monotonic w.r.t. each variable on $R^+ = \{x \mid x_i \geq 0 \quad \forall i \in \{1, \ldots, n\}\}$. These functions are called *rough monotonic functions*. We should remember that in order to use such a class of functions, indicator variables must be non-negative and of increasing preference.

Although this subclass represents only small part of variety of monotonic functions on R^+, it posses good predictive capabilities for considered applications.

Another approach is to impose monotonicity constraints for the points from a given dataset X. Suppose that we want to incorporate information that the utility function is increasing w.r.t. variable x_i. The monotonicity condition for a continuous function is

$$\frac{\partial U(x)}{\partial x_i} \geq 0. \tag{20}$$

For quadratic utility function (17), this constraint can be written as follows:

$$c_i + 2d_i x_i + \sum_{k=1}^{i-1} d_{ki} x_k + \sum_{k=i+1}^{n} d_{ik} x_k \geq 0. \tag{21}$$

If we implement (21) for all points from dataset $X = \{x^j | j \in \{1, \ldots, m\}\}$, we get the set of linear constraints

$$c_i + 2d_i x_i^j + \sum_{k=1}^{i-1} d_{ki} x_k^j + \sum_{k=i+1}^{n} d_{ik} x_k^j \geq 0,$$
$$j = 1, \ldots, m \tag{22}$$

that makes the utility function to be close to monotonic.

Using a similar approach, we can impose a more general type of monotonicity. Suppose that we know that some set of points $Y = \{y^i | i \in \{1, \ldots, r\}\}$ is ordered according to our preference, and we want to find a utility function that is monotonic w.r.t. this preference on a set Y. We can impose the following constraints on the utility function:

$$U(x^1) \leq U(x^2) \leq \cdots \leq U(x^r) \tag{23}$$

for quadratic utility function (17), these constraints are presented as

$$\sum_{k=1}^{n} d_k x_k^i x_k^i + \sum_{k=1}^{n-1} \sum_{l=k+1}^{n} d_{kl} x_k^i x_l^i + \sum_{k=1}^{n} c_k x_k^i$$

$$\leq \sum_{k=1}^{n} d_k x_k^{i+1} x_k^{i+1} + \sum_{k=1}^{n-1} \sum_{l=l+1}^{n} d_{kl} x_k^{i+1} x_l^{i+1}$$

$$+ \sum_{k=1}^{n} c_k x_k^{i+1}, \quad i = 1, \ldots, m. \tag{24}$$

Although these constraints are quadratic w.r.t. indicator variables, they are linear w.r.t. parameters we want to find.

3.6. Non-monotonicity

While, for some indicators, we can incorporate monotonicity constraints, for others it may be beneficial to use other properties, such as convexity and concavity. Let us consider the continuous indicator 'age', a_1, mentioned earlier. Suppose that middle age is preferable to young and old age, and the preferences is specified as

$$\{0-18\} < \{18-20\} < \{21-24\} < \{25-29\}$$
$$< \{30-55\},$$
$$\{85-\infty\} < \{76-84\} < \{60-75\} < \{56-59\}$$
$$< \{30-55\}.$$

The preference for age is expressed by the non-monotone criterion over the whole age range. We can incorporate this into the credit-scoring model in two ways.

(1) We can assume that the utility function is concave w.r.t. variable x_1. This can be assured by calculating the second derivative of the utility function w.r.t. x_1 and including the following constraint in LP (18):

$$\frac{\partial^2}{\partial x_1^2} U(x) \leq 0.$$

If the utility function is quadratic in x, the resulting optimization problem is linear in control variables.

(2) We can change coding of the age datafield as follows:

Age	< 18	18–20	21–24	25–29	30–55	56–59	60–75	76–84	> 84
Code	0	1	2	3	4	3	2	1	0

With this coding, the utility function is monotonic w.r.t. the age field.

4. CREDIT CARDS SCORING

We explain our approach with the credit cards scoring problem, which earlier was considered in Bradley *et al.* (1999) and Fisher (1936).

4.1. Dataset and problem statement

The dataset consists of 150 credit card applications submitted to the National Bank of Greece during the period 1995–1996. Each credit card application includes 25 fields. However, only seven fields selected by a credit manager were included in the analysis[¶] (see, Table I):

(1) *Family status* (three-point scale): divorced (least preferred), single, married/widowed (most preferred).
(2) *Profession* (10-point scale): from 0 — unemployed (least preferred) to 9 — lawyers, doctors, teachers or professors (most preferred).
(3) *Business telephone number* (three-point scale): not available (least preferred), not required, available (most preferred). For some professions, it is difficult or impossible to declare a business phone number (e.g. retired applicant, shipment, etc.). These professions were included in the 'not required' field.
(4) *Residence* (three-point scale): rental (least preferred), house-guest/homestead, privately owned (most preferred).
(5) *Bank account for credit card payments* (binary): bank account for credit card payments is not available (least preferred), available (most preferred).
(6) *Age* (multiple-values scale): preference is expressed by a nonmonotone criterion. The least preferred are young and old customers,

and the most preferred are middle-aged customers. Applications are split into the following groups,

$$\{0-18\} < \{18-20\} < \{21-24\}$$
$$< \{25-29\} < (30-55),$$
$$\{85-\infty\} < \{76-84\} < \{60-75\}$$
$$< \{56-59\} < \{30-55\}.$$

(7) *Years in business* (continuous scale): from 0 (least preferred) to 35 (most preferred). Unlike the 'age-field', preference is expressed by a monotonic criterion.

Although the annual income of the applicants could be considered as an important criterion in the credit card evaluation, it was not considered in this case study because of the two following reasons:

(a) Credit cards are commonly used to cover daily expenses, which do not require a significant amount of credit. In such cases, an income of a credit card applicant does not have a significant effect on the acceptance or rejection of a credit card application. However, it could be a significant factor for high credit limits.
(b) The annual income of an applicant in Greece may not provide a reliable measure representing the true financial status of credit card applicants.

Table I summarizes the fields and their values.

We know the class of each credit card application. The 150 applications are split into three classes:

1. *Accepted applications (Class I_1):* 74 applications.
2. *Uncertain applications (Class I_2):* 52 applications that were considered difficult to evaluate by a local loan officer of the bank.
3. *Rejected applications (Class I_3):* 24 applications.

This trichotomous (i.e. three group) classification is more flexible than the two-group classification. It allows a credit analyst to define an uncertain area, indicating that some credit card applications need to be reconsidered in the second stage of the analysis in order to determine their characteristics.

[¶]Compared to Damaskos (1997) and Zopounidis *et al.* (1998), we made two minor modifications in the codification of the data. This does not affect the results, but simplifies the approach. First, we have measured the profession field on the 10-point scale (fractional values are considered in Damaskos (1997) and Zopounidis *et al.* (1998)). Second, the minimal value for all fields is equal to 0 (minimal value equals 1 in Damaskos (1997) and Zopounidis *et al.* (1998)).

Table I. Codification of data in the credit cards scoring case study

Field #	1	2	3	4	5	6	7
Field/value	Family status	Profession	Business phone	Residence	Bank account	Age	Years in business
0 (Less Preferred)	Divorced	Unemployed	Not available	Rental	Not available	$<18, >84$	Continuous criterion
1	Single	Handicraftsmen, businessmen	Not required	Guests, homestead	Available	$\{18\text{--}20\}, \{76\text{--}84\}$	
2	Married divorced	Restaurateurs, butchers, bakers	Available	Privately owned		$\{21\text{--}24\}, \{60\text{--}57\}$	
3		Merchants				$\{25\text{--}29\}, \{56\text{--}59\}$	
4		Insurers, cosmeticians, wine makers				$\{30\text{--}55\}$	
5		Farmers, technicians, shipmen, constructors					
6		Employees in private sector					
7		Engineers, employees in the public sector, military officers					
8		Retired					
9 (Most preferred)		Lawyers, pharmacists, doctors, teachers, professors					

Table I maps applications to the set of points $X = \{x^1, \ldots, x^{150}\}$ in R^7 space. This set is decomposed into three subsets $I_1 \cup I_2 \cup I_3$, $I_{i \neq j} \cap I_j = \emptyset$. With our approach, we want to find thresholds u_1, u_2 and a utility function $U(x)$ that classifies the set $X = \{x^1, \ldots, x^{150}\}$, i.e.

{Application i is accepted}

$\Leftrightarrow i \in I_1 \Leftrightarrow u_2 \leq U(x^i)$,

{Decision on application i is postponed}

$\Leftrightarrow i \in I_2 \Leftrightarrow u_1 \leq U(x^i) < u_2$,

{Application i is rejected}

$\Leftrightarrow i \in I_3 \Leftrightarrow U(x^i) < u_1$.

We consider utility functions, which are linear in decision variables and linear or quadratic in indicator variables (fields in Table I). To find a utility function classifying the dataset with mini-mal error, we solve (16) or (18) linear programming problems.

4.2. Numerical experiments

To compare classifications capabilities of different models, we considered several classes of utility functions in combination with different codification schemes, see Table II. Discretized data is obtained by converting the original 'continuous' variable into set of 'discrete' variables according to the rule discussed in Section 3.4. For the models with linear utility function we consider monotonic constraints, for the models with quadratic utility function we consider 'rough' monotonicity constraints discussed in Section 3.5.

To estimate the performance of the considered models, we have used the 'leave-one-out' cross validation scheme. For discussion of this scheme and other cross validation approaches, see, for instance Efron and Tibshirani (1994). Let us denote by $m = 150$ the number of applications in

Table II. List of considered models, classes of utility functions, and data codifications

Model identificator	Type of data in fields 1–4	Type of utility function	Type of monotonicity
CLN	Non-discretized	Linear	No
CLM	Non-discretized	Linear	Monotonic
CQN	Non-discretized	Quadratic	No
CQM	Non-discretized	Quadratic	Rough
DLN	Discretized	Linear	No
DLM	Discretized	Linear	Monotonic
DQN	Discretized	Quadratic	No
DQM	Discretized	Quadratic	Rough

Table III. Calculation results for the considered models (AVR = average value of misclassification percentage, STDV = standard deviation of misclassification percentage)

	In-sample		Out-of-sample	
	AVR (%)	STDV (%)	AVR (%)	STDV (%)
CLN	4.04	0.19	10.00	30.10
CLM	5.29	0.33	10.00	30.10
CQN	0	0	11.33	11.33
COM	2.65	0.22	10.00	30.10
DLN	1.98	0.36	8.67	28.23
DIM	3.33	0.26	10.00	30.10
DON	0.01	0.11	18.00	38.55
DQM	4.06	0.57	7.33	26.16

the dataset. For each model (defined by the class of the utility function and codification according to Table II), we conducted m experiments. By excluding applications, x_i, one-by-one from the set X, we constructed m training sets,

$$Y_i = X \setminus \{x_i\}, \quad i = 1, \ldots, m.$$

For each set Y_i, we solved (16) or (18) optimization problems and found optimal parameters of the utility function $U(x)$ and thresholds U_{TH}. Also, we computed the number of misclassified applications M_i from the set Y_i. Let us introduce the variable

$$P_i = \begin{cases} 0, & U(x) \text{ correctly classified } x_i, \\ 1, & \text{otherwise.} \end{cases} \quad (26)$$

Denote by $E_{\mathrm{in-sample}}$ an estimate of in-sample error, which is calculated as follows:

$$E_{\mathrm{in\text{-}sample}} = \frac{1}{m} \sum_{i=1}^{m} \frac{M_i}{(m-1)} = \frac{1}{m(m-1)} \sum_{i=1}^{m} M_i. \quad (27)$$

In the last formula, the ratio $M_i/(m-1)$ estimates the probability of in-sample misclassification in the set Y_i. We averaged these probabilities to estimate the in-sample error.

Denote by $E_{\mathrm{out-of-sample}}$ an estimate of the error. It is calculated as the ratio of the total number of misclassified out-of-sample objects in m experiments to the number of experiments

$$E_{\mathrm{out\text{-}of\text{-}sample}} = \frac{1}{m} \sum_{i=1}^{m} P_i. \quad (28)$$

For the experiment we have chosen $\delta = 1$, $s = 0.00001$. The experiments were conducted with a Pentium III, 700 MHz in C/C++ environment. Linear programming problems were solved by the CPLEX 7.0 package. Calculation results are provided in Table III. Table IV represents distribution of prediction results.

Figure 5 represents in-sample and out-of-sample errors for different models. Models with quadratic utility functions (no constraints), CQN and DQN, have zero in-sample error. Among considered models, these models are the most 'flexible'. The smallest out-of-sample error is obtained by DQM-algorithm with a rough monotonic quadratic utility function. Also, DLN has relatively small out-of-sample error. Figure 5 shows that the best forecasting results (out-of-sample characteristics) achieved by models that fit into data with the large values of misclassification error (in-sample characteristics). Moreover only DQM model does not make 2-class mistake, when the application from the third class is predicted to be in the first class.

It is worth mentioning that discretizing continuous fields dramatically increases the number of variables in optimization problem (12) and the computing time. Imposing constraints on utility functions also increases the computing time. However, for all considered optimization problems, the computing time is less than 1 second. The CPLEX LP solver, which was used in this study, can solve large-scale problems with thousands of datapoints.

5. ANALYSIS

The considered models are defined by the class of the utility functions and the codification of the training data. We considered linear and quadratic utility functions with and without monotonicity constraints on coefficients. The main characteris-

Table IV. Distribution of mispredicted applications

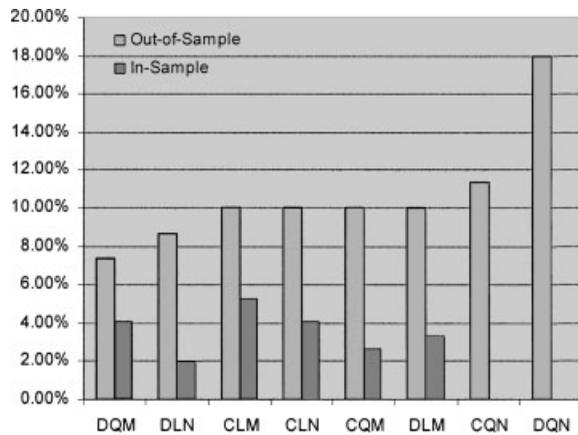|	Class								
Actual	1	1	1	2	2	2	3	3	3
Predicted	1	2	3	1	2	3	1	2	3
Model	Number of application								
CLN	21	3	0	3	46	3	1	5	68
CLM	21	3	0	3	46	3	1	5	68
CQN	21	3	0	5	43	4	1	4	69
COM	22	2	0	3	45	4	1	5	68
DLN	20	4	0	1	48	3	1	4	69
DLM	20	4	0	4	45	3	1	3	70
DON	17	7	0	9	38	5	1	5	68
DQM	23	1	0	2	47	3	0	5	69

Figure 5. In-sample and out-of-sample errors for different models (C/D = continuous/discrete data, Q/L = quadratic/linear utility function, M/N = with/without monotonicity constraints).

Figure 6. In-sample and out-of-sample performance *vs.* the size of training dataset. Increasing of the size of the training set leads to a larger in-sample error and a smaller out-of-sample error. For sufficiently large datasets, the in-sample and out-of-sample errors are quite close (the model is saturated with data).

tics of the classification model are in-sample and out-of-sample errors. Detail discussion of the theoretical background of the classification algorithms (in-sample *vs* out-of-sample) is beyond the scope of this paper. Readers interested in these issues can find relevant information, for instance, in the book by Vapnik (1998). Theoretically, classification models demonstrate the following characteristics, see Figure 6. For small training sets, the model fits the data with 0 in-sample error, whereas the expected out-of-sample error is large. With an increase of the size of the training set, the expected in-sample error diverges from zero (the model cannot exactly fit the training data). Increasing the size of the training set leads to a larger expected in-sample error and a smaller
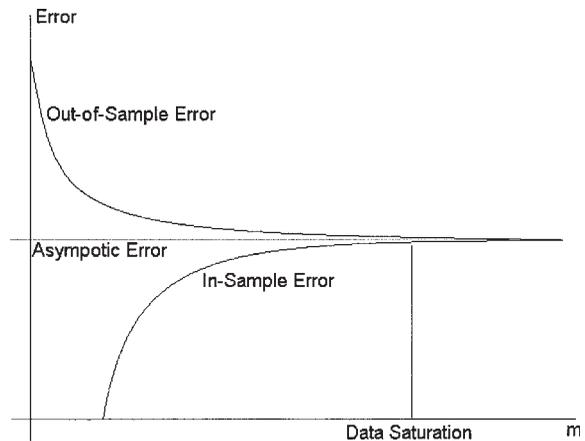
expected out-of-sample error. For sufficiently large datasets, the in-sample and out-of-sample errors are quite close. In this case, we say that the model is saturated with data.

We say that the class of models A is more flexible than class of models B if class A includes class B. For instance, models with quadratic utility functions are more flexible than models with linear utility functions. Including constraints on the utility function reduces the flexibility of the class. Figure 7 illustrates theoretical in/out-of-sample characteristics for two classes of models with different flexibilities. For small training sets, the
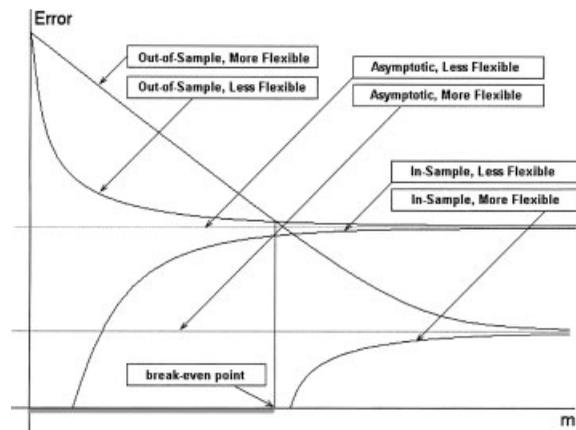
Figure 7. In-sample and out-of-sample performance for different models: for small training sets, the less flexible model gives a smaller out-of-sample error (compared to the more flexible model) and, consequently, predicts better than the more flexible model. The more flexible model outperforms less flexible models for large training sets.

less flexible model gives a smaller expected out-of-sample error (compared to the more flexible model) and, consequently, predicts better than the more flexible model. However, the more flexible models outperform the less flexible models for large training sets (more flexible models have a smaller expected out-of-sample error compared to less flexible models). This is because the less flexible model requires less data for saturation than the more flexible model.

Let us discuss the calculation results obtained with the models considered in this case study. Table III shows that for models with quadratic utility function, the in-sample error is smaller than that for models with linear function (with similar constraints and codification): CQN *vs* CLN, CQM *vs* CLM, DQN *vs* DLN, DQM *vs* DLM. Moreover, in-sample error equals zero for models DQN and CQN. However, out-of-sample error is greater for models DQN and CQN than for the models DLN and CLN, respectively. The model with the quadratic utility function is 'more flexible' than the model with the linear utility function. The former fits the data better (smaller in-sample error). However, a small in-sample error does not guarantee a good out-of-sample classification, as it is in the case of models DQN and CQN. This reflects saturation characteristics of a model: a more flexible model may need a quite large training dataset to provide a good prediction,

while a less flexible model is saturated with data more quickly than more flexible model. Therefore, for small datasets, the less flexible models tend to outperform (out-of-sample) more flexible models. However, for large datasets, more flexible models outperform (out-of-sample) less flexible models.

In addition to the type of the utility function, the constraints and the data format contribute to the flexibility of a model. A model becomes less flexible if we impose additional constraints on the utility function. The experiments confirm this statement: the models without constraints have better in-sample characteristics (smaller in-sample error) than the same models with constraints. The other component of flexibility of a model is data format: models with discrete data are more flexible than ones with continuous data, because many discrete fields represent one continuous field.

Finalizing the discussion, we want to emphasize that choice of class of the utility functions plays a critical role for good out-of-sample performance of the algorithm. An excessive flexibility of the model may lead to 'overfitting' and poor prediction characteristics of the model.

6. CONCLUDING REMARKS

We have studied a general methodology for classifying objects. It is based on finding an 'optimal' classification utility function belonging to a prespecified class of functions. We have considered linear and quadratic utility functions (in indicator parameters) with monotonicity constraints. The constraints on utility functions were motivated by expert judgments. Selecting a proper class of the utility functions and imposing constraints plays a critical role in the success of the suggested methodology. For small training sets, flexible models perform quite poorly and reducing flexibility (imposing constraints) may improve forecasting capabilities of the model. With constraints, we can adjust the flexibility of the model to the size of the training dataset. For considered classes of utility functions, the optimal utility function can be found using linear programming techniques. Linear programming leads to fast algorithms, which can be used for large datasets.

We have applied the developed methodology to a credit cards scoring problem. The case study was performed with a dataset of credit card applications submitted to the National Bank of Greece. We found that for the considered dataset, the best

out-of-sample characteristics (smallest out-of-sample error) are delivered by quadratic utility functions with positivity constraints on the control variables (coefficients of the utility function). These constraints make the utility function monotone w.r.t. indicator variables. The results of numerical experiments are in line with the general considerations about the comparative performance of models having different flexibilities.

This study was focused on the evaluation of computational characteristics of the suggested approach. Although the obtained results are data specific, we can conclude that the approach leads to quite robust classification techniques. By changing the flexibility of the model we can handle a wide range of datasets, from very small to extremely large. In the future, we plan to conduct a theoretical study, investigating the performance characteristics of the suggested models.

REFERENCES

Bradley PS, Fayyad UM, Mangasarian OL. 1999. Mathematical programming for data mining: formulations and challenges. *INFORMS Journal on Computing* **11**(3): 217–238.

Capon N. 1982. Credit scoring systems: a critical analysis. *Journal of Marketing* **46**: 82–91.

Coffman JY. 1986. The proper role of tree analysis in forecasting the risk behavior of borrowers. Management Decision Systems, Atlanta, MDS Reports 3,4,7.

Damaskos XS. 1997. Decision models for the evaluation of credit cards: application of the multicriteria method ELECTRE TRI. Master Thesis, Technical University of Crete, Chania, Greece (in Greek).

Eisenbeis RA. 1978. Problems in applying discriminant analysis in credit scoring models. *Journal of Banking and Finance* **2**: 205–219.

Efron B, Tibshirani RJ. 1994. An Introduction to the bootstrap. Chapman & Hall: New York.

Fisher RA. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**: 179–188.

Hand DJ, Oliver JJ, Lunn AD. 1996. Discriminant analysis when the classes arise from a continuum. *Pattern Recognition* **31**: 641–650.

Henley WE. 1995. Statistical aspects of credit scoring. Ph.D. thesis, Open University.

Konno H, Gotoh J, Uho T. 2000. A cutting plane algorithm for semi-definite programming problems with applications to failure discrimination and cancer diagnosis. Tokyo Institute of Technology, Center for research in Advanced Financial Technology, Working Paper 00-5, May.

Konno H, Kobayashi H. 2000. Failure discrimination and rating of enterprises by semi-definite programming. *Asia-Pacific Financial Markets* **7**: 261–273.

Makowski P. 1985. Credit scoring branches out. *Credit World* **75**: 30–37.

Mangasarian OL. 1965. Linear and nonlinear separation of patterns by linear programming. *Operations Research* **13**: 444–452.

Mangasarian O, Street W, Wolberg W. 1995. Breast cancer diagnosis and prognosis via linear programming. *Operations Research* **43**: 570–577.

Myers JH, Forgy EW. 1963. The development of numerical credit evaluation systems. *Journal of American Statistics Association* **58**(September): 799–806.

Pardalos PM, Michalopoulos M, Zopounidis C. 1997. On the use of multicriteria methods for the evaluation of insurance companies in Greece. In *New Operational Approaches for Financial Modeling*, Zopounidis C. (ed); Physica-Verlag: Berlin-Heidelberg; 271–283.

Thomas LC. 2000. A survey of credit and behavioral scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting* **16**: 149–172.

Titterington DM. 1992. Discriminant analysis and related topics. In *Credit Scoring and Credit Control*, Thomas LC, Crook JN, Edelman DB (eds). Oxford University Press: Oxford; 53–73.

Vapnik V. 1998. *Statistical Learning Theory*. John Wiley & Sons, Inc: New York.

Wiginton JC. 1980. A note on the comparison of logit and discriminant models of consumer credit behaviour. *Journal of Financial and Quantitative Analysis* **15**: 757–770.

Zopounidis C, Pardalos P, Doumpos M, Mavridou T. 1998. Multicriteria decision aid in credit cards assessment. In *Managing in Uncertainty: Theory and Practice*, Zopounidis C, Pardalos P. (eds). Kluwer Academic Publishers: Dordrech; 163–178.

Zopounidis C, Doumpos M. 1997. Preference desegregation methodology in segmentation problems: the case of financial distress. In *New Operational Approaches for Financial Modeling*, Zopounidis C. (ed.); Physica-Verlag: Berlin-Heidelberg; 417–439.