World Academy of Science, Engineering and Technology
International Journal of Economics and Management Engineering
Vol:5, No:3, 2011

An Optimal Subclass Detection Method for Credit Scoring

Luciano Nieddu, Giuseppe Manfredi, Salvatore D'Acunto, and Katia La Regina

Abstract—In this paper a non-parametric statistical pattern recognition algorithm for the problem of credit scoring will be presented. The proposed algorithm is based on a clustering k-means algorithm and allows for the determination of subclasses of homogenous elements in the data.

The algorithm will be tested on two benchmark datasets and its performance compared with other well known pattern recognition algorithm for credit scoring.

Keywords—Constrained clustering, Credit scoring, Statistical pattern recognition, Supervised classification.

Open Science Index, Economics and Management Engineering Vol:5, No:3, 2011 publications.waset.org/51/pdf

I. Introduction

CREDIT scoring has its setting in the wide context of financial credit.

For many years the decision to grant credit has been taken judgmentally by credit analysts, mainly on the basis of their subjective evaluation of the risk of default. The rapid growth in the last years, in both the availability and the use of consumer credit, however, has led to a rise in the use of more formal and objective methods, known as credit scoring, to help credit providers quantify and manage the financial risk involved in granting credit and to decide, quickly and in a more objective way, whether or not to provide credit.

The US sub-prime mortgage crisis has revealed the impact of credit risk decisions on the global e local economy. Various financial institutions have suffered important losses as a result of customer payment defaults. Therefore the development of reliable, efficient and fast credit risk decision support system has a key role in enhancing the evaluation decision by getting faster and more accurate decisions.

Credit scoring can be formally defined as a statistical technique aimed at the prediction of the probability that a loan applicant, or an existing borrower, will default or become delinquent [1].

The credit scoring approach of managing the loan granting was first introduced in the 1940s but evolved and developed significantly in the 1960s, when credit cards were introduced and their use started increasing.

L. Nieddu is with the Faculty of Economics, LUSPIO University, Rome, 00145-Italy e-mail: l.nieddu@luspio.it (see http://www.luspio.it/)

G. Manfredi is full professor of Statistics at the Giustino Fortunato University (see http://www.unifortunato.eu/)

Salvatore D'Acunto is associate professor of Economics at the Giustino Fortunato University

Katia La Regina is assistant professor at the Giustino Fortunato University

Credit scoring realizes a sort of automation of the credit granting process whose benefits concern both the lenders and the borrowers. Advantages deriving from such an automation coincide with enabling credit providers to focus on only information related to credit risk, with increasing speed and consistency of the loan application process, thus enabling financial institutions to quantify the risks associated with granting credits and to determine the interest rate to be charged, etc

In recent years credit scoring has been used for loans concerning homes and small business, as well as for insurance applications and renewals.

The automation of the credit grant process is realized through the construction of credit scoring models, which are statistically derived models aimed at the orientation of the process of decision making for loan applications.

The construction of credit scoring models generally refers to a methodology involving: i) the selection of a *training set*, which is a sample of customers whose solvency is previously known and therefore are classifiable as "good" and "bad", according to their repayment performance over a given period of time; ii) the definition of the data set to be analyzed, generally defined on the basis of the loan applications; iii) the performance of statistical (or other quantitative) analysis of the data to derive a credit scoring model.

Traditional statistical methods are the most common among the several techniques used in the construction of credit scoring models [19]. In particular, Fisher's discriminant analysis[11] was first intensively applied in the construction of the earliest credit scoring models. Logistic regression [15] was then proposed as an alternative, being less restrictive than discriminant analysis.

Other techniques, such as genetic algorithm, k-nearest neighbour (see, e.g. [13],[14], [17]) have been used as well in the construction of credit scoring models, although infrequently. In the recent years data mining techniques and neural networks have been widely applied [20].

Data mining techniques have been performed particularly through the decision tree [16] approach, being the resulting decision trees interpretable and visualizable quite easily.

In this paper we propose a credit scoring methodology based on a statistical pattern recognition algorithm. The proposed algorithm is non-parametric, i.e. no distributional assumptions are required on the data. It is based on a constrained k-means clustering algorithm [3] and allows for subclasses determination inside a class.

Open Science Index, Economics and Management Engineering Vol:5, No:3, 2011 publications.waset.org/51/pdf

The proposed algorithm will be tested on benchmark datasets from the UCI Repository of Machine Learning[1] and its performance compared with other well known pattern recognition techniques for credit scoring, namely: Classification Trees, Logistic Regression and k-nearest neighbour (k-nn) (see, e.g. [16],[13], [15] [11]).

The paper is organized as follows: in section II the proposed algorithm will be presented together with some general remarks on the data on the representativeness of the training sample with respect to the population. In Section III the experimental setup will be described, together with the results of the performances of the four techniques obtained in a cross-validation framework. Then in Section IV some conclusions will be drawn.

II. The Algorithm - T.R.A.C.E.

The algorithm presented in this paper (T.R.A.C.E.: Total Recognition by Adaptive Classification Experiments) is a supervised classification algorithm [14], i.e. a data set of elements with known classes is supposed to be available. The performance of the algorithm is assessed via cross-validation [11].

Given a data set of n pattern vectors in $\Re^p$, assume a partition defined on the dataset, i.e. each pattern vector is assigned to one and only one of k known classes. Such a dataset will be called *piecewise linearly separable*. Let assume a Euclidean norm defined on the dataset and let ψ be a function from $\Re^p$ onto the set $\mathcal{C} = \{1, 2, \ldots, k\}$ which maps each pattern vector $\mathbf{x}_j$, $j = 1, \ldots, n$ into the class $c \in \mathcal{C}$ that it belongs to. T.R.A.C.E. begins computing the barycentre of each class, yielding an initial set of k barycentres. Then the Euclidean distance of each pattern vector from each barycentre is computed. If each pattern vector is closer to the barycentre of its class the algorithm stops, otherwise there will be a non empty set $\mathcal{M}$ of pattern vectors which belong to a class and are closer to a barycentre of a different class. In $\mathcal{M}$ select the pattern vector $\mathbf{x}_w$ that is farthest from the barycentre of its class. This pattern vector will be used as a seed for a new barycentre for class $\psi(\mathbf{x}_w)$. A k-means algorithm [3] will then be performed for all the pattern vectors in class $\psi(\mathbf{x}_w)$ using, as starting points, the set of barycentres for class $\psi(\mathbf{x}_w)$ and the vector $\mathbf{x}_w$. Once the k-means has been performed the set of barycentres will be composed of $k + 1$ elements. The barycentres at the new iterations need not be computed for all classes, but only for class $\psi(\mathbf{x}_w)$, since the barycentres for the other classes have remained unchanged. In the following step the distance of each pattern vector from all the barycentres is computed anew, and so is the set $\mathcal{M}$ (see Figure 1).

If $\mathcal{M}$ is not empty then the pattern vector in $\mathcal{M}$ which is farthest from a barycentre of its own class is once again selected to serve as a seed for a new barycentre. This procedure iterates until the set $\mathcal{M}$ is empty. The

[1]UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

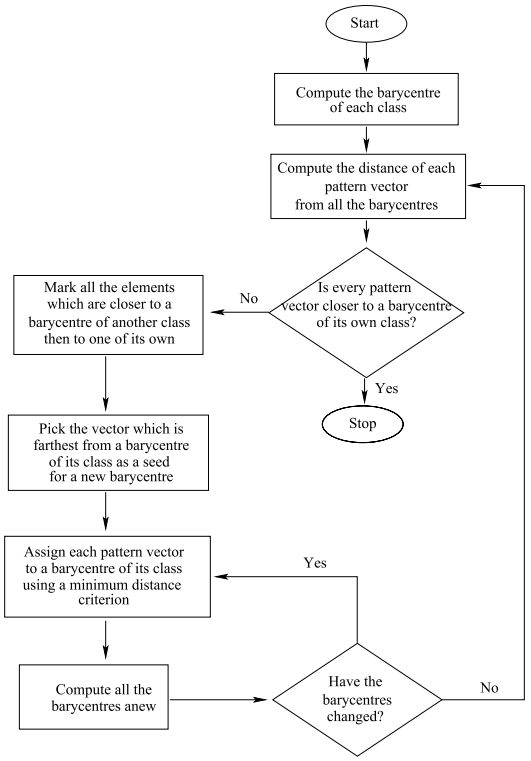


Fig. 1: : T.R.A.C.E.: flow-chart

convergence of the T.R.A.C.E. in a finite number of steps has been proved in various ways (see [7],[8]).

Upon convergence, T.R.A.C.E. yields a set of barycentres which, in the worst case, are in a number equal to the number of elements in the dataset and which has a lower bound in the number of classes.

The aim of this algorithm is to find *subclasses* in the dataset which can be used to classify new vectors of unknown class. It is worth noticing that if the partition defined on the dataset is consistent with the features considered, i.e. if the pattern vectors are linearly separable, then T.R.A.C.E. generates a number of barycentres equal to the number of classes. On the other hand, if the dataset is not linearly separable, then T.R.A.C.E. continues splitting the classes until the subclasses obtained are linearly separable. It is obvious that it can continue splitting until all the subclasses are composed of only one vector (singleton). It will not converge only if two vectors in the dataset belong to different classes and are represented by the same pattern vector [7],[8]. This problem can be easily overcome increasing the dimension of the vector space.

Once T.R.A.C.E. has converged the sets of barycentres can be used to classify new query points assigning the new element to the class of the barycentre it is closest to. If elements from the training set are used as query points then the algorithm always classify them correctly because, once converged, all pattern vectors in the training set are closer to a barycentre of their own class.

Open Science Index, Economics and Management Engineering Vol:5, No:3, 2011 publications.waset.org/51/pdf

A. The data problem

If a classifier is defined according to this method on a piecewise linearly separable training set, a completely correct partition of the training set will be returned if the algorithm is allowed to run until convergence [7].

If the algorithm is applied on a collection of objects, in which there is no clustering structure (i.e. the classification is spurious) the algorithm will still run until convergence, provided that the dataset is piecewise linearly separable, but since the class of each element is spurious, the number of barycentres will be very large, almost equal to the number of elements in the dataset. Such a situation can then be considered as a meter of the spuriousness of the clustering structure in the data. An excessive number of barycentres may arise even in presence of some meaningful relationships in the data; possible causes are:

- the pattern vectors show a clustering structure in the data but this structure is however swamped by noise and so data reduction technique may be useful [12], [5].
- there is a clustering structure in the data, but the number of features collected is not sufficient enough to discriminate adequately, therefore, due to the lack of sufficient information, elements that theoretically belong to different classes show pattern vectors very similar due to the lack of sufficient information. These elements will create, in training, subclasses with few or just one element (singletons). Therefore the presence of many barycentres with just one or very few elements may be considered indicative of the fact that the characteristics collected on the data (pattern vector) are not sufficient to discriminate between classes. If the clustering structure is really present in the data then this problem can be easily overcome by increasing the dimension of the pattern vector through the collection of further information on the data (see e.g. [6], [4]) or using non linear transformation on the pattern vector (see e.g. [2]).
- undersampling of some subclasses in the population.

The latter case can cause problems in verification due to data coherence often caused by a small training sample size or by undersampling in specific subclasses of the population under study. Assume that a particular subclass is underrepresented in the data set, e.g. the data set contains only one instance of that particular subclass. If this item falls in the training sample, then it will be set aside to form a distinct barycentre (singleton), which will be composed only of a single element. Nothing more should happen and, of course, because of the singleton sample the barycentre vector, understood as the mean vector of a set of like objects, will be not a good estimate of the population values. On the other hand, if this objects, which will constitute a singleton barycentre set in training, falls in verification it will be assigned to the wrong class because it will find no opportune barycentre in the training set, since there it figured as a singleton and will be nearer to a barycentre of another class, as otherwise in training it would not have formed a new barycentre. Thus because of reduced size of the sample classification errors will be made, *which would not occur if the training set was much larger or more representative.* If the training sample is not collectively exhaustive this will give rise to error. Thus the training set must be *consistent* in the sense of collective exhaustiveness of the patterns in the population. Hence, to help improve the representativeness of the training set, it is often worthwhile to sample the training set with a *stratified* sample instead of a simple random sample. This procedure will be called *labeling.* To do this it is advantageous to sample less than proportionally those objects that appear in very small subgroups. Thus a very small probability of being chosen in verification is given to those objects that in classifying the complete data set turn out to form a singleton barycentre and are the only member of that subgroup. Such a probability should be so small, to all intents and purposes, as to exclude that pattern from being chosen in verification and a slightly larger probability is chosen if a doubleton barycentre set is formed, so that on average one object of the couple may be chosen but not both. In the same way, structured sampling probabilities should be chosen for 3, 4 and 5 element barycentre sets, so that on average not more than half of the constituent elements are likely to be chosen. This will be implemented in practice by excluding from choice objects which are assigned to singleton barycentres and restricting to not more than half those objects which are assigned to these subgroups with low consistency, see [10].

The labeling procedure has been displayed in meta language in Figure 2.

Step1 **Let**
- $\mathbf{x}_j$, $j = 1, \ldots, n$ be the pattern vectors in the data set
- $\mathbf{B}_0$ be the set of k initial barycentres $\mathbf{b}_i$, $i = 1, \ldots, k$

Step2
Compute the distances of each x_j from all the $b_i \in B_t$
Let $\mathcal{M}$ be the set of $\mathbf{x}_w$ that are closer to a barycentre of a class different from their own.
$t \leftarrow 0$

Step3 **while** $M \neq \emptyset$
- **Let** $\mathbf{x}_s$, $s \in \mathcal{M}$ be the vector with the greatest distance from its own barycentre.
- $c \leftarrow \psi(\mathbf{x}_s)$
- **Let** $B_{t+1} \leftarrow B_t \cup \mathbf{x}_s$
- for all the elements of class c perform a k-means routine using as starting points the barycentres of B_{t+1} that belong to class c
- $t \leftarrow t + 1$
- **Compute** the distances of each $\mathbf{x}_j$ from all the $\mathbf{b}_i \in B_t$
- **Let** $\mathcal{M}$ be the set of $\mathbf{x}_w$ that are closer to a barycentre of a class different from their own.

end

Step4 **for** ($\mathbf{b}_i \in B_t$)
case $\#\mathbf{b}_i$
1 : label the element in $\mathbf{x}_j \in \mathbf{b}_i$ for training
$\{2, 4\}$: label $\frac{1}{2}$ of the elements $\in \mathbf{b}_i$ for training
$\{3, 5\}$: label $\frac{2}{3}$ of the elements $\in \mathbf{b}_i$ for training
end
end

Fig. 2: : Labeling algorithm in meta-language

Open Science Index, Economics and Management Engineering Vol:5, No:3, 2011 publications.waset.org/51/pdf

III. Experimental Results

A. Experimental Setup

The proposed algorithm has been compared with three other well established techniques used in credit scoring, namely nearest-neighbour classifier, logistic regression and classification trees.

The dataset used are the two benchmark datasets made public from the University of California Irvine repository of machine learning. The first data set, called Australian credit data by [18] concerns credit card applications. It is made up of 690 observations and 15 variables (class variable included) This dataset is interesting because there is a good mix of variables (continuous, nominal with small numbers of values, and nominal with larger numbers of values).

The second data set, called German credit data, was provided by Prof. Hofmann in Hamburg in the numeric version provided by Strathclyde University. In the latter version it is made up of 1000 observations and 25 variables (class variable included).

To assess the performance of the various classification techniques, first the resubstitution correct classification rates have been computed, training and testing the algorithms on the whole data both for the Australian and for the German dataset. The results are depicted in Table I. No preprocessing of the data was performed whatsoever.

TABLE I: Resubstitution correct classification rate

	TRACE	k-nn	Logistic	CART
AUS	1.0000	1.0000	0.8754	0.8826
GER	1.0000	1.0000	0.7850	0.7870

As expected, T.R.A.C.E. and $k-$nn ($k = 1$) achieve perfect recognition on the training set. Therefore if the training sample can be considered a good representation of the whole population, then the two algorithms can be expected to perform well in classifying new entities. Logistic and classification trees, on the other hand, perform very well, although they do not reach perfect recognition. The performance of both Logistic and classification tree is much better on the Australian dataset, suggesting that the German data could be considered a more difficult problem.

The resubstitution classification rate is know to be a overestimation of the performance of any algorithms. To obtain a more reliable estimate a much better approach consists in cross-validation. In the following we have applied 10%-cross-validation to assess the performance of the various algorithms, i.e. the dataset is randomly split into two parts, 10% of the data are randomly selected for testing (*test set*) and 90% of the data are randomly selected for training (*training set*) the algorithm. The performance of the trained algorithm on the test set should be a good indicator of the performance of the algorithm on new data since the test set is independent of the training set. To obtain a more stable estimate this procedure is repeated 100 times, each time randomly selecting training set and test set.

B. Results

In the following the cross-validation results of the performance of T.R.A.C.E. will be reported and will be compared with $1-$nn, Logistic regression and classification trees.

To fine-tune the proposed algorithm it has been applied on the original data and on data after standardization of each variable, obtained subtracting the mean and dividing by the standard deviation. T.R.A.C.E. has been applied both in the normal mode and in the labeling mode (stratified). In Table II and Table III the results of the performance of T.R.A.C.E. on the Australian and on the German dataset over 100 trials have been depicted.

TABLE II: Cross-validation results for the proposed algorithm - Australian credit dataset

	Mean	Min	1st.Qu	Median	3rd.Qu	Max
Original	0.651	0.513	0.609	0.662	0.691	0.803
Standized	0.774	0.652	0.750	0.773	0.806	0.884
Stratified	0.814	0.667	0.780	0.815	0.844	0.922
Strat. St.	0.873	0.779	0.846	0.868	0.908	0.956

TABLE III: Cross-validation results for the proposed algorithm - German credit dataset

	Mean	Min	1st.Qu	Median	3rd.Qu	Max
Original	0.639	0.546	0.610	0.640	0.670	0.740
Standized	0.661	0.540	0.633	0.659	0.688	0.770
Stratified	0.753	0.635	0.723	0.747	0.781	0.854
Strat. St.	0.792	0.670	0.768	0.798	0.818	0.900

The average performance together with the worst (Min) best (Max) and the $1^{st}, 2^{nd}$ and 3^{rd} quartiles of the distribution of the correct classification rate over the 100 trials have been detailed in both Tables. In both dataset the performance of T.R.A.C.E. is better on standardized data than on original data. The best performances are obtained using labeling (stratified) with standardization, implying that both datasets have variables with quite different ranges and that both dataset show some subclasses that have been undersampled, i.e. there are some homogenous clusters (subclasses) in the datasets that are made up of very few elements, therefore standardization and labeling are necessary on the datasets. Hence, in the following, the comparisons will be made on standardized data with stratification.

In Table IV the average performances over 100 trials of the four techniques on the Australian dataset are displayed, together with the worst (Min) and best (Max) performance and with the $1^{st}, 2^{nd}$ and 3^{rd} quartiles of the distribution of the correct classification rate.

In Figure 3 the boxplots for the distributions of the correct classification rates for the four techniques have been depicted.

Almost all techniques work fairly well on the data, with performances that are comparable and on average around $86 - 87\%$ correct classification. The only technique which does not perform optimally is the 1-nn, with an average

TABLE IV: Cross-validation results comparisons - Australian credit dataset

	Mean	Min	1st.Qu	Median	3rd.Qu	Max
TRACE	0.873	0.779	0.846	0.868	0.908	0.956
k-nn	0.656	0.478	0.620	0.652	0.696	0.812
CART	0.858	0.754	0.841	0.855	0.884	0.971
Logistic	0.866	0.754	0.841	0.870	0.899	0.942

TABLE V: Cross-validation results comparisons - German credit dataset

	Mean	Min	1st.Qu	Median	3rd.Qu	Max
TRACE	0.792	0.670	0.768	0.798	0.818	0.900
k-nn	0.668	0.530	0.637	0.670	0.700	0.790
CART	0.747	0.640	0.720	0.750	0.772	0.860
Logistic	0.770	0.640	0.740	0.770	0.792	0.850

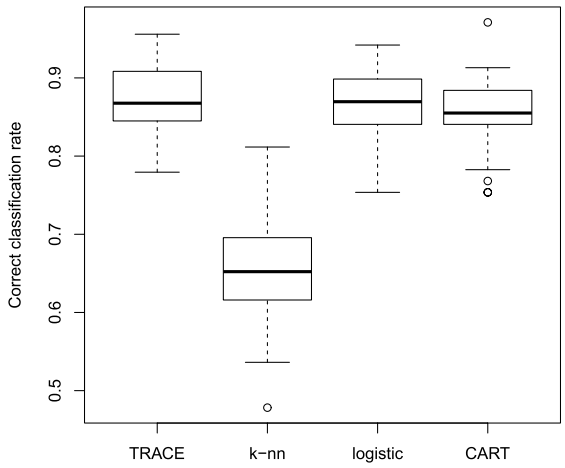


Fig. 3: Australian Data - boxplots for the distributions of the correct classification rates for TRACE, k-nn (k=1), Logistic and Classification Trees (CART)

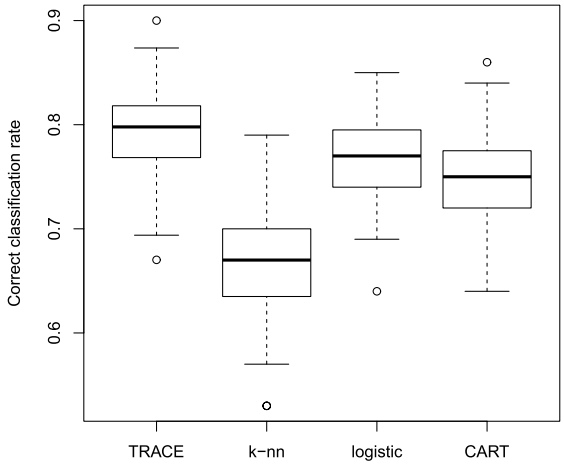


Fig. 4: German Data - boxplots for the distributions of the correct classification rates for TRACE, k-nn (k=1), Logistic and Classification Trees (CART)

correct classification around 66% and, in the worst case, obtaining also a classification rate under the random classification rate which is equal to 50% (see [11]). From the boxplot (Figure 3) we can observe that the performance of T.R.A.C.E. is on average comparable and slightly better than the ones of Logistic and CART, and all perform better than 1−nn. This could be due to the fact that 1−nn is too sensitive to noise and errors in the data, the classification being performed only on the class of the closest element in the training set.

T.R.A.C.E., on the other hand, classifying according to the class of the barycentre closest to the query point, is more capable of handling noise (additive white noise can be averaged-out during the computation of the barycentre).

In Table V analogous results are detailed for the German credit database and in Figure 4 the corresponding boxplots have been displayed. As can be observed, the average performance of the proposed algorithm is better than all the other techniques. Once again the performances of T.R.A.C.E., Logistic and CART are still pretty much comparable, while 1−nn still performs worse than the other techniques, although the difference in performance now is not so pronounced. None of the considered techniques goes below the random correct classification rate.

IV. Conclusions

A statistical pattern recognition algorithm based on the k-means clustering technique has been presented in this paper. The performance of the proposed technique more then holds its own when compared with well established techniques such as k-nn, Logistic regression and Classification Trees and therefore could be used as an automatic support technique in problems of credit scoring. One of the main features of the proposed technique is that it is not black-box-like, i.e. given a credit applicant, not only it can classify him/her as good or bad with a very high precision, but also, being based on the k-means algorithm, it produces a set of subclasses of credit applicants (all homogenous inside the subclass) which can be considered as *prototypes* for the whole class of credit applicants. Therefore not only the algorithm can tell with a high precision whether the applicant is "good" or "bad" according to the class of the barycentre which the applicant is closest to, but also can give a description of the type of applicant via the description of the class of the barycentre which the applicant has been assigned to.

Open Science Index, Economics and Management Engineering Vol:5, No:3, 2011 publications.waset.org/51/pdf

Acknowledgment

The authors would like to thank the Giustino Fortunato University (Italy) for supporting and funding this research.

References

[1] L.J. Mester, What's the point of credit scoring?, *Business Review*, Federal Reserve Bank of Philadelphia, 1997
[2] C. Cifarelli, L. Nieddu, O. Seref, and P. Pardalos. K-t.r.a.c.e.: A kernel k-means procedure for classification. *Computers and Operations Research*, 34(10):3154–3161, 2007.
[3] A. D. Gordon. *Classification*. Chapman & Hall Ltd, London; New York, 1999.
[4] G. Grimaldi, C. Manna, L. Nieddu, G. Patrizi, and P. Simonazzi. A diagnostic decision support system and its applications of the choice of suitable embryos in human assisted reproduction. *Central European journal of Operations Research*, 10(1):29–44, April 2002.
[5] I. T. Jolliffe. *Principal Component Analysis*. Springer Verlag, New York, 2002.
[6] A. Lozano, G. Manfredi, and L. Nieddu. An algorithm for the recognition of levels of congestion in road traffic problems. *Mathematics and Computers in Simulation*, 2007.
[7] L. Nieddu and G. Patrizi. Formal properties of pattern recognition algorithms: A review. *European Journal of Operational Research*, 120:459–495, 2000.
[8] L. Nieddu and G. Patrizi. Optimization and algebraic techniques for image analysis. In M. Lassonde, editor, *Approximation, Optimization and Mathematical Economics*, pages 235 –242. Physica-Verlag, 2001.
[9] S. Watanabe. *Pattern Recognition: Human and Mechanical*. Wiley, New York, 1985.
[10] H. S. Konjin. *Statistical Theory of Sample Design and Analysis*. North Holland, Amsterdam, 1973.
[11] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John wiley & Sons, 1992.
[12] S. Watanabe. Karhunen-Loéve expansion and factor analysis. Theoretical remarks and applications. *Transactions of the fourth Prague conference on Information Theory, Statistical Decision functions, Random Processes*, pp. 635-660, 1965
[13] R. O. Duda and P. E. Hart. *Pattern Classification and scene analysis* Wiley, New York, 1973.
[14] S. Watanabe. *Pattern Recognition: human and mechanical* Wiley, New York, 1985.
[15] P. McCullagh and J.A. Nelder. *Generalized Linear Models, second edition.*, Chapman and Hall, 1989
[16] T. Hastie, R. Tibshirani, J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition.*, Springer, 2009
[17] V. Desai, J. Crook and G. Overstreet. Credit Scoring Models in the Credit Union Environment Using Neural Networks and Genetic Algorithms. *Computer Journal of Mathematics Applied in Business and Industry*, vol. 8 no. 4, pp. 324-346, 1997.
[18] R. Quinlan. Simplifying Decision Trees. *International Journal of Man Machine Studies*, vol. 27 no. 6, pp. 221-234, 1977.
[19] C. Thomas. A Survey of Credit and Behavioral Scoring: Forecasting Financial Risks of Lending to Customers. *International Computer Journal of Forecasting*, vol. 16 no. 2, pp. 149-172, 2000.
[20] A. Vellido, G. Lisboa nd J. Vaughan. Neural Networks in Business: a Survey of Applications. In *Proceedings of Expert Systems with Applications*, Australia, pp. 51-70, 1999.

Luciano Nieddu Luciano Nieddu is an assistant professor of Statistics at the Faculty of Economics (LUSPIO University - Rome/Italy). Has received his Ph.D. in Operation Research from "Sapienza" University of Rome. His research areas include pattern recognition, statistical modeling and computer vision.

Giuseppe Manfredi Giuseppe Manfredi is full professor of Statistics with the Giustino Fortunato University (Italy). His research area includes statistical inference, statistical modeling and discriminant analysis.

Salvatore D'Acunto Salvatore D'Acunto is associate professor of Economics at the Second University of Naples and adjunct professor at the Giustino Fortunato University (Italy). His research area includes two-countries growth models, economic theory of nonprofit organizations and bank behaviour modelling.

Katia La Regina Katia La Regina is an assistant professor with the Giustino Fortunato Telematic University (Benevento-Italy). His research area include statistical inference, statistical modeling and discriminant analysis.

Open Science Index, Economics and Management Engineering Vol:5, No:3, 2011 publications.waset.org/51/pdf