Analysis of Methods for Generating Classification Rules Applicable to Credit Risk

Patricia Jimbo Santana[1], Augusto Villa Monte[2], Enzo Rucci[2,3], Laura Lanzarini[2], and Aurelio Fernández Bariviera[4]

[1] *School of Administration Science, Central University of Ecuador, Quito, Ecuador*
prjimbo@uce.edu.ec
[2] *III LIDI, School of Computer Science, UNLP, La Plata, Bs.As., Argentina*
{avillamonte, erucci, laural}@lidi.info.unlp.edu.ar
[3] *CONICET, La Plata, Bs.As., Argentina*
[4] *Departament of Business, Universitat Rovira i Virgili, Reus, Spain*
aurelio.fernandez@urv.cat

Abstract

Credit risk is defined as the probability of loss due to non-compliance by the borrower with the required payments in relation to any type of debt. When financial institutions select their customers correctly, they can reduce their credit risk. To achieve this, they use various classification methodologies to sort customers based on their risk, analyzing a set of variables such as reputation, leverage, income and so forth. The extensive analysis and processing of these variables is quite time-consuming, partly because the data to be analyzed are not homogeneous. In this paper, we present an alternative method that operates on nominal and numeric attributes, which allows obtaining a predictive model that uses a reduced set of classification rules aimed at reducing credit risk. When the number of rules used decreases, credit analysts need less time to make their decisions, which will also result in better customer service. The methodology proposed here was applied to two databases of the UCI repository and two real databases of Ecuadorian banks that grant various types of credit. The results obtained have been satisfactory. Finally, our conclusions are discussed and future research lines are suggested.

Keywords: Classification rules, Credit scoring, Competitive Neural Networks, Particle Swarm Optimization

Received 13 February 2017 / Accepted 18 April 2017

1 Introduction

Current global economy involves people applying for credit lines for various purposes, such as production, business, consumption, housing, real state, microcredits, and even public investment. As the economy grows, the requirements for granting credit lines have increased, and the different characteristics and behaviors of different customers have to be taken into account. As a result of this, financial institutions have to analyze large numbers of microeconomic variables before granting a line of credit, and then, based on that analysis, either grant the credit that was requested based on their ability to establish a payment plan, or reject the request.

Oftentimes, financial institutions concern about recovering their money, since credit beneficiaries usually behave in a confused and unpredictable manner. This is opposed to the response speed that is required from the process to grant credits, since the goal is to attract as many customers as possible. As a result, financial institutions must improve loan approval accuracy to avoid non-compliance risks.

On the other hand, thanks to the advances made in technology, there are currently countless processes that log their operations automatically, generating large history information repositories. This record includes not only information from different types of observations, but also the results of previous decisions. This drives an interest for learning from past situations, looking to identify the criteria that were used.

Data mining has provided an answer to this problem through different techniques that model available information with no prior hypothesis required, which means that, when analyzing credit variables, financial analysis responses can be obtained.

The objective of this paper is modeling credit risk information using classification rules. The correct identification of the most relevant characteristics will be of great help for financial analysts to make their decisions.

To measure the performance of the method proposed here, different solutions are analyzed considering in particular model simplicity in relation to:

- Number of rules: a reduced set of rules results in a better model.

- Average length of rule antecedent: a lower number of conditions used to form the antecedent of each rule results in a model that is easier to interpret.

An association rule is an expression with the following format

IF condit1 THEN condit2

where both conditions are conjunctions of propositions of the (attribute=value) type and whose only restriction is that the attributes included in the antecedent of the rule must not be part of its consequent [1]. When the set of association rules has the same attribute in the consequent, it is a set of classification rules [2, 3].

In this article, we present a method to obtain classification rules that combines a neural network with an optimization technique. Emphasis is on achieving good coverage with a reduced number of rules.

Section 2 briefly describes some related articles, Section 3 details the method proposed, Section 4 presents the results obtained, and Section 5 presents a summary of the conclusions along with possible future work lines.

2 Related Work

In the 1960s, capital market development in the United States faced the need to start using more scientific models to assess corporate economic strength. The first z score model was developed by Altman and presented in [4]. Twenty years later, by the end of the 90s, the same author published a survey of techniques used in the financial sector [5]. This work does not explicitly describe the application of hazard rate or partial likelihood models, although it does explain the use of statistical techniques probit and logit together with state transition techniques and other techniques known as "actuarial default likelihood derivation" linked to past bond defaults. With the new millennium, specific developments for applying survival analysis to credit risk measurement came to light [6, 7, 8].

In the last few decades, there has been an increase in consumption credit. In our market, savings and credit cooperatives are considered as a growing industry. Not only has the number of credit card holders increased, especially in emerging economies, but there has also been an increase in the number of small consumption credits. For instance, in these economies it is very common for families to buy household appliances using credit card installment plans. In several countries, it is also common that stores that sell household appliances partner up with a financial institution to offer their customers a quick line of credit. The existence of such financial instrument helps increase sales. This partnership generates a conflict of interests. On the one hand, the store that sells household appliances wants to sell their products to all potential customers; therefore, it is in their best interest to promote an attractive credit policy. On the other hand, the financial entity wants to maximize their income from credits, and to that effect, they implement stringent monitoring controls on their losses over granted loans. The goal is the implementation of transparent policies between the stores that offer household appliances and their partner financial institutions. There are also financial institutions that offer consumption credits or microcredits directly to their customers, and it is also in their interest to minimize risk. One possible way for developing such a policy is the objective identification of relevant characteristics in credit beneficiaries to help decide whether to grant the loan or not by building a suitable model.

Regardless of the model to be built, there are two problems that usually affect history information of credits granted: first, is imbalanced class data, and the second problem is the large number of customer attributes that have to be analyzed before granting the credit line [9]. Both issues will negatively affect the performance of any model that is built. In the literature, there are solutions proposed for these issues.

In [10], the authors propose four alternatives to solve class imbalance in a credit risk problem. One of the options is predefining class likelihood distribution and, based on that, selecting the examples to be used. A second option would be using a weight matrix so that not all of them have the same significance. For instance, those in the minority class can increase their value by a certain number of times. A third option is based on using a loss matrix. The final option is using a sub-sampling process, which consists in using a subset of input data to build the model, removing instances (typically from the majority class). In all cases, the performance of the method selected was measured using the data set selected to build a classification tree.

As regards input space reduction, the authors in [11] used association rules to select relevant attributes that were then used to train a decision tree with the C4.5 algorithm [12]. The latter is one of the most commonly used methods, since it can operate with both numeric and nominal attributes, and it supports missing data. It has parameters that control the pruning level of the tree, showing the most important characteristics of the problem at the expense of reduced classification accuracy.

On the other hand, there are two possible ap-

proaches to model credit risk – descriptive and predictive. The former is usually solved by means of grouping techniques, self-organizing maps, or SOMs, being one of the most commonly used ones. These networks were used in [13] to sort customers into several groups, analyzing provider characteristics and the likelihood of non-compliance for each group.
Classification-rule-based models can solve the problem from both points of view, since, even though they are predictive in nature, if the set is simple enough to interpret and analyze, they can also be used to describe the decisions made.
If working with classification rules, the literature includes different tree-based methods for building them, such as C4.5 [12], or trimmed tree-based methods, such as PART [14]. In either case, it is essential that the set of rules obtained covers the examples with a preset error level. Tree-based rule building methods are partitive and based on various attribute metrics in order to assess their coverage ability. Other methods combining models to improve classification accuracy have also been defined. Such is the case of the method proposed in [15], where a fuzzy SVM (support vector machine) that considers the output from several secondary SVMs is used to help a bank establish a more reliable system to assess their customers.

3 Methodology

In this paper, we present a methodology that can be considered hybrid in nature that is based on the combination of particle swarms with competitive neural networks. The latter are used to start the search process at promising positions. Even though there are rule-generation methods that use PSO [16], when operating on nominal attributes the body of available examples should be large enough to cover all search space areas, which is not always feasible. The result is a poor initialization of the population, which in turn causes a premature convergence. To solve this problem, and at the same time reduce rule generation time, the performance of several methods that combine fixed and variable population was compared. PSO starts with two competitive neural networks – LVQ (Learning Vector Quantization) and SOM (Self-Organizing Maps). In the literature, there are methods that use PSO to determine the optimal number of competitive neurons in the network, such as [17]. This is not the case of our proposal, since the optimization technique is used here to identify the most representative characteristics that will be included in rule antecedents.

3.1 Learning Vector Quantization

Learning Vector Quantization (LVQ) is a supervised classification algorithm that is based on centroids or prototypes [18]. It can be interpreted as a competitive neural network formed by three layers. The first layer is just an input layer. The second layer is where competence takes place. The output layer is responsible for the classification process. Each neuron in the competitive layer is associated to a number vector whose dimension is the same as that of the input examples, and a label that indicates the class that it is going to represent. Once the adaptive process finishes, these vectors will contain the information related to the classification centroids or prototypes. There are several versions of the training algorithm. The one used in this article is described below.

When the algorithm is started, the number K of centroids to be used must be indicated. This allows defining the architecture for the network, since the number of input entries and output results are given by the problem.
Centroids are initialized taking K random examples. Examples are then entered one by one, and centroid position is then adapted. To do so, the centroid that is closest to the example being analyzed is selected using a preset distance measurement. Since this is a supervised process, it is possible to determine if the example and the centroid belong or not to the same class. If the centroid and the example do belong to the same class, the centroid is "moved closer" to the example in order to strengthen representation. If, on the contrary, they belong to different classes, the centroid is "moved away". These movements are done by means of a factor or adaptation speed that allows weighing the distance for the move.
This process is repeated until modifications are below a preset threshold, or until the examples are identified with the centroids themselves in two consecutive iterations, whichever happens first.
For the implementation used in this article, the second nearest centroid is also analyzed and, should it belong to a different class than that of the example, and should it be at a distance that is less than 1.2 times the distance to the first centroid, the "moving away" step is applied.

Several variations of LVQ are described in [18].

3.2 Self-Organization Maps (SOM)

The SOM (Self-Organizing Maps) neural network was defined by Kohonen in 1982 [18]. Its main application is grouping all available information, and it is characterized by its ability to preserve input data topology. Just as LVQ, it is a partitive clustering technique, since it associates each example to an average vector or centroid. However, it adds the concept of neighborhood for centroids, allowing that similar groups are closer together within the architecture. There is no such characteristic in LVQ. For this reason, it is commonly used as visualization tool and to reduce the number of dimensions of the input space. It can be represented as a two-layer structure: the input layer, whose function is only to allow the entry of information to the network, and the competitive layer, which is responsible for the grouping task. The neurons that form this second layer are connected and have the ability of identifying the number of “hops” or connections that separate them from each of the other neurons in this level. Each competitive neuron is associated to a weight vector or centroid represented by the values of the arcs that reach this neuron from the input layer. Therefore, the SOM network interacts with two information structures: one in relation to the centroids linked to the competitive neurons, and the other one that is responsible for establishing proximity around neurons. This style, unlike other methods such as the K-means method [19], offers additional information about clusters, since the neurons that are close together within the architecture may represent similar groups in the input data space.

3.3 Generating Classification Rules with PSO

Particle Swarm Optimization is a population-based metaheuristic algorithm proposed by Kennedy and Eberhart [20] where each individual in the population, called particle, represents a possible solution to the problem and changes following three factors: its knowledge of the environment (its fitness value), its historical knowledge or previous experiences (its memory), and the historical knowledge or previous experiences of neighboring individuals (its social knowledge).

Using PSO to generate classification rules that can operate on nominal and numerical attributes requires a combination of the methods mentioned above, since the attributes that will be part of the antecedent have to be selected and the value or range of values they can take has to be determined (discrete-continuous combination).

Since this is a populational technique, the required information has to be analyzed for each individual in the population. A decision has to be made between representing a single rule or the entire set for each individual, and the representation scheme has to be selected for each rule. Given the objectives proposed for this work, the Iterative Rule Learning (IRL) [21] approach was followed, where each individual represents a single rule and the solution to the problem is built from the best individuals obtained after a sequence of runs. Using this approach implies that the populational technique will be applied iteratively until achieving the desired coverage and obtaining a single rule in each iteration: the best individual in the population Additionally, a fixed-length representation was chosen, where only the antecedent of the rule will be coded and, given the approach adopted, an iterative process will be carried out to associate all individuals in the population to a preset class, which does not require consequent codification.

3.4 Method Proposed

Rules are obtained through an iterative process that analyzes non-covered examples in each class, starting with the largest classes. Each time a rule is obtained, the examples that are correctly covered by the rule are removed from the input data set. The process continues until all examples are covered or until the number of non-covered examples in each class is below the established minimum support or until a maximum number of tries has been done to obtain a rule, whichever happens first. It should be noted that, since the examples are removed from the input data set as they are covered by the rules, the rules operate as a classification list. That is, in order to classify a new example, the rules must be applied in the order in which they were obtained, and the example will be classified with the class that corresponds to the consequent of the first rule whose antecedent is verified for the example at hand.

Since neural networks only operate with numerical data, nominal attributes are represented by means of dummy code that uses both binary digits and the different options that may be present in such nominal attribute. Also, before starting the training process, each dimension that corresponds to a numerical attribute is linearly escalated in [0,1]. The similarity measurement used is the Euclidean distance. Once training is finished, each centroid will contain approximately the average of the examples it represents.

To obtain each of the rules, the class to which the consequent belongs is first determined. Seeking high-support rules, the method proposed will start by analyzing those classes with higher numbers of non-covered examples. The minimum support that any given rule has to meet is proportional to the number of non-covered examples in the class upon rule generation. That is, the minimum required support for each class decreases as iterations are run, as the examples in the corresponding class are gradually covered. Thus, it is to be expected that the first rules will have a greater support than the final ones.

In algorithm 1, the pseudo-code of the method proposed is shown. For more details, see [22, 23].

Algorithm 1 Pseudocode of the proposed method

```
Train network using all training examples.
Calculate the minimum support for each class.
while (termination criterion is not reached) do
    Choose the class with the highest number of non-covered examples.
    Build a reduced population of individuals from centroids.
    Evolve the population using variable population PSO.
    Obtain the best rule for the population.
    if (the rule meets support and confidence requirements) then
        Add the rule to the set of rules.
        Consider the examples classified by this rule as correctly covered.
        Recalculate the minimum support for this class.
    end if
end while
```

4 Data and Results

To measure the performance of the method proposed here, two databases from the UCI repository and two real databases from Ecuadorian institutions were used. For these two, credit applications and awarded credit operations were analyzed using the following attributes: status; date of application; credit target; province; amount requested; amount authorized; purpose of the credit line; how much cash the customer has, bank accounts, investments, other assets, liabilities and salary of the applicant; information verification date; authorization date; approval date / rejection date; bank accounts, investments, other assets, liabilities and salary of applicant spouse. If the applicant is a small business, the information requested includes business income and expenses. Applications can be rejected or accepted. If it is accepted, the status is sorted with other credits that were paid out with no incidents and those that have some delay in investment recovery. Similarly, overdue credits are sorted, depending on credit procedures, into those that are less than 90 days overdue and those that are more than 90 days overdue (start of legal actions), which can be considered to be matured.

Four variations of the method proposed were measured that combine two types of PSO, one with fixed population and one with variable population, initialized with two different competitive neural networks: LVQ and SOM. The solutions obtained were compared with the C4.5 and PART methods. The procedures for finding classification rules in the methods proposed and the control methods are different. C4.5 is a pruned tree whose branches are exclusionary and allow classifying the examples. PART returns a list of rules that are equivalent to those generated by the classification method proposed, but in a deterministic manner. PART works by building partial trees. Each tree is created in a way similar to that proposed for C4.5, but during the process, generation errors are calculated for each branch. These errors determine when tree generation must end.

A total of 30 separate runs were carried out for each method. For fixed population PSO, a competitive network of 30 neurons was used, while in the case of variable population PSO, starting size was 20 neurons. PART was run with a confidence factor of 0.3 for the pruned tree. For the remaining parameters, the default values were used.

Tables 1, 2, 3 and 4 summarize the results obtained when each method was applied to each database, considering not only rule set coverage accuracy, but also the simplicity of the model obtained. This simplicity is reflected on the average number of rules obtained and the average number of terms used in the antecedent.

As regards accuracy, Figures 1, 3, 5 and 7 show the confidence intervals for the average accuracy obtained with each method for each database using a confidence level of 0.05. Figures 1 and 3 show that the variations of the method proposed were able to successfully solve both repository cases, while Figures 5 and 7 indicate that the two real cases were more accurately classified by the algorithms based on C4.5 partition and PART.

However, in all cases, the cardinality of the model offered by the four variations of the method proposed is markedly lower than that of the two control methods. Even though the difference in accuracy between both types of method is within a range of 1-3 percent points, it should be noted that the accuracy of the PSO-based classification is very good and comparable to that obtained with the other methods. As regards the number of rules, this value is 10-20 times larger in partition methods.

Figures 2, 4, 6 and 8 show the simplicity of each of the models obtained for each database. Simplicity was measured by the total number of conditions that are included in the entire model for each case; i.e., the number of rules in the model multiplied by the average length of each rule (antecedent conditions).

It should be noted that the information from both real cases corresponds to consumption loans. These operations handle amounts that are much lower than mortgage loans, and quick decisions

Table 1: Results obtained with Australian database, UCI repository

Method	Precision	# Rules	Length Anteced.
SOM+PSO	**0.8584** ±0.0140	3.0100 ±0.0316	1.3525 ±0.0650
SOM+varPSO	**0.8536** ±0.0126	3.0600 ±0.0699	1.7258 ±0.1084
LVQ+PSO	**0.8641** ±0.0130	3.0200 ±0.0421	1.3925 ±0.0569
LVQ+varPSO	**0.8543** ±0.0106	3.1100 ±0.1286	1.7258 ±0.1038
C4.5	**0.8540** ±0.0061	18.6066 ±2.1500	4.8638 ±0.2598
PART	0.7358 ±0.0340	33.4800 ±1.9028	2.4820 ±0.0829

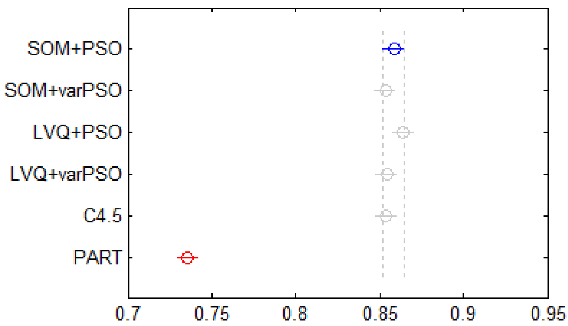

Figure 1: Confidence intervals for the average accuracy obtained with each method for the Australian database.

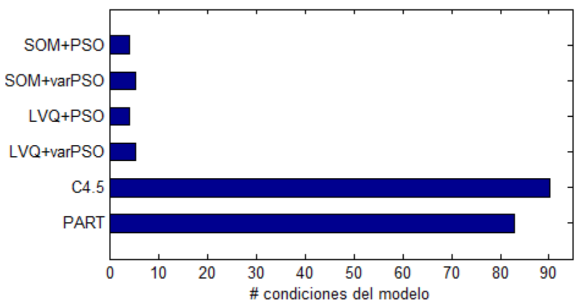

Figure 2: Model simplicity obtained with each method for the Australian database.

Table 2: Results obtained with German database, UCI repository

Method	Precision	# Rules	Length Anteced.
SOM+PSO	**0.6984** ±0.0220	6.3444 ±1.6094	2.5248 ±0.3045
SOM+varPSO	**0.7020** ±0.0139	6.3909 ±1.3888	2.4756 ±0.2248
LVQ+PSO	0.6823 ±0.0298	6.3555 ±1.6194	2.4468 ±0.3458
LVQ+varPSO	**0.6981** ±0.0231	6.5700 ±0.9129	2.5548 ±0.2268
C4.5	**0.7105** ±0.0072	85.0266 ±4.4466	5.6155 ±0.1678
PART	0.6940 ±0.0130	71.0600 ±1.8257	2.9978 ±0.0774

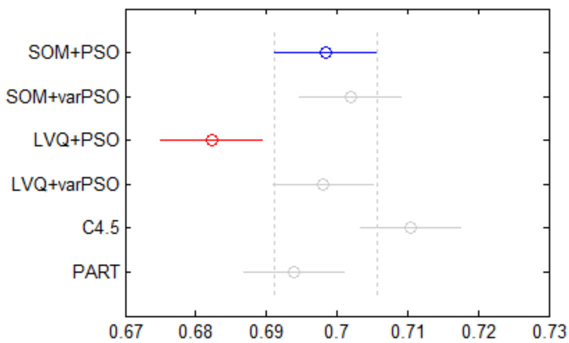

Figure 3: Confidence intervals for the average accuracy obtained with each method for the German database.

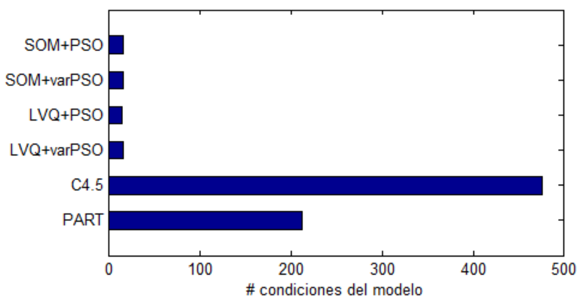

Figure 4: Model simplicity obtained with each method for the German database.

Table 3: Results obtained with database from Cooperative

Method	Precision	# Rules	Length Anteced.
SOM+PSO	0.7913 ±0.0027	3.8250 ±0.3862	1.7102 ±0.0749
SOM+varPSO	0.7912 ±0.0021	4.7000 ±0.8445	1.8697 ±0.2261
LVQ+PSO	0.7923 ±0.0038	4.0749 ±0.4787	1.6464 ±0.0845
LVQ+varPSO	0.7965 ±0.0040	4.7750 ±0.9394	1.7308 ±0.0840
C4.5	**0.8105** ±0.0011	114.2600 ±6.0543	9.6762 ±0.1143
PART	0.8054 ±0.0023	42.3566 ±2.1661	4.6956 ±0.0880

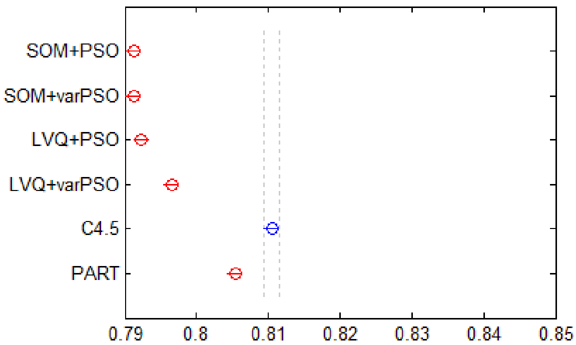

Figure 5: Confidence intervals for the average accuracy obtained with each method for the database from the Cooperative.

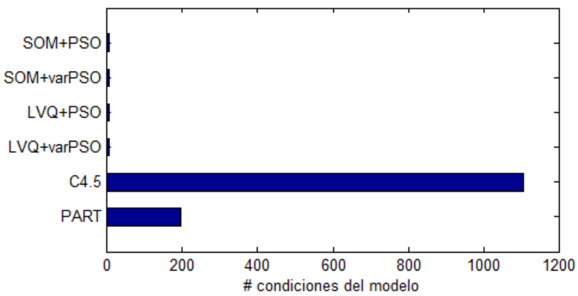

Figure 6: Model simplicity obtained with each method for the database from the Cooperative.

Table 4: Results obtained with Ecuadorian bank database

Method	Precision	# Rules	Length Anteced.
SOM+PSO	0.9254 ±0.0063	3.5666 ±0.2081	3.1905 ±0.4328
SOM+varPSO	0.9529 ±0.0028	4.0142 ±0.3184	2.3164 ±0.2695
LVQ+PSO	0.9336 ±0.0139	3.6833 ±0.1471	2.6933 ±0.2149
LVQ+varPSO	0.9470 ±0.0069	3.9333 ±0.2658	2.3983 ±0.2092
C4.5	**0.9778** ±0.0003	153.5733 ±5.1686	11.2348 ±0.1564
PART	**0.9761** ±0.0007	80.9400 ±2.2033	4.7650 ±0.0687

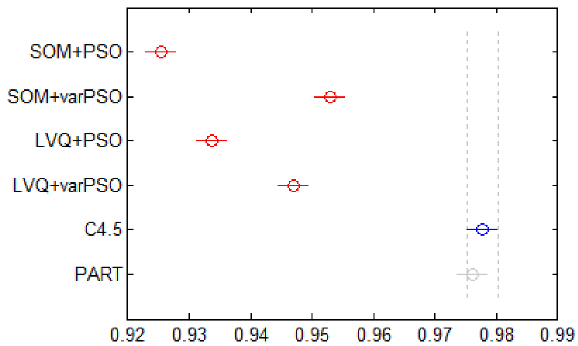

Figure 7: Confidence intervals for the average accuracy obtained with each method for the database from the Ecuadorian bank.

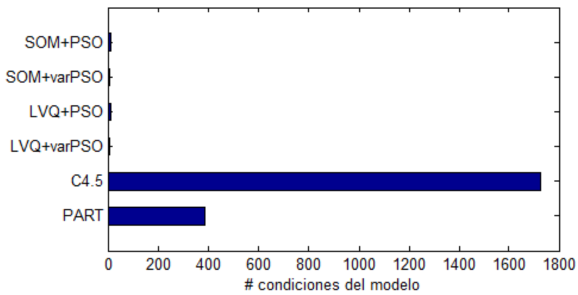

Figure 8: Model simplicity obtained with each method for the database from the Ecuadorian bank.

are required because they are usually agreed with the customer through an on-line service.
A loss of accuracy is acceptable in this type of situations, since the volume of operations will make up for any incorrect decisions. However, being able to make a decision based on a short questionnaire with no more than 10 questions highly increases the chances of a successful operation. No customer will go through an on-line questionnaire as long as those proposed by the other methods. Consequently, there is a sort of balance between simplicity and accuracy. Since credit rules must be simple to be able to give a quick answer to customers, the method proposed here is considered to be a suitable solution for this problem.

5 Conclusions

A new method for generating classification rules based on the combination of PSO and competitive neural networks has been presented; a preliminary version of this work can be found in [24]. The method has been tested with two real credit databases from a credit and savings cooperative and an Ecuadorian bank, as well as two public databases from the UCI repository (UC Irvine Machine Learning Repository). Results have been satisfactory. The measurements obtained allow stating that the method proposed significantly reduces the number of rules required while maintaining an acceptable level of accuracy.

It should be noted that the objective of this research work is finding an intuitive model for credit scoring that offers an accuracy level comparable to that achieved by popular reference models. The results obtained suggest that the simplification of the rules used to make the decision generates transparency in the credit scoring process, which could result in an improved reputation for financial institutions.

Future research lines should consider adding the analysis of a set of micro- and macro- economy variables to obtain a simpler model while keeping an adequate level of accuracy.

References

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, (San Francisco, CA, USA), pp. 487–499, Morgan Kaufmann Publishers Inc., 1994.

[2] C. Aggarwal, *Data Mining: The Textbook*. Springer International Publishing, 2015.

[3] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems, Elsevier Science, 2011.

[4] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The Journal of Finance*, vol. 23, no. 4, pp. 589–609, 1968.

[5] E. I. Altman and A. Saunders, "Credit risk measurement: Developments over the last 20 years," *Journal of Banking and Finance*, vol. 21, no. 11-12, pp. 1721–1742, 1997.

[6] D. Duffie and K. Singleton, *Credit Risk: Pricing, Measurement, and Management*. Princeton Series in Finance, Princeton University Press, 2012.

[7] K. Roszbach, "Bank lending policy, credit scoring, and the survival of loans," *The Review of Economics and Statistics*, vol. 86, no. 4, pp. 946–958, 2004.

[8] A. Saunders and L. Allen, *Credit Risk Management In and Out of the Financial Crisis: New Approaches to Value at Risk and Other Paradigms*. Wiley Finance, Wiley, 2010.

[9] K. Andric and D. Kalpic, "The effect of class distribution on classification algorithms in credit risk assessment," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1241–1247, May 2016.

[10] S. Birla, K. Kohli, and A. Dutta, "Machine learning on imbalanced data in credit risk," in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1–6, Oct 2016.

[11] X. Mei and Y. Jiang, "Association rule-based feature selection for credit risk assessment," in *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, pp. 301–305, May 2016.

[12] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[13] Q. l. Chen and J. b. Lin, "Integrating of business intelligence and crm in banks: An empirical study of som applied in personal customer loans in taiwan," in *2015 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, pp. 68–73, Nov 2015.

[14] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, (San Francisco, CA, USA), pp. 144–151, Morgan Kaufmann Publishers Inc., 1998.

[15] Z. X. Li, "A new method of credit risk assessment of commercial banks," in *2016 International Conference on Robots Intelligent System (ICRIS)*, pp. 34–37, Aug 2016.

[16] Z. Wang, X. Sun, and D. Zhang, *A PSO-Based Classification Rule Mining Algorithm*, pp. 377–384. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[17] C. Hung and L. Huang, "Extracting rules from optimal clusters of self-organizing maps," in *2010 Second International Conference on Computer Modeling and Simulation*, vol. 1, pp. 382–386, Jan 2010.

[18] T. Kohonen, *Self-Organizing Maps*. Springer Series in Information Sciences, Springer Berlin Heidelberg, 2012.

[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, Calif.), pp. 281–297, University of California Press, 1967.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

[21] G. Venturini, "Sia: A supervised inductive algorithm with genetic search for learning attributes based concepts," in *Proceedings of the European Conference on Machine Learning*, ECML '93, (London, UK, UK), pp. 280–296, Springer-Verlag, 1993.

[22] L. Lanzarini, A. Villa Monte, G. Aquino, and A. De Giusti, *Obtaining Classification Rules Using lvqPSO*, pp. 183–193. Cham: Springer International Publishing, 2015.

[23] L. Lanzarini, A. Villa Monte, and F. Ronchetti, "Som+pso: A novel method to obtain classification rules," *Journal of Computer Science and Technology*, vol. 15, pp. 15–22, 4 2015.

[24] P. Jimbo Santana, A. Villa Monte, E. Rucci, L. C. Lanzarini, and A. Fernández Bariviera, "An exploratory analysis of methods for extracting credit risk rules," in *XIII Workshop Bases de datos y Minería de Datos (WBDMD). XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*, pp. 834–841, Oct 2016.