IJICT International Journal of Information & Communication Technology

ITRC

Volume 3- Number 2- March 2011 (57-65)

**Research Note**

# Credit Scoring Using Colonial Competitive Rule-based Classifier

Javad Basiri
Dept. of Electrical & Computer Engineering
University of Tehran
Tehran, Iran
basiri@ut.ac.ir

Fattaneh Taghiyareh
Dept. of Electrical & Computer Engineering
University of Tehran
Tehran, Iran
ftaghiyar@ut.ac.ir

Mohammad Siami
Dept. of Industrial Engineering
Iran University of Science & Technology
Tehran, Iran
m_siami@ind.iust.ac.ir

Mohammad Reza Gholamian
Dept. of Industrial Engineering
Iran University of Science & Technology
Tehran, Iran
gholamian@iust.ac.ir

Received: October 29, 2010 - Accepted: January 19, 2010

*Abstract*— **Credit scoring is becoming one of the main topics in the banking field. Lending decisions are usually represented as a set of classification tasks in consumer credit markets. In this paper, we have applied a recently introduced rule generator classifier called CORER[1] (Colonial cOmpetitive Rule-based classifiER) to improve the accuracy of credit scoring classification task. The proposed classifier works based on Colonial Competitive Algorithm (CCA). In order to approve the CORER capability in the field of credit scoring, Australian credit real dataset from UCI machine learning repository has been used. To evaluate our classifier, we compared our results with other related well-known classification methods, namely C4.5, Artificial Neural Network, SVM, Linear Regression and Naïve Bayes. Our findings indicate superiority of CORER due to better performance in the credit scoring field. The results also lead us to believe that CORER may have accurate outcome in other applications of banking.**

***Keywords-credit scoring; CORER; colonial competitive algorithm; rule-based classifier; classification; finance and banking;***

## I. INTRODUCTION

Economical crisis in recent decade lead the banks and credit institutes to have more attention to credit risk. Credit scoring, one of the first credit risk evaluator and important way for customer account evaluation, improves cash flow, reduces credit risk and helps for managerial decision. Customer credit scoring aims at dividing loan applicants into two main categories; loan applicants with good credit and the ones with bad credit. Applicants with good credits are more likely to repay their loans and vice versa, the applicants with bad credits are less probable to repay[1]. The importance of customer credit scoring has been highlighted in many recent studies[2, 3] making it a critical subsystem in customer relationship management [4] such that even one percent improvement in the accuracy of model can strongly affects its performance and responsiveness [1, 5].

Generally speaking, credit scoring is a binary classification problem which distinguishes good and

[1] Tool used to remove the core of a fruit.

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]

bad loan applicants [6]. Many algorithms such as decision trees [2, 7], neural networks [3, 8], genetic programming [9] and logistic regression analysis [1, 7, 10] have been conducted to predict worthy customer but nevertheless this problem remains a hot topic in financial research.

"For good classifiers, superior accuracy may be one of the most important performance measures" [2]. In recent years, many methods [2, 7, 8, 11] have been proposed for credit scoring. All of these researches have tried to increase the model accuracy.

In this paper we introduce the CORER (Colonial cOmpetitive Rule-based classifiER) algorithm to the credit scoring literature to increase the accuracy of credit scoring task. This classifier, like other rule-based classifiers, extracts one rule from the training set in each time [12]. The CORER algorithm works based on CCA (Colonial Competitive Algorithm), a recently-developed evolutionary optimization algorithm [13]. To the best of our knowledge, credit scoring literature does not contain any reference (yet) to the CORER algorithm.

Some algorithms such as decision tree induction methods or rule-based classifiers such as CN2 [14] and PART [15] examine only one feature at a time. This characteristic may introduce some constraints to achieve a high accuracy in data classification task. The CORER classifier considers all the features of the objects from the first iteration to the final iteration. It may omit a specific feature from a specific rule in the current iteration and add that feature to that rule, in the next iteration and vice versa. In each time, it tries to improve the performance of current rules.

The remaining of the paper is arranged as follows. In section 2, we review the credit scoring and related works, while section 3 describes the CORER classifier in details. In section 4, we explain the experimental results on a benchmark dataset. Finally, section 5 concludes the paper.

## II. CUSTOMER CREDIT SCORING

Credit scoring is one of the most important financial risk forecasting to customer lending [10] methods and there are many definitions of credit scoring.

Most of definitions have emphasized that credit scoring is a classification method which classifies customers into two main categories: customer with good and bad credit.

Using a credit scoring method with a high accuracy has some benefits for banks such as:[16]

- Risk evaluation for customer
- Reduce cost for credit evaluation
- Faster and easier decision making for loan requests

Credit scoring is obtained from the customer experiences. Credit analyst's decisions tended to be according to the view that what mattered was the 5Cs: the Character of the consumer, the Capital, the Collateral, the Capacity and the economic Conditions (Fig. 1) [10].

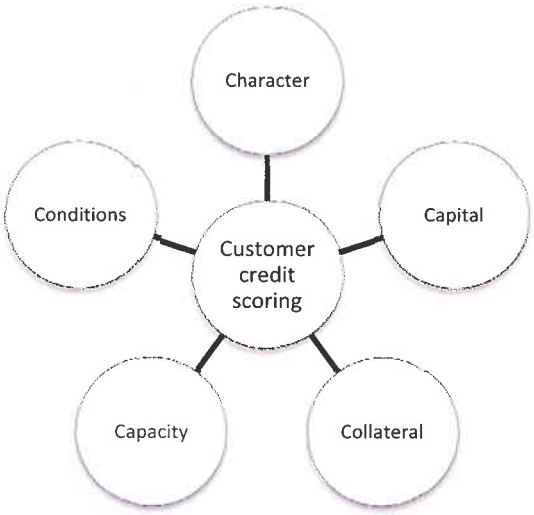



Figure 1.Customer credit scoring 5Cs framework [10]

Based to the mentioned customer credit evaluation or credit scoring modeling with the extremely increase of applicants, it is impossible to conduct the work manually. Thus, many researches and many algorithms are presented in this field for predict customer credit easily and with a high accuracy [1].

Many approaches have been conducted to predict the credit scoring. These approaches have been classified into two categories: Statistical and Artificial Intelligence models [1, 16]. Linear Discriminant Analysis (LDA) and Logistic Regression Analysis (LRA) are first statistical methods that have been proposed in credit scoring literature [10]. Low accuracy of these models leads the researchers to use artificial intelligence and data mining methods [1] such as decision tree (DT) [2, 7], artificial neural networks (ANNs) [3, 8], support vector machine (SVM) [1, 17, 18], Genetic Programming (GP) [9] for credit scoring to reach superior accuracy in their models. Most of the AI techniques are trying to improve the accuracy of model and increasing the number of instances that have been diagnosed correctly good or bad [2].

There are some research gaps in the credit scoring literature including:

- Low performance of neural network in small datasets with irrelevant attributes [9].
- Applying statistical techniques to credit scoring is somehow problematic. Some theoretical assumptions may be not suitable in practice for credit scoring. For instance, consider the multivariate normality assumptions given for independent variables; these are frequently violated in the practice of credit scoring. Therefore, as mentioned above, some of these techniques are theoretically invalid for finite samples [1].
- About decision tree algorithms, as some nodes may have similar probabilities and the approach is vulnerable to noise, they may have a less optimal solution [18].

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]

• Support vector Machine (SVM) methods have problem in choosing best input features and kernel parameters [17].

This paper tries to improve the above problems by using an efficient classifier method that is introduced in the next section.

## III. CORER CLASSIFIER

CORER classifier works based on CCA algorithm a recently-developed evolutionary optimization algorithm. So, in this section, first we review the CCA algorithm. Thereafter, we describe the CORER classifier in details.

### A. CCA Algorithm

CORER classifier extracts rules based on CCA (Colonial Competitive Algorithm). CCA algorithm is a new optimization algorithm which introduced by Atashpaz et al. [13]. It starts with an initial population that is called country. Based on cost function, the countries are divided into two categories, colonies and imperialists. An imperialist with its colonies form an empire. There is a competition between the empires and weak empires fall down and powerful ones expand, during the competition. The CCA reaches to a state in which only one empire remains. The procedure of this algorithm is as follows:

1. Select some random points on the function and initialize the empires.
2. Move the colonies toward their imperialist (Assimilating).
3. Change the position of some colonies randomly (Revolution).
4. If there is a colony in an empire with lower cost than the relevant imperialist, exchange their positions.
5. Compute the total cost of empires.
6. Unite similar empires.
7. Pick the weakest colony from the weakest empire and give it to the empire with the most likelihood to possess it (Imperialistic competition).
8. Eliminate the powerless empires.
9. If stop conditions satisfied, stop, if not go to 2 [12, 13, 20].

### B. CORER

The CORER works like other rule-based classification algorithms and extracts one rule at a time. We can see the flowchart of CORER in Fig. 2.

#### 1) Preprocessing Phase

When we use the CORER classifier, the domains of the applied features should be in the form of categorical type. To convert numerical feature A to a categorical type, first, the values of A are sorted in ascending order. The midpoint between each pair of adjacent values is a possible split point. The midpoint $MidPT_i$ between two values $a_i$ , $a_{i+1}$ of A is computed as follows:

$$MidPT_i = \frac{a_i + a_{i+1}}{2} , i = 1, 2, \ldots\ldots, N$$

Where, N is the number of existing tuples in dataset D.

For each $MidPT_i$, the following formula is computed:

$$Info_{Ai}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Where, v = 2, $D_1$ is the set of tuples in the dataset D which satisfying A <= $MidPT_i$, $D_2$ is the remaining tuples and $Info(D_j)$ is defined by the following expression:

$$Info(D) = -\sum_{i=1}^{m} p_i \times log_2(p_i)$$

Where, pi is the probability that a tuple belongs to class Ci and is defined by $|C_{i,D}|/ |D|$ and m is the number of classes in the dataset D. We split A at $MidPT_k$ where it has the minimum $Info_A(D)$ between all midpoints. (Line 1.1 in fig. 4 indicates this step).

#### 2) Creating Initial Rules

According to [12], the left hand side of each rule contains some bits for each feature. The right hand side of each rule, indicates the class label. This label is assigned to the instances that are covered with the left hand side. CORER considers one bit for each category of each feature. For example, considering the features $F_1$, $F_2$, $F_3$ and the class label $F_c$ as follows

$F_1 \in \{large, medium, small\}$
$F_2 \in \{yes, no\}$
$F_3 \in \{high, medium, low\}$
$F_c \in \{Yes, No\}$

Rule length in this case is 10 bits, 8 bits for three features and 2 bits for the class label. Consider the following rule:

| $F_1$ | $F_2$ | $F_3$ | | $F_c$ |
|---|---|---|---|---|
| 111 | 10 | 101 | → | 01 |

It means If ($F_1$ = large or medium or small) and ($F_2$ = yes) and ($F_3$ = high or low) then the class label is No.

Notice that a feature involving all 1's, matches all value of a feature and is equivalent to ignore that conjunctive term [21]. So, in this example, the value of $F_1$ is irrelevant, and the rule is interpreted as:

If ($F_2$ = yes) and ($F_3$ = high or low) then the class label is No.

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]

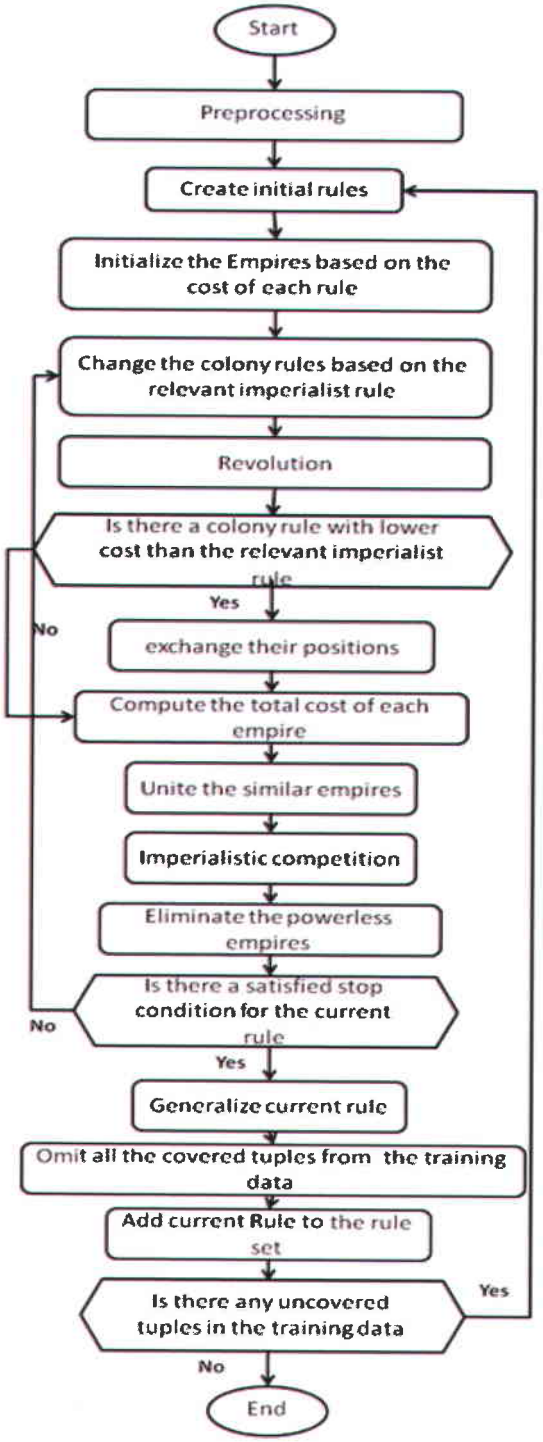



Figure 2. Flowchart of CORER classifier [12]

Considering this strategy, some initial countries are created randomly. (Line 3.1 in fig. 4 indicates this step).

*3) Initialize the Empires*

Coverage and accuracy are two important factors [22] for a rule and are defined as follows:

$$accuracy(R_i) = \frac{n_{correct}}{n_{covers}} \qquad (1)$$

$$coverage(R_i) = \frac{n_{covers}}{|D_{tr}|}$$

Where, $R_i$ is a rule (a member of current population), $n_{correct}$ is number of training records correctly classified by $R_i$, $n_{covers}$ indicates number of training records covered by $R_i$ and $D_{tr}$ indicates the training set. Considering both accuracy and coverage, the cost function $F_{cost}$ has been simply defined as follows [12]:

$$F_{cost}(R_i) = 1 - (\alpha * accuracy(R_i) + \beta * coverage(R_i))$$
$$\text{S.t: } \alpha + \beta = 1, \quad \alpha, \beta > 0 \qquad (2)$$

After generating initial rules (countries), $F_{cost}$ is computed for each rule. Initial imperialists are a specific number of rules with the lowest $F_{cost}$. Each remaining rules will be assigned to a particular imperialist as the colonies of that imperialist [12]. Clearly, in each empire, the best rule is the imperialist of that empire. (Lines 3.2 to 3.4 in fig. 4 indicate this step).

*4) Changing the Rules*

This process is like the movement stage in CCA algorithm. CORER considers the distance between each colony rule and its imperialist rule as the number of different corresponding bits for each feature F. Then, number $\omega$ between 0 and d is selected randomly. Thereafter, $\omega$ bits of the current feature of the current rule (they are different from corresponding bits of the imperialist rule) are selected randomly to convert to their Supplement (0 to 1 and 1 to 0). After this process, the $\omega$ selected bits of the current rule are similar to the corresponding bits of imperialist rule. (Line 3.5 in fig. 4 shows this step).

*5) Revolution*

According to [12] in this phase, CORER chooses some rules of each empire and recreates them randomly. This process is like the creating initial rules. (Line 3.6 in fig. 4 shows this step).

*6) Exchanging the Position of a Colony and its Relevant Imperialist*

After changing the colony rules, if there is a colony rule with a lower cost than the related imperialist, CORER exchanges their positions. (Lines 3.7 to 3.8 in fig. 4 show this step).

*7) Computing the Total Cost*

CORER calculates the total cost of each empire based on [13] using the following formula:

$$T.C._n = Cost(imperialist_n) + \xi mean\{Cost(colonies\ of\ empire_n)\}$$

Where, $T.C._n$ represents the total cost of n-th empire and $\xi$ is a positive number lower than 1. (Lines 3.9 to 3.10 in fig. 4 show this step).

*8) Uniting the Similar Empires*

Based on the CCA algorithm, if bits variation between two imperialist rules is lower than a specific threshold, two related empires are merged. (Line 3.11 in fig. 4 shows this step).

*9) Imperialistic Competition*

There is a competition for one (or more) colony of weakest empire between all the other empires to own this (these colonies). Fig. 3 indicates this process as a big picture. (Line 3.12 in fig. 4 shows this step).

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]

*10) Eliminating the Powerless Empires*

CORER eliminates the empire which loses all of its colonies. (Line 3.13 in fig. 4 shows this step).

*11) Checking the Stop Condition for the Current Rule*

Reaching two conditions, CORER finishes the learning of current rule:

1. Only one empire remains.
2. Number of iteration reaches to a particular threshold.

When one of the above conditions satisfied, CORER has a learned classification rule. This rule is the imperialist position of the only remaining empire or the best imperialist position among all the imperialist rules of all the empires. (Lines 3.14 to 3.15 in fig. 4 show this step).

*12) Rule Generalization*

The learned rule R is generalized by Foil-Prune criteria [22] using the following formula:

$$FOIL_Prune(R) = \frac{pos - neg}{pos + neg}$$

Where pos is the number of tuples which correctly classified by R and neg is the number of tuples which classified by R incorrectly. CORER omits a particular feature if the Foil-Prune for the pruned rule (a rule without that feature) is higher than the non-pruned rule. (Line 3.15.1 in fig. 4 shows this step).

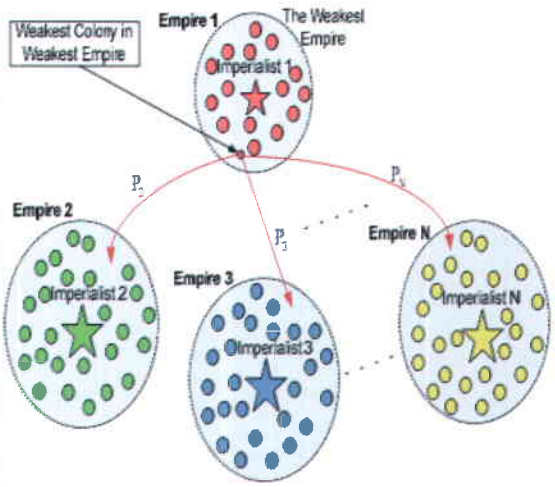



Figure 3. The imperialistic competition [13]

**Classifier** CORER
**Input** : Training data set $D_{tr}$
**Output** : The set of decision rules $R_{set}$
**Procedure** CORER ($D_{tr}$)

1. **For** every numerical feature $F_i$
   1.1. Preprocessing ($F_i$ )
2. **End For**
3. **While** ∃ tuple $T_i \in D_{tr}$
   3.1. Create initial rules R.
   3.2. **For** every rule $R_i$
      3.2.1. Compute the cost function $F_{cost}(R_i)$
   **3.3. End For**
   3.4. Initialize the Empires E based on $F_{cost}(R_i)$.
   3.5. Change the colony rules based on the relevant imperialist rule.
   3.6. Revolution
   3.7. **For** every empire $E_i \in E$
      3.7.1. **If** ∃ colony rule $R_i \in E_i$ with relevant imperialist rule $R_j$ and $F_{cost}(R_i) < F_{cost}(R_j)$
         3.7.1.1. Exchange the position of $R_i$ and $R_j$.
      **3.7.2. End If**
   **3.8. End For**
   **3.9. For** every empire $E_i$
      3.9.1. Compute total cost $T.C(E_i)$
   **3.10. End For**
   3.11. Unite the similar empires.
   3.12. Pick the weakest colony (colonies) from the weakest empire and give it (them) to the empire that has the most likelihood to possess it (Imperialistic competition).
   3.13. Eliminate the powerless empires.
   3.14. **If** none of the stop conditions for the current rule satisfied
      3.14.1. go to 3.5.
   **3.15. Else**
      3.15.1. Generalize current rule $R_i$ by Foil-Prune criteria.
      3.15.2. Omit all the covered tuples of $D_{tr}$.
      3.15.3. Add current Rule R to $R_{set}$.
   3.16. **End If**
4. **End While.**

Figure 4. The procedure of CORER classifier [12]

*13) Omitting the Covered Tuples*

After the generalization process, all the tuples in the training set $D_{tr}$ which covered by the current rule are omitted. Using the refined $D_{tr}$ , a new classification rule will be extracted in the next iteration. (Line 3.15.2 in fig. 4 shows this step).

After eliminating the covered tuples, CORER checks the number of remaining tuples in $D_{tr}$. If there is not any uncovered tuples in $D_{tr}$, the rule extraction is finished.

Fig. 4 depicts the procedure of CORER classifier. The procedure from 3.1 to 3.16 is called stage and the procedure from 3.5 to 3.14 is called iteration.

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]

The main characteristic of CORER is its comprehensive view on all the existing features which many other classification algorithm such as Decision Trees for example ID3 [23], or rule-based classifiers for example RIPPER [24] and CN2 [14] do not have this characteristic [12].

## IV. EXPERIMENTAL RESULTS AND CAMPARISON

In this paper for evaluating our credit scoring model we have used 5 steps procedure that have be shown in fig 5.

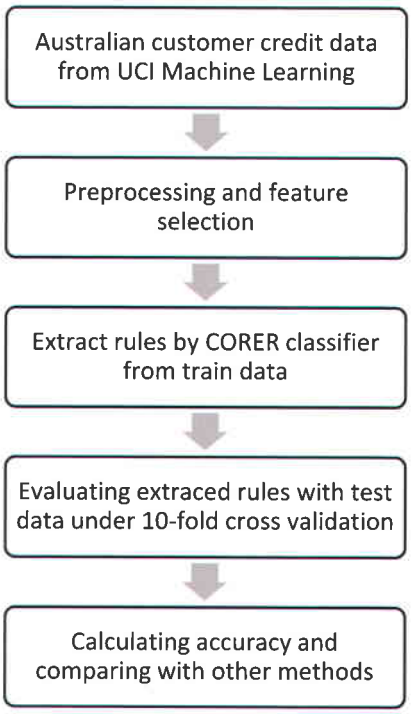



Figure 5. Credit scoring prediction procedure

### A. Dataset

Australian credit real dataset from UCI Repository of Machine Learning Databases (http://www.niaad.liacc. up.pt/statlog/datasets.html) has been used to evaluate the CORER classifier on the customer credit data. This dataset consists of 690 samples, with 307 good applicants and 383 bad ones. Each sample contains 15 features, including 6 nominal and 8 numeric features. Also, 15th feature is the class label which says that the customer have a good or bad credit. Full information about Australian dataset has been illustrated in table 1.

Table 1. Real world dataset from UCI repository

| Name | Number of Classes | Instances | Nominal features | Numeric features | Total features |
|---|---|---|---|---|---|
| Australian | 2 | 690 | 6 | 8 | 14 |

### B. Preprocessing and feature selection

Feature selection algorithms aim at identification of features which are the best for the prediction [25]. Chi-Square algorithm [26] has been used as the feature selection algorithm in this paper. The top 10 selected features have been used for all applied classifiers.

### C. Train CORER and extract Rules

1. Half of initial rules are created randomly and each other initial rules mentions to a particular record in the training set.

2. In our experiments parameter α has been set to 0.95 and β to 0.05 in the equation (2). Also, these values have been gained based on our results on the training set.

3. Number of iterations of CORER has been set to 1500 for the used data set. Also, number of initial rules has been set to 50, and number of initial imperialist to 8.

### D. Test and validation method

In this paper ten-fold cross validation has been applied to dataset as the testing method. Also, we have evaluated the performance of CORER classifier by comparing it with some well-known classifiers in the credit scoring literature, such as Decision Tree (DT) [1, 7], Support Vector Machine (SVM) [1, 19], Linear Regression Analysis (LRA) [1, 7] and Artificial Neural Network (ANN) [3, 8]. The accuracy (equation (1)) of CORER has been compared with the applied classification methods.

### E. Prediction accuracy and analysis

For implementation of base learners, such as DT (j48 module), LRA (logistic module), ANN (MultiLayerPerceptron module), SVM (SMO module) and BN (naïvebayes module) we have used Weka (release 3.6.1) [27] on the Australian credit data set, over the ten-fold cross validation. Also, Matlab version 7.6 was used for CORER.

First, we consider the results of the mean accuracy of CORER and other five classifiers (DT, LRA, ANN, SVM and BN) which has illustrated in fig. 6. As it can be seen, based on the mean accuracy, CORER has the best result on this dataset, in comparison with all the other applied classifiers.

In fig. 7 minimum accuracy of CORER and five other classifiers on the Australian credit data set over the ten-fold cross validation have been shown. Also, in this case CORER has outperformed all the other prediction algorithms.

Fig. 8 depicts maximum gained accuracy of different applied classifiers over 10 folds of ten-fold cross validation. As we can see, both DT and ANN algorithms have the highest maximum accuracy and the CORER is ranked second.

Fig. 9 illustrates standard deviation of different classifiers. DT and SVM are ranked first and second in this item. Although the SVM has the best standard deviation, but the mean accuracy of this algorithm is about 3% lower than CORER classifier. It means that almost in all folds of ten-fold cross validation, the accuracy of CORER is more than SVM classifier.

[ Downloaded from journal.iirc.ac.ir on 2024-03-28 ]

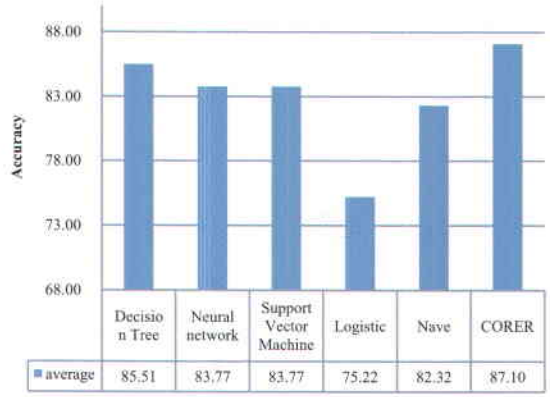



Figure 6. The Mean Accuracy

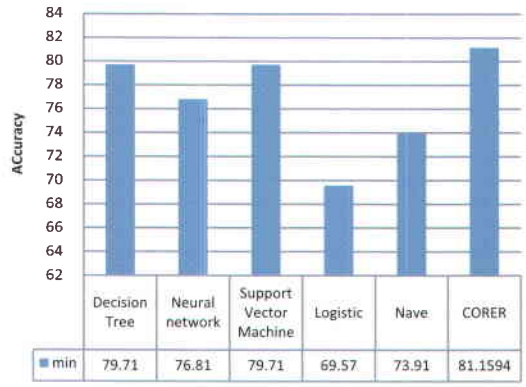



Figure 7. The Minimum Accuracy

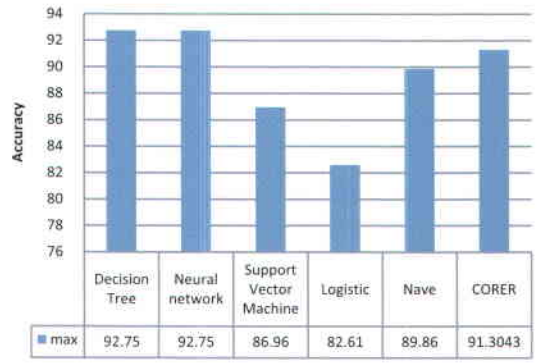



Figure 8. The Maximum Accuracy

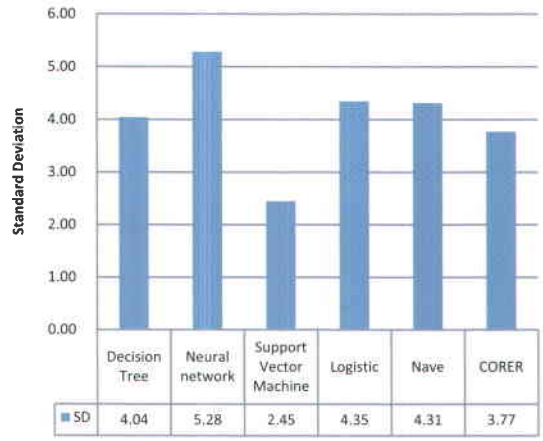



Figure 9. The Standard Deviation

CORER achieves the best mean accuracy in comparison with the other applied classification algorithms because of two main reasons. First of all, it has a comprehensive view on the all existing features. It may eliminate a specific feature of a specific rule in the current iteration and add that feature to that rule, in the next iteration and vice versa. Second, CORER works based on CCA algorithm. CCA has been used in several applications and experimental results of previous researches show that this method converges to a better point in comparison with the other applied algorithms [28-30]. According to the CCA algorithm, based on the cost function, CORER tries to improve the current rules performance in each iteration. It is worth mentioning that, there is not any iteration that leads to new rules with lower performance compared to previous iterations. Hence, the position of an imperialist rule with a colony one was changed only when the colony rule works better than its imperialist.

According to [12], CORER works very well in classification datasets with few features where there are few categories in each feature. It means that CORER performance is sensitive to the rule length and it works better for smaller rules. In conclusion, CORER algorithm can act better compared to other approaches when the rule length is not very long (about 30 bits).

Based on the experimental results, CORER may provide better performance for some critic fields which need more precise approaches. Also, since based on the previous studies, even one percent improvement in the accuracy of applied can strongly affects its performance and responsiveness of credit scoring systems [1, 5],our findings show that CORER classifier has better performance compared to the well-known algorithms in the credit scoring domain and also lead us to believe that CORER may have better outcome in other applications of banking.

## V. CONCLUSIONS

Credit risk assessment is becoming one of the most important topics in the field of financial risk management .In this paper we have applied a recently introduced rule generator classifier called CORER, in order to improve the accuracy of credit scoring classification task. In each stage, the applied classifier extracts one rule from the training set, based on CCA algorithm. CORER has two important properties. First, it has a comprehensive view on all the existing features. Second, it works based on CCA algorithm. We have tested the CORER by applying it to Australian credit real dataset from UCI Repository of Machine Learning Databases. CORER results gave us enough confidence to be sure about its superiority to many other classifiers due to our comparison with some other well-known classification methods.

Some future research directions also emerge. First, large datasets of credit scoring should be tested to further evaluation of CORER classifier. Another approach for improvement of credit scoring accuracy, could be gained by ensemble the output of CORER

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]

with other powerful classifiers using data fusion methods.

## ACKNOWLEDGMENTS

We thank the Iran Telecommunication Research Center for financial support. We wish to thank the developers of Weka. We also express our gratitude to the donors of the different datasets and the maintainers of the UCI Repository.

## REFERENCES

[1] G.Wang, J.Hao , J.Ma , H.Jiang "A comparative assessment of ensemble learning for credit scoring" Expert system with applications vol.38, 2011, pp.223-230.

[2] Zhang, D., X. Zhou, et al "Vertical bagging decision trees model for credit scoring," Expert system with applications, vol. 37, 2010, pp. 7838-7843.

[3] N.C.Hsieh "Hybrid mining approach in the design of credit scoring models," Expert system with applications vol.28, 2005 , pp. 655-665.

[4] E.Ngai, L. Xiu, and D. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification," Expert Systems with Applications, vol.36, 2009, pp. 2592-2602.

[5] W.Chen, C. Ma, and L. Ma, "Mining the customer credit using hybrid support vector machine technique," Expert Systems with Applications, vol.36, 2009, pp. 7611-7616.

[6] F.Chen and F. Li, "Combination of feature selection approaches with SVM in credit scoring" Expert Systems with Applications,vol. 37, 2010 pp. 4902-4909.

[7] T.Leea,Ch.Ch.Chiub,Y.-Ch.Chouc,Ch.J.Lud "Mining the customer credit using classification and regression tree and multivariate adaptive regression splines," Expert system with applications, vol.50, 2006, pp. 1113-1130.

[8] Ch.F.Tsai, J.W.Wu "Using neural network ensembles for bankruptcy prediction and credit scoring," Expert Systems with Applications, vol.34, 2008, pp. 2639-2649.

[9] C.S.On, J.Jeng, Huang, G.HshiungTzeng "Building credit scoring model using genetic programming", Expert System with application, vol.29, 2005, pp.41-47.

[10] L.C. Thomas "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers" international journal of forcasting vol.16, 2000, pp.149-172.

[11] N.C.Hsieh and L.-P. Hung "A data driven ensemble classifier for credit scoring analysis." Expert Systems with Applications, vol.37, 2010, pp.534-545.

[12] J. Basiri, F. Taghiyareh and S. Gazani, "CORER: A New Rule Generator Classifier," 13th IEEE CSE, Hong Kong, December 2010.

[13] E. Atashpaz Gargari, C. Lucas. "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition". IEEE Congress on Evolutionary Computation, Singapore, 2007, pp 4661-4667.

[14] P. Clark and T. Niblett, "The CN2 Induction Algorithm. Mach," Learn. 3, 4 March 1989, pp.261-283.

[15] E.Frank and Witten, I. H. "Generating Accurate Rule Sets Without Global Optimization," In Proceedings of the Fifteenth international Conference on Machine Learning (July 24 - 27, 1998). J. W. Shavlik, Ed. Morgan Kaufmann Publishers, San Francisco, CA, pp.144-151.

[16] L.Nanni, A.Lumini "An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring", Expert system with applications, vol.36, 2009, 3028-3033.

[17] C.L.Huang, M.C.Chen, C-J.Wang "credit scoring with datamining approach based on support vector machine," Expert System with application vol.37, 2007, pp. 847-856.

[18] A.Ghorbani , F.Taghiyareh, C.Lucas "Application of locally linear model tree on customer churn prediction" SOCPAR, 2009.

[19] S.TunLi ,W.Shiue , M.H.Huang "The evaluation of consumer loans using support vector machines," Expert System with application, vol.30, 2006, pp.772-782.

[20] M.T.Mahmoudi, N. Forouzideh, C.Lucas, F. Taghiyareh. "Artificial Neural Network Weights Optimization based on Imperialist Competitive Algorithm," Seventh International Conference on Computer Science and Information Technologies, Yerevan, Armenia, 28 September - 2 October, 2009.

[21] K.A.De Jong, Spears, W. M., and Gordon, D. F. "Using Genetic Algorithms for Concept Learning," Mach. Learn, Vol.13, 1993, pp.161-188.

[22] Han, J. and M.Kamber, "Data Mining: Concepts and Techniques" Morgan Kaufmann Publishers Inc, 2000.

[23] J.R.Quinlan," Induction of Decision Trees," Mach. Learn. vol.1, 1986, pp.81-106.

[24] W. W. Cohen, "Fast effective rule induction," In Machine Learning: the 12th International Conference, Lake Taho, CA, 1995, pp. 115-123. Morgan Kaufmann.

[25] J. Hadden, A. Tiwari, R. Roy, D. Ruta, "Computer assisted customer churn management: State-of-the-art and future trends", Computers and Operations Research, vol. 34, no. 10, 2007, pp. 2902-2917.

[26] X. Jin, A. Xu, R. Bie, P. Guo," Machine Learning Techniques and Chi-Square Feature Selection for Cancer Classification Using {SAGE} Gene Expression Profiles," Data Mining for Biomedical Applications, {PAKDD} Workshop, Bio{DM} Proceedings, vol. 3916, Springer, 2006, pp. 106-115.

[27] H. Witten and E. Frank, Data Mining "Practical Machine Learning Tools and Techniques," 2nd ed, Morgan Kaufmann, 2005 <http://www.cs.waikato.ac.nz/ml/weka>.

[28] E. Gargari, F. Hashemzadeh, R. Rajabioun et al, "Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process," International Journal, vol.1, 2008, pp 337 - 355.

[29] R. Rajabioun, E. Atashpaz-Gargari, and C. Lucas, "Colonial competitive algorithm as a tool for Nash equilibrium point achievement," Computational Science and Its Applications-ICCSA 2008, pp. 680-695.

[30] A. Biabangard-Oskouyi, E. Atashpaz-Gargari, N. Soltani et al, "Application of Imperialist Competitive Algorithm for Material Properties Characterization from Sharp Indentation Test," Int J Eng Simul, vol. 10, 2009.



**Javad Basiri** received his B.Sc. degree in Information Technology Engineering from University of Isfahan, in 2008 and his M.Sc. degree from University of Tehran, 2011. Currently he has started his PhD at University of Tehran, Electrical and Computer Eng., IT Department. His research interests include Customer Relationship Management, Data Mining, Data Fusion, Recommender Systems, Evolutionary Algorithms, and Information Retrieval.



**Fattaneh Taghiyareh** is Assistant Professor of Computer Engineering-Software & Information Technology, at the University of Tehran, where she has served since 2001. She received a Ph.D. in Computer Engineering- Parallel Algorithm Processing from the Tokyo Institute of Technology in 2000. Her research interests include Human-Centered Computing (HCC) applied to Learning Management Systems and based on Multi-Agent Systems. She is currently working on "Group Collaborative Learning" as well as "Creative Thinking" with an existential approach. In addition, Web Service composition and interoperability are another concerns of her research.

[ Downloaded from journal.iricac.ir on 2024-03-28 ]

Currently she's been envolved in a national project in cooperation with ITC Organization to stablish e-Learning Laboratory as a pilot for National Learning Network.



**Mohammad Siami** received the B.Sc. degree from University of Isfahan, Iran, in 2008. He is currently working toward the M.Sc. degree in Department of Industrial Engineering at the Iran University of Science & Technology at Tehran. His research interests include Business Intelligence, Customer Relationship Management, Data Mining, Data Fusion and Credit Scoring in banking industry.



**M. R. Gholamian** received his B.Sc. and Ph.D. degree in Industrial Engineering from Amirkabir University of Technology, Tehran, and his M.Sc. degree in Industrial Engineering from Isfahan University of Technology, Isfahan in 1998. He is currently an Assistant Professor in the Department of Industrial Engineering at the Iran University of Science & Technology at Tehran. His research interests are in intelligent systems and multicriteria decision making. He has published four books, 50 papers in the international conferences and 15 papers in the international journals.

[ Downloaded from journal.itrc.ac.ir on 2024-03-28 ]