A Novel Imperialist Competitive Algorithm for Automated Mining of Association Rules

Fariba Khademolghorani
Department of Computer Engineering
Islamic Azad University
Najafabad Branch, Iran
Fariba.Khadem@sco.iaun.ac.ir

ABSTRACT: *Association rule mining can be considered as an optimization problem. A lot of algorithms have been introduced in the area, but they suffer from several limitations. Recently, imperialist competitive algorithm (ICA) has been introduced for solving different optimization problems. This paper proposes a novel ICA algorithm for automated mining of the interesting and readable association rules without considering the minimum support and the minimum confidence thresholds. In this algorithm, the convergence rate and the computational efficiency of ICA have been improved. These modifications on ICA includes modification of the modeling the assimilation policy and combining it with a mutation operator of a genetic algorithm, which lead to increasing the exploration of the algorithm and on the other hand lead to the improvement of the convergence rate. The value of the mutation probability is automatically determined without requiring to be specified by the user. The experimental results indicate the efficiency of this algorithm in comparison with the methods of mining association rules based on the basic ICA and the genetic algorithm. Thus, these modifications are not only useful for association rule mining, but also it can be extended to other optimization problems.*

Keywords: Association Rules, Imperialist Competitive Algorithm, Genetic Algorithm, Evolutionary Algorithm

Received: 11 July 2011, Revised 5 August 2011, Accepted 8 August 2011

1. Introduction

With the development of information technology, there are many different kinds of information databases, such as medical data, financial data, and marketing transaction data. In recent decades, data mining techniques have been the most widely applied to find the critical hidden knowledge from these databases and predict the future behavior of them. Data mining can be categorized into several tasks, including association rule, sequential patterns, time series, clustering, classification, etc [1].

Association rules play very important role in recent scenario of data mining and basically use for finding out the useful patterns, relation between items found in the database of transactions [2]. Transactional database refers to the collection of transaction records, which in most cases are sales records. When a data set consist the continuous values, it becomes hard to mine the data and some special techniques need to be prepared [3]. Association rule consists of many modified algorithms based on Agrawal, Apriori [4], SETM [5], AIS [4], DIC [6] and other methods that focus on improving its efficiency and accuracy.

However, these algorithms have their limitations. They have to mine association rules in two stages separately. In these methods, rules with high occurrence in database are considered as the best rules, whereas most of these rules can easily be predicted by the users. Therefore, they are not interesting. Also, they mine rules with a large number of attributes, which are not understandable for the user. Hence, the user will never use them. Moreover, two parameters, minimum support and minimum confidence thresholds, are always specified by the decision-maker him/herself or through trial-and-error; and hence, these algorithms lack both objectiveness and efficiency.

In recent years, in addition to common methods, some evolutionary algorithms have been used with multi-objective functions. Evolutionary algorithms such as genetic, ant colony, simulated annealing and particle swarm optimization have been used for mining association rules. In order to get the aforementioned objectives, the presented work which is almost like the method in [7], proposes an efficient imperialist competitive algorithm for mining interesting and understandable association rules, which it does not need the determination of the minimum support and the minimum confidence thresholds. Also, it extracts these rules as a single run, whereas the previous methods mine them in two stages. This paper emphasizes more on the improvement of the convergence rate and increasing the exploration of the algorithm to global optimal solutions. Therefore, some modifications have been made to the basic ICA, such as adding the mutation operator with adaptive rate and improving the modeling the assimilation policy. The values of these two parameters are automatically determined; hence this algorithm is very user friendly. In this way, we improve ICA and our proposed method in [7].

The rest of this paper is organized as follows: Section 2 briefly introduces the general background about the association rules, related works and the imperialist competitive algorithm; Section 3 describes the proposed algorithm in details, then the experimental results will be discussed in Section 4. Finally, conclusion and future works will be presented in Section 5.

2. Background

2.1 Definition of association rule

Agrawal et al in 1993 [4] first proposed an association rule algorithm in order to analyze the customers' market basket. They showed that some hidden relationships exist between purchased items in transactional databases. Therefore, these results can help decision-makers understand customers' purchasing behavior [8]. An association rule is in the form of A=>C, where A and C represent antecedent and consequent itemsets of the rule, respectively. The general transaction database ($D = \{T_1, T_2, \ldots, T_n\}$) can represent the possibility that a customer will buy product C after buying product A and $A \cap C = \phi$. $I = \{I_1, I_2, \ldots, I_n\}$ is a set of all items appearing in the transaction database. Any non-empty subset x from I is called an itemset, an itemset that contains k items is called a k-itemset. The basic framework for mining association rules includes two stages. First, find frequent itemsets; second, generate association rules based on these frequent itemsets. However, two measures of support and confidence are introduced for evaluating association rules. These are calculated from (1) and (2) equations, respectively [1]:

a) Support (A=>C): Finding itemsets with their supports above the minimum support threshold which is called "frequent itemset".

$$\text{Sup}\,(A => C) = \frac{\text{number of transactions which contain A \& C}}{\text{number of transactions in the database}} \tag{1}$$

b) Confidence (A=>C): Using frequent itemsets found in (1) to generate association rules that have confidence levels above the minimum confidence threshold.

$$Conf\,(A => C) = \frac{\text{number of transactions which contain A \& C}}{\text{number of transactions which contain A}} \tag{2}$$

These rules which satisfy both minimum support and minimum confidence threshold are called strong association rules [1].

2.2 Related works

Association rule mining is one of the most important techniques in data mining. There has been introduced many association rule mining algorithms based on the proposed methods by Agrawal in [4] and [9]. In addition, an approach has been introduced for mining association rules in large relational tables including both quantitative and categorical attributes [10]. On the other hand, recently optimization methods for mining association rules have been applied.

The idea of using a genetic algorithm has been employed for discovering only frequent itmesets [11]. In another study, a new method using a genetic algorithm has been proposed for finding negative and optimized association rules [3]. In [12], other researchers applied a genetic algorithm for discovering association rules from a manufacturing information system (MIS) dataset. The results of applying this method showed that the genetic algorithm is more efficient.

In another study, an ant colony system has been applied for mining multi-dimensional rules. The test results indicated that the proposed method could extract more condensed rules than Apriori method [13]. Later, this method was combined with the clustering method to provide more meticulous rules [14]. Also, a method of extracting association rules by using multi-objective genetic algorithm has been proposed in [15].

In [16], a method of exploring frequent itemsets has been proposed by combining PSO with Ant algorithm. This method, in comparison with GAR algorithm in [11] is faster and has more accuracy but it can discover only frequent itemsets and GAR has the same limitation. In [8], by using particle swarm algorithm has been proposed a method for mining association rules.

In [17], has been extracted association rules based on the genetic algorithm. In this method, relative confidence was used as the fitness function. But only rules with fixed length were extracted. Also, the fitness function of this method is in such a way that it is trapped into local optimum, and hence many rules are generated. Later, other researchers improved it by defining a new multi-objective fitness function [18]. In [19] simulated annealing algorithm has been used to develop a multi-objective rule mining method.

In another study, other researchers used the gravitational search algorithm to develop a multi-objective association rule mining method [20]. The results of applying this method in comparison with the proposed method in [8] showed that it is more able to discover global solutions and individuals evolve into the convergent proper positions with a higher fitness value.

But so far, there is no which uses imperialist competitive algorithm for mining association rules, except in [7]. Therefore, in this paper by improving the proposed method in [7], we tried to provide a useful method for extracting association rules. In the next section we briefly introduce this evolutionary algorithm.

2.3 Imperialist competitive algorithm

Imperialist competitive algorithm (ICA) is a new evolutionary algorithm that has been recently introduced for solving different optimization problems. This algorithm is a global search strategy and is based on the sociopolitical competition among empires [21].

Like other evolutionary algorithms, ICA starts with an initial population of individuals which are called countries and the countries are separated to two different sorts based on their cost, and defined as imperialist and colony respectively, which together form empires. The division of all the colonies of initial countries is based upon the power of the imperialist. This means that the more powerful imperialist, have the more colonies. After dividing all colonies among imperialists and creating the initial empires, these colonies start moving toward their relevant imperialist state based on assimilation policy. In contrast with the assimilation policy, there is a revolution in the countries of each empire. In ICA, revolution causes a country to suddenly change its socio-political characteristics. The revolution increases the exploration of the algorithm and prevents the hasty convergence of countries to local minimums. The total power of an empire is defined by the imperialist's power and percentage of the colonies' power. Then the imperialist competition begins among the empires. All empires try to dominate the colonies of other empires under their control. Then, the imperialistic competition will gradually result in an increase in the power of powerful empires and a decrease in the power of weaker empires. This results in the collapse of weak empires. Finally, these processes will hopefully cause all the countries to converge to a situation in which there exists only one empire in the world and all the other countries are colonies of that empire and they have the same position and power as the imperialist [21].

3. MICA in association rule mining

This section describes MICA algorithm which is proposed in this paper for mining interesting, readable and understandable association rules and it is based on one of the latest evolutionary algorithms which is called the imperialist competitive algorithm (ICA). The mica is one of the most important materials in technology which is compatible with the concept of our algorithm and its application in data mining to explore the interesting and readable of association rules, which is important in this area; hence, this algorithm has been named MICA (Mining association rules via a novel ICA). Since, ICA has been introduced as a method based on the minimum value-finding; in this work, it has been changed into a searcher of the maximum value in the space. The proposed algorithm consists of a series of stages that Figure 1 illustrates the algorithm structure with an overview. It is noticeable that evolutionary algorithms are not necessarily appropriate for solving any optimization problems. On the other hand, there are different methods for solving an optimization problem with applying a specified evolutionary algorithm; which many researchers have shown in their researches.

In this paper, first, by evaluating the strengths and weaknesses of ICA algorithm and by considering the association rule mining problem, ICA algorithm has been improved. These modifications are added in the two stages of the algorithm structure in Figure 1, where the margins are shown in bold. For example, in this study, for revolution modeling, ICA has been combined with a mutation operator of a genetic algorithm, which leads to increasing the exploration of the algorithm. Also, unlike the other genetic algorithms, this method does not require to specifying the mutation rate before the start of the algorithm. Hence, it is very user friendly.

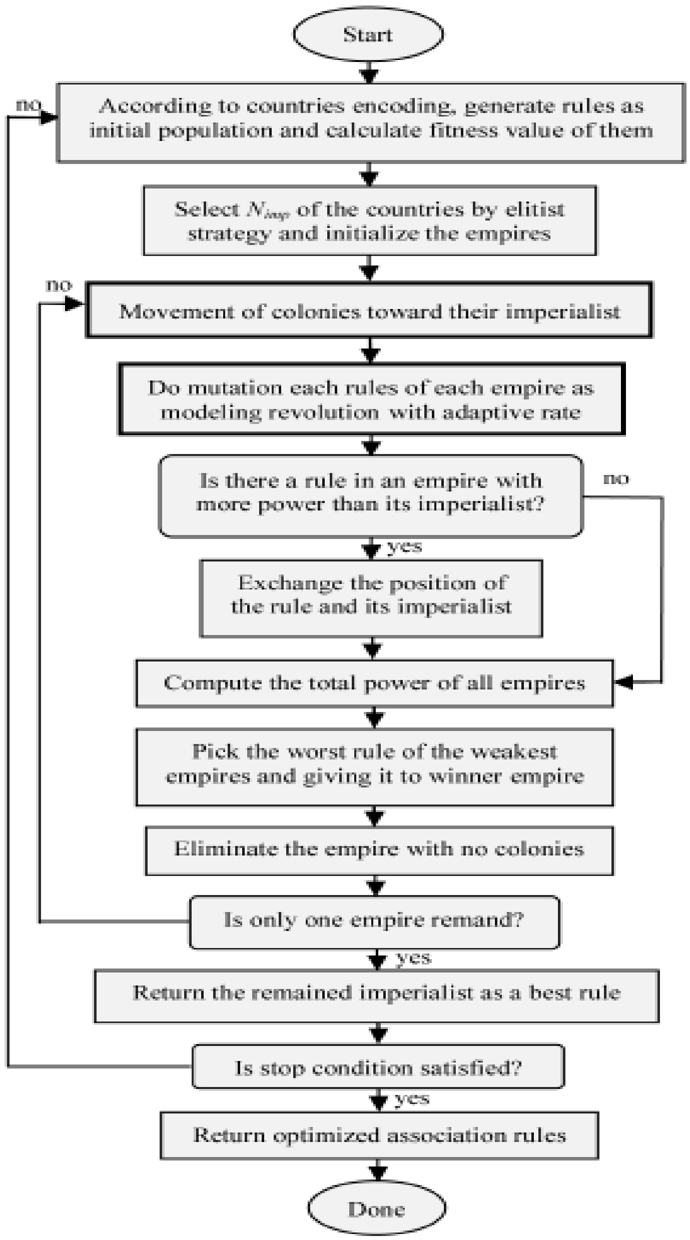

Figure 1. The structure of the proposed algorithm

Moreover, in the stage of the moving colonies toward the relevant imperialist, with considering association rule mining problem, a method is presented which reduces the re-computation time. Finally, this modification leads to increasing the convergence rate of the algorithm, as well as offering promising results.

The following sub-sections describe some important parts of MICA algorithm which are explained: countries representation, modeling the assimilation policy, modeling the revolution with adaptive rate, and finally fitness function of the proposed algorithm is explained.

3.1 Countries representation

In this work, each country represents an association rule and each rule contains of a series of decision variables which represents the status of every item (attribute) in the rule. Each attribute is considered as one dimension of the solution. The structure of each country's representation in MICA algorithm is shown in Figure 2.

Figure 2. The structure of a countries representation as a rule

Every country has *n* decision variables in lieu of *n* items in any database, where the *j*th variable which is known as S_{ij} represents the antecedent or consequent of *j*th item of the rule *i* and can take three values: '0', '1' or '2'. In this way, if S_{ij} is '0', it means that the *j*th attribute is in the antecedent of the rule *i* and if it is '1', this attribute is in the consequent of the rule *i*. If it is '2', it means that this attribute is not in this rule. Suppose that in a dataset, there are five attributes: egg, cake, milk, bread and sugar, respectively. Consider following example,

If *cake* ***and*** *milk* ***then*** *sugar*

Now, following proposed approach and country encoding, for simplicity usage, this rule can be represented as 20021.

3.2 Modeling the assimilation policy

The movement of colonies in each of the empire toward the relevant imperialist is the motivation of the assimilation policy. This means that the colonies can be like their imperialist and improve their positions.

In this work, for modeling the assimilation policy in mining association rule problem, a method has been introduced that is almost like the proposed method in [7]. In this way, for each colony, some attributes of the relevant imperialist are copied randomly in the colony. But unlike the proposed method in [7], here these attributes are just selected randomly from among different bits between the bits of the colony's rule and the imperialist's rule, which leads to the increase of the convergence of the algorithm and the reduction of the re-computation time. Therefore, this approach is more appropriate.

In this method, first, for each colony, the amount of *d* as the distance between the colony's rule and the imperialist's rule is calculated. *d* is the number of different bits between the bits of the colony's rule and the imperialist's rule. For example, in Figure 3, different bits between the imperialist and the colony have been determined; it also shows that this distance between them is four.

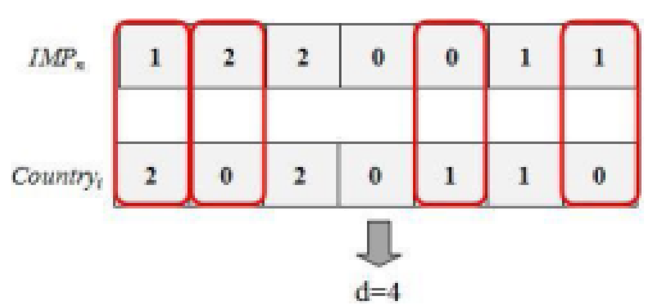

Figure 3. Calculate the distance between the imperial'rule and the colony' rule

Then, for each colony, the random number *n*, between *0* and *d* with uniform distribution is selected. Then, *n* bits of the colony are changed randomly which are different between the colony's rule and the imperialist's rule. It means that all bits selected as '0' are converted to '1'; all bits selected as '1' are converted to '2' and all bits selected as '2' are converted to '0'.

3.3 Modeling the revolution with adaptive rate

In contrast to the assimilation policy, there is a revolution in the colonies of each empire. In this work, for modeling the revolution, ICA has been combined with a mutation operator of a genetic algorithm. It has been modeled in the following manner: for each colony of an empire, a real random number between 0 and 1 is selected. If this number is lower than the mutation rate, the colony will have the chance of mutation occurring. Therefore, the position of an attribute in the colony's rule is changed, which has a great effect on the support, confidence and fitness value of that rule. This is done with the aim of moving the position of empires colonies to the global optimized solution. Also, unlike the other genetic algorithms and the proposed method in [7], this method does not require to specifying the mutation rate before the start of the algorithm and its value is calculated for each empire automatically as (3):

$$P^{n}_{m_{col_i}} = 1 - \left| \frac{fit^{n}_{col_i}}{\sum_{j=1}^{N_{col \in n}} fit^{n}_{col_j}} \right| \quad (3)$$

fit^{n}_{coli} is the fitness of *i*th colony. P^{i}_{coli} is the mutation probability for ith colony. 1- P^{i}_{coli} is the normalized cost of the ith colony. Since the weak colonies are more likely to be collapsing, in this method the deviational behavior of these colonies will be automatically more. This is done with the aim of moving the position of weak colonies from the unfavorable spaces of the problem toward better solutions and finally increasing the power of the empire. In this way, the individuals with high fitness search the space more locally and the individuals with low fitness search the space more globally. Modeling the revolution with the rate of adaptive changes increases the exploration of the algorithm and prevents the hasty convergence of countries to local optimum solutions.

It is noteworthy that after any changes on the dimensions of any individual, its rule validity will be checked out. Every individual's rule should have at least one attribute in its antecedent and one in its consequence.

3.4 Fitness function

The fitness function should be determined to the specific search spaces, thus choice of fitness function is very important to get the desired results. This paper focuses on mining interesting and readable rules.

a) Interestingness

Mining association rule is a task that extracts some hidden information, it must discover those rules that have a comparatively less occurrence in the entire database which are more interesting for the users; discovering such rules is more difficult. For classification rules, measure like information gain, can be useful. But it is not efficient for evaluating the association rules. Therefore, interestingness measure in [15] is used in the fitness function which is computed by (4).

$$interestin\ gness_i = \left[\frac{\sup_i(A \cup C)}{\sup_i(A)} \right] \text{ x } \left[\frac{\sup_i(A \cup C)}{\sup_i(C)} \right] \text{ x } \left[1- \frac{\sup_i(A \cup C)}{|R|} \right] \quad (4)$$

This relation has three parts: [$\sup_i(A \cup C)/\sup_i(A)$] indicates the probability of creating the rule ith depending on the antecedent part; [$\sup_i(A \cup C)/\sup_i$(C)] shows the probability of creating rule ith depending on the consequent part. In fact most of these are interesting rules in which the rate of acquired information is approximately the same in both antecedent and consequent parts of the rule. In this parameter the support count of the rule antecedent and the support count of the rule consequent are used. In the third part of this parameter, [$\sup_i(A \cup C)/|R|$] gives the probability of generating the rule ith depending on the whole dataset. In this parameter $|R|$ is the total number of records in the database. So complement of this probability will be the probability of not generating the rule. Because, those rules have a very high support count and high frequency will be less interesting, and such rules are easily predictable by the users.

b) Readability

Readability is another objective used in MICA. This parameter rewards the shorter rules with a smaller number of attributes. Readability and comprehensibility of rules that are important in data mining are increased. It is known that larger rules have more redundant information. In result, it is difficult to understand the rules and the user will never use them. Readability has been computed by (5) where $|\, attribute_i(A \cup C)\, |$ is the number of attributes that exist in total rule i and $|attribute|$ is the total number of attributes in database.

$$readability_i = 1 - \frac{|\, attribute_i(A \cup C)\, |}{|\, attributes\ (D)\, |} \tag{5}$$

In this paper, the weighted sum fitness function is calculated for every individuals of the population by (6). In fact, all objectives have been weighted in order to give them different importance.

$$fitness_i = \sum \omega_j O_{ij} \ , \quad 0 \le \omega_j \le 1 \tag{6}$$

Here, ω_j the weight for objective j, and O_{ij} is fitness of the jth objective of rule i. Each ω_j will be specified by the percentage of user's interests and one might increase or decrease the effects of parameters of the fitness function. It means that they do not need to assign for each database in spite of some thresholds such as minimum support and minimum confidence in the previous methods. Also, it should be noted that another objective can be added.

4. Implementation and experimental results

MICA algorithm was implemented and executed in C#, on a PC with Intel Dual-Core 2.1 GHz operator on a 2GB Ram. The setting of the used parameters is shown in Table 1. The values of the weighted coefficients for the interestingness and readability objectives which have been introduced in the proposed fitness function were selected as 0.9 and 0.1, respectively. These coefficients are specified by the percentage of user's interests. Hence, they do not need to reinitialize for each database in spite of some thresholds such as minimum support and minimum confidence.

$N_{Country}$	N_{Gen}	N_{Imp}
70	100	10

Table 1. The used parameters valuess for running MICA

The proposed method was executed and evaluated on four real datasets: balance scale, nursery, car evaluation, basket market. For doing examinations we used the basket market dataset existing in Clementine 12.0 tool [22] and three datasets in UCI at http://www.ics.uci.edu/~mlearn. The specifications of datasets used are given in Table 2. All attributes in datasets are categorical except basket market dataset; therefore we converted them into the Boolean datasets. It means that every attribute with any amount is considered as an item. The basket market dataset includes 11 binary attributes and it has the least attributes in this work. Nursery dataset has the most records over than 10000 examples and also, it has the more attributes than the rest of the dataset.

Dataset	$N_{Records}$	$N_{Attributes}$
Balance Scale	625	23
Nursery	12960	32
Car Evaluation	1728	28
Basket Market	1000	11

Table 2. The specification of datasets

The first experiment is evaluating the convergence rate and the performance of the proposed algorithm to discover the best rules by considering the introduced objectives in this study. Therefore, MICA is compared with MINICA [7] and the method of mining association rules which is based on the GA [15]. The initial setting of the used parameters values for the GA [15] is given in Table 3.

N_{POP}	N_{Gen}	$P_{Crossover}$	$P_{Mutation}$ (5 point)
70	100	0.8	0.02

Table 3. The used parameters values for the proposed GA

In order to compare the experiments fairly, we examined them by the presented fitness function in this work. Also, we examined MINICA algorithm with the same parameters as in Table 1. This experiment was done with 20 replications. So, the mean of the results was calculated. Figures 4 to 7 illustrate the convergence speed and accuracy of these algorithms and the lines in these figures show the fitness of the best imperialist in each generation by any of the three algorithms.

As it is depicted in Figure 6, MINICA and MICA algorithms have a better performance than the GA algorithm. Although, MINICA has a better fitness at the 15th iteration, finally, MICA has a better ability to discover the best rules.

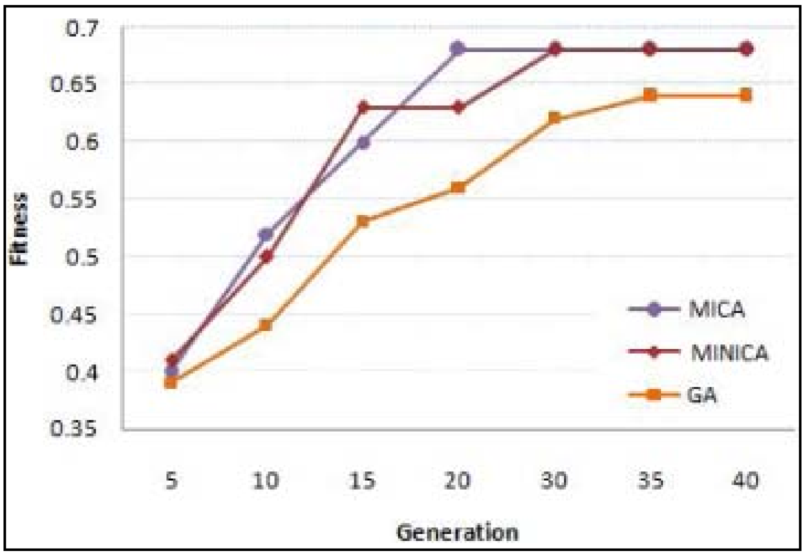

Figure 4. Convergence rate on the Balance Scale dataset

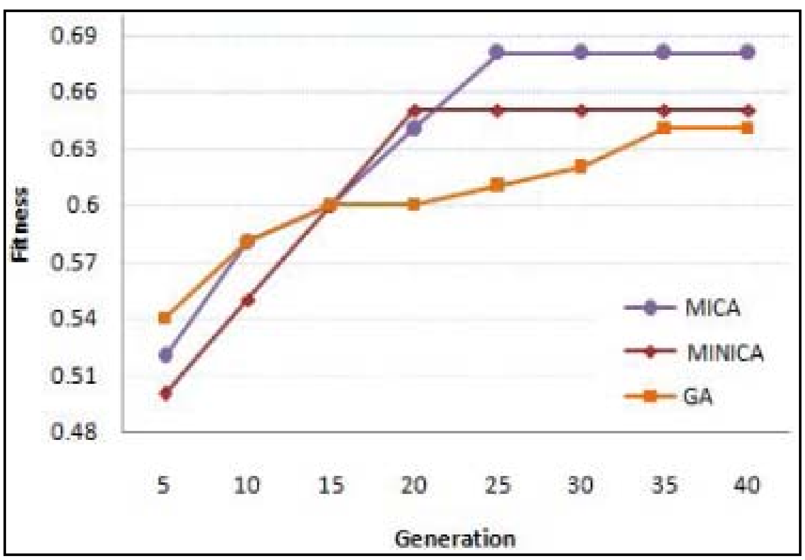

Figure 5. Convergence rate on the Nursery dataset

In the Figure 5, the performance of MINICA on the car evaluation dataset is better than the GA [15] but MICA indicates a better performance both in convergence speed and in obtaining the quality of the association rules in compaction to the other two algorithms.

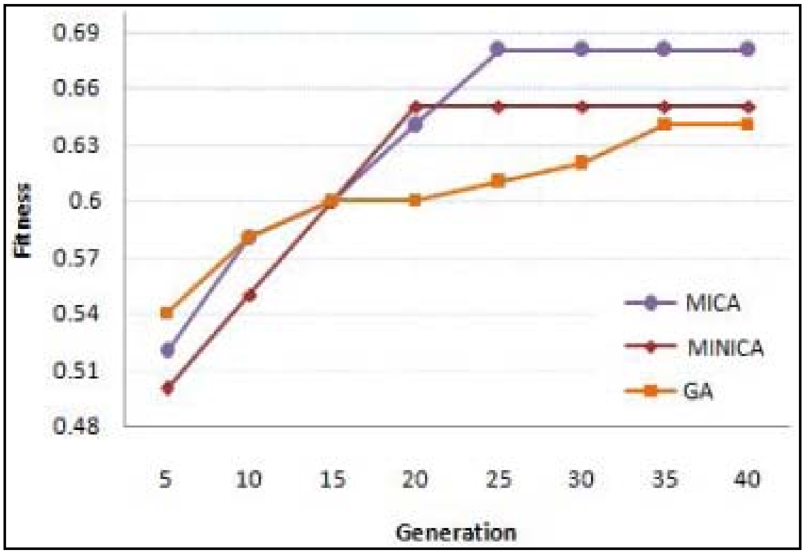

Figure 6. Convergence rate on the Car Evaluation dataset

According to Figure 6, at the first 10 iterations, the GA algorithm has a better fitness than MICA and MINICA algorithms, but from the 15th iteration to the end, two other algorithms have been achieved to better rules. As it can be seen, MINICA has converged faster than MICA. But it is not an important issue, because MICA has been able to escape from the local solutions and reach to global optimal solutions.

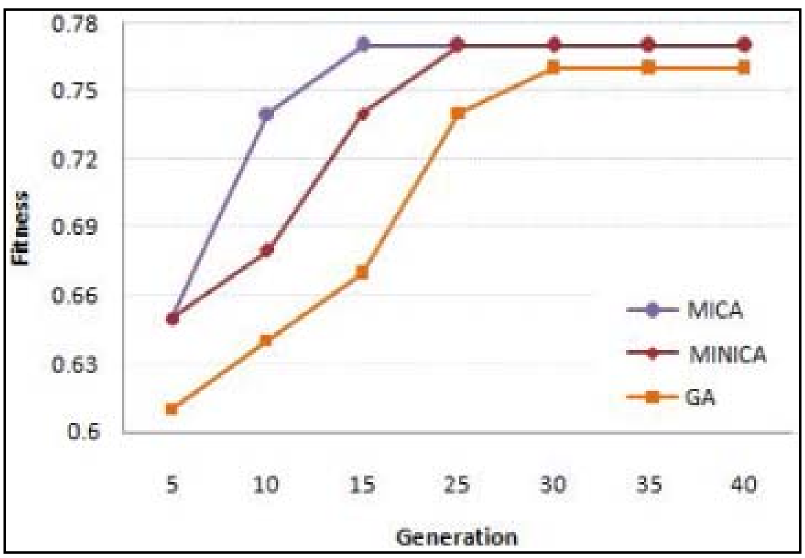

Figure 7. Convergence rate on the Basket Market dataset

In Figure 7, it is observed that MICA and MINICA have a better performance in the quality of optimal solution and in convergence speed rather than the GA algorithm. Also, MICA has a better convergence speed in comparison to MINICA.

As it was seen, MICA and MINICA algorithms that are based on ICA have better performance with least error in compare with the algorithm based on the GA for mining association rule problem. In ICA, individuals have more interacted with each other and thus these algorithms are more likely to discover global solution. Also, MICA has a better efficiency than MINICA algorithm both in the quality of optimal solutions and in convergence speed. On the other hand, it has better ability to avoid hasty

Parameters	Datasets				
	Methods	Balance Scale	Nursery	Car Evaluation	Basket Market
Support %	MICA	7.22	18.11	6.12	15.88
	MINICA	7.24	18.21	6.14	15.87
	GA [15]	5.32	17.55	5.61	14.38
Confidence %	MICA	57.10	71.04	45.68	58.27
	MINICA	57.02	69.02	43.54	58.24
	GA [15]	48.37	67.42	39.33	52.77
$N_{Attributes}$	MICA	2.5	9.1	3.1	2.3
	MINICA	2.5	9.3	3.2	2.3
	GA [15]	4.2	9.4	3.9	3.2
Readability	MICA	0.89	0.71	0.90	0.79
	MINICA	0.89	0.71	0.89	0.79
	GA [15]	0.80	0.70	0.86	0.70
N_{Rules}	MICA	31	35	20	10
	MINICA	32	35	19	10
	GA [15]	28	35	22	9
Cosine	MICA	0.48	0.59	0.44	0.59
	MINICA	0.47	0.56	0.38	0.57
	GA [15]	0.39	0.52	0.33	0.51

Table 4. Comparisons of the results

convergence, because MICA is based on the improved ICA, which has been introduced in this work by the generalization of the basic ICA. In fact, by modification of the modeling the assimilation policy and by combining ICA with a mutation operator of a genetic algorithm with adaptive rate, an individual with a higher fitness searches the space with slow movements and hence searches the space more locally. Also, an individual with a lower fitness searches the space with rapid movements and searches the space more globally. These figures depicted that the modifications applied are really useful.

The results of the final test in this paper have been shown in Table 4, which is about the comparison of the details of the mean results obtained from MICA, MINICA [7] and the proposed GA in [15]. The mean number of best different mined rules, the mean number of attributes contained in the rules, the mean of the readability, support, confidence and cosine value of these rules are shown in this table. The cosine measure can be used for evaluating the interestingness of the association rules. According to this table, MICA presents better results in comparison with MINICA; on the other hand, MINICA has better results in comparison with the association rules mining method in [15]. MICA and MINICA algorithms could discover rules with appropriate support and high confidence value in comparison with the GA in [15]. Also, the number of attributes obtained in the rules by these two methods is smaller; hence, the readability of these rules is more. As mentioned before, larger rules are more likely to contain redundant information.

In addition, in these two methods, the mean cosine value which can be used for evaluating the interestingness of the rules is more than the GA [15]. On the other hand, with a closer look at the obtained mean results by MICA and MINICA, we can say that MICA can provide better results on some datasets (such as, nursery and car evaluation that have large attributes and records) with a more difference and it can provide the desired results on other datasets with a little difference.

As mentioned before, this paper has focused more on the improvement of the running time by using less iteration and less re-computation as well as increasing the exploration of the algorithm to global optimal solutions. These purposes have been achieved through the introduced modifications in sections 3.2 and 3.3. According to the all the results of the experiments, the conclusion drawn stated that MICA had considerably higher efficiency.

5. Conclusion and future works

This paper proposed a novel ICA algorithm for mining association rules without considering the minimum support and the minimum confidence thresholds which are difficult to determine for each database. The algorithm extracts automatically the readable and interesting rules as a single run. In addition, other objectives can be added to the fitness function. This algorithm has been called MICA, which is extending of MINICA algorithm. This paper emphasized more on the improvement of the performance both in convergence rate and in obtaining the quality of the association rules, so to achieve these purposes, several modifications have been presented that includes modifying the modeling the assimilation policy and combining that algorithm with a mutation operator of a genetic algorithm with adaptive rate. The most important aspect of this algorithm is that the value of the mutation probability is automatically determined; hence this algorithm is very user friendly. In this way, modeling the assimilation policy and modeling the revolution in ICA and our proposed method in [7] have been improved. In MICA, an individual with a higher finesse searches the space with slow movements and hence searches the space more locally. Also, an individual with a lower fitness searches the space with rapid movements and searches the space more globally. MICA's performance was evaluated through several tests in various real datasets. The results of all experiments in this work indicate that MICA has a better ability to discover global solutions with a fast convergence rate in comparison to MINICA and the mining association rules which is based on the GA [15].

Regarding that MICA algorithm has an appropriate structure for parallel architecture, so the parallelization of this method can be useful to reduce the run-time. Moreover, in this paper, for association rule mining, ICA has been improved; this improved version can be applied to other optimization problems as a future work. In addition, this algorithm can be extended for mining the weighted association rules.

References

[1] Han, J., Kamber, M. (2006). Data Mining: Concepts and Techniques, Morgan Kaufmann in Elsevier, Second Edition, San Francisco.

[2] Agrawal, R., Imielinksi, T., Swami, A. (1993). Database mining: a performance perspective, In Proceedings of IEEE Transactions on Knowledge and Data, 5 (6) 914–925.

[3] Saggar, M., Agrawal, A.K., Lad, A. (2004). Optimization of association rule mining using improved genetic algorithms, *In.*: Proc. of the IEEE International Conference on Systems Man and Cybernetics, p. 3725–3729.

[4] Agrawal, R., Imielinski, T., Swami, A.N. (1993). Mining association rules between sets of items in large databases, *In.*: Proc. of ACM SIGMOD international conference on Management of data, Washington, 22 (2) 207–216.

[5] Houtsma, A., Swami, M. (1993). Set-oriented mining of association rules, Research Report RJ 9567, IBM Almaden

Research Center, San Jose, California.

[6] Birn, S., Motvani, R., D.Ullman, J., Tusr, S. (1997). Dynamic itemset counting and implication rules for market basket data, *In.*: Proc. of the ACM SIGMOD international conference on Management of data, New York, 26 (2) 255–264.

[7] Khademolghorani, F. (2011). An effective algorithm for mining association rules based on Imperialist Competitive Algorithm, *In.*: Proc. of the sixth International Conference on Digital Information Management (ICDIM 2011), Melbourne, Austria, p. 6-11.

[8] Kuo, R.J., Chao, C.M., Chiu, Y.T. (2009). Application of particle swarm optimization to association rule mining, Elsevier Applied Soft Computing, 11(1) 326-336.

[9] Agrawal, R., Srikant, R. (1994). Fast algorithm for mining association rules in large databases, Research Report RJ 9839, IBM Almaden Research Center, San Francisco, CA, USA, p. 487-499.

[10] Srikant, R., Agrawal, R. (1996). Mining quantitative association rules in large relational tables, *In.*: Proc. of ACM SIGMOD international conference on Management of data, Montreal, 25 (2) 1-12.

[11] Mata, J., Alvarez, J., Riquelme, J. (2002). Discovering numeric association rules via evolutionary algorithm, *In.*: Proc of sixth Pacific–Asia conference on knowledge discovery and data mining PAKDD-02 (LNAI), Taiwan. p. 40-51.

[12] Li, C., Yang, M. (2004). Association rule data mining in manufacturing information system based on genetic algorithms, *In.*: Proc. of the 3rd International Conference on Computational Electromagnetics and Its Applications, p. 153–156.

[13] Kuo, R.J., Shih, C.W. (2007). Association rule mining through the ant colony system for National Health Insurance

Research Database in Taiwan, Computers an Mathematics with Applications, Tarrytown, NY, USA, 54 (11-12) 1303– 1318.

[14] Kuo, R.J., Lin, S.Y., Shih, C.W. (2007). Discovering association rules through ant colony system for medical database in Taiwan, International Journal of Expert Systems with Applications, 33 (3) 794-808.

[15] Ghosh, A., Nath, B. (2004). Multi-Objective Association Rule Mining Using Genetic Algorithm, Elsevier Information Sciences, 163 (1) 123–133.

[16] Badawy, O.M., Habib, M.I., Sallam, A.A. (2008). Quantitative Association Rule Mining Using a Hybrid PSO/ACO Algorithm (PSO/ACO-AR), *In.*: Proc. of Arab Conference on Information Technology (ACIT'2008), Hammamet, Tunisia, p. 1-9.

[17] Yan, X., Zhang, Ch., Zhang, Sh. (2009). Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support, processing of Expert Systems with Applications, 36 (2) 3066-3076 .

[18] Qodmanan, H., Nasiri, M., Minaei-Bidgoli, B. (2010). Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence, *In.*: Proc. of Elsevier Expert Systems with Applications, 38 (1) 288-298.

[19] Nasiri, M., Taghavi, L., Minaei-Bidgoli, B. (2010). Multi-Objective Rule Mining using Simulated Annealing Algorithm, Soft Computing in Journal of Convergence Information Technology (JCDA), Korea, 5 (1) 60-68.

[20] Khademolghorani, F., Baraani, A., Zamanifar, K. (2011). Efficient mining of association rules based on gravitational search algorithm, IJCSI International Journal of Computer Science Issues, Mahebourg , 8 (4) 1-8.

[21] Atashpaz-Gargari, E., Lucas, C. (2007). Imperialist Competitive Algorithm An Algorithm for Optimization Inspired by Imperialistic Competition, IEEE Cogress on Evolutionary Compution, Singapore, p. 4661-4667.

[22] SPSSInc. Clementine 12.0 Algorithms Guide, SPSSInc, http://www.spss.com.

Author biographies

Fariba Khademolghorani obtained her BS degree in Computer Engineering in 2007 and her MS degree in 2011 at the Faculty of Computer Engineering of Islamic Azad University of Najafabad in Iran. Her research interests include: swarm Intelligence, data mining, soft computing and evolutionary algorithms. She published over eight papers in data mining in national journals and national conferences. Also, she proposed two papers in international journal and international conference. She continues her researches in the field of association rules mining from databases with different types of attributes by using evolutionary algorithms.