International Journal of Academic Research in Computer Engineering
Print ISSN: 2476-7638 and Online ISSN: 2538-2411
Vol. 2, No. 1, Pages. 34-39, May 2018

International Journal of Academic Research in Computer Engineering

Volume 2, Number 1, Pages 34–39, May 2018
www.ijarce.org

IJARCE

Finding the Best Threshold in Association Rule Mining Using the Frog Leaping Algorithm

Ayub Hamidi [1], Madeh Shokri[2*], and Aso Bozorgpanah [3]

1. Institute of Applied Science and Technology, Jahad Daneshgahi of Kurdistan, Kurdistan, Iran.
2. Institute of Applied Science and Technology, Jahad Daneshgahi of Kurdistan, Kurdistan, Iran.
3. Department of Computer and IT Engineering, University of Kurdistan, Kurdistan, Iran.

Receive Date 2017.05.11; Accepted Date: 2017.12.10, Published Date: 2018.05.15

*Corresponding Author: Madeh Shokri (madeh.shokri@gmail.com)

Abstract
In exploring association rules, most of the previous studies have been conducted on the optimization of the efficiency, however, determination of the backup, and safety threshold, has a great effect on the quality of the association rules, and is still of importance. In the traditional algorithms for association rule mining, the two parameters of the confidence, are always determined by the deciding user, using trial and error. Despite the high impact of this task on the performance of the algorithms for association rule mining, these algorithms do not have a good performance in Big Data. Therefore, designing a method to automatically find the best value for these parameters, in Big Data, is a necessity. In this article, a new method for enhancing the calculative efficiency in Big Data to determine the best threshold is suggested. In this method, using the Shuffle Frog Leaping Algorithm (SFLA), first the best fitness for each frog is determined, then the minimum support and confidence are specified. The results show that the SFLA could yield better threshold than genetic and a-priori algorithms. The superiority of the suggested method is that it has the capability of functioning in Big Data, and it is more probable to escape the local optimum trap than the genetic algorithm; moreover, its convergence is quick and its search accuracy is high.

Keywords: Shuffle Frog Leaping Algorithm, Big Data, Association Rule Mining

1. Introduction

With the increasing development of technology, an extensive amount of data has been created, and consequently, different databases have been made to store high volume information. Therefore, in using data mining algorithms in order to access hidden information in the data is increasing every day [1-3]. Using this technique in the recent decade has provided the users with a lot of tools so that they could extract meaningful knowledge and useful patterns form the database[4].

The information acquired from the large amount of data is very useful in the 21st century. Of course, it they should be accurate, legible, and understandable. Regarding the high usages of data mining, one of the problem is that its process requires manual work for completion [4, 5].

Relationships of association rules show mutual dependence between a large set of data items [4, 6-8]. Association rule mining is one of the most important data mining algorithms for finding the hidden rules and properties which play a significant role in the decision-makings [4, 9].

In association rule mining, two parameters if minimum support and confidence has always been determined by the decision-making user, using trial and error [7, 8] who largely affects the algorithm efficiency. However, in a Big Data, it creates millions of rules most of which will be

useless, and therefore, the traditional method of using a-priori algorithm will have less efficiency. Although a-priori algorithm is one of the most famous algorithms in association rule mining and optimized algorithms have been used to improve the efficiency of a-priori algorithm [5,6], however, it has an unignorably weakness: that the parameters are calculated subjectively.

Therefore, the required method needs to be able to automatically find the best value for the two parameters of support and confidence in Big Data. The main goal of this article is to present an optimized method to find the threshold value, the minimum support and confidence in Big Data. Hence, we used SFLA and conducted some tests using Matlab in order to achieve the goal of the study. Results show that the SFLA could determine these parameters quickly and accurately, and also increase the efficiency of data mining in Big Data. On the other hand, the search accuracy of SFLA concerning convergence, in comparison with other algorithms is better and its convergence is more stable.

2. Shuffle Frog Leaping Algorithm

SFLA was first introduced by EUSUFF and LANSEY in 2003[7]. It is one of the metaheuristic optimization algorithms which is modeled from the social behavior of frogs, and taxonomically, it is classified among the behavioral or memetic algorithms, which is briefly called frog algorithm or frog leaping algorithm. As we mentioned earlier, this method is inspired from the communities of frogs which are formed based on the way they look for food, which is by transforming the information among themselves in the ponds, they find their way to the food and the frog which is closer to the fold leads the other frogs to the same direction [8, 10].

In recent years, SFLA has had successful applications in optimization problems. Its main usage is in defining the values of nervous system and control system, and wherever the PSO genetic algorithm[9,11] and [1,2,9,10] could be used[3]. SFLA is not just an optimization algorithm; it is also a tool to show human beings' social cognition, and intelligent agents based on social psychology. Some scientists believe that science is optimized by mutual social behaviors, and thinking is not just a private, but a social action. In SFLA there are some creatures, which we call frogs and are scattered in the search space of a function the value of which we intend to optimize. As an optimization algorithm, SFLA provides a population-based search in which each individual changes its location as time passes. This algorithm functions based on the rules of probability, random search, and population. Each frog represents an insoluble method in the question. The location of the i^{th} frog in a d-dimensional space is $x = (xi_1, xi_2, \ldots, xi_d)$; and the amount of food available for these frogs which is the same as their competence, is f(xi). Locations which have more competence, have more food, and the goal of the frogs is to find more food. Therefore, using SFLA the frogs try to find more food. The procedure of implementing the algorithm is such that first, the population is descending sorted by competence; that is, the best frog is placed in the index 1[12, 13]. The frogs are divided into (m) groups which is called memeplex. Each memeplex consists of (n) frogs. In dividing the population into (m) memeplex, the second frog is given to the second memeplex, and the $(m)^{th}$ frog to the $(m)^{th}$ memeplex. Then. The $(m+1)^{th}$ frog is given to the 1^{st} memeplex, and thus continues the process of sorting so that, there are (n) frogs in each memeplex. Now we carry out the memetic evaluation in each memeplex:

Memetic evaluation is a kind of local search; Based on triangular distribution, (q) frogs ($q < n$) from each memeplex is put into a sub-memeplex, so that each group will have a subgroup: $Pj=2(n+1-j)/n(n+1)$, $j=1,2, \ldots, n$, (triangular distribution formula)

- ✓ According to this distribution, the frogs with highest competence in each memeplex has better chance for selection.
- ✓ After creating the memeplex, their worst frog is replaced by the best frog from the total population.

In each memeplex, the location of the best and worst frog is found according to competence which respectively is Xb and Xw. Then, using equation [1-3], a candidate location is obtained [12-14]:

$$D_i = Rand. (X_b - X_w) \quad (1)$$

$$New\ X_w = Old\ X_w + D_i \ , D_{min} <= D_{max} \quad (2)$$

Rand is a random number generating function with uniform distribution in the interval of [1,0] and is equal to the length of displacement, the amount of which in each dimension should be within the interval of $[D_{min} , D_{max}]$ [15,16].

After obtaining the location, the value of its competence is measured and if it is better than the previous location, X_w moves to its new location; otherwise, (1) and (2) are repeated, with the only difference that instead of X_b , the location of the best frog among all the memeplex, that is X_g is used.

If after this stage competence value of the obtained location from equation (2) is better than the previous location of X_w , this frog is initialized, that is, its new location is determined randomly. Then, the frogs from sub-memeplex are placed in memeplex. Now we combine the memeplex [17].
We put all the frogs in the groups in the X array and sort them in descending order. If the stop condition is met, the algorithm stops, otherwise we sort the frogs once again. We should note that, the stop condition could be a number (N) from the number of all the repetitions in which the frog with best memetic pattern is not optimized anymore[17].

3. Suggested Method

Our suggested method consists of two parts: processing and mining. In the first part the collecting and preprocessing are explained and in the second part, the algorithm implementation for association rule mining is discussed. The procedure of implementation is shown in figure (1) as flowchart. In the first step, the intended database for the algorithm is prepared. In fact, it is in this step which we perform the preprocessing on the database. In the second step, we perform the binary transformation on the database so that the data could be converted and prepared for frog leaping algorithm implementation. The third step is to create primary population, and to divide the frogs into groups and to define the best general and local frog. In the fourth step, the frog leaping algorithm is implemented in order to find the threshold value of the minimum support and confidence in the obtained population from the previous step, so that we could find the optimized value by finding the best frog. In the fifth step, using the suggested algorithm, the best threshold value is presented and the results from the fifth step are used by data rule mining algorithms to achieve better rule.

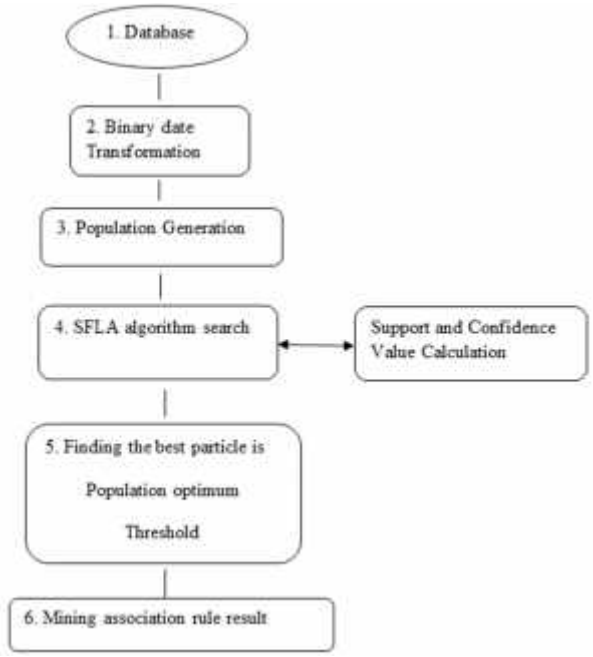

Figure 1. The stages of data mining in the present study

Informative data are among the most valuable assets and life resources. After being processed, data are converted into information which after classification and analysis are converted into knowledge. Access to information when the data are kept in a bad way or when there is not any systematic method to access them, is very difficult, so as they may not even be usable. Therefore, in order to achieve an appropriate information system, the data should be classified and stored logically, so that they could be used easier and faster, analyzed with more efficiency, and therefore, managed better [15,16,18,19].
In this paper, the data from foodmart 2000 database have been used. In this database, we used sales fact 1997 table to evaluate the frog leaping algorithm. The table has 85837 records from the sales of a shop and each record has eight features [20]. We have used a random sample of 3000 records from the sales fact 1997 table. In the suggested algorithm to generate hidden positive and negative rules, we have used SFLA. In this research, our frog represents the positive rule and each rule consists of two sections: the preceding and the succeeding [21]. Figure (2) shows a positive rule and the 10 squares on the left shows the preceding section and the 10 squares on the rights show the succeeding section. Each square represents a field from the database.

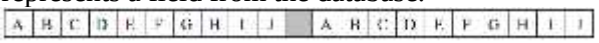

Figure 2. Representation of the movement of a frog

As an example, figure 3 is the representation of the following rule or condition:

If (product _id = 43and store _id = 1) Then (customer _id = 3, store _sales = 3)

Figure 3. Representation of the movement of the frog in a database

In order for the implementation of the suggested method we used Matlab R2010b. We used Matlab in order to show the way the frogs move toward the best target. The frogs are directed in the population toward an optimized answer using the fitness function: in this article, in order to obtain the coefficients of support and confidence, we do the following actions:

In order to generate conditional rules like formula (1), we have used two standards in cost(p) function, to measure the quality of association rules.

if a ⟶ b (1)

$$s = \frac{sup(A \cup B)}{N}$$

$$c = \frac{sup(A \cup B)}{sup(A)}$$

Here, N is the number of all the transactions and sup(A∪B) is the number of the repetition of items A and B in all the transactions. The function of cost(p) is in a manner that first, the support and confidence value for each frog is obtained. After sending the frogs to the fitness function, the frog which has the highest level of fitness is used to lead the other frogs toward the best rule. We define the fitness function in this research as follows:

Fitness = n_1 * support + n_2 * confidence – n_3 * NA

In this relationship, the number of the features participating in the generated rule and the coefficients of a, are used to control each of the parameters' effect in the fitness function, and therefore, they could be adjusted preferably. The first and second part of this function is related to the calculation of the support and confidence of the rule. Considering these two parameters together in the calculation of the generated function, seems essential; since the confidence and support alone cannot be a measure for judging the quality of the generated rule. It is obvious that, a high-quality rule is one in which both of these factors have a high value. On the other hand, we know that in rules with high lengths, the possibility of the existence of extra features which results in the reduction of the quality of the generated answer is high. Therefore, in the third step, we decided to concentrate more on the rules with smaller length and higher legibility, understandability and quality, which are highly significant in data mining. First, the total of F frogs is generated randomly, each of which represents a rule that the frogs are sorted and placed descending in the meme lexes, based on the triangular distribution. Then, using the motioned formulas, the best and worst frogs in the group are found, and the worst frog is optimized by the best frog; then, the groups are sorted descending in an array, and if the stopping condition is met, the algorithm stops, that is, the best frog is not optimized anymore and if the condition is met, the frog leaping algorithm is performed until the end of the condition. This means that the best frogs are found and the support and confidence of the best frogs are shown as the minimum support and minimum confidence. Therefore, we could use these values to discover better and more association rules.

4. The Results of the Stimulation

We implemented and tested this method in several sets of data bases, and obtained interesting results. As we mentioned earlier, in this article, we tested the results on a set of data called foodmart 2000. We used the sales fact table 1997 in foodmart 2000 database to conduct the study. The table has 85387 records, and we randomly used the 3000 records from sales fact table 1997 and supposed the parameters of the SFLA as the table below:

Table 1. Parameters of frog leaping algorithm

F_{max}	Number of memeplexes	N1	N2	N3	Number of Repetitions
200	300	0.8	0.8	0.2	60

We implemented the algorithm with two different values for population which yielded the results from Table 2.

These results shown in Table 2 are much better than the results from [2]. In figure 3 the obtained values for confidence and in figure 4, the obtained values for support in the two algorithms are compared with each other. The red diagram is the suggested algorithm in this study, and the blue diagram is the suggested algorithm in genetics [1-3].

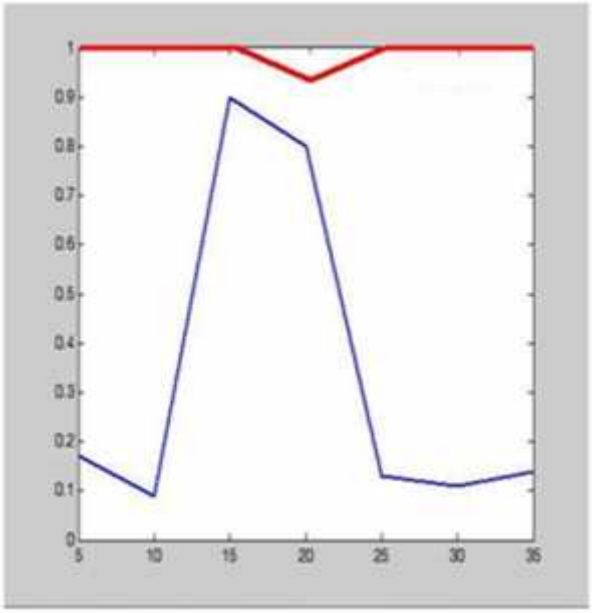

Figure 3. The Diagram for comparing the two algorithms for confidence

Based on the results, the frog leaping algorithm can suggest better and more efficient rules using association rule mining; since the results show that the suggested algorithm could highly increase the efficiency of the association rule mining algorithms in Big Data. Therefore, with the quick and exponential growth of the size of data, the need for such algorithms in Big Data is more felt.

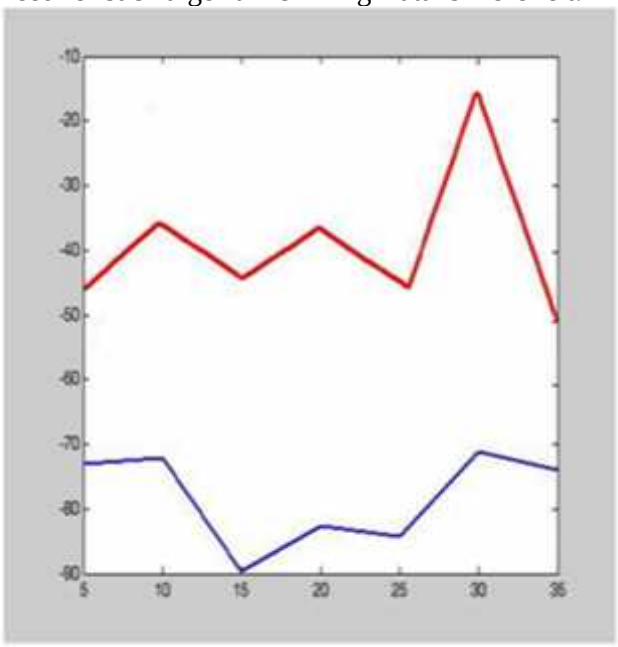

Figure 4. The Diagram for comparing the two algorithms for support

Table 2. The Results from the Proposed Algorithm

The Best Rule	The Implementation Time	The minimum support discovered	The minimum confidence discovered	The Maximum Number of Rules	Sample Size	Population Size	Condition
N4=131 n6=3,7 --->n5=4 cost>0,7012	83	0.0089	1	280	3000	5	1
N8=4→n5=2 cost>0,8805	113	0.0097	1	650	3000	10	2
N3=110 --> n4=3 cost>0,7943	180	0.0091	0.92	935	3000	15	3
N3=9462 --> a6 = cost>0,7456	254	0.0097	1	1330	3000	20	4
N8=2 --> n4=3 cost >0,894	341	0.0089	1	1700	3000	25	5
N5=131 n6=5,1 n8=3 --> n4=3 cost > 0,7932	354	0.186	1	2010	3000	30	6
N8=4 -- > n6 =4 cost>0,8974	403	0.0059	1	2350	3000	35	7

4. Conclusion

The problem of large amount of data which we face in the current century, and the increasing growth of databases, motivated us to suggest an algorithm for association rule mining in Big Data [22]. Today's association rule mining algorithms, like Apriori have one major flaw: the minimum support and confidence was calculated manually by the user and using trial and error; However, using frog leaping algorithm in this article, we solve the problem and specified the assignment of these two parameters accurately and quickly, and finally, we compared the results with the genetic algorithm, and found out that the frog leaping algorithm has obtained good results. First, using the frog leaping algorithm, we could find the

minimum support and confidence, and by applying the results in Apriori algorithm or other association rule mining algorithms, we could increase the efficiency of these algorithms in Big Data.

References

[1] P. Vyas and A. Chauhan, Comparative Optimization of Efficient Association Rule Mining through PSO and GA, in Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on, pp. 258–263, 2013.

[2] K. Sarath and V. Ravi, Association rule mining using binary particle swarm optimization, Eng. Appl. Artif. Intell, Vol. 26, No. 8, pp. 1832–1840, 2013.

[3] H. Barati and M. Sadeghi, An efficient hybrid MPSO-GA algorithm for solving non-smooth/non-convex economic dispatch problem with practical constraints, Ain Shams Eng. J., 2016.

[4] M. Gupta, "Application of weighted particle swarm optimization in association rule mining," Int. J. Comput. Sci. Informatics, Vol. 1, pp. 2231–5292, 2012.

[5] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici, A priori convergence of the greedy algorithm for the parametrized reduced basis method, ESAIM Math. Model. Numer. Anal., Vol. 46, No. 3, pp. 595–603, 2012.

[6] O. Solyalı and H. Sural, A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem, Transp. Sci., Vol. 45, no. 3, pp. 335–345, 2011.

[7] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," J. Water Resource Plan. Management, Vol. 129, No. 3, pp. 210–225, 2003.

[8] D. Mora-Melia, P. L. Iglesias-Rey, F. J. Martinez-Solano, and P. Munoz-Velasco, "The Efficiency of Setting Parameters in a Modified Shuffled Frog Leaping Algorithm Applied to Optimizing Water Distribution Networks," Water, vol. 8, no. 5, p. 182, 2016.

[9] F. F. Razi and V. Shahabi, Forming the stock optimized portfolio using model Grey based on C5 and the Shuffled frog leap algorithm, J. Stat. Manag. Syst., Vol. 19, No. 3, pp. 397–421, 2016.

[10] S. Q. Ali and H. M. Hasanien, Shuffled Frog Leaping Algorithm for Multi-objective Design Optimization of Transverse Flux Linear Motor, Electr. Power Components Syst., Vol. 44, No. 11, pp. 1307–1315, 2016.

[11] P. Roy, P. Roy, and A. Chakrabarti, Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect, Appl. Soft Comput., Vol. 13, No. 11, pp. 4244–4252, 2013.

[12] M. A. Ahandani and H. Alavi-Rad, Opposition-based learning in shuffled frog leaping: An application for parameter identification, Inf. Sci. (Ny), Vol. 291, pp. 19-42, 2015.

[13] M. Eusuff, K. Lansey, and F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Eng. Optim., Vol. 38, No. 2, pp. 129–154, 2006.

[14] J. Zhao, M. Hu, H. Sun, and L. Lv, Shuffled frog leaping algorithm based on enhanced learning, Int. J. Intell. Syst. Technol. Appl., Vol. 15, No. 1, pp. 63–73, 2016.

[15] Z. Zhen, D. Wang, and Y. Liu, Improved shuffled frog leaping algorithm for continuous optimization problem, in Evolutionary Computation, 2009. CEC'09. IEEE Congress on, 2009, pp. 2992-2995.

[16] H. Liu, F. Yi, and H. Yang, Adaptive grouping cloud model shuffled frog leaping algorithm for solving continuous optimization problems, Comput. Intell. Neurosci., Vol. 2016, p. 25, 2016.

[17] J. Ebrahimi, S. H. Hosseinian, and G. B. Gharehpetian, Unit commitment problem solution using shuffled frog leaping algorithm, IEEE Trans. Power Syst., Vol. 26, No. 2, pp. 573–581, 2011.

[18] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng., Vol. 26, No. 1, pp. 97–107, 2014.

[19] S. G. Kalko and G. Stolovitzky, Bioinformatics: Data Mining and Big Data, Handb. Transl. Med., p. 66, 2016.

[20] M. Chen, S. Mao, and Y. Liu, Big data: A survey, Mob. Networks Appl., vol. 19, no. 2, pp. 171–209, 2014.

[21] S. Madden, from databases to big data, IEEE Internet Computing, vol. 16, no. 3, pp. 4–6, 2012.

[22] A. Defazio, J. Domke, and T. S. Caetano, Finito: A faster, permutable incremental gradient method for big data problems, in ICML, 2014, pp. 1125–1133.