Research Journal of Applied Sciences, Engineering and Technology 7(13): 2627-2633, 2014
DOI:10.19026/rjaset.7.578
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 11, 2012 Accepted: July 04, 2013 Published: April 05, 2014

Research Article

An Innovative Potential on Rule Optimization using Fuzzy Artificial Bee Colony

K. Sathesh Kumar and M. Hemalatha
Department of Computer Science, Karpagam University, Coimbatore, India

Abstract: This study adapted an improved algorithm based on Artifical Bee Colony Optimization. It is not possible to justify that all the rules generated by fuzzy based apriori algorithm produce optimum result. Thus optimization of the result generated was carried out by Fuzzy Apriori algorithm using Fuzzy Artifical Bee Colony Optimization (FABCO), it's worth noting that a significant findings were revealed. FABCO is used for optimization of rules to get the best classification accuracy. The proposed method was compared with the traditional Artifical bee colony optimization and the particle swarm optimization. The current work proved a better classification performance compared to un-pruned rules.

Keywords: Fuzzy ABC algorithm, fuzzy apriori, fuzzy association rule, fuzzy datamining, rule optimization

INTRODUCTION

The increase in Real-World optimization problems, in recent years made situations to develop an exact and performing algorithm which is unguaranteed to solve the optimization problems. Many analytical methods do not provide the feasible solution, which can be obtained using optimization algorithm within reasonable time consumption the ability to give optimal solution to every optimization run enables to select the feasible solution to provide the criteria. Most of the real life combinations problems can be optimized in major areas on computer science and mathematics, optimization can be categories into four types combinational optimization, Evolutionary algorithm, Gradient methods and stochastic optimization. Most important population based search engine algorithm used in operational research and computer science is Bee Colony optimization. This kind of algorithm performs enables neighborhood search and random search or it can use both type of search optimization. Swam Intelligence fields very recently follow the directions of introducing Bee colony optimization techniques. The agents are artificial bees which solves complex optimization problems. This behavior of swam characteristics with social insects, communication between the colony insects very certain

Artificial Bee Colony (ABC) algorithm is a kind of swam intelligence used for solving combinational optimization problems, this algorithm proposed by Karaboga in 2005 is based on a particular intelligence behavior of the swam this algorithm is recently introduced optimization algorithm based on foraging behavior of the bee algorithm for solving rule optimization problems, the ABC algorithm mainly contains three groups first one employee bee second is onlookers and scouts first half of the colony consist of the employee bees and second half consist of onlooker bees for every food source there is only one employee bees, in other words the number of employee bee is equal to food source, the employee bee of an abounded food source become a scouts.

Association is one of the data mining techniques among the various data mining methods, rules etc to finds the necessary patterns from the raw set of patterns. Association helps to categorize the patterns with relations. Necessary association rules are mined from available association rules by most association rule mining technique for decision making. Fuzzy association rule mining uses to create a fuzzy logic and it can be used to convert numerical attributes to fuzzy attributes. The popular fuzzy association rule mining algorithm that is available today fuzzy apriori and its different variations it has been well supported for the fuzzy numerical analysis. This proposed study adapted a Fuzzy Apriori algorithm using Fuzzy Artifical Bee Colony Optimization (FABCO). The integration of FABCO improved result significantly produced by Fuzzy Apriori algorithm. FABCO is used for optimization of rules to get the best classification accuracy.

LITERATURE REVIEW

One of the most popular research tasks in data mining is the discovery of frequent itemsets and association rules. Fuzzy association rules is generally easily understandable to humans because of the linguistic terms associated with the fuzzy sets. In addition many researchers have been proposed fuzzy

Corresponding Author: M. Hemalatha, Department of Computer Science, Karpagam University, Coimbatore, India

association rule mining techniques. Fuzzy C-Means algorithm and membership values are generated (Radha and Rajagopalan, 2010) from that quantitative attribute are partitioned into several fuzzy sets which is deal with the data classification problems. Planning based algorithm and cluster based genetic algorithm (Mehmet and Reda, 2006) which dynamically adjusts the fuzzy sets to provide maximum profit under user specified linguistic minimum support and confidence terms. To increase consistency and reliability of the algorithm, results are optimized by Apriori algorithm using an Ant colony optimization algorithm (Badri *et al*., 2011). Integrated fuzzy techniques with data mining algorithm (Au and Chan, 2003) involving a large database generating the rules by fuzzy association rule mining. Proposed an efficient algorithm named Fuzzy Cluster-Based (FCB) along with its (Amir and Reza, 2011) parallel version named Parallel Fuzzy Cluster-Based (PFCB) algorithm. Extracting necessary knowledge through multilevel fuzzy association rule mining to implicit in transactions database (Pratima *et al*., 2010) with different support at each levels. This novel algorithm integrates the fuzzy set concept and the apriori algorithm (Guy *et al*., year) this method for extracting fuzzy association rules between weighted key phrases in collections of text documents. A Rule Based Simplification algorithm and Fuzzy Association Rule mining algorithm for the generate a set of rules can be directly used as a quantitative model of the system (Pach *et al*., 2005). This study contributes towards the discovering fuzzy association rules that based on temporal pattern (Stephen *et al*., 2012). The novel method of the 2-tuple linguistic representation determines fuzzy association rules, at the same time maintaining the interpretability of linguistic terms. Iterative Rule Learning (IRL) with a Genetic Algorithm (GA) simultaneously induces rules and tunes the membership functions.

METHODOLOGY

Association rule mining (a-priori): The Association Rule rules designed is as follows:

IF NFC and RIA THEN BI
IF QC and VLSI THEN IPR

Where NFC, RIA, QC, VLSI, refer to eight subjects out of which student has to choose four. The rules above shows that if a student takes NFC and RIA then the probability is high that he will choose BI too; similarly if he chooses QC and VLSI then the probability is high that he will take IPR. There is no limit on the number of antecedents in the rules, but there is a constraint on the number of consequents and i.e.:

Number of consequents = 1

The limit doesn't make any harm, because if in case the user wants to see the confidence value of a rule that contains the more than one consequents (Liu, 2007) can do the same by taking two rules from our system which allows its intersection.

Apriori algorithm:

General process: Association rule generation is usually split up into two separate steps:

- First, minimum support is applied to find all frequent itemsets in a database
- Second, these frequent itemsets and the minimum confidence constraint are used to form rules

While the second step is straight forward, the first step needs more attention.

Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over *I* and has size 2*n*-1 (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items *n* in *I*, efficient search is possible using the downward-closure property of support (also called anti-monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets (Agrawal and Srikant, 1994).

Apriori Algorithm Pseudocode

Pseudocode Apriori (T, minsupport) {//T is the database and L1 = {frequent items};
For (k = 2; L_{k-1} ! = ∅; k++) {
C_k = candidates generated from L_{k-1}
// that iscartesian product L_{k-1}×L_{k-1} and eliminating any k-1 size itemset that is not frequent
for each transaction t in database do
{
# increament the count of all candidates in C_k that are contained in t
L_k = candidates in C_k with minsupport
}//end for each
}//end for
Return $U_k L_k$;
}

As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation) and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length *k* from item sets of length *k*-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent *k*-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates. Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation creates large numbers of subsets (the algorithm attempts to load as many as possible up, the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all 2|*S*| - 1 of its proper subsets.

Fuzzy association rules: Based on classical association rule mining, a new approach has been developed expanding it by using fuzzy sets. The new fuzzy association rule mining approach emerged out of the necessity to mine quantitative data frequently present in databases efficiently. Algorithms for mining quantitative association rules have already been proposed. When dividing an attribute in the data into sets covering certain ranges of values, we are confronted with the sharp boundary problem:

- Rules F-APACS()
- Begin
- For all d Є D
- For all $\mathcal{L}_{pq}, \mathcal{L}_{jk} \varepsilon\ \mathcal{L}$, P ≠ j do
- deg $\mathcal{L}_{pq}\mathcal{L}_{jk}$ += min$\left(\mu_{L_{pq}}(d[I_p]),,\mu_{L_{jk}}(d[I_j])\right)$
- For all $\mathcal{L}_{pq}, \mathcal{L}_{jk} \varepsilon\ \mathcal{L}, p \neq j$ do
- If interesting $(\mathcal{L}_{pq}, \mathcal{L}_{jk})$then
- $\mathcal{R} = \mathcal{R} \cup$ rulegen $(\mathcal{L}_{pq}, \mathcal{L}_{jk})$
- Return $(\mathcal{R})$
- end

The algorithm starts with a data set. The linguistic terms are represented by fuzzy sets *Lpq*, *Ljk* and the degree to which *d* is represented by is summarized in *deg Lpq Ljk*. The interestingness of an association rule is calculated using the adjusted difference measure (Au and Chan, 1998). The algorithm first searches the database and returns a complete set containing all attributes of the database. In a second step, a transformed fuzzy database is created from the original one. The user has to define the sets to which the items in the original database will be mapped. After generating the candidate itemsets, the transformed database is scanned in order to evaluate the support and after comparing the support to the predefined minimum support, the items with a too low support are deleted. Tzung *et al*. (1999) the frequent itemsets *FK* will be created from the candidate itemesets *CK*. New candidates are being generated from the old ones in a subsequent step. *Ck* is generated from *Ck*-1 as described for the Apriori algorithm. The following pruning step deletes all itemsets of *Ck* if any of its subsets does not appear in *Ck*-1. Finally, the association rules are generated from the discovered frequent itemsets (Delgado, 2003).

Artificial Bee Colony (ABC) algorithm: In ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution (Dervis and Bahriye, 2009). The number of the employed bees or the onlooker bees are equal to the number of solutions in the population.
The main steps of the algorithm are highlighted below:

- Initialize Population
- Repeat
- Place the employed bees on their food sources
- Place the onlooker bees on the food sources depending on their nectar amounts
- Send the scouts to the search area for discovering new food sources
- Memorize the best food source found so far
- Until requirements are met

Totally, ABC algorithm employs four different selection processes:

1. A global probabilistic selection process, in which the Probability value is calculated by (2) used by the on looker bees for discovering promising regions.
2. A local probabilistic Selection process carried out in a region by the employed bees and the onlookers depending on the visual information such As the color, shape and fragrance of the flowers (sources) (bees will not be able to identify the type of nectar source until they Arrive at the right location and discriminate among sources growing there based on their scent) for determining a food source Around the source in the memory as defined (1).
3. A local selection called greedy selection process carried out by Onlooker and employed bees in that if the nectar amount of the candidate source is better than that of the present one, the bee forgets the present one and memorizes the candidate source produced by (7). Otherwise, the bee keeps the present one In the memory.
4. A random selection process carried out by scouts as defined in (8). It is clear from the above explanation that there are three control parameters in the basic ABC: The number of food sources which is equal to the number of employed or

onlooker bees (SN), the value of limit and the Maximum Cycle Number (MCN). In the case of honeybees, the recruitment rate represents a measure of how quickly the bee colony finds and exploits a newly discovered food source. Artificial recruiting could similarly represent the measurement of the speed with which the feasible solutions or better quality solutions of the difficult optimization problems can be discovered. The survival and progress of the bee colony are dependent upon the rapid discovery and efficient utilization of the best food resources. Similarly; the successful solution of difficult engineering problems is connected to the relatively fast discovery of good solutions especially for the problems that need to be solved in real time. In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. Detailed pseudo-code of the ABC algorithm is given below (Radha and Rajagopalan, 2010):

1 : Initialize the population of solutions X_i, I = 1,…., SN
2 : Evaluate the population
3 : Cycle = 1
4 : Repeat
5 : Produce new solutions ti for the employed bees by using the following equation:

$$v_{ij} = X_{ij} + \emptyset_{ij}\ (X_{ij} - X_{kj}) \quad (1)$$

6 : Apply the greedy selection process for the employed bees
7 : Calculate the probability values Pi for the solutions xi by the equation as follows:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (2)$$

8 : Produce the new solutions t_i for the onlookers from the solutions x_i selected depending on P_i and evaluates them
9 : Apply the greedy selection process for the onlookers
10 : Determine the abandoned solution for the scout, if exists and replace it with a new randomly produced solution xi by:

$$X_i^j = X_{min}^j + \text{rand } [0, 1]\ (X_{max}^j - X_{min}^j) \quad (3)$$

11 : Memorize the best solution achieved so far
12 : Cycle = cycle + 1
13 : Until cycle = MCN

Proposed work: The proposed work presents an ABC algorithm for the specified problem to minimize the number of Fuzzy association rules. Fuzzy based Apriori algorithm uses transaction data set which deploys user interested support and confidence value resulting the fuzzy association rule set. These fuzzy association rule sets are discrete and continuous hence; need to prune weak rule sets. Optimization of result is needed, henceforth; Fuzzy Artificial Bee Colony Optimization (FABCO) algorithm for fuzzy association rule was accomplished. Using the fitness value for finding the probability of occurrence of the each rule the greedy selection process will be decided and then optimized by fuzzy association rule set which is generated. In this proposed work, the adoption of FABCO the resultant rules of fuzzy association rules are considered as the population. In order to find the best rules, the fitness value of each rule is identified. Here, the fitness value is calculated based on the roulette wheel. The fitness value is a function of which chromosome is tested for its suitability to the problem in hand. After producing v_i within the boundaries, a fitness value for a minimization problem can be calculated to the solution v_i by:

$$Fitness_i = \left. \begin{matrix} 1/(1+f_i)\ if\ f_i \geq 0 \\ 1+abs(f_i)\ if\ f_i < 0 \end{matrix} \right\}$$

Proposed ABCO Algorithm:

- Initialize the population of rules
- Evaluate the population
- Cycle = 1
- Repeat
- Produce new solutions for the employed bees by using the following equation:

$$v_{ij} = X_{ij} + \emptyset_{ij}(X_{ij} - X_{kj}) \quad (4)$$

- Apply greedy selection process for the employed bees
- Calculate the probability values $P_{i,j}$ for the rules $x_{i,j}$ using fitness of the solution:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (5)$$

- For each onlooker bee, produce a new solution v_i by equation:

$$v_{ij} = X_{ij} + \emptyset_{ij}\ (X_{ij} - X_{kj}) \quad (6)$$

- In the neighborhood of the solution selected depending on p_i and evaluate it
- Apply selection process between v_i and x_i based on greedy method for onlooker bee
- If Scout Production Period (SPP) is completed, determine the abandoned solutions by using-limit

parameter for the scout, if it exists, replace it with a new randomly produced solution by:

$$X_i^j = X_{min}^j + \text{rand } [0, 1] \, (X_{max}^j - X_{min}^j) \quad (7)$$

- Memorize the best solution achieved so far
- Cycle = cycle + 1
- Until cycle = MCN

EXPERIMENTAL RESULTS

For measuring the performance of the proposed FABCO which is a bio-inspiration methodology, we have performed experiments with different datasets that are selected from the UCI Machine Learning Repository. There are nearly 187 data sets currently maintained by the UCI Machine Learning research group. We have tested on the three popular data sets in the experiment they are Voting, Iris and Wine shown the Table 1. To show the performance of FABCO classification algorithm, we compared the performance of our proposed method with three different existing classification techniques KNN, Particle swarm optimization and Artificial bee colony optimization.

Parameter setting of FABCO: In FABCO approach, we have used the number of fold to calculate the cross validation is ten. During the rule discovery period, there are three basic control parameters: the number of food sources which is equal to the number of employed or onlooker bees (SN), the value of limit and the maximum cycle number. Furthermore, they are the most important parameters in this algorithm, since these three parameters will determine the time consuming of finding the rule set and the quality of rule set. Therefore, suitable values are needed to balance these two factors. The values of limit are automatically assigned to the maximum and the minimum value for each feature. With several times attempt, SN is set to 20 and the maximum cycle number set to 2000. After the final rule set has generated, two parameters are required by the class prediction, which are quality weight (α) and coverage weight (β). Both of them were set to 0.5 in this experiment.

Table 2 shows compare the efficiency and the accuracy of the FABCO, it was compared with the ABC and the PSO. The dataset used in this rule optimization is a voting dataset. To estimate the reduction process, we calculated the average supports of acquired rules on testing data set. Also we tested the primitive rules on these records. The Table 3 and Fig. 1 shows the results. As presented in table the accuracy of fuzzy association rules increased 8 after 81% reduction of number of rules using FABCO. This shows a good result for reduction of rules by this approach.

Table 1: Description of three UCI benchmark dataset

Dataset	Features	Instances	Class
Voting	16	435	2
Wine	13	178	3
Iris	4	150	3

Table 2: Performance accuracy of classification in %

	KNN	PCO	ABC	FABCO
Voting	0.95209	0.96532	0.97091	0.97865
Iris	0.94280	0.95710	0.96820	0.98410
Wine	0.96590	0.97460	0.98070	0.98960

Table 3: Accuracy percentage of proposed technique

Techniques	Primitive rules	Reduced rules
PSO	0.31	0.296
ABC	0.29	0.283
FABCO	0.23	0.207

Table 4: Rule pruning accuracy percentage of proposed technique

	Accuracy	
Techniques	Before pruning	After pruning
PSO	86	90
ABC	90	92
FABCO	93	95

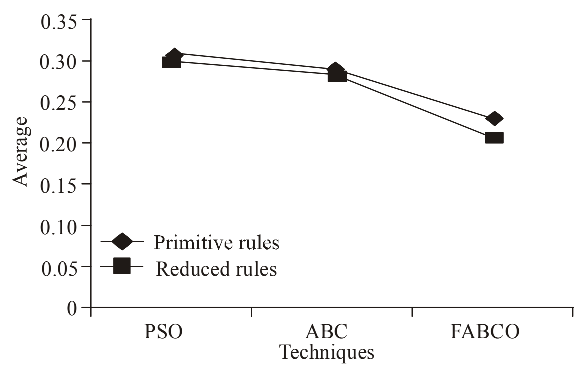

Fig. 1: Accuracy percentage of proposed technique

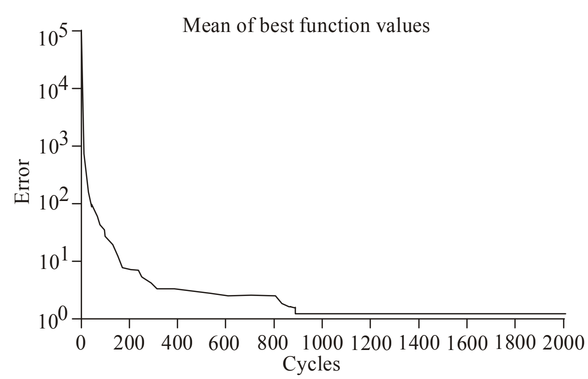

Fig. 2: Using the function rosenbrock

The Table 4 shows the accuracy percentage of proposed technique with the other two existing approaches namely Particle Swarm Optimization (PSO) and Artificial Bee Colony optimization (ABC). The result shows that the proposed study has performed a significant improvement in enhancing the capability of the classification of dataset by pruning the rules generated by fuzzy association rule mining.

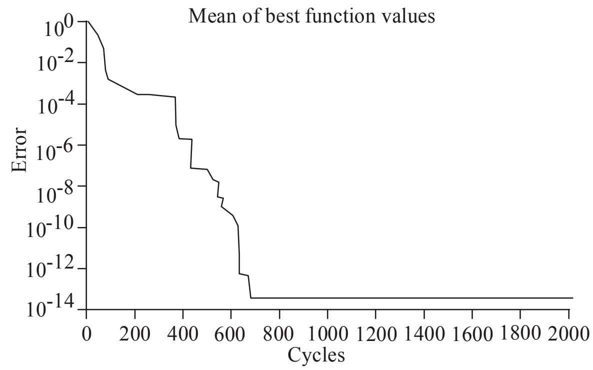

Fig. 3: Using the function griewank

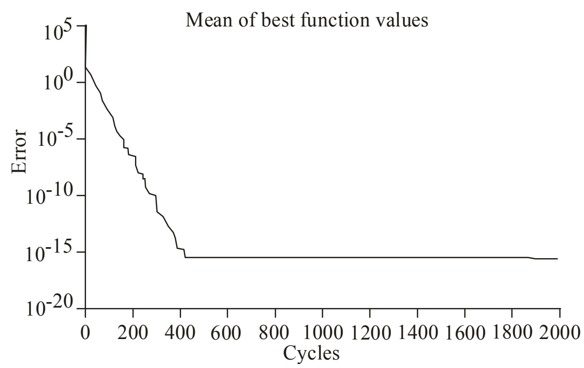

Fig. 4: Using the function rastrigin

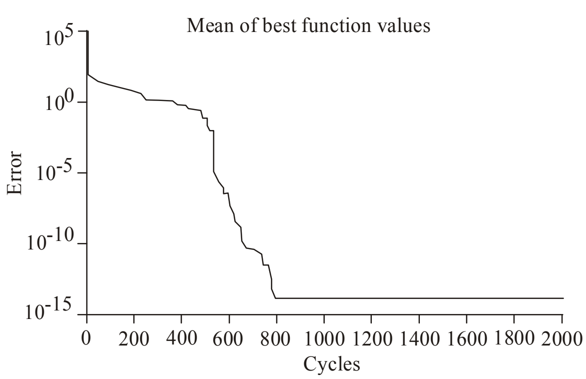

Fig. 5: Using the function sphere

The best Mean value of FABCO with four different function is displayed in the Fig. 2 using the function rosenbrock the proposed approach holds the mean value of 1.22568 and standard deviation is 1.68234 with the iteration of 2000 cycles.

Figure 3 shows the function griewank was implemented with the proposed approach which holds the mean value is 4.18924e-014 and standard deviation is 7.13125e-014 with the iteration of 2000 cycles and Fig. 4 the function rastrigin was implemented with proposed approach which holds the mean value is 9.4739e-015 and standard deviation is 1.64093e-014 with the iteration of 2000 cycles.

Figure 5 the function sphere was applied with the proposed approach and its mean value is 1.48018e-016 and standard deviation is 8.63957e-017 with the iteration of 2000 cycles.

CONCLUSION

The great number of rules extracted from data mining algorithms, especially fuzzy association rule mining, considerably minimizes the effectiveness of the rules and turns it difficult for the user to use and decide on these rules. This is attributed to the generation of more number of rules. All the generated rules do not produce best classification results and also it takes more number of times. To overcome this problem we proposed fuzzy based artificial bee colony optimization algorithm which picks the best rules from the given population of rules. The proposed approach was compared with the particle swarm optimization and the artificial bee colony optimization. Based on the proportion of rules, reduction and accuracy of techniques, the proposed method outperforms existing ones.

REFERENCES

Agrawal, R. and R. Srikant, 1994. Fast Algorithms for Mining Association Rules in large databases. Proceeding of 20th International Conference on Very Large Data Bases.

Amir, E. and S. Reza, 2011. Two efficient algorithms for mining fuzzy association rules. Int. J. Mach. Learn. Comput., 1(5): 510-515.

Au, W.H. and K.C.C. Chan, 1998. An effective algorithm for discovering fuzzy rules in relational databases. Proceedings. IEEE World Congress on Computational Intelligence, IEEE International Conference on Fuzzy Systems, pp: 1314-1319.

Au, W.H. and K.C.C. Chan, 2003. Mining fuzzy association rules in a bank-account database. IEEE T. Fuzzy Syst., 11(2): 238-248.

Badri, P., K.C. Vijay, K.K. Rajneesh and Y.K. Rana, 2011. Optimization of association rule mining apriori algorithm using ACO. Int. J. Soft Comput. Eng., 1(1): 24-26.

Delgado, M., 2003. Fuzzy association rules: An overview. Proceeding of BISC Conference.

Dervis, K. and A. Bahriye, 2009. A comparative study of artificial bee colony algorithm. Appl. Math. Comput., 214: 108-132.

Guy, D., S. Moti, L. Mark, L. Marina and K. Abraham, year. An Apriori-like algorithm for Extracting Fuzzy Asso citation Rules between Key phrases inText Documents.

Liu, B., 2007. Web Data Mining: Exploring Hyperlinks. Contents and Usage Data. Springer, Heidelberg.

Mehmet, K. and A. Reda, 2006. Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining. Appl. Intell., 24: 7-15.

Pach, F.P., F. Szeifert, S. Nemeth, P. Arva and J. Abonyi, 2005. Fuzzy association rule mining for data driven analysis of dynamical system. Hungarian J. Ind. Chem. Veszp., 33(1-2): 57-67.

Pratima, G., K. Neelu and K.R. Pardasani, 2010. A model for mining multilevel fuzzy association rule in database. J. Comput., 2(1): 58-64.

Radha, R. and S.P. Rajagopalan, 2010. Generating membership values and fuzzy association rules from numerical data. Int. J. Comput. Sci. Eng., 2(8): 2705-2715.

Stephen, G.M., M.A. Gongora, A.A. Hopgood and S. Ahmadi, 2012. Temporal fuzzy association rule mining with 2-tuple linguistic representation. Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2012). Brisbane, pp: 1-8.

Tzung, P.H., C.S. Kuo and S.C. Chi, 1999. A fuzzy data mining algorithm for quantitative values. Proceedings of the 3rd International Conference Knowledge-Based Intelligent Information Engineering Systems, pp: 480-483.