Fast Mining of Fuzzy Association Rules

Amir Ebrahimzadeh and Reza Sheibani, *Member, IACSIT*

***Abstract*—Fuzzy association rules described by the natural language are well suited for the thinking of human subject and will help to increase the flexibility for supporting user in making decisions or designing the fuzzy systems. However, the efficiency of algorithms needs to be improved to handle real-world large datasets. In this paper, we present an efficient algorithm named fuzzy cluster-based (FCB) along with its parallel version named parallel fuzzy cluster-based (PFCB). The FCB method is to create cluster tables by scanning the database once, and then clustering the transaction records to the i-th cluster table, where the length of a record is i. moreover, the fuzzy large itemsets are generated by contrasts with the partial cluster tables. Similarly, the PFCB method is to create cluster tables by scanning the database once, and then clustering the transaction records to the i-th cluster table, which is on the i-th processor, where the length of a record is i. moreover, the large itemsets are generated by contrasts with the partial cluster tables. Then, to calculate the fuzzy support of the candidate itemsets at each level, each processor calculates the support of the candidate itemsets in its own cluster and forwards the result to the coordinator. The final fuzzy support of the candidate itemsets is then calculated from these results in the coordinator. We have performed extensive experiments and compared the performance of our algorithms with two of the best existing algorithms.**

***Index Terms*—Fuzzy association rules, cluster table, parallel.**

I. Introduction

Highlight Relational database have been widely used in data processing and support of business operation, and there the size has grown rapidly. For the activities of decision making and market prediction, knowledge discovery from a database is very important for providing necessary information to a business. Association rules are one of the ways of representing knowledge, having been applied to analyze market baskets to help managers realize which items are likely to be bought at the same time [1]. For example, rule {P}→{Q} represent that if a customer bought P, then he should buy Q at the same time .Formally, the problem is stated as follows:

Let I={ $i_1, i_2, \ldots, i_m$} be a set of literals ,called items , D be a set of transaction , where each transaction T is a set of items such that $T \subseteq I$. A unique identifier TID is given to each transaction. A transaction T is said to contain A, a set of item in I, if $A \subseteq T$. An association rule is an implication of the form "A→ B", where $A \subseteq I$, $B \subseteq$ I, and $A \cap B = \emptyset$. Usually, an association rule $A \rightarrow B$ can be obtained if its degree of support and confidence is greater than or equal to the pre-specified threshold respectively, i.e

Dsupp($A\rightarrow B$)=$|AB|/|D| \geq$ Min_supp, and Dcon$f$($A\rightarrow B$)=$|AB|/|A| \geq$Min_conf,

where $|A|$ is the number of transaction that contain A, and $|D|$ is the total number of transaction in database D.

Initially, Agrawal et al. [2] proposed a method to find the large itemsets. Subsequently, Agrawal et al. [3] also proposed the Apriori algorithm.

In recent year, there have been many attempts to improve the classical approach [3],[4].Since real world application usually consist of quantitative values, mining quantitative association rules have been carried out by partitioning attribute domains and the transforming the quantitative values into binary ones to apply the classical mining algorithm [5] . However, using the classical approach for partitioned intervals may lead to the problem of sharp boundaries for interval [6] .

In dealing with the "sharp boundary problem "in partitioning, fuzzy sets, which can deal with the boundary problem naturally, have been used in the association rule mining domains[7]-[12].

However, these algorithms must scan a database many times to find the fuzzy large itemsets. Therefore as the database size becomes larger and larger, a better way is to mine association rules in parallel. A parallel algorithm for mining fuzzy association rules have been proposed in[13].

A fuzzy association rule understood as a rule of the form $A\rightarrow B$ where A and B are now fuzzy subsets rather than crisp subsets. The standard approach to evaluate the significance of fuzzy association rules is to extend the definition of well-known support and confidence measure to fuzzy association rule:

Dsupp($A\rightarrow B$)= ($\sum A(x) \otimes B(y)$) / $|D|$,

Dconf ($A\rightarrow B$)=($\sum A(x) \otimes B(y)$) / $\sum A(x)$,

where A(x) and B(y) denotes the degree of membership of the element x and y with respect of the fuzzy sets A and B respectively, $\otimes$ is a t-norm [14]. Large fuzzy itemset and effective fuzzy association rules can be determined by the proposed fuzzy support and the fuzzy confidence, respectively .In this paper, an effective algorithm named fuzzy cluster based (FCB) algorithm along with it's parallel version is proposed.

These mining algorithms consist of three parts:

1) Quantitative attributes are partitioned into several fuzzy sets by the fuzzy c-means (FCM) algorithm[15];
2) Discovering frequent fuzzy attributes;
3) Generating fuzzy association rules with at least a minimum confidence from frequent fuzzy attributes. In this paper we firstly describe the sequential algorithm (FCB), secondly we propose it's parallel version, then experiment result are given to show the performance of the proposed algorithms. last is conclusion.

Manuscript received June 25, 2012; revised August 10, 2012.

A. Ebrahimzadeh is with the Sama technical and vocational training college, Islamic Azad University, Mashhad branch, Mashhad, Iran (e-mail: ebrahimzadehamir@gmail.com).

R. Sheibani is with the Department of Computer, Mashhad Branch, Islamic Azad University, Mashhad, Iran (e-mail: reza.shni@gmail.com)

II. PARTITIONING FUZZY SET

Fuzzy set was proposed by Zadeh , and the division of the features into various linguistic values has been widely used in pattern recognition and fuzzy inference. From this, various results have been proposed, such as application to pattern classification by Ishibuchi et al [16], the fuzzy rules generated by wang and Mendel [17] , and methods for partitioning feature space were also discussed by many researchers. In this paper, we view each attribute as a linguistic variable, and the variable are divided into various linguistic values. A linguistic variable is a variable whose values are linguistic words or sentences in a natural language. For example, the values of the linguistic variable 'Age' may be 'close to 30' or ' very close to 50' and referred to as linguistic values. In FCB algorithm, quantitative attributes are partitioned into several fuzzy sets by the FCM algorithm[15].

III. FCB ALGORITHM

The performance is dramatically decreased in the process of many fuzzy association rules algorithms. This is due to the fact that a database is repeatedly scanned to contract each candidate itemset with the whole database level by level in process of mining fuzzy association rules. Thus, we propose an efficient method for discovering the fuzzy large itemsets. for better understanding, we first describe the sequential version of the algorithm with an example and then move on to the parallel version.

A. Fuzzy Cluster-Based Algorithm (Sequential Implementation)

After quantitative attributes are partitioned into several fuzzy sets by FCM algorithm, The sequential algorithm employs some efficient cluster tables to represent database D by a single scan of the database, following by contrasts with the partial cluster tables.

Fig. 1 is the algorithmic form of sequential algorithm , which, for ease of presentation, is divided into three parts . Part 1 gets a set of large 1-itemsets and creates M cluster tables ,scan the database once and cluster the transaction data. If the length of transaction record is K , transaction record will be stored in the table, named cluster_tabel(k), $1\leq k \leq M$, where M is the length of the longest transaction record in database. Meanwhile, the set of large 1-itemsets, L_1,is generated.

Part 2 generates the set of fuzzy candidate k_itemsets C_k . the procedure is similar to the candidate generation of Apriori algorithm [3] .

Part 3 determines the set of fuzzy large k-itemsets L_k, as shown in Fig. 3. when the length of candidate itemset is k , the support is calculated with reference to the cluster_tabel(k) . Then it is contacted with the Cluster_Table (k+1),(k+2),… .

B. An Example of FCB Algorithm

We provide an example to to explain the application of our algorithm. There are 20 records in the database. An example is shown in table I. Each transaction in table I consist of pair (x,t) such that x is an item and t is the number of item x in transaction. Part 1 gets a set of large 1-itemset and create four cluster table are shown in table II: (a), (b), (c)and (d). Then to find the fuzzy support of each fuzzy candidate 2-itemset algorithm start from cluster–table (2), And calculate the fuzzy support of candidate itemset in this cluster-table. Next the same does in cluster–table (3) and cluster–table (4). Finally, the fuzzy support of candidate itemset is the sum of Fuzzy support in cluster–table (2), cluster–table (3) and cluster–table(4).

- Algorithms Table_based_Clustering_pruning (D, Minsup)
- Input: D, Minsup
- Output: Answer(Answer = U L_k , for $1\leq k\leq M$)
- Begin
- 1) cluster_Table_Create(D,Minsup);
- 2) for (k=2; $L_{k-1} \neq ø$; k++) do{
- 3) C_k= Candidate_itemset_Gen(L_{k-1});
- 4) L_k=Large_itemset_Gen(C_k);
- 5) }
- 6) Answer=U L_k ;
- End

Fig. 1. Main program for the fcb algorithm.

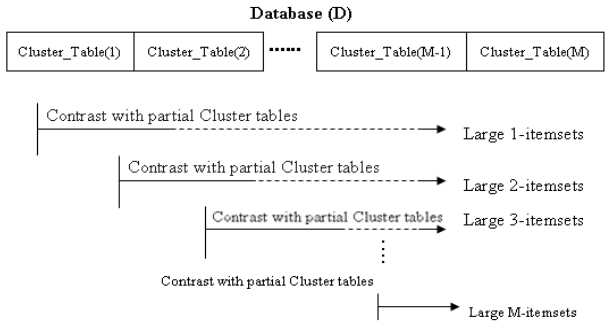


Fig. 2. Fcb algorithm need the constrants with only partial cluster tables.

- Procedure Large_Itemset_Gen(C_k)
- Input: C_k
- Output: L_k
- Begin
- 1)While($C_k \neq ø$)do{
- 2) pick c from C_k;
- 3) support(c)=0;
- 4) for (i=k; i ≤ max_length; i++) do{
  5) temp = the fuzzy support of c appearance in the Cluster_Table(i);
  6) support(c) = support(c) + temp ;
  /*compute support of fuzzy itemset c*/
  7) }
  8) support (c) = support (c) / | D| ;
  9) if (support (c) ≥ Minsup) then{
- 10) put c into L_k;
- 11) }
- 12) }End.

Fig. 3. Procedure of fuzzy large k-itemsets Generation for fcb

Similary ,to find the fuzzy support of each fuzzy candidate 3-itemset , the support of candidate itemset in cluster–table (3) and cluster–table (4) claculate . Then the support of candidate itemset is sum of its support in this claster-tables.

TABLE I: AN EXAMPLE OF TRANSACTION DATABASE

100	**(1,1)** **(2,7)** **(3,3)**	*600*	*(1,2)* *(3,1)* *(5,2)*	*1100*	*(1,5)* *(2,1)* *(3,1)*	*1600*	*(1,1)* *(4,4)* *(5,3)*
200	(2,2)	700	(3,3)	1200	(3,3) (5,2)	1700	(2,4) (3,3)

300	(1,4) (5,5)	800	(2,3) (3,2) (5,1)	1300	(2,3) (3,1) (4,3) (5,1)	1800	(5,1)
400	(1,5) (3,2) (4,1) (5,2)	900	(1,1) (2,2) (3,4) (4,2)	1400	(3,5) (4,1)	1900	(1,1) (3,1) (4,1)
500	(1,2) (3,1)	1000	(4,1)	1500	(2,4) (3,2) (4,1)	**2000**	**(1,3) (2,3) (3,3) (4,1)**

TABLE II: FOUR CLUSTER TABLES A CLUSTER_ TABLE (1)

TID	*1*	*2*	*3*	*4*	*5*
200	0	2	0	0	0
700	0	0	3	0	0
1000	0	0	0	1	0
1800	0	0	0	0	**1**

b)Cluster_ table(2)

TID	*1*	*2*	*3*	*4*	*5*
300	4	0	0	0	5
500	2	0	1	0	0
1200	0	0	3	0	2
1400	0	0	5	1	0
1700	0	4	3	0	**0**

c)Cluster_ table(3)

TID	*1*	*2*	*3*	*4*	*5*
100	1	7	3	0	0
600	2	0	1	0	2
800	0	3	2	0	1
1100	5	1	1	0	0
1500	0	4	2	1	0
1600	1	0	0	4	3
1900	1	0	1	1	**1**

d)Cluster_ table(4)

TID	*1*	*2*	*3*	*4*	*5*
400	5	0	2	1	2
900	1	2	4	2	0
1300	0	3	1	3	1
2000	3	3	3	1	**0**

IV. PFCB ALGORITHM

A. Partition the Quantitative Attributes

Quantitative attributes are partitioned into several fuzzy sets by the FCM algorithm in sequential algorithm.

As the database size becomes larger and larger, FCM algorithm requires lots of computation power, main memory and disk I/O. Lamehamedi H. presents the parallel fuzzy *c*-means (PFCM) algorithm [18].The PFCM algorithm is developed following a master/slave approach. The computation is iterative and consists of *s* slaves controlled by the master. In order to implement the parallel algorithm for mining fuzzy association rules on the distributed linked

PC/workstation, we improve the master/slave approach to the single program/multi data approach. Indeed, the PFCM algorithm is shown in Fig. 4.

PFCM1. The values taken by each attribute are regarded as the initial set of patterns. Partition the initial set of patterns among the processes. Each process will get *n/s* patterns. Where *n* is the number of patterns and *s* is the number of the processes launched.

$$\{\underbrace{x_1, x_2, \quad , x_{n/s}}_{\text{Process 1}} \mid \underbrace{x_{(n/s)+1}, \quad , x_{2n/s}}_{\text{Process 2}} \mid \quad \mid \underbrace{x_{(s-1)(n/s)+1}, \quad , x_n}_{\text{Process s}}\}$$

PFCM2. Initialize the $v_i, i = 1,2, \quad ,c$ on the root process, and broadcast them to all processes.

PFCM3. Each process receives $v_i, i = 1,2, \quad ,c$, and computes the membership values of the patterns it holds. Each process *j* operates separately on its subset of data $\{x_k, k = (j-1)n/s+1, \quad , jn/s\}$, this step is the end of the initialization part.

$$u_{ik,0} = [\sum_{j=1}^{c} (\| x_k - v_{i,0} \|_A / \| x_k - v_{j,0} \|_A)^{2(m-1)}]^{-1} \ \forall i,k$$

For *t*=1 to *T*

PFCM4. Each process computes:

$$\alpha_{i,j} = \sum_{k=1}^{size\,x} (u_{ik,t-1})^m x_k$$

$$\beta_{i,j} = \sum_{k=1}^{size\,x} (u_{ik,t-1})^m$$

where $sizex = n/s$ is the number of patterns receive by each process.

PFCM5. Each process *j* sends these results ($\alpha_{i,j}$ and $\beta_{i,j}$) to the root process, which then aggregates to compute $v_{i,t}$ and broadcasts to all processes.

$$v_{i,t} = (\sum_{j=1}^{s} \alpha_{i,j}) / (\sum_{j=1}^{s} \beta_{i,j}) \quad i = 1,2, \quad ,c$$

PFCM6. Each process receives the value of the cluster centers and computes the membership values of the patterns. Each process operates separately on its subset of data.

$$u_{ik,t} = [\sum_{j=1}^{sizex} (\| x_k - v_{i,t} \|_A / \| x_k - v_{j,t} \|_A)^{2(m-1)}]^{-1} \ \forall i,k$$

PFCM7. Each process computes the error. The portion of error at each process *j* is computed and then sent to the root process.

$$error_j = \sum_{k=1}^{size\,x} \sum_{i=1}^{c} (u_{ik,t-1} - u_{ik,t})^2 \quad for\ j = 1,2, \quad ,s.$$

PFCM8. The errors are aggregated at the root process:

$$E_t = (\sum_{j=1}^{s} error_j)^{1/2}$$

if $E_t < \varepsilon$, then stop.

Fig. 4. Man program of PFCM algorithm.

B. PFCB Algorithm

In the parallel version (PFCB algorithm), after quantitive attributes are partitioned into several fuzzy set, each cluster is handled by one processor (cluster *i* is handled by rocessor *i*). Additionally, another processor is required as the coordinator. Therefore, we need *M*+1 processor where then M is the maximum length of transaction. The PFCB method creates cluster table by scaning the database once, and then clustering the transaction records to the i-th cluster table, which is on the i-th processor, where the length of a record is *i*. Similar to the sequential version, L_1 is created of this stage.the creating of C_k from L_{k-1} is done similarly to the apriori algorithm and on the coordinator. Then, at each level, to create L_k from C_k the coordinator will send the C_k set to all the processors with numbers equal or greater than *k*. After receiving the C_k, each processor calculates the fuzzy support of each itemset in C_k, in its own cluster, and send the results back to the coordinator. The coordinator after getting back all the result, will compute the fuzzy support of each itemset to create L_k. obviously, at each level *i*, there are *i*-1 idle processors. The Fig. 5 and Fig. 6 show the working of this algorithm in coordinator and other processors.

C. Algorithms Coordinator

(D, Minsup)
Input: D, Minsup
Output: Answer (Answer = U L_k , for $1 \le k \le M$)
Begin
1) scan DB and create cluster i on machine i, Where the length of a record is i (1=<i=<M) that M is the longest of transactions), also compute L_1 .
2) for (k=2; ;k++) do {
2-1- C_k= candidate-itemset-Gen(L_{k-1});
2-2- send C_k to all machine with number equal or greater than k;
2-3- receive results from all machine with number equal or bigger than k;
2-4- calculate the fuzzy support of each itemset by adding the fuzzy supports in each cluster.
2-5- add itemset with support greater than minsup to L_k;
If (L_k=Ø) send halt message to all processor.
Halt;
}
Answer=ỤL$_k$; }

Fig. 5. Man program for the pfcb algorithm.

- Procedure Larg_Itemset_Gen(C_k)
- Input: C_k
- Output: L_k
- Begin
- For (k=2; ;k++) do {
- Recive C_k from coordinator ;
- While ($C_k \neq 0$)do {
- Pick c from C_k;
- Support(c)=0;
- Temp= the fuzzy support of c in cluster of this machine; }
- -send the fuzzy supports in this cluster to coordinator. }

Fig. 6. Procedure of fuzzy large *k*-itemsets generation for pfcb.

V. Experimental Results

To evaluate the efficiency of the FCB method, we have implemented the FCB along with fuzzy Apriori_like algorithm, Using Microsoft visual C# on a Pentium III 600 MHz PC with 256MB of available physical memory.

The test database is real-life database. In this experiment, the efficiency of the FCB algorithm is compared to the Apriori_like algorithm. The number of linguistic value in each attribute is 3.

1) 60000 transaction records of experimental data are sampled randomly from the real-life Database. The test database contains 10 items, in which the longest transaction record contains 7 items.

The performance of FCB algorithm is compared to Apriori –like algorithm under various users specified minimum support (MinSup), such that 0.50%, 0.40%, 0.30%, and 0.20%. The results are shown in Fig. 7. you can show that whenever the minsupport decreases, the gap between algorithms becomes more evident.

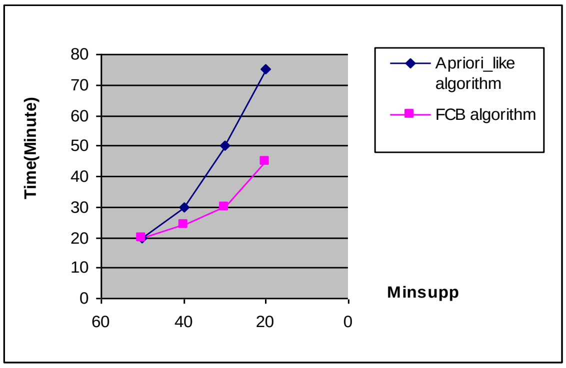
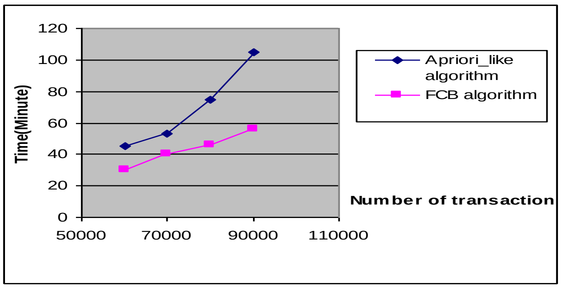


Fig. 7. Performance of fcb and apriori-like algorithm on 60000 records.



. Fig. 8. Performance of fcb and apriori-like algorithm at minsupport 0.30%.

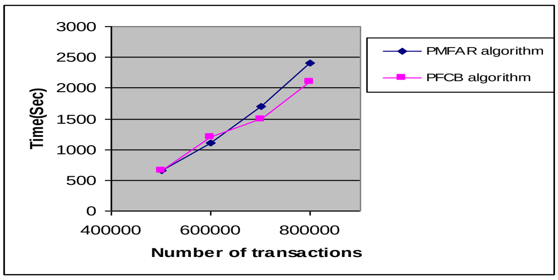


Fig. 9. The performance of pfcb is compared to pmfar algorithm at minsup 0.30%.

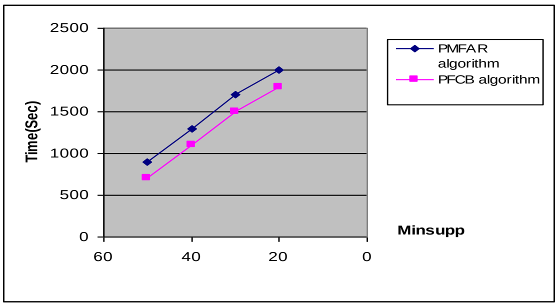


Fig. 10. The performance of pfcb is compared to pmfar on 700000 transactions.

2) 60000, 70000, 80000 and 90000 records of experimental data are sampled randomly from real-life database. The number of attribute is again 10. The performance of FCB algorithm is compared to apriori-like algorithm where minimum support is 0.30% (Fig. 8).

When the number of transaction increases, again the gap between algorithms increases too. We implemented our parallel algorithm for mining fuzzy association rules a long with PMFAR algorithm on the distributed linked PC/workstation. This workstation consists of eight computers with 128,000 KB of real memory, which are interconnected via a 10M/100M hub. We use the parallel message passing software MPICH2.The experiment is implemented on the pervious real life dataset, With 10 items in with the longest transaction record contains 7 items. In the experiment, attributes are partitioned into three fuzzy sets. Let minimum fuzzy support be 0. 30%, let minimum fuzzy confidence be 0.1.

1) 500000,600000,700000,800000 records of experimental are sampled from dataset. the number of linguistic value in each attribute is 3.The performance of PFCB is compared to PMFAR algorithm.(Fig. 9)
2) 700000 records of experimented data are sampled randomly from dataset. The performance of PFCB and PMFAR are compared with various user specified min

sup. (Fig. 10).

Experiments shows that when the number of transactions is increased and minsup is decreased our algorithm outperform PMFAR algorithm.

VI. Conclusion

In this paper we propose an efficient algorithm for mining fuzzy association rules. The FCB algorithm along with its parallel version creates cluster table to aid discovery of fuzzy large itemsets. The characteristics of FCB are the following. It only requires a single scan of the database, following by contrast with the partial cluster tables. In PFCB algorithm, there is a cluster in each machine. For calculating the fuzzy support of each itemset in C_k, the fuzzy support of C_k itemset in each cluster will be estimated and the result of this calculation will be sent to the coordinator. The coordinator after receiving, the consequences from each machine, will be calculate the final support of itemset. Eventually estimate the big itemset in each level. Experiments with real life database show that FCB and PFCB have a good performance.

References

[1] J. W. Han and M. kamber, "Data mining: Concepts and techniques," *Morgan Kaufmann*, San Francisco, 2001.

[2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 1993, pp. 207-216.

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkm-o, "Fast discovery of association rules, in: U.M Fayyad, G. piatetsky-shapiro,p.Smyth,r.Uthurusamy(Eds.), Advances in Knowledge Discovery and Data Mining," *AAAI press*, Menlo park, 1996. pp. 307-328.

[4] H. Mannila, H. Toivonen, and I. Verkamo, "Efficient algorithms for discovering association rules, piatetsky-Shapiro. G. and Frawley,W.J.(eds.) knowledge Discovery in Databese," *AAAI press/The MITpress*, MenIo park, Califomia, 1991.

[5] R. Srikant and R. Agrawal. "Mining Quantitative Association Rules in Large Rolational Tables," *Proc. of 1996 ACM-SIGMOD Internat. Conference Management of Data*. pp.12, Montreal, Canada, 1996.

[6] Gyenesei, "A fuzzy approach for mining quantitative association rules," TUCS technical reports 336, University of Turku, Department of computer Science, Lemminkisenkatu 14, Finland, 2000.

[7] G. Chen, Q. Wei, and E. E. Kerre, "Fuzzy data mining: Discovery of fuzzy generalized association rules.G,. Bordogna and G.pasi,, (eds.)," *Recent Issues on Fuzzy Databases.* Springer-Verlag, 2000.

[8] G. Chen, P. Yan, and E. Kerre, "Mining fuzzy implication–based association rules in quantitative database," *Proceeding of FLINS2002*.

[9] M. Delgado, D. Sanchez, and M. A. Vila, "Acquisition of fuzzy association rules from medical data. S.Barro and R. Marin," (eds.) *Fuzzy Logic in Medicine*. Physical Verlag, 2000.

[10] W.-H. Au and K. C. Chan, "An effective algorithm for discovering fuzzy rules in relation databases," *Proc. IEEE World Congress on Computational Intelligence*, pp. 1314-1319, 1998.

[11] Y.GAO ,J.MA ,L.MA , A new algorithm for mining fuzzy associations rules, proceedings of the third international conference on machine learning and cybernetic ,shanghai ,26-29 August 2004 ,pp.1635-1639.

[12] M.Kaya ,R.Alhajj ,Genetic algorithm based framework for mining fuzzy associations rules, Fuzzy sets and systems 152(2005) 587-601.

[13] B. Xu, J. Lu, Y. Zhang, L. Xu, H. Chen, and H. Yang, "Parallel algorithm for mining fuzzy association rules," In *International Conference on Cyberworlds*, Singapore, 2003.

[14] E. Hiillermeier and J. Beringer, "Mining implication-based fuzzy association rules in database, B. Bouchon-Meunier, L. Foulloy, and R>R.Yager," (eds.) *Intelligent Systems for Information Processing: From representation to Applications*. Elsevier, 2003.

[15] J. J. Lu, Z. L. Song, and Z. P. Qian, "Mininglinguistic valued association rules," *Journal of Software*, 2001, vol. 12, no. 4, pp. 607-611.

[16] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selection fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transaction on Fuzzy Systems* vol. 3, no. 3, pp. 206-270, 1995.

[17] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transaction on Systems*, Man, and Cybernetics vol. 22, no. 6, pp. 1414-1427, 1992.

[18] H. Lamehamedi, A. D. Bensaid, and E.-G. Kebbal, "Adaptive programming: Application to a semisupervised point prototype clustering algorithm," *International Conference on Parallel and Distributed Processing Techniques*, Las Vegas, Nevada, USA, 1999, pp. 2753-2759.

Amir Ebrahimzadeh is a Faculty member of Sama technical and vocational training college, Islamic Azad University, Mashhad branch, Mashhad, Iran. He is born on 12 September in Mashhad. He gained his B.S. Degree in Computer Software engineering, Azad University of Mashhad, 2003. And his M.S. Degree in Computer Software engineering at Islamic Azad University Mashhad Branch, 2005.His major field of study is data mining.

Reza Sheibani is a Faculty Member of Computer Engineering Department of Islamic Azad University, Mashhad Branch, Mashhad, Iran. He is born on 20 April in Mashhad. He gained his B.S. Degree in Computer Software engineering, Ferdowsi University of Mashhad, 1998. And his M.S. Degree in Computer Software engineering at Islamic Azad University South Tehran Branch, 2002.His major field of study is data mining.