An Algorithm For Mining Fuzzy Association Rules

Reza Sheibani , Amir Ebrahimzadeh ,Member, IAUM

***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 huge datasets in real word. in this paper, we present an efficient algorithm named Fuzzy Cluster-Based Association Rules(FCBAR).**

The FCBAR method is to create cluster tables by scanning the database once, and then clustering the transaction records to the k_th cluster table, where the length of a record is k. Moreover, the fuzzy large itemsets are generated by contrasts with the partial cluster tables. This prunes considerable amount of data, reduces the time needed to perform data scans and requires less contrast. Experiments with the real-life database show that FCBAR outperforms fuzzy Apriori_like algorithm , a well–known and widely used association rules algorithm.

Index Terms **—Cluster Table , Fuzzy Association Rules**

1. INTRODUCTION

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} represents 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 , …, 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 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 ⊆ T . An association rule is an implication of the form "A→ B" , where A ⊆ I , B ⊆ I , and A∩B=ø . Usually ,an association rule A → 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→ B)=|AB |/|D| ≥ Min_supp ,and
Dcon*f*(A→B)=|AB|/|A|≥Min_conf,
where |A | is the number of transaction that contain A , and |D | is the total number of transaction in database D.

The authors are faculty of Software Engineering ,Islamic Azad University ,mashhad branch ,Ostad yousefi street P.O.BOX 91735-413 , Mashhad,Iran.

Generally, there are two phases for mining association rules. In the first phase, we first find all the large itemsets, the supports of large itemsets are larger than or equal to the minimal support specified by user. If there are k items in a large itemset, then we call it a large k-itemset . We can find that a subset of a large itemset must also be large. Subsequently, we use the large itemsets generated in the first phase to generate effective association rules. If the confidence of an association rule is larger than or equal to the minimum confidence specified by user, then it is effective. The key work for finding the association rules is to find all large itemsets.

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 years, 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 approaches 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 large itemsets.

A fuzzy association rule understood as a rule of the form A→ 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→ B)= (∑A(x)⊗ B(y)) / |D|,
Dconf (A→ B)=(∑ A(x)⊗ B(y)) / ∑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, ⊗ is a t-norm [13]. 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 Association Rules (FCBAR) is proposed. For the proposed algorithm , quantitative attributes are divided into various linguistic values . The reminder of this paper is organized as follows : The cases for fuzzy partitioning are explained in section 2 (as defined in [14]).

We present our algorithm in section 3. In section 4 experimental results are given to show the performance of the proposed algorithm . Section 5 is the conclusion .

2. 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 [15] , the fuzzy rules generated by wang and Mendel [16] , 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 . Triangular memberships functions are used for each linguistic value as defined in [14]. Hence, each linguistic value is a fuzzy number, which is a fuzzy subset in the universe of discourse that is both convex and normal .

Notations used in this case are stated as:

K : number of linguistic values in a linguistic variable ;

d : number of attributes of database relation, that d >=1 ;

$A_{K,i(m)}^{X(m)}$: i(m)th linguistic value of k various linguistic value defined in X(m) , where 1≤ i(m) ≤ k;

$\mu_{K,i(m)}^{X(m)}$: membership function of $A_{K,i(m)}^{X(m)}$;

t_p : Pth tuple of a database relation, that $t_p = (t_{p1}, t_{p2}, \ldots, t_{pd})$ and p ≥ 1 .

A quantitative attribute can be divided into K various linguistic values (K=2,3,4…).

For example, for the attribute 'Age' (range from 0 to 60), we describe K=2, K=3 in figure1.

$\mu_{K,i(m)}^{Age}$ Can be represented as follows:

$$\mu_{K,i(m)}^{Age}(x) = \max\{1 - |x - a_{i(m)}^{K}| / b^{K}, 0\} \quad (1)$$

Where

$$a_{i(m)}^{K} = mi + (ma-mi).(i(m) -1) / (k-1) \quad (2)$$

$$b^{K} = (ma-mi) / (k-1) \quad (3)$$

Where ma is the maximum value of the attribute's domain, and me is the minimum value. It is clear that ma=60 and mi= 0 for 'Age'.

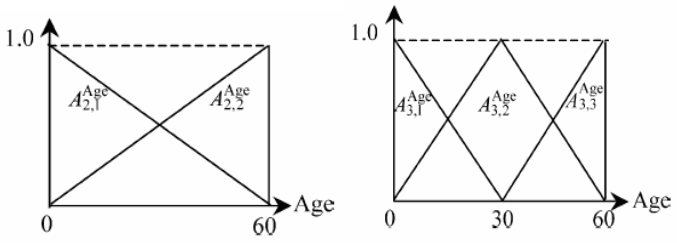
Generally, $A_{K,i(m)}^{Age}$ can be described in a linguistic sentence such as:

$A_{K,1}^{Age}$: young ,and below 60/(K -1) (4)

$A_{K,K}^{Age}$: old ,and above [60– 60/(K–1)] (5)

$A_{K,i(m)}^{Age}$: close to (i - 1). [60 – 60/(K – 1)]

and Between (i(m) - 2).[60– 60 / (K–1)] and i(m) . [60 – 60/(K – 1)] , for 1< i(m)<K (6)



a) k=2 for 'Age' b) k= 3 for 'Age'

Fig 1

3. OUR ALGORITHM (FCBAR)

The performance is dramatically decreased in the process of many fuzzy association rule 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. If an alternative method can decrease the number of database scans, and reduce the number of contrasts, the performance efficiently will be improved . Thus, we propose an efficient FCBAR method for discovering the fuzzy large itemsets, and the main characteristics are in follow. FCBAR only requires a single scan of the transaction database, following by contrasts with the partial cluster tables. Not only this prunes considerable amount of data reducing the time needed to perform data scan and requiring less contrast, but also ensures the correctness of the mined results.

A FCBAR Algorithm

The FCBAR employs some efficient cluster tables to represent database D by a single scan of the database, following by contrasts with the partial cluster tables.

Algorithms Table_based_Clustering_pruning (D, Minsup)
Input: D, Minsup
Output: Answer(Answer = U L_k , for 1≤k≤M)
Begin
1) cluster_Table_Create(D,Minsup);
2) for (k=2; $L_{k-1} \neq \emptyset$; 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.2. Main program for the FCBAR.

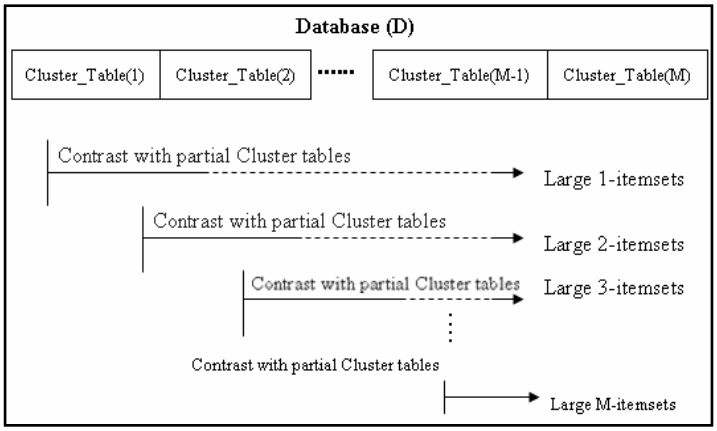


Fig.3. FCBAR need the contrast with only partial cluster tables.

Fig.2 is the algorithmic form of FCBAR , which, for ease of presentation, is divided into three parts .
Part 1 gets a set of fuzzy 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–table(k), 1≤ k ≤ M , where M is the length of the longest transaction record in database. Meanwhile, the set of fuzzy 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 4. when the length of candidate itemset is k , the support is calculated with reference to the cluster–table(k) . Then it is contacted with the Cluster–Table(k+1), Cluster–Table (k+2),… .

Procedure Larg_Itemset_Gen(C_k)
Input: C_k
Output: L_k
Begin
1)While($C_k \neq \emptyset$)**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.4.Procedure of fuzzy large k-itemsets Generation for FCBAR.

B An Example Of FCBAR

We present an example to explain the application of our algorithm. There are 20 records in the database.
The example is shown in table 1.
Each transaction in table 1 consists of pairs (x,t) such that x is an item and t is the number of item x in transaction.
Part 1 gets a set of fuzzy large 1-itemset and create four cluster tables are shown in table 2: (a), (b), (c)and (d).
Then to find the fuzzy support of each fuzzy candidate 2-itemset, we start from cluster–table(2) And calculate the fuzzy support of candidate itemset in this cluster–table . Next we do the same in cluster–table(3) and cluster–table(4). Finally ,the fuzzy support of candidate itemset is the sum of Fuzzy supports in cluster–table(2), cluster–table(3) and cluster–table(4).
To find the fuzzy support of each fuzzy candidate 3-itemset we calculate the support of candidate itemset in cluster–table(3) and cluster–table(4) . Then the support of candidate itemset is sum of its supports in these cluster–tables.

Table 1
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 2

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–able(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

4. EXPERIMENT

To evaluate the efficiency of the proposed method, we have implemented the FCBAR, 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 FCBAR algorithm is compared to the Apriori_like algorithm .

(1) 60000 transaction records of experimental data are sampled randomly from the real-life Database, with k=3, that k is the number of linguistic values in each attribute . The test database contains 10 items , in which the longest transaction record contains 7 items. The performance of FCBAR algorithm is compared to Apriori algorithm under various user specified minimum support (minsupp), such as 50%, 40%, 30%, 20% . the result are shown in Fig.5.

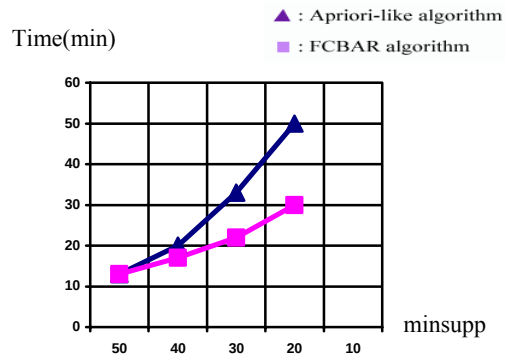


Fig.5.Performance of FCBAR and Apriori-like algorithm on 60000 records.

We can show that whenever the minsupport decreases ,the gap between algorithms becomes more evident.

(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 FCBAR algorithm is compared to apriori-like algorithm where minimum support is 30% (Fig.6).

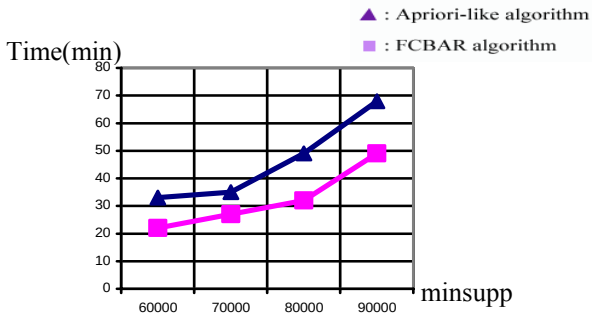


Fig.6.Performance of FCBAR and Apriori-like algorithm at minsupport 30%.

When the number of transaction increases ,again the gap between algorithms increases too.

(3) 60000 records are sampled from test database. Then the performance of algorithms are compared with various number of attributes where minsupp is 30% (Fig 7).

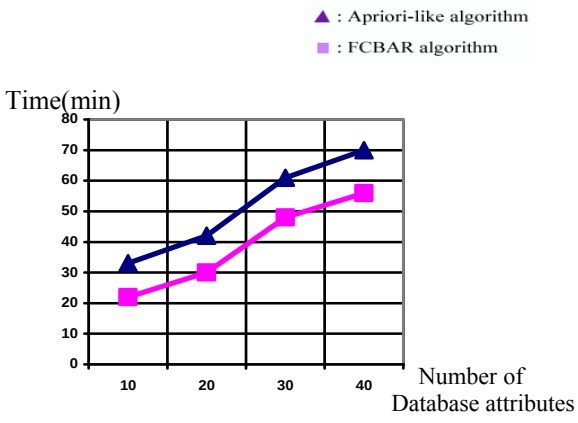


Fig.7.Performance of FCBAR and Apriori-like algorithm on 60000 records with minsupport 30%.

Experiments show that FCBAR algorithm outperforms Apriori-like algorithms when the number of attributes increases.

5. CONCLUSIONS

In this paper we proposed the efficient algorithm for mining fuzzy association rules. The FCBAR algorithm creates cluster table to aid discovery of fuzzy large itemsets. Contrasts are performed only against the partial cluster tables that were created in advance.

Experiments with the real-life database show that FCBAR outperforms Apriori_like algorithm, a well-known and widely used association rule. When there is an increase in the number and the size of pattern discovered, the performance gap between the algorithms becomes more evident. The characteristics of FCBAR are the following.

It only requires a single scan of the database, following by contrast with the partial cluster tables. Not only this prunes considerable amount of data reducing the time needed to perform data scans and requiring less contrast , but also ensures the correctness of the mined result.

Moreover a parallel version of our algorithm can be implemented. Such that each CPU can be utilized to process a cluster table; thus the performance will be improved.

REFERENCES

[1] J.W.Han,M.kamber, data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2001.

[2] R.Agrawal, T. Imielinski, 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,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] Wai-HoAu, and K.C.CChan. 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] 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.

[14] Y.Hu, R.Chan, G.Tzng : Discovering fuzzy association rules using fuzy partition methods. Knowledge-Based System 16 (2003) 137-147.

[15] H.Ishibuchi,K.Nozaki,N.Yamamoto,H.Tanaka, Selection fuzzy if-then rules for classification problems using genetic algorithms, IEEE Transaction on fuzzy Systems 3 (3) (1995) 206-270.

[16] L.X.Wang , J.M.Mendel ,Generating fuzzy rules by learning from examples , IEEE transaction on systems, Man , and Cybernetics 22(6) (1992) 1414 -1427.