International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011
ISSN: 1793-8201

Frequent Itemsets from Multiple Datasets with Fuzzy data

Praveen Arora, R. K. Chauhan and Ashwani Kush

Abstract—**Association rules from large Data warehouses are becoming increasingly important. In support of this trend, the paper proposes a new model for finding frequent itemsets from large databases that contain tables organized in a star schema with fuzzy taxonomic structures. The study aims to incorporate the previous developed algorithms on mining fuzzy generalized association rules and Mining Association rules in Entity relationship Models to discover a new algorithm. The paper focuses on the extraction of multi level linguistic association rules from multiple tables and examines the performance of extracted rules. An example given in the study demonstrates that the proposed mining algorithm can derive multi level fuzzy association rules from multiple datasets in a simple and effective manner.**

Index Terms—**Association rules, Data Mining, Fuzzy Data, ER Models**

I. Introduction

An Association Rules mining is an important process in data mining, which determines the correlation between items belonging to a transaction database and various algorithms have been presented for efficiently mining them in large databases. There are very few published results on how to mine association rules when data resides in multiple tables with fuzzy taxonomic structures. The proposed framework is extended to deal with more general, flexible and linguistic knowledge in multi level fuzzy association rules from multiple tables. Crisp Association Rules Mining algorithms can mine only binary attributes that can potentially introduce loss of information due to the sharp ranges where as Fuzzy association rules use fuzzy logic to convert numerical attributes to fuzzy attributes and corresponds better to intuition and prevent overestimation of boundary cases as attribute values are represented in the interval [0,1], instead of having just 0 and 1, and to havetransactions with a given attribute represented to a certain extent (in the range [0, 1]) which replaces crisp binary attributes with fuzzy ones Mangalampalli and Pudi [15]. It helps user to grasp the concept easier and the scheme works on multiple tables so that it can take advantage of the knowledge already embedded in Entity relationship model regarding the relationships between the database entities. The discovery of interested association relationships among huge amounts of business transaction records can help in many business decision-making processes [3]. The problem of mining multi level linguistic association rules is to find all rules that satisfy a user-specified minimum support and minimum confidence from tables that are designed using ER models and that contain fuzzy data. This study presents an efficient algorithm that generates all significant association rules between sets of items in a large database of customer transactions.

Rest of the paper is organized as; Section 2.0 presents recent studies about various existing association rule mining algorithms, Section 3.0 presents the newly discovered algorithm that will find Frequent Itemsets from Multiple Datasets with fuzzy data, Section 4.0 gives the illustration of the newly discovered algorithm using a small example, Section 5.0 describes the results obtained. Section 6.0 concludes the study with a vision of future work.

Manuscript received September 4, 2009.

Praveen is working at JaganNath Institute of Mgmt. Sciences, Delhi, India. She can be reached at praveen@jimsindia.org.

Ram Kumar is in DCSA, Kuurkshetra University Kurukshetra India. He is Chairman and Professor and can be reached at rkc.dcsa@gmail.com

Ashwani Kush is in Computer Science department at university college, Kurukshetra University India. He can be reached at akush20@gmail.com

II. Recent Studies

The problem of mining association rules has been discussed in [4]. It is an AIS algorithm known to be the first published algorithm to generate all large itemsets in a transaction database. The AIS algorithm makes multiple passes over the entire database. During each pass, it scans all transactions. In the first pass, it counts the support of individual items and determines which of them are large or frequent in the database. Large itemsets of each pass are extended to generate candidate itemsets. After scanning a transaction, the common itemsets between large itemsets of the previous pass and items of this transaction are determined. These common itemsets are extended with other items in the transaction to generate new candidate itemsets. A large itemset *l* is extended with only those items in the transaction that are large and occur in the lexicographic ordering of items later than any of the items in *l*. To perform this task efficiently, it uses an estimation tool and pruning technique. The estimation and pruning techniques determine candidate sets by omitting unnecessary itemsets from the candidate sets. Then, the support of each candidate set is computed. Candidate sets having supports greater than or equal to min support are chosen as large itemsets. These large itemsets are extended to generate candidate sets for the next pass. This process terminates when no more large itemsets are found. The disadvantage is that this results in unnecessarily generating and counting too many *candidate* itemsets that turn out to be *small*.

Set–Oriented Mining for association rules in relational Databases is described by [5] where an algorithm **SETM** has been developed with the desire to use SQL to compute large itemsets. The *Candidate* itemsets are generated on-the-fly as the database is scanned, but counted at the end of the pass. New *candidate* itemsets are generated the same way as in AIS algorithm [4], but the TID of the generating transaction is saved with the *candidate* itemset in a sequential structure. At the end of the pass, the support count of *candidate* itemsets is determined by aggregating this sequential structure

In addition to having same disadvantage as of the AIS algorithm [4], also it is that for each *candidate* itemset, there are as many entries as its support value. *Apriori* and *AprioriTid* algorithms [6] are used to discover association rules between items in a large database of sales transactions. Results reveal that these algorithms always outperform the earlier algorithms **AIS** and **SETM**. The study also emphasizes how the best features of the *Apriori* and *AprioriTid* can be combined into a hybrid algorithm, called AprioriHybrid. Experiments reveal that AprioriHybrid scales linearly with the number of transactions. The execution times decrease little as the number of items in the database increases. As the average transaction size increases, the execution times increase only gradually. In another study [7] the properties of association rule discovery in relations has been discussed. The Basic algorithm proposed has been based on the same basic idea of repeated passes over the database as in AIS algorithm [4] with the difference that the Basic algorithm makes careful use of the combinatorial information obtained from previous passes and in this way avoids considering many unnecessary sets in the process of finding the association rules. Experimental results of the algorithm shows improvement when compared against the previous results, and is also simple to implement. Studies on mining association rules find rules at single concept level, but mining association rules at multiple concept levels may lead to the discovery of more specific and concrete knowledge from data [8]. In this study, a top-down progressive deepening method is developed for mining multiple level association rules from large transaction databases. Concept hierarchy handling, methods for mining flexible multiple-level association rules, and adaptation to difference mining requests are also discussed in the study. [9, 10] introduce the problem of mining generalized association rules where a database of transactions consists of a set of items, and taxonomy (is-a-hierarchy) on the items. The paper finds associations between items at any level of the taxonomy. The study replaces each transaction with an "extended transaction" that contains all the items in the original transaction as well as all the ancestors of each item in the original transaction. Any of the earlier algorithms are then run on these transactions to get generalized association rules. But this Basic approach has been found to be slow. It presents two algorithms *Cumulate* and *EstMerge* for finding generalized association rules. [11] proposed a method to handle quantitative attributes for which each attribute is assigned several fuzzy sets. Fuzzy sets handle numerical values better than existing methods because fuzzy sets soften the effect of sharp boundaries. The fuzzy set concept is better than the partition method because fuzzy sets provide a smooth transition between member & non-member of a set. The paper uses Significance and certainty factor to determine the satisfiability of itemsets & rules. In many real life applications, the related taxonomic structures may not be necessarily crisp, rather certain fuzzy taxonomic structures reflecting partial belonging of one item to another may pertain [12]. For example, Carrot may be regarded as being both Fruit and Vegetable, but to different degrees. Here, a sub-item belongs to its super-item with a certain degree.

A crisp taxonomic structure assumes that the child item belongs to its ancestor with degree 1. But in a fuzzy taxonomy; this assumption is no longer true. Different degrees may pertain across all nodes (item sets) of the structure. The study focuses on the issue of mining generalized association rules with fuzzy taxonomic structures. The study extends **Apriori** and **Fast** algorithm to allow discovering the relationships between data attributes upon all levels of fuzzy taxonomic structures. Various sub-algorithms have also been developed.

Current data mining algorithms [13] handle databases consisting of a single table. This study addresses the problem of mining association rules in databases consisting of multiple tables and designed using the entity-relationship model. To address this issue the study introduces the notion of entity and join support and presents two algorithms: algorithm **Apriori Join,** for mining the outer join of a star schema tables using the knowledge of the schema, and algorithm **Apriori Star**, for directly mining the star schema database. A study by [14] aims at dealing with the fuzzy association rules of the form X→Y where X and Y can be collections of fuzzy sets. It incorporates fuzziness in the exact taxonomies that reflect partial belongings among itemsets. A number of sub-algorithms as Apriori fast algorithms (GAR), an algorithm to deal with fuzzy taxonomies (FGAR), and An algorithm to deal with linguistic hedges (HFGAR) have been introduced to express meaningful knowledge in a more natural and abstract way.

III. Proposed Study

In real databases, typically a number of tables will be defined. In this paper, we examine the problem of mining association rule from a set of relational tables. In particular we are interested in the case where the tables form a star structure as shown in the Figure 1 below. A star schema consists of *fact table* (FT) in the center and multiple dimension tables.

The Product table of the above structure is given in the table Table 1 below. In this table the attribute Product_name is fuzzy and the fuzzy taxonomic structure over the attribute Product_name is given in the figure 1.

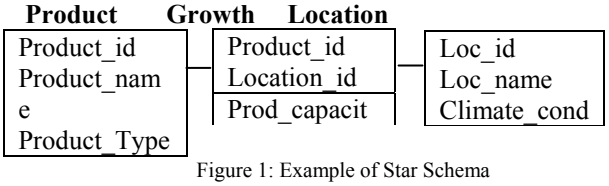

Figure 1: Example of Star Schema

The paper focuses on finding rules from multiple tables that contain fuzzy data and are arranged in star schema. If traditional data mining algorithms [12,13] are used to discover association rules in such a case then the join of these tables will affect the efficiency and cost of the algorithm used. To overcome this problem the study focuses on extending the traditional algorithms in such a way that the mining of multi level fuzzy association rules become fairly simple. During the fuzzy Association Rule Mining process, the original dataset is transformed into an extended one [12] with attribute values in the interval [0, 1] using the equation given in [12]:

TABLE 1 : PRODUCT

Prod_id	Prod_name	Type
001	Wheat	Club
002	Wheat	Durum
003	Corn	White
004	Corn	Yellow
005	Bean	Azuki
006	Bean	Lentils
007	Peanut	Runner
008	Sesame	Indian
009	Sesame	Ethiopian
010	Rapeseed	Canola
011	Pear	Bartlett
012	Pear	Comice
013	Grape	White
014	Citrus	Lemon
015	Citrus	Orange

$$\mu_{xy} = \max_{\forall l:\, x \to y} (\min_{\forall e \in l} (\mu_{le})) \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

The proposed study focuses on developing an algorithm that will use fuzzy logic for finding fuzzy association rules from tables that are designed using ER models and for this in order to process this extended dataset, we need fuzzy measures analogous to crisp support and confidence, which are defined as degree of support and degree of confidence [12].

$$\mu_{tX} = Support_{tX} = \min_{x \in X} (\max_{a \in t} (\mu_{xa})) \ldots\ldots\ldots (2)$$

$$Dsupport\ (X \Rightarrow Y) = \Sigma count\ (\mu_{tZ}) / |T|. \ldots\ldots (3)$$

$$Dconfidence\ (X \Rightarrow Y) = Dsupport\ (X \Rightarrow Y) / Dsupport\ (X) \ldots\ldots (4)$$

$$= \Sigma_{t \in T} count\ (\mu_{tZ}) / \Sigma_{t \in T} count\ (\mu_{tX})$$

The study in this paper presents an algorithm that uses three phases in generating fuzzy association rules for multiple tables. The following notations are used in the study:

E = fuzzy Entity table

μ = fuzzy membership of any itemset

count = Total count of the degrees of a particular itemset

K=iterations in the pseudocode

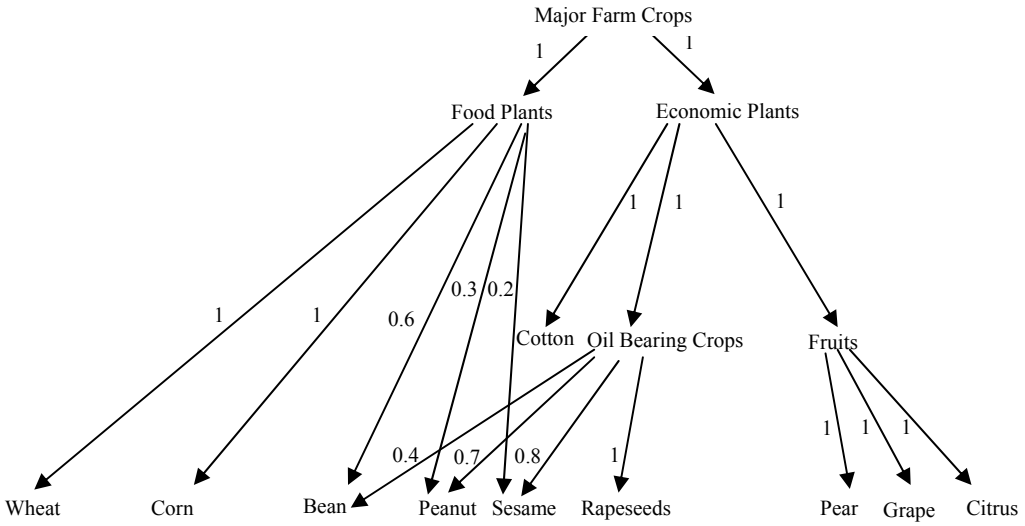

Figure 2: Example of fuzzy taxonomic structures

C_k = Candidate Set in iteration K

c.Entity_Sup=Entity Support of one of the Candidate itemsets

c.join_sup=Join support of one of the Candidate itemsets

FK.next = Record of total number of occurrence of a particular foreign key.

A. First Phase of the Algorithm:

During the first phase, the algorithm determines the degree to which leaf level item belong to its ancestors using the equation number 1 given above. The leaf item, parent item and their corresponding degree is then kept in an Extended table T'.

The occurrence of each foreign key in the relationship table is then maintained in a counter array FK.next. In crisp items, the degree to which leaf level item belong to its parent is always 1 but in case of fuzzy taxonomy, since a leaf node item may belong to more than one parent, their degrees are calculated using fuzzy logic. The candidate collection is initialized to all 1-item sets from all the entity tables.

TABLE 2: LOCATION

Loc_id	Loc_name	Specialist in Food Product/Oil
1001	Punjab	Oil
1002	Rajasthan	FoodProduct
1003	Rajasthan	Oil
1004	Haryana	Food product
1005	Haryana	Oil
1006	Assam	Food Product
1007	Sikkim	Oil
1008	UP	Oil

1009	Himachal	Food Product
1010	Himachal	Oil

B. Second Phase of the Algorithm:

In the second phase, Scanning of all the entity tables is performed and during the scan of one table E_i compute the entity support for all item sets from E_i while also computing their join support using the results obtained from FK.next item sets whose Σ*count* values are greater than min_support x |T|. These item sets are called *frequent item sets*. The item sets from Candidate that have entity support or Join support greater than minimum support are placed into the collection Frequent and the entity item sets that have join support greater than minimum support and the join item sets that have join support greater than minimum support are placed into the collection AllFrequent.

C. Third phase of the Algorithm:

From the Frequent collection generate new candidates using the *apriori_gen* method [13] and all the newly generated item sets are placed into Candidate. All entity item sets in Candidate from all entity tables are then scanned and their entity and join supports are computed. Entity tables and relationship tables are the joined and join support of all the join itemsets are then calculated.

TABLE 3: GROW

Prod_id	Loc_id	Sent_for_Productio n
001	1001	Oil
002	1003	Food Product
003	1004	Food Product
005	1001	Food Product
005	1002	Food Product
005	1005	Food Product
006	1001	Oil
006	1004	Food Product
007	1004	Oil
008	1004	Oil
009	1004	Oil
011	1007	Food Product

The algorithm then moves to phase 2 and the process keeps on repeating until all frequent itemsets are received and the Candidate collection becomes empty.

Algorithm

1) Determine the degree to which leaf item belongs to its ancestor using equation (1).
2) Repeat until all leaf nodes in taxonomy
3) Repeat until all interior nodes in taxonomy
4) μ (LNi, INj) = $\max_{\forall l: INj \rightarrow LNi} (\min \forall_{e \text{ on } l} (\mu_{le})$ //LNi=leaf node, INi=Interior(Parent)Node
5) Insert leaf node, parent node and the degree to which leaf node belongs to parent node in the Transaction table T'.
6) Repeat until all foreign key f in Relationship table R
7) f++
8) FK.next=f
9) Set K=1
10) Set $C_{k=}$ candidate 1-itemsets from all the Entity tables from E_1 through E_n.

Figure 4: Pseudo-code for phase 1.

1) Repeat for all Entity tables E_i where i=1 to n
2) Repeat for all itemsets I belongs to Ei
3) Compute the sum of all the degree ∑count that are associated with the transaction in T
4) If (Σcount >= (min_sup x |T|))
5) Compute Entity Support from Ei
6) Compute Join support from FK.next
7) FK.next++
8) C_k = I
9) Frequent F = (if c. Entity_Sup || c. join_sup Є C_k >= min_sup) // c = candidates of Ck
10) AllFrequent AF = (c.E.entity_Sup || c.J.join_sup Є C_k >= min_sup) // c.E =candidates of Entity table

Figure 5: Pseudo-code for phase 2.

1) C_k=apriori_gen(F, min_sup) // [11]
2) If C_k = Ø then Exit.
3) Repeat for all c.E Є C_k
4) Repeat for all I Є E_i where i=1 to n
5) compute Entity Support
6) compute Join Support
7) R * E_1 * E_2 * ….. * E_n to form join table JT.
8) Repeat for all c.J Є C_k
9) compute join support from JT
10) Increment K by 1
11) Go to phase 2.

Figure 6: Pseudo-code for phase 3.

IV. ILLUSTRATION OF THE ABOVE PSEUDOCODE FOR MINING FREQUENT ITEMSETS FROM MULTIPLE DATASETS WITH FUZZY DATA.

The new algorithm described above is applied on the tables designed using entity relationship model. For our experiment we took a very small database of three tables containing **Product**, **Location** and a transaction table **Grow** as shown in Tables 1, 2 and 3 respectively. A join table is also formed on the fly during each pass of the algorithm by joining all the entity and Relationship tables. Entity Support and Join Support is calculated for itemsets that belong to Entity table and join support is calculated for itemsets that belong to Join table. The support of an entity item set with respect to its entity table is called *entity support*. The support of an (entity or join) item set with respect to the table Join is called *Join support*. To compute Entity Support we count the number of rows from Entity table that contain the itemset and then divide this count with the total number of rows in the Entity table. To compute Join Support we have to count the number of rows that contain the itemset and that belong to Join Table and then divide this value by the cardinality of Join table.

The study will focus on finding rules such as: Oil bearing Crops ⇒ Haryana which implies that the Production of Oil bearing crops like Bean, Peanut, Sesame and Rapeseed grows more in Haryana where the item bean partially belongs to Food plants with degree $\mu_{\text{Food plantsBean}}$ ie 0.6 and also partially belongs to oil bearing plants with degree $\mu_{\text{OilbearingplantsBean}}$ ie 0.4. Similarly Peanut, Sesame also belongs to both food plants and Oil bearing Crops with two different degrees. As can be seen from Table 1 and Table 2, both Oil bearing Crops and

Haryana belong to two different entities designed using ER models. In this example the attribute Bean, Peanut and Sesame of table **Product** is first converted into fuzzy taxonomic structure reflecting partial belonging of one item to another. Fig 2 explains the concept.

The Algorithm given above is used to find such kind of rules. The fuzzy extensions that will be presented in this study will enable us to discover not only crisp generalized association rules but also fuzzy generalized association rules when databases consisting of several tables organized in a schema within the framework of fuzzy taxonomic structures. Strong Association rules between items of fuzzy nature existing in multiple tables can be calculated that will undoubtedly help in understanding things in broad spectrum.

V. Results

The study focuses on finding multi level association rules from databases that contain multiple tables designed using entity relationship models with fuzzy data which were undiscovered in the previous studies. A new algorithm has been developed in this study to discover such rules. Table 4 shows the frequent itemsets and the final results of the study are given in table 5. For our study the min_support is 10% and min_confidence is 40%.

Table 5 shows the final result of the study: The interesting rules of the study. Rule Oil bearing Plants→ Haryana implies that the production of Oil from Oil bearing plants is mostly done at Haryana with degree of support 22.5% and degree of confidence 66.17% which meets our min_sup and min_conf threshold. Degree of Support is calculated as:

TABLE 4: Frequent Itemsets

1-ItemSet	**ΣCount**	**Entitysupport**
Wheat	2	13.33%
Bean	2.8	18.66%
Sesame	1.6	10.66%
Oil-bearing Plants	5.1	34%
Punjab	2	20%
Rajasthan	3	30%
Haryana	6	60%
Sikkim	1	10%
2-ItemSet	**ΣCount**	**Joinsupport**
Oil bearing Plants, Haryana	3.5	29.16%

TABLE 5: Interesting Rules

2-Itemset	**DSupport**	**Dconfidence**
Oil bearing plants→ Haryana	22.5%	66.17%

min(0.6,1)+min(0.6,1)+min(0.7,1)+min(0.8,1)+min(0.8,1)=3.5

∑count value of (Oil bearing plants, Haryana) = 3.5

DSupport = ∑count (Oil bearing plants, Haryana)/No. of Transactions * 100 = 3.5/15*100 = 22.5%

Degree of confidence (Oil bearing plants, Haryana) = ∑count (Oil bearing plants, Haryana)/∑count (Oil bearing plants) * 100 = 3.5/5.1 *100 = 66.17%

The study will help the management of the Supermarkets in making their business plans that includes what to put on sale, how to design coupons, how to maximize the profits. Analysis of the transaction data which includes the customer personal data and the goods that customers purchase converted into fuzzy taxonomic structures reflecting partial belonging of item to another will be the approach in order to improve the quality of such decisions.

VI. Conclusion

Traditional approaches handles crisp and fuzzy data very well but very less published results show that databases that contain multiple tables with fuzzy data having taxonomy can be handled efficiently. The Proposed algorithm is discovered by extending these traditional algorithms and helps to find the multi level fuzzy association rules in Entity –Relationship modeled databases, which is capable to handle multiple tables. The study analyzes how the attributes of several entities appear together. The Study also analyzes the rules with respect to the relationships existing between the entities and their ancestors. If several relationships exist between two or more entities, then the fuzzy association rules between their attributes and ancestors are examined with respect to each such relationship. The discovered algorithm uses the join and entity supports in determining frequent item sets. By considering the entity support it does not eliminate from the result entity item sets that are frequent with respect to their entity table but not with respect to the relationship table and it also allows the computation of correct support and confidence for rules existing among attributes of the same entity table.

Future enhancement can be achieved out by investigating efficient algorithms for computing the support of item sets relative to various connectors that may exist for a set of fuzzy attributes and to examine the effect of the topology of the entity-relationship diagram on the resources required for these computations. The algorithm described above is implemented using JAVA to find frequent itemsets efficiently.

References

[1] Frawley, W.J., Piatetsky-Sapiro, G., and Matheus, C.J. (1992). "knowledge discovery in databases: An overview"

[2] Ishibuchi, H., Yamamoto, T. and Nakashima, T. (2001). ICDM Proceedings of the IEEE International Conference on Data Mining. pp. 241-248.

[3] Han, J., Kamber, M. (2001). "Data Mining: Concepts and Techniques"., Harcourt India Pvt Ltd.

[4] Agrawal, R., Imielinski, T., and Swami, A. (1993, May). Mining Association Rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, pp. 207-216, Washington, D.C.

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

[6] Agrawal, R., and Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. of the VLDB conference, Santiago, Chile. Expanded version available as IBM Research Report RJ9839, June 1994.

[7] Mannila, H., Toivonen, H., and Verkamo, A.I. (1994, July). Efficient algorithms for discovering association rules. In KDD-94: AAAI workshop on Knowledge discovery in databases, pp. 181-192, Seattle, Washington.

[8] Han, J., and Fu, Y. (1995, September). Discovery of Multiple-level Association Rules from Large Databases. Proceedings of the 21st International Conference on VLDB, Zurich, Switzerland.

[9] Srikant, R., and Agrawal R. (1995). Mining Generalized Association Rules, proceedings of the 21st VLDB Conference, Zurich, Switzerland.

[10] Srikant R. and Agarwal R. (1996). Mining quantitative association rules in Large Relational Tables, in proceedings of the ACM SIGMOD International Conference on Management of Data, pp 1-12, Montreal, Quebec, Canada.
[11] Kuok, C.H., Fu, A., and Wong, M.H. (1998). Mining Fuzzy association rules in databases ACM SIGMOD Record, 27(1), ACM Press.
[12] Chen, G., Wei, Q., and Kerre E. (2000). "Fuzzy Data Mining: Discovery of Fuzzy Generalized Association Rules". In Bordagna and Pasi (eds.), Recent Research issues on Management of Fuzziness in Databases, Physica-verlag (Springer).
[13] Cristofor, L., and Simovici, D. (2001-2002). "Mining Association Rules in Entity-Relationship Modeled Databases", Technical Report, UMB.
[14] Chen, G., and Wei, Q. (2002). Fuzzy association rules and the extending Mining Algorithms, Information Sciences: An International Journal, 147, pp.201-228.
[15] Mangalampalli A. and Pudi V.(2009), Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets., In proceedings of IEEE International Conference on Fuzzy Systems., Jeju Island, Korea.