Association Rule Mining over Fuzzy Taxonomy for Databases with Multiple Tables

Dr. Praveen Arora[1] , Dr. Priyanka Gandhi[2]

1. Professor, Department of Management, Jagan Institute of Management Studies, Rohini.

2. Associate Professor, Department of Management, Jagan Institute of Management Studies, Rohini

***Abstract*—**In this study, the problem of mining association rules in databases with many tables of fuzzy data with taxonomy and tables created using entity-relationship (ER) models is discussed. The majority of the data mining methods in use today deal with datasets that just contain one table. For several tables with ambiguous data and taxonomic structures, few techniques are effective. The "Fuzzy Generalized Rules for ER Models" algorithm is proposed in this paper. In order to find a new algorithm, the study intends to combine the previously published methods Extended Apriori and Apriori star. The research will aid in standardizing algorithms for extracting relevant information from database tables containing information with ambiguous taxonomic structures.

***Keywords*— Association Rules, Data Mining, Fuzzy data, ER models,**

I. INTRODUCTION

Using massive data sets, Association Rules Mining (ARM) extracts intriguing and unexpected rules. The main goal of ARM is to identify all subsets of items or attributes that regularly appear in a large number of database records or transactions and, in addition, to extract rules on how the existence of one subset of items affects the presence of another subset [12]. The values close to the borders would be over- or underestimated as a result of the quantitative values being divided into clear sets. This issue can be solved by fuzzy sets by allowing partial memberships to various sets. The computations required to cope with the various data structures are made possible by fuzzy set theory. It makes the set fuzzy rather than crisp by allowing the intervals to overlap. Items can then display partial membership to many sets, solving the issue of abrupt boundaries [3]. Fuzzy sets are generalised sets that permit a range of membership for their constituent parts. Typically, the membership degree structure [7] is chosen using the actual unit interval [0; 1]. The study focuses on the multi-level linguistic association rules' extraction from various tables and evaluates the effectiveness of the retrieved rules. Finding 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 is the challenge of mining multi-level linguistic association rules. With a sizable database of customer transactions, this work proposes an effective algorithm that creates all meaningful association rules between groups of objects. With the use of this investigation, rules can be found that the traditional quantitative technique would have missed.

The remainder of the paper is divided into five sections: Section 2 introduces recent studies, Section 3 discusses the shortcomings of current systems, Section 4 discusses the proposed study, and Section 5 introduces the recently developed algorithm, Extended Apriori Fast, which will discover Generalized association rules for ER models with fuzzy data. For a sample database of customer transactions, the algorithm is used, and Section 6 shows the results of the implementation. A suggestion for further research appears in Section 7 of the report.

II. RECENT STUDIES

Rules for mining associations have been a topic of discussion [2]. The first documented AIS algorithm to produce all huge itemsets in a transaction database, according to knowledge. It concentrated on improving databases with the features required to handle decision support query processing. The objective of this programme was to find qualitative rules. Only one item in the consequent can be used with this strategy. In other words, association rules take the form XIj | where Ij is a single item in the domain I, X is a group of objects, and is the confidence of the rule. The AIS algorithm scans the full database several times. As the database is searched, candidate itemsets are created and counted instantly. The drawback is that this causes needlessly large numbers of little candidate itemsets to be generated and counted.Set–Oriented Mining for association rules in relational Databases is described by [9] where an algorithm **SETM** has been developed with the desire to use SQL to compute large itemsets. Main features are:

Candidate itemsets are generated on-the-fly as the database is scanned, but counted at the end of the pass.

1. New *candidate* itemsets are generated the same way as in AIS algorithm, but the TID of the generating transaction is saved with the *candidate* itemset in a sequential structure.
2. 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, also it is that for each *candidate* itemset, there are as many entries as its support value.

Apriori and *AprioriTid* algorithms [1] 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 [11] 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 [2] 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. [13, 14] 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. [10] 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 [5]. 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 [6] 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 [4] 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. DEMERITS IN EXISTING SYSTEM

The goal of the study is to create an algorithm that can find fuzzy generalized association rules for multiple tables that makes use of the fact that the fuzzy data exists in a hierarchical taxonomy (concept hierarchy) to produce various association rules at various levels in the taxonomy for databases with multiple tables created using ER Models. Traditional data mining algorithms must first join entity tables and relationship tables in order to be computed, which has a negative impact on the effectiveness and cost of the algorithm used to find association rules in such environments (where either the data of the single table is used with fuzzy taxonomic structures or even if used from multiple tables then the concept of fuzzy data is not introduced). It has been noted that very little work has been put towards creating multi-level fuzzy data mining association rules for several tables created using ER Models.

IV. PROPOSED STUDY

When databases made up of multiple tables structured in a schema within the framework of fuzzy taxonomic structures, the fuzzy extensions that will be described in this study will allow us to uncover both crisp and fuzzy generalised association rules. It is possible to construct strong association rules between fuzzy elements that are included in different tables, which will surely aid in grasping a wide range of concepts.

An example of such a rule can be young ⇒ Meat which implies that customer of the age group 20-30 and 30-40 might turn to buy Meat where the age group 30-40 partially belongs to Young with degree $\mu_{young30\text{-}40}$. The following example finds above mentioned fuzzy generalized rule.

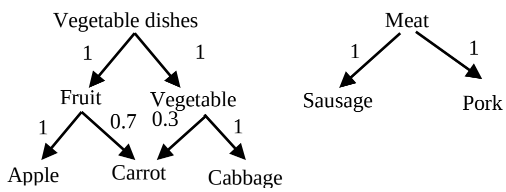


Fig 1: Example of fuzzy taxonomic structure over item

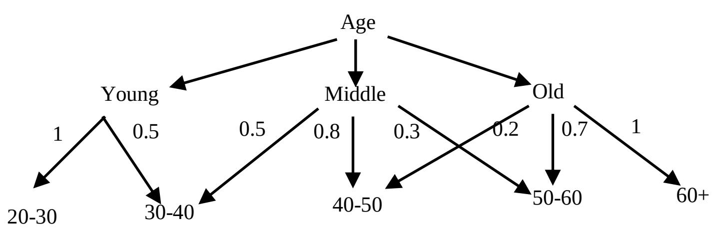


Fig 2: Example of fuzzy taxonomic structure of Age

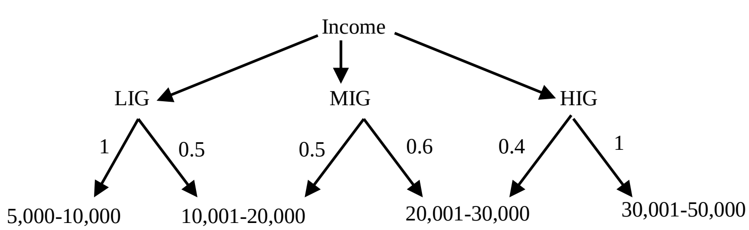


Fig 3: Example of fuzzy taxonomic structures of Income

The study aims to extend the previous developed algorithms Extended Apriori and Apriori star to discover a new algorithm Extended Apriori Fast which is given below.

V. IMPLEMENTATION

<u>Algorithm Extended Apriori Fast</u>

Determine the degree to which leaf item belongs to its ancestor.

1. **forall** leaf nodes LNi Є Taxonomy **do**
   forall interior nodes INj Є Taxonomy **do**
   μ (LNi, INj) = $\max_{\forall l:\, INj \rightarrow LNi}$ (min $\forall_{e \text{ on } l}$ (μ_{le})
   insert into Transaction T' // Extended Transaction set
   values LNi, INj, μ (LNi, INj)
   endfor **endfor**
2. **forall** foreign key f Є R do // R = Relationship Table.
   f++; **endfor**
3. Set K=1; C_k = Itemsets (E_1, E_2,…., E_n).
4. **forall** entity tables E_i (where i=1 to n)
   forall itemsets I Є E_i
   Compute Σcount // sum of all the degree that are associated with the transaction in T
   If (Σcount >= (min_sup x |T|))
   Compute Entity Support and join support
   C_k = I, K++
   endif
   endfor; **endfor**
5. Frequent F = (if c. Entity_Sup || c. join_sup Є Ck >= min_sup) // c = candidates of Ck
   AllFrequent AF = (c.E.entity_Sup || c.J.join_sup Є Ck >= min_sup) // c.E =Entity Itemset; c.J = Join Itemset
6. If C_k = Ø then Exit.
7. **forall** c.E Є C_k
   forall I Є E_i (where i=1 to n)
   Compute Entity Support
   Compute Join Support
   endfor; endfor
8. R * E_1 * E_2 * ….. * E_n to form join table JT.

In this phase **Extended Apriori Fast Algorithm** is implemented which uses fuzzy logic for finding the fuzzy association rules from multiple tables. For this implementation the small dataset was taken from a supermarket for the goods that customers have purchased. As shown in Table 1 ancestors of the table data are added in Transaction table to form Transaction' and as shown in Table 2 and 3 ancestors are added in table Customer to form extended table Customer'. These ancestors are added so as to find multi level linguistic association rules.

The min-support threshold was taken 40% and min-confidence was taken 60%. Here, it is emphasized that these thresholds are context-dependant, which should be defined according to the concrete situation. Here, by a frequent item, it is meant that the itemset whose *Dsupport* is more than min-support threshold. Extended Apriori Fast algorithm is applied on Table 1 and 2 and frequent 1-itemsets are generated that are given in Table 3. Σcount values of the candidate itemsets are calculated and then Entity Support and Join Support of the Item sets that belong only from a single Entity table are calculated and Join support of Item sets that belong to multiple tables are calculated. 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*. Join support for any item set can be calculated but the entity support is defined only for entity item sets.

Table 3: Frequent Table

Frequent Item sets	Σcount	Dsupport
{Pork}	4	40%
{Fruit}	2.3	50%
{Vegetable}	2.7	33%
{Vegetable dishes}	5.1	63%
{Meat}	5.5	75%

{Young}	2.0	55%
{Middle}	1.5	41%
{LIG}	1.1	55%

eg, Σcount value of the Item Sets is calculated as:

Σcount {Fruit, Meat} = min (1, 1) +min (0.3, 0.3) +min (0.1, 0.1) =1.4

According to Extended Apriori Fast Algorithm, using the Σcount values given in the above table, Entity Support and Join Support of the Item sets that belong only from a single Entity table are computed and Join support of Item sets that belong to multiple tables are computed. Entity Support is calculated by counting the number of rows of Entity table that contains the itemset. Find the smallest value among the items of the itemsets in the tuple and add it with other smallest values among the items of the itemsets of other tuples in the table containing the itemset. Then divide this count by number of distinct entity keys to find the percentage of entity support. To compute join support, count number of rows that contain the itemset. Determine the values of these itemsets in the table and select the smallest value. This smallest value is then multiplied with the number of occurrences of its primary key in the transaction table. Divide this value by the no. of rows in the Transaction table. For a join itemset, compute its join support by counting the number of rows of *Join* table containing the itemset Find the smallest value among the items of the itemsets in the tuple and add it with other smallest values among the items of the itemsets of other tuples in the table containing the itemset. Then divide this count by number of rows in the table *Join*.

The table below presents the Entity and Join Support of Entity Item Sets.

e.g. Entity Support {Young, LIG} = min (1, 0.5) +min (0.5, 1) =1 =1/3*100=33.33%

No. of occurrences of Custid 1 is 2 in Transaction and for Custid 3 is 1 then:

Join Support (Young, LIG}= (0.5*2)+(0.5*1)=1.5 = 1.5/6*100 =25%

Table 4: Entity Support and join support of Entity Item Sets

2-Item Set	**Σcount**	**Entity Support**	**Join Support**
{Pork, Fruit}	0.8	13%	11%
(Pork, Vegetable}	1.5	28%	21%
{Pork, Veg. Dishes}	1.8	29%	28%
{Pork, Meat}	3.5	60%	50%
{Fruit, Vegetable}	0.8	20%	10%
{Fruit, Veg. Dishes}	3.1	50%	40%
{Fruit, Meat}	1.4	28%	22%
{Veg. Dish}	2.7	48%	44%
{Vegetable, Meat}	2.3	37%	38%
{Veg. Dish, Meat}	3.0	47%	49%
{Young, Middle}	0.7	20%	10%
{Young, LIG}	1.4	33.33%	25%

{Middle, LIG}	0.8	19%	9%

The frequent itemsets from the study are shown in table 5. The table below presents the Dsupport for frequent Item Sets. Degree of support of the frequent itemsets are calculated as:

$\sum$*count* value of {VegDishes, Meat} = 2.9

Dsupport = 2.9/6*100 = 48.3%

Table 5: Dsupport for frequent item sets

Frequent Item sets	Dsupport
{Young}	50%
{Middle}	43%
{LIG}	50%
{Pork}	50%
{Fruit}	40%
{Vegetable}	43%
{Vegetable dishes}	73%
{Meat}	70%
{Young, Meat}	41.6%
{VegDishes, Meat}	48.3%

In this section the proposed algorithm has been evaluated. The basic advantage of the new scheme is that it allows rules to be formulated using vague linguistic expressions that prevents the overestimation of boundary cases. 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. This factor helps in quick recovery of data and gives better results. The proposed algorithm finds rules that previous algorithms have not worked upon.

For the experiment, a small database of two tables has been taken as shown in Tables 1, 2 respectively. A join table is also formed on the fly during each pass of the algorithm by joining Customer C' and Transaction T' tables. In order to get association rules from such fuzzy tables, it is required to calculate Degree of Support and Degree of Confidence that should be greater than or equal to user specified minimum support and minimum confidence respectively. Degree of support of all the frequent itemsets are calculated and are shown in table 5. Degree of confidence of all 2-itemsets using the formula given as under:

For instance, *Dconfidence* (Young $\Rightarrow$ Mea t) = 42/50*100 = 82.33 which is greater than 60%. Table 6 lists those rules discovered which satisfy the given thresholds min_sup= 40% and min_conf=60%.

Table 6: The rules satisfying min-support and min_conf.

Association Rules $\mathbf{X \Rightarrow Y}$	***Dsupport*** $\mathbf{X \Rightarrow Y}$	***Dconfidence*** $\mathbf{X \Rightarrow Y}$
Young $\Rightarrow$ Meat	41.67%	82.33%
Vegetable dishes $\Rightarrow$ Meat	48.33%	65.91%
Meat $\Rightarrow$ Vegetable dishes	48.33%	69.05%

Table 6 shows final results of the study. As shown in the table rule young $\Rightarrow$ Meat implies that customer of the age group 20-30 and 30-40 might turn to buy Meat where the age group 30-40 partially belongs to Young with degree $\mu_{young30-40}$. In this example the attributes **age** and **Income** of the **customer C'** table were first converted into fuzzy taxonomic structures respectively reflecting partial belonging of one item to another given in fig 2 and 3 respectively.

Here Young and Meat both belong to two different tables which satisfy our requirement of multiple tables. Young is a subclass of its super class Age which belongs to fuzzy taxonomic structure over the attribute Age that again satisfies our requirement of fuzzy data in tables. Such rules given in table 6 are fuzzy generalized association rules for multiple tables.

To assess the relative performance of the Proposed Extended Apriori Fast, experiments has been carried out on the number of transactions varying from one table to another table of the database. Synthetic data has been generated using ARTool to evaluate the performance of the algorithms.

The characteristics of the datasets on which the experiments were performed using Extended Apriori Fast are shown in table 7. The results of the Existing algorithms Apriori Join and Apriori Star along with the results of the discovered algorithm Extended Apriori Fast is shown on the same parameters in table 8.

Table 7: Characteristics of test

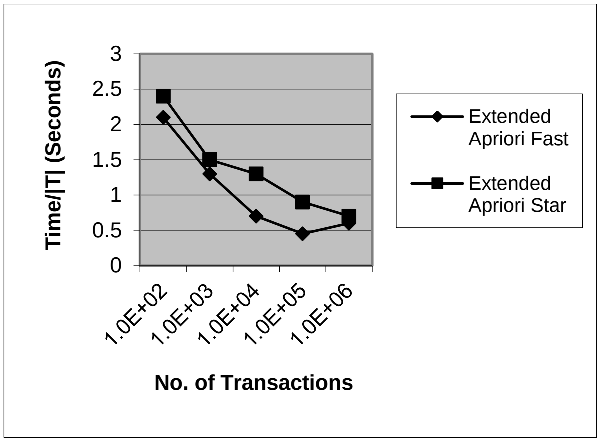
MinSup	D1	D2	D3
Extended Apriori Star [19] results			
0.5	6.7	12.9	20.0
0.4	7.9	16	31
0.3	8.1	15.37	31.33
0.2	09	21.0	43.98
0.1	23	45	78.33
Extended Apriori Fast results			
0.5	8.2	16.5	35.3
0.4	9.4	29.3	16.1
0.3	18.57	49	67.6
0.2	19	39.2	70
0.1	22.3	58	185

dataset

Characteristics	D1	D2	D3
Entity E1			
No. of Transactions	10000	30000	50000
Avg. Transaction Size	8	7	9
No. of Items	50	50	50
No. of frequent Patterns	100	100	100
Avg. pattern length	6	6	6
Entity E2			
No. of Transactions	80	170	380
Avg. Transaction Size	4	4	4
No. of Items	10	10	10
No. of frequent Patterns	8	8	8
Avg. pattern length	2	2	2
Entity E3			
No. of Transactions	1000	2000	4000
Avg. Transaction Size	6	7	7
No. of Items	30	30	30
No. frequent Patterns	55	55	54
Avg. pattern length	4	4	4

Table 8: Test Results

It has been observed that the Proposed Extended Apriori Fast algorithm scale linearly as the number of tuples of the database increases as shown in table 8. This is major advantage as number of transactions increase. This factor helps in getting better results. It is further observed that the execution times increase for all Extended Apriori star [19] and Extended Apriori Fast algorithms as the minimum support is reduced because the total number of large and candidate itemsets increase as shown in Graph 1. The performance of Proposed Extended Apriori-Star has not been upto mark as evident from graph but the reason is that Proposed Algorithm finds multi level fuzzy association rules for ER models. The justification can further be explained as; In case of Extended Apriori star, these algorithms mine multi level association rules for ER models but do not use fuzzy data in their tables. In case of Extended Apriori star, the algorithm mines multi level association rules with fuzzy data but do not use ER models. Both these drawbacks are taken into consideration in the proposed study. It is observed that the Proposed Algorithm is always a little slower than Extended Apriori star as the algorithm works on a database where all entity instances are participating to the Relation R. In such situations the proposed algorithm finds frequent join itemsets at each step and thus builds the join of the tables and also scans the entity tables in a separate step. The algorithm also computes the degree between each node and its ancestor. Thus, the algorithm takes much time than existing algorithms but the discovered algorithm takes the advantage of the knowledge already embedded in an entity relationship model regarding the relationships between the database entities. This help in generating more accurate results and gives better recovery of data. Although the discovered algorithm is little slower, but it gives more useful results than the existing algorithms.



Graph 1: Performance of EAS and EAF with diff support levels

VI. CONCLUSION

Apriori algorithm is used to identify associations between items in a set of data. This is particularly useful for making decisions based on the data. When dealing with clear-cut or "crisp" data, the traditional Apriori algorithm can be used to find strong association rules. However, when dealing with more complex, fuzzy data, an extended version of the Apriori algorithm is needed. This method can find fuzzy association rules, which help identify relationships at a higher level. To use the extended Apriori method, each leaf item in a taxonomic structure is added to a transaction set, which is then extended to create a new transaction set. In the case of fuzzy taxonomic structures, the transaction set is created by adding not only the ancestors of each leaf item, but also the extents to which those predecessors are supported by transactions.

High-level rules are often more complex and significant, but they cannot be applied to numerous tables. To address this issue, the multi-table-capable Extended Apriori Fast algorithm is used to find fuzzy association rules. This method calculates the degree of association between each leaf node and its ancestor across multiple tables, and replaces the count operation with the join and entity supports to determine common item sets.

The Extended Apriori Fast method also takes into account entity supports, which allows frequent entity item sets to be identified even if they are not frequent with respect to the relationship table. This helps to compute the correct support and confidence for rules that exist among attributes of the same entity table.

REFERENCES

1. 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.
2. 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.
3. Bakk Lucas Helm. Master Thesis on Fuzzy Association rules - An implementation in R. Vienna 2007.
4. Chen, G., and Wei, Q. (2002). Fuzzy association rules and the extending Mining Algorithms, Information Sciences: An International Journal, 147, pp.201-228.
5. 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).
6. Cristofor, L., and Simovici, D. (2001-2002). "*Mining Association Rules in Entity-Relationship Modeled Databases*", Technical Report, UMB.
7. Gottwald, SeigFried. Universes of Fuzzy Sets and Axiomatizations of Fuzzy set theory. Studia Logica volume 82, Number 2, March, Springer, 2006.
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. Houtsma, M., and Swami, A. (1993, October) Set-oriented mining of association rules. Research Report RJ 9567, IBM Almaden Research Center, San Jose, CA.
10. Kuok, C.H., Fu, A., and Wong, M.H. (1998). Mining Fuzzy association rules in databases ACM SIGMOD Record, 27(1), ACM Press.

11. 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.

12. Peter P. Wakabi-Waiswa, Venansius Baryamureeba. International Journal of Computing and ICT Research. Vol 2. No 1. June 2008.

13. 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.

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

15. P Arora, RK Chauhan, A Kush, " Association Rule Mining for Multiple Tables With Fuzzy Taxonomic structures ", International Journal of Computer Theory and Engineering 2 (6), 866

16. P Arora, P Gandhi, G Sharma, S Saxena, "Study about Rule Mining for Multiple Tables with Fuzzy Data", Novel Research Aspects in Mathematical and Computer Science Vol. 1, 38-47, 2022

17. P Arora, RK Chauhan, A Kush "Frequent Itemsets from Multiple Datasets with Fuzzy data", International Journal of Computer Theory and Engineering vol. 3, no. 2, pp. 255-260, 2011.

18. P Arora, RK Chauhan, A Kush "Mining fuzzy generalized association rules for ER models", International Journal of Information Technology and Knowledge Management (IJITKM), 2008, vol-1, issue-2, pp 191-198.

19. P Arora, RK Chauhan, A Kush "Frequent Itemsets from Multiple Datasets with Fuzzy data", International Journal of Computer Theory and Engineering vol. 3, no. 2, pp. 255-260, 2011.

20. P Arora, RK Chauhan, A Kush ;Association Rule Mining For Multiple Tables With Fuzzy Taxonomic Structures; to be published in International Journal of computer theory and Engineering(IJCTE);, vol 2, issue 6, pp 866-870, December 2010