Association Rule Mining for Multiple Tables With Fuzzy Taxonomic Structures

Praveen Arora, R. K. Chauhan and Ashwani Kush

Abstract—Most of the existing data mining algorithms handle databases consisting of single table to find association rules an large databases. Few algorithms work on multiple tables having fuzzy data with taxonomic structures. This paper proposes ‘Multi level Fuzzy rules for ER Models’ algorithm. The study focuses on the issue of mining association rules in databases having multiple levels containing fuzzy data with taxonomy and tables to be designed using Entity-Relationship (ER) Models. The study aims to incorporate the previous developed algorithms Extended Apriori and Apriori star to a new algorithm. The study will help in standardizing algorithms for finding appropriate results from database tables containing data with fuzzy taxonomic structures.

Index Terms—Association Rules, Data Mining, Fuzzy data, ER models

I. INTRODUCTION

Association Rules mining (ARM) discovers interesting and unexpected rules from large data sets. The major aim of ARM is to find the set of all subsets of items or attributes that frequently occur in many database records or transactions, and additionally, to extract rules on how a subset of items influences the presence of another subset [1]. The subdivision of the quantitative values into crisp sets would lead to over- or underestimating the values near the borders. Fuzzy sets can overcome that problem by allowing partial memberships to different sets. Fuzzy set theory provides the tools needed to do the computations in order to be able to deal with the different data structure. It allows the intervals to overlap, making the set fuzzy instead of crisp. Items can then show a partial membership to more than one set, overcoming the sharp boundary problem [2]. Fuzzy sets are generalized sets which allow for a graded membership of their elements. Usually the real unit interval is chosen as the membership degree structure [3]. The study focuses on the extraction of multi level linguistic association rules from multiple tables and examines the performance of extracted rules. 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. Using this study, rules can be discovered that might have got lost with the standard quantitative approach. Rest of the paper has been organized as, Section-2 presents recent studies wherein we present the brief of various existing association rule mining algorithms, Section-3 presents the newly discovered algorithm Extended Apriori Star that will find Generalized association rules for ER models with fuzzy data. The algorithm is applied on large database of customer transactions and Section-4 presents the implemented results. Section-5 concludes the study with an idea of future work.

II. RECENT STUDIES

The problem of mining association rules has been discussed by agrawal et al [4]. It is an AIS algorithm known to be the first published algorithm to generate all large item sets in a transaction database. It focused on the enhancement of databases with necessary functionality to process decision support queries. This algorithm was targeted to discover qualitative rules. This technique is limited to only one item in the consequent. That is, the association rules are in the form of \( X \Rightarrow I_1 | \alpha \), where \( X \) is a set of items and \( I_1 \) is a single item in the domain \( I \), and \( \alpha \) is the confidence of the rule. The AIS algorithm makes multiple passes over the entire database. Candidate itemsets are generated and counted on-the-fly as the database is scanned. 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 Houtsma [5] where an algorithm \( \text{SE TM} \) 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.
- 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 [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 APriori Hybrid. Experiments reveal that Apriori Hybrid 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 by Manila et al [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 by Han[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. Srikant et al [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. Kuok [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 as by Chen [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 by Cristofor [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 Chen [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

The Proposed study focuses on developing an algorithm that can find fuzzy generalized association rules for multiple tables which uses the existence of a hierarchical taxonomy (concept hierarchy) of the fuzzy data to generate different association rules at different levels in the taxonomy for databases containing multiple tables designed using ER Models. If traditional data mining algorithms are used to discover association rules in such environments (where either the data of the single table is used with fuzzy taxonomic structures and even if used from multiple tables then the concept of fuzzy data is not introduced) then first a join of entity tables and relationship table needs to be computed which in turn adversely affects the efficiency and cost of the algorithm used. It has been observed that very less work has been done on development of multi level fuzzy data mining association rules for multiple tables.

![Fig 1: Example of fuzzy taxonomic structure over item](image-url)
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. 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 $\mu_{\text{young, 30-40}}$. The following example finds above mentioned fuzzy generalized rule.

The study aims to extend the previous developed algorithms. The proposed algorithm Extended Apriori Star has been explained in next section.

**IV. IMPLEMENTATION**

**Algorithm Extended Apriori Star**

1) $\mu_{xy} = \bigoplus \mu \bigotimes$ // determine the degree to which leaf item belongs to its ancestor.

$$\forall l: x \to y \quad \forall eonl$$

2) **forall** leaf nodes $L_{Ni} \in \text{Taxonomy}$ **do**

   **forall** interior nodes $IN_{j} \in \text{Taxonomy}$ **do**

   $\mu_{(L_{Ni}, IN_{j})} = \max_{l \in L_{Ni}} \min_{e \in l} (\mu_{le})$

   **insert** into Transaction $T'$ // Extended Transaction set

   **values** $L_{Ni}, IN_{j}, \mu (L_{Ni}, IN_{j})$

   **endfor**

3) **forall** foreign key $f \in R$ **do**

   $f_{++}$;

   **endfor**

4) Set $K=1$; $C_{k} = 1$-Itemsets ($E_{1}, E_{2}, \ldots, E_{n}$).

5) $K_{++}$;

6) **forall** entity tables $E_{i}$ (where $i=1$ to $n$)

**forall** itemsets $I \subseteq E_{i}$

Compute $\Sigma_{\text{count}}$ // sum of all the degree that are associated with the transaction in $T$

**If** ($\Sigma_{\text{count}} \geq (\min_{\text{sup}} \times |T|)$)

- Compute Entity Support from $E_{i}$
- Compute Join support from results of step-3

$C_{k} = I$

**endif**

**endfor**;

**endif**;

1) **Frequent** $F = (\text{if c. Entity\_Sup} \parallel \text{c. join\_sup} \in C_{k} \geq \min_{\text{sup}}) // c = \text{candidates of Ck}$

   All Frequent $AF = (\text{c.E.entity\_Sup} \parallel \text{c.J.join\_sup} \in C_{k} \geq \min_{\text{sup}}) // c.E = \text{Entity Itemset}; c.J = \text{Join Itemset}$

2) $C_{k} = \text{apriori\_gen}(F, \text{min\_sup})$

3) If $C_{k} = \emptyset$ then Exit.

4) **forall** c.E $\in C_{k}$

   **forall** $L_{Ni} \in \text{E}_{i}$ (where $i=1$ to $n$) Compute Entity Support

   compute Join Support

   **endfor**;

$R \times E_{1} \times E_{2} \times \ldots \times E_{n}$ to form join table $JT$.

5) $R * E_{1} * E_{2} * \ldots * E_{n}$ to form table $JT$.

6) **forall** c.J $\in C_{k}$

   compute Join support to form table $JT$

   **endfor**

6) Go to step 5.

In this phase Extended Apriori Star 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, we should emphasis that these thresholds are context-dependant, which should be defined according to the concrete situation. Here, by a frequent item set we mean the item set whose $D_{\text{support}}$ is more than min-support threshold. Extended Apriori star algorithm is applied on Table 1 and 2 and Frequent 1-itemsets are generated that are given in Table 3. We calculate the $E_{\text{count}}$ values of the candidate itemsets and then find the Entity Support and Join Support of the Item sets that belong only from a single Entity table and calculate the Join support of Item sets that belong to multiple tables. Table 1 and Table 2 have been shown in the end of the article.

![Fig 2: Example of fuzzy taxonomic structure of Age](image2.png)

![Fig 3: Example of fuzzy taxonomic structures of Income](image3.png)
Table 3 contains all Item Sets who’s Entity Support or Join Support is greater than or equal to min_sup.

eg. \( \Sigma \text{count} \) value of the Item Sets is calculated as:

\[
\Sigma \text{count} \{\text{Fruit, Veg. Dishes}\} = \min (1, 1) + \min (0.7, 0.7) + \min (0.7, 0.7) = 2.4
\]

According to Extended Apriori Star Algorithm, using the \( \Sigma \text{count} \) values given in the above table we now find the Entity Support and Join Support of the Item sets that belong only from a single Entity table and calculate the Join support of Item sets that belong to multiple tables.

The table 4 below presents the Entity and Join Support of Entity Item Sets.

The frequent itemsets from the study are shown in table 5. The table 5 presents the Dsupport for frequent Item Sets. 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:

\[
D_{\text{confidence}}(\text{Young} \Rightarrow \text{Meat}) = \frac{41}{50} \times 100 = 82.33\% \text{ which is greater than 60\%.}
\]

Table 6 lists those rules discovered which satisfy the given thresholds min_sup= 40% and min_conf=60% , thus showing final results of the study.

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

\( D_{\text{support}} = 2.9/6*100 = 48.3\% \)

TABLE 4: ENTITY SUPPORT AND JOIN SUPPORT OF ENTITY ITEM SETS

TABLE 5: DSUPPORT FOR FREQUENT ITEM SETS

V. Results
For our experiment we took a very small database of two tables 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 we need 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, \( D_{\text{confidence}} (\text{Young} \Rightarrow \text{Meat}) = 42/50*100 = 82.33 \) which is greater than 60%.

Table 6 lists those rules discovered which satisfy the given thresholds min_sup= 40% and min_conf=60% , thus showing final results of the study.

As shown in the Table 6 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 \( \mu_{\text{young30-40}} \). 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 Figure 2 and Figure 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.

VI. Conclusion
By the Apriori algorithm crisp data can be handled for finding the association rules which are required for making the decision. With extended Apriori algorithm, which finds
fuzzy association rules helps to discover the related higher level due to the strong association rules. In extended Apriori Star algorithm each leaf item in taxonomic structures are added into the transaction set \( T \) in order to form a so-called extended transaction set \( T' \). In the case of fuzzy taxonomic structures, \( T' \) is generated by not only adding to \( T \) all the ancestors of each leaf item in fuzzy taxonomic structures, but also the degrees that the ancestors are supported by the transactions in \( T \). The rules at high levels often reflect more abstract and meaningful business semantics. But it is also not able to handle multiple tables. The Extended Apriori Star algorithm, which is able to handle multiple tables, finds the fuzzy association rules. In the Extended Apriori Star algorithm, we should compute the degree between each leaf node and its ancestor of more than one table, while we do not need to do so in the classical algorithm. The second is that we replace \( \Sigma \text{count} \) operation with \( \text{count} \) operation. And the third is that Extended Apriori Star 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.

REFERENCES


Transaction^

<table>
<thead>
<tr>
<th>TransactionNo</th>
<th>CustomerID</th>
<th>Apple</th>
<th>Carrot</th>
<th>Cabbage</th>
<th>Pork</th>
<th>Sausage</th>
<th>Fruit</th>
<th>Vegetable</th>
<th>VegDishes</th>
<th>Meat</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.7</td>
<td>0.3</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>300</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>400</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.7</td>
<td>1</td>
</tr>
<tr>
<td>500</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>600</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Customer^

<table>
<thead>
<tr>
<th>CustomerID</th>
<th>CustomerName</th>
<th>Age</th>
<th>Income</th>
<th>Young</th>
<th>Middle</th>
<th>Old</th>
<th>LIG</th>
<th>MIG</th>
<th>HIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Suneel</td>
<td>25</td>
<td>17500</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Renu</td>
<td>42</td>
<td>25000</td>
<td>0</td>
<td>0.8</td>
<td>0.2</td>
<td>0</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>3</td>
<td>Rajesh</td>
<td>35</td>
<td>10000</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>