EFFICIENT FUZZY APRIORI ASSOCIATION RULE MINING TO FIND CO-OCCURANCE RELATIONSHIP

Anubha Sharma¹, asst. Prof. Nirupama tiwari²
Department of Computer Science & Engineering/Shriram College of Engineering & Management [SRCEM] Banmore, Gwalior (MP)/India, 474003

Abstract: Data mining is sorting through data to identify patterns and establish relationships. Association rule mining is a well established method of data mining that identifies significant correlations between items in transactional data. Measures like support count, comprehensibility and interestingness, used for evaluating a rule can be thought of as different objectives of association rule mining problem. In this paper we proposed efficient fuzzy apriori association rule mining technique to find all co-occurrence relationships among data items. Our technique has three steps. Firstly, the Apriori principle which allows considerably reduced the search space with discover the frequent item set. Secondly proposed an approach for finding fuzzy sets for quantitative attributes in a database by using k-medoid clustering techniques and finally employs techniques for mining of fuzzy Apriori Associate rules. We also find fuzzy Apriori Association rule measured by Leverage. Our experimental results showed better performance than previous work.

Keywords: Data Mining, Association Rule Mining, Fuzzy Apriori Association, Fuzzy Apriori Association with Leverage.

I. INTRODUCTION

In Association, the relationship of a particular item in a data transaction on other items in the same transaction is used to predict patterns. Prediction analysis is related to regression techniques. The key idea of prediction analysis is to discover the relationship between the dependent and independent variables, the relationship between the independent variables. Sequential Pattern analysis seeks to find similar patterns in data transaction over a business period. Existing algorithms [2] and [10-11] for mining association rules are mainly worked on a binary database, termed as market basket database. On preparing the market basket database, every record of the original database is represented as a binary record where the fields are defined by a unique value of each attribute in the original database. The fields of this binary database are often termed as an item. Storing of this binary database, to be used by the rule mining algorithms, is one of the limitations of the existing algorithms. Another aspect of these algorithms is that they work in two phases. The first phase is for frequent item-set generation. Frequent item-sets are detected from all-possible item-sets by using a measure called support count (SUP) and a user-defined parameter called minimum support. The second phase generates the rules using another user-defined parameter called minimum confidence [2] and support [3-5]. In this paper we proposed efficient fuzzy apriori leverage association rule mining technique to find all co-occurrence relationships among data items.

II. LITERATURE SURVEY

A. Association Rules

Association rules are if and then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An association rule has two parts, an antecedent (if) and a consequent (then). Association rule is expressed as X=>Y, where X is the antecedent and Y is the consequent. Each association rule has two quality measurements, support and confidence. Support implies frequency of occurring patterns, and confidence means the strength of implication [1-3] and [7-10].

B. Rules Interestingness Measures

The aim of the association rules is to reveal interesting relations between data. For that reason certain are used which evaluate the level of importance of each rule. These are:

Confidence: The confidence of an association rule is the proportion of the isolates that are covered by the LHS of the rule that are also covered by the RHS. Values of confidence near value 1 are expected for an important association rule [8-11].

Leverage: The leverage of an association rule is the proportion of additional isolates covered by both the LHS and RHS above those expected if the LHS and RHS were independent of each other. Leverage takes values inside [-1, 1]. Values equal or under value 0, indicate a strong independence between LHS and RHS. On the other hand values near 1 are expected for an important association rule [1].
III. RELATED WORKS

Farah Hanna Al-Zawaidah and Yosef Hasan Ibara and Marwan Al-Abed Abu-Zanona et. al. presented a novel association rule mining approach that can efficiently discover the association rules in large databases. The proposed approach is derived from the conventional Apriori approach with features added to improve data mining performance. They developed a visualization module to provide users the useful information regarding the database to be mined and to help the user manage and understand the association rules. [1].

Praveen Arora, R. K. Chauhan and Ashwani Kush et. al. is to find 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 paper focuses on finding rules from multiple tables that contain fuzzy data and are arranged in star schema. [3].

K. Sangeetha , Dr. P. S. Periasamy , S. Prakash et. al. present the main focus of this research work is to propose an improved association rule mining algorithm to minimize the number of candidate sets while generating association rules with efficient pruning time and search space optimization. The relative association with reduced candidate item set reduces the overall execution time. The scalability of this work is measured with number of item sets used in the transaction and size of the data set. [4].

In this paper The problem of mining quantitative data from large transaction database is considered to be an important critical task. Researchers have proposed efficient algorithms for mining of frequent itemsets based on Frequent Pattern (FP) tree like structure which outperforms Apriori like algorithms by its compact structure and less generation of candidate itemsets mostly for binary data items from huge transaction database. [5].

This work presents a new foundational approach to Fuzzy Weighted Associative Classifiers where quantitative attributes are discretized to get transformed binary database. In such data base each record fully belongs to only one fuzzy set [6].

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. 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 [7].

IV. PROPOSED METHODOLOGY

In this paper we proposed efficient fuzzy apriori association rule mining technique to find all co-occurrence relationships among data items. Our technique has three steps. Firstly, the Apriori principle which allows considerably reduced the search space with discover the frequent item set. Secondly proposed an approach for finding fuzzy sets for quantitative attributes in a database by using k-mediod clustering techniques and finally employs techniques for mining of fuzzy Apriori Associate rules. And also find fuzzy Apriori Association rule measured by Leverage.

Definitions:

- **Support**
  The rule $X \Rightarrow Y$ holds with support $s\%$ of transactions in D contains $X \cup Y$. Rules that have a $s$ greater than a user-specified support is said to have minimum support.

- **Confidence**
  The rule $X \Rightarrow Y$ holds with confidence $c\%$ of the transactions in D that contain $X$ also contain $Y$. Rules that have a $c$ greater than a user-specified confidence is said to have minimum confidence.

- **Itemset**: An itemset is a set of items. A $k$-itemset is an itemset that contains $k$ number of items.

- **Frequent itemset**: This is an itemset that has minimum support.

- **Candidate set**: This is the name given to a set of itemsets that require testing to see if they fit a certain requirement [1] and [5].

A. **Discovering Frequent Itemsets Using Apriori Algorithm**

The proposed of our method is the A Priori algorithm. Our contributions are in providing novel scalable approaches for each building block. We start by counting the support of every item in the dataset and sort them in decreasing order of their frequencies. Next, we sort each transaction with respect to the frequency order of their items. We call this a horizontal sort. Furthermore, we are careful to generate the candidate itemsets in sorted order with respect to each other. We call this a down word sort. When itemsets are both horizontally and downword sorted, we call them fully sorted. As we show, generating sorted candidate itemsets (for any size $k$), both horizontally and down word, is computationally free and maintaining that sort order for all subsequent candidate and frequent itemsets requires careful implementation, but no cost in execution time. This conceptually simple sorting idea has implications for every subsequent part of the algorithm.

A.1 **Candidate Generation**
Candidate generation is the important first step in each iteration of Apriori. Typically it has not been considered a bottleneck in the algorithm and so most of the literature focuses on the support counting. However, it is worth pausing on that for a moment. Modern processors usually manage about thirty million elementary instructions per second. We devote considerable attention to improving the efficiency of candidate generation, too.

### A.2 Indexing

There is another nice consequence of generating sorted candidates in a single pass: we can efficiently build an index for retrieving them. In our implementation and in the following example, we build this index on the least frequent item of each candidate (k + 1)-itemset.

The structure is a simple two-dimensional array. Candidates of a particular size k+1 are stored in a sequential file, and this array provides information about offsetting that file. Because of the sort on the candidates, all those that begin with each item I appear contiguously. The exact location in the file of the first such candidate is given by the i
th element in the first row of the array. The ith element in the second row of the array indicates how many bytes are consumed by all (k + 1)-candidates that begin with item I.

#### B. Down Word-Sorted Apriori Algorithm

The changes that have come out of this sorting are far-reaching and have impacted every phase of the algorithm.

**Algorithm:** The proposed down word-Sorted Apriori algorithm

**INPUT:** A dataset D and a support threshold s

**OUTPUT:** All sets that appear in at least s transactions of D F is set of frequent itemsets 
C is set of candidates 

C ← U  
Scan database to count support of each item in C  
Add frequent items to F  
Sort F least-frequent-first (LFF) by support (using quick-sort)  
Output F  
for all f ∈ F, sorted LFF do  
for all g ∈ F, supp(g) ≥ supp(f), sorted LFF do  
Add {f, g} to C  
end for  
Update index for item f  
end for  
while |C| > 0 do  
{Count support}  
for all t ∈ D do  
for all i ∈ t do  
Relevant-Cans ← using index, compressed cans from file that start with I for all Compressed-Cans ∈ Relevant-Cans do  
if First k – 2 elements of Compressed-Cans are in t then  
Use compressed candidate support counting technique to update appropriate support counts  
end if  
end for  
end for  
Add frequent candidates to F  
Output F  
Clear C  
{Generate candidates}  
Start ← 0  
for 1 ≤ i ≤ |F| do  
if i = |F| OR fi is not near-equal to fi−1 then  
Create super candidate from f-start to fi−1 and update index as necessary  
Start ← i  
end if  
end for  
{Candidate pruning—not needed!}  
Clear F  
Reset hash

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters. Apriori, and improve it quite significantly by introducing what we call a down word sort.

#### B.1 Discovering Fuzzy Sets using Clustering
The proposed method uses a known clustering algorithm to find the medoids of k clusters. The whole process of automatically discovering fuzzy sets can be subdivided into four steps:

1. Transform the database to make clustering possible (the value of all the attributes has to be positive integer).
2. Find the k medoids of the transformed database using a clustering method.
3. For each quantitative attribute, fuzzy sets are constructed using the medoids.
4. Generate the associated membership functions.

We can get our membership function looking at the definition of the sets above. For the fuzzy set with mid-point \( a_{ij} \), the membership function looks as follows: If \( x \leq a_{ij} \), the membership of \( x \) is 0. Also for \( x \geq a_{ij} \), \( \mu_x \geq 0 \) because in both cases, the value lies outside the range of the fuzzy set. If \( x \) takes exactly the value of the mid-point \( a_{ij} \), the membership is 1. For all other cases, we have to use a formula in order to compute the specific membership.

Generate membership functions (triangular function):

\[
f_i(x) = \begin{cases} 
1, & \text{if } a_{ij}^+ - x \\
\frac{x - \min_j}{a_{ij}^+ - \min_j}, & \text{if } \min_j \leq x \leq a_{ij}^+ \\
\frac{\max_j - x}{\max_j - a_{ij}^+}, & \text{if } a_{ij}^- \leq x \leq \max_j \\
0, & \text{otherwise} 
\end{cases}
\]

A distinction between two types of fuzzy sets has been introduced. These two types are called equal space fuzzy sets and equal data points fuzzy sets. Equal space fuzzy sets are symmetrical and all occupy the same range in the universal set. In contrary, equal data points fuzzy sets cover a certain number of instances and thus are not symmetrical.

C. Algorithm for Fuzzy Apriori Association Rule Mining

The algorithm first searches the database and returns the complete set containing all attributes of the database. In a second step, a transformed fuzzy database is created from the original one. The user has to define the sets to which the items in the original database will be mapped. After generating the candidate itemsets, the transformed database is scanned in order to evaluate the support and after comparing the support to the predefined minimum support, the items with a too low support are deleted. The frequent itemsets \( F_k \) will be created from the candidate itemsets \( C_k \). New candidates are being generated from the old ones in a subsequent step. \( C_k \) is generated from \( C_{k-1} \) as described for the Apriori algorithm in step 1. The following pruning step deletes all itemsets of \( C_k \) if any of its subsets does not appear in \( C_{k-1} \).

D. Candidate Pruning

The candidate generation so effective is its aggressive candidate pruning. In recent, the probability that a candidate is generated is shown to be largely dependent on its best testset that is, the least frequent of its subsets. Classical A Priori has a very effective candidate generation technique because if any itemset \( c \setminus \{c_i\} \) for \( 0 \leq i \leq k \) is infrequent the candidate \( c = \{c_0, \ldots, c_i\} \) is pruned from the search space. By the A Priori Principle, the best testset is guaranteed to be included among these. However, if one routinely picks the best test set when first generating the candidate, then the pruning phase is redundant.

In our method, on the other hand, we generate a candidate from two particular subsets, \( f_k = c \setminus \{c_k\} \) and \( f_{k-1} = c \setminus \{c_{k-1}\} \). If either of these happens to be the best testset, then there is little added value in a candidate pruning phase that checks the other \( k-2 \) size \( k \) subsets of \( c \). Because of our least-frequent-first sort order, \( f_0 \) and \( f_1 \) correspond exactly to the subsets missing the most frequent items of all those in \( c \). We observed that usually either \( f_0 \) or \( f_1 \) is the best testset. Finally, the association rules are generated from the discovered frequent itemsets.

E. The Fuzzy apriori mining Associate rules are composed of two steps

1. Find all itemsets that have fuzzy support (FS<X,A>) above the user specified minimum support. These itemsets are called frequent itemsets.
2. Use the frequent itemsets to generate the desired rules. Let X and Y be frequent itemsets. We can determine if the rule X \( \Rightarrow \) Y holds by computing the fuzzy confidence FC<<X, A>,<Y, B>> and this value is larger than the user specified minimum confidence value.

\[
\text{FS}_{<X,A>} = \sum_{a \in T} \prod_{x \in a} \mu_x (a_i \in A, t_i \in X) \\
D = \{d_1, d_2, \ldots, d_k\}: \text{transactions}, D[X\setminus A] \text{ with } X \text{ and } Y \text{ is attributes and } A \text{ and } B \text{ is the corresponding fuzzy sets in } X. \\
\text{For Confidence of Rule } A \Rightarrow C \\
\text{=Support(A and C)/Support(A)}
\]
Fuzzy Apriori Association rule measured by Leverage:

\[
\text{Leverage} = \Pr(L, R) - \Pr(L) \cdot \Pr(R)
\]

In Fuzzy Form the Formula is:

\[
FL<\alpha_{\lambda},\beta_{\gamma}> = \sum_{i=1}^{n} \prod_{c \in C} \mu_{c}(c \in C, t_{i}) \cdot \sum_{a \in A} \prod_{t \in T} \mu_{a}(a \in A, t_{i}) \cdot \sum_{b \in B} \prod_{r \in R} \mu_{b}(b \in B, r_{i})
\]

The rules are discovered based on the specified threshold values for support. For each rule, the frequency counts for the LHS and RHS of each rule is given, as well as the values for confidence, lift, leverage, and conviction. Note that leverage and lift measure similar things, except that leverage measures the difference between the probability of co-occurrence of L and R as the independent probabilities of each of L and R, i.e., in other words, leverage measures the proportion of additional cases covered by both L and R above those expected if L and R were independent of each other.

V. EXPERIMENTAL RESULTS

We implement our proposed fuzzy apriori association rule mining technique using JAVA platform in which jdk1.6 version is used and NETBEANS IDE6.9 is used for the graphics and analysis. To evaluate our proposed mechanism for efficient association rule mining we first find the fuzzy apriori association rule, after then we measured fuzzy association rule using leverage. Table 2 represent the fuzzy apriori association leverage rule with support, confidence, frequent set, generation time, fuzzy apriori of rules with leverage, total fuzzy apriori of nodes created, total support value increments, and total storage (Bytes) on completion after transaction of same data set used for table1.

<table>
<thead>
<tr>
<th>support</th>
<th>45</th>
<th>55</th>
<th>85</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>confidence</td>
<td>35</td>
<td>25</td>
<td>45</td>
<td>30</td>
</tr>
<tr>
<td>Frequent Sets</td>
<td>185</td>
<td>65</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>Generation Time</td>
<td>0.11 sec</td>
<td>0.05 sec</td>
<td>0.05 sec</td>
<td>0.03 sec</td>
</tr>
<tr>
<td># of Rules</td>
<td>785</td>
<td>244</td>
<td>9</td>
<td>76</td>
</tr>
<tr>
<td>Total # of Nodes Created</td>
<td>387</td>
<td>205</td>
<td>129</td>
<td>139</td>
</tr>
</tbody>
</table>

Table 2: Fuzzy Apriori Association Rule with Leverage

Table 3 represents the comparison association rule value of previously proposed FARMA, and our proposed Fuzzy Apriori and Fuzzy Apriori with leverage techniques.

<table>
<thead>
<tr>
<th>support</th>
<th>20</th>
<th>30</th>
<th>70</th>
<th>80</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>confidence</td>
<td>80</td>
<td>70</td>
<td>10</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>FARMA</td>
<td>18706</td>
<td>18592</td>
<td>778</td>
<td>130</td>
<td>7784</td>
</tr>
<tr>
<td>Fuzzy Apriori</td>
<td>3150</td>
<td>1500</td>
<td>80</td>
<td>60</td>
<td>418</td>
</tr>
<tr>
<td>Fuzzy Apriori Leverage</td>
<td>3124</td>
<td>1481</td>
<td>45</td>
<td>50</td>
<td>402</td>
</tr>
</tbody>
</table>

Table 3: Comparison Association Rule of previous work and proposed Fuzzy Apriori and Fuzzy Apriori with leverage

Figure 1 represents the comparison association rules graph of previously proposed FARMA, and our proposed Fuzzy Apriori and Fuzzy Apriori with leverage techniques. With respect to all the support values (20, 30, 50, 70, and 80) our proposed Fuzzy Apriori and Fuzzy Apriori with leverage techniques shows better performance than FARMA [1].
VI. CONCLUSION AND FUTURE WORKS

Discovering association rules is at the heart of data mining. Mining for association rules between items in large database of sales transactions has been recognized as an important area of database research. In this paper we proposed efficient fuzzy apriori association rule mining technique to find all co-occurrence relationships among data items. Our technique has three steps. Firstly, the Apriori principle which allows considerably reduced the search space with discover the frequent item set. Secondly proposed an approach for finding fuzzy sets for quantitative attributes in a database by using clustering techniques and finally employs techniques for mining of fuzzy Apriori Associate rules and also find fuzzy Apriori Association rule measured by Leverage. We compare association rules of previously proposed FARMA with our proposed Fuzzy Apriori and Fuzzy Apriori leverage techniques. The work presented in the paper can be extended for multi-level association rule mining and the work also can be enhanced to generate multi-dimensional association rules.

ACKNOWLEDGEMENTS

We wish to thank Dr. Anil Kishore Saxena (Director) & Prof. Manoj Ramiya (HOD of CSE), Shriram College of Engineering & Management, Gwalior, M.P., India, for his insightful comments and suggestions.

REFERENCES