Generating associations rule mining using Apriori and FPGrowth Algorithms

J.Suresh[1], P.Rushyanth[2], Ch.Trinath[3]
[1] *M. Tech Student of CSE dept,*
KL University, Vijayawada,
[2] *M.Tech Student of CSE dept,*
KL University, Vijayawada,
[3] *M. Tech Student of CSE dept,*
KL University, Vijayawada,

***Abstract*- Mining frequent itemsets from the large transactional database is one of the most challenging problems in data mining. In many real world scenarios, the data is not extracted from single data source but from distributed and heterogeneous ones. The discovered knowledge is expected to better business operations. In data mining methods association rule mining is one of the most popular one. However, mining informative patterns using association rules often results in a very large no of founding patterns, leaving the analyst with the task to go through all the rules and discover interesting ones In this paper we present a generating associations rules using Apriori and FPGrowth algorithms.**

***Index Terms*- heterogeneous data, discovered knowledge, complex data, associations, frequent items, informative patterns.**

I. INTRODUCTION

Now a day's all enterprise data mining applications are depend upon the data mining application, such as mining public service data and telecom services activities. All categories involve complex data sources, these are available in the form of multiple large scale, distributed, and heterogeneous data sources changing information about business transactions, business impact and user preferences. In these situations, all business people expect the discovered knowledge to present a full picture of business settings rather than one view based on a single source. It is challenging to mine for comprehensive and informative knowledge in such complex data suited to real-life decision needs by using the existing methods.

Association Rule Mining applying to finds application in market basket analysis. The market analysts would be interested in identifying frequently purchased items. Here we have two strategies for measures the process of association rule mining, those are confidence and support. The confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern. The support of an association rule is the percentage of task-relevant data transactions for which the pattern is true.

The origin analysis of the marketing bucket is an exploration of association rules represents one of the main applications in data mining. Their popularity is based on an efficient data processing by means of algorithms. Being given a set of transactions of the clients, the purpose of the association rules is to find correlations between the sold articles. Knowing the associations between the offered products and services helps those who have to take decisions to implement successful marketing techniques. By means of the Rapid Miner application we design several processes which generate frequent item sets, on the basis of which were then generated association rules. This paper includes two processes, the first one uses the Apriori algorithm and the second one uses the two algorithms *FPGrowth* and *Create Association Rules.*

This paper is organized in the fallowing scenario: in section 2 we present Apriori algorithm, in section 3 we present the FP-Growth algorithm, in sections 4 we present two process developed for generating association rules and in section 5 we present conclusions of the research.

II. APRIORI ALGORITHM

The Apriori Algorithm [4] is an influential algorithm for mining frequent item sets for Boolean association rules. Apriori is the best algorithm for mine association rules. It uses a breadth-first search technique counting the support of item sets and uses a candidate generation function which exploits the downward closure property of support. An apriori uses a bottom up approach method, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. This algorithm

to generate all frequent item sets and confident association rules, which was given together with the introduction of this mining problem. The property of Apriori is Any subset of frequent item set must be frequent.i.e., if {*AB*} is a frequent item set, both {*A*} and {*B*} should be a frequent item set Iteratively find frequent item sets with cardinality from 1 to *k (k*-item set*)* Use the frequent item sets to generate association rules. The algorithm terminates when no further successful extensions are found. The main property of this algorithm is all non-empty subsets of a frequent item set must also be frequent. the process of to find the frequent patterns by using an Apriori Algorithm is shown in below.

1. Scan all the transaction. Create possible itemsets.

2. Let the Hash table of size 8.

3. For each bucket assign a candidate pairs using the ASCII values of the itemsets.

4. Each bucket in the hash table has a count, which is increased by 1 each item an item set is hashed to that bucket.

5. If the bucket count is equal or above the min support count, then the bit vector is set to 1 else it is set to 0.

6. The candidate pairs that hash to locations where the bit vector bit is not set are removed.

7. Modify the transaction database to include only these candidate pairs.

Pseudo-code

Input D, is a database of transactions Min_sup, the Min threshold support

Output Lk Maximal frequent itemsets in D Ck Set of Candidate k-itemsets.

Method:

1. L1 =Frequent items of length 1.

2. For (k=1; Lk! =ϕ; k++) do.

3. Ck+1=candidates generated from Lk.

4. For each transaction t in database D do.

5. Increment the count of all candidates in Ck+1 that are contained in t.

6. Lk+1 =candidates in Ck+1 with minimum support

7. end do

8. Return the set Lk as the set of all possible frequent itemsets

The main notation for association rule mining that is used in Apriori algorithm is the following.

1) A k –itemset is a set of k items.

2) The set Ck is a set of candidate k-itemsets that are potentially frequent.

3) The set Lk is a subset of Ck and the set of k-itemsets that are frequent.

For generating associations in this algorithm, we find the Support and Confidence of the transaction dataset. An example of association rule is Contains (T,"Milk") → Contains (T, "Bread") [Support= 4%, Confidence=40%]

The interpretation of the rule as follows:

• 40% of transactions that contains Milk also contains Bread;

• 4% of all transactions contain both of these items.

The calculations of the Support(S) and Confidence(C) are very simple in Apriori algorithm as follows:

1. $\text{conf}(X \Rightarrow Y) = \frac{Sup(X \cup Y)}{Sup(X)}$.
2. $\text{sup}(X \Rightarrow Y) = \frac{Sup\ (X \cup Y)}{Count\ of\ items\ (X)}$.

Let's assume Data set D={(1,2,3),(2,3,4),(1,2,4),(1,2,5),(1,3,5)},From this data set we find the Support(S),Confidence(C) for itemset (1,2) is

Now Support(S) = $\frac{Matched\ Item\ sets\ (1,2)}{Total\ No.of\ itemsets} = \frac{3}{5} = 60\%$.

Now find Confidence(C) = $\frac{Sup(1 \cup 2)}{Sup(1)} = \frac{3}{4} = 75\%$.

III. FP-GROWTH ALGORITHM

The FP-Growth Algorithm [6] is an alternative way to find frequent itemsets without using candidate key generations, thus improving performance. For so much it uses a divide-and-conquer strategy. Here we store the data base in the primary storage and to calculate the support of all generated sets of patterns. The FP/Growth algorithm uses a combination between the horizontal model and the vertical model of a database. In this all elements are saved in tree structure and each element has a pointer attached to all transactions containing it. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the itemset association information.

In simple words, this algorithm works as follows: first it compresses the input database creating an FP-tree instance to find frequent items. After the first step it divides the compressed database into a set of databases, each one associated with one frequent pattern. After that each database is mined separately. Using this approach, the FP-Growth reduces the search costs looking for short patterns recursively and then concatenating then in the long frequent patterns. In large databases, it's not possible to hold the FP-tree in main memory. This approach cope with this problem is to firstly partition the database into a set of smaller databases. Then after construct an FP-tree from each of these smaller databases.

The following section describes the algorithm for FP-Growth algorithm for generating associations.

Input: D is a transactional DB, user-specified threshold and min sup,;

Output: L, ALL frequent itemsets in D

1 L1=find frequent 1-itemsets (D);

2 **for** k = 2; Lk−1 6= □; k + + do

3 Ck = apriori gen (Lk−1);

4 **for each** transaction t in D do

5 Ct = subset (Ck, t);

6 **for each** candidate c ∈Ct do

7 c.count + +;

8 **end**

9 **end**

10 Lk = {c ∈Ck |c.count ≥ min sup};

11 end

12 **return** L = ∪kLk ;

Process for Construction of FP-Tree

I. Scan database once and find out frequent items F. Sort F in support count descending order L.

II. Create the root of FP-Tree and label it as null.

III. Scan database again. For each transaction, select and sort frequent items in L-order. Create a branch in the tree if there is no common prefix in the path of the tree. The counting is performed for the items in the transaction along the path of the tree.

IV. An item header table is used to record the occurrences of items via a chain.

The formal statement of association rule mining is Let I=I1, I2, …, In be a set of n different attributes, T is a transaction that holds a set of items such that T ⊆ I, D be a database with different transaction records Ts. An association rule is an consequence in the form of A→B, where A, B ⊂ I are sets of items called item sets, and A ∩ B = Ø. A is called originator while B is called resultant, the rule means A implies B.

Example: Let we take the transaction database, DB, be (the first two columns of) Table 1 and the minimum support threshold be 3.

Transaction ID	Items Bought	(Ordered) Frequent Items
100	f; a; c; d; p; i; m; g	f; c; a; m; p
200	a; b; c; f; l; m; o	f; c; a; b; m
300	b; f; h; j; o	f; b
400	b; c; k; s; p	c; b; p
500	a; f; c; n; l; p; m; e	f; c; a; m; p

Table 1: The Transaction data set

A compact data structure is designed based on the following observations.

1. Since only the frequent items will play the role in the frequent pattern mining, it is necessary to perform one scan of DB to identify the set of frequent items.
2. If we store set of frequent items of each transaction in some compact structure, it may avoid repeated scanning of DB.
3. If multiple transactions share an identical frequent item sets, then, they can be merged into one with the number of occurrences registered as count. It is easy to check if two sets are identical in all the frequent items in different transactions are sorted according to a fixed order.
4. If two transactions share a common prefix value, according to some sorted order of frequent items, Using prefix structure we can merge the shared parts as long as the count is register properly. If the frequent items are sorted in their frequency of descending order, then there are better chances to share more prefix strings.

With these observations, we can construct a frequent pattern tree as follows.

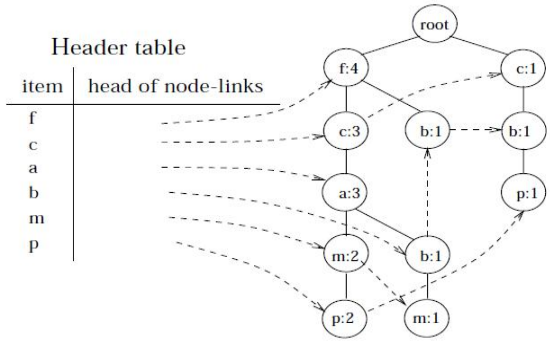

Figure: The FP-tree built based on the data in Table

Here we scan DB it derives a list of frequent items, {(f : 4);(c : 4);(a : 3);(b : 3);(m : 3);(p : 3)}, (the number after ":" indicates the support), and with items ordered keep in descending order. This ordering is most important because each path of a tree will follow this order. The frequent items in each transaction are listed ordering in the rightmost column of Table . Now we scan all transactions individually and create tree, it is shown in above figure.

IV. PROCESSES GENERATING ASSOCIATIONS USING APRIORI AND FPGROWTH ALGORITHM

The initial process uses the *Apriori* algorithm to find the frequent patterns and also generate association rules based on the frequent sets discovered. The process is presented in figure a[7].

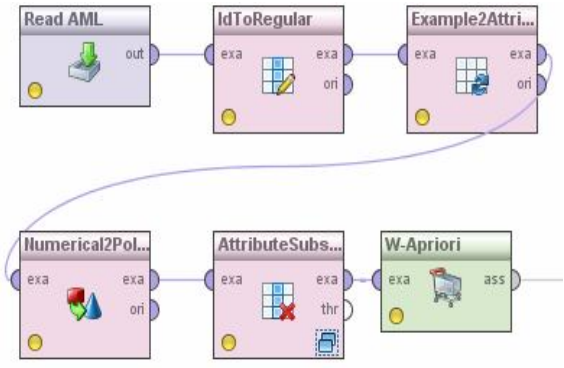

Fig. a. Generating association rules by using the *W-Apriori* Algorithm

The next process uses the *FP-Growth* algorithm to determine the frequent item sets. the frequent item sets discovered based on the *Create Association Rules* algorithm to generate association rules. The same data sets was used in apriori process shown in figure a, namely the same values for minimum support and confidence. The process is presented in figure b [7].

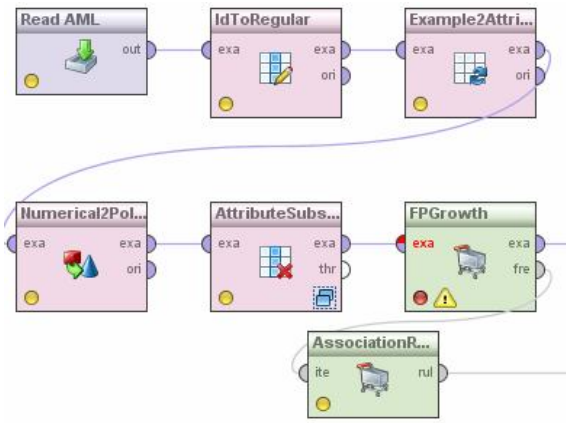

Fig. b. Generating association rules using the *FP-Growth* and the *Association Rules* algorithms

The frequent item sets are generated by using the FPGrowth algorithm. This algorithm calculates all frequent item sets in the given database, and constructs a FP-Tree structure from transactions of a database. The FP-Tree structure is used for compressing data which are stored in the memory. All frequent sets of items are obtained from this structure.

A major advantage of the algorithm *FP Growth* algorithm compared with others of the same type algorithm is it can be applied to larger data sets and it uses only two scans of the data. The frequent sets are searched for positive entries from the data base. The entry data set must contain bionominal attributes. If the data contains other types of attributes preprocessing operators must be used to transform the data set. The necessary operators are the transformation operator which change the type of values from numerical attributes into nominal attributes and then from nominal attributes into binominal attributes.

Association Rule mining process

Association rules mining is a technique for finding interesting rules from the transactional databases [1]. An association rule is an expression of the form $A \rightarrow B$, where A and B are subsets of the set of items. Here A is referred as the set of antecedents and B is referred to as the set of consequents. The subsets A and B are disjoint. The importance of a rule is evaluated by its support and confidence. The support of a rule is the fraction of all transactions where the set of antecedents A and the set of consequents B apply simultaneously. The support of a rule is a measure the number of transactions in data set. The higher the support, the more important the rule is. The confidence of a rule is the fraction of transactions containing the set of antecedents A which also contain the set of consequents B. A minimum support and confidence thresholds are usually pre-specified before mining for association rules. The following are process for generating association rules.

1. Find all frequent itemsets:
   •Each support S of these frequent itemsets is at least equal to a pre-determined min_sup.
2. Generate the strong association rules from frequent itemsets:

   •These rules must be frequent item sets and must satisfy min_sup and min_conf.

V. CONCLUSION

In data mining applications an association rules play a major role to find interesting patterns in complex data sets. The frequent sets of articles must be previously generated in order to obtain these association rules. Here two most common algorithms are used for this type of actions are the Apriori and FP-Growth. An apriori which generate both frequent sets and association rules and FP-Growth generates frequent sets of articles, which are then used by Create Association Rules to generate association rules. Although the Apriori algorithm processes data in a different manner from the algorithms *FPGrowth* and *Create Association Rules*, eliminating the sets of articles which are not frequent.

REFERENCES

[1] T. Imielinski,R. Agrawal and T. Swami, Mining association rules between sets of items in large databases. In *Proc., ACM SIGMOD Conf. On Manag. of Data*, pages 207–216, Washington, D.C,1993.

[2] Craus M., Archip A., *A Generalized Parallel Algorithm for Frequent Itemset Mining* and Proceedings of the 12th WSEAS International Conference on Computers, Greece ,Heraklion, 2008, pg. 520-523.

[3] R. Srikant and R. Agrawal, Mining Generalized Association rules, 1999, pg. 407–419.

[4] Daniel Hunyadi, *Improvements of Apriori Algorithms*, First International Conference on Modeling and Development of Intelligent Systems – MDIS, October 22-25, 2009, Sibiu, Romania, "Lucian Blaga" University Publishing House.

[5] R. Srikant and R. Agrawal. Fast algorithms for mining association rules in largest databases. In J.Bocca, C. Zaniolo and M. Jarke, editors, *Proc. Int. Conf. on Very Large Data Bases*, pages 478–499, Santiago, Chile, 1994.

[6] Han J., Pei J., Mao R and Yin Y., *Mining frequent patterns without candidate generation: A frequent-pattern tree approach*, data mining and knowledge discovery, 2003.

[7] Daniel Hunyadi, Performance comparison of Apriori and FP-Growth algorithms in generating association rules, Proceedings of the European Computing Conference "Lucian Blaga" University Publishing House.

[8] H. Zhang, Y. Zhao, C. Zhang and L. Cao, "Combined Association Rule Mining," in *Proc. PAKDD*, 2008, pp. 1069–1074.

[9] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, Q. Chen ,H. Pinto, U. Dayal, and M.-C. Hsu, "Mining sequential patterns by pattern-growth: The Prefix Span Approach," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1424–1440, Nov. 2004.

[10] Y. Zhao, H. Zhang, C. Zhang ,L. Cao, and H. Bohlscheid, "Combined Pattern Mining: From Learned Rules to Actionable Knowledge," in *Proc. AI*, 2008, pp. 393–403.

BIOGRAPHY

Jogi.suresh received the B.Tech degree in Information Technology from JNTU affiliated College. He is currently pursuing M.Tech. in Computer Science and Engineering from K.L.University,Vijayawada.

P.Rushyanth received the B.Tech. degree in Information Technology from JNTU affiliated College. He is currently pursuing M.Tech In Computer Science and Engineering from K.L.University, Vijayawada.

Ch Trinath received the B.Tech degree in Computer Science and Engineering from JNTU affiliated College. He is currently pursuing M.Tech. in Computer Science and Engineering from K.L.University,Vijayawada.