Association Rules and Mining Frequent Itemsets using Algorithms

Sukhvir Singh
Asst. Professor, Department of Computer Science & Engineering
NC College of Engineering
Israna, Panipat
boora_s@yahoo.com

Jagdeep Singh
Research scholar Department of Computer Science & Engineering
NC College of Engineering
Israna, Panipat
jagga_6202@yahoo.co.in

Abstract—**Association rules are the main technique for data mining. Association rule mining is to find association relationships among large data sets. Mining frequent patterns is an important aspect in association rule mining. In this paper,Apriori algorithm is analyzed, which is classic one in the Association Rules Algorithms and summarizes problems existing in the algorithm. Analyzing of the frequent itemsets problem for Association Rules in data mining**.

Keywords-Data mining, Apriori Algorithm, Association rules, Frequent itemsets.

1. INTRODUCTION

Data mining based on association rules can be divided into two parts: finding all frequent itemsets, and generating reliable association rules from all frequent itemsets. Frequent itemsets mining plays an essential role in association rules mining. It is a very time-consuming procedure. Frequent itemset and association rules mining problems have attracted increasing research interest. Association rule mining has received considerable attention from database practitioners and researchers because of its applicability in many areas. At present, the most important association rule mining algorithm is Apriori[1,2] put forward by R.Agrawal, it is a classical algorithm for mining the frequent itemsets. One of the most important algorithms is mining association rules, which was first introduced in [3, 4]. Association rule mining has many important applications in our life. An association rule is of the form X => Y. And each rule has two measurements: support and confidence.Many algorithms for mining association rules from transactions database have been proposed [5, 6, 7] since Apriori algorithm was first presented.

2. ASSOCIATION RULE MINING

Association rule mining finds interesting associations and/or correlation relationships among large set of data items. Association rules show attribute value conditions that occur frequently together in a given dataset.[3] A set of items is defined as an itemset and represents items found in a dataset.The associations that Apriori finds are called *Association rules*. An association rule has two parts. The *Antecedent* is a subset of items found in sets of data. The *Consequent* is an item that is found in combination with the antecedent. Two terms describe the significance of the association rule. The *Confidence* is a percentage of data sets that contain the antecedent. The *Support* is a percentage of the data sets with the antecedent that also contain the consequent.

A. ASSOCIATION RULES

The concept of the association rules was first proposed by R.Agrawal.It is used to describe the patterns of customers' purchase in the supermarket.[4]The association rules can be formally defined as

Definition 1: Let $I=\{i1,i2,i3,\ldots.in\}$be finite itemsets. D is a transactional database. Where ik(k {1,2,…...m})is an item, and *Tid* is the exclusive identifier of transaction T in transactional database.

Definition 2: Let $X\subset I, Y\subset I$, and $X\cap Y=\varnothing$.The implication of the form $X=>Y$ is called an association rules. Where $X\subset I, Y\subset I$ are purchases patterns of customer.

Definition 3:Let D is a transactional database.If the percentage of transactions in D that contain X U Y is $s\%$,the rule $X=>Y$ holds in D with Support s.If the percentage of transactions in D containing X that also contain Y is $c\%$,the rule $X=>Y$ has Confidence c. The definitions of probability are,

$$\text{Support}(X => Y) = P(X \cup Y) \quad (1)$$

$$\text{Confidence}(X => Y) = P(Y|X) \quad (2)$$

Rules that satisfy both minimum support threshold (*minsup*) and minimum confidence threshold (*minconf*) are called strong rules.
Definition 4: If the support of itemsets *X* is greater than or equal to minimum support threshold, *X* is called frequent itemsets.If the support of itemsets *X* is smaller than the minimum support threshold,*X* is called in frequent itemsets.

B. RELATED PROPERTIES
According to definition 4, we can get the following properties:

- Property 1: Let $Y \subseteq I$ and $X \subseteq Y$ .If *Y* is frequent iternsets,*X* is also frequent itemsets.
- Property 2: Let $Y \subseteq I$ and $X \subseteq Y$.If the *X* is infrequent itemsets,*Y* is also infrequent itemsets.

The correctness of property 1 and property 2 can be gained easily.These properties can be used to reduce searching width of mining association rules.

3. APRIORI ALGORITHM

Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k − 1. Then it prunes the candidates which have an infrequent sub pattern.The algorithm terminates when frequent itemsets can not be extended any more.But it has to generate a large amount of candidate itemsets and scans the data set as many times as the length of the longest frequent itemsets.
Classical Apriori algorithm:

(1) C1 = {candidate 1-itemsets};

(2) L1 = { c C1 |c.count≥minsup}};

(3) for (k=2; Lk−1≠Ø; k++) do begin
(4) Ck=apriori-gen(Lk-1);
(5) for all transactions t D do begin
(6) Ct=subset(Ck,t);
(7) for all candidates c Ct do
(8) c.count++;
(9) end
(10) Lk={c Ck |c.count≥minsup}
(12) end
(13) Answer= Lk;

ILLUSTRATION OF ALGORITHM

In the original decision Table 1, U is the concerned universe, a, b, c, d are condition attributes, e is decision attribute. [3]The least support is minsup=1. Reduce the decision table according to the algorithm presented in the above section:
Attribute reduction. Only attribute a is e-omissible in the original table, so delete attribute a to form a new decision table.
Value reduction. Calculate attribute support and rule support of every attribute to form candidate set C1 with 1 condition attribute.
Check every item in the new table, if rule support of an item is less than or equal to the least support minsup, delete the item from C1; if attribute support of an item is equal to rule support, transfer the item to rule set R as determinate rule. Form candidate set C2 with 2 condition attributes. Combine two items that have the same decision attribute in C1 to form new item with 2 condition attributes by extending C1. Then deal with them according to the above method. Form candidate set C3 with 3 condition attributes. When candidate set C4 with 4 condition attributes is empty, stop the algorithm. At last we get the reduced decision Table 2 yielding the following rules:

(1) $(d,1) \Rightarrow (e,3)$

(2) $(b,2) \wedge (c,2) \Rightarrow (e,3)$

(3) $(b,1) \wedge (c,1) \Rightarrow (e,3)$

(4) $(b,2) \wedge (c,1) \wedge (d,2) \Rightarrow (e,2)$

(5) (b,1)∧ (c,2) ∧ (d,2) ⇒(e,1)

TABLE 1 THE ORIGINAL DECISION TABLE

U	a	b	c	d	e
1	1	2	1	1	3
2	1	2	2	1	3
3	1	2	1	2	2
4	1	1	1	1	3
5	2	1	2	2	1
6	2	2	2	1	3
7	2	2	2	2	3
8	2	1	2	1	3
9	2	1	1	1	3
10	3	2	2	2	3
11	3	2	1	2	2
12	3	1	1	2	3
13	3	1	1	1	3
14	3	2	2	1	3
15	3	1	2	2	1
16	3	1	2	1	3

TABLE 2 THE REDUCED DECISION TABLE

U	b	c	d	e	Rule Support
1	--	--	1	3	9
2	2	2	--	3	5
3	1	1	--	3	4
4	2	1	2	2	2
5	1	2	2	1	2

4. FP-GROWTH ALGORITHM

In [9], Han, Pei et al. proprosed a data structure
called FP-tree (frequent pattern tree). FP-tree is a highly compact representation of all relevant frequency information in the data set. Every path of FP-tree represents a frequent itemset and the nodes in the path are stored in decreasing order of the frequency of the corresponding items. A great advantage of FP-tree is that overlapping itemsets share the same prefix path. So the information of the data set is greatly compressed. It only needs to scan the data set twice and no candidate itemsets are required.
An FP-tree has a header table. The nodes in the header table link to the same nodes in its FP-tree. Single items and their counts are stored in the header table by decreasing order of their counts. Fig.1a shows an example of a data set while Fig.1b shows the FPtree constructed by that data set with minsup = 30%.

Transaction
a b c
a c e f
d f
a b c
a c e g
b c

Fig 1a.A Data set

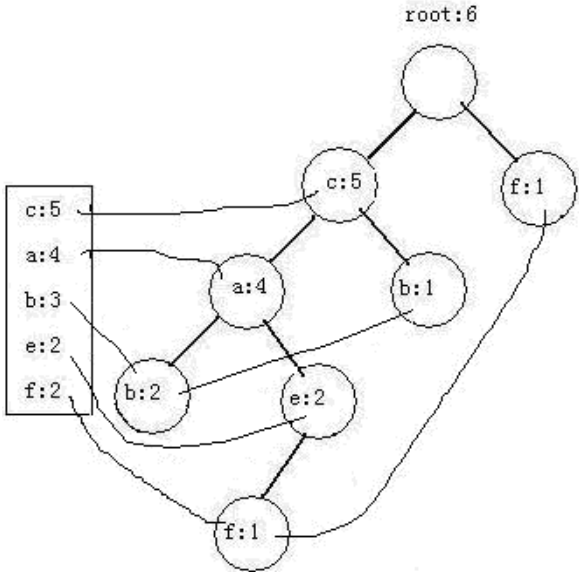

Fig.1b. FP-tree Constructed by the above data set.

The disadvantage of FP-Growth is that it needs to work out conditional pattern bases and build conditional FP-tree recursively. It performs badly in data sets of long patterns.

5. CONCLUSIONS AND FUTURE WORK

This paper, present a method that only scans the data set twice and builds FP-tree once using Apriori algorithm while it still needs to generate candidate itemsets. The future work is to further improve the Apriori- Algorithm and test more and larger datasets.

6. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami.“Mining associcion rules between sets of items in large databases”. In SIGMOD Conference, pages 207-216, 1993.

[2] R. Agrawal, R. Srikant, “Fast algorithms for mining association rules”, Proceedings of the 20th Very Large DataBases Conference (VLDB’94), Santiago de Chile, Chile,1994, pp. 484-48

[3] Yuliang Ma,Zhizeng Luo,“Value Reduction in rough sets based on Apriori algorithm”National Hi-tech Research and development Program of China,2008

[4] Wanjun Yu,Xiaochun Wang, “The Research Of Inproved apriori Algorthim for Mininh Asociation Rules”IEEE,2008

[5] Agrawal, R., Srikant, R., & Vu, Q, “Mining association rules with item constraints”, In The third international conference on knowledge discovery in databases and data mining, Newport Beach, California, 1997, pp. 67-73.

[6] J.Han, Y. Fu, “Discovery of multiple-level association rules from large database”, In The twenty-first international conference on very large data bases, Zurich, Switzerland, 1995, pp. 420-431.

[7] Fukuda, T., Morimoto, Y., Morishita, S., & Tokuyama, T., “Mining optimized association rules for numeric attributes”, In The ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, 1996, pp. 182-191.

[8] J.Han, J.Pei and Y.Yin., “Mining frequent patterns without candidate Generation”, in: Proceeding of ACM SIGMOD International Conference Management of Data,2000, pp. 1-12.