Mining Frequent Itemset Based on Sorting Technique

Madhu Bala Babel[1], B.L. Pal[2]

[1]*M.Tech Scholar, Dept. of CSE, Mewar University, Gangrar, Chittorgarh (Raj.)*
[2]*Asst. Professor, Dept. of CSE, Mewar University, Gangrar, Chittorgarh (Raj.)*

***Abstract*- Mining frequent itemset is a very important concept in Data Mining. Association rules are used for analysing costumer behavior and the relationship among the data items. Generally market basket analysis uses the Association rules to identify the costumer purchasing behavior. Apriori algorithm and Fp-Tree algorithms are the two most important algorithm for mining frequent itemsets. This paper presents a new and effective algorithm for mining frequent itemsets, which uses the sorting technique. This algorithm reduces the full scan of the whole database for finding the frequent itemset.**

***Keywords: Association rule, Data mining, Frequent itemset, minimum support, Support, Subset*.**

I.INTRODUCTION

Association rule mining is a problem in data mining. The main purpose of mining association rule is to findout the association among the data items and to analyse the costumer purchasing behavior. All the itemsets that appear more than or equal to minimum support are the frequent itemset. Mining frequent itemset problem is used by Market basket application in which it is analysed that which kind of products are purchased together by costumer.

Database representation for Itemset:

The database of item sets can be stored in memory by following 3 ways:

- Horizontal representation: In this representation each row contains the transaction identifier followed by data items that are present in that transaction.[1]

Table1. Horizontal Database representation [4]

Transaction Identifier	List of items
T1	A,B
T2	C,D,E
T3	A,B,CF
T4	A,C,D
T5	A,D,F
T6	B,E,I
T7	A,B,D,E
T8	D,E,F,I

- Vertical representation: In this representation each column contains the data item followed by transaction identifier in which the data items is present.[1]

Table 2. Vertical Database representation [4]

List of items						
A	B	C	D	E	F	I
T1	T1	T2	T2	T2	T3	T6
T3	T3	T3	T4	T6	T5	T8
T4	T6	T4	T5	T7	T8	
T5	T7		T7	T8		
T7			T8			

- Bit vector representation: In this representation the database is represented in a 2-dimension table of bit vector. In this row represents the transaction identifier and the column represents the data items. When the item presents in the transaction then the corresponding value is 1, otherwise the value is 0.[1]

Table 3. Bit Vector Database representation [1]

Transaction Identifier	Items						
	A	B	C	D	E	F	I
T1	1	1	0	0	0	0	0
T3	1	1	1	0	0	1	0
T7	1	1	0	1	1	0	0
T4	1	0	1	1	0	0	0
T5	1	0	0	1	0	1	0
T6	0	1	0	0	1	0	1
T2	0	0	1	1	1	0	0
T8	0	0	0	1	1	1	1

II. BASIC CONCEPT

Let I={i1,i2,i3…} is a set of items. Let DB is a database containing all transactions. Each transaction (TI) contains some items, means $TI \subseteq I$ .A unique

identifier is associated with each transaction, which uniquely identify that transaction. Let X and Y are two itemsets. Then the association rule is of the form $X \rightarrow Y$, where $X \cap Y = \Phi$. Frequent itemset is searched for association rule. But in this paper the major concern is to search frequent itemsets.[2]
Freq(X) is the number of transaction in database DB that contains the itemset X. It is also called the support for the itemset X. An itemset is frequent if its support is greater than or equal to minimum support(S), that is specified by the user.

Mining frequent itemset:

Mining frequent itemset plays important role in the field of data mining. Support of each transaction is counted from database (DB) to find the frequent itemset. Mining frequent itemset problem is motivated by Market-Basket application, where each tuple in transaction database (DB) is a transaction purchased by costumer. There are three basic technique for generating frequent itemset.[3]

- Horizontal layout based mining techniques
  - Apriori algorithm
  - DHP algorithm
  - Partition
  - Sample
  - A new improved Apriori algorithm

- Vertical layout based mining techniques
  - Eclat algorithm

- Projected database based mining techniques
  - FP-tree algorithm
  - H-mine algorithm

III. PROPOSED APPROACH

This new approach is based on the concept of sorting the database according to data items. This approach reduces the full scan of whole database each time when finding an itemset is frequent or not.

This approach uses the both horizontal and bit vector representation of database. In this proposed approach initially horizontally database representation is sorted according to data items. Then the bit vector representation of database is created. After that it is counted, how many times an item is presented in all transaction. The items whose count is greater than or equal to minimum support combinedly make a maximal frequent itemset. And discard those items whose count is smaller than minimum support.

After finding the maximal frequent itemset, all subsets of maximal frequent itemset containing two or more items is created. Then each subset is searched in the horizontal database representation to find it is frequent or not. If the number of times the itemset came in horizontal representation is less than minimum support than the itemset is not frequent. And because of the horizontal representation is sorted according to items the searching of subset in database is stopped when a higher order item is found in database. Because there is no possibility of finding the subset after that so this algorithm reduces to scan of full database.

If an itemset is not frequent then its superset will not be frequent. So if an itemset is found infrequent then algorithm will not search for its superset. Thus algorithm does not search for sets whose subsets is not frequent.

IV. PROPOSED ALGORITHM

(1) [Maintain sorted horizontal representation of DB.]
First of all sort all the itemsets in database (DB) in accordance to their items.

(2) [Creating bit vector table.]
Then make 2D table for Transaction versus Items(Transaction/Items) and put 1 in the table if item presents in corresponding transaction, else put 0. Count the number of times the item is coming in different transactions.

(3) [composing maximul frequent itemset.]
All items whose count>=minimum support(S), jointly make the maximul frequent itemset. The number of items in maximul frequent itemset is n. Discard all itemsets whose count is < minimum support(S).

(4) Then generate all subsets of maximul frequent itemset that have two or more items.

(5) [Scanning database which is created in step 1, for finding 2 item's subset is frequent or not.]
After generating subsets, while subsets' item=2:
If freq(itemset)>=minimum support(S)
Then: itemset is frequent
Else
Itemset is infrequent.

(6) while subsets' item>2 to n:
If any subset of subset is not frequent
Then: do not search database for this subset.
[because if any set is not frequent then it's superset can not be frequent.]

Else

[Search the database for this subset.]

If freq(itemset)>=minimum support(S)
Then: itemset is frequent
Else
Itemset is infrequent.

(7) Exit.

Example:

Table 4. Example

Transaction Identifier	List of Items
T1	A,B
T2	C,D,E
T3	A,B,C,F
T4	A,C,D
T5	A,D,F
T6	B,E,I
T7	A,B,D,E
T8	D,E,F,I

Minimum support(S) is 3 then generation of frequent itemset for the above database using the proposed approach is as follows:

Step 1. Creating the sorted horizontal representation of transaction database (TB).

Table 5. Sorted database representation

Transaction	Itemsets
T1	A,B
T3	A,B,C,F
T7	A,B,D,E
T4	A,C,D
T5	A,D,F
T6	B,E,I
T2	C,D,E
T8	D,E,F,I

Step 2. Creating the bit vector representation for transaction versus items.

Table 6. Counting and bit vector database representation

Transaction Identifier	Items						
	A	B	C	D	E	F	I
T1	1	1	0	0	0	0	0
T3	1	1	1	0	0	1	0
T7	1	1	0	1	1	0	0
T4	1	0	1	1	0	0	0
T5	1	0	0	1	0	1	0
T6	0	1	0	0	1	0	1
T2	0	0	1	1	1	0	0
T8	0	0	0	1	1	1	1
Count	5	4	3	5	4	3	2

Step 3. The count for the item I is less than minimum support(S=3). So it is not included in maximul frequent itemset.

Step 4. The maximul frequent itemset is {A,B,C,D,E,F}.

Step 5. The subsets of maximul frequent itemset are as:

{A,B} {A,C} {A,D} {A,E} {A,F} {B,C} {B,D} {B,E} {B,F} {C,D} {C,E} {C,F} {D,E} {D,F} {E,F} {A,B,C} {A,B,D} {A,B,E} {A,B,F} {B,C,D} {B,C,E} {B,C,F} {C,D,E} {C,D,F} {D,E,F} {A,B,C,D} {A,B,C,E} {A,B,C,F} {B,C,D,E} {B,C,D,F} {C,D,E,F} {A,B,C,D,E} {A,B,C,D,F} {B,C,D,E,F} {A,B,C,D,E,F}.

Step 6. Itemset or subset whose items=2 are {A,B} {A,C} {A,D} {A,E} {A,F} {B,C} {B,D} {B,E} {B,F} {C,D} {C,E} {C,F} {D,E} {D,F} {E,F}. So algorithm starts scanning the sorted horizontal database for all these subsets.

Let for {A,B},the algorithm scans the transactions T1, T3, T7, T4. After scanning the T4, which has items A,C,D. Because as the data items are sorted according to items and A,B will not come after A,C,D. So the algorithm terminates the searching database for A,B. This step continues for all subsets whose items=2.

As the count for {A,B} in database is 3, so it is frequent.

Step 7. For itemset or subset whose items>2 to n. If any subset of itemset or subset is infrequent then the itemset (subset) is infrequent and the database is not searched for the itemset (subset).

Let for itemset {A,B,D}. Its subsets are {A,B} {A,D} {B,D}. As {A,B} and {A,D} are frequent but {B,D} is not frequent. So {A,B,D} is also not frequent and it is not searched in database. If for any itemset, all of its subsets are frequent then database is scanned to check that the itemset is frequent or not. This step is repeated for all subsets whose item are from 3 to n.

Step 8. After scanning the database for all itemset (subset) the algorithm terminates.

So the frequent itemset for the above database is:

{A,B} {A,D} {D,E} .

CONCLUSION

As the database is sorted, this algorithm reduces the full scan of database. As the higher order item than subset came in the database the algorithm stops scanning the database for the that subset. Also if any subset of an itemset is infrequent then the itemset is also infrequent. So the algorithm not search for that itemset.

REFERENCES

[1] Robin Singh Bhadoria (2011) - international journl of computer technology and application, ISSN:2229-6093, Vol 2 (5), 1328-1333, Analysis of Frequent Item set Mining on Variant Datasets.

[2] Qihua Lan, Defu Zhang, Bo Wu (2009) - Global Congress on Intelligent Systems, 978-0-7695-3571-5/09 $25.00 © 2009 IEEE DOI 10.1109/GCIS.2009.387

[3] Bharat Gupta (2011), A Better Approach to Mine Frequent ItemsetsUsing Apriory and FP-Tree Approach.

[4] Varsha Mashoria, Anju Singh (2013), IOSR Journal of Engineering (IOSRJEN), ISSN: 2250-3021, ISSN: 2278-8719 Vol. 3, Issue 1(Jan. 2013), ||V1|| PP 58-64 Literature Survey on Various Frequent Pattern Mining Algorithm.

AUTHOR

I Madhu Bala Babel received my B.E. degree in 2009 in Computer Science and Engineering from Rajasthan University, Jaipur. I am pursuing my M.Tech degree from Mewar University, Chittorgarh, Raj. My research interests include Algorithm Analysis and Design.