A Review On Frequent Pattern Mining Algorithms

Sunita Murjani[1], Dhwani Dave[2]
[1]Assistant Prof. Computer Science and Engineering Department
Dr. Jivraj Mehta Institute Of Technology,Mogar Anand
Gujarat,India.
sunitas_murjani@yahoo.com
[2]Assistant Prof. Computer Science and Engineering Department
Dr. Jivraj Mehta Institute Of Technology,Mogar Anand
Gujarat,India.
dave.dhwani@gmail.com

***Abstract*: Frequent itemsets play an essential role in many data mining tasks thattry to find interesting patterns from databases, such as association rules,correlations, sequences, episodes, classifiers, clusters and many more of whichthe mining of association rules is one of the most popular problems. The original motivation for searching association rules came from the need toanalyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Association rules describehow often items are purchased together. For example, an association rule"beer ⇒ chips (80%)" states that four out of five customers that boughtbeer also bought chips. Such rules can be useful for decisions concerning product pricing, promotions, store layout and many others.**

***Keywords*—Data mining, Frequent items, itemset**

I. INTRODUCTION

Data mining is the extraction of useful patterns and relationships from data sources, such as databases, texts, the web etc. It has nothing to do however with SQL, OLAP, data warehousing or any of that kind of thing. It uses statistical and pattern matching techniques .Many areas of science, business, and other environments deal with a vast amount of data, which needs to be turned into something meaningful, knowledge. Many website owners and SEO professionals use different statistical packages to make sense of their data, as do many other professionals. Data mining is often overlooked when in fact it can provide very interesting information that statistical methods are unable to produce or produce properly. These data mining methods give you a lot more control. Various techniques have been developed in data mining amongst which primarily frequent pattern mining or Association rule mining is very important which results in association rules. These rules are applied on market based analysis, medical applications, science and engineering, music data mining, banking etc for decision making. Association rules are used to unearth relationships between apparently unrelated data in a relational database

Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data mining can be viewed as a result of the natural evolution of information technology. The database system industry has witnessed an evolutionary path in the development of the functionalities Such as data collection and database creation, data management (including data storage and retrieval).

Various application of data mining

- Banks -to detect which customers are using which products so they can offer the right mix of products and services to better meet customer needs –cross sell and up sell.
- Business
  1) advertising,
  2) Customer modeling and CRM (Customer Relationship management)
  3) e-Commerce
  4) fraud detection
  5) investments
  6) manufacturing
  7) targeted marketing
- Web search engines
- Credit card companies -to assist in mailing promotional materials to people who are most likely to respond
- Stock Market – market timing, stock selection, risk analysis
- Telecommunications
- Retail – market basket analysis to help determine marketing strategies
- Enrollment Management

Frequent Pattern Mining

Frequent Pattern mining plays essential role in many important data mining task such as mining Association rules, correlations, sequential patterns, multi dimensional patterns, max patterns. Frequent pattern mining finds the frequent pattern from very large database and increases efficiency of computation. Thus effective and efficient frequent pattern mining is a major research problem. Frequent Pattern are used in association rule mining. These rules are applied on market based analysis, medical applications, science and engineering, music data mining , banking etc for decision making. For example, In market based analysis , when customer buys items, some of them are having dependencies on each other. Finding these dependencies can be very helpful to market based analysis and these dependencies can be found by association rule mining in which important patterns are found. This is known as extracting pattern from database. From these patterns association rules are found.

Fundamentals

Table 2.1 An Example

TID	ITEMS
1	A,B
2	A,C,D,E
3	B,C,D,E
4	A,B,C,D
5	A,B,C,E

Table 2.1 illustrates an example of market basket transactions. Each row in this table corresponds to a transaction, which contains a unique identifier labeled TID and a set of items bought by a given customer. Retailers are interested in analyzing the data to learn about the purchasing behavior of their customers. Such valuable information can be used to support a variety of business-related applications such as marketing promotions, inventory management, and customer relationship management.

Item set and support count

Let I = {i1,i2,i3,…….in} be the set of all Items , T={t1,t2,t3,…..,tm} be the set of all transactions. Each transaction t contains a subset of items chosen from I. In association analysis, a collection of zero or more items is termed an item set. If an item set contains k items, it is called a k-item set. For instance, {B,C,D} is an example of a 3-itemset. The null (or empty) set is an item set that does not contain any items.

The transaction width is defined as the number of items present in a transaction. A transaction t_i is said to contain an item set X if X is a subset of t_j.. For example, the second transaction shown in Table 1 contains the item-set {A,C} but not {A,B}. An important property of an item-set is its support count, which refers to the number of transactions that contain a particular item set. In the data set shown in Table 1, the support count for {B,C,D} is equal to two because there are only two transactions that contain all three items.

Association Rules:

An association rule is having two important things support and confidence. It is an implication expression of the form P → Q , where P and Q are disjoint item sets, i.e., $P \cap Q = \emptyset$. The strength of an association rule can be measured in terms of its support and confidence.

Support is the number of transactions in which the association rule holds. It is the percentage of transactions that demonstrate the rule. Suppose the support of an item is 0.4%, it means only 0.4 percent of the transaction contain purchasing of this item.

Support (PQ)=Support count of(P∪Q)/Total number of transactions in database

Support is an important measure because a rule that has very low support may occur simply by chance. A low support rule is also likely to be uninteresting from a business perspective because it may not be profitable to promote items that customers seldom buy together. For these reasons, support is often used to eliminate uninteresting rules. Support also has a desirable property that can be exploited for the efficient discovery of association rules.

Confidence is the conditional probability that, given A present in transaction, B will also be present

Confidence (PQ) = Support count of (P∪Q) / Support(Q)

Confidence, on the other side, determines the reliability of the conclusion made by a rule. For a given rule P → Q, the higher the confidence, the more likely it is for Q to be present in transactions that contain P. Confidence also provides an estimate of the conditional probability of Q given P.

The aim of association rule is to discover all association problems having support and confidence not less than the given value of threshold. If the support and confidence of item set of database is less than minimum support and confidence then that item set is not frequent item set.

II. CLASSICAL APRIORI ALGORITHM

Apriori algorithm is basic and very popular algorithm which is proposed by R. Agrawal in1994.[1] It is used for to find frequent item sets among whole database. It is used when we have to find some important data among large database. It uses the breath first search technique. Apriori algorithm uses iterative search method to find frequent item set. It repeatedly scans the whole database to count support value of item set. If we need to find K item sets then this classical algorithm requires scanning database K-1 times. First it finds 1-dimensional frequent item set is denoted as L1 then 2-dimensional is denoted by L2 and so on until no more frequent item sets found.[2][3]

Classical apriori algorithm has two steps: 1) Join step 2) Prune step

1) The join step: Join operation is performed by itself to find new candidate sets. To find L_k a set of candidate k-item sets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k.[1]

2) The prune step**:** In this step examine all members of C_k which are may or may not be frequent, but all of the frequent k-item sets are included in C_k. A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k. To reduce the size of C_k, the Apriori first count the support value of each member and compare with predefined minimum support value. Which are not satisfy the min support value are been deleted.[2]

Algorithm:-

Input: D, Database of transactions; min_sup, minimum support threshold
Output: L, frequent itemsets in D [3]
Method:

```
(1) L1=find_frequent_1-itemsets(D);
(2) for(k=2; $L_{k-1}\neq\Phi$; k++){
(3) $C_k$=apriori_gen($L_{k-1}$, min_sup);
(4) for each transaction t∈D{
(5) Ct=subset($C_k$,t);
(6) for each candidate c∈Ct
(7) c.count++ ;
(8) }
(9) Lk={ c∈$C_k$ |c.count≥min_sup }
(10) }
(11) return L=Uk$L_k$ ;
```

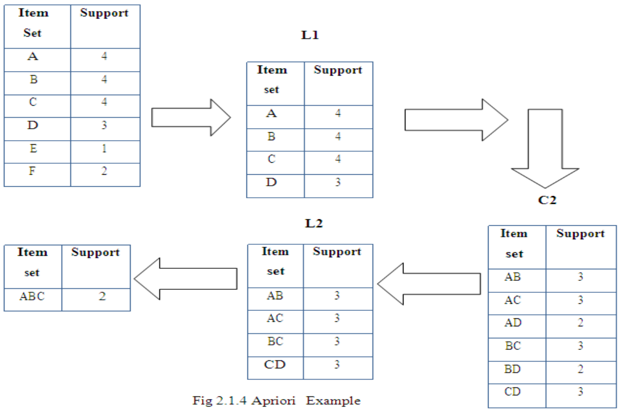
Procedure apriori_gen(L_{k-1}:frequent(k-1)-itemsets)

```
(1) for each itemset l1∈ $L_{k-1}${
(2) for each itemset l2∈ $L_{k-1}${
(3) if(l1 [1]= l2 [1])∧ (l1 [2]= l2 [2]) ∧…∧(l1 [k-2]=
        l2 [k-2]) ∧(l1 [k-1]< l2 [k-1]) then {
(4) c = l1∞l2;
(5) if has_infrequent_subset(c, $L_{k-1}$) then
(6) delete c;
(7) else add c to $C_k$ ;
(8) }}}
 (9) return $C_k$;
```

Procedure has_infrequent_subset(c: candidate k-itemset; L_{k-1}:frequent(k-1)-itemsets)

```
 (1) for each(k-1)-subset s of c {
 (2) if s ∉ $L_{k-1}$ then
 (3) return true; }
 (4) return false;
```

Example:-



Item Set	Support
A	4
B	4
C	4
D	3
E	1
F	2

L1

Item set	Support
A	4
B	4
C	4
D	3

C2

Item set	Support
AB	3
AC	3
AD	2
BC	3
BD	2
CD	3

L2

Item set	Support
AB	3
AC	3
BC	3
CD	3

Item set	Support
ABC	2

Fig 2.1.4 Apriori Example

Advantages:
1)This is very simple and can be easily implemented.

Disadvantages:
1) It requires iterative scans of database.
2) It produce large candidate set so requires more memory and time.

III. REVIEW ON FREQUENT PATTERN MINING ALGORITHMS

A. Improved Efficiency of Apriori using Transaction Reduction and Matrix[4]

In this paper [4], Vipul Mangla presents the improved Apriori Algorithm, which is used to mine association rules. In this Algorithm transactions and items are represented in Matrix form. The rows of the matrix represent the transaction and the columns of the matrix represent the items. The elements of matrix A are:
A= [aij] = 1, if transaction i has item j
otherwise
A =[aij] = 0
We assume minimum support and confidence is given.
The sum of the jth column vector gives the support of jth Item.
And the sum of the ith row vector gives the S-O-T, that is, size of ith transaction (no. of items in the transaction).
Now we generate the item sets. For, 1–frequent item set, we check if the column sum of each column is greater than minimum support. If not, the column is deleted. All rows with rowsum=1 (S-O-T) are also deleted. Resultant matrix will represent the 1-frequent item set. Now, to find 2-frequent itemsets, columns are merged by AND-ing their values. The

resultant matrix will have only those columns whose columnsum>=min_support. Additionally, all rows with rowsum=2 are deleted. Similarly the k_{th} frequent item is found by merging columns and deleting all resultant columns with columnsum<min_support and rowsum=k.When matrix A has 1 column remaining, that will give the k_{th} frequent itemset.

B. Dynamic Approach for Frequent Patterns Mining Using Transposition of Database [5]

In this Paper[5], Sunil Joshi Dynamic Approach for Frequent Patterns Mining Using Transposition of Database) for mining frequent patterns which are based on Apriori algorithm and used Dynamic function for Longest Common Subsequence [1]. The main distinguishing factors among the proposed schemes is the database stores in transposed form and in each iteration database is filter /reduce by generating LCS of transaction id for each pattern. Time taken by the algorithm to find frequent items compare to classical Apriori is very small.

The mining algorithm works over the entire database file, first transpose the database and count the number of item and transaction string generated for each item. Sort the item numbers. Now apply Apriori like Algorithm in which first we calculate frequent pattern C1.it reduces un-frequent pattern and its transaction details also. For each pass we apply following sequence of operation until condition occurred. First generate the candidate pattern and prune by Apriori method. To count the support , instead of whole database for each pruned pattern we find longest common subsequence and length of transaction string of pattern’s item and also stored new pattern and its transaction string so that next iteration we trace above string. To find longest common subsequence we used dynamic programming approach which faster then traditional approach.Write pruned pattern list with transaction string. So that in next pass we used this pattern list instead of all pattern list. An advantage of this approach is in each iteration database filtering and reduces, so each iteration is faster then previous iteration

C. An improved apriori algorithm(Scan database only twice)[6]

In this Paper [6], Girja Shankar introduced an improved Apriori algorithm that will reduce the number of scan whole database as well as reduce the redundant generation of sub items and the final one is to prune the candidate item sets according to min-support. To achieve these goals we introduce the concept of Global power set and database optimizations. An improved Apriori algorithm reduces s system resources occupied and improved the efficiency of the system.

To enhance the efficiency of production of the frequent item sets, in this paper [6] we discusses two problems of the Apriori algorithm. First, we need to scan the database multiple times and Second, it will generate large candidate item sets, which will increase the time and space complexity. To overcome these defects we first find frequent_one_itemset of database then generate power set of the frequent_one_itemset and initialized itemset count=0. Call this power set as Global power set. When we scan database for item set counting, first we delete items from transaction which is not present in frequent_one_item set list. This step will reduce the extra generation of candidate item sets. After delete process we generate Local power set of remaining items of the transaction and compare with the global power set. When match fund increase the item set count by one. This step will reduce the multiple scan of database. These steps will use for increase the efficiency of the algorithm. The advantages of this approach is it needs only two scans of database to generate frequent item set And It takes less time compare to classical apriori .(since there is no join operation) But It can not handle dynamic updates in database.

D. Improved Apriori Algorithm based on matrix[8]

In this paper[8] X. Luo uses matrix to represents transactions and items. If a item is present in transaction then it is indicated by 1 and 0 indicates absence of item in transaction. This algorithm deals with only two values 0 and 1 to find frequent itemset.

Table 2.2.6..1 Sample database

Tid	Items	I1 I2 I3 I4 I5
T1	I1,I2,I5	1 1 0 0 1
T2	I2,I4	0 1 0 1 0
T3	I2,I3	0 1 1 0 0
T4	I1,I2,I4	1 1 0 1 0
T5	I1,I3	1 0 1 0 0
T6	I2,I3	0 1 1 0 0
T7	I1,I2	1 1 0 0 0
T8	I1,I2I3,I5	1 1 1 0 1
T9	I1,I2,I3	1 1 1 0 0

For 1-itemset matrix represented is used (i.e.)

MAT(I1) = 100110111
MAT(I2) = 111101111
MAT(I3) = 001011011
MAT(I4) = 010100000
MAT(I5) = 100000010

Now by counting the number of 1’s in the matrix we can easily find the occurrences of that item.

For 2-itemset we can multiply the binary representation of the items to get the occurance of that items together.

To find how many times item I_j and I_k are appearing together we have to multiply the MAT(I_j) and MAT(I_k).

i.e $MAT(I_j,I_k)=MAT(I_j) * MAT(I_k)$.

MAT(I2,I5)=MAT(I2) *MAT(I5) =
100000010 * 111101111 = 100000010
Mat(I2,I5) = 100000010

Then support of these two items can be calculated as follows:
Support (I2,I5)= (Nos. of times Appearing together/Tot. Transaction) = 2 / 9

Similarly the same procedure can be followed for all possible itemset. This algorithm needs to scan the database only once and also does not require to find the candidate set when searching for frequent itemset.

IV. CONCLUSION

Throughout the last decade, a lot of people have implemented and compared several algorithms that try to solve the frequent itemset mining problem as efficiently as possible. Based on the candidate generation and scanning time we can easily analyze time complexity and space complexity of the algorithms.

REFERENCES

[1]Rakesh Agrawal, ”Fast algorithm for Mining Association Rule”, Proceedings *of the ACM SIGMOD International Conference Management of Data, Washington, 1993, pp.207-216.*

[2] Ekta Garg, Meenakshi Bansal, “A Survey on Improved Apriori Algorithm”, *International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 7, July – 2013.*

[3] Jaishree Singh, Hari Ram, Dr. J.S. Sodhi ”Improving efficiency of Apriori Algorithm Using Transaction Reduction”, *International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.*

[4] Vipul Mangla , “Improving The Efficiency of Apriori Algorithm in Data Mining” *International Journal of Engineering and Innovative Technology, ISSN No- 2277-3754 Vol 3 Issue 3 September-2013.*

[5] Sunil Joshi ,” An Implementation Of Frequent Pattern Mining Algorithm Using Dynamic Function “ , *International Journal Of Computer Applications, ISSN No-0975-8887 Vol 9 No 9 , November-2010.*

[6]Girja Shankar, Latita Bargadiya, “A New Improved Apriori Algorithm For Association Rules Mining” , *International Journal of Engineering Research & Technology (IJERT) ISSN No- 2278-0181 Vol. 2 Issue 6, June –2013* .

[7] X. Luo and W. Wang, “Improved Algorithms Research for Association Rule Based on Matrix,” *2010 International Conference on Intelligent Computing and Cognitive Informatics*, pp. 415– 419, Jun. 2010.