ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

Improved Generation of Frequent Item Sets using "Apriori Algorithm"

P.Suresh[1], K.N. Nithya[2], K.Murugan[3]

HOD, Dept of Computer Science, Salem Sowdeswari College - Govt. Aided, Salem, India[1]

Assistant Professor, Department of Computer Science, Sri Sakthi Kailaash Women's College, Salem, TN[2]

Assistant Professor, Department of Computer Science, Govt. College for Women, Kolar, India[3]

Abstract: Association Rule is a vital technique applied in the task of Data Mining to search and generate associations among large number of databases. A variety of algorithms have been introduced and used to find the relationships existing in databases. A very popular algorithm is Apriori algorithm. Though it is a very efficient process, it takes a considerable time to scan large amount of transactions in database to generate frequent item sets. Therefore an attempt is made to create an Improved Apriori algorithm that reduces the amount of scans needed to give away the frequent sets.

Keywords: Association Rule, Frequent Item Sets, Original Apriori algorithm, Improved Apriori algorithm.

I. INTRODUCTION

Association Rule was introduced by Aggarwal and Srikant in the year 1993. Association mining has been used in many application domains. One of the best known is the business field where discovering of purchase patterns or association between products is very useful for decision making and effective marketing. Extracting association rules is the core of Data Mining [5].Association rule mining is referred as a "Market Basket Analysis". However in last few years application areas have increased significantly. Some examples of recent applications are:

- Finding patterns in biological databases.
- Extraction of knowledge from software engineering metrics.
- Web personalization, text mining etc.
- discovering knowledge from agricultural databases,
- Survey data from agricultural research, data about soil and cultivation,
- Data containing information linking geographical conditions and crop production and so on.

Finding frequent item sets is more expensive since the number of itemsets grows exponentially with the number of items [3]. Association rules mining is to discover the associations and relations among item sets of large data. Association rules mining is an important branch of data mining research, and association rules is the most typical style of data mining [1].

A. Definition:
Let I=I1, I2, ... , Im be a set of m distinct attributes, T be transaction that contains a set of items such that $T \subseteq I$, D be a database with different transaction records Ts. An association rule is an implication in the form of $X \Rightarrow Y$, where X, $Y \subset I$ are sets of items called itemsets, and $X \cap Y = \emptyset$. X is called "antecedent" while Y is called "consequent"; the rule means X implies Y. There are two important basic measures for association rules, support(s) and confidence(c) [2].

B. Support(S):
The support supp(S) of an itemset S is defined as the proportion of transactions in the data set which contain the item set.

supp(S) = no. of transactions which contain the itemset S/ total no. of transactions

C. Confidence(C):
The confidence of a rule is defined as the total number of items in the transactions that contains the antecedents.

$$\mathrm{Conf}(X \rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X)$$

II. FREQUENT ITEM SET

The term is used to determine the items that occur in common in all the transactions in a database. If an itemset is not frequent, any of its superset is never frequent [4].Frequent item set acts as a backbone in association rule mining.
Frequent patterns are ones that occur at least a user-given number of times (minimum support) in the dataset. They allow us to perform essential tasks such as discovering association relationships among items, correlation, sequential pattern mining, and much more[6]. Thus the other items can be excluded from the process helping in easier derivation of association rules. The transaction that has at least a probable relationship between the items is found by using the "Minimum Support Value". Based on this value, we proceed to identify the frequent item set.

III. APRIORI ALGORITHM

This is a classical algorithm that forms associativity between the items in a database to generate frequent item sets. Two main steps involved are;

A. Candidate Generation:
Candidates are those items taken in to account with a minimum support value.

B. Prune:
The items which are less than minimum support value are removed from the candidate list.

Original Apriori Algorithm:

```
procedure Apriori (T, minSupport) { //T is the database and minSupport is the minimum support}
L1= {frequent items};
for (k= 2; Lk-1 !=ø; k++) {
Ck= candidates generated from Lk-1
//that is cartesian product Lk-1 x Lk-1 and eliminating any k-1 size itemset that is not
//frequent
for each transaction t in database do{
#increment the count of all candidates in Ck that are contained in t
Lk = candidates in Ck with minSupport
}//end for each
}//end for
return$U_kL_k$;
}
```

IV. PROPOSED ALGORITHM

In the improved Apriori algorithm, we are going to introduce an additional process of noting down the corresponding transactions ID's while generating the frequent 1-itemset. The notification of the transactions will help in avoiding scanning overall transactions.
Steps:

1. Input the database and the minimum support value
2. Find the frequent 1-item set(L1) and note down its corresponding Transaction IDs
3. Prune (L1) whose item set value< minimum support value.
4. Generate the candidates (C1) with the Transaction Ids.
5. For frequent 2-item set, we are going to apply the Intersection Operation to the corresponding transaction Ids got from step 2.
6. Count the total transactions present and compare with the Minimum Support Value.

Improved Apriori Algorithm:

```
procedure Apriori (T, minSupport) { //T is the database and minSupport is the minimum support}
L1= {frequent items};
T1={Transaction IDs for the frequent items};
for (k= 2; Lk-1 !=ø; k++) {
Ck= candidates generated from Lk-1;
Tk= transactions ID values of Lk-1;
for each transaction t in database do{
Tk = Lk∩Lk-1;
#increment the count of all candidates in Ck that are contained in t;
# count the value of Tk.
Tk = candidates in Tk with minSupport
}//end for each
}//end for
return$U_kL_k$;
}
```

V. RESULT AND DISCUSSION

1. The table 1 shows the database that consists of list of transactions and the itemsets occurring in each transaction. Let the minimum support value be 3.

TABLE 1

TRANSACTIONS	ITEM LIST
T1	I1,I2,I5
T2	I2,I4
T3	I2,I4
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

2. Generate frequent 1-item set and note the corresponding Support value and Transaction ID(TIDs).

1-ITEM SET	SUPPORT VALUE	TID	
I1	6	T1,T4,T5,T7, T8,T9	
I2	7	T1,T2,T3,T4, T6,T8,T9	
I3	5	T5,T6,T7,T8, T9	
I4	3	T2,T3,T4	
I5	2	T1,T8	DELETED

Here, the item I5 has 2 as the support value. So it is deleted. Now we have

1-ITEM SET	SUPPORT VALUE	TID
I1	6	T1,T4,T5,T7,T8,T9
I2	7	T1,T2,T3,T4,T6,T8,T9
I3	5	T5,T6,T7,T8,T9
I4	3	T2,T3,T4

3. Generate the frequent 2-item set. For the combination of any two items, we apply the intersection process of its respective TIDs. We note only the common transactions present and eliminate the other transactions. This process is done to search for the Items only in these transactions than searching from the beginning. For Example, take the first 2-item set (I1, I2).

a) Note down the common transactions between I1 and I2. They are T1, T4, T8, T9. Search for the presence of (I1, I2) only in these four transactions. Others are unsearched.

b) Count te total number of transactions that has this item set. The Support value = 4.

Likewise, we find the support values for the remaining 2-item sets and eliminate as before. The table below shows the result;

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

2-ITEM SET	COMMON TIDs	SUPPORT VALUE	
I1,I2	T1,T4,T8,T9	4	
I1,I3	T5,T7,T8,T9	4	
I1,I4	T4	1	DELETED
I2,I3	T6,T8,T9	3	
I3,I4	No common transactions	0	DELETED

c) The frequent 2-item sets are;

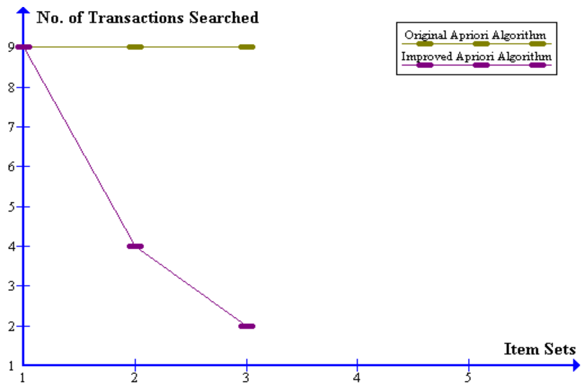
2-ITEM SET	COMMON TIDS	SUPPORT VALUE
I1,I2	T1,T4,T8,T9	4
I1,I3	T5,T7,T8,T9	4
I2,I3	T6,T8,T9	3

4. The frequent 3-item set is evaluated as explained above. Therefore the result we get is(I1,I2,I3). Thus these 3 items has the maximum possibility of associations to occur in the database.

Comparison in Number of Transactions Searched:

ITEMSETS	ORIGINAL APRIORI ALGORITHM	IMPROVED APRIORI ALGORITHM
1-Item Set	9	9
2-Item Set	9	4
3-Item Set	9	2

Graphical Representation:



VI.CONCLUSION

From the above results and discussions, it is clear that the number of transactions involved in searching for the frequent item sets is considerably reduced, therefore enhancing the execution of Apriori algorithm. We hope that by using the above Improved Apriori algorithm, we can search large number of Data Bases with less effort by completely reducing the transaction size, hence, arriving at the generation of association rules accurately.

REFERENCES

1. "Research of an Improved Apriori Algorithm in Data Mining Association Rules" Jiao Yabing International Journal of Computer and Communication Engineering, Vol. 2, No. 1, January 2013
2. "Mining Frequent Itemsets – Apriori Algorithm", Laboratory Module 8
3. "Analysis on Improved Pruning in Apriori Algorithm, Jovita Vani Sequeira, Zahid Ansari, Volume 5, Issue 3, March 2015, International Journal of Advanced Research in Computer Science and Software Engineering.
4. "The Apriori algorithm: Data Mining Approaches Is To Find Frequent Item Sets From A Transaction Dataset, Abhang Swati Ashok, JoreSandeep S., International Journal of Innovative Research in Science, Engineering and Technology,Volume 3, Special Issue 4, April 2014
5. "AN Improved Apriori Algorithm For Association Rules",Mohammed Al-Maolegi1, Bassam Arkok2,International Journal on Natural Language Computing (IJNLC) Vol. 3, No.1, February 2014
6. "An Efficient Data Mining Technique for Generating Frequent Item sets ",K. Geetha, Sk.Mohiddin, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013
7. "Text Classification Using Data Mining", S. M. Kamruzzaman, Farhana Haider, Ahmed Ryadh Hasan

BIOGRAPHIES

Dr. P. Suresh is a Head, Department of Computer Science, Salem Sowdeswari College [Govt. Aided], Salem. He received the M.Sc., Degree from Bharathidasan University in 1995, M.Phil Degree from Manonmaniam Sundaranar University in 2003, M.S (By Research) Degree from Anna University, Chennai in 2008, PGDHE Diploma in Higher Education and Ph.D., Degree from Vinayaka Missions University in 2010 and 2011 respectively in Computer Science. He is an Editorial Advisory Board Member of Elixir Journal. His research interest includes Data Mining and Natural Language Processing. He is a member of Computer Science Teachers Association, New York.

Mrs. K.N. Nithya is an Assistant Professor in the Department of Computer Science, Sri Sakthi Kailaash Women's College. She received the B.C.A degree and M.C.A degree from Periyar University. Her research area of interest includes Data Mining and Mobile Computing.

Mr. K. Murugan is an Assistant Professor and Head of the Department of Computer Science at Government College for Women, Kolar. His current area of research interest is Computer Networks and its applications. He has successfully guided 15 candidates for M.Phil. He has been teaching computer Science for the past 17 Years. He completed his Master Degree in Computer Science at Bharathidasan University, Master of Philosophy in Computer Science at Manonmaniam Sundaranar University, Master of Engineering in Computer Science and Engineering at Anna University.