International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 4, Issue 1 (July 2014) PP: 01-05

Efficient Temporal Association Rule Mining

Geetali Banerji[1], Kanak Saxena [2]
[1]*Professor, Computer Science, IINTM, GGSIPU, New Delhi*
[2] *Professor, Department of Computer Application, SATI, Vidisha (M.P.),*

Abstract: *Associationship is an important component of data mining. In real world applications, the knowledge that is used for aiding decision-making is always time-varying. However, most of the existing data mining approaches rely on the assumption that discovered knowledge is valid indefinitely. For supporting better decision making, it is desirable to be able to actually identify the temporal features with the interesting patterns or rules. This paper presents a novel approach for mining Efficient Temporal Association Rule (ETAR). The basic idea of ETAR is to first partition the database into time periods of item set and then progressively accumulates the occurrence count of each item set based on the intrinsic partitioning characteristics. Explicitly, the execution time of ETAR is, in orders of magnitude, smaller than those required by schemes which are directly extended from existing methods because it scan the database only once.*

Keywords: *Apriori Algorithm, Association Rules, ETAR, Frequent Patterns, Itemset, Transactional Database, Temporal database.*

I. INTRODUCTION

Due to a wide variety of application potentials of association rules, the problem of association rule discovery has been studied for several years. Most previous work overlooks time components [9, 10], which are usually attached to transactions in databases also most of them require multiple passes over the database resulting in a large number of disk reads and placing a huge burden on the I/O subsystem. It becomes much tedious to mine the association rules as the data are growing more and more like mountain. Hence it is important in developing techniques in such a way that interesting rules are mined effectively from huge databases.

This work addresses temporal issues of association rules beside its execution time. The results of experiments show that many time-related association rules [11], that would have been missed with traditional approaches can be discovered with the techniques and approaches presented in this paper. The major concerns in this paper are the identification of the valid period for the item sets.

In case of Marketing, it is important that in database, some items which are infrequent in whole dataset but may be frequent in a particular time period. If these items are ignored then associationship result will no longer accurate. Marketing persons, who expect to use the discovered knowledge, may not know when some item set are required or whether it still is valid in the present, or if it will be valid sometime in the future. For supporting better decision making, it is desirable to be able to actually identify the temporal features with the interesting patterns or rules.

The paper is organized as follows. Section 2 discuss about association rules followed by an efficient apriori algorithm with single scan of data base. Section 4 introduces the proposed ETAR algorithm. Section 5 is concern with empirical results, followed by conclusions.

II. ASSOCIATION RULES

Mining association rules is particularly useful for discovering relationships among items from large databases [3]. A standard association rule is a rule of the form X→ Y which says that if X is true of an instance in a database, so is Y true of the same instance, with a certain level of significance as measured by two indicators, support and confidence. The mining process of association rules can be divided into two steps.

2.1. Frequent Itemset Generation

Generate all sets of items that have support greater than a certain threshold, called minsupport.

2.2. Association Rule Generation

From the frequent itemsets, generate all association rules that have confidence greater than a certain threshold called minconfidence [4]. Apriori is a renowned algorithm for association rule mining primarily because of its effectiveness in knowledge discovery [5].However; there are two bottlenecks in the Apriori algorithm. One is the complex frequent itemset generation process that uses most of the time, space and memory. Another bottleneck is the multiple scan of the database [6].

Linked list of item ids	No of transactions containing items	Linked list of Transaction ids

Fig. 1 Structure of each array element / node

The following section discuss about the improved version of apriori algorithm.

III. AN IMPROVED APRIORI BASED ALGORITHM WITH SINGLE SCAN

As in [8], the authors have proposed an improved apriori based approach with single scan of database. In this approach, frequent patterns can be mine by just a single scan of database. This is achieved by carrying out certain changes at the algorithmic level. In First scan, the complete transactional database is copied into an array of items ids containing pointer to the Linked list of Transaction id (see Fig 1). Each iteration outcome is new candidate set with frequent item sets, the old Linked lists are removed, which leads to efficient utilization of memory as well as fast access. The algorithm is as follows:

Algorithm

```
1)  [Connect to Database] Open Table.
2)  [Create a dynamic array to store all the transaction Id containing specific Item Id]
    Create an array where array size = No. of items. Each array element should contain Item Id (IID),
    Count (initialize to zero), and Address (add. of first transaction containing the specified item of a
    linked list, initialize to NULL).Initialize DeletedIID to NULL. Flag = "N".
3)  [Process all Transactions]
    Repeat step 4 and step 5 Until EOF.
4)  Read transaction record.
5)  [Copy the transaction ID into various items]
    Create a node and copy TID into corresponding Items and increment the Count.
    [End of Loop]
6)  [All the items with their Transaction Id are copied into the Data Structure]
    Close Transactional Database.
7)  [Items below minsupport are removed]
    Delete the items below minimum support and release the memory.
8)  Sort the DS in descending order according to Count.
9)  [Find frequent item sets]
    Read first Item DS as PrevItem
10) [Process all the items]
    Repeat step 11 thru step 17 until no more items is left
11) Read next Item as CurrItem
12)  Repeat steps 13 thru step 16 until no more element is left
13)  [Check for the item set contained in deleted one]
    If Flag = "Y" then
    Search PrevItem and CurrItem in DeletedIID
    If not found then
    For each TID common in PrevItem and CurrItem copy both the items with TID into a new   DS (check
    for duplicacy).
    Else
                    Go to Step 17
    EndIF
    Else if Flag = "N"
    For each TID common in PrevItem and CurrItem copy both the items with TID into a new   DS (check
    for duplicacy).
    EndIF
14) Increment the Count.
15) Read Next CurrItem.
    [End of loop in step 12]
16) Delete items with Count less than minimum support and copy them into a new DS DeletedIID.
17) Read Next PrevItem.
    [End of loop in step 10]
18) Delete the previous DS.
19)  Flag = "Y"
20) Repeat step 8 thru step 18 until no common TID left.
```

21) *The item set which finally exists is the most frequent item set.*

The above algorithm does not consider the temporal aspect of transactions. The proposed algorithm tackles efficient utilization of memory as well as temporal aspect. The following section discuss proposed algorithm.

IV. EFFICIENT TEMPORAL ASSOCIATION RULE MINING

In this approach we can mine frequent patterns as per the time interval specified by the user. The tuples, in which date of purchase is same as the time interval, copied into a linked list. The major magnetism of this algorithm is that the size of the linked list keeps on reducing. The algorithm is as follows:

1) *[Input the Time interval]*
   Read TI
2) *[Copy the transactions, where date of purchase matches the TI]*
   Create the linked list (OLDL) and copy item id's (IID), Transaction ids (TID) , date of purchase (DATE) and store the number of transactions where item is present (COUNT) .
3) *[Generate frequent k=1-Item set]*
   Delete the nodes with COUNT below minimum support.
4) *Repeat Step 5 to Step 8 until no common TID or the existing item set COUNT is below minimum support.*
5) *Traverse the linked list and copy item with common TID into a new linked list (NEWL).*
6) *[Generate frequent k+1-item sets]*
   Delete the nodes with COUNT below minimum support from NEWL.
7) *IF NEWL is non empty THEN copy NEWL into OLDL.*
8) *Delete NEWL.*
9) *OLDL contains the most frequent item set.*

V. EMPIRICAL RESULTS

Our data set contains 6 transactions containing transaction id, list of items purchased and date of purchase. This data set has been used for two different algorithm mentioned in section 3 and 4. In case of ETAR it is used for two different intervals. Table 1 depicts the data set. In this case minimum support is 2.

Table 1. Transactional Database

Transactions (TID)	List of Items (IID)	Date(DT)
100	c, d, e, f, g, i	<*,01,04>
200	a, c, d, e, m, b	<*,01,04>
500	a, c, d ,e, b	<*,01,04>
400	a, c, d ,h	<*,06,04>
600	a, b, d, h, i	<*,06,04>
300	a, b, d, e, g, k	<*,06,04>

5.1 Improved Apriori Based Algorithm with Single Scan of Database algorithm

After applying the algorithm discussed in section 3, we found that the most frequent item set is (a, b, c, d, e).

5.2 ETAR algorithm with time interval (TI) = <*, 01, 04>

Table 2 depicts candidate-1 item sets for the specified time interval. Table 3 to table 6 shows the status of linked list (in tabular form) for different candidate sets. Table 7 shows the most frequent item set.

Table 2. Candidate 1-item sets

IID	COUNT	TID		
a	2	200	500	
m	1	200		
b	2	200	500	
c	3	100	200	500
d	3	100	200	500
e	3	100	200	500
f	1	100		
g	1	100		
i	1	100		

Table 3. Frequent 1-item sets

IID	COUNT	TID		
a	2	200	500	
b	2	200	500	
c	3	100	200	500
d	3	100	200	500
e	3	100	200	500

Table 4. Frequent 2-item sets

IID		COUNT	TID		
c	d	3	100	200	500
c	e	3	100	200	500
c	b	2	200	500	
c	a	2	200	500	
d	e	3	100	200	500
d	a	2	200	500	
d	b	2	200	500	
e	a	2	200	500	
e	b	2	200	500	
a	b	2	200	500	

Table 5. Candidate 3 & 4 item sets

IID			COUNT	TID		
c	d	e	3	100	200	500
c	d	a	2	200	500	
c	b	d	2	200	500	
c	d	e, a	2	200	500	
c	d	a, b	2	200	500	
c	e	a	2	200	500	
c	e	b	2	200	500	

Table 6. Frequent 5 item sets

IID			COUNT	TID	
c	d	e, a	2	200	500
c	d	e, b	2	200	500
c	e	a, b	2	200	500

Table 7. Frequent 4 item sets

IID	COUNT	TID	
c, d, e, a, b	2	200	500

From table 7, it is found that the most frequent item set is (a, b, c, d, e).

5.3 ETAR algorithm with time interval (TI) = <*, 06, 04>

Table 8 depicts candidate-1 item sets and table 9 shows most frequent 1 item sets for the specified time interval. ETAR algorithm is applied for the specified time interval.

Table 8. Candidate 1-item sets

IID	COUNT	TID		
a	3	400	600	300
c	1	400		
b	2	600	300	
d	3	400	600	300
h	2	400	600	
i	1	600		
g	1	300		
k	1	300		
e	1	300		

Table 9. Frequent 1-item sets

IID	COUNT	TID		
a	3	400	600	300
b	2	600	300	
d	3	400	600	300
h	2	400	600	

The most frequent item sets found for the specified time interval is (a, b, d).

On the basis of above patterns, it is found that item set (a, b, d) is popular during the complete calendar year. The stock of these items should be maintained throughout the year. Whereas (c, e) is popular for a particular time period only.

VI. CONCLUSIONS

In this paper, we have proposed an Algorithm which gives an efficient time sensitive approach for mining frequent item in the dataset. Discovered rule is easy to understand. Efficient Temporal Association Rules Algorithm can mine frequent patterns with a single scan of database. It is found that different association rules are discovered while considering different time intervals associated to it.

REFERENCES

[1] Syed Khairuzzaman Tanbeer , Chowdhury , Farhan Ahmed and Byeong-Soo Jeong , Parallel and Distributed Algorithms for FP mining in large Databases, *IETE Technical Review Vol. 26, Issue 1 , 55-65, Jan 2009.*

[2] Umarani V., Punithavalli M., “A Study of effective mining of Associative rules from huge databases”, IJCSR International Journal of Computer Science and Research, Vol. 1 Issue 1, 2010

[3] Yu-Chiang Li, Jieh-Shan Yeh, Chin-Chen Chang, “Efficient Algorithms for Mining Shared-Frequent Itemsets”, In Proceedings of the 11th World Congress of Intl. Fuzzy Systems Association, 2005.

[4] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, "Association Rule Mining and its Application to MPIS", 2003.

[5] Agrawal, R. and Srikant, R., Fast algorithms for mining association rules. In Proc.20th Int. Conf. Very Large Data Bases, 487-499, 1994.

[6] Sotiris Kotsiantis, Dimitris Kanellopoulos,” Association Rules Mining: A Recent Overview", GESTS International Transactions on Computer Science and Engineering, Vol.32, No: 1, pp. 71-82, 2006.

[7] R. Agrawal, T. Imielinski and A.Swami Mining association rules between sets of items of large databases, *Proc. of the ACM SIGMOD Intl'l Conf. On Management of Data, Washington D.C., USA*,.207– 216, 1993.

[8] Geetali Banerji, Kanak Saxena, An Improved Apriori Based Algorithm with Single Scan of Database, National Conference on Converging Technologies Beyond 2020, April 2011, UIET, Kurukshetra University, India

[9] Verma Keshri, Vyas O.P., Efficient calendar based temporal association rule, SIGMOD Record Vol. 34, No. 3, Sept 2005

[10] Yingjiu Li, Peng Ning, X. Sean Wang ,Sushil Jajodia R :“Discovering calendar- based temporal association rules”, Data & Knowledge Engineering volume 4,Elesvier publisher, Volume 44, 193-214 ,2003

[11] Claudio Bettini, X. Sean Wang R: “Time Granularies in databases, Data Mining, and Temporal reasoning 2000., 230, ISBN 3-540-66997-3, Springer-Verlag, July 2000, 230 pages. Monograph.