International Journal of Scientific & Engineering Research, Volume 3, Issue 12, December-2012
ISSN 2229-5518

Incremental Association Rule Mining Through Vertical Transaction ID

Kamlesh Malpani, Dr.Parashu.Ram Pal

Abstract— Association rule mining is a popular data mining technique which gives us valuable relationships among different items in a dataset. In dynamic databases, new transactions are appended as time advances. This may introduce new association rules and some existing association rules would become invalid. Thus, the maintenance of association rules for dynamic databases is an important problem. Several incremental algorithms, is proposed to deal with this problem. In this paper we proposed algorithm VTII (Vertical Transaction Id Intersections). This algorithm reduces a number of times to scan the database (old and new) to generate frequent pattern. As a result, the algorithm has execution time faster than that of previous Algorithms. This paper also conducts experiments to show the performance of the proposed algorithm. The result shows that the proposed algorithm has a good performance.

Keywords—Association rule, Dynamic maintenance, Incremental , Vertical**.**

1. INTRODUCTION

Database is dynamic when new transactions are inserted and deleted from the database. This may introduce new association rules and some existing association rules would become invalid. As a brute force approach, apriori may be reapplied to mining the whole dynamic database when the database has been changed. However, this approach is very costly even if small amount of new transactions is inserted into a database. Thus, the association rule mining for a dynamic database is an important problem. Several research works have proposed several incremental algorithms to deal with this problem.

2.EXISTING WORKS

2.1 FUP [Cheung et al. (1996)]

FUP first scans the incremental part of the dataset and detects (i) the looser single itemsets, i.e. the itemsets that become infrequent due to the inclusion of the incremented part and (ii) it finds the candidate frequent itemsets. Then the whole dataset (i.e. the old and new together) is scanned to find their support in the complete dataset. Next, it performs similar operations iteratively for k-itemsets. Finally, after multiple scanning of the dataset it finds all the maximal frequent sets.

2.2FUP2 [Cheung et al. (1997)]

This algorithm works on a dynamic dataset where new records may be inserted and some of the existing records may be deleted. It extracts the rules from the final dataset by considering both the deleted parts and the newly added part.

2.3Borders [Feldman et al., (1999)]

This algorithm finds the frequent itemsets from the dynamic dataset, using the frequent itemsets already discovered from the old dataset. Here, an infrequent item set is termed as border set if all the non empty proper subsets of it are frequent. Due to the insertion of new records to the dataset, some of the border sets may become frequent, and is termed as promoted border set. For that- the border sets of the old dataset also have to be maintained along with the frequent sets derived. Based on the promoted border set.

2.4 Modified borders [Das and Bhattacharyya, (2005)]

This algorithm is a modified version of the borders algorithm that minimizes unnecessary candidate generations. However, this algorithm uses an additional

user parameter apart from the parameter support count which are sensitive. With proper tuning of these parameters only- a better performance of the algorithm is possible. When this additional parameter's value is closer to the support count, the algorithm converges to the borders algorithm. Depending on this parameter, the border sets has been divided into four different sets B', B'', B''' and B''''. The probability of becoming promoted border set is the highest for the elements of B' and lowest for B''''.

Illustrate through example

	TID	Items
D	**T1**	**A B C**
	T2	**A F**
	T3	**A B C E**
	T4	**A B D F**
	T5	**C F**
	T6	**A B C**
	T7	**A B C E**
	T8	**C D E**
	T9	**B D E**

Table 1
Generate one item set with support count

Item	Support Count
A	**6/9**
B	**6/9**
C	**6/9**
D	**3/9**
E	**4/9**
F	**3/9**

Table 2
Generate frequent one item set with minimum support

Item	Minimum support Count
A	**6/9**
B	**6/9**
C	**6/9**
E	**4/9**

Table 3

Now for two item set use joining in table 3 each item is join with every other item. Now calculate support count for each two item set

Item set	Support Count
AB	**5/9**
AC	**4/9**
AE	**2/9**
BC	**4/9**
BE	**3/9**
CE	**3/9**

Table 4

Delete those item which support count less then the given support count

Item set	Minimum Support Count
A,B	**5/9**
A,C	**4/9**
B,C	**4/9**

Table 5

Now for three item set use joining in table 6. Finally we got three frequent item set with minimum support.

Item set	Support Count
ABC	**4/9**

Table 6

Frequent one item set{A},{B},{C}
Frequent two item set {A,B}, {A,C},{B,C}
Frequent three item set {ABC}
Based on our survey and experimental analysis, it has been observed that:
(1)The algorithms work on market basket encoded data, which causes information loss.
(2)They work in two phases, which causes them computationally expensive.
(3) Some of the algorithms work for incremented dataset only; they cannot handle the updated Dataset due to deletion;
(4)May need multiple scanning of the whole dataset.
(5) Huge number of rules may be generated, based on the user parameters.

3 PROPOSED METHODS

	TID	Item set
D -	**T1**	**A B C**
	T2	**A F**
	T3	**A B C E**
D	**T4**	**A B D F**
	T5	**C F**
	T6	**A B C**
	T7	**A B C E**
	T8	**C D E**

Item
AB
AC
AE
BC
BE
CE

D +	T9	A, B D E
	T10	A,B,D
	T11	D,F
	T12	A,B,C,D

Table 7

So after adding some new transaction the updated data base D' is now for frequent pattern support count 40%

TID	Item set
T4	A B D F
T5	C F
T6	A B C
T7	A B C E
T8	C D E
T9	A, B D E
T10	A,B,D
T11	D,F
T12	A,B,C,D

Table 8

Now we convert the table2 into vertical format resulted table

Item	TID
A	T4, T6, T7, T9, T10, T12
B	T4, T6, T7, T9, T10, T12
C	T5, T6, T7, T8, T12
D	T4, T8, T9, T10, T11, T12
E	T7, T8, T9
F	T4, T5, T11

Table 9

Now assign a code to each row now use Transaction Id Intersection over vertical transaction

Table 10

Delete those row which has support count less then the given support count

Table 11

Item	TID	Row ID	Support Count
A	T4, T6, T7, T9, T10, T12	R1	6
B	T4, T6, T7, T9, T10, T12	R2	6
C	T5, T6, T7, T8, T12	R3	5
D	T4, T8, T9, T10, T11, T12	R4	6
E	T7, T8, T9	R5	3
F	T4, T5, T11	R6	3

Table 12

Now perform intersection between Row ID

Row ID Intersection	Transaction sets	Support count	S/D
R1 ∩ R2 (A,B)	{T4, T6, T7, T9,T10, T12}	5	S
R1 ∩ R3 (A,C)	{T6, T7, T12}	3	D
R1 ∩ R4 (A,D)	{T4, T9, T10, T12}	4	S
R2 ∩ R3 (B,C)	{T6, T7, T12}	3	D
R2 ∩ R4 (B,D)	{T4, T9, T12}	3	D
R3 ∩ R4 (C,D)	{T8, T12}	2	D

Table 13

Delete those row which support count less then the given support count so resulted table

Row ID Intersection	Transaction sets	Support count
R1 ∩ R2 (A,B)	{T4, T6, T7, T9,,T10 T12}	5
R1 ∩ R4 (A,D)	{T4, T9, T10, T12}	4

Table 14

Now finally take intersection between R1, R2 and R3

Row ID Intersection	Transaction sets	Support count
R1 ∩ R2 ∩ R4 {A,B,D}	{T4,T9,T10,T12}	4

Table 15

Frequent one item set {A}, {B},{D}
Frequent two item set {A, B}, {A,D},{B,D}
Frequent three item set {ABD}

4 EXPERIMENT

To evaluate the performance of promising frequent algorithm, algorithm is implemented and tested on a PC with a 2.8 GHz Pentium 4 processor, and 1 GB main memory. The experiments are conducted on a synthetic dataset, called T10I4D10K. The technique for generating the dataset is proposed by Agrawal and etc. [1]. The synthetic dataset comprises 20,0 transactions over 10 unique items, each transaction has 5 items on average Firstly, the proposed algorithm is used to find frequent item set from an original database . Then, several sizes of incremental databases, i.e. 10%, 20%, and 30% of the original database, are added to the original database. For comparison purpose, FUP algorithm is also used to find association rules from the same original database and the same incremental databases. The experimental results with various minimum support thresholds are shown in Table and Figure. From the results, the proposed algorithm has better running time than that of FUP algorithm .

Min_sup	Algorithm	Execution time (sec.)		
		Percent of Incremental database size		
		10%	20%	30%
40%	VTII	145	133	102
	FUP	185	156	127
50%	VTII	134	118	98
	FUP	167	141	124
60%	VTII	133	99	89
	FUP	156	132	101

Table 16

When support count is 40%

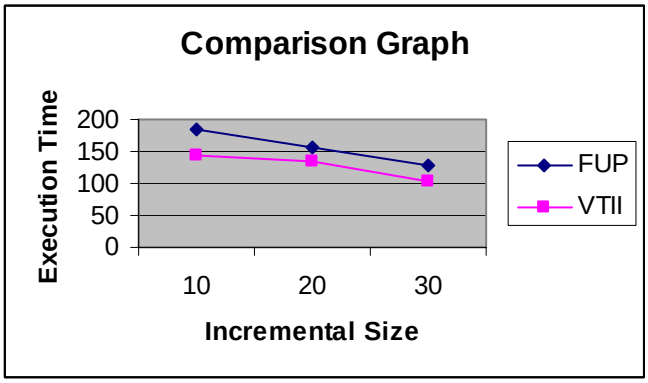

Figure 1

When Support count 50%

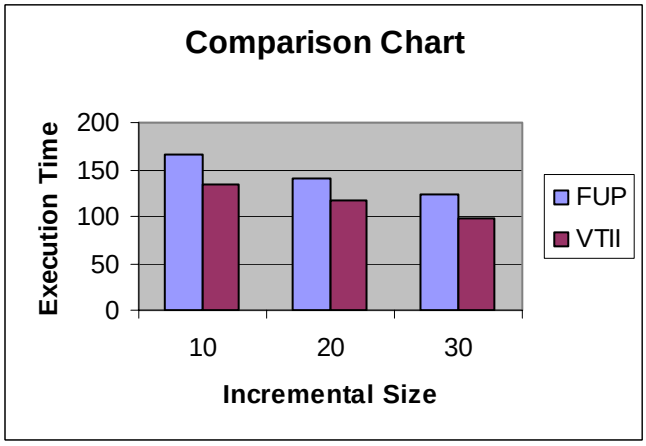

Figure 2

4 REFERENCES

(1) Rahman , Mohammad.M AL-Widyan Philadelphia, "Reduce Scanning Time Incremental Algorithm (RSTIA) of Association rules" Academic Research International2, September Volume 1, Issue 2, September 2011 .
(2) N.L. Sarda N. V. Srinivas "An Adaptive Algorithm for Incremental Mining of Association Rules" (IEEE Xplore)
(3) Siddharth Shah ,N. C. Chauhan ,S. D. Bhander "Incremental Mining of Association Rules" Vol. 3 (3) , 2012,4071-4074. (Journal)
(4) Wei-Guang Teng and Ming-Syan Chen "Incremental Mining on Association Rules ".
(5) Animesh Tripathy1, Subhalaxmi Das2 & Prashanta Kumar Patra3, "An Association Rule Based Algorithmic Approach to Mine Frequent Pattern in Spatial Database System" Vol. 1, No. 2, July-December 2010, pp. 357-363 (Journal).

(6) Sandhya Rani Jetti, Sujatha D, "Mining Frequent Item Sets from incremental database : A single pass approach", Volume 2, Issue 12, December-2011 1 ISSN 2229-5518
(7) Ratchadaporn Amornchewin, "Incremental Association Rule Mining Using Promising Frequent Itemset Algorithm "
(8) Chelliah Balasubramanian*, Karuppaswamy "A mining method for tracking changes in temporal association rules from an encoded database" Vol.1(1), 2009, 1-8 (Journal)
(9) Nibedita Panigrahi Konark, "An Efficient Algorithm for Mining Of frequent items using incremental model", Volume-1, Issue-1,2011
(10) Wuzhou Dong, Juan Yi, Haitao He, Jiadong Ren, "An incremental algorithm for frequent pattern mining based on bit-sequence", Volume3, Number9, October 2011 doi: 10.4156 / ijact.vol3.issue9.4
(11) Ahmed Taha1, Mohamed Taha1, Hamed Nassar2, Tarek F. Gharib3 Journal of Intelligent Learning Systems and Applications "DARM: Decremental Association Rules Mining", 2011,3,181-189 doi:10.4236 / jilsa.2011.
(12) Romanas Tumasonis, Gintautas Dzemyda "A probabilistic algorithm for mining frequent sequences"
(13) Jia-Dong Ren and Xiao-Lei Zhou, "A New Incremental Updating Algorithm for Mining Sequential Patterns", 318-321, 2006 ISSN 1549-3636 (Journal)
(14) Jia-Dong Ren and Xiao-Lei Zhou, "A New Incremental Updating Algorithm for Mining Sequential Patterns ", 318-321, 2006 ISSN 1549-3636 © 2005 Science Publications . (Journal)
(15) Politehnica University of Timisoara, Bd. Vasile Parvan 2, Timisoara, "A Comparative Study of Association Rules Mining Algorithms", WWW: http: / / www.cs.utt.ro / ~stefan
(16) Rakesh Agrawal Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules" .
(17) Maria-Luiza Antonie Osmar R. Za¨ıane "Mining Positive and Negative Association Rules: An Approach for Confined Rules "
(18) Qiankun Zhao Nanyang and Sourav S. Bhowmick Nanyang "Association Rule Mining: A Survey "

Kamlesh Malpani
Ph.D. Scholar, Department of Computer Engineering, Pacific Academy of Higher Education and Research, Udaipur (Raj.)

Dr.Parashu.Ram Pal
PG Department of Computer Application,
Ajay Kumar Garg Engineering College
Ghaziabad, U.P. India