Comparative Study Over Classical Apriori and DSIM Methods

Vijay Kumar Verma, Dr. Kanak Saxena

[1]*Dept. of Computer Science & Engg.*

Lord Krishna College of Technology Indore

[2]*Department of Computer Applications*

Samrat Ashok Technological Institute

Vidisha (M.P.), India

vijayvermaonline@gmail.com

Abstract: *Fining frequent item set is a key issue in data mining; the Apriori algorithms use candidate itemsets to generate Frequent item set , but this approach is highly time-consuming because of self joining and prunining . To look for an algorithm that can avoid the generating of vast volume of candidate itemsets, DSIM (Data-Set Intersection Method) algorithm uses set intersection method to find the maximal frequent itemset.This process is performed by deleting items in infrequent 1-itemset and merging duplicate transaction repeatedly; the process is performed by generating intersections of transactions and deleting unneeded subsets recursively. This algorithm differs from all other methods which are used for discovering maximal frequent itemset.*

Index Terms: *data mining, maximum frequent itemsets, candidate itemsets, intersection*

I. Generating frequent itemsets (frequent patterns) is a key problem in data mining, and it's widely used in applications concerning association rules[9] . Because all frequent itemsets are considered implicitly in the maximum frequent itemset (MFI), the issue of discovering frequent itemset can be converted to the issue of discovering maximal frequent itemset. Besides, only maximal frequent itemset is needed in some of the data mining applications instead of the frequent itemset .

Classical Apriori Algorithm

TID	Itemset
T001	I2, I3, I5, I6, I8
T002	I2, I3, I6, I7, I8
T003	I0, I4, I7
T004	I1, I4, I9
T005	I2, I3, I4, I5, I8
T006	I3, I4, I8, I9
T007	I1, I4, I6, I8
T008	I0, I2, I4, I6
T009	I2, I3, I7
T010	I0, I4, I5, I9

Table I. Data Set (Transactions)

Tid	Itemset	Support
1	I0	3
2	I1	2
3	I2	5
4	I3	5
5	I4	7
6	I5	3
7	I6	4
8	I7	3
9	I8	5
10	I9	3

Table II. Minimum Support Count

Generating 1 item set with support count Form the table II we only take those item in the next State whose support count is 3 or greater than 3. Table III represent the data set with minimum support count

TID	Itemset	Support Count
1	I0	3
2	I2	5
3	I3	5
4	I4	7
5	I5	3
6	I6	4
7	I7	3
8	I8	5
9	I9	3

Table III. (1 Itemset with minimum support)

In table IV we use the concepts of self joining thatI0,is join withI1,I2,I3…I9 similarly we use this concepts forI1,I2…..I9 and we get the table IV with two item set .In table IV we also show The support count for two item set Is shown in table IV

TID	Itemset	Count
1	I0, I2	1
2	I0 , I3	0
3	I0, I4	3
4	I0, I5 I0, I5	1
5	I0, I6	1
6	I0, I7	1
7	I0, I8	0
8	I0,I9	1

Table IV Continutes....

TID	Itemset	Count
9	I2, I3	4
10	I2, I4	2
11	I2, I5	2
12	I2, I6	3
13	I2, I7	2
14	I2, I8	3
15	I2, I9	0
16	I3, I4	2
17	I3, I5	1
18	I3, I6	2
19	I3, I7	2
20	I3, I8	4
21	I3, I9	1
22	I4, I5	2
23	I4, I6	2
24	I4, I7	1
25	I4, I8	2
26	I4, I9	3
27	I5, I6	1
28	I5, I7	0
29	I5, I8	2
30	I5, I9	1
31	I6, I7	1
32	I6, I8	3
32	I6, I9	0
34	I7, I8	1
35	I7, I9	0
36	I8, I9	1

Table IV (2 Itemset with support count)

In table V we takes only those items whose support count is greater the or equal to 3.In Table there are only 7 row with two item set and have the support count 3 or greater.

TID	Item set	Count
1	I0, I4	3
2	I2, I3	4
3	I2 ,I6	3
4	I2 ,I8	3
5	I3 ,I8	4
6	I4 ,I9	3
7	I6 ,I8	3

Table V (2 item set with minimum support count)

Table VI is generated form table V again we are using the concepts of Selfjoing that is we join {I0, I4} with {I2, I3}, {I2, I6}….. {I6, I8}.similarly we use this concepts with other sets .In table VI some row has count(*) . Here from VI we will consider only those item set in the next step whose all subset(two item subset) are present in tabel V otherwise we reject the item set from table VI.

In table VII there are only two row exit because the set{I2,I3,I8} has all its subset in table V Similarly set {I2,I6,I8} has all its sub set in table V .

Here set {I2, I3, I8} set with frequent item set.

TID	Itemset	Count
1	I0 ,I2, I3, I4	*
2	I0, I2, I4 ,I6	*
3	I0, I2, I4 ,I8	*
4	I0, I3,I4, I8	*
5	I0, I4,I9	*
6	I0, I4,I6,I8	*
7	I2, I3, I6	*
8	I2, I3, I8	3
9	I2, I3, I4, I9	*
10	I2, I3, I6, I8	*
11	I2, I6, I8	2
12	I2,I3,I6,I8	*
13	I2,I4 ,I6 ,I9	*
14	I2,I4,I8	*
15	I2,I3,I8	3
16	I2, I4 ,I8 ,I9	*
17	I2,I6,I8	2
18	I3, I4 ,I8,I9	*
19	I3, I6 ,I8	*
20	I4,I9,I6,I8	*

TABLE VII.(3 Itemset with support count)

So the most frequent itemset {I2,I3,I8}

TID	Itemset	Count
1	I2,I3,I8	3
2	I2,I6,I8	2

DSIM Algorithm

This algorithm is divided into two phase the first phase is DISTILLATION of data set and second is INTERSECTION PRUNING. Both are used in order to reduce the length of item set and the volume of data set

A. <u>Distillation of Data Set</u>

Based on length of item set, first screen out the data set in descending order. Then move transactions with support count bigger then minimal support threshold to a frequent item set , and delete all sub set of those transaction to distill the data set

Step 1: Screen out data set, find frequent one item set;

Step 2: Screen out the data set, delete all infrequent 1-itemset from all transactions; then integrate identical transactions. Then sort the data-set in descending order of length of item set to form a new data set denoted as DS.

Step 3: Process every transaction T^a in S with minimum support count greater then threshold. Move these T^a in one set denoted as SF and delete all T^3 (T^3 ,” T^2 3>2)

Step 4: Delete all non-MFI from SF.

Step 5: End

B. <u>Intersection Method</u>

Assume data set denoted by DS and minimum support threshold is $

Step 1: distill data set DS with Distillation data set method; if %DS%< $, end of process. Of current data set.

Step2: Find intersection of Tª and Tn (a < n d" I) ; merge all intersection into a new data set DS1; delete transaction Tn(Tn,"Tª); If %DS1% e" $, then go to step 1 to perform anther intersection pruning method for DS1

Step3: Use the vertical data format of DS to find the intersection of Tj and Ti (j=2, 3, 4, ..., m<n; j<id"n), merge all intersections into a new data-set DS1, go to step 1 to perform another intersection pruning circle for DS1; when the volume of the remaining data-set is less than $, stop finding intersections of Tj and Ti, terminate the process for current data-set.

Step4: End;

C. Illustrate investigation though example

The following example shows how to discover MFI using DSIM for transaction database DS (Table8) with minimum support threshold as 4 .

TID	Itemset
T001	I2, I3, I5, I6, I8
T002	I2, I3, I6, I7, I8
T003	I0, I4, I7
T004	I1, I4, I9
T005	I2, I3, I4, I5, I8
T006	I3, I4, I8, I9
T007	I1, I4, I6, I8
T008	I0, I2, I4, I6
T009	I2, I3, I7
T010	I0, I4, I5, I9

Table 8. Transaction Data-Set DS

Step 1: Distill transaction data set DS using distillation method. Result shown in table 9;

TID	Iteam set	Count
1	I2, I3, I6,I8	2
2	I2, I3, I4, I8	1
3	I4 I6, I8	1
4	I3, I4, I8	1
5	I2, I4, I6	1
6	I2, I3	1

Table 9 Result distillation methods

Step 2: Find intersections of T1 and Ti (i=2, 3... 7), merge all intersections into data-set DS1, as shown in table 10

TID	Iteamset	Count
1	I2, I3, I8	3
2	I6, I8	3
3	I3, I8	3
4	I2, I6	3
5	I2, I3	3

Table 10 Intersection Data-Set for T1 in Table 9

Step 3: Distill the data-sets in Table 10; as this example, the result remains no change.

Step 4: Find intersections of T1 and Ti (i=2, 3, 4, 5) in Table 10 respectively, merge them into a new data-set DS1

TID	Iteamset	Count
1	I3,I8	4
2	I2, I3	4

Because T3 and T5 are subset of T1 in Table 10, delete T3 and T5;

Step 5: Distill the data-set in Table 11, produce frequent itemset {{I3, I8}:4, {I2, I3}:4}; Table 11 is now empty after Distillation;

Step 6: Back to Table 10, T3 and T5 has been deleted, we only need to find the intersection of T2 and T4; but the length of T2 and T4 are both 2, no need to find intersection of them.

Step 7: Back to Table 9, because T6 has been deleted, we only need to find the intersections of T2 and Ti (i=3,4,5,7); merge all intersections into a new data-set DS1,as shown in Table 12

Table 12.Intersection : Data-Set for T2 in Table 2

TID	Iteamset	Count
1	I4, I8	2
2	I4, I3, I8	2
3	I2, I4	2

Because T4 ," T2 in Table 9, it should be deleted.

Step 8: Distill the data-set in Table 10; after Distillation the result is empty;

Step 9: The original data-set DS has 10 transactions; Table 9 shows that 30% (3 transactions) of them has been processed; condense again the remaining data-set in Table 9, and the result is empty. The process ends.

Step 10: Merge all resulting frequent itemsets, and delete all non-frequent maximal itemsets, the final result of MFI is {{I3, I8}: 4, {I2, I3}: 4}. The steps above use 14 times of intersection calculations for MFI; compared with other Apriori-like algorithms, its simplicity and efficiency is explicit.

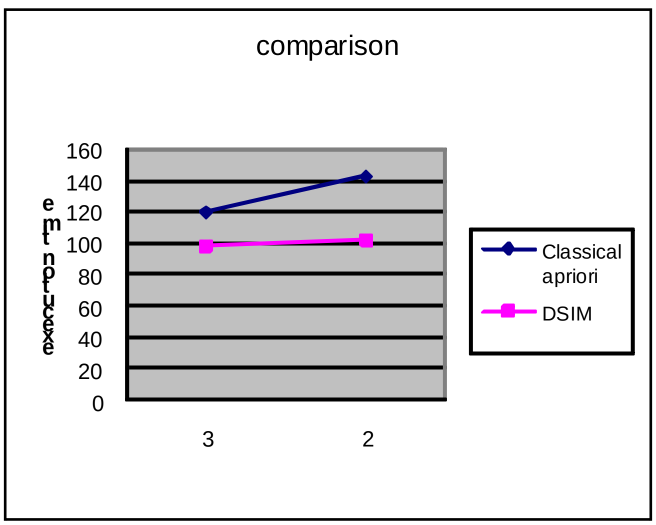
Note: Because the volume of the example data-set DS is small (only 10), the above process does not include the utilizing of vertical data format; the reason of introducing vertical data format is to reduce the number of times of finding the intersections.

Comparison between Classical Apriori algorithm vs DSIM Algorithm Processor P-IV

This table representing the minimum support and execution time for Classical apriori algorithm and DSIM Algorithm :

Minimum support	Time taken to execute (In millisecond) Classical Apriori algorithm	Time taken to execute (In millisecond) DSIM Algorithm
3	**120**	**98**
2	**143**	**102**

Graph representing the comparison of Classical Apriori algorithm and DSIM Algorithm when minimum support varing:



References

[1] Association Rules Mining Algorithm Zhihua Xiao Department of Information System and Computer Science National University of Singapore Lower Kent Ridge Road Singapore

[2] International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, August 2010 DOI :

[3] Association Rules Mining Algorithm Zhihua Xiao Department of Information System and Computer Science National University of Singapore

Lower Kent Ridge Road Singapore119260 xiaozhih@)iscs.nus.edu.sg

[4] Association Rule and Quantitative Association Rule Mining among Infrequent Items Ling Zhou Stephen Yau Dept. of Mathematics, Statistics and Computer Science Dept. of Mathematics, Statistics and Computer Science University of Illinois at Chicago University of Illinois at Chicago 773-378-0126 312-996-3065 lzhou5@uic.edu yau@uic.edu

[5] A Comparative Study of Association Rules Mining Algorithms Cornelia Gyõrödi*, Robert Gyõrödi*, prof. dr. ing. Stefan Holban** *Department of Computer Science, University of Oradea, Str. Armatei Romane 5, 3700, Oradea, Romania, Phone: +40 (0) 59 432-830, e-mail: rgyorodi@rdsor.ro

[6] An Implementation of Frequent Pattern Mining Algorithm using Dynamic Function Sunil Joshi Department of Computer Applications Samrat Ashok Technological Institute Vidisha (M.P.), India Dr. R. S . Jadon Department of Computer Applications Madhav Institute of Technology and Science Gwalior (M.P.), India Dr. R. C. Jain Professor, Director Samrat Ashok Technological Institute Vidisha (M.P.), India

[7] Generalized Association Rule Mining Using Genetic Algorithms Peter P. Wakabi-Waiswa, Venansius Baryamureeba and K. Sarukesi

[8] Mining Positive and Negative Association Rules: An Approach for Confined Rules Maria-Luiza Antonie Osmar R. Za¨ýane

[9] Improved Association Mining Algorithm for Large Dataset IJCEM International Journal of Computational Engineering & Management, Vol. 13, July 2011 ISSN (Online): 2230-7893

[10] An Algorithm for Frequent Pattern Mining Based On Apriori Goswami D.N.*, Chaturvedi Anshu. **Raghuvanshi C.S.***

[11] An Improved Apriori-based Algorithm for Association Rules Mining 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery

[12] Fast Algorithms for Mining Association Rules Rakesh Agrawal Ramakrishnan Srikant

[13] Induction of Association Rules: Apriori Implementation Christian Borgelt and Rudolf Kruse

[14] Efficient Mining First-Order Frequent Patterns Jan Blat´ak Masaryk University, Brno, Czech Republic.