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

ABSTRACT
An Important Problem in Data Mining in Various Fields like Medicine, Telecommunications and World Wide Web is Discovering Patterns. Frequent patterns mining is the focused research topic in association rule analysis. Apriori algorithm is a classical algorithm of association rule mining. Lots of algorithms for mining association rules and their mutations are proposed on basis of Apriori Algorithm. Most of the previous studies adopt Apriori-like algorithms which generate-and-test candidates and improving algorithm strategy and structure but no one concentrate on the structure of database. A simple approach is if we implement in Transposed database then result is very fast. Recently, different works proposed a new way to mine patterns in transposed databases where a database with thousands of attributes but only tens of objects. In this case, mining the transposed database runs through a smaller search space. In this paper, we systematically explore the search space of frequent patterns mining and represent database in transposed form. We developed an algorithm (termed DFPMT—A 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. Our solutions provide faster result. A quantitative exploration of these tradeoffs is conducted through an extensive experimental study on synthetic and real-life data sets.

Keywords- Longest Common Subsequence, Transposition of Database,Frequent Pattern mining

I. INTRODUCTION
Frequent Pattern Mining is most powerful problem in association mining. Most of the algorithms are based on algorithm is a classical algorithm of association rule mining [2, 3, 4]. Lots of algorithms for mining association rules and their mutations are proposed on basis of Apriori Algorithm [2, 3, 4, 5, 6]. Most of the previous studies adopt Apriori-like algorithms, which generate-and-test candidates and improving algorithm strategy and structure. Several modifications on apriori algorithm are focused on algorithm Strategy but no one-algorithm emphasis on representation of database. A simple approach is if we implement in Transposed database then result is very fast. Recently, different works proposed a new way to mine patterns in transposed databases where a database with thousands of attributes but only tens of objects [2]. In many example attribute are very large than objects or transaction In this case, mining the "transposed" database runs through a smaller search space. In apriori algorithm each phase is count the support of prune pattern candidate from database. No one algorithm filters or reduces the database in each pass of apriori algorithm to count the support of prune pattern candidate from database. We propose a new dynamic algorithm for frequent pattern mining in which database represented in transposed form. And for counting the support we find out by longest common subsequence approach and after finding pattern longest common subsequence is stored or update in database so that next time instead of whole transaction we search from these filter transaction string.

II. FREQUENT PATERN MINING
Frequent Itemset Mining came from efforts to discover useful patterns in customers' transaction databases. A customers' transaction database is a sequence of transactions (T = t1. . . tn), where each transaction is an itemset (ti ⊆ I). An itemset with k elements is called a k-itemset. In the rest of the paper we make the (realistic) assumption that the items are from an ordered set, and transactions are stored as sorted itemsets. The support of an itemset X in T, denoted as suppT(X), is the number of those transactions that contain X, i.e. suppT(X) = |{tj : X ⊆ tj}|. An itemset is frequent if its support is greater than a support threshold, originally denoted by min supp. The frequent itemset mining problem is to find all frequent itemset in a given transaction database.

The first, and maybe the most important solution for finding frequent itemsets, is the APRIORI algorithm [5,7]. Later faster and more sophisticated algorithms have been Suggested, most of them being modifications of APRIORI [5, 7]. Therefore if we improve the APRIORI algorithm then we improve a whole family of algorithms. We assume that the reader is familiar with APRIORI [1, 2, 3, 4, 5, 6, 7] and we turn our attention to its central data structure.

Most of these algorithms adopt an Apriori-like method: generates a candidate pattern by extending currently frequent pattern and then test the candidate. During this process, many infrequent patterns are generated.

III. TRANSPOSITION OF DATABASE

To avoid confusion between rows (or columns) of the original database and rows (columns) of the "transposed" database, we define a database as a relation between original and transposed representations of a database in Table-1. The attributes are A = {a1, a2, a3, a4} and the objects are O = {o1, o2, o3}. We use a string notation for object sets or itemsets, e.g., a1a3a4 denotes the itemset {a1, a3, a4} and o2o3 denotes the object set {o2, o3}. This dataset is used in all the examples between two sets: a set of attributes and a set of objects.

IV. DYNAMIC FUNCTION

The longest common subsequence problem is one of the common problems which can be solved efficiently using dynamic programming. "The Longest common subsequence problem is, we are given two sequences X=<x1,x2----------xn> and Y=<y1,y2---------ym> and wish to find a maximum length common subsequence of X and Y" for example : if X=<A,B,C,B,D,A,B> and Y=<B,D,C,A,B,A> then The sequence <B, C, B, A> longest common subsequence. Let us define C [i, j] to be the length of an LCS of the sequences xi and yj. If either i=0 or j=0, one of the sequence has length 0, so the LCS has length 0. The Optimal substructure of the LCS Problem gives the recursive formula in fig.1

TABLE I. TRANSPOSITION OF DATABASE

DATABASE D

Object	Attribute Pattern
O1	a1a2a3
O2	a1a2a3
O3	a2a3a4

TRANSPOSED DATABASE D^T

Attribute	Object Pattern
a1	O1O2
a2	O1O2O3
a3	O1O2O3
a4	O3

$$C(i,j)=\begin{cases}0 & \text{if } i=0 \text{ or } j=0\\ C(i-1,j-1)+1 & \text{if } i,j>0 \text{ and } xi=yj\\ \max(c(i,j-1),c(i-1,j)) & \text{if } i,j>0 \text{ and } xi\neq yj\end{cases}$$

Figure 1. Longest Common Subsequence Recursive Formula

V. ALGORITHM

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

A. *Algorithm DAPS (Algorithm for Pruning with Support)*

I. Compute k-1 Subset of k-itemset
II. Generate Itemset Transition string from Filter Transposed Database
III. Computer LCS for each item in itemset using Transition string.
IV. If length of LCS >∂ then itemset is frequent.

B. *Algorithm DFPMT (Dynamic Approach for Frequent Patterns Mining Using Transposition of Database)*

I. Convert Database in Transpose Form D^T
II. Compute F1 of all Frequent Items
III. C1:= D^T (Only Frequent Item row with Transition id string)
IV. K: =2.
V. While Lk-1≠ { } do
VI. Compute Ck of all candidate k-1 Itemsets
// New Algorithm which prune and count support using LCS //
VII. Compute Lk=DAPS (Ck)
VIII. K:=K+1

VI. EXPLANATION WITH EXAMPLE WHICH SUPPORT THE ARGUMENTS

Study the following transaction database .A={A1,A2,A3,A4,A5,A6,A7,A8,A9>, Assume σ=20%, Since T contains 15 records, it means that an itemset that is supported by at least three transactions is a frequent set and output shown in fig. 2.

TABLE II. GIVEN DATASET

A1	A2	A3	A4	A5	A6	A7	A8	A9
1	0	0	0	1	1	0	0	0
0	1	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0
1	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0	0

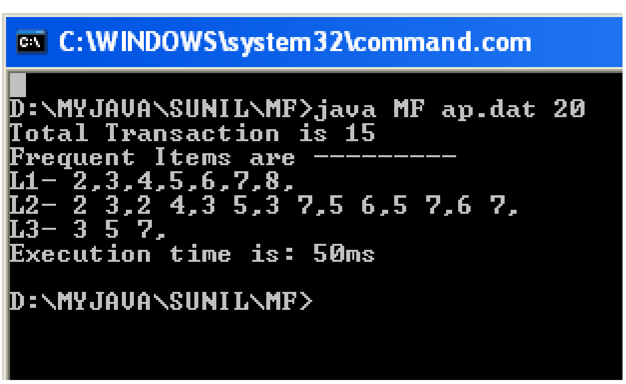

Figure 2. Frequent Pattern with support 20%

TABLE III. TRANSPOSITION OF DATABASE WITH TRANSACTION ID

Item Id	Transaction id String	Count
1	1,14	2
2	2,4,6,7,13,15	6
3	4,6,10,11,14,15	6
4	2,3,6,13	4
5	1,3,5,8,10,11,12,14	8
6	1,5,7,12,13	5
7	3,5,7,10,11,13,14	7
8	1,2,9,12	4
9	7,15	2

Now Apply Algorithm

1) Pass 1

TABLE IV. FREQUENT ITEM SET

Item Id	Transaction id String	Count
2	2,4,6,7,13,15	6
3	4,6,10,11,14,15	6
4	2,3,6,13	4
5	1,3,5,8,10,11,12,14	8
6	1,5,7,12,13	5
7	3,5,7,10,11,13,14	7
8	1,2,9,12	4

L1 :={ 2, 3, 4, 5, 6, 7, 8}

2) Pass 2

Generate candidate for k=2

C2:={{2,3},{2,4},{2,5},{2,6},{2,7},{2,8},{3,4},{3,5},{3,6},{3,7}, {3,8},{4,5},{4,6},{4,7},{4,8},{5,6},{5,7},{5,8}, {6, 7}, {6, 8}, {7, 8}}

After Apply DAPS Algorithm

TABLE V. RESULT OF DAPS ALGORITHM

Item Id	Transaction id String	Count
2,3	4, 6,15	3
2,4	2, 6,13	3
3,5	10,11,14	3
3,7	10,11,14	3
5,6	1,5,12	3
5,7	3,5,10,11,14	5
6,7	5,7,13	3

L2:= {(2,3),(2,4),(3,5),(3,7),(5,6),(5,7), (6,7)}

3) Pass 3

Generate Candidate for k: = 3

C3:={{2,3,4},{3,5,7},{5,6,7}}

After Apply DAPS Algorithm

Item Id	Transaction id String	Count
3,5,7	10,11,14	3
5,6,7	5	1

L3 :={(3, 5, 7)}

L: =L1UL2UL3

VII. EXPERIMENTAL RESULTS

In this section we performed a set of experiments to evaluate the effectiveness of the frequent pattern mining using dynamic function method. The algorithm DFPMT was executed on a Pentium 4 CPU, 2.4GHz, and 10 GB of RAM computer. It was implemented in Java. The experiment database sources are T40I4D100K, provided by the QUEST generator of data generated from IBM's Almaden lab. The experimental dataset consists of two kinds of data whose records are set to 5K and 100K the testing results of experiments are showed in Fig.3.

In the Fig.3, the horizontal axis represents the number of support in database and the vertical axis represents mining time

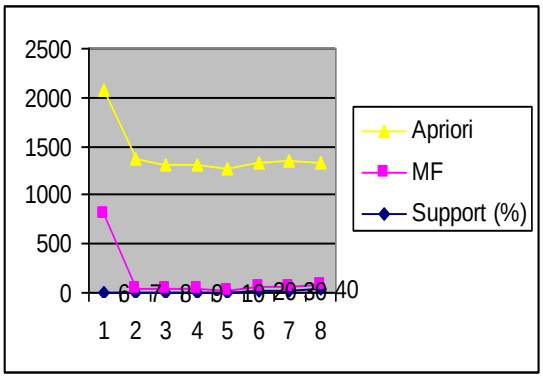

Figure 3. The test results of apriori and DFPMT

(Second) of the same database with the algorithm Apriori And DFPMT. The two curves denote different time cost of the algorithm Apriori and DFPMT with different minsup.

VIII. CONCLUSION

Determining frequent objects (item sets, episodes, sequential Patterns) is one of the most important fields of data mining. It is well known that the way candidates are defined has great effect on running time and memory need, and this is the reason for the large number of algorithms. We presented a new research trend on frequent pattern mining is expecting transpose representation to relieve current methods from the traditional bottleneck, providing scalability to massive Data sets and improving response time. In order to mine patterns in databases with more columns than rows, we proposed a complete framework for the transposition: we gave the item set in the transposed database of the transposition of many classical transactions ID. Then we gave a strategy to use this framework to mine all the itemset satisfying.

We used dynamic approach which is better than tradition approach for finding longest common subsequence. We also presented a new research trend on filtering the database in each iteration. Our implementation can be further improved if parallelism is used to store reduced pattern. Further investigations are needed to clear the possibilities of this technique.

ACKNOWLEDGMENT

We thank Sh. Jitendra Agrawal and Sh. K K Shrivastava for discussing and giving us advice on its implementation.

REFERENCES

[1] Sunil Joshi et al: accepted research paper in The **IEEE** 2010 International Conference on Communication software and Netweorks (ICCSN 2010) ***on "A Dynamic Approach for Frequent Pattern Mining Using Transposition of Database"*** *from* 26 - 28 February 2010

[2] B. Jeudy and F. Rioult, Database transposition for constrained closed pattern mining, in: Proceedings of Third International Workshop on Knowledge Discovery in Inductive Databases (KDID) co-located with ECML/PKDD, 2004.

[3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules,In Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp. 487–499.

[4] J. Han, Research challenges for data mining in science and engineering. In NGDM 2007.

[5] R. Agrawal, R. Srikant, Mining sequential patterns, In Proceedings of the 11th International Conference on Data Engineering, 1995, pp. 3

[6] A fast APRIORI implementation Ferenc Bodon∗ Informatics Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences H-1111 Budapest, L´agym´anyosi u. 11, Hungary

[7] B. Goethals. Survey on frequent pattern mining. Technical report, Helsinki Institute for Information Technology,03.

[8] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *The International Conference on Very Large Databases*, pages 487–499, 1994.

[9] Improving Frequent Patterns Mining by LFP XU Yusheng, MA Zhixin, CHEN Xiaoyun, LI Lian School of Information Science and Engineering Lanzhou University Lanzhou, China, 730000 e-mail:{xuyusheng, mazhx, chenxy, lil}@lzu.edu.cn Tharam S. Dillon School of Information System Curtin University Perth, Australia

[10] Finding Longest Increasing and Common Subsequences in Streaming Data David Liben-Nowell_ y dln@theory.lcs.mit.edu Erik Vee_ z env@cs.washington.edu An Zhu_ x anzhu@cs.stanford.edu November 26, 2003

[11] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. *In Proc. of the ACM SIGMOD Conference on Management of Data*, pages 207–216, 1993.

[12] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328, 1996.

[13] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. L. P. Chen, editors, *Proc. 11th Int. Conf. Data Engineering, ICDE*, pages 3–14. IEEE Press, 6–10 1995.

[14] F. Bodon and L. R´onyai. Trie: an alternative data structure for data mining algorithms. *to appear in Computers and Mathematics with Applications*, 2003.

[15] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Record (ACM Special Interest Group on Management of Data),26(2):255*, 1997.

[16] D. W.-L. Cheung, J. Han, V. Ng, and C. Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *ICDE*, pages 106–114, 1996.

[17] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. Naughton, and P. A. Bernstein, editors, *2000 ACM SIGMOD Intl. Conference onManagement of Data*, pages 1–12. ACM Press, 05 2000.

[18] R. Agarwal, C. Aggarwal, and V. V. V. Prasad: A Tree Projection Algorithm for Generation of Frequent Itemsets.*Journal of Parallel and Distributed Computing* (special issue on high performance data mining), (to appear), 2000.

[19] R. Agrawal, T. Imielinski, and R. Srikant: Mining association rules between sets of items in large databases. SIGMOD, May 1993.

[20] D. Burdick, M. Calimlim, J. Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In Proc. of 17th Int'l Conf. on Data Engineering, pp. 443-452, 2001.

[21] Efficient Mining of Weighted Frequent Pattern Ove Data Streams Farhan Ahmed,Tanbeer 2009

AUTHORS

Sunil Joshi is presently working as a Ass. Professor, Computer Applications at Samrat Ashok Technological Institute Vidisha (M.P). He has 9 years teaching experience and 2 years research experience. His research areas include Data mining.

Dr. R S Jadon is presently working as a Head, Computer Applications at Madhav Institute of Technology and Science, Gwalior. He has 12 years research experience. He has presented research papers in more than 30 national and international conferences
And published more than 30 papers in national and international journals. His research areas include Video Data Processing.

Dr. R. C. Jain is presently working as a Director and Head, Computer Applications at Samrat Ashok Technological Institute Vidisha He has 30 years teaching experience and 15 years research experience. He has presented research papers in more than 100 national and international Conferences and published more than 100 papers in national and international journals. His research areas include Data mining and Network security.