ISSN 1512-1232

An Efficient Algorithms for Generating Frequent Pattern Using Logical Table With AND, OR Operation

Kamlesh Malpani[1], P. R. Pal[2]

[1]PG Department of Computer Science, Shri Vaishnav Institute of Management
Indore, M.P. India, Email: malpani_k1@rediffmail.com

[2]Department of Computer Applications, Ajay Kumar Garg Engineering College
Ghaziabad, U.P. India, E-mail: prpal@rediffmail.com

Abstract:

Frequent Pattern Mining plays an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, Market basket analysis is a useful method of discovering customer purchasing patterns by extracting associations or co-occurrences in transactional databases[1]. Information obtained from the analysis can be used in marketing, sales, service, and operational strategies, In this paper, we propose a new algorithm based Logical Operation (AND,OR). In this algorithms we are using simple Logical operation (AND, OR) on data set containing items. We use simple table to perform AND, OR operation to avoid joining and pruning. The advantage of this new technique is fast operation on dataset containing items and provides facilities to avoid unnecessary scans to the database

1 Introduction

Data mining is the process of extracting patterns from data. It is becoming as an increasingly important tool to transform these data into information. Frequent itemsets mining is a popular and important, first step in data mining for analyzing data sets across a broad range of applications. It plays an essential role in many important data mining tasks.

Let I = { I1, I2, I3, …, Im} be a set of items. Let D be the transactional database,

where each transaction T is a set of items such that T ⊆ I. Each transaction is associated with an

identifier TID[3]. A set of items is referred as itemset. An itemset that contains K items is a K-itemset. The number of transactions in which a particular itemset exists gives the support or frequency count or count of the itemset. If the support of an itemset I satisfies the minimum support threshold, then the itemset I is a frequent itemset.

Classified based on the completeness of patterns to be mined, the levels of abstraction involved in the rule set, the number of data dimensions involved in the rule, the types of values handled in the rule, the kinds of rules to be mined, the kinds of patterns to be mined. The classification of algorithms for frequent itemset mining is Apriori-like algorithms, frequent pattern growth based algorithms It is impractical to generate the entire set of frequent itemsets for the very large databases . There is much research on methods for generating all frequent itemsets efficiently[3]. Most of these algorithms use a breadth-first approach, i.e. finding all k-itemsets before considering (k+1) itemsets. The performance of all these algorithms gradually degrades with dense datasets.

The main drawback of frequent itemsets is they are very large in number to compute or store in computer. This leads to the introductions of closed frequent itemsets and maximal frequent itemsets. An itemset X is closed in a data set S if there exists no proper superitemset Y such that Y has the same support count as X in S. An itemset X is closed frequent itemset in set S if X is closed and frequent in S. an itemset X is a maximal frequent

ISSN 1512-1232

itemset in set S if X is frequent and there exists no super-itemset Y such that X Y and Y is frequent in.

S. Maximal frequent itemset mining is efficient in terms of time and space when compared to frequent itemsets and closed frequent itemsets because both are subsets of maximal frequent itemset. Some of the algorithms developed for mining maximal frequent

2 Apriori Algorithms

Apriori algorithm is an influential algorithm for mining frequent itemsets for Boolean association rules. It uses a Level-wise search, where k-itemsets (Anitemset that contains k items is a k- itemset) are used to explore (k+1)-itemsets, to mine frequent itemsets from transactional database for Boolean association rules[4].

First, the set of frequent 1-itemsets is found. This set is denoted L1. L1 is used to find L2, the frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database.

Apriori property: All non-empty subsets of a frequent itemset must also be frequent.

It performs the following tasks:

1. Reducing the search space to avoid finding of each Lk requires one full scan of the database

2. If an itemset I does not satisfy the minimum support threshold, min_sup, the I is not frequent, that is, P (I) < min_sup

3. If an item A is added to the itemset I, then the resulting itemset (i.e., IUA) cannot occur more frequently than I. Therefore, I UA is not frequent either, that is, P (I UA) < min_sup.

A two step process is followed, consisting of join and prune actions[5].

1. The join step: To find Lk, a set of candidate k-itemsets is generated by joining Lk-1 with itself. This set of candidates is denoted Ck. The join, Lk-1 with Lk_1, is performed, where members of Lk-1 are joinable if they have (k_2) items in common.

2. The prune step: Ck is a superset of Lk, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in Ck. A scan of the database to determine the count of each candidate in Ck would result in the determination of Lk (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to Lk). Ck, however, can be huge, and so this could involve heavy computation. To reduce the size of Ck, the Apriori property is used as follows. Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (k-1)-subset of a candidate k-itemset is not in Lk_1, then the candidate cannot be frequent either and so can be removed from Ck. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets[5].

2.1 Limitations

1. The algorithm is of low efficiency, such as firstly it needs to repeatedly scan the database, which spends much in I/O.

2. Secondly, it create a large number of 2- candidate itemsets during outputting frequent 2-itemsets.

3. Thirdly, it doesn't cancel the useless itemsets during outputting frequent k-itemsets.

2.2 Methods to Improve Apriori's Efficiency

•*Hash-based itemset counting*: A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent.

ISSN 1512-1232

•*Transaction reduction*: A transaction that does not contain any frequent k-itemsetis useless in subsequent scans.

•*Partitioning:* Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB.

•*Sampling*: mining on a subset of given data, lower support threshold + a method to determine the completeness.

•*Dynamic itemset counting*: add new candidate itemsets only when all of their subsets are estimated to be frequent.

3 Proposed Algorithm

Our algorithm is an effective algorithm for mining association rules in large databases .Like the Apriori algorithm, our algorithm mines association rules in two steps. In the first step compute frequent itemsets using logic OR and AND operations. The Implemented algorithm gains significant performance improvement over the Apriori algorithm.

3.1 Generation of Frequent Itemsets

The implemented algorithm generates frequent itemsets through evolutionary iterations based on two tables, the item details table and the transaction table.

3.2 Transforming a transaction details into a logical Table

The Logical table with element values of „1 or „0 ,where items are present in the transaction means 1 otherwise 0. Finally, a column vector Ck is utilized to store the reference count of all frequent k-itemsets in the kth iteration.The reference count on a k-itemset can be obtained by counting the number of 1 s in the corresponding row of logical table .

3.3 Generation of Frequent k-itemsets

Frequent k-itemsets can be generated through the following iteration:

Repeat

1. Read a pair of different rows from a logical table.
2. go to step 3 (i.e., until a new k-itemset has been found).
3. Performing AND operation on the two rows of Logical table, correspond to the rows of step2.

The result shows that, which transactions contain this new k- itemset. And then counting the number of 1 s in the result to get the reference count of this new k-itemset. If the count is less than the number of transactions required by the minimum support, the new k-itemset is discarded.

After the generation of frequent k-itemset, the logical table of the k-itemset and its corresponding reference count vector are *kept in frequent itemset table for generating association rules.*

4 llustrative example

4.1 Ariori Algorithms

ISSN 1512-1232

TID	List of Items
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1, I2,I3

Table 1 Table with 1 item set and support count

Item Set	Sup_ Count
I1	6
I2	7
I3	6
I4	2
I5	2

Table 2 Table with 1 item set and minimum support count

Item Set	Sup_ Count
I1	6
I2	7
I3	6
I4	2
I5	2

Table 3

Item Set
I1,I2
I1,I3
I1,I4
I1,I5
I2,I3
I2,I4
I2,I5
I3,I4
I3,I5
I4,I5

Table 4 Table with 2 item set

Item Set	Sup_Count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

Table 5 Table with 2item set and support count

Item Set	Sup_Count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

Table 6 Table with 2 itemset and minimum support count

Itemset
I1,I2,I3
I1,I2,I5

Table 7 with frequent item set

4.2 Proposed Algorithms

ISSN 1512-1232

Table containing transactions

TID	List of Items
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1, I2,I3

Table 8

	Items				
	I1	I2	I3	I4	I5
I1	1	0	0	0	0
I2	0	1	0	0	0
I3	0	0	1	0	0
I4	0	0	0	1	0
I5	0	0	0	0	1
T100	1	1	0	0	1
T200	0	1	0	1	0
T300	0	1	1	0	0
T400	1	1	0	1	0
T500	1	0	1	0	0
T600	0	1	1	0	0
T700	1	0	1	0	0
T800	1	1	1	0	1
T900	1	1	1	0	0
Support Count	6	7	6	2	2

Table 9 Table with item present or absent in transaction Also with their support count

Two Item set	I1, I2	I1, I3	I1, I5	I2, I3	I2, I4	I2 , I5
I1	1	1	1	0	0	0
I2	1	0	0	1	1	1
I3	0	1	0	1	0	0
I4	0	0	0	0	1	0
I5	0	0	1	0	0	1
T100	1	0	1	0	0	1
T200	0	0	0	0	1	0
T300	0	0	0	1	0	0
T400	1	0	0	0	1	0
T500	0	1	0	0	0	0
T600	0	0	0	1	0	0
T700	0	1	0	0	0	0
T800	1	1	1	1	0	1
T900	1	1	0	1	0	0
Sup count	4	4	2	4	2	2

Table 10 Table with 2 Itemset with support count

Three item set	I1,I2,I3	I1,I2,I5
I1	1	1
I2	1	1
I3	1	0
I4	0	0
I5	0	1
T100	0	1
T200	0	0
T300	0	0
T400	0	0
T500	0	0
T600	0	0
T700	0	0
T800	1	1
T900	1	0
Sup Count	2	2

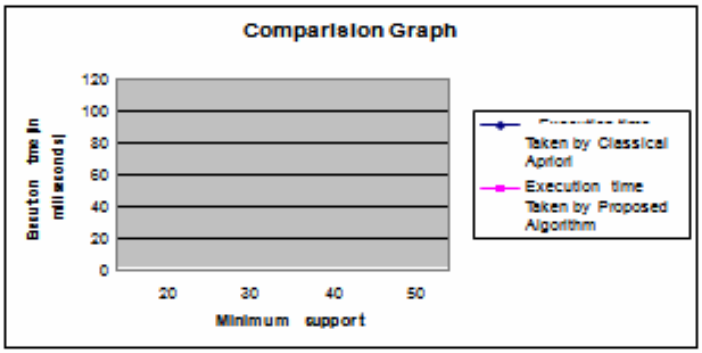
Table 11 Table with frequent itemset

5 Comparison with graph

In order to show the performance of the proposed algorithm, we conducted an experiment

ISSN 1512-1232

using the Apriori algorithm and proposed algorithm. The algorithms were implemented in Dot Net and tested on a Windows XP platform. The test database are taken from easyday shopping mall The number of items N is set to 50; D is the number of transactions; T is the averages size of transactions, and I is the average size of the maximum frequent itemsets. Graph show results for different numbers of minimum supports. The results show that the performance of our algorithm is much better than that of the Apriori algorithm. This is because the greater the minimum support, the more less candidate itemsets the Apriori algorithm has to determine, and also the Apriori algorithm join and pruning processes take more time to execute. However, and it spends less time calculating k- supports with the logical item table .



6 Conclusions

The most common application of association rule mining is market basket analysis. In this paper, An Efficient algorithm for mining association rules using Logical Table based approach is proposed. The main features of this algorithm are that it only scans the transaction database once, it does not produce candidate itemsets, and In addition, it stores all transaction data in bits, so it needs less memory space and can be applied to mining large databases

References

1. Frequent Itemset Generation Using Hashing-Quadratic Probing Technique by M. Krishnamurthy European Journal of Scientific ResearchISSN 1450-216X Vol.50 No.4 (2011), pp. 523-532
2. Improved Association Mining Algorithm for Large Dataset Tannu Arora1, Rahul Yadav 2 IJCEM International Journal of Computational Engineering & Management, Vol. 13, July 2011 ISSN (Online): 2230-7893 www.IJCEM.org
3. A Fast Algorithm for Mining Multilevel Association Rule Based on Boolean Matrix 1Pratima Gautam 2 K. R. Pardasani (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, 746-752
4. Proposing an Efficient Method for Frequent Pattern Mining Vaibhav Kant Singh, Vijay Shah, Yogendra Kumar Jain, Anupam Shukla, A.S. Thoke, Vinay Kumar Singh, Chhaya Dule, Vivek Parganiha
5. An Efficient Data Mining Approach on Compressed Transactions Jia-Yu Dai, Don-Lin Yang, Jungpin Wu, and Ming-Chuan Hung PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, VOLUME 30, JULY 2008, ISSN 1307-688433
6. An Improved Apriori-based Algorithm for Association Rules Mining Huan Wu, Zhigang Lu, Lin Pan, Rongsheng Xu Wenbao Jiang 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery

ISSN 1512-1232

7. An Algorithm for Frequent Pattern Mining Based On Apriori Goswami D.N.*, Chaturvedi Anshu. **Raghuvanshi C.S.*** (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 942-947
8. An Efficient Algorithm for Mining Of frequent items using incremental model Prof. Dr.Prashant Patnaik Mr. Sanjay Padhi
9. Mining Dynamic Databases using Probability-Based Incremental Association Rule Discovery Algorithm by Ratchadaporn Amornchewin Journal of Universal Computer Science, vol. 15, no. 12 (2009), 2409-2428 submitted: 15/12/08, accepted: 25/6/09, appeared: 28/6/09 NJ.UCS
10. DARM: Decremental Association Rules Mining Ahmed Taha1, Mohamed Taha1, Hamed Nassar2, Tarek F. Applications, 2011, 3, 181-189 doi:10.4236/jilsa.2011.33019, Published Online August2011 (http://www.SciRP.org/journal/jilsa)
11. GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets KARAM GOUDA karam g@hotmail.com Data Mining and Knowledge Discovery, 11, 1–20, 2005 _c 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands. Department of Mathematics, Faculty of Science, Benha, Egypt, MOHAMMED J. ZAK
12. Mining Positive and Negative Association Rules: An Approach for Confined Rules Maria-Luiza Antonie Osmar R.Za¨ıane Department of Computing Science, University of Alberta
13. An incremental algorithm for frequent pattern mining based on bit-sequence by Wuzhou Dong, Juan Yi, Haitao He, Jiadong Ren
14. algorithm for mining time varying frequent itemsets d.sujatha1, prof.b.l.deekshatulu2 Journal of Theoretical and Applied Information Technology © 2005 - 2009 JATIT. All rights reserved.
15. Mining Dynamic Databases using Probability-Based Incremental Association Rule Discovery Algorithm Ratchadaporn AmornchewinJournalof Universal Computer Science, vol.15, no.12 (2009), 2409-2428 submitted: 15/12/08, accepted: 25/6/09, appeared: 28/6/09, NJ.UCS Comparison

Article received: 2012-06-04