Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 5 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

An Improved Apriori Algorithm based on Matrix Data Structure

By Shalini Dutt, Naveen Choudhary & Dharm Singh

Maharana Pratap University of Agriculture and Technology, India

Abstract- Mining regular/frequent itemsets is very important concept in association rule mining which shows association among the variables in huge database. the classical algorithm used for extracting regular itemsets faces two fatal deficiencies .firstly it scans the database multiple times and secondly it generates large number of irregular itemsets hence increases spatial and temporal complexties and overall decreases the efficiency of classical apriori algorithm.to overcome the limitations of classical algorithm we proposed an improved algorithm in this paper with a aim of minimizing the temporal and spatial complexities by cutting off the database scans to one by generating compressed data structure bit matrix(b_matrix)-and by reducing redundant computations for extracting regular itemsets using top down method. theoritical analysis and experimental results shows that improved algorithm is better than classical apriori algorithm.

Keywords: *Apriori algorithm, frequent itemsets, association rule.*

GJCST-C Classification : *E.1*

Strictly as per the compliance and regulations of:

An Improved Apriori Algorithm based on Matrix Data Structure

Shalini Dutt [α], Naveen Choudhary [σ] & Dharm Singh [ρ]

Abstract- Mining regular/frequent itemsets is very important concept in association rule mining which shows association among the variables in huge database. The classical algorithm used for extracting regular itemsets faces two fatal deficiencies .firstly it scans the database multiple times and secondly it generates large number of irregular itemsets hence increases spatial and temporal complexities and overall decreases the efficiency of classical apriori algorithm. to overcome the limitations of classical algorithm we proposed an improved algorithm in this paper with a aim of minimizing the temporal and spatial complexities by cutting off the database scans to one by generating compressed data structure bit matrix(b_matrix)-and by reducing redundant computations for extracting regular itemsets using top down method. Theoretical analysis and experimental results shows that improved algorithm is better than classical apriori algorithm.

Keywords: apriori algorithm, frequent itemsets, association rule.

I. Introduction

In last few decades data has become so vast that extracting information from this huge data becomes very important issue in data mining . hence data mining brought into scene from last few years. Mining association rules is important process in data mining which shows relationship among the variables or affairs stored in data warehouse, database and other information repositories. Association rule mining is two step process. First it generates regular/frequent itemset set of items having count equal or greater than user specified parameter i.e., minimum support and second it discovers association rules from these frequent itemsets. In this regard first association rule mining algorithm apriori algorithm was proposed in 1994 to discover regular itemset. Limitations of apriori results in lot of research in the field of data mining to build more efficient algorithms in respect of space and time. This paper puts forward an improved algorithm using matrix data structure with simply counting rows and columns and transaction reduction strategies using top down approach for finding out largest regular itemset to smallest regular itemset.

In this way, it can greatly reduces complexity and increases the efficiency of improved algorithm.

Author α σ ρ: Department of Computer Science and Engineering, College of Technology and Engineering, Maharana Pratap University of Agriculture and Technology, Udaipur, Rajasthan, India.
e-mail: sd.dutt7@gmail.com, naveenc121@yahoo.com, singhdharm@hotmail.com

The remaining section of this paper is organized as follows: Section 2 contains Apriori Algorithm. In section 3, elaborate the proposed improved algorithm for extracting regular itemset and. Section 4 contains experimental results and conclusion.

II. Apriori Algorithm

The apriori algorithm is standard and classical algorithm for mining regular/frequent itemsets (if an itemset satisfies minimum threshold i.e, min_support , it is called regular itemset. The set of regular k-itemsets is commonly denoted by LK.) brought by R. Agarwal and R. Shrikant in 1994, that leads to generate association rules called association rule mining. It uses bottom–up and iterative approach known as breadth first search (level wise search where k-itemset used to discover k+1 itemset). It generates all regular itemsets(a set of items is referred as an itemset ,n-itemset consist of n items) and apriori property is introduced to reduce the search space .

Apriori property→all non-empty subsets of a regular itemset must also regular. For example if {1,2,3} is a regular itemset, then {(1,2),(1,3),(2,3),(1),(2)and(3)} are must be regular itemset. it uses two key operations first, Join operation→ to discover regular k-itemset, a set of candidate k-itemset is generated by joining L_{K-1} with itself. Second, Prune operation→discard the itemset if support count of itemset is less than minimum support required and also discard the itemset if it is not following apriori property , remove itemset that have subset that is not regular. apriori algorithm generates regular itemset as follows First , regular 1-itemsets are discovered by scanning the transactional database to calculate the count of each item and select the items those satisfying required minimum support , denoted by L_1. Next L_1, regular 1-itemset is used to discover L_2 (set of regular 2-itemset) and L_2 is used to discover L_3 (set of regular 3-itemset) and so on until no more regular itemsets can be discovered and generation of each L_k needs one complete scan of the database . Advantage of this algorithm is , it is simple and easy to mine regular itemsets if database is small. also faces two fatal deficiencies. First, it scans the database multiple times, so greatly high the I/O cost and second, generate large no. of candidate itemsets if database is huge and overall decreases the efficiency of algorithm.

III. Improved Apriori Algorithm

The improved algorithm proposed in this paper works in two phases. In first phase required compressed data structure i.e, b_matrix is constructed and then this compressed data structure is used in second phase to generate regular itemsets. This algorithm employs top-down approach to discover regular item sets from largest regular item set to smallest regular itemset.

Algorithm steps

1. In first phase b_matrix is constructed for the given transactional database. rows in b_matrix represents each transaction and column represents items in transactional database. In b_matrix, each cell will contain values either 1 or 0 for showing the existence of items in transactional database. Entry value will be 1, if the item is present in the respective row else 0, if the item is absent in the row. With two more columns count and redundant transaction counter (TC). Here count column represents the size of row(the sum of total no of 1's in that particular row) and remove those columns whose sum is not equal or greater than predefined min_support value and then update count column. If row is duplicated in database then it is represented by the value in the redundant transaction counter column and delete unnecessary duplicate transaction/row and if row is not duplicate then redundant transaction counter column is set to 1. Then rearrange the b_matrix in descending order based on count column. This is our required compressed data structure and here the phase 1 of our improved algorithm completes.
2. Now generate regular itemsets directly from b_matrix. Select first row from b_matrix and match its count value with next row count respectively. If the next row count is more or equal to the processing row count then do AND operation among the rows, if result is same to the processing row item set structure then increase the count value of support of processing row item set by one and continue this procedure of matching and AND operation through rest of the rows in b_matrix and then check the value of total support. If it is greater or equal to predefined min_support count then extract the item set and its subsets and move them to frequent array list. The same procedure will be repeated for rest of the transactions in b_matrix until all rows are not checked.

The gain of improved algorithm is that it lessen the no. of comparisons to mine largest regular item set for duplicate transactions and transactions having smaller item sets in size that is count value (since they do not have all the items of row under process) and another major advantage is once largest regular item set is discovered then its subsets are searched and moved into frequent array list. While searching for next largest regular item set it checks first, transaction under processing is previously present in frequent array list because of prior largest itemset and its subsets, if itemset is already in frequent array list, it avoids number of comparisons needed to calculate the support count of itemset. Hence decreases number of scans and time needed to extract the regular itemset.

IV. Illustration

Consider the implementation of this improved algorithm through a sample below. TABLE-I shows a transactional database consist of 9 transactions. Set the minimum support counts as 3(min_support=3).

Table 1

TID	ITEMS
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

a) Phase-1

Step1- Scan the transactional database and convert it into desired compressed data structure that is b_matrix M_{9*6}. Where each row represents one transaction and column represents distinct items in whole transactional database and last column i.e, **count** represents the size of row. In b_matrix entry value will be 1, if item is present in the corresponding row else 0,if item is not present in the corresponding row.

Step2- After this rearrange the b_matrix in desending order based on **count** column after removing , those columns whose sum is not equal or greater than required minimum support value. Here min_support is 3, hence remove columns for items 4 and 5 and update **count** column and also merge the duplicate rows in TC column of b_matrix to reduce the computations for redundant rows for finding regular item set.

b) Phase-2

Step3- Now select first row TID-8 and extract its itemset {1, 2, 3} and calculate its support count in b_matrix using AND operation with rows having count value equal or greater than its own count value. If AND operation results in same item set structure as processing row's item set structure, then increase its support count value. after complete AND operation , check value of support count of item set , if it is equal or greater than required min_support than it is frequent/regular , then move itemset with its subset into frequent array list and move to next row. Here item set support is less than required min_support; hence it is not regular move to next row.

Table 2 : (b_matrix)

TID	I1	I2	I3	COUNT	TC
T8	1	1	1	3	2
T1	1	1	0	2	2
T3	0	1	1	2	2
T5	1	0	1	2	2
T2	0	1	0	1	1

Step4- select next row TID-1 and extract its item set {1, 2}. After AND operation its support count is 4, hence it is regular, move item set with its subset into frequent array list. So here regular /frequent array list is {(1, 2), (1) and (2)}. And move to next row.

Step5- select next row TID-3 and extract its item set {2,3}.first check item set in frequent array list , if it is present in frequent array list , then it is regular , no need of AND operation with other rows to calculate its support count. But TID-3 is not present in frequent array list. So do AND operation with rows TID-8,1and 5.after AND operation its support count is 4. Hence it is regular, move {2, 3} {2} and {3} into frequent array list. Now frequent array list is {(1, 2), (2, 3), (1), (2) and (3)}. Move to next row.

Step6- select next row TID-5 and extract its item set {1,3}. Check in frequent array list, not found. Do AND operation with rows TID-8, 1 and 3 and calculate its support count i.e, 4. Hence it is regular, move {1, 3}, {1} and {3} into frequent array list.

Step7- select last row TID-2 and extract its item set {2}. Check in frequent array list, found. Hence it is regular no need of further set of AND operation to calculate its support count. The final frequent array list will be ***Frequent array list*** : {(1,2),(2,3),(1,3),(1),(2)and(3)} extracted in less time by avoiding unnecessary comparisons as per our improved algorithm.

V. Experimental Results

All the experiments are carried out on core i7 Intel based PC machine with 2 GB main memory, running on window 7 operating environment and the program code is written in java. Our experimental benchmark dataset is taken from artificial data set of IBM that is, T1014D100K datasets. there are 100000 data records / affairs and 870 items in T1014D100K dataset. The improved algorithm is compared with classical apriori algorithm using same hardware, dataset and minimum support requirement. The output of both the algorithms is same which demonstrates that our algorithm is competent. the time computation is started from the spot when the file is read into the memory until all the regular itemsets are generated. redundant rate in transactional database is high hence using compressed data structure we eliminate it and reduces space and time complexities in proposed algorithm. from the results shown in figure 1. we can conclude that time and space expenses are lesser as compared to classical algorithm in proposed algorithm.

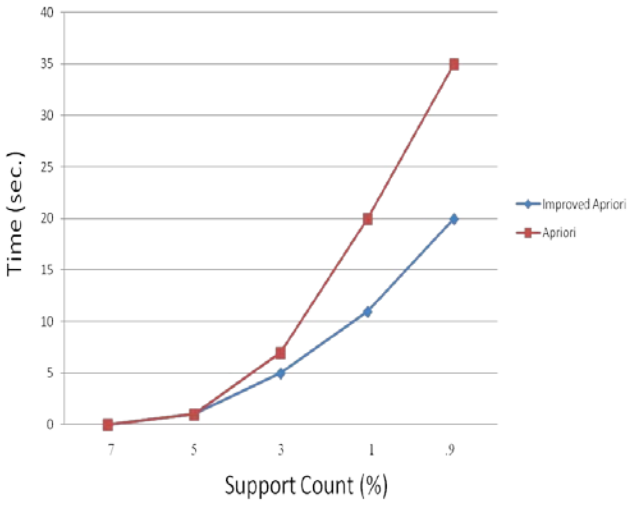

Figure 1 : Execution Time of database

VI. Conclusion

This proposed algorithm for mining regular itemset using bit matrix needs only single scan of whole transactional database to construct compressed data structure . hence greatly reduces I/O cost and it also doesn't generate irregular itemset. So improved algorithm decreases temporal complexity and spatial complexity and have higher efficiency as compared to our classical apriori algorithm.

References Références Referencias

1. Agrawal,R. and Srikant,R. 1994 . Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.
2. J Han, "Data Mining Concepts and Techniques "SecondEdition. Morgan Kaufmann Publisher, 2006,pp.123-134.
3. Hu Ji-ming and Xian Xue-feng. 2006 "The Research and Improvement of Apriori for association rules mining ". Computer Technology and Development 16(4) 99~104.
4. CHEN,Z. CAI,S. SONG,Q and ZHU,C. 2011 "A Improved Aproiri Algorithm based on Pruning Optimization and Transaction Reduction," AMISEC , 2nd IEEE International Conference, pp1908-1911.
5. Agrawal.R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases Clin Proceedings of the 20th International Conference on Very Large Databases, I 994:487-499
6. Rui Chang and Zhiyi Liu , " An Improved Apriori Algorithm," ICEOE 2011, IEEE International Conference, vol. 1, pp v1- 476 - v1- 478.

7. Huiying Wang and Xiangwei Liu, "The Research of Improved Association Rules Mining Apriori Algorithm,"FSKD 2011, 8Th IEEE International Conference, vol 2, pp961-964.
8. Qiang Ma. Improved Algorithm based on Apriori Algorithm[J]. Development and application of computer,2010,23(2):6-7.
9. Yubo Jia, Guanghu Xia, Hongdan Fan, Qian Zhang and Xu Li, "An Improved Apriori Algorithm Based on Association Analysis," ICNDC 2012, 3rd IEEE International Conference, pp208-211.
10. Jingyao Hu, "The Analysis on Apriori Algorithm Based on Interest Measure," ICCECT 2012, IEEE International Conference, pp1010-1012