IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. I (Jan – Feb. 2015), PP 48-52
www.iosrjournals.org

A classification of methods for frequent pattern mining

Patel Atul R.[1], Patel Tushar S.[2]
[1]*ME (CSE Student), S. P. B. Patel Engineering College, Mehsana, Gujarat, India*
[2]*IT Department, S. P. B. Patel Engineering College, Mehsana, Gujarat, India*

Abstract: *Data mining refers to extracting knowledge from large amounts of data. Frequent pattern mining is a heavily researched area in the field of data mining with wide range of applications. Frequent itemsets is one of the emerging task in data mining. A many algorithms has been proposed to determine frequent patterns. Apriori algorithm is the first algorithm proposed in this field. An Apriori algorithm having two major limitation first generate huge candidate itemsets and second more times scan the database. Problem, to be solved some methods for frequent itemset mining in the paper. Three major factors used in frequent itemset mining such as time, scalability, efficiency. In this paper we have analyze various algorithm for frequent itemset mining such as CBT-fi, Index-BitTableFI, Hierarchical Partitioning, Matrix based Data Structure, Bitwise AND, Two-Fold Cross-Validation and binary based Semi-Apriori Algorithm also discuss advantages & disadvantages of the frequent itemset mining algorithm.*

Keywords: *Frequent, K-itemset, Mine*

I. Introduction

Data mining is the process of discovering interesting knowledge from large amounts of data stored in database, data warehouse, or other information repositories. Popular research area of data mining was started by the tasks of frequent item set mining and association rule induction. The huge research efforts devoted to these tasks have led to a variety of sophisticated and efficient algorithms to find frequent item sets. Among the best-known approaches are Apriori, Eclat and FP-growth [7]. Frequent pattern mining is the process of searching recurring relationships in a given dataset. Frequent patterns are patterns (i.e. itemsets) that appears in a dataset frequently. A set of items, i.e. computer and antivirus that appears frequently together in a transaction dataset is a frequent itemsets. Frequent patterns mining like Frequent itemsets find frequent itemsets from the small database and/or large database, where the database are either transactional or relational. The frequent itemset mining is the process of finding out frequent itemsets from the DB. Frequent itemsets such as 1-frequent, 2-frequent, 3-frequent. . . . . k-frequent itemsets.

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases such as association rules, correlations, sequences, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. Frequent item set mining is a data analysis method, which was originally developed for market basket analysis and which aims at finding regularities in the shopping behaviour of the customers of supermarkets, mail-order companies and online shops.[7]**.**

Frequent pattern mining was first proposed by Agrawal et al. (1993) for market basket analysis in the form of association rule mining. It analyses customer buying habits by finding associations between the different items that customers place in their "shopping baskets". For instance, if customers are buying computer, how likely are they going to also buy antivirus (and what kind of antivirus) on the same trip to the supermarket? Such information can lead to increased sales by helping retailers do selective marketing and arrange their shelf space. Association rules describe how often items are purchased together.

The frequent itemset mining algorithms such as, Apriori, AprioriTID, DHP, FDM, DIC, PincerSearch, CARMA, Eclat, Diffset, FP-growth, DSM-FI, PRICES, FHARM, H-mine, DTFIM, GIT-tree, Scaling Apriori, FPG ARM etc.

In this paper, section 2 discuss the problem definition of the frequent itemset mining; section 3 discuss the various algorithm for frequent itemset mining; section 4 discuss analysis and discussion; Finally section 5 concludes the paper.

II. Problem Defination

In Apriori algorithm, Generate the huge number of candidate itemsets and more time scan to database. Also major challenge found in frequent pattern mining is a large number of result patterns. This problem further to be solved that various method and algorithm for frequent itemset mining. The main aim of the frequent itemset mining to optimize the process of finding patterns which should be efficient, scalable and detect the important patterns.

III. Classification

Following various algorithm for frequent itemset mining.

(1) CBT-fi Algorithm

A Compact Bittable frequent itemset was proposed to reduce the number of transactions from the original database (called Transaction Reduction) to find the frequent itemsets. This algorithm reduces the number of scans in the database and improves the efficiency and reduces the execution time. Also reduce the number of scan from the DB. The process begins with, the transaction database can be transformed into a binary matrix M, in which each row corresponds to a transaction and each column corresponds to an item. Therefore the bit-table contains 1 if the item is present in the current transaction and 0 otherwise. After in M compute the column wise bit count for each item and eliminate the items column whose bit count is less than min_sup value. And find out the frequent 1-itemset. Next step is to sort the frequent single items in ascending order based on the support count. Create a Compact Bit Table steps is to the cluster(group) the similar transaction (row) based on the decimal value of each row is denoted as record-count vector(rcv) and also compute the bit count for each transaction(row) is denoted as bitcount vector(bcv) as CB Table. Finally find out the frequent 2-itemset,frequent 3-itemset and so on from CB Table with rcv and bcv [5]. The detail parameters are shown in Table 1.

Table 1. CBT-fi algorithm parameters

Storage Structure	Array based
Technique	CBT saves the memory considerably by clustering the similar transactions.
Memory Utilization	Require less amount of memory
Databases	Suitable for sparse dataset, medium dataset, dense dataset
Time	Execution time is less where minimum support is low

(2) Index-BitTableFI Algorithm

This algorithm based on BitTableFI. Methods for mining frequent itemsets have been implemented using a Bit Table structure. BitTableFI is such a recently proposed efficient BitTable-based algorithm, which exploits BitTable both horizontally and vertically. Although making use of efficient bit wise operations, BitTableFI still may suffer from the high cost of candidate generation and test. To address this problem, a new algorithm Index-BitTableFI is proposed. Index-BitTableFI also uses BitTable horizontally and vertically. To make use of BitTable horizontally, index array and the corresponding computing method are proposed. By computing the subsume index, those itemsets that co-occurrence with representative item can be identified quickly by using breadth-first search at one time. Then, for the resulting itemsets generated through the index array, depth-first search strategy is used to generate all other frequent itemsets. Thus, the hybrid search is implemented, and the search space is reduced greatly. The advantages of the proposed methods are as follows. On the one hand, the redundant operations on intersection of tidsets and frequency-checking can be avoided greatly; On the other hand, it is proved that frequent itemsets, including representative item and having the same supports as representative item, can be identified directly by connecting the representative item with all the combinations of items in its subsume index [1]. The detail parameters are shown in Table 2.

Table 2. Index-BitTableFi algorithm parameters

Storage Structure	Array based
Technique	Index-BitTableFI generates frequent itemsets from BitTable, index array and subsum index.
Memory Utilization	Require less amount of memory
Databases	Efficient for dense dataset
Time	Execution time is less

(3) Hierchical Partationing Algorithm

A Hierarchical Partitioning Algorithm is proposed for the problem of mining frequent itemsets when the size of the database is very large. Hierarchical partitioning for mining frequent itemsets in large databases based on a novel data structure called the Frequent Pattern List (FPL). One of the major features of the FPL is its ability to partition the database, and thus transform the database into a set of sub-databases of manageable sizes. As a result, a divide-and-conquer approach can be developed to perform the desired data-mining tasks. Experimental results show that hierarchical partitioning is capable of mining frequent itemsets and frequent closed itemsets in very large databases [4]. The detail parameters are shown in Table 3.

Table 3. Hierarchical Partitioning algorithm parameters

Storage Structure	Array based
Technique	Hierarchical Partitioning Algorithm based on FPL data structure.
Memory Utilization	Require different amount of memory at different point (size) of time
Databases	Suitable for sparse dataset, medium dataset
Time	Execution time is less

(4) Matrix based Data Structure Algorithm

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 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 [6].The detail parameters are shown in Table 4.

Table 4. Matrix based Data Structure algorithm parameters

Storage Structure	Array based
Technique	Find the frequent itemsets from b_matrix and only one time scan the whole database
Memory Utilization	Require less amount of memory
Databases	Suitable for sparse dataset, medium dataset and dense dataset
Time	Execution time is less compare to Apriori

(5) Bitwise AND Algorithm

According to Bitwise AND Algorithm, the data flow is based on its characteristics, the approach puts forward a new frequent itemsets mining algorithm based on bitwise and computation. Algorithm uses basic window for unit, and update sliding window in memory using arrays structure maintenance item of frequent information, finally by the frequent items between the bitwise and operations to get all the frequent itemsets. Algorithm in each basic window goes into the sliding window after dynamically update arrays, analysis and experiment shows that the algorithm has better performance [2]. The detail parameters are shown in Table 5.

Table 5. Bitwise AND algorithm parameters

Storage Structure	Array based
Technique	Find the frequent itemsets from basic window
Memory Utilization	Require less amount of memory
Databases	Suitable for sparse dataset, medium dataset and dense dataset
Time	Required less amount of time

(6) Two-Fold Cross-Validation Algorithm

Apriori TS-Tree Two-Fold Cross-Validation algorithm based on Apriori Two-Fold Cross-Validation algorithm. This technique use a TS-tree (total support tree) to generate the few candidate items. Main disadvantage of this technique to must partition of the database and used TS-tree. Apriori TS-tree Cross Validation Optimized uses TS-tree for efficient implementation of validation steps in based algorithm. In this technique first partition of whole database T_1 and T_2. After TS-tree is created for T_1 and T_2. The tree contains set F^{μ}_{T1} & F^{μ}_{T2} all μ-frequent itemsets in partition T_1 & T_2 respectively. Finally find the frequent itemset for union of the F^{μ}_{T1} and F^{μ}_{T2}. In experiment execution time is less compare to Apriori [3]. The detail parameters are shown in Table 6.

Table 6. Two-Fold Cross-Validation algorithm parameters

Storage Structure	Array based
Technique	Find the frequent itemsets from TS-tree
Memory Utilization	Require much amount of memory
Databases	Suitable for sparse dataset, medium dataset, dense dataset
Time	Execution time is less compare to Apriori in experiment result

(7) Semi-Apriori Algorithm

A binary based Semi –Apriori technique that efficiently discovers the frequent itemsets. Frequent itemsets generation produce extremely large numbers of generated itemsets that make the algorithms inefficient. The reason is that the most traditional approaches adopt an iterative strategy to discover the itemsets, that's require very large process. Furthermore, the present mining algorithms cannot perform efficiently due to high and repeatedly database scan. A new binary-based Semi-Apriori technique that efficiently discovers the frequent itemsets. Extensive experiments had been carried out using the new technique, compared to the existing Apriori algorithms, a tentative result reveal that our technique outperforms Apriori algorithm in terms of execution time [7]. The detail parameters are shown in Table 7.

Table 7. Semi-Apriori algorithm parameters

Storage Structure	Array based
Technique	Find the frequent itemsets from three stages and reduce the candidate itemsets

Memory Utilization	Require less amount of memory
Databases	Suitable for sparse dataset, medium dataset, dense dataset
Time	Execution time is less compare to Apriori where minimum support is low the execution is fast in experiment result

IV. Analysis And Discussion

We have study about various algorithm such as, CBT-fi, Index-BitTableFI, Hierchical Partationing, Matrix based Data Structure, Bitwise AND, Two-Fold Cross-Validation and binary based Semi-Apriori Algorithm for frequent itemset mining. The advantages and disadvantages of various methods are shown in Table 7.

Table 7. Advantages and disadvantages

Methods	Advantages	Disadvantages
CBT-fi	(1) Reduce the transaction using rcv and bcv, row/bit count vector (2) Use less amount of memory or which clusters(groups) the similar transaction into one and forms a compact bit-table structure which reduces the memory consumption (3) Scan database only once (4) Find out 2-itemset,3-itemset, up to k-itemset based on CB Table with rcv and bcv	(1) must count the rcv and bsv
Index Bit-Table	(1) Execution time is less where similar transaction avoid greatly (2) Search space is reduced greatly (3) Processing : cost is low and efficiency is high	(1) only efficient for dense dataset
Hierarchical Partitioning	(1) There is no extra cost for re-scanning the original database (2) HP can be used with memory based algorithm for large database	(1) Because of database is always being updated and content always changing & size always increasing
Matrix Data Structure	(1) Only one time scan the whole database (2) Reduce I/O cost (3) Higher efficiency compare to classical Apriori algorithm (4) Reduce the execution time (5) Execution time is less compare to Apriori	(1) Must use a bit matrix
Bitwise AND Algorithm	(1) Better performance dynamic update array, analysis of database (2) Require less amount of memory (3) More accuracy and efficiency	(1) Must use a sliding window that increase cost
Two Fold Cross Validation Model	(1) Few candidate items generated (2) Execution time is less compare to Apriori	(1) Must partition of the database & use a TS-Tree
Semi-Apriori Algorithm	(1) Reduce the candidate itemsets (2) Rrduce the total number of database passes (3) Execution time is less compare to Apriori	(1) Minimum support is high require more execution time

V. Conclusion

The analyze of some methods for frequent itemsets mining. CB-Table algorithm better then the BitTableFI and Index-BitTableFI algorithm. A Compact BitTable approach for mining frequent itemsets (CBT-fi) which clusters(groups) the similar transaction into one and forms a compact bit-table structure which reduces the memory consumption as well as frequency of checking the itemsets in the redundant transaction.when Index-BitTableFI algorithm does not reduced the transaction and efficient for only dense dataset. The problem of mining frequent itemsets when the size of the database is very large which used a Hierarchical Partitioning approach.

The classical Apriori 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 use to a matrix base data structure. Bitwise AND algorithm provide the more accuracy and more efficiency. Two-Fold Cross-Validation algorithm better than Apriori algorithm because Apriori generate huge candidate itemset. When Two-Fold Cross-Validation algorithm generate few candidate itemset and used to a TS-Tree.

A Semi-Apriori algorithm better then Apriori algorithm. This algorithm reduce the candidate itemset, avoid the repeated database scan and better performance compare to Apriori.

References

[1]. Wei Song, Bingru Yang and Zhangyan Xu, " Index-BitTableFI: An improved algorithm for mining frequent itemsets," Proceedings of the Elsevier, March 2008.

[2]. Guoxiaoli, Fengli and Guoping, " Research on Mining Frequent Itemsets based on Bitwise AND Algorithm," Proceedings of the IEEE, 2011.

[3]. Predrag Stanisic and Savo Tomovic, " Frequent Itemset Mining Using Two-Fold Cross-Validation Model," Proceedings of the IEEE, 2012.

[4]. Fan-Chen Tseng, " Mining frequent itemsets in large databases: The hierarchical partitioning approach," Proceedings of the Elsevier, 2013.

[5]. A.Saleem Raja and E.George Dharma Prakash Raj, " CBT-fi: Compact BitTable Approach for Mining Frequent Itemsets," Proceedings of the Advances in Computer Science: an International Journal, sep 2014.

[6]. Shalini Dutt, Naveen Choudhary and Dharm Singh., " An Improved Apriori Algorithm based on Matrix Data Structure," Proceedings of the Global Journal of Computer Science and Technology: C Software & Data Engineering, 2014.

[7]. Sallam Osman Fageeri, Rohiza Ahmad and Baharum B. Baharudin, " A Semi-Apriori Algorithm for Discovering the Frequent Itemsets," Proceedings of the IEEE, 2014.

[8]. Christian Borgelt, "Simple Algorithms for Frequent Item Set Mining."European Center for Soft Computing,Asturias, Spain.

[9]. Jiawei Han and Kamber, Data Mining: concept and Techniqes, second edition, Elsevier, 2006.