A Collaborative Approach of Frequent Item Set Mining: A Survey

Arpan Shah, Pratik Patel

Parul Institute of technology, VADODARA, GUJARAT,INDIA

Summary

Data mining defines hidden pattern in data sets and association between the patterns. In data mining, association rule mining is key techniques for discovering useful patterns from large collection of data. Frequent iemset mining is a step of association rule mining. Frequent itemset mining is used to gather itemsets after discovering association rules. In this paper, we have explained fundamentals of frequent itemset mining. We have defined present's techniques for frequent item set mining. From the large variety of capable algorithms that have been established we will compare the most important ones. We will organize the algorithms and investigate their run time performance.

Key words:

Data mining, Association Rules, Frequent Item set Mining, FP-growth, Minimum Support

1. Introduction

Frequent item set mining is one of the most significant and general topic of research for association rule mining in data mining research area. As performance of association rule mining depends upon the frequent itemsets mining, thus is necessary to mine frequent item set efficiently. A frequent itemset is an itemset that occurs frequently. In frequent pattern mining to check whether an itemset occurs frequently or not we have a parameter called support of an itemset [1]. An itemset is termed frequent if its support count is greater than the minimum support count set up initially.Association rule is faced by X→Y where X and Y are item sets and their intersection is null i.e. $X \cap Y= \{\}$.The support of an association rule is the support of the union of X and Y, i.e. XUY. X is called the head or antecedent and Y is called the tail or consequent of the rule .The confidence of an association rule is defined as the percentage of rows in D containing itemset X that also contain itemset Y,

i.e, CONFIDENCE (X → Y) =P (X|Y) = SUPPORT (XY)/SUPPORT (X).

A large number of algorithms were proposed by many researchers for generation of frequent itemsets, firstly Apriori, like algorithms are proposed but due to their large number of candidate generation, more database scan, slow processing and for some cases when support threshold is low then frequent patterns generation become doubtful because of high memory dependency, huge search space and large I/O required in these type of algorithms. In this paper, new algorithms have been studied like FP-Growth and their variations to reduce the memory requirements, to decrease I/O dependencies, and also to reduce the pruning strategies for efficiently generation of frequent itemsets [1].

2. Frequent Item Set Mining Based Algorithm

2.1 Horizontal layout based data mining Algorithms

In this each row of record characterizes a transaction identifier (TID), tracked by a set of items.

2.1.1. Apriori Algorithm

Apriori algorithm is, the most typical and significant algorithm for mining frequent itemsets. Apriori is used to discover all frequent itemsets in a given database. The apriori algorithm uses the apriori principle, which says that the item set I containing item set X is never large if item set X is not large or all the non-empty subset of frequent item set must be frequent also. Based on this principle, the apriori algorithm generates a set of candidate item sets whose lengths are (k+1) from the large k item sets and prune those candidates, which does not contain large subset. Then, for the rest candidates, only those candidates that satisfy the minimum support threshold are taken to be large (k+1)-item sets. Without scanning the transactions, the apriori generate item sets by using only the large item sets found in the previous pass. Steps involved are:

1. Generate the candidate 1-itemsets (C1) and write their support counts during the first scan.

2. Find the large 1-itemsets (L1) from C1 by eliminating all those candidates which does not satisfy the support criteria.

Manuscript received November 5, 2015
Manuscript revised November 20, 2015

3. Join the L1 to form C2 and use apriori principle and repeat until no frequent itemset is found.

2.1.2. Direct Hashing and Pruning (DHP) Algorithm:

DHP can be derived from apriori by introducing additional control. To this purposes DHP makes use of an additional hash table that aims at limiting the generation of candidates in set as much as possible. DHP also progressively trims the database by discarding attributes in transaction or even by discarding entire transactions when they appear to be subsequently useless. In this method, support is counted by representing the items from the element list into the container which is distributed according to support known as Hash table. As the new item set is met if item exist before then increase the container count else insert into new container. Thus in the end the container whose sup count is a lesser amount of the minimum support is removed from the candidate set.

2.1.3. Partitioning Algorithm:

Partitioning algorithm is to find the frequent elements on the source of dividing database into n partitions. It overcomes problem of memory for large database which do not appropriate into main memory because minor parts of database without problems fit into main memory. The algorithm executes in two phases. In the first phase, the Partition algorithm logically divides the database into a number of non-overlapping partitions. The partitions are considered one at a time and all large itemsets for that partition are generated. At the end of phase I, these large itemsets are merged to generate a set of all potential large itemsets. In phase II, the actual support for these itemsets is generated and the large itemsets are identified. The partition sizes are chosen such that each partition can be accommodated in the main memory so that the partitions are read only once in each phase.

2.1.4. Sampling Algorithm:

Random sampling is a method of selecting n units out of a total N, such that every one of the Cn N distinct samples has an equal chance of being selected. It is just created in the idea to choice an arbitrary sample of the itemset R from the database in its place of whole database D [4]. The sample is designed like a whole sample is held in the main memory. Thus, we try to find the frequent candidates for the sample only and there is chance to find error of global frequent elements in that sample hence lower threshold support is used rather than actual minimum support to find the frequent candidates in local to sample [4].

2.1.5. Dynamic Item set Counting (DIC):

This is an alternative to Apriori Itemset Generation. In this itemsets are dynamically added and deleted as transactions are read. It is based on the downward release property in which this calculates the itemsets to different point of time regarding the scan. This algorithm also used to ease the number of database for discovering the frequent itemsets by just counting the new element at any fact of time finished the run time [4, 5].

2.1.6 Continuous Association Rule Mining Algorithm (CARMA):

CARMA takes the computation of large itemsets online. Being online, CARMA indications the present association rules to the user and permits the user for converting the parameters, minimum support and minimum confidence, at any matter through the first scan of the database. It wants at most 2 database scans. CARMA creates the itemsets in the first scan and appearances counting all the itemsets in the second scan [4]. It also increments the counts of given itemsets, that are subset of the transaction after reading all transaction. Then it produces new itemsets from the transaction, if all direct subsets of the itemsets are currently potentially large respecting the current minimum support and the portion of the data that is read. For more perfect assessment of whether an itemset is theoretically large, it computes an upper bond for the count of the itemset, which is the total amount of its current count and evaluate of the number of amounts before the itemset is produced. The asset of the amount of occurrences is computed when the itemset is of first generated [4].

2.2 Vertical Layout Data Mining Algorithm:

For vertical layout data set, each column communicates to an item, swapped by a TID list, which is the list of rows that the item occurs.

2.2.1. Eclat Algorithm:

Equivalence Class Clustering and bottom up Lattice Traversal is known as ECLAT algorithm. This algorithm is also used to perform item set mining. It uses TID set intersection that is transaction id intersection to compute the support of a candidate item set for avoiding the generation of subsets that does not exist in the prefix tree. For each item store a list of transaction id. An algorithm, for all frequent itemset denoted by ‘i’ new database is created Di. It is appropriate for small datasets and needs less time for frequent pattern generation than apriori. Compared with other algorithms like Apriori, FP- growth etc, it does not create the candidate item sets. This algorithm scans the database only once and creates the vertical data base, which identifies each item in the list of transactions that supports the items [15].

2.3. Projected Layout Data mining Algorithm:

This kind of dataset uses divide and conquer strategy to extract itemsets therefore it counts the support more powerfully then Apriori based algorithms. Tree projected layout techniques use tree structure to conserve and extract the itemsets [4].

2.3.1. FP-Growth Algorithm:

Fp–growth is also called as projected database based data mining techniques, which generates frequent itemset without candidate generation. It uses tree based structure. The problem of Apriori algorithm was distributed with, by introducing a different data structure, called frequent pattern tree, or FP-tree then based on this structure an FP-tree created pattern fragment growth method was established. It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum support. FP-growth traces the set of concurrent items [3, 4, 5, 6, and 7].
Advantages:-
1. Only 2 passes over data-set
2. “Compresses” data-set
3. No candidate generation
4. Much faster than Apriori
Disadvantages:-
1. FP-Tree may not fit in memory!!
2. FP-Tree is expensive to build

2.3.2 H mine Algorithm

A memory-based, efficient pattern-growth algorithm, H-mine is for mining frequent patterns for the datasets that can fit in memory [4]. A simple, memory-based hyper-structure, H-struct, is planned for fast mining. H-mine has polynomial space complexity and is thus more space efficient than pattern-growth methods like FP-growth and tree projection when mining sparse datasets, and also more effective than apriori-based systems which produce a huge amount of number of elements. H-mine has very limited and exactly expectable space overhead and is faster than memory-based apriori and FP-growth. H-mine needs an H-struct new data structure for mining purpose known as hyperlinked structure. It is used upon the dynamic adjustment pointers which help to retain the procedure expected tree in memory so, H- mine designed in frequent pattern to mine data form dataset that can fit into main memory [4, 17].

3. Conclusion

The overall purpose of the data mining process is to extract knowledgeable information from a large data set and transform it into a logical structure for advance use. Association rules prove to be the most effective technique for frequent pattern matching over a decade. This paper gives a brief survey on different approaches for mining frequent itemsets using association rules. The performance of algorithms reviewed in this paper based on support count, size of datasets, nature.

References

[1] S. Neelima, N. Satyanarayana and P. Krishna Murthy3,”A Survey on Approaches for Mining Frequent Itemsets”, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-87

[2] Jeetesh Kumar Jain, Nirupama Tiwari, Manoj Ramaiya, “A Survey: on Association Rule Mining”, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 1, January–February2013,pp 2065-2069

[3] Aakansha Saxena, Sohil Gadhiya , “A Survey on Frequent Pattern Mining Methods Apriori, Eclat, FP growth”, 2014 IJEDR | Volume 2, Issue 1 | ISSN: 2321-9932

[4] Bharat Gupta, Dr. Deepak Garg, “FP-Tree Based Algorithms Analysis: FP Growth, COFI-Tree and CT-PRO”, International Journal on Computer Science and Engineering (IJCSE) 2013

[5] Wei Zhang, Hongzhi Liao.Na Zhao, “Research on the FP Growth Algorithm about Association Rule Mining”, 2008 International Seminar on Business and Information Management

[6] SakthiNathiarasan1, Kalaiyarasi2, Manikandan3, Literature Review on Infrequent Itemset Mining Algorithm, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 8, August 2014

[7] Varsha Mashoria, Anju Singh, “Literature Survey on Various Frequent Pattern Mining Algorithm”, IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719 Vol. 3, Issue 1 (Jan. 2013), ||V1|| PP 58-64

[8] Vikas Kumar, Sangita Satapathy, “A Review on Algorithms for Mining Frequent Itemset Over Data Stream”, Volume 3, Issue 4, April 2013 ISSN: 2277 128X

[9] Pradeep Rupayla, Kamlesh Patidar, “A Comprehensive Survey of Frequent Item Set mining Methods”, IJETAE, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

[10] Neelesh Kumar Kori, Ramratan Ahirwal, Dr. Yogendra Kumar Jain, “Efficient Frequent Itemset Mining Mechanism Using Support Count”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 1, Issue 1, September 2012

[11] Shipra Khare , Prof. Vivek Jain, “A Review on Infrequent Weighted Itemset Mining using Frequent Pattern Growth”,

(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 1642-1647

[12] Le Wang, Lin Feng Jing Zhang, Pengyu Liao, “An Efficient Algorithm of Frequent Itemsets Mining Based on MapReduce”, Journal of Information & Computational Science 11:8 (2014) 2809–2816 May 20, 2014

[13] S.Vijay Jeetesh Kumar Jain, Nirupama Tiwari, Manoj Ramaiya, “A Survey: on Association Rule Mining”, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 1, January –February 2013, pp.2065-2069

[14] S.Vijayaranii el.al. “Mining Frequent Item Sets over Data Streams using Éclat Algorithm” International Conference on Research Trends in Computer Technologies (ICRTCT-2013).

[15] Jian Pei , Jiawei Han , Hongjun Lu , Shojiro Nishio , Shiwei Tang , Dongqing Yang “H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases”.

[16] H. Altay Güvenir, “An Algorithm for Mining Association Rules Using Perfect Hashing and Database Pruning’s

[17] T. Karthikeyan1 and N. Ravikumar, “A Survey on Association Rule Mining”International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2014.