ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 8, August 2014

Literature Review on Infrequent Itemset Mining Algorithms

Sakthi Nathiarasan[1], Kalaiyarasi[2], Manikandan[3]

M.E-Student, Department of CSE, Adhiyamaan College of Engineering, Hosur, India[1,2]

Assistant Professor, Department of CSE, Adhiyamaan College of Engineering, Hosur, India[3]

Abstract: The aim of Association Rule Mining is to find the correlation between data Items based on frequency of occurrence. Infrequent Itemset mining is a variation of frequent itemset mining where it finds the uninteresting patterns i.e., it finds the data items which occurs very rarely. Considering weight for each distinct items in a transaction independent manner adds effectiveness for finding frequent itemset mining. Several articles related to frequent and weighted infrequent itemset mining were proposed. This paper focus on reviewing various Existing Algorithms related to frequent and infrequent itemset mining which creates a path for future researches in the field of Association Rule Mining.

Keywords: Infrequent Itemset mining, Association Rule Mining, weight, Correlation

I. INTRODUCTION

Association Rule mining(ARM)[1] is one of the popular technique used to find the correlation between the data items in the database based on some statistical measures but not considering the interesting of the business users. ARM is one of the oldest technique in data mining. The goal of ARM is to find the relationship, correlation among different data sets in the database. Itemset mining is an exploratory data mining technique widely used for discovering valuable correlations among data. Frequent itemsets mining is a core component of data mining and variations of association analysis, like association rule mining. Infrequent itemsets are produced from very big or huge data sets by applying some rules or association rule mining algorithms like Apriori technique, that take larger computing time to compute all the frequent itemsets. Extraction of frequent itemsets is a core step in many association analysis techniques. The frequent occurrence of item is expressed in terms of the support count. However, significantly less attention has been paid to mining of infrequent itemsets, but it has acquired significant usage in mining of negative association rules from infrequent itemset, fraud detection where rare patterns in financial or tax data may suggest unusual activity associated with fraudulent behavior, market basket analysis and in bioinformatics where rare patterns in microarray data may suggest genetic disorders. Several frequent item set mining including Apriori, FP-Growth algorithm, AFOPT algorithm, NONORDFP algorithm, FP-GROWTH* algorithm, Broglet's FP-Growth, DynFP-Growth algorithm, Enhanced FP-Growth algorithm, IFP-min Algorithm and Transaction mapping algorithm were proposed. And this paper discuss about literature review on various frequent and infrequent itemset mining algorithms.

II. LITERATURE REVIEW

A. The Apriori Algorithm

Apriori[2] was the first proposed algorithm in association rule mining, to identify the frequent itemsets in the large transactional database. Apriori works in two phases. During the first phase it generates all possible Itemsets combinations. These combinations will act as possible candidates. The candidates will be used in subsequent phases. In Apriori algorithm, first the minimum support is applied to find all frequent itemsets in a database and Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

Apriori Algorithm:

```
procedureApriori (T, minSupport)
{
 //T is the database and minSupport is the minimum
support
L1= {frequent items};
for (k= 2; Lk-1 !=∅; k++)
{
Ck= candidates generated from Lk-1
for each transaction t in database
do
{
 #increment the count of all candidates in Ck that are
contained in t
 Lk = candidates in Ck with minSupport
 }//end for each
 }//end for
returnUk Lk ;
}
```

The main drawback of Apriori is the generation of large number of candidate sets. The efficiency of apriori can be improved by Monotonicity property, hash based technique, Partioning methods and so on.

B. FP-growth Algorithm

The drawback of Apriori can be improved by Frequent pattern Growth algorithm[3].This algorithm is implemented without generating the candidate sets. This algorithm proposes a tree structure called FP tree structure,

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 8, August 2014

going to collect information from the database and creates an optimized data structure as Conditional pattern. Initially it Scans the transaction database DB once and Collects the set of frequent items F and their supports and then Sort the frequent itemsets in descending order as L, based on the support count. This algorithm reduces the number of candidate set generation, number of transactions, number of comparisons.

Algorithm FP-growth:

Input:

-A transactional database *DB* and a minimum support threshold ξ.

Output:

- frequent pattern tree, FP-tree

/*phase1: */

(1) Scan the transactional database.
(2) Collect the set of frequent items *F* and their supports. Sort *F* in support descending order as *L*. /* the *list* of frequent items*/
(3) [3] Create the root of an FP-tree, *T*, and label it as "root" /* for each transaction do */
(4) Select and sort the frequent items in *Trans* according to the order of *L*.
(5) perform the insert-tree function /* call insert-tree function recursively */ /*phase2: */

Input:

An FP-treeconstructed in the above algorithm,
D – transaction database;
s – minimum support threshold.

Output:

-The complete set of frequent patterns.

(1) call the FP-Growth function
(2) check if the tree has a single path,
(3) then for each combination (denoted as *B*) of the nodes in the path *P* do
(4) generate pattern B ∪ A with support=minimum support of nodes in B
(5) else
(6) construct B's conditional pattern base and B's conditional FP-tree

C. AFOPT Algorithm

Liu et al[4] investigated the algorithmic performance space of the FP-growth algorithm. AFOPT algorithm uses dynamic ascending frequency order for both the search space exploration and prefix-tree construction, it uses the top-down traversal strategy. AFOPT algorithm utilizes dynamic ascending frequency for the item search space ,adaptive representation for the conditional database format, physical construction for the conditional database construction, and top-down traversal strategy for the tree traversal. The dynamic ascending frequency search order can make the subsequent conditional databases shrink rapidly. As a result, it is useful to use the physical construction strategy with the dynamic ascending frequency order.

D. NONORDFP Algorithm

The computational time and space complexity of the FP-Growth algorithm was improved by NONORDFP algorithm. Racz[5] proposed thecompact representation of FP-tree allows faster allocation, traversal, and optionally projection. It contains less administrative information about the items in the database and allows more recursive steps to be carried out on the same data structure, with no need to rebuild it.

E. FP-Growth Algorithm*

Grahne et al [6], found that 80% of CPU was used for traversing the FP-trees. Fpgrowth* algorithm uses FP-tree data structure in combination with the array-based and incorporates various optimization techniques. Array-based technique is used to reduce the traversal time of FP-tree. It reduces the memory consumption compared to FP-growth Algorithm.

F. Broglet's FP-Growth

FP-Growth algorithm can be improved by Broglet's FP growth algorithm[6] . Initially it scans the frequencies of the items and all infrequent items, that is, all items that appear in fewer transactions than a user-specified minimum number are discarded from the transactions, since, they can never be part of a frequent item set. The items in each transaction are sorted, so that they are in descending order with respect to their frequency in the database. It reduces the computational cost in FP-Growth.

G. DynFP-growth Algorithm

The main drawback of the Aprioi-like methods is at the candidate set generation and test. This problem was taken into consideration by introducing a novel, compact data structure, named frequent pattern tree[3], or FP-tree, is not unique for the same logical database. This approach can provide a very quick response to any queries even on databases that are being continuously updated. Because the dynamic reordering process, Gyorodi C et al [7] proposed a modification of the original structures, by replacing the single linked list with a doubly linked list for linking the tree nodes to the header and adding a master-table to the same header.

H. Enhanced FP-Growth Algorithm

Enhanced-FP[8], which does its work without any prefix tree and any other complex data structure. It processes the transactions directly, so its main strength is its simplicity. It initially scans the supports of the items and is calculated. The items whose support count is less than minimum support are discarded and specified as infrequent items. Then the items in the database are sorted in ascending order with respect to their support. And the initial transaction database is converted in to a set of transaction list, with one list for each item. These lists are stored in array, each of which contains a pointer to the head of the list. And the Transaction lists are traversed from left to right for finding all the frequent item set that contain the item the list corresponds to. Before a transaction list is processed, its support count is checked, if it exceeds than minimum support count than there must be a frequent item set.

I. IFP-min Algorithm

IFP min algorithm[9] that uses a recursive approach to mine minimally infrequent Item sets(MIIs). The infrequent

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 8, August 2014

item sets are then reported and it gets pruned from the database. The items presented in the modified database are individually frequent. This algorithm then selects the MIIs and it divides into two non-disjoint sets as residual database and projected database. First the IFP-min algorithm is applied to residual database, where the MIIs are reported, if the database has the single item then it is considered to be the item itself or as an empty set. Then IFP-min algorithm is applied to projected database. The itemsets in the projected database share the lf-item as a prefix. The MIIs obtained from the projected database by recursively applying the algorithm are compared with those obtained from residual database. If an itemset is found to occur in the second set, it is not reported; otherwise, the lf-item is included in the itemset and is reported as an MII of the original database. The use of residual tree is to reduce the computational time.

J. Transaction mapping Algorithm

The transaction tree is similar to FP-tree but there is no header table or node link. The transaction tree[10] has compact representation of all the transactions in the database. Each node in tree has an id corresponding to an item and a counter that keeps the number of transactions that contain this item in this path. Here we can compress transaction for each itemset to continuous intervals by mapping transaction ids into a different space to a transaction tree. Advantage of this algorithm is the performance can be improved compared to FP-Growth, FP-Growth* algorithms.

Algorithm:

Input:

-Databse DB

Output:

-all infrequent item sets

(1) scan the database and identify the infrequent item sets.
(2) construct the transaction tree with the count for each node.
(3) Construct the transaction interval lists.
(4) Construct the lexicographic tree in a depth first order keeping only the minimum amount of information necessary to complete the search.

K. The Infrequent Weighted Itemset Miner Algorithm

IWI Miner is a FP-growth-like mining algorithm[11] that performs projection-based itemset mining. Hence, it performs the main FP-growth mining steps:

(a) FP-tree creation and

(b) recursive itemset mining from the FPtree index. Unlike FP-Growth, IWI Miner discovers infrequent weighted itemsets instead of frequent (unweighted) ones. To accomplish this task, the following main modifications with respect to FP-growth have been introduced:

(i) A novel pruning strategy for pruning part of the search space early and
(ii) a slightly modified FP-tree structure, which allows storing the IWI-support value associated with each node.

Algorithm (IWI Miner(T, E))

Input:

-T, a weighted transactional dataset

Input:

-E, a maximum IWI-support threshold

Output:

-F, the set of IWI satisfying E

(1) F=0 /*Initialization*//*scan T and count the IWI-support of each item */
(2) count the infrequent weighted item sets with the support value.
(3) create header table which is a data structure which holds information about total weight values.
(4) for each transaction, create equivalent transaction.
(5) create an FP-Tree, for each transaction.
(6) Iterate the process until all transactions are traced.
(7) create conditional pattern base
(8) calculate weight value.
(9) obtain the infrequent item sets.

To reduce the complexity of the mining process, IWI Miner adopts an FP-tree node pruning strategy to early discard items (nodes) that could never belong to any itemset satisfying the IWI-support threshold. Hence, an item(i.e., its associated nodes) is pruned if it appears only in tree paths from the root to a leaf node characterized by IWI-support value greater than E.

L. Minimal Infrequent Weighted ItemSet Miner

The MIWIMining procedure is similar to IWIMining[1][11]. However, since MIWI Miner focuses on generating only minimal infrequent patterns, the recursive extraction in the MIWIMining procedure is stopped as soon as an infrequent itemset occurs. It finds both the infrequent itemsets and minimal infrequent itemset mining. The advantage of MIWI algorithm is reduction in generating the candidate sets, reducing the computational Time, improved the efficiency of algorithm performance compared to FP-Growth algorithm.

III. CONCLUSION

Weighted Itemset mining is an exploratory information mining system generally utilized for uncovering profitable connections among information. The main advantage to perform Infrequentitemset mining was to improve the profit of rarely found datasets in the transactions. The first attempt is to find the frequent item set mining and then discover the infrequent weighted item sets. Several frequent itemset mining algorithms as Apriori to FP-growth are there but generation of candidate sets is large. As per the analysis of all the existing algorithms MIWI is the most effective algorithm, which computes in very less computing time, improves the efficiency of performance when the database is large, computes the weighted transaction.

REFERENCES

[1] Agrawal R, Imielinski T, &Swami , A. “Mining association rules between sets of items in large databases”.In proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, 1993.
[2] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. 20th Int’l Conf. Very Large Data Bases (VLDB ’94), pp. 487-499, 1994.

[3] Luca Cagliero and Paolo Garza "Infrequent Weighted Itemset Mining using Frequent Pattern Growth", IEEE Transactions on Knowledge and Data Engineering, pp. 1- 14, 2013.
[4] Liu,G. , Lu ,H. , Yu ,J. X., Wang, W., & Xiao, X.. "AFOPT:An Efficient Implementation of Pattern Growth Approach", In Proc. IEEE ICDM'03 Workshop FIMI'03, 2003.
[5] BalazesRacz," nonordfp: An FP-Growth Variation without Rebuilding the FP-Tree**",** 2nd Int'l Workshop on Frequent Itemset Mining Implementations FIMI2004
[6] Grahne O. and Zhu J. "Efficiently Using Prefix-trees in Mining Frequent Itemsets", In Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining, 2004.
[7] Cornelia Gyorodi, Robert Gyorodi, T. Cofeey& S. Holban – "Mining association rules using Dynamic FP-trees" – în Proceedings of The Irish Signal and Systems Conference, University of Limerick, Limerick, Ireland, 30th June-2nd July 2003, ISBN 0-9542973-1-8, pag. 76-82.
[8] Grahne G. and Zhu J., "Efficiently Using Prefix-Trees in mining Frequent Item sets," Proc. ICDM 2003Workshop Frequent Item set Mining Implementations, (2003).
[9] A.Gupta, A. Mittal, and A. Bhattacharya, "Minimally Infrequent Itemset Mining Using Pattern-Growth Paradigm and Residual Trees," Proc. Int'l Conf. Management of Data (COMAD), pp. 57-68, 2011.
[10] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," Procedings of ACM SIGMOD Intnational Conference on Management of Data, ACM Press, Dallas, Texas, pp. 1-12, May 2000.
[11] Han, J. , Pei, J. , & Yin, Y. "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD '96), Page 205-216, 2000.

BIOGRAPHIES

A. Sakthi Nathiarasan received the B.Tech in Information Technology from Sri Krishna College of Engineering andTechnology, Coimbatore, India in the year of 2013. He is currently doing his M.E degree in Computer Science and Engineering in Adhiyamaan College of Engineering, Hosur and he published 5 research articles in international journals and conferences. His area of interests includes Adhoc Networks, Cryptography and Network security, Utility Mining, Genetic Algorithms and Autonomic Computing.

P. Kalaiyarasi received the B.E degree in Computer Science and Engineering from Sona College of Technology, Salem, India. After that she worked as Software Tester in Cognizant Technology Solutions, Chennai. She is currently doing her M.E degree in Computer Science and Engineering in Adhiyamaan College of Engineering, Hosur and she published 3 research articles in international journals and conferences. Her area of interests includes Software Engineering, Object Oriented Software Development, Data Mining and Optimization Techniques.

M. Manikandan is currently working as Assistant professor in the department of Computer Science and Engineering in Adhiyamaan College of Engineering, Hosur, India. He obtained his M.E Degree from Arunai Engineering College, Tiruvannamalai under Anna University. He is having an experience of about 7 years and published many research papers in various international journals and conferences. His areas of interests includes Artificial Intelligence, Neural Networks, Algorithm Analysis and Soft Computing.