Survey on Infrequent Itemset Mining Using Frequent Pattern Growth

S.Nandhini, Mr.M.Yogesh prabhu, Dr.S.Gunasekaran

Abstract— Itemset mining is an effective area of research due to its successful application in various data mining scenarios like finding association rules. There are two types of itemset mining namely, Frequent Itemset Mining and Infrequent Itemset Mining. The research society has focused on the Infrequent Weighted Itemset Mining problem. The infrequent weighted itemset are item sets whose frequency of occurrence in the analyzed data is less than or equal to a maximum threshold. Two algorithms are reviewed to find rare itemset, that are infrequent weighted itemset (IWI) and Minimal Infrequent Weighted Itemset (MIWI) and this is based on the frequent pattern-growth paradigm. Finally performance analysis of an algorithm has been shown in terms of execution time.

Index Terms— Itemset mining, Infrequent itemset, Frequent pattern growth, Association rule, Weighted Itemset, Minimal infrequent patterns, Residual trees.

1 Introduction

DATA MINING is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. Frequent itemsets are the items that appear frequently in the transactions. The main goal of frequent itemset mining is to identify all the itemsets in the transaction data set, which are frequently purchased. Item sets are defined as a non empty set of items. If itemset is with k-different items is termed as a k-itemset. For example {bread, butter, milk} may denoted as a 3-itemset in a supermarket transaction. The Apriori algorithm is the initial solution for the frequent pattern mining problem.

The problems of Apriori, which generates more candidate sets and require more scans of database. To overcome the problems of Apriori FP-Growth has been proposed. The uses of FP Tree data structure without any candidate generation and using only two database scans. Mining infrequent patterns is a challenging task because there are an enormous number of such patterns that can be derived from a known data set. More exclusively, the key issues in mining infrequent patterns are: (1) how to identify interesting infrequent patterns, and (2) how to efficiently discover them in large data sets.

- *S.Nandhii is currently pursuing masters degree program in Software Engineering in Coimbatore Institute of Engineering andTechnology, India,. E-mail: nandhiniunique@gmail.com*
- *Mr.M.Yogesh Prabhu is currently working in Computer Science engineering in Coimbatore Institute of Engineering andTechnology, India,. E-mail:yogeshprabhu1985@gmail.com*
- *Dr.S.Gunasekaran is currently working as head and Professor in Computer Science engineering in Coimbatore Institute of Engineering andTechnology, India,E-mail:gunaphd@yahoo.com*

2 Literature Survey

2.1 Association Rule Mining

Association Rule is an important type of knowledge representation to find implicit relationships among the items present in large number of transactions. Rakesh Agrawal et al.[2] introduced association rules for discovering regularities between products in large-scale transaction Given $I = \{i_1, i_2, \ldots\ldots i_n\}$ as the item space, which is the set of items, a transaction may be defined as the subset of *I.* The support of an itemset X in a dataset D, denoted as *support*D(X), is defined as *count*D(X)/|D|, where count D(X) is the number of transactions in D containing X. An itemset is to be frequent(large) if support is larger than a user-specified value (also called minimum support(*min_sup*)).An Association is the implication of the form [$X \rightarrow Y$, *sup* , *conf*], where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The support of $X \cup Y$ (*sup*) in the transactions is larger than *min_sup*, furthermore when *X* appears in transaction, *Y* is likely to appears in the same transaction with the probability of *confidence*.

2.2 Apriori Algorithm Overview

Apriori algorithm for frequent item set mining and association rule over transactional database. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. To identify the frequent item sets in the large transaction database. Two stages of Apriori algorithm, first stage count item occurrence and generate candidate item set and second count support candidate item.

Apriori Algorithm

I = {Input as item sets};
For (k = 2; L_k-1≠0; k++) do begin C_k = apriori gen (L_k-1);
For all transactions t ε D do begin C_t = sub set (C_k, t);
For all candidates c ε C_t do C.count++;
End L_k = {c ε C_k | c: count ≥ minsup}
end Output = Ụ$_k$ L_k;

2.3 FP-Growth Algorithm Overview

The FP-Growth Algorithm is also to find frequent itemsets without using candidate generations, thus improving performance. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the itemset association information.

This algorithm works as follows:First it compresses the input database creating an FP-tree instance to represent frequent items. After this first step it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately. Using this strategy, the FP-Growth reduces the search costs looking for short patterns recursively and then merging them in the long frequent patterns, offering good selectivity.

FP-Growth Algorithm performs:

Step 1: Construction of FP-tree.

Step 2: Extracts frequent itemsets directly from the FP-tree.

Construction of FP-Tree:

1. Create the root of the tree, labeled with "null".
2. Scan the database D a second time. (First time we scanned it to create 1-itemset and then L).
3. The items in each transaction are processed in L order (i.e. sorted order).
4. A branch is created for each transaction with items having their support count.
5. Whenever the same node is encountered in another transaction, we just increment the support count of the common node or Prefix.
6. To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links.
7. Now, The problem of mining frequent patterns in database is transformed to that of mining the FP-Tree.

In large databases, it's not possible to hold the FP-tree in the main memory. A strategy to cope with this problem is to firstly partition the database into a set of smaller databases (called projected databases), and then construct an FP-tree from each of these smaller databases.

2.4 Minimal Infrequent Itemset Mining

A new algorithm designed specifically for finding minimal infrequent itemset. They faces the issue of Infrequent Itemset Mining problem. The problem status that, consider $I= \{i_1,i_2,\ldots i_L\}$ be a set of items. An itemset is a subset $I \subseteq I$. The cardinality of I, denoted by $|I|$, is the number of items in the itemset. A dataset, $D = \{t_1,t_2,\ldots,t_R\}$, is a collection of R transactions of the form $t_i = (i,T_i)$, where *i* is the transaction identifier (TID) and $Ti \subseteq I$. We denote by $|D|$ the number of transactions in the dataset. Given an itemset I, a transaction T is said to contain I if $I \subseteq T$. The support set of an itemset I with respect to the dataset D is $D(I) = \{t_i \in D : I \subseteq T_i\}$.

Given a dataset D and an integer threshold τ, we say an itemset I is

τ-occurrent if $|D(I)| = \tau$

τ-frequent if $|D(I)| \geq \tau$

τ-infrequent if $|D(I)| < \tau$

MINIT (MINimal Infrequent iTemsets) is the algorithm developed to find minimal τ-infrequent itemsets. The following steps explains MINIT algorithm, Initially, a ranking of items is prepared by computing the support of each of the items and then creating a list of items in ascending order of support. Minimal τ-infrequent itemsets are discovered by considering each item i_j in rank order, recursively calling MINIT on the support set of the dataset with respect to i_j considering only those items with higher rank than i_j, and then checking each candidate MII against the original dataset.

2.5 Infrequent Weighted Itemset Mining

IWI Miner is a FP-growth-like mining algorithm that performs projection-based item set mining. FP-growth mining steps: 1. FP-tree creation 2. Recursive item set mining from the FP tree index. 3. IWI Miner discovers infrequent weighted item sets instead of frequent (unweighted) ones. FP-growth have been modified and new algorithm is 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,€)

Input-Weighted transaction dataset with support value €

1) F=0
2) Count item IWI (T)
3) Construct FP tree
4) For all weighted transaction
5) Calculate Equivalent transaction
6) For all transaction create and insert into FP tree

Output- Set of satisfying €

2.6 Infrequent Weighted Itemset Mining

MIWI Miner focuses on generating only minimal infrequent patterns, the recursive extraction in the MIWI Mining procedure is stopped as soon as an infrequent item set occurs. It finds both the infrequent item sets and minimal infrequent item set mining.

IWI mining (T, € ,Prefix)

Input- Tree, a FP tree

Output- The set of IWIs

1) F=0 initialization
2) Create header table holds for all items i in tree
3) Generate a new item set I with prefix and support of item i
4) I – Infrequent item
5) Construct I as conditional pattern and FP tree
6) Select the infrequent items from the set
7) Remove from Tree and finally apply recursive mining

3 COMPARISION RESULT

IWI Miner and MIWI Miner performance on standard synthetic datasets are analyzed. The comparison of MIWI Miner and MINIT performance, in terms of execution time, on synthetic data sets with different characteristics. Fig. 3.1 reports the execution times achieved by varying the maximum support threshold in the range [0, 200] on two IBM synthetic data sets with 100,000 transactions and two representative average transaction length values (i.e., 10 and 15). The synthetic data sets are characterized by a fairly sparse data distribution.

TABLE 1

MIWI MINER AND MINIT IN TERMS OF EXECUTION TIME

Algorithm	Support Threshold				
	0.1	0.2	0.3	0.4	0.5
MINIT	500	387	226	198	94
MIWI	104	85	53	41	24

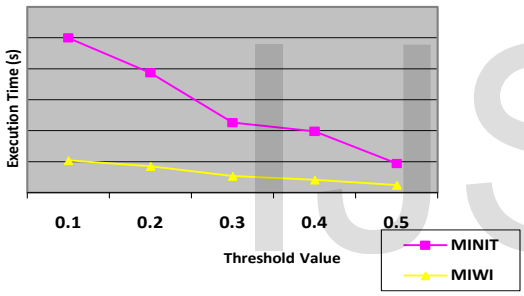

Figure 1: Comparison between MIWI Miner and MINIT in terms of execution time

4 CONCLUSION

This survey deals about the various frequent itemset mining algorithm and the new algorithm for finding minimal infrequent itemset are reviewed. Two FP Growth-like algorithms that accomplish IWI and MIWI mining efficiently are also reviewed and the performance analysis of MIWI and MINIT also showed with respect to execution time. The future work will be discovering maximum and minimum infrequent itemset by integrating existing algorithm with residual trees.

REFERENCES

[1] Savasere, E. Omiecinski, and S.B. Navathe, "An efficient algorithm for mining association rules in large databases," Intl. Conf. on Very Large Databases, pp. 432–444, 1995.

[2] Ashish Gupta, Akshay Mittal and Arnab Bhattacharya "Minimally Infrequent Itemset Mining using Pattern-Growth Paradigm and Residual Trees", 17th International Conference on Management of Data (COMAD), 2011.

[3] X. Wu, C. Zhang, and S. Zhang "Efficient mining of both positive and negative association rules", ACM Transaction Information System, vol. 22, issue 3, pp. 381–405, 2004.

[4] Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*. p. 207. doi:10.1145/170035.170072. ISBN 0897915925.

[5] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining Using Weighted Support and Significance Framework," Proc. nineth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '03), pp. 661-666, 2003.

[6] Luca Cagliero and Paolo Garza, "Infrequent Weighted Itemset Mining Using Frequent Pattern Growth," IEEE TRANSACTION ON KNOWLEDGE AND DATA ENGINEERING,VOL.26,NO.4, APRIL 2014

[7] D.J. Haglin and A.M. Manning, "On Minimal Infrequent Itemset Mining," Proc. Int'l Conf. Data Mining (DMIN '07), pp. 141-147, 2007.

[8] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 1-12, 2000.

[9] C.K. Chui, B. Kao, and E. Hung, "Mining Frequent Itemsets from Uncertain Data," Proc. 11th Pacific Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD '07), pp. 47-58, 2007.