A Survey on Frequent Patter Mining to Find Minimum Errors

S. Mounika[1],
M.E(CSE)1st Year[1],
Department Of Computer Science And Engineering,
K.S.Rangasamy College Of Technology,
Namakkal, India.

P. Senthil Raja[2],
Assistant Professor[2],
Department Of Computer Science And Engineering,
K.S.Rangasamy College Of Technology,
Namakkal, India.

***Abstract*— Association rule mining is the process of finding frequent patterns, associations, correlations, or connecting structures among sets of items or objects in transactional databases, relational databases, and additional information repositories. Association rule mining is also known as frequent pattern mining. Frequent Pattern Mining is used to mine the frequent patterns. Number of frequent pattern generated in frequent pattern mining depends on the frequent patterns which imposes a large challenges on visualizing, understanding and further analysis of the generated patterns. This leads to find a minimum representative pattern set. Frequent patterns have anti-monotone property. The assets states that if a pattern is frequent, then all of its subsets must also be frequent. The two algorithms MinRP set and FlexRP set, are used here to explain the problem in frequent pattern mining. MinRP set produces the smallest solutions that possibly have in practice and it takes a finite amount of time to terminate. The amount of time increases when the number of patterns is high. MinRP set is very space-intense and time-intense on some dense datasets when the number of frequent patterns is huge. MinRP set is similar to RPglobal. RPglobal scheme states a greedy method to discover representative patterns among the exposed frequent itemsets. FlexRP set is urbanized based on MinRP set. It provides one extra constraint K, which allows users to make a exchange between good association and the number of representative patterns.**

KeyTerms — Frequent pattern mining, MinRPset,FlexRpset.

I.INTRODUCTION

Frequent patterns proves an interesting relationships between attribute-value pairs that occur frequently in a given data set. Association rules are derived from frequent patterns, where the associations are commonly used to check up the purchasing patterns of customers in a store. Frequent itemsets play an vital role in many data mining tasks that try to discover motivating patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems[13]. The unique inspiration for searching association rules came from the require to study so called shop transaction data, that is, to examine customer actions in terms of the purchased products. Association rules explain how often items are purchased together. For example, an association rule “beer) chips (80%)” states that four out of five customers that bought beer also bought chips. Such rules can be helpful for decisions about product pricing, promotions, accumulate plan and many others. Frequent itemset mining algorithms can be categorized into three classes: 1) Apriori-based, parallel formatting method, with Apriori as its representative, 2) projection-based, horizontal formatting, pattern growth method, which may discover some compressed data structure such as FP-tree, as in FP-growth and 3) vertical formatting method, such as CHARM [13].The universal agenda among these methods is to use a min_support threshold to ensure the generation of the correct and complete set of frequent itemsets, based on the popular Apriori property [13]: Every subpattern of a frequent pattern must be frequent (also called the downward closure property). All the subsets of these frequent long patterns are frequent too based on the anti-monotone property. This leads to an unexpected increase in the number of frequent patterns. The enormous amount of patterns can simply become a restricted access for understanding and added analyze frequent patterns[1]. A pattern X is a set of items in I, that is, $X \leq I$. If a transaction t Є D contains all the items of a pattern X, then we say t supports X and t is a supporting transaction of X. Let T(X) be the set of transactions in D behind pattern X. The support of X, denoted as supp(X), is defined as |T(X)|. If the support of a pattern X is larger than a user-particular entry min _sup, then X is called a frequent pattern[1]. Entire set of frequent patterns frequently contains a lot of redundancy. It is attractive to group parallel patterns together and represent them using one single pattern. Closed itemsets and non-derivable itemsets are lossless forms of compressing frequent itemsets, i.e. the full list of frequent itemsets and associated frequency counts (used for computing association rules) can be accurately derived from the dense representation. Maximal itemsets permit better compression when compared with closed patterns, but the representation is lose. In MinRPset and FlexRPset, a representative pattern can represent its subsets only. To further reduce the number of representative patterns, we drop this form to allow a representative pattern to represent extra patterns.

II. BACKGROUND

Frequent pattern increase method which is used to discover the frequent itemsets without candidate generation. The FP-tree is mined by start0ing from each frequent length-1 pattern (as an initial suffix pattern), construct its conditional pattern base (a "subdatabase", which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then construct its conditional FP-tree,and the stage mining recursively on such a tree. Pattern-growth is one of several powerful frequent pattern mining methodologies, where a pattern (e.g., an itemset, a subsequence, a subtree, or a substructure) is frequent if its incidence frequency in a database is no less than a particular smallest amount supports entrance. The (frequent) pattern-growth method mines the data set in a divide-and-conquer way: It first derives the set of size-1 frequent patterns, and for each pattern p, it derives p's projected (or conditional) database by data set partitioning and mines the projected database recursively. Since the data set is decomposed progressively into a set of much smaller, pattern-related projected data sets, the pattern-growth method effectively reduces the explore space and leads to high efficiency and scalability. The approach has several distinct features:

1. The method preserves the essential groupings of the original data elements for mining.
2. The method partitions the data set to be examined as well as the set of patterns to be examined by database projection.

III.RELATED WORK

Apriori Approach

Apriori employs an iterative approach known as a level-wise search, where I-itemsets are used to explore (I+1) itemsets. The Apriori applicant produce-and-test method appreciably reduces the size of candidate sets, primary to good performance gain. On the other hand , it can suffer from two nontrivial costs:

1.It may still need to produce a enormous number of nominee sets

2.It may require to repeatedly look into the whole database and verify a huge set of nominee by pattern alike.

Profile-Based Approach:

The amount of frequent patterns can be very bulky. The number of pattern generators is better than that of closed Patterns[2]. The number of non-derivable patterns can also be bigger than that of closed patterns on some datasets. The number of maximal patterns is much smaller than the number of blocked patterns. All frequent patterns can be in good health from maximal patterns, but their maintained information is lost[9]. Profiles to review patterns. A review consists of a master pattern, a support and a possibility sharing vector, which contains the possibility of the objects in the master pattern. The set of patterns represented by a profile are subsets of the master pattern, and their support is designed by multiplying the support of the profile and the possibility of the equivalent items. There are several drawbacks with this profile-based approach:1)It makes conflicting assumptions.2)There is no error assurance on the projected support of patterns.3)The proposed algorithm for generating profiles is very slow because it needs to scan the original dataset repeatedly. 4) The boundary between frequent patterns and rare patterns cannot be resolute using profiles.

Reconfigurable Platform:

Mimic the inner memory layout of the original pattern mining software algorithm while achieving a higher throughput .Reconfigurable Systolic architecture for frequent pattern mining. Systolic tree structure is used to store the support counts for nominee patterns in pipelined approach that reads the support counts and takes pruning assessment . A Field Programmable Gate Array is an integrated circuit designed to be configured by a customer or a designer after developed – hence "field-programmable". The FPGA configuration is generally particular using a hardware description language (HDL). A model is described using a FGPA platform. The advantage of FGPA is to parallelize algorithms at the operand level granularity. The advantages are: 1)It decreses the mining time. 2)The mining speed of the systolic tree was several times faster than the FP-tree for long frequent patterns**.**

Unordered trees:

The aim is to mine the restrictedly embedded sub tree patterns from a set of rooted labeled unordered trees. Apriori based techniques are used to produce all candidate sub trees level by level during two efficient rightmost expansion operation. Tree matching and pattern matching in general are very valuable operations in these applications. The techniques used here are:1) Restrictedly implanted subtrees is used to find the hidden relationships in unordered trees. 2) FRESTM- frequent restrictedly implanted is algorithm used to solve the tree mining problem with value to time and space complexities.

The advantages are: 1) Formulation of a new frequent restrictedly implanted subtree mining problem.2) Integral design of a set of techniques based on the Apriori standard. The disadvantages are: 1) Tree size becomes larger, more patterns are found. 2) More time is spent in the mining process. 3) More time is spent on discovering the frequent subtrees.

Mining colossal frequent patterns:

Colossal patterns are dangerous to many applications, mainly in domains like bioinformatics. Large patterns are called colossal patterns. A mining advance called Pattern-Fusion is used to capably find a excellent estimate to the huge patterns. Pattern-Fusion is able to fuse minute

frequent patterns into massive patterns by taking leaps in the pattern search space. The advantages are: 1) Pattern-fusion gives elevated quality colossal pattern mining..2) Determine the distance between two arbitrary pattern sets. The disadvantages are: 1) Redundancy problem. 2) Downward closure property leads to an explosive number of frequent patterns.

Maximum length frequent itemsets:

The use of frequent item sets has been restricted by the high computational cost as well as the large number of resulting item sets. Here is to discover the frequent item sets with a maximum length. It generates the maximum length frequent item sets by adapting a pattern section growth line of attack based on the FP-tree structure. The techniques used are: 1) Conditional Pattern base Pruning is mostly used to prune the conditional transactions and also used to identify the frequent itemset longer than frequent itemset detected so far. 2) Frequent Item Pruning is used to find all frequent items in the provisional pattern base. This imposes a stricter condition on the selection of conditional transaction. The advantages are: 1) Maximum Length of frequent item sets can be professionally well-known even if the database is very large . 2) Mining long frequent item sets is advantages for FP-growth method. The disadvantage is that it consumes more time and space.

Top-K Frequent Closed Itemsets:

Mining task here is to supply the top-k frequent closed item sets of length no less than min_l.TFP is developed for mining such item sets without mins_support. Mine all the item sets instead of only the closed ones. There is no lowest amount of length constraint. The technique used here is search space pruning method. The advantages are: 1) The small transactions are not incorporated. 2)It gives high presentation.

IV.PROBLEM STATEMENT

Let I = $\{i_1, i_2, \ldots, i_n\}$ be a set of its itemsets. An itemset X is a nonempty subset of I. The length of itemset X is the number of items enclosed in X, and X is called an l-itemset if its length is l. A transaction database TDB is a set of transactions. An itemset X is enclosed in transaction<tid , Y> if $X \leq Y$. A pattern is closed if it is more frequent than all of its supersets. If a pattern X1 is non-closed, then there exists another pattern X2 such that X1<X2 and supp(X2) = supp(X1). X be a set of some items in I. From an association rule , implication of form X I_j , where X is a set of items in I and I_j is a single item in I that is not present in X.

Given the set of transactions T, generating all rules that satisfy constraints of two different forms:

1. *Syntactic Constraints:*

These rules involve limitations on items that can become visible in a rule. For example, we may be engrossed only in rules that have a specific item I_x appearing in the following, or rules that have a specific item I_y appearing in the predecessor.

2. *Support Constraints:*

These constraints alarm the number of transactions in T that support a rule. The support for a rule is distinct to be the part of transactions in T that convince the union of items in the consequent and antecedent of the rule. Support should not be confused with confidence. While confidence is a measure of the rule's strength, support corresponds to statistical significance.

The distance between two patterns is defined based on their supporting transaction sets.

Definition 1(D(X 1, X2)): Given two patterns X1 and X2, the distance between them is defined as

$$D(X1, X2) = 1 - \frac{|T(X1) \cap T(X2)|}{|T(X1) \cup T(X2)|}$$

Definition 2 : Given a real number and two patterns X1 and X2, here X1 is by X2 if X1 $\in$ X2 and D(X1,X2).

The objective is to choose the smallest amount set of patterns. The selected patterns are representative patterns. Representative patterns need not to be frequent. Here the problem is to find the minimum representative pattern set. So the two algorithms are used.

V. RPGLOBAL AND RPLOCAL ALGORITHM

RPglobal and RPlocal are two algorithms which is used to find the representative pattern set. RPglobal first produces the set of patterns that can be enclosed by each pattern, and then employs the greedy algorithm [5]. The optimality of RPglobal is determined by the optimality of the greedy algorithm, so the solution produced by RPglobal is almost the most excellent solution which can be used. However, RPglobal is very time-consuming and space-consuming. If the number of frequent patterns is not large then it is feasible only. RPlocal is developed based on FPclose [6]. It integrates frequent pattern mining with representative pattern finding. RPlocal is very professional, but it produces more representative patterns than RPglobal. Due to their vast memory usage and time consuming these algorithms are not used. Instead MinRPset and FlexRPset are used.

VI. MINRPSET ALGORITHM

MinRPset is used to find the smallest solution by consuming less space and time. Let F be the set of frequent patterns in a dataset D with respect to threshold min sup, and F_1 be the set of patterns with maintain no less than min sup(1-) in D. Given a pattern X $\in$ F_1 and C(X) denote the set of frequent patterns that can be covered by X. Here C(X) $\in$ F. By downward closure property if X is frequent then X $\in$ C(X). The

following are the working steps of minimum representative pattern set:

RULE1: Mine all patterns in F_1 and generate C(X) which is a set of frequent patterns that X covers for each pattern $X \in F_1$.

RULE2: From above step the result obtained is $|F_1|$ sets. The elements of $|F_1|$ sets are frequent patterns in F.

RULE3: The greedy algorithm is used for polynomial time approximation algorithm.

Let C(X)s is the main blockage of the MinRPset algorithm when F and F_1 are large and to find C(X)s over a large F for a large number of patterns F_1. Inorder to improve the efficiency of MinRPset the closed patterns alone should be considered, a structure called CFP-tree to find C(X)s should be used and a light-weight compression technique to compress C(X)s.

VII. THE FLEXRPSET ALGORITHM

While the number of frequent patterns is large on a dataset, the MinRPset algorithm becomes very slow due to search the subsets over a large CFP-tree for a huge number of patterns. Fitting C(X)s into a main memory becomes a bottleneck. To solve this problem, instead of searching C(X)s for all closed patterns can selectively generate C(X)s such that every frequent pattern is covered a sufficient number of times, in the hope that the greedy set cover algorithm can still find a near-optimal solution. The fewer the number of C(X)s generated, the more efficient the algorithm is. This is the basic idea of the FlexRPset algorithm. The FlexRPset algorithm uses a parameter *K* to control the minimum number of times that a frequent pattern needs to be covered. The depth-first order to traverse a CFP-tree from left to right. With the increase of parameter *K*, more information are gathered, hence less representative patterns are generated. The operation time of FlexRPset becomes longer. It is observed that the C(X)s generated at a smaller *K* value can be re-used at a larger *K* value. This leads to the incremental FlexRPset algorithm. It starts from *K*=1 and works like FlexRPset. If the number of representative patterns generated at *K*=1, then it stops. Otherwise, it increases *K* to 10 and generates C(X)s. The FlexRPset algorithm uses a parameter K to control the minimum number of times that a frequent pattern needs to be covered.

VIII .CONCLUSION

Here is to find the minimum representative pattern sets with minimum error guarantee. The two algorithms namely MinRPset and FlexRPset are used. Both algorithms first mine frequent patterns, and then find representative patterns in a post-processing step, while RPlocal integrates frequent pattern mining with representative pattern finding. Due to the use of the post-processing strategy, MinRPset and FlexRPset have the following additional benefits besides producing fewer representative patterns:

1) Users may not know what value should be used at the beginning. The post-processing strategy allows users to try different values without mining frequent patterns multiple times. This is especially beneficial on very large datasets.

2) In MinRPset and FlexRPset, it is easy to keep record of the set of patterns covered by each representative pattern. This information is useful for users to inspect individual representative patterns in more details.

3) We can relax the conditions on covered to further reduce the number of representative patterns .

MinRPset and FlexRPset have some drawbacks. On some dense datasets, MinRPset and FlexRPset with a large *K* value are often much slower than RPlocal. Both MinRPset and FlexRPset create fewer representative patterns than earlier work RPlocal. MinRPset is often more expensive than RPlocal. FlexRPset takes one extra parameter *K*, which allows users to make a trade-off between result size and running time. Users can make the trade-off conveniently using the incremental approach. When *K* is small, FlexRPset is usually faster than or has similar running time with RPlocal. Allow the users to relax the conditions in the problem definition to further reduce the number of representative patterns. Hence this approach is a very flexible approach to finding representative patterns.

REFERENCES

[1] Association Rules between Sets of Items in Large Rakesh Agrawal, Tomasz Imielinski and Arun Swami “Mining Databases”.
[2] Bart Goethals and Mohammed J. Zaki “advances in Frequent Itemset Mining Implementations:Report on FIMI’03”.(ref2)
[3] Artur Bykowski and Christophe Rigotti “A Condensed Representation to Find Frequent Patterns”2001.(ref9)
[4] Jiawei Han and Micheline Kamber “Data Mining: Concepts and Techniques(2nd edition)”2006.(ref4)
[5] Chao Wang and Srinivasan Parthasarathy “Summarizing Itemset Patterns Using Probabilistic Models”2006.(ref20)
[6] Guimei Liu, Haojun Zhang, and Limsoon Wong “A Flexible Approach to Finding Representative Pattern Sets”2014.(base paper)
[7] Agrawal R. Imielinski T. and Swami A. N. (2011) “Mining Association Rules between Sets of Items in Large Databases”, in Proceeding ACM SIGMOD, pp. 207-216.
[8] Agarwal R. Aggarwal C. and Prasad V. V. V. (2011), “Depth First Generation of Long Patterns”. in Proceeding ACM SIGMOD, pp.108-118.
[9] Agrawal R. and Srikant R. (2012)”Fast Algorithm for Mining Association Rules” in Proceeding VLDB, pp 487-499.
[10] Brin S. R., Motwani, J. Ullman and Tsur S. (2010) ”Dynamic Itemset Counting and Implication Rules for Market Basket Data”in Proceeding ACM SIGMOD, pp. 255-264.
[11] Feida Zhu, Xifeng Yan, Jiawei Han, Philip S. Yu and Hong Cheng(2004) , “Mining Colossal Frequent Patterns by Core Pattern Fusion”, PP.5-22.
[12] Han J. and Yin.Mining Y. (2010) “Frequent Patterns without Candidate Generation”. In Proceeding ACM SIGMOD, pp. 1-12.
[13] Jin, Abu-Ata M., Xiang Y. and Ruan N. (2012) “Effective and efficient itemset pattern summarization: Regression-based approaches,” in Proceeding KDD, Las Vegas, pp. 399–407, USA.
[14]D. Xin, H. Cheng, X. Yan, and J. Han, “Extracting redundancy aware top-k patterns,” in Proc. KDD, Philadelphia, PA, USA, 2006, pp. 444–453.
[15] F. N. Afrati, A. Gionis, and H. Mannila, “Approximating a collection of frequent sets,” in *Proc. KDD*, Washington, DC, USA, 2004,pp. 12–19.

[16] J. Pei, G. Dong, W. Zou, and J. Han, "Mining condensed frequent pattern bases," *Knowl. Inform. Syst.*, vol. 6, no. 5, pp. 570–594, 2004.
[17] X. Yan, H. Cheng, J. Han, and D. Xin, "Summarizing itemset patterns:A profile-based approach," in *Proc. KDD*, Chicago, IL, USA,2005, pp. 314–323.
[18] R. Jin, M. Abu-Ata, Y. Xiang, and N. Ruan, "Effective and efficient itemset pattern summarization: Regression-based approaches," in *Proc. KDD*, Las Vegas, NV, USA, 2008, pp. 399–407.
[19] A. K. Poernomo and V. Gopalkrishnan, "CP-summary: A concise representation for browsing frequent itemsets," in *Proc. KDD*, New York, NY, USA, 2009, pp. 687–696.
[20] C. Wang and S. Parthasarathy, "Summarizing itemset patterns using probabilistic models," in *Proc. KDD*, Philadelphia, PA, USA, 2006, pp. 730–735.
[21] M. Mampaey, N. Tatti, and J. Vreeken, "Tell me what I need to know: Succinctly summarizing data with itemsets," in *Proc. KDD*, San Diego, CA, USA, 2011, pp. 573–581.
[22] R. Rymon, "Search through systematic set enumeration," in *Proc.KR*, 1992, pp. 539–550.
[23] J. Wang, J. Han, and J. Pei, "Closet+: Searching for the best strategies for mining frequent closed itemsets," in *Proc. KDD*, New York, NY, USA, 2003, pp. 236–245.
[24] T. Westmann, D. Kossmann, S. Helmer, and G. Moerkotte, "The implementation and performance of compressed databases,"*SIGMOD Rec.*, vol. 29, no. 3, pp. 55–67, 2000.
[25] K. Zhao, B. Liu, J. Benkler, and W. Xiao, "Opportunity map:Identifying causes of failure—a deployed data mining system,"in *Proc. SIGKDD*, New York, NY, USA, 2006, pp. 892–901.