A Survey on Finding Representative Pattern Sets in Frequent Pattern Mining

Mrs. Neha Chaure[1], Prof. Dr.S.S. Sane[2]

[1]Computer Department K.K.Wagh Instutute Of Engineering Education & Research, Nashik

Abstract- *A large number of frequent patterns are produced by frequent pattern mining. So challenge is visualization, understanding and analysis of the generated patterns. Therefore find a small number of representative patterns that best approximate all other patterns. Thus to find a minimum representative pattern set, MinRPset algorithm is developed. This algorithm provides error guarantee. If the number of frequent closed patterns is less than one million then the solution produced by MinRPset is smallest. It also takes a sufficient amount of time to finish. MinRPset is space consuming as well as time consuming on some dense datasets. The dense dataset contains large number of frequent closed patterns. The another algorithm called FlexRPset is developed to solve this problem. This algorithm provides one extra parameter K which allows users to make a trade-off between resultant size and efficiency. For making the trade off conveniently, an incremental approach can be adopted. The algorithm MinRPset and FlexRPset produces fewer representative patterns than RPlocal which is an efficient algorithm developed for solving the same problem.*

Index Terms*-Frequent pattern mining, Representative patterns*

I. INTRODUCTION

The Frequent pattern mining is important in the data mining area. Firstly it was introduced by Agrawal et al. in 1993 [1]. Usually frequent pattern mining is performed on a transaction database $D = \{t_1, t_2, \ldots, t_n\}$, where t_j is a transaction containing a set of items, $j \in [1, n]$. Let $I = \{i_1, i_2, \ldots, i_m\}$ be the set of distinct items. These items are appearing in D. A pattern X is a set of items in I, i.e. $X \subseteq I$. If a transaction $t \in D$ contains all the items of a pattern X, then it is said that t supports X and t is a supporting transaction of X. Let T(X) be the set of transactions in D which supports pattern X. The support of X is denoted as supp(X) and is defined as $|T(X)|$. If the support of a pattern X is larger than a user-specified threshold min_sup, then X is called a frequent pattern. The task of frequent pattern mining is tofind all the frequent patterns in D with respect to min_sup from given a transaction database D and a minimum support threshold min_sup.

Many efficient algorithms are developed for mining frequent patterns [2]. Now the focus is on how to efficiently mine frequent patterns and howto effectively utilize them. Frequent patterns have the anti-monotone property. This property states that if a pattern is frequent then all of its subsets must be frequent too.

The complete set of frequent patterns contains a lot of redundancy. Many frequent patterns have similar items and supporting transactions. So it will be better to group similar patterns together and representthem using one single pattern. This is the concept of frequent closed pattern [3]. Let X be a pattern and S be the set of patterns appearing in the same set of transactions as X, i.e. $S = \{Y|T(Y) = T(X)\}$. The longest pattern contain in S is called a closed pattern and all the other patterns in S are subsets of it. The closed pattern of S is selected. This pattern can represent all the patterns in S. All the frequent patterns and their exact support can be recovered from the set of frequent closed patterns.
For finding a minimum representative pattern set, two algorithms are developed MinRPset and FlexRPset. MinRPset uses a tree structure called CFP-tree (Condensed Frequent Pattern Tree) [7] to store frequent patterns compactly. Also CFP tree structure supports efficient retrieval of patterns that are δ covered by a given pattern.

The MinRPset is only slower than RPlocal. The algorithm FlexRPset is developed based on MinRPset. To make a tradeoff between efficiency and the number of representative patterns selected, it provides one extra parameter k. When $K = \infty$, then FlexRPset approaches MinRPset. FlexRPset becomes faster when k decreases, but it produces more representative patterns. FlexRPset is faster than RPlocal when k=1.

II. LITERATURE SURVEY

Many approaches are used to developeefficient algorithms for mining frequent patterns. They can be classified as the Apriori family algorithms and the pattern-growth based algorithms.

Agrawal et al. [1] proposed Apriori algorithm on the basis of Apriori property. This algorithm is an iterative algorithm. This algorithms takes a transactional dataset and support threshold (min_sup) as an input and output is exact matching frequent patterns which contain support greater thanmin_sup. It assumes that the dataset is very well preprocessed and noise free. But in real world, datasets are dirty and contain missing and noisy values.

It is difficult for users to set min-sup threshold to obtain the desired results. If min-sup is set too large, then there may be a small number of frequent patterns and this is not the desirable result. If the min-sup is set too small, then there may be many redundant un-useful frequent patterns. This not only take a large processing time for mining but also increase the complexity of filtering un-interesting frequent patterns. Apriori works in two phases. In the first phase, it generates all possible itemsets combinations. These combinations of itemsets will act as possible candidates. These candidates will be used in latter phases. In the first phase, the minimum support is applied. This support is applied to find all frequent itemsets in a database and later on rules in Apriori algorithm are formed. These rules are formed using frequent itemsets and the minimum confidence constraint. The main drawback of Apriori is that the generation of large number of candidate sets. The efficiency of Apriori can be improved by using monotonicity property, hash based technique, partioning methods and so on.

The drawback of Apriori can be overcomed by Frequent pattern Growth algorithm[21].This algorithm is implemented without generating the candidate sets. For performing mining, this algorithm uses a tree structure called FP (Frequent Pattern) tree structure. The information collected from the database is stored in FP tree. Initially this algorithm scans the transaction database only once and collects the set of frequent items F and their supports. Then sort these frequent itemsets in descending order based on the support count. Therefore this algorithm reduces the number of candidate set generation, number of transactions and number of comparisons. The pattern growth approach mines all frequent itemsets on the basis of FP-tree. The main strategy of pattern growth approach is that it traverses search space in depth first order and on each node of search space it mines frequent itemsets on the basis of conditional patterns and creates child FP-tree. The major advantage of using pattern-growth approach is that it removes the costly candidate generate operation, which is required in Apriori algorithm.

The two greedy methods called RPglobal and RPlocal are developed by Xin. These guarantees compression bound but with higher computational complexity. RPlocal is close to RPglobal. These algorithms can reduce the number of closed frequent Patterns. The two steps are performed in RPglobal method. The firststep is to find all the frequent patterns that can cover itemset. The second step is to find the set-covers that is to find the set of representative patterns. In RPlocal, the depth-first search scans each pattern twice. In the first scan, visit from its parent and in the second scan, visit after finishing the calls to its children. After a pattern is visited in its second time, all the patterns that can possibly cover it have been enumerated. The other patterns are not able to cover it. So output a pattern in its second visit. The RPlocal sequentially scans the output patterns. During scanning, at any time when an uncovered pattern (calledprobe pattern) is found then the algorithm finds the current largest set that is a representative pattern.This pattern covers the largest set. The difference between the RPlocal method and the RPglobal method is the selections of the probe patterns.

Han et al. developed an FP-growth method that mines complete set of frequent itemsets without generating candidate. FP-growth uses technique called divide and conquer. In this, first database is scanned. After scanning, a list of frequent items is generated. In this list, items are ordered by frequency in descending order. Then this frequency descending list is compressed into a tree called frequent pattern tree or FP-tree.

Frequent pattern mining is a time-consuming process mainly for very large datasets. Therefore it is better to use a strategy “mining once and using many times". A main challenge while mining frequent patterns from large data set is that such mining generates a large number of patterns by considering min_sup threshold. This happens when min_sup is set low because if a pattern is frequent then its subpatterns are also frequent. A large pattern contains exponential number of smaller and frequent sub-patterns. This problem is overcomed by using concepts such as closed frequent pattern mining and maximal frequent pattern mining.

A pattern X is a closed frequent pattern in a database D if X is frequent in D and there does not existproper super-pattern Y such that support of X and Y is same in D. A pattern X is a maximal frequent pattern in database D if X is frequent and there does not exist super-pattern Y such that $X \subset Y$ and Y is frequent in D. X is also called as max-pattern. The set of max-patterns is more compact. This set does not contain the complete support information. This information is related to all corresponding frequent patterns of it. Pasquier et al. proposed mining of frequent closed itemsets.

The other pattern-growth method for mining sequential pattern called Prefix Span (i.e. Prefix-projected Sequential Pattern mining) is developed by Jian Pei. PrefixSpan mines the complete set of patterns without generating candidate subsequence. It is efficient and faster than Apriori based approach. The FreeSpan (Frequent pattern-projected Sequential Pattern mining) reduces the candidate subsequence generation. FreeSpan is based on the property that if an itemsetX is infrequent then any sequence whose projected itemset is a superset of X cannot be a sequential pattern. For a sequence $A=(s_1 \ldots s_i)$, the itemset $s_1 U \ldots U\ s_i$ is called A’s projected itemset. FreeSpan mines sequential patterns by partitioning the search space and projecting the sequence subdatabases based on the projected itemsets. This is done recursively.

To reduce pattern set size other concepts, such as generators [8], disjunction-free generators [9], δ-free sets [10],non-derivable patterns [11], maximal patterns [12], top-k frequent closed patterns [13] and redundancy-aware top-k patterns [14] are proposed . The number of non-derivablepatterns is larger than that of closed patterns on some datasets. The number of maximal patterns is smaller than the number of closed patterns. All frequent patterns can be recovered from maximal patterns. But their support information is lost.

Several approaches have been developed to make a tradeoff between pattern set size and the precision of pattern support. One such approach is developed by Xin et al. [4] . Another approach is proposed by Peiet al. [15]. It uses absolute error bound and heuristic algorithms to mine a minimal condensed pattern-base. This is a superset of the maximal pattern set. All the frequent patterns and their support can be restored from a condensed pattern-base with error guarantee.

Yan et al. [16] uses profiles for summarizing patterns. A profile consists of a master pattern, a support and a probability distribution vector. A profile-based summarization finds k representative patterns. These are called as master patterns. A probability distribution vector contains the probability ofthe items in the master pattern. The set of patterns represented by a profile are subsets of the master pattern and their support is calculated by multiplying the support of the profile and the probability of the corresponding items. For summarizing a collection of patterns using k profiles, Yan et al. partition the patterns into k clusters and uses a profile to describe each cluster. The advantage is that this profile model is capable of recovering frequent patterns plus their supports. There are many drawbacks with this profile-based approach. Some drawbacks are: it makes contradictory assumptions. The patterns represented by the same profile are supposed to be similar in both item composition and supporting transactions. So the items in the same profile are expected to be strongly correlated. The support of patterns is calculated from a profile. The items in the same profile are expected to be independent. There is no error guarantee on the support of patterns. The algorithm for generating profiles is very slow because it needs to scan the original dataset repeatedly. The boundary between frequent patterns and infrequent patterns cannot be determined using profiles.

Several improvements are made to the profile based approach. Jin et al. [17] developed a regression based approach to minimize restoration error. They cluster patterns based on restoration errors instead of similarity between patterns So the lowerrestoration error can be achieved. But there is no error guarantee on the restored support. CP-summary [18] uses conditional independence. The conditional independence is used to reduce restoration error. It adds one more component to each profile: a pattern base and the new profile is called c-profile. The items in a c-profile are expected to be independent with respect to the pattern base. CP-summary provides error guarantee on calculated support. Wang et al. [19] make generalization on another representation of frequent patterns i.e. non-derivable patterns. Wang et al. use Markov Random Field (MRF) to summarize frequent patterns. The support of a pattern is calculated from its subsets. This is similar to non-derivable patterns. Markov Random Field model is expensive to learn. It does not provide error guarantee on calculated support. The goal of Mampaey et al. [20] is to summarize data instead with a collection of non-redundant patterns. A probabilistic maximum entropy model is used in this approach.

III. CONCLUSION

The main problem in frequent pattern mining is that it needs to scan database repeatedly and also generation of candidates. Several algorithms are developed to solve the frequent itemset mining problem as efficiently as possible. The pattern-growth approach is more efficient than Apriori. It uses a divide-and-conquer strategy to project and partition a large database recursively into a smaller set of patterns. RPglobalis time-consuming and space-consuming. RPlocal is very efficient, but it produces more representative patterns than RPglobal. The Frequent pattern growth uses FP-tree. FP-tree is used to Preserve complete information for frequent pattern mining. For mining frequent closed itemsets, FPclose algorithm is developed .FreeSpan and PrefixSpan are projection based approaches. They reduce the candidate sequence generation. The Markov Random Field model is easy to understand but not easy as compared to profiles . For finding minimum representative pattern sets two algorithms, MinRPset and FlexRPset are developed. Both these algorithms mines frequent patterns first and then find representative patterns. They use CFP-tree to store and retrieve frequent patterns.

ACKNOWLEDGMENT

I would like to take this opportunity to express my profound gratitude and deep regard to Prof.Dr.S.S.Sane,head of computer engineering department, K.K.Wagh Institute of Engineering Education and Research, Nashik for his exemplary guidance, valuable feedback and encouragement. I express my warm thanks to Prof.Dr.K.N.Nandurkar, Principal, K. K. Wagh Institute of Engineering Education and Research,Nashik for their support and guidance. I would also like to give my sincere gratitude to all the friends and colleagues.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in Proc. SIGMOD, Washington, DC, USA, 1993, pp. 207–216.

[2] B. Goethals and M. J. Zaki, "Advances in frequent itemset mining implementations: Introduction to FIMI'03," in Proc. ICDM, 2003.

[3] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules," in Proc. 7th ICDT, Jerusalem, Israel, 1999, pp. 398–416.

[4] D. Xin, J. Han, X. Yan, and H. Cheng, "Mining compressed frequent-pattern sets," in Proc. 31st Int. Conf. VLDB, Trondheim, Norway, 2005, pp. 709–720.

[5] V. Chvatal, "A greedy heuristic for the set-covering problem," Math.Oper.Res., vol. 4, no. 3, pp. 233–235, 1979.

[6] G. Grahne and J. Zhu, "Efficiently using prefix-trees in mining frequent itemsets," in Proc. FIMI, 2003.

[7] G. Liu, H. Lu, and J. X. Yu, "CFP-tree: A compact disk-based structure for storing and querying frequent itemsets," Inf. Syst., vol. 32, no. 2, pp. 295–319, 2007.

[8] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal, "Mining minimal non-redundant association rules using frequent closed itemsets," in Proc. 1st Int. Conf. CL, London, U.K., 2000, pp. 972–986.

[9] A. Bykowski and C. Rigotti, "A condensed representation to find frequent patterns," in Proc. PODS, New York, NY, USA, 2001, pp. 267-273.

[10] J.-F. Boulicaut, A. Bykowski, and C. Rigotti, "Free-sets: A condensed representation of Boolean data for the approximation of frequency queries," Data Mining Knowl. Discov., vol. 7, no. 1, pp. 5–22, 2003.

[11] T. Calders and B. Goethals, "Mining all non-derivable frequent itemsets," in *Proc. PKDD*, Helsinki, Finland, 2002, pp. 74–85.

[12] R. J. Bayardo, "Efficiently mining long patterns from databases," in*Proc. SIGMOD*, New York, NY, USA, 1998, pp. 85–93.

[13] J. Wang, J. Han, Y. Lu, and P. Tzvetkov, "TFP: An efficient algorithm for mining top-k frequent closed itemsets," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 5, pp. 652–664, May 2005.

[14] D. Xin, H. Cheng, X. Yan, and J. Han, "Extracting redundancyaware 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][17] 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.

[18] 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.

[19] C. Wang and S. Parthasarathy, "Summarizing itemset patterns using probabilistic models," in *Proc. KDD*, Philadelphia, PA, USA,2006, pp. 730–735.

[20] 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

[21] Luca Cagliero and Paolo Garza "Infrequent Weighted Itemset

[22] Mining using Frequent Pattern Growth", IEEE Transactions on Knowledge and Data Engineering, pp. 1-14, 2013.