Resolving Top-k High Utility Item sets Mining Problem without Setting Minimum Utility Thresholds

[1]N.HARSHA VARDAN, [2] T. KISHORE BABU
[1]*M. Tech Student, Andhra Loyola Institute of Engineering and Technology, Krishna District, Andhra Pradesh, India*
[2]*Assistant Professor, Andhra Loyola Institute of Engineering and Technology, Krishna District, Andhra Pradesh, India*

ABSTRACT– ***The problem of frequent itemset mining is popular. But it has some important limitations when it comes to analyzing customer transactions. An important drawback is that purchase quantities are not taken into account. Thus, an item may only appear once or zero time in a transaction. A second major drawback is that all items are viewed as having the same importance, utility of weight. Thus, frequent pattern mining may find many frequent patterns that are not interesting. To address these limitations, the problem of frequent item set mining has been redefined as the problem of High-Utility Item set (HUI) mining. In this paper we propose top-k high utility item set mining, where k is the desired number of HUIs to be mined. In this propose work, we are using two efficient algorithms to mine HUIs without set minimum utility threshold.***

1. INTRODUCTION

Mining high utility item sets from a transactional database refers to the discovery of item sets with high utility like profits. Although a number of relevant approaches have been proposed in recent years, but they incur the problem of producing a large number of candidate item sets for high utility item sets. Such a large number of candidate item sets degrades the mining performance in terms of execution time and space requirement. The situation may become worse when the database contains lots of long transactions or long high utility item sets. An emerging topic in the field of data mining is utility mining which not only considers the frequency of the item sets but also considers the utility associated with the item sets. The main objective of High Utility Item set Mining is to identify item sets that have utility values above a given utility threshold. Thus Utility mining plays an important role in many real-time applications and is an important research topic in data mining system to find the item sets with high profit. In this paper we present a literature review of the present state of research and the various algorithms for high utility item set mining. In this paper we are proposing a new framework for Top-k high utility web access patterns, where k is the desired number of HUIs to be mined. Frequent item set mining maintains to the discovery of associations as well as correlations among items in large transactional and relational data sets. With large amounts of data continuously being collected as well as stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationship between large amounts of business transaction records can help in many decision making processes such as catalogue design cross marketing, and customer shopping behavior analysis. The common framework utilized for these algorithms is to use min_support threshold to ensure the generation of the correct and complete set of frequent item sets. However, it is very difficult for users to set an appropriate minimum support because it highly depends on data types. If it is set too high, no result item sets are found while too small value makes an enormous number of result patterns which cause inefficiencies in terms of computation time and memory usage. Thus, it requires multiple trials for users to find an appropriate minimum support value, which costs a lot[9].To address this issue, top-k frequent item set mining has been proposed. Top-k FIM mines the most frequent k item sets without using the minimum support value from the user. The research of the FIM has been developed into the weighted frequent pattern mining and progressed to the high utility item set mining (HUIM). In utility mining, each item is associated with a utility and an occurrence count in each transaction (e.g. quantity). The utility of an item set represents its importance, which can be measured in terms of weight, value, quantity or other information depending on the user specification. An item set is called high utility item set (HUI) if its utility is no less than a user-specified minimum utility threshold min_util. HUI mining is essential to many applications such as streaming analysis, market analysis, mobile computing and biomedicine. Although top-k HUI mining is essential to many applications, developing efficient algorithms for mining such patterns is not an easy task.

Because of the massive, real-timing and dynamic property of data streams, mining algorithms over data streams needs to be more efficient on both running time and memory usage.

In this paper we propose two efficient algorithms such as TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in One phase) algorithms. The TKU algorithm adopts a compact tree-based structure named UP-Tree to maintain the information of transactions and utilities of item sets. TKU inherits useful properties from the TWU model and consists of two phases. In phase I, potential top-k high utility item sets (PKHUIs) are generated. In phase II, top-k HUIs are identified from the set of PKHUIs discovered in phase I. On the other hand, the TKO algorithm uses a list-based structure named utility-list to store the utility information of item sets in the database. It uses vertical data representation techniques to find out top-k HUIs in only one phase.

II. RELATED WORK

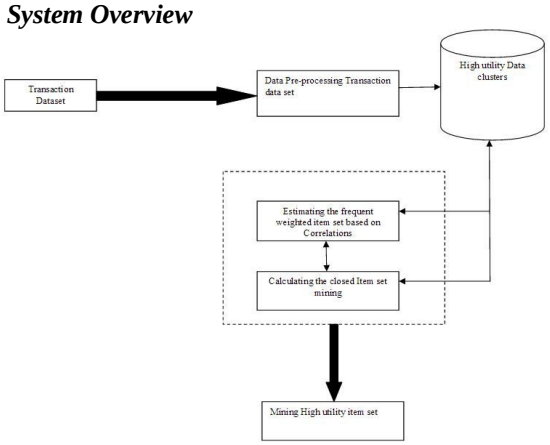
Mining high utility item sets from a transactional database refers to the detection of item sets with high utility like profits. However, a number of relevant approaches have been proposed in recent years, they incur the problem of producing a large number of candidate item sets for high utility item sets. Such a large number of candidate item sets degrades the mining performance in terms of execution time and space requirement. The condition may become worse when the database contains lots of long transactions or long high utility item sets. In this paper, we propose an efficient algorithm, namely UP-Growth (Utility Pattern Growth), for mining high utility item sets with a set of techniques for pruning candidate item sets. The information of high utility item sets is maintained in a special data structure named UP-Tree (Utility Pattern Tree) such that the candidate item sets can be generated efficiently with only two scans of the database. The performance of UP-Growth was evaluated in comparison with the state-of-the-art algorithms on different types of datasets. The experimental results show that UP-Growth not only reduces the number of candidates effectively but also outperforms other algorithms substantially in terms of execution time, especially when the database contains lots of long transactions.

R. Agrawal and R. Srikant, T. Imielinski proposed apriori algorithm, it is used to obtain frequent itemsets from the database. In miming the association rules this author have the problem to generate allassociation rules that have support and confidence greater than the user specified minimum support as well as minimum confidence respectively. The first pass of the algorithm simply counts item occurrences to determine the large 1-itemsets. First it generates the candidate sequences and then it chooses the large sequences from the candidate ones. Next, the database is scanned and the support of candidates is counted. The second step involves generating association rules from frequent item sets. Candidate item sets are stored in a hash-tree. The Hash-Tree node contains either a list of item sets or a hash tables. Apriori is a classic algorithm for frequent item set mining and association rule learning over transactional databases. After identifying the large item sets, only those item sets are allowed which have the support greater than the minimum support allowed. Apriori Algorithm generates lot of candidate item sets and scans database every time. When a new transaction is added to the database then it should rescan the entire database again.

Existing methods of association rule mining consider the appearance of an item in a transaction, whether or not it is purchased, as a binary variable. However, customers may buy more than one of the same item, and the unit cost may vary among items. Utility mining, a generalized form of the share mining model, attempts to overcome this problem. Since the Apriori pruning strategy cannot identify high utility item sets, developing an efficient algorithm is crucial for utility mining. This study proposes the Isolated Items Discarding Strategy (IIDS), which can be applied to any existing level-wise utility mining method to reduce candidates and to improve performance. The most efficient known models for share mining are SHFSM and DCG, which also work adequately for utility mining as well. By applying IIDS to SHFSM and DCG, the two methods FUM and DCG+ were implemented, respectively. For both synthetic and real datasets, experimental results reveal that the performance of FUM and DCG+ is more efficient than that of SHFSM and DCG, respectively. Therefore, IIDS is an effective strategy for utility mining.

III. FRAMEWORK

A. System Overview



In this paper, we propose two efficient algorithms named TKU (mining Top-K Utility itemsets) and

TKO (mining Top-K utility item sets in One phase) are proposed for mining the complete set of top-k HUIs in databases without the need to specify the min_util threshold. The TKU algorithm adopts a compact tree-based structure named UP-Tree to maintain the information of transactions and utilities of item sets. Second, we investigate the properties of the TKU and TKO algorithms and develop different strategies to effectively raise the border thresholds in both algorithms.

Advantages of the Framework:

1. Mining the complete set of top-k HUIs in databases without the need to specify the min_util threshold.
2. The construction of the UP-Tree and prune more unpromising items in transactions, the number of nodes maintained in memory could be reduced and the mining algorithm could achieve better performance.

B. Utility Pattern (UP)-Tree

To facilitate the mining performance and avoid scanning original database repeatedly, we will use a compact tree structure, named UP-Tree (Utility Pattern Tree), to maintain the information of transactions and high utility item sets. Two strategies are applied to minimize the overestimated utilities stored in the nodes of global UP-Tree.

Each node N in UP-tree consists of a node N:item, overestimated utilityN:nu, support count N:count, a pointer to the parent node N:parent and a pointer N:hlink to the node which has the same name as N:name. The root of the tree is a special empty node which points to its child nodes. The support count of a node N along a path is the number of transactions contained in that path that have the item N:item. N:nu is the overestimated utility of an item set along the path from node N to the root. In order to facilitate efficient traversal, a header table is also maintained. The header table has three columns, Item, TWU and Link. The nodes in a UP-tree along a path are maintained in descending order of their TWU values. All nodes with the same label are stored in a linked list and the link pointer in the header table points to the head of the list.

C. TKU Algorithm

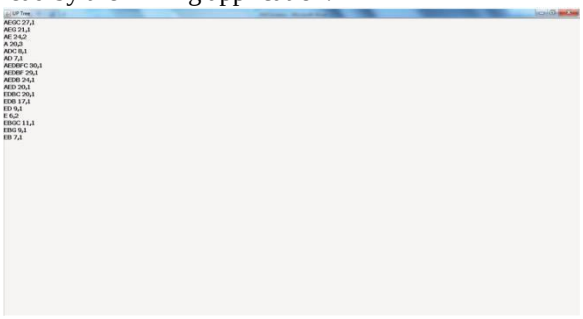
The TKU algorithm accepts a compact tree-based structure named UP-Tree to maintain the information of transactions as well as utilities of item sets.

D. TKO Algorithm

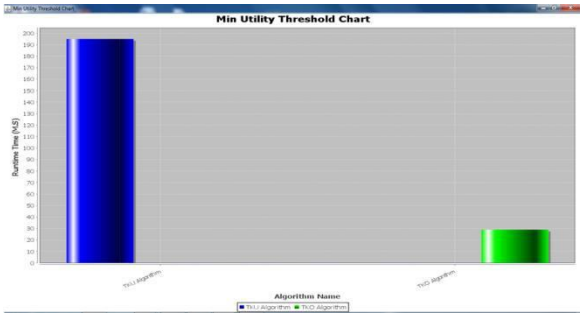
This algorithm can discover top-k HUIs in only one phase. It utilizes the basic search procedure of HUI-Miner and its utility-list structure. Whenever an item set is generated by TKO, its utility is calculated by its utility-list without scanning the original database. We first describe a basic version of TKO named TKO Base and then the advanced version, which includes several strategies to increase its efficiency.

IV. EXPERIMENTAL RESULTS

In this experiment we first read the transaction dataset and after we need to load profit database to read by the mining application.



From Transaction as well as Profit database, we can compute the Transaction utilities and we can construct the UP-tree.



By using UP-Tree results the TKU and TKO algorithms will work. Finally, we can see the top-k high utility itemsets without setting min_util thresholds.

V. CONCLUSION

In this paper, we conclude that we solved the top-k high utility item sets mining problem without setting min_util thresholds. For that, we proposed an efficient two algorithms named as TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in One phase). These two algorithms are used UP-Tree algorithm. From experimental results, we can say the proposed algorithms are efficient to mine the top-k high utility item sets.

REFERENCES

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[2] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, “Efficient tree structures for high-utility pattern mining in incremental databases,”IEEE Trans. Knowl. Data Eng., vol. 21, no. 12,

[3] K. Chuang, J. Huang, and M. Chen, "Mining top-k frequent patterns in the presence of the memory constraint," VLDB J., vol. 17, pp. 1321–1344, 2008.

[4] R. Chan, Q. Yang, and Y. Shen, "Mining high-utility itemsets," in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.

[5] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.

[6] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Mining top-k association rules," in Proc. Int. Conf. Can. Conf. Adv. Artif. Intell., 2012, pp. 61–73.

[7] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Novel concise representations of high utility itemsets using generator patterns," in Proc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput. Sci., 2014, vol. 8933, pp. 30–43.

[8] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2000, pp. 1–12.

[9] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," in Proc. IEEE Int. Conf. Data Mining, 2002, pp. 211–218.

[10] S. Krishnamoorthy, "Pruning strategies for mining high utility itemsets," Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015.

[11] C. Lin, T. Hong, G. Lan, J. Wong, and W. Lin, "Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases," Adv. Eng. Informat., vol. 29, no. 1, pp. 16–27, 2015.

[12] G. Lan, T. Hong, V. S. Tseng, and S. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," Expert Syst. Appl., vol. 41, no. 11, pp. 5071–5081, 2014.

[13] Y. Liu, W. Liao, and A. Choudhary, "A fast high utility itemsets mining algorithm," in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.

[14] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in Proc. ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.

[15] J. Liu, K. Wang, and B. Fung, "Direct discovery of high utility itemsets without candidate generation," in Proc. IEEE Int. Conf. Data Mining, 2012, pp. 984–989.

[16] Y. Lin, C. Wu, and V. S. Tseng, "Mining high utility itemsets in big data," in Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining, 2015, pp. 649–661.

[17] Y. Li, J. Yeh, and C. Chang, "Isolated items discarding strategy for discovering high-utility itemsets," Data Knowl. Eng., vol. 64, no. 1, pp. 198–217, 2008.

[18] J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W. K. Liao, A. Choudhary, and G. Memik, NU-MineBench version 2.0 dataset and technical report [Online]. Available: http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html, 2005.

[19] G. Pyun and U. Yun, "Mining top-k frequent patterns with combination reducing techniques," Appl. Intell., vol. 41, no. 1, pp. 76–98, 2014.

Mr. T.Kishore Babu is working as Assistant Professor in Andhra Loyola Institute of Engineering, Vijayawada.

Nadakudhiti V Harsha Vardhan was born in Vijayawada, Andhra Pradesh on June 12, 1994. he graduated from the Jawaharlal Nehru Technological University, Kakinada. His special fields of interest included Data Mining and Network Security. Presently he is studying M. Tech in Andhra Loyola Institute of Engineering, Vijayawada.