RESEARCH ARTICLE OPEN ACCESS

Mining Top K-HUI Patterns By using RUC Strategy

Riswana.P.P [1], Divya.M [2]
AWH Engineering College, KTU University
Department of Computer science & Engineering
Kuttikkatoor, Kozhikode, India

ABSTRACT

High utility itemsets (HUIs) mining is an emerging topic in data mining, which refers to discovering all itemsets having a utility meeting a user-specified minimum utility threshold min_util. However, setting min_util appropriately is a difficult problem for users. Generally speaking, finding an appropriate minimum utility threshold is a tedious process for users. If min_util is set too low, too many HUIs will be generated, which may cause the mining process to be very inefficient. On the other hand, if min_util is set too high, it is likely that no HUIs will be found. To address this limitation a novel algorithm TKOA is proposed for mining top K-HUIs in One phase without candidate Generation. Also another algorithm ALG2 is proposed for mining HUIs which is completely based on User Frequency.

Keywords *:—* Utility Mining, High Utility Itemset, Frequent Itemset Mining, Top –k pattern Mining.

I. INTRODUCTION

Data mining is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. Finding interesting patterns has been an important data mining task, and has a variety of applications, for example, genome analysis, condition monitoring, cross marketing, and inventory prediction, where interestingness measures play an important role. With frequent pattern mining a pattern is regarded as interesting if its occurrence frequency exceeds a user specified threshold. For example, mining frequent patterns from a shopping transaction database refers to the discovery of sets of products that are frequently purchased together by customers. However, a user's interest may relate to many factors that are not necessarily expressed in terms of the occurrence frequency.

Utility mining emerged recently to address the limitation of frequent pattern mining by considering the user's expectation or goal as well as the raw data. In utility mining, each item is associated with a utility (e.g. unit profit) and an occurrence count in each transaction (e.g. quantity). An itemset is called high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold min_util.

However, efficiently mining HUIs in databases is not an easy task because the downward closure property used in FIM does not hold for the utility of itemsets. In other words, pruning search space for HUI mining is difficult because a superset of a low utility itemset can be high utility. To tackle this problem, the concept of transaction-weighted utilization (TWU) model was introduced to facilitate the performance of the mining task. In this model, an itemset is called high transaction-weighted utilization itemset (HTWUI) if its TWU is no less than min_util, where the TWU of an itemset represents an upper bound on its utility.

Although many studies have been devoted to HUI mining, it is difficult for users to choose an appropriate minimum utility threshold in practice. Depending on the threshold, the output size can be very small or very large. Besides, the choice of the threshold greatly influences the performance of the algorithms. If the threshold is set too low, too many HUIs will be presented to the users and it is difficult for the users to comprehend the results. On the contrary, if the threshold is set too high, no HUI will be found. To find an appropriate value for the min_util threshold, users need to try different thresholds by guessing and re-executing the algorithms over and over until being satisfied with the results. This process is both inconvenient and time-consuming.

Most of the prior utility mining algorithms with the itemset share framework does not address the above challenges and adopt a two-phase, candidate generation approach, that is, first find candidates of high utility patterns in the first phase, and then scan the raw data one more time to identify high utility patterns from the candidates in the second phase. The challenge is that the number of candidates can be huge, which is the scalability and efficiency bottleneck.

To precisely control the output size and discover the itemsets with the highest utilities without setting the thresholds, a promising solution is to redefine the task of mining HUIs as mining top-k high utility itemsets (top-k HUIs). The idea is to let the users specify k, i.e., the number of desired itemsets, instead of specifying the minimum utility threshold. Setting k is more intuitive than setting the threshold because k represents the number of itemsets that the users want to find whereas choosing the threshold depends primarily on database characteristics, which are often unknown to users.

Therefore a novel Algorithm named TKOA is proposed which identify the complete set of top-k HUIs in one phase without the need to specify the min_util threshold. For TKO, the novel strategies RUC, RUZ and EPB for pruning the search space is integrated. TKOA is completely profit oriented, so in addition to TKOA another algorithm named ALG2 which is completely based on User Frequency is proposed for mining HUIs.

The rest of the paper is organized as follows. Section 2 presents the literature survey, followed by mining Top K-HUI patterns by RUC Strategy. Section 3.1 presents system Architecture. Section 3.2 presents TKOA which is completely profit oriented and Section 3.3 presents ALG2 algorithm which is based on user frequency followed by Conclusion and Future work.

II. RELATED WORKS

Related Works deals with the techniques which contribute to the development of Mining Top K- HUIs. There are a number of works related on different types of HUIs Mining. Some prior techniques introduced for mining High Utility Itemsets are briefly listed.

Junfu Yin, Zhigang Zheng, Longbing Cao [1] proposed an efficient algorithm, USpan, for mining high utility sequential patterns is specified and presented. USpan, introduces the lexicographic quantitative sequence tree to extract the complete set of high utility sequences and design concatenation mechanisms for calculating the utility of a node and its children with two effective pruning strategies. However compared to HUI mining, utility based sequence analysis faces critical combinational explosion and computational complexity.

Jingyu Shao, Junfu Yin, Wei Liu and Longbing Cao [2], proposed an algorithm named Combined Utility-Association Rule Mining (CUARM), which enables the generation of itemsets that are both high utility and are strongly associated, by considering the relationship between items/ itemsets, which provide users with actionable knowledge.

Philippe Fournier-Viger, Cheng-Wei Wu, Souleymane Zida, Vincent S.Tseng [3] proposed an algorithm named FHM which reduces the number of join operations by introducing a novel structure named EUCS(Estimated Utility Co-occurrence Structure). However FHM supports only static databases and is time-consuming.

Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow [4] proposed an efficient algorithm for mining high utility itemsets from transactional databases. In this work, two algorithms, namely utility pattern growth (UP-Growth) and UP-Growth+, is proposed for mining high utility itemsets with a set of effective strategies for pruning candidate itemsets. The information of high utility itemsets is maintained in a tree-based data structure named utility pattern tree (UP-Tree) such that candidate itemsets can be generated efficiently with only two scans of database. Since this framework requires multiple phases to scan the database it is highly complex.

Sen Su, Shengzhi Xu, Xiang Cheng, Zhengyi Li, and Fangchun Yang [5] proposed a private FP-growth algorithm, which is referred to as PFP-growth. The PFP-growth algorithm consists of a pre-processing phase and a mining phase. In the pre-processing phase, to improve the utility and privacy trade off, a novel smart splitting method is proposed to transform the database. For a given database, the pre-processing phase needs to be performed only once. In the mining phase, to offset the information loss caused by transaction splitting, we devise a run-time estimation method to estimate the actual support of itemsets in the original database. The limitation is that if itemsets of the same length is generated simultaneously, the no. of FP-trees stored in memory will grow at an exponential rate.

Junqiang Liu, Ke Wang, Benjamin C.M. Fung [6] proposed an algorithm, d^2HUP, namely Direct Discovery of High Utility Patterns, for mining high utility itemsets in one phase without candidate generation. It is an integration of the depth-first search of the reverse set enumeration tree, the pruning techniques that drastically reduces the number of patterns to be enumerated, and a novel data structure that enables efficient computation of utilities and upper bounds. However since d^2HUP searches unwantedly this is not suitable for large databases.

III. MINING TOP K-HUI PATTERNS BY USING RUC STRATEGY

To precisely control the output size and discover the itemsets with the highest utilities without setting the thresholds, a promising solution is to redefine the task of mining HUIs as mining top-k high utility itemsets (top-k HUIs). The idea is to let the users specify k, i.e., the number of desired itemsets, instead of specifying the minimum utility threshold. Mining Top K-HUI patterns introduces an RUC (Raising the threshold by the Utilities of Candidates) strategy. This strategy can be incorporated with any one-phase mining algorithm where itemsets are found with their utilities. It adopts the Top K-List structure to maintain top-k HUIs, where itemsets are sorted by descending order of utility. Initially, Top K-List is empty. When an itemset X is found by the search procedure and its utility is no less than min_util, X is added to Top K-List. If there are more than k itemsets already in Top K-List, min_util can be safely raised to the utility of the k-th itemset in Top K-List. After that, itemsets having a utility lower than the raised min_util are removed from Top K-List.

A. System Architecture

Mining Top K-HUI patterns by raising the threshold consists of the following mining Layers, TKOA and ALG2.

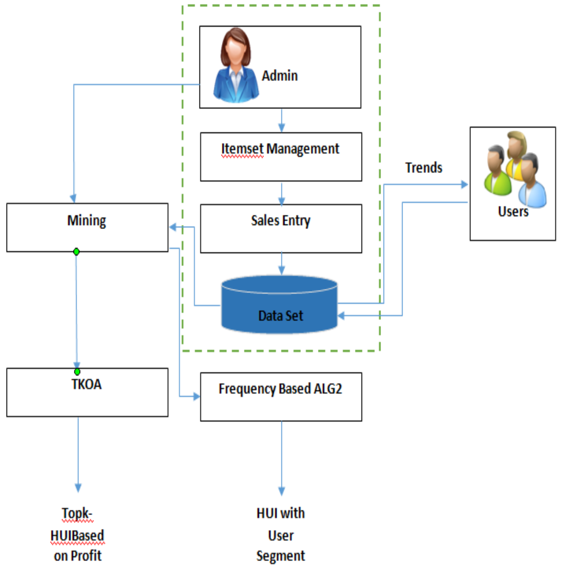

Fig 1: Mining Top K-HUI patterns Architecture

The Architecture starts from Admin who tries to Login by using Username and Password. Basically Admin have different roles in finding Top K-HUI patterns which include the following.

Itemset Management: The Itemset Management consists of an Item Entry. The Item Entry Table includes Name, Category, Brand, Cost Price and Selling Price of the items to be entered. Once a new Item is saved with these details it is listed on the Table with an ID number. Entering of the item is done by the Admin. There is a provision to add images of the item to be saved. Based on the item entered in the table the Transaction is created for Mining Top K-HUI.

Sales Entry: In this Transaction ID, Category, Item Name, Customer Name, cost price is already set. By choosing a customer Name details are entered and saved. Multiple transactions are possible here for the same customers. A new customer account can be created by entering Username and password.

Dataset: After managing Items and Sales the details are stored in a database which can be used to retrieve Top K-HUI patterns through Mining phase. The database helps the customers to view latest Trends based on the items stored.

Mining: The Mining phase takes the details from the database to find Top K-HUI patterns. It includes two Algorithms namely, TKOA which finds Top K-HUI based on profit and ALG2 which finds HUI with user segment.

IV. TKOA

The TKOA (mining Top-k utility itemsets in One phase Algorithm) uses a list-based structure named utility-list to store the utility information of itemsets in the database. It uses vertical data representation techniques to discover top-k HUIs in only one phase. It utilizes the basic search procedure of HUI-Miner and its utility-list structure. Whenever an itemset is generated by TKOA, its utility is calculated by its utility-list without scanning the original database. TKOA includes RUC, RUZ strategies to increase its efficiency.

Algorithm

```
Input:  i) Transaction Set
        ii) item and profit
        iii) K-value
    1.  Compute TWU
    2.  Sort TWU in ascending Order
    3.  Travel in each Transaction
    •   Re-order the transaction items based on TWU in
        increasing order.
    4.  Find Initial Utility List of item with TWU > 0 in
        increasing Order
    5.  Z-element removal
    6.  Find distinct Prefix after Z-element removal – Class
    7.  Travel in each Class
    •   Create Utility List of ith item in Class and travel in
        the utility list
    •   Sum (XUtil) > threshold, then insert the item
    •   If no space, then apply RUC strategy
    •   Sum(XUtil) + Sum(RUtil) > threshold
    •   X= X items + Ritems
    8.  End
```

RUC Strategy

```
Input: K-value, X, Utility
    1.  Check HUI List is full or not
       a) If NO, then add X with Utility
       b) If YES,
    •   Find lowest Utility from HUI-List
    •   Compute it with Utility
    •   If greater, remove lowest utility item from HUI Set
    •   Set threshold = min utility
    •   Add X with Utility
    2.  Return
```

TKOA initially sets the min_util threshold to 0 and initializes a min-heap structure Top K-List for maintain-ing the current top-k HUIs during the search. The algorithm then scans D twice to build the initial utility-lists. Then, TKOA explores the search space of top-k HUI using a procedure that we name Top K-HUI-Search. It is the combination of a novel strategy named RUC (Raising threshold by Utility of Candidates) with the HUI-Miner search procedure [14].

During the search, TKO updates the list of current top-k HUIs in Top K-List and gradually raises the min_util threshold by the information of Top K-List. When the algorithm terminates, the Top K-List captures the complete set of top-k HUIs in the database.

V. ALG2

A new Algorithm named ALG2 is introduced for mining Top K-HUI patterns which completely relies on user frequency. ALG2 is not profit oriented. Here the items are re-ordered in descending order based on the item Frequency. From the Re-Ordered Transaction get all the patterns by pruning. In this Algorithm User Interest and Transaction number is considered. The user can view the latest Trends of the item and give Likes. Finally the Items with highest Count is listed with customer ID.

Algorithm

```
1.  Get all Transactions
2.  Table Traversal to get frequency of item
3.  Apply Threshold based on U-frequency and T-frequency
4.  Reorder items in the transaction by descending order
5.  Travel in Reorder Transaction
      •  Split transaction into items
      •  Set Parent=0
      •  Get first item of splitted transaction
      •  Check the item is contained in child nodes of parent
      •  If yes, Frequency updation is done.
      •  If no, new node created, parent= new node_id
6.  Continue  step 5 until all items in the transaction covered
7.  Pattern Formation
      •  Get all nodes in tree to a set
      •  Travel each node in the set
      •  Check nodes frequency > Threshold , then
      •  Add the item to HUI
8.  Travel each item in HUI
9.  Find Customer _id of each HUI
10.  Append it to HUI set
11. End
```

VI. CONCLUSION AND FUTURE WORK

This paper presents Mining Top K-HUI patterns by using two algorithms namely TKOA which is completely profit oriented and ALG2 which is based on user frequency. The TKOA utilizes the basic search procedure of HUI-Miner and its utility-list structure. Utility is calculated by its utility-list without scanning the original database. In ALG2, based on the count of an item/itemsets HUIs with customer ID are listed.

As a future work the two Algorithm, TKOA (profit based) and ALG2 (user frequency based) can be combined for pruning the search space. Top K-HUI mining task can be extended to discover different types of top-k high utility patterns such as top-k high utility episodes, top-k high utility web access patterns and top-k mobile high utility sequential patterns.

REFERENCES

[1] J. Yin, Z. Zheng and L. Cao, 2012, ‘USpan: An Efficient Algorithm for Mining High Utility Sequential Patterns’, in Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).

[2] Jingyu Shao, Junfu Yin, Wei Liu and Longbing Cao,2015 ‘Mining Actionable Combined Patterns of High Utility and Frequency’, IEEE.

[3] Philippe Fournier-Viger, Cheng-Wei Wu, Souleymane Zida, Vincent S.Tseng, 2014, ‘FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning’, Proc.21st international Symposium on Methodologies for intelligent Systems(ISMIS 2014), Springer, LNAI, pp.83-92.

[4] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE,’ Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases’.

[5] Junqiang Liu, Ke Wang, Benjamin C.M. Fung, 2016, ‘Mining High Utility Patterns in One Phase without Generating Candidates’ IEEE Transactions on Knowledge and Data Engineering.

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

[7] P. Fournier-Viger, C. Wu, V. S. Tseng, ―Mining Top-K Association Rules,‖ in Proc. of Int’l Conf. on Canadian conference on Advances in Artificial Intelligence, pp. 61–73, 2012.

[8] P. Fournier-Viger, V. S Tseng, ―Mining Top-K Sequential Rules,‖ in Proc. of Int’l Conf. on Advanced Data Mining and Applications, pp. 180-194, 2011.

[9] R. Agrawal and R. Srikant, ―Fast Algorithms for Mining Association Rules,‖ in Proc. of Int'l Conf. on Very Large Data Bases, pp. 487-499, 1994.