Dynamic Algorithms for Mining Top-K High Service Component sets

T.Jithendar[1] G.Venkatesh[2], Mesa Kalpana[3]

[1,2] Assistant Professor, Department of Computer Science and Engineering, Guru nanak Institutions technical campus, Hyderbad, T.S,India.

[3] Assistant Professor, Department of Computer Science and Engineering ,Aarushi Group of Institution college of Engineering warangal, TS, 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 by trial and error 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. In this paper, we address the above issues by proposing a new framework for top-k high utility itemset mining, where k is the desired number of HUIs to be mined. Two types of efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One phase) are proposed for mining such itemsets without the need to set min_util. We provide a structural comparison of the two algorithms with discussions on their advantages and limitations. Empirical evaluations on both real and synthetic datasets show that the performance of the proposed algorithms is close to that of the optimal case of state-of-the-art utility mining algorithms.**

I. INTRODUCTION

FREQUENT item set mining (FIM) is a fundamental research topic in data mining. However, the traditional FIM may discover a large amount of frequent but low-value item sets and lose the information on valuable item sets having low selling frequencies. Hence, it cannot satisfy the requirement of users who desire to discover item sets with high utilities such as high profits. To address these issues, utility mining emerges as an important topic in data mining and has received extensive attention in recent years. In utility mining, each item is associated with a utility (e.g. unit profit) 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 [23] and biomedicine [4]. However, efficiently mining HUIs in databases is not an easy task because the downward closure property [1], [8] used in FIM does not hold for the utility of item sets. In other words, pruning search space for HUI mining is difficult because a superset of a low utility item set can be high utility. To tackle this problem, the concept of transaction weighted utilization (TWU) model [13] was introduced to facilitate the performance of the mining task. In this model, an item set is called high transaction-weighted utilization item set (HTWUI) if its TWU is no less than min_ util, where the TWU of an item set represents an upper bound on its utility. Therefore, a HUI must be a HTWUI and all the HUIs must be included in the complete set of HTWUIs. A classical TWU model-based algorithm consists of two phases. In the first phase, called phase I, the complete set of HTWUIs are found. In the second phase, called phase II, all HUIs are obtained by calculating the exact utilities of HTWUIs with one database scan. 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. A large number of HUIs also causes the mining algorithms to become inefficient or even run out of memory, because the more HUIs the algorithms generate, the more resources they consume. 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 _ V.S. Tseng and C.-W. Wu are with the Department of Computer Science, National Chao Tung University, 1001 University Road, Hsinchu City, Taiwan. E-mail: vtseng@cs.nctu.edu.tw, silvemoonfox@hotmail.com. _ P. Fournier-Viger is with the Department of Computer Science, University of Moncton, Moncton, NB, Canada. E-mail: philippe.fournier-viger@umoncton.ca. _ P.S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, and the Institute for Data Science, Tsinghua University, Beijing, China. E-mail: psyu@cs.uic.edu. Manuscript received 25 Nov. 2014; revised 2 July 2015; accepted 7 July 2015. Date of publication 22 July 2015; date of current version 3 Dec. 2015.Recommended for acceptance by X. Xiong. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2015.2458860 over until being satisfied with the results. This process is both inconvenient and time-consuming. To precisely control the output size and discover the item sets 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 item sets (top-k HUIs). The idea is to let the users specify k, i.e., the number of desired item sets, instead of specifying the minimum utility threshold. Setting k is more intuitive than setting the threshold because k represents the number of item sets that the users want to find whereas choosing the threshold depends primarily on database characteristics, which are often unknown to users. Using a parameter k instead of the min_util threshold is very desirable for many applications. For example, to analyze customer purchase behavior, top-k HUI mining serves as a promising solution for users who desire to know "What are the top-k sets of products (i.e., item sets) that contribute the highest profits to the company?" and "How to efficiently find these item sets without setting the min_util threshold?". Although top-k HUI mining is essential to many applications, developing efficient algorithms for mining such patterns
is not an easy task.

II.EXISTING SYSTEM:

- The traditional FIM (Frequent itemset mining) may discover a large amount of frequent but low-value itemsets and lose the information on valuable itemsets having low selling frequencies. Hence, it cannot satisfy the requirement of users who desire to discover itemsets with high utilities such as high profits.
- To address these issues, utility mining emerges as an important topic in data mining and has received extensive attention in recent years. In utility mining, each item is associated with a utility (e.g. unit profit) and an occurrence count in each transaction (e.g. quantity).
- The utility of an itemset represents its importance, which can be measured in terms of weight, value, quantity or other information depending on the user specification. An itemset is called high utility itemset (HUI) if its utility is no less than a user-specified minimum utility threshold min_util.
- In recent years, high utility itemset mining has received lots of attention and many efficient algorithms have been proposed, such as Two-Phase, IHUP, IIDS, UPGrowth, d2HUP and HUI-Miner. These algorithms can be generally categorized into two types: twophase and one-phase algorithms.

DISADVANTAGES OF EXISTING SYSTEM:

- Although many studies have been devoted to HUI mining, it is difficult for users to choose an appropriate minimum utility threshold in practice.
- The existing studies may perform well in some applications, they are not developed for top-k

high utility itemset mining and still suffer from the subtle problem of setting appropriate thresholds.

III.PROPOSED SYSTEM:

In this paper, we address all of the above challenges by proposing a novel framework for top-k high utility itemset mining, where k is the desired number of HUIs to be mined.

Major contributions of this work are summarized as follows:

- First, two efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets 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 itemsets. TKU inherits useful properties from the TWU model and consists of two phases.
- In phase I, potential top-k high utility itemsets (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 itemsets in the database.
- It uses vertical data representation techniques to discover top-k HUIs in only one phase.

ADVANTAGES OF PROPOSED SYSTEM:

- Two efficient algorithms TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One phase) are proposed for mining such itemsets without setting minimum utility thresholds.
- TKO is the first one-phase algorithm developed for top-k HUI mining, which integrates the novel strategies RUC, RUZ and EPB to greatly improve its performance.
- Empirical evaluations on different types of real and synthetic datasets show that the proposed algorithms have good scalability on large datasets and the performance of the proposed algorithms is close to the optimal case of the state-of-the art two-phase and one-phase utility mining algorithms.

IV.IMPLEMENTATION

- Admin

In this module, the Admin has to login by using valid user name and password. After login successful he can perform some operations such as view and authorize users, Adding Categories Sub-Categories, Adding Product Posts for by Selecting Category and Sub-Categories, Viewing Top- K Utility Item Set Keywords, Viewing all Products in terms of Construction of UP-Tree, Viewing all High Utility Item set Mining Products, Viewing All User Search History and Finding Top K Products Results in Chart.

Viewing and Authorizing Users:

In this module, the admin views all users details and authorize them for login permission. User Details such as User Name, Address, Email Id and Mobile Number.

Add Categories, Sub-Categories and Product Posts:

In this module, the admin adds Categories, Sub-Categories and Product Posts. The Product Posts are added by selecting particular category and Sub-Category and Product Details such as, Product Title, Price, Description and Image of that Product.

View all Products with Ranks and Comments:

In this module, the admin can see all the uploaded products with product ranks and comments. The Product details contain Product title, description, price, and image.

The Comment details include commented user, their comment and the date of comment.

View Top-K Utility Item Sets Keywords:

In this module, the all keywords which are all used very frequently and less frequently will be displayed in a Rank (No. of times used) in a Top-K Order.

View all Products in terms of Construction of UP-Tree:

In this, the admin can see all the products in a Tree Format. In this Tree, Firstly (On Top) Category then

Sub-Category and lastly (at Bottom) Product Posts will be displayed.

View all high Utility Item Set Mining Products:
In this, the top 5 Mining products will be displayed along with their details based on ranks. The Product details contain Product title, description, price, and image.

Find Top K Products Results in Chart:
In this, the top K number of products will be displayed based on top rank of products in a chart based on the value selected from the combo box.

- User

In this module, there are n numbers of users are present. User should register before performing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user can perform some operations like viewing their profile details, searching for products based on product description, searching products and viewing them in a UP-Tree Format, Viewing Own Search History and Finding Top K Product Item Sets by selecting category and Top K Value.

Viewing Profile Details:
In this module, the user can see their own profile details, such as their address, email, mobile number, profile Image.

Search Products:
In this, the user search for products based on product description. The matched results will be displayed in two ways: Exact Matched and Related Products. Related Products are the products which are not exactly matched for user entered keyword and they are belong to the same category of exactly matched products category.

Search and View Products in UP-Tree Format:
In this, the user search for products based on product description and the matched products will display in a UP-Tree Format. In a Tree there would be three layers. In a first top layer the Category name and in a second layer the Sub-Category Name and in a last layer the Product Title would be shown and user can see the product details by clicking on product name.

Finding Top K Item Sets:
In this, the user finds Product Items Sets based on Category and Top k Value. The Result is the top K number of products from the Selected Category.

V.CONCLUSION

In this paper, we have studied the problem of top-k high utility item sets mining, where k is the desired number of high utility item sets to be mined. Two efficient algorithms TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in One phase) are proposed for mining such item sets without setting minimum utility thresholds. TKU is the first two-phase algorithm for mining top-k high utility item sets, which incorporates five strategies PE, NU, MD, MC and SE to effectively raise the border minimum utility thresholds and further prune the search space. On the other hand, TKO is the first one-phase algorithm developed for top-k HUI mining, which integrates the novel strategies RUC, RUZ and EPB to greatly improve its performance. Empirical evaluations on different types of real and synthetic datasets show that the proposed algorithms have good scalability on large datasets and the performance of the proposed algorithms is close to the optimal case of the state-of-theart two-phase and one-phase utility mining algorithms [14], [25]. Although we have proposed a new framework for top-k HUI mining, it has not yet been incorporated with other utility mining tasks to discover different types of top-k high utility patterns such as top-k high utility episodes, top-k closed high utility item sets, top-k high utility web access patterns and top-k mobile high utility sequential patterns. These leave wide rooms for exploration as future work.

REFERENCE

[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, pp. 1708–1721, Dec. 2009.

[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 item sets mining algorithm," in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.

[14] M. Liu and J. Qu, "Mining high utility item sets 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 item sets 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.

[20] T. Quang, S. Oyanagi, and K. Yamazaki, "ExMiner: An efficient algorithm for mining top-k frequent patterns," in Proc. Int. Conf. Adv. Data Mining Appl., 2006, pp. 436 – 447.

[21] H. Ryang and U. Yun, "Top-k high utility pattern mining with effective threshold raising Strategies," Knowl.-Based Syst., vol. 76, pp. 109–126, 2015.

[22] H. Ryang, U. Yun, and K. Ryu, "Discovering high utility itemsets with multiple minimum supports," Intell.Data Anal., vol. 18, no. 6, pp. 1027–1047, 2014.

[23] B. Shie, H. Hsiao, V. S. Tseng, and P. S. Yu, "Mining high utility mobile sequential patterns in mobile commerce environments," in Proc. Int. Conf. Database Syst. Adv. Appl. Lecture Notes Comput. Sci., 2011, vol. 6587, pp. 224–238.

[24] P. Tzvetkov, X. Yan, and J. Han, "TSP: Mining top-k closed sequential patterns," Knowl. Inf. Syst., vol. 7, no. 4, pp. 438–457, 2005.

[25] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253–262.