ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

Efficient Algorithms for Mining High Utility item sets from Transactional Databases

Sharayu Fukey, P. M. Chawan

PG Student, Department of Computer Engineering, VJTI, Mumbai, India

Associate Professor, Department of Computer Engineering, VJTI, Mumbai, India

ABSTRACT: Frequent Item sets discovery of sets of items that are frequently purchased together by customer. High Utility Item sets (HUIs) refers to discovery of an item set with high utility such as streaming analysis, market analysis, mobile computing and biomedicine. HUIs consider with particular attribute of the item set. HUIs mining is defined as qualitative representation of user preferences. The primary goal of mining is to discover hidden patterns and unexpected trends in data set with HUI and the secondary goal is to discover Top-k results. Point of Item sets (POI) is to provide personalized recommendation of particular aspects such as restaurants, movie theaters. Recommendation is depends upon user previous interest or previous search options.HUIs mining identifies item sets where its attribute satisfies threshold. According to the threshold, items categorized as High Utility Item sets and Low Utility item sets. Resulting candidate item sets further undergoes pruning techniques so as to reduce no. of candidate item set considering. Efficient framework can be designed for mining Top-k HUIs set where k is no. of HUIs to be mined and defining the recommendation and POI. HUI mining is used to find Top-k results to enhance decision making process. HUIs mining is useful for building powerful search engines to display Top-k results. IT has many application like market analysis, stream analysis and biometric.

KEYWORDS: Utility mining, high utility item sets, high utility item set mining, top-k, pattern mining, parallel mining, top-k high utility item set mining, Data mining ,frequent item set, transactional database.

I. INTRODUCTION

A search engine is an information retrieval system designed to help find information stored on a computer system. The search results are usually presented in a list and are commonly called hits. Search engines help to minimize the time required to find information and the amount of information which must be consulted. Search engines provide an interface to a group of items that enables users to specify criteria about an item of interest and have the engine find the matching items. The criteria are referred to as a search query. In the case of text search engines, the search query is typically expressed as a set of words that identify the desired concept that one or more documents may contain. There are several styles of search query syntax that vary in strictness. Data mining is use to find hits according to some conditions. Data mining is concerned with analysis of large volumes of data to automatically discover interesting regularities or relationships which in turn leads to better understanding of the underlying processes. The primary goal is to discover hidden patterns, unexpected trends in the data. 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 is 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 and biomedicine. 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 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 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 over until being satisfied with the results. This process is both inconvenient and time-consuming.

Search engine is information retrieval system in which the search results are usually presented in a list. Search engine is considering the data mining to find out the appropriate information. In existing application or web page it takes the frequent item sets mining which is depend upon frequency of the item sets. Frequent item set mining does not consider attribute. For consideration of attribute uses HUI mining. Introducing a shopping web page consist of search engine which gives top-k high utility item sets.

In this k no of item sets is not fixed. Following are contribution in shopping web page:

- High Utility Item sets: HUI generated by considering particular attribute. For shopping web page defining attribute in terms of ratio. Ratio of hit, like and buy. Utility of the item set is decided by hit:like:buy ratio. Secondly considering attribute as a prize.
- Top-k results: providing k no of results to the user. K value is not defined.
- Recommendation: in shopping application Recommendation is based on user history but which is store in browsers cookie. If those cookies are deleted then recommendation will not get generate. Recommendation is depends upon user previous interest or previous search options.
- Point of interest (POI): Point of Item sets (POI) is to provide personalized recommendation of particular aspects such as restaurants, movie theaters. According to user interest suggesting some more products which are related to it.

II. REVIEW OF LITERATURE

Vincent S. Tseng, Senior Member, IEEE, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, Fellow, “Efficient Algorithms for Mining Top-K High Utility Itemsets” explains, High utility item sets (HUIs) mining is an emerging topic in data mining, which refers to discovering all item sets 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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

high, it is likely that no HUIs will be found. In this paper, address the above issues by proposing a new framework for top-k high utility item set mining, where k is the desired number of HUIs to be mined. Two types of efficient algorithms named 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 the need to set min_util.

Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, "Efficient Algorithms for Mining High Utility Item sets from Transactional Databases" explains, 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 algorithms 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 situation may become worse when the database contains lots of long transactions or long high utility item sets. This paper propose two algorithms, namely utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility item sets with a set of effective strategies for pruning candidate item sets. The information of high utility item sets is maintained in a tree-based data structure named utility pattern tree (UP-Tree) such that candidate item sets can be generated efficiently with only two scans of database. The performance of UP-Growth and UP-Growth+ is compared with the state-of-the-art algorithms on many types of both real and synthetic data sets. Experimental results show that the proposed algorithms, especially UP Growth+, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime, especially when databases contain lots of long transactions.

R.Nandhini, Dr.N.Suguna "Shrewd Technique for Mining High Utility Item set via TKU and TKO Algorithm" explains, High utility item sets (HUIs) mining is an emanate topic in data mining, which refers to discovering all item sets having a utility meeting a user-specified minimum utility threshold. We used three efficient algorithms AppriorCH(Apriori-based algorithm for mining High utility closeditem set), Apriori HC-D (AprioriHC algorithm with removing unpromising and isolated items) and CHUD (Closed High Utility item set Discovery) algorithm. However, setting min_util appropriately is a hard problem for users. Finding an appropriate minimum utility threshold by trial and error is a tough process for users. Setting very low min_util, HUIs will be generated in large, which may cause the mining process to be very inefficient. In another way, by setting too high min_util, no HUI will be found. So the above issues is addressed by proposing a framework for mining top-k high utility item set , where k is the desired number of HUIs to be mined. Two types of proficient algorithms named 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 the need to set min_util.

T.Vinothini, V.V.Ramya Shree "Mining High Utility Item sets from Large Transactions using Efficient Tree Structure" International explains, mining high utility item sets from a transactional database refers to the discovery of item sets with high utility like profits. It is an extension of the frequent pattern mining. Although a number of relevant algorithms 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 situation may become worse when the database contains lots of long transactions or long high utility item sets. Association rule mining with item set frequencies are used to extract item set relationships. Frequent pattern mining algorithms are designed to find commonly occurring sets in databases. Memory and run time requirements are very high in frequent pattern mining algorithms. Systolic tree structure is a reconfigurable architecture used for utility pattern mining operations. High throughput and faster execution are the highlights of the systolic tree based reconfigurable architecture. The systolic tree mechanism is used in the frequent pattern extraction process for the transaction database. Systolic tree based rule mining scheme is enhanced for Weighted Association Rule Mining (WARM) process, used to fetch the frequently accessed transaction database with its weight values. The dynamic dataset is weight assignment scheme uses the item request count and span time values. The proposed system improves the weight estimation process with span time, request count and access sequence details.

J. Yin, Z. Zheng, L. Cao, Y. Song, and W. Wei, "Mining top-k high utility sequential patterns," explains a novel framework called top-k high utility sequential patternmining to tackle this critical problem. Accordingly, an efficient

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

algorithm, Top-k high Utility Sequence (TUS for short) mining, is designed to identify top-k high utility sequential patterns without minimum utility. In addition, three effective features are introduced to handle the efficiency problem, including two strategies for raising the threshold and one pruning forfiltering unpromising items. Our experiments are conducted on both synthetic and real datasets.

III. EXISTING SYSTEM APPROCH

Frequent item set mining is a fundamental research topic in data mining (FIM) 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_utility. HUI mining is essential to many applications such as streaming analysis, market analysis, mobile computing and biomedicine.

We have prepared some proposed algorithms in related work. But all these algorithms obtain the problem of generating a large number of candidate item sets. Such a large number of candidate item sets downgrades the mining performance in terms of execution time and space. If algorithm produces huge number of candidate item sets, then higher processing time it consumes. Utility pattern growth (UP-Growth) and UP-Growth+ algorithm conquer this limitation. These algorithms found high utility item sets by using adequate strategies. The information of high utility item sets is managed in a tree-based data structure named utility pattern tree (UP-Tree) such that candidate item sets can be produced efficiently with only two scans of database. 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 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.

IV. PROPOSED SYSTEM ARCHITECTURE

UP-growth and UP-Growth+ algorithm discovering high utility item sets is efficiently. By applying the proposed strategies of these algorithms the number of generated candidate item sets can be highly decreased in phase I and high utility item sets can be identified more efficiently in phase II. This technique is useful on static datasets. It did not consider the adaption of database. Our proposed system will perform on transactional database which is related to data mining. i.e. deletion or insertion of one or more records from database will consider on database history. To achieve this it uses the existing approach. Proposed system can avert unnecessary or repetition of calculations by using previous results when a database is updated, or when the threshold value is replaced. UP-Growth Algorithm: The UP-Growth is one of the productive algorithms to generate high utility item sets depending on construction of a global UP Tree. In this system using UP Growth+ maintain the item set dataset by creating UP tree. TKU and TKO find out high utility item sets by both pattern mining and parallel mining. At first stage TKU is applied on the item set database with both parallel and pattern mining. In second step TKO is applied on the item set database with both parallel and pattern mining. The efficient Top-k high utility item sets are generated. Two efficient algorithms named TKU (mining Top-K Utility item sets) 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 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

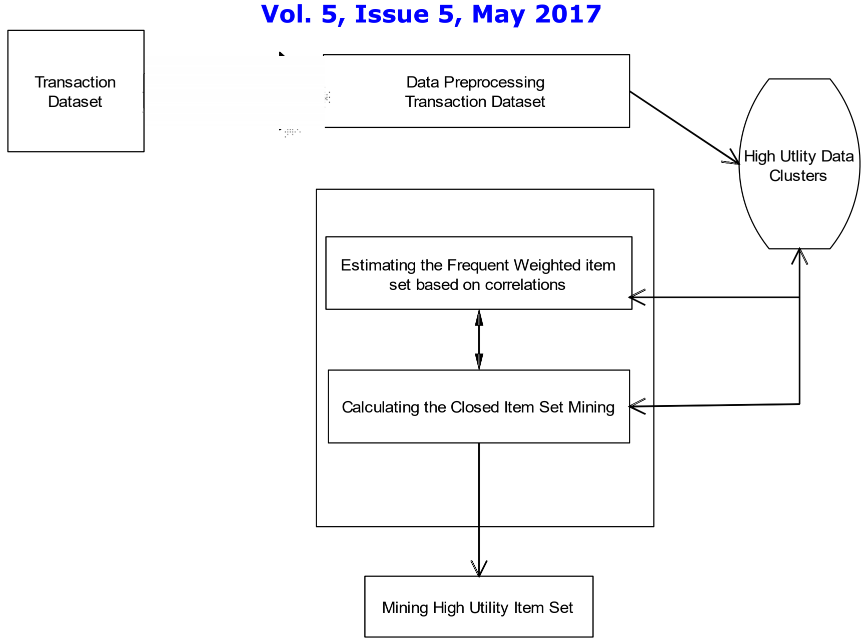


Fig. Proposed System Architecture

V. SYSTEM ANALYSIS

High Utility Item sets mining from transactional database constructed in 4 modules.

Module 1: Administrator
The administrator preserve database of the transactions made by customers. In the daily market basis, each day a new product is let go, so that the administrator would add the product or items, and update the new product view the stock details.

Module 2: Customer
Customer can purchase the number of items. All the purchased items history is stored in the transaction database

Module 3: Construction of UP-Tree [1]
1. First scan:-
a) Initially Transaction Utility (TU) of each transaction is counted. Then TWU of each single item is also assembled.
b) Removing global unpromising items.
c) Utilities of unpromising items are excluded from the TU of the transaction.
d) Then remaining promising items in the transaction are arranged according to the descending order of TWU.
2. Second scan:-
a) UP-Tree is generated by inserting transactions.
The function is performed as follows:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

1. If N has a child node ChN such that ChN.item= Ij, then increment ChN.count by 1; Otherwise, create a new child node ChN with ChN.item= Ij, ChN.count=1, ChN.parent=N and ChN.nu=0.
2. Increase ChN.nu.
3. Call Insert_Reorgnized_Transaction.

Module 4: UP-Growth+ Algorithm [1]

Applying UP-Tree to the UP-Growth takes further execution time for Phase II. A modified algorithm i.e. UP-Growth+ curtail the execution time by effectively identifying high utility item sets. It measures the Maximum transaction Weighted Utilization (MTWU) from all items and considering multiple of min-sup as a user specified threshold value.

Let S be the system that describes dataset i.e. set of transaction with profit of item as input to system with calculation of transaction utility, transaction weighted utility, recognized transaction utility, up tree construction, UP growth algorithm, Improved UP growth algorithm and this all gives output as high potential utility item sets.

Finite set of item sets I={I1, I2, . . . Im}

Transaction Database D={T1, T2,….Tm} where each T is subset of I

Item sets X={I1, I2, ….Il}

Transaction Utility TU(Tr)=EU(Tr, Tr) where EU external utility

Total Utility= ∑TU(Tr)

High utility item set: An item set X is called high utility item set (HUI) if U(X) is no less than a user-specified minimum utility threshold min_util otherwise X is a low utility item set.

High utility item set mining: The goal of HUI mining is to discover fHUI(D, d).

Transaction-weighted utilization: TWU(X)=∑TU(Tr).

High TWU item set: An item set X is a high TWU item set iff TWU(X) _ abs_min_util.

Top-k high utility item set: An item set X is called top-k high utility itemset (top-k HUI) in D iff there are less than k item sets whose utilities are larger than EU(X) in fHUI(D, 0).

Optimal minimum utility threshold: An absolute minimum utility threshold d_ is called optimal minimum utility threshold iff there does not exist a threshold d>d*such that |fHUI(D,0)|= |KH| . An equivalent definition is that d*= min{U(X)| X€KH}.

Module 5: TKU and TKO algorithm

Top K Utility Item sets (TKU) and mining Top K Utility Item sets in One phase (TKO) takes High utility mining on item set database. For mining it will consider both parallel and pattern mining.

Steps for TKU:

1. Construct up tree.
2. Generating PKHU from UP-Tree.
3. Identify Top k HUIs from PKHUIs.

Potential top-k high utility item set: An item set is called Potential top-K High Utility Item set (PKHUI) if its estimated utility value (i.e., TWU) and MAU are no less than the min_utilBorder threshold.

TKHUI Algorithm :

Steps of TKHUI

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

Procedure Mining
Input: All HUI tree T_s and header tables Hs in the current window, an item set based item set (base –item set is initialized as null), as list TKValueList , minimum utility value minUti.
Output :TKHUIs
Begin
1. Find top-k maximal total utility value of itemset in Hs to TKValueList.
2. Add a field add-information to each leaf-node

(1) For each item Q in HL do from the last item of HL and HL is one HS
//Step 1:Calculate utility information of the node Q
(2) Float twu=0,BU=0,SU=0,NU=0;
(3) For each header table H in Hs do
(4) For each node N for the item Q in the tree T corresponding to H do
(5) BU+=T.N.bu;
(6) SU+=T.N.su;
(7) NU+=T.N.nu;// N.nu is a utility for item Q in the list N.piu
(8) End For
(9) End For
(10) Twu=BU+SU
//Step 2 :Generate new itemset and create new sub tree and header table
(13)If(twu>= minUti) then
(14) base-itemset={Q};
(15) create a sub HUI tree subT and a header table subH for base-itemset ;
(16) sub-Mining(SubT,SubT,base-itemset,TKValueList,minUti);
(17)Remove item Q from itemset base-itemset;
(18)End If
// Step 3:Pass add-information on node Q to parent node
(19) Move each node's bac-info to its parent;
(20) End For
(21) Delete item set whose value are less than minUti from TKHUIs;
(22) Return TKHULS:
END

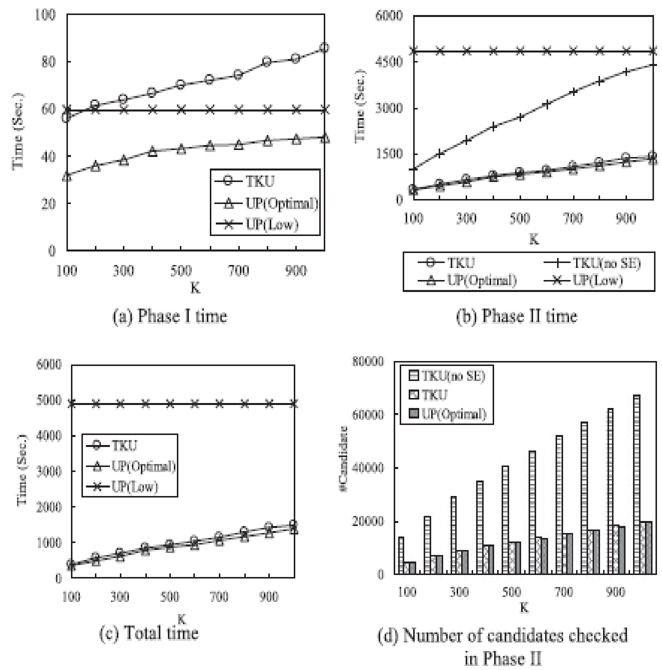
VII. EXPERIMENTAL SET UP

According to Vincent S. Tseng, Senior Member, IEEE, Cheng-Wei Wu, Philippe Fournier Viger, and Philip S. Yu, Fellow, "Efficient Algorithms for Mining Top-K High Utility Item sets", Fig. shows the performance of the algorithms on a very large dataset Chainstore. Because the runtime of TKUBase on this dataset is very slow (e.g., over 24 hours when k=1), we instead use UP-Growth with a low minimum utility threshold (0.01 percent) as the baseline (denoted as UP (Low) in the experiments). The Fig. shows the runtime of the algorithms for phase I. Since the threshold of UP (Low) is fixed, its runtime remains the same when k is varied. In Fig. a, shows TKU performs more operations to apply strategies to raise the threshold step by step. Fig. b shows the runtime for Phase II of the algorithms. In Fig. b, TKU is slightly slower than UP(Low). Fig. c shows the total runtime of the algorithms. Globally, TKU is much faster than UP(Low). This is because UP(Low) needs to check all candidates in Phase II, whereas TKU only needs to check some of them thanks to strategy SE. Fig. d shows the number of candidates checked by the algorithms in Phase II. Since checks all candidates, its performance is the worst. TKU

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

generates much more candidates in phase I, the number of candidates that need to be checked by TKU is close to that of UP(Optimal) in Phase II. This is because TKU avoids checking some candidates by using the SE strategy.



(a) Phase I time
(b) Phase II time
(c) Total time
(d) Number of candidates checked in Phase II

Expected result is to get HUIs from the key word feed in the search engine. HUIs are generated according to the ratio of hit:view:buy attributes. Secondly it will consider according to prize. Recommendation will be generated according to the user previous access and history. POI will be generated according to the particular item selected by the user and it will show the item sets regarding that particular item.

VII.CONCLUSION

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. TKU 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 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-threat two-phase and one-phase utility mining algorithms.

REFERENCES

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases" IEEE Transaction on Know and Data Eng, Vol. 25, No. 8, AUGUST 2013.

[2] R.Nandhini, Dr.N.Suguna "Shrewd Technique for Mining High Utility Item set via TKU and TKO Algorithm" International Journal of Computer Science and Information Technologies 2015.Zhou Zhao, Deng Cai, Xiaofei He and YuetingZhuang,"User Preference Learning for Online Social Recommendation",DOI 10.1109/ TKDE. 2016.2569096, IEEE Trans., 2016.

[3] T.Vinothini, V.V.Ramya Shree "Mining High Utility Item sets from Large Transactions using Efficient Tree Structure" International Journal of Last Tread in Engineering and Technology (IJLTET)

[4] J. Yin, Z. Zheng, L. Cao, Y. Song, and W. Wei, "Mining top-k high utility sequential patterns," in Proc. IEEE Int. Conf. Data Mining, 2013, pp. 1259–1264.

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

[6] 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. 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.

[7] 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.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.

[8]Hongzhi Yin, Xiaofang Zhou, Bin Cui, Hao Wang, Kai Zheng and Quoc Viet Hung Nguyen," Adapting to User Interest Drift for POIRecommendation",IEEE Trans,2016.

[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. Krishnamurthy, "Pruning strategies for mining high utility item sets," 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 item sets 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 item sets 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 item sets," Data Knowl. Eng., vol. 64, no. 1, pp. 198–217, 2008.

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

[19] 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.