Mining High Utility Itemsets using TKO and TKU to find Top-k High Utility Web Access Patterns

Ms. Sharda Khode, Dr. Sudhir Mohod
BDCOE, Sevagram RTMNU Nagpur, BDCOE, Sevagram RTMNU Nagpur, BDCOE
Sharda.khode16@gmail.com, sudhir_mohod@rediffmail.com

Abstract - Mining high utility itemsets from a transactional database refers to the discovery of itemsets 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 itemsets for high utility itemsets. Such a large number of candidate itemsets 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 itemsets. An emerging topic in the field of data mining is utility mining which not only considers the frequency of the itemsets but also considers the utility associated with the itemsets. The main objective of High Utility Itemset Mining is to identify itemsets 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 itemsets with high profit. In this paper we present the implementation of first module where pre-processing of dataset is done to remove unpromising data from web usage and product base dataset by using TopKRules and also 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. Two types of efficient algorithms named TKU and TKO are proposed for mining such itemsets. In this paper we present a literature review of the present state of research and the various algorithms for high utility itemset mining.

Keywords- High utility itemset mining, Web Access Patterns, TKO, TKU

I. INTRODUCTION

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional and relational data sets. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationship among huge amounts of business transaction records can help in many decision making processes such as catalogue design, cross marketing, and customer shopping behaviour analysis. The common framework used for these algorithms is to use min_support threshold [7] to ensure the generation of the correct and complete set of frequent itemsets. 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 itemsets 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 itemset mining [9] has been proposed. Top-k FIM mines the most frequent k itemsets without using the minimum support value from the user. The research of the FIM has been developed into the weighted frequent pattern mining [12] and progressed to the high utility itemset mining (HUIM) [14, 15]. 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. HUI mining is essential to many applications such as streaming analysis [20], market analysis [23], mobile computing [19] and biomedicine [18].

Although top-k HUI mining is essential to many applications, developing efficient algorithms for mining such patterns is not an easy task. It poses four major challenges as discussed below.

First, the utility of itemsets is neither monotone nor antimonotone [16, 17]. In other words, the utility of an itemset may be equal to, higher or lower than that of its supersets and subsets. Therefore, many techniques developed in top-k frequent pattern mining that rely on anti-monotonicity to prune the search space cannot be directly applied to top-k high utility itemset mining.

The second challenge is how to incorporate the concept of top-k pattern mining with the TWU model. Although the TWU model is widely used in utility mining, it is difficult to adapt this model to top-k HUI mining because the exact utilities of itemsets are unknown in phase I. When a HTWUI is generated in phase I, it cannot guarantee that its utility is higher than other HTWUIs and that it is a top-k HUI before performing phase II. To guarantee that all the top-k HUIs can be captured in the set of HTWUIs, a naive approach is to run the algorithm with min_util. However, this approach may face the problem of a very large search space.

The third challenge is that the min_util threshold is not given in advance in top-k HUI mining. In traditional HUI mining, the search space can be efficiently pruned by the algorithms by using a given min_util threshold. If an algorithm cannot raise the min_utilBorder threshold effectively and efficiently, it would produce too many intermediate low utility itemsets during the mining process, which may degrade its performance in terms of execution time and memory usage. Thus the challenge is to design effective strategies that can raise the min_util threshold as high as possible and as quickly as possible, and further reduce as much as possible the number of candidates and intermediate low utility itemsets produced in the mining process.

The last challenge is how to effectively raise the min_utilBorder threshold without missing any top-k HUIs. A good algorithm is one that can effectively raise the threshold during the mining process. However, if an incorrect method for raising the threshold is used, it may result in some top-k HUIs being pruned. Thus, how to raise the threshold efficiently and effectively without missing any top-k HUI is a crucial challenge for this work.

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

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 [19] to maintain the information of transactions and utilities of itemsets. TKU inherits useful properties from the TWU model and consists of two phases.

II. LITERATURE REVIEW

Title: Mining association rules between sets of items in large databases
Authers: R. Agrawal and R. Srikant, T. Imielinski
This paper [1] proposed apriori algorithm, it is used to obtain frequent itemsets from the database. In miming the association rules this auther have the problem to generate all association rules that have support and confidence greater than the user specified minimum support and 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 itemsets. Candidate itemsets are stored in a hash-tree. The hash-tree node contains either a list of itemsets or a hash table. Apriori is a classic algorithm for frequent itemset mining and association rule learning over transactional databases. After identifying the large itemsets, only those itemsets 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.

Title: Efficient Data Mining for Path Traversal Patterns
Authors: J. Han, J. Wang, Y. Lu, and P. Tzvetkov
This paper [10] proposed frequent pattern tree (FP-tree) structure, an extended prefix tree structure for storing crucial information about frequent patterns, compressed and develop an efficient FP-tree based mining method is Frequent pattern tree structure. Pattern fragment growth mines the complete set of frequent patterns using the FPgrowth. It constructs a highly compact FP-tree, which is usually

substantially smaller than the original database, by which costly database scans are saved in the subsequent mining processes. It applies a pattern growth method which avoids costly candidate generation. FP-growth is not able to find high utility itemsets.

Title: A fast high utility itemsets mining algorithm
Authors: Y. Liu, W. Liao, and A. Choudhary
In this paper [13] two-phase algorithm for finding high utility itemsets is proposed. The utility mining is to identify high utility itemsets that drive a large portion of the total utility. Utility mining is to find all the itemsets whose utility values are beyond a user specified threshold. Two-Phase algorithm, it efficiently prunes down the number of candidates and obtains the complete set of high utility itemsets. This paper explain transaction weighted utilization in Phase I, only the combinations of high transaction weighted utilization itemsets are added into the candidate set at each level during the level-wise search. In phase II, only one extra database scan is performed to filter the overestimated itemsets. Two-phase requires fewer database scans, less memory space and less computational cost. It performs very efficiently in terms of speed and memory cost both on synthetic and real databases, even on large databases. In Two-phase, it is just only focused on traditional databases and is not suited for data treams. Two-phase was not proposed for finding temporal high utility itemsets in data streams. However, this must rescan the whole database when added new transactions from data streams. It need more times on processing I/O and CPU cost for finding high utility itemsets.

Title: A fast agorithm for mining high utility itemsets
Author: S.Shankar, T.P.Purusothoman, S. Jayanthi, N.Babu
This paper [11] proposed a novel algorithm Fast Utility Mining (FUM) which finds all high utility itemsets within the given utility constraint threshold. To generate different types of itemsets the authors also suggest a technique such as Low Utility and High Frequency (LUHF) and Low Utility and Low Frequency (LULF), High Utility and High Frequency (HUHF), High Utility and Low Frequency (HULF).

Title: Top-K high utility itemset mining based on Utility List Structures
Authors: Serin Lee, Jong Soo Park
In This paper [4] a new algorithm, TKUL-Miner, to mine top-k high utility itemsets efficiently proposed. It utilizes a new utility-list structure which stores necessary information at each node on the search tree for mining the itemsets. The proposed algorithm has a strategy using search order for specific region to raise the border minimum utility threshold rapidly. Moreover, two additional strategies for calculating smaller overestimated utilities are suggested to prune unpromising itemsets effectively.

Title: Efficient algorithms for mining Top-K high utility itemsets
Authors: Vincent S. Tseng, Cheng-wei Wu, Philippe Fournier-Viger and Philip S. Yu
In this paper [2] a novel framework for top-k high utility itemset mining, where k is the desired number of HUIs to be mined is proposed. 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.

Title: Utility Pattern Mining Algorithm bases on Improved Utility Pattern Tree
Authors: Shuning Xing, Fangai Liu, Jiwei Wang, Lin Pang, Zhenguo Xu
In this paper [5] a process of UP-Tree by introducing a Fast Utility Tree (FU-Tree) is proposed. In this method, they introduce the LinkQueue to reduce the number of scanning the original database and adopt prefix utility to minimize the overestimated utility. The theoretical analyses and experimental results show that FU-Tree outperforms UP-Tree in the time consumption of construction trees, and enhances the efficiency of mining high utility itemsets.

Title: Mining High Utility Patterns in One Phase without Generating Candidates
Authors: Junqiang Liu, Benjamin C.M. Fung
In this paper [8] a novel algorithm that finds high utility patterns in a single phase without generating candidates is proposed. The novelties lie in a high utility pattern growth approach, a lookahead strategy, and a linear data structure. Concretely, our pattern growth approach is to search a reverse set enumeration tree and to prune search space by utility upper bounding. This algorithm also look ahead to identify high utility patterns without enumeration by a closure property and a singleton property. Also linear data structure enables this to compute a tight bound for powerful pruning and to directly identify high utility patterns in an efficient and scalable way, which targets the root cause with prior algorithms.

III. MOTIVATION

The basic motivation is to design effective strategies that can raise the min-util internally as high as possible and as quickly as possible and further reduce the number of candidates and

intermediate low utility item-sets. It is helpful to know 'what are the top-k sets of products that contribute the highest profits to the company' and to analyse customer purchase behaviour, and to discover the item-sets with highest utilities without setting the thresholds and to precisely control the output size.

IV. PROBLEM DEFINITION

High utility itemsets (HUIs) mining 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. 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 will 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. For mining such itemsets we will use two efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in one phase) without setting minimum utility threshold. We will apply this approach to find new utility mining task to discover different types of top-k high utility web access patterns.

V. OBJECTIVE

A. To achieve good scalability under different parameters like time and database size.

B. To reduce search space, memory consumption and multiple database scan.

C. To discover different types of top-k high utility web access patterns.

VI. PROPOSED WORK

Calculating and displaying the top-k high utility itemsets requires complete chain process. This process is depicted in the architecture diagram. As per this architecture first we calculate the Transactional Utility and Transactional Weighted Utility by scanning the Transactional Database. This scan will be the first database scan.

Next step is to mine the minimum utility threshold. This is the most important step in utility itemset mining. The minimum utility threshold can be any value regardless of dataset. If the minimum threshold is very less, then very large amount of irrelevant data will come or if it is very high, then very less amount of data will be retrieved. Therefore, a new approach will be followed which will calculate the minimum utility threshold on the basis of the database.

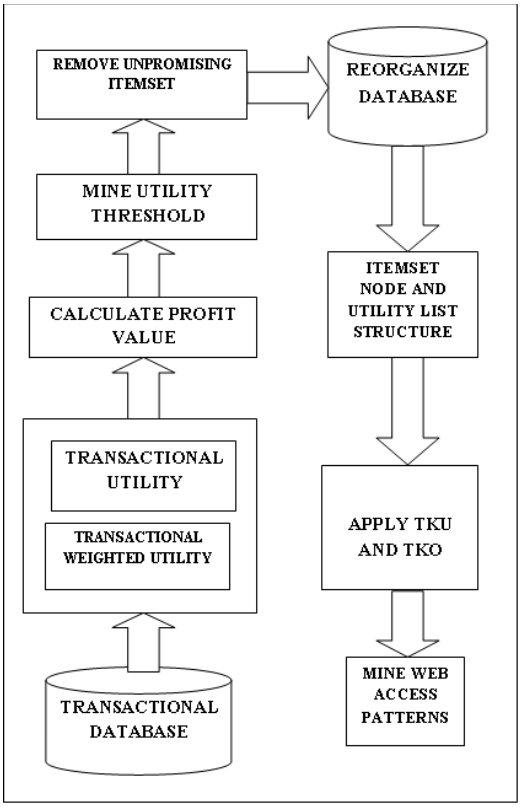
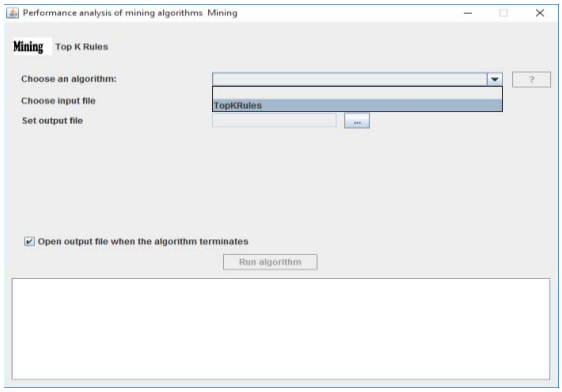


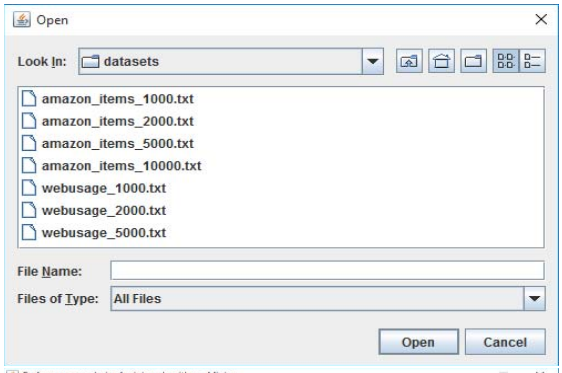
Figure: Architecture Diagram

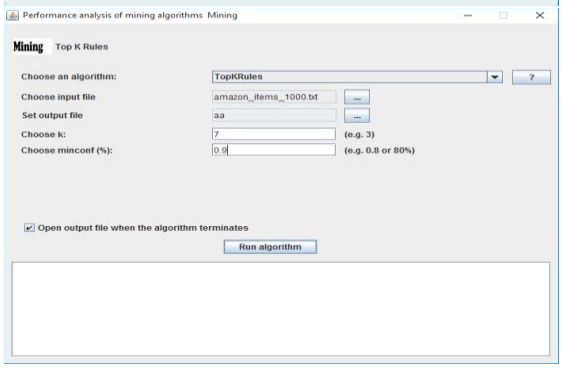
Now the unpromising itemsets will be removed because they won't generate any profit. Then, the Database will be reorganized in the order of the profit values. This will be the second database scan. This will show the most promising itemsets above the less promising itemsets.

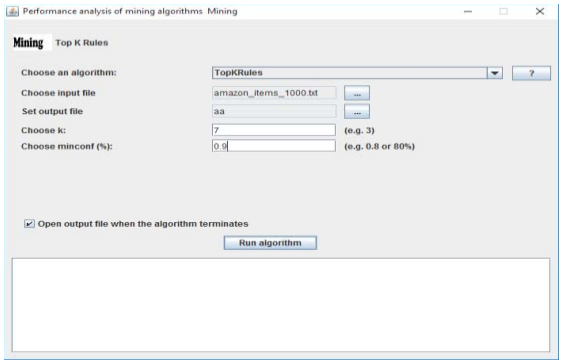
After reorganizing the database which shows the most promising itemsets a Itemset Node and Utility-List Structure is created, then this structure is mined the itemsets using TKUL algorithm, after that TKU and TKO algorithms will be proposed for mining such itemsets. The TKU algorithm adopts a compact 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. Then the web access patterns are mined.

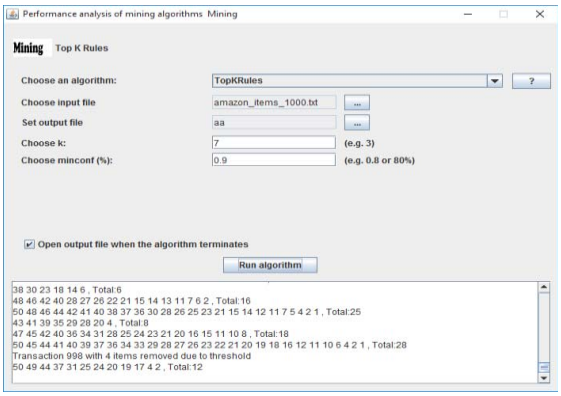
VII. IMPLEMENTATION OF TopKRule


Performance analysis of mining algorithms Mining
Mining Top K Rules
Choose an algorithm: TopKRules
Choose input file
Set output file
☑ Open output file when the algorithm terminates


Open
Look In: datasets
amazon_items_1000.txt
amazon_items_2000.txt
amazon_items_5000.txt
amazon_items_10000.txt
webusage_1000.txt
webusage_2000.txt
webusage_5000.txt
File Name:
Files of Type: All Files


Performance analysis of mining algorithms Mining
Mining Top K Rules
Choose an algorithm: TopKRules
Choose input file: amazon_items_1000.txt
Set output file: aa
Choose k: 7 (e.g. 3)
Choose minconf (%): 0.9 (e.g. 0.8 or 80%)
☑ Open output file when the algorithm terminates


Performance analysis of mining algorithms Mining
Mining Top K Rules
Choose an algorithm: TopKRules
Choose input file: amazon_items_1000.txt
Set output file: aa
Choose k: 7 (e.g. 3)
Choose minconf (%): 0.9 (e.g. 0.8 or 80%)
☑ Open output file when the algorithm terminates


Performance analysis of mining algorithms Mining
Mining Top K Rules
Choose an algorithm: TopKRules
Choose input file: amazon_items_1000.txt
Set output file: aa
Choose k: 7 (e.g. 3)
Choose minconf (%): 0.9 (e.g. 0.8 or 80%)
☑ Open output file when the algorithm terminates

```
38 30 23 18 14 6 , Total:6
48 46 42 40 28 27 26 22 21 15 14 13 11 7 6 2 , Total:16
50 48 46 44 42 41 40 38 37 36 30 28 26 25 23 21 15 14 12 11 7 5 4 2 1 , Total:25
43 41 39 35 29 28 20 4 , Total:8
47 45 42 40 36 34 31 28 25 24 23 21 20 16 15 11 10 8 , Total:18
50 45 44 41 40 39 37 36 34 33 29 28 27 26 23 22 21 20 19 18 16 12 11 10 6 4 2 1 , Total:28
Transaction 998 with 4 items removed due to threshold
50 49 44 37 31 25 24 20 19 17 4 2 , Total:12
```

VIII. CONCLUSION

In this paper we are presenting a literature survey on various algorithms used for mining high utility itemsets. We have focused on TKU and TKO algorithms for mining top-k high utility web access patterns. It has better performance than other algorithms in terms of runtime, especially when databases contain huge amount of long transactions. High utility itemset mining is a research area of utility based descriptive data mining. Utility based data mining is used for finding itemsets that contribute most to the total utility in that database. The discovered patterns can be used for better web page access prediction. This paper presents overview of algorithms for web page prediction along with the implementation part of TopKRules which is used for the pre-processing of dataset to remove unpromising data from Web usage and product base dataset.

REFERENCES

[1] R. Agrawal and R. Srikant, T. Imielinski, A. Swami, “Mining association rules between sets of items in large databases”, in proceedings of the ACM SIGMOD International Conference on Management of data, pp.207-216, 1993.

[2] Vincent S. Tseng, Cheng-wei Wu, Philippe Fournier-Viger and Philip S. Yu, “Efficient algorithms for mining Top-K high utility itemsets”, in IEEE Transactions on Knowledge and data engineering, vol.28 no.1 January 2016.

[3] J. Pisharath, Y. Liu, W.K. Liao, A. Choudhary, G. Memik, and J. Parhi, (2005). Numinebench version 2.0 dataset and technical report. Available at <http://cucis.ece.northwestern.edu/papers/DMS/MineBench.html>.Accessed on June 2015.

[4] Serin Lee, Jong Soo Park, “Top-K high utility itemset mining based on Utility List Structures”, In Proceedings of IEEE International Conference on Data Mining(ICDM), Maebashi, pp. 101 - 108, 2016

[5] Shuning Xing, Fangai Liu, Jiwei Wang, Lin Pang, Zhenguo Xu, “Utility Pattern Mining Algorithm bases on Improved Utility Pattern Tree”, in 8thbinternational symposium on computational intelligence and design, pp.258 – 261,2015.

[6] Smita R. Londhe, Rupali mahajan and Bhagyashree Bhoyar, “Overview on methods for mining high utility itemset from transactional database”, in international journal of scientific engineering and

research (IJSER), Volume1 Issue 4, December 2013.

[7] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," In Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Vol. 1215, pp. 487-499, 1994.

[8] Junqiang Liu, Benjamin C.M. Fung "Mining High Utility Patterns in One Phase without Generating Candidates" in ," IEEE Transaction Knowledge in Data Engineering, vol. 10, no. 12, pp. 1-14, Dec.2015.

[9] A.W.-C. Fu, R.W.-W. Kwong, and J. Tang, "Mining n-most interesting itemsets," In Proceeding of International Symposium on Methodologies for Intelligent Systems (ISMIS), Charlotte, Vol. 1932, pp. 59-67, 2000.

[10] M.-S. Chen, J.-S. Park, and P.S. Yu, "Efficient Data Mining for Path Traversal Patterns," IEEE Trans.Knowledge and Data Eng., vol. 10, no. 2, pp. 209-221, Mar. 1998.

[11] S.Shankar, T.P.Purusothoman, S. Jayanthi, N.Babu, "A fast agorithm for mining high utility itemsets", in Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464.

[12] W. Wang, J. Yang, and P. Yu, "Efficient mining of weighted association rules (WAR)," In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, pp. 270-274, 2000.

[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] Y. Liu, W. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," In Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Vol. 3518, pp. 689-695, 2005.

[15] C.F. Ahmed, S.K. Tanbeer, B.S. Jeong, and Y.K. Lee, "Efficient tree structures for high utility pattern mining in incremental databases," IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 12, pp. 1708-1721, 2009.

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 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.

[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] S. Lee and J.S. Park, "High utility itemset mining using transaction utility of itemsets," KIPS Transactions on Software and Data Engineering, Vol. 4, No. 11, pp. 499-508, 2015.

[24] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," In *Proceedings of IEEE International Conference on Data mining(ICDM)*, Maebashi, pp. 211-218, 2002.