SIMULATION AND ANALYSIS OF EFFICIENT ALGORITHMS FOR MINING TOP-K HIGH UTILITY ITEMSETS

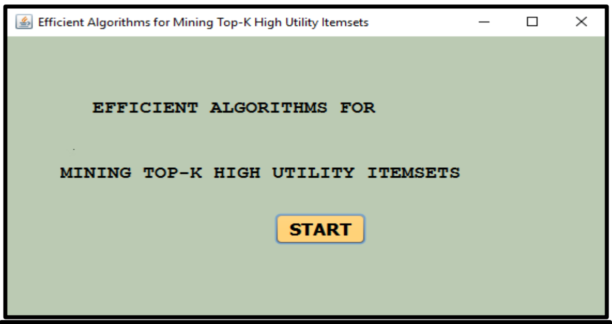
Surbhi Choudhary[1], Devendra Nagal[2], Swati Sharma[3]
[1]PhD Research Scholar, Dept. of Computer Applications, JNU Jodhpur
[2,3]Faculty, Dept. of Electrical Engineering, JNU Jodhpur

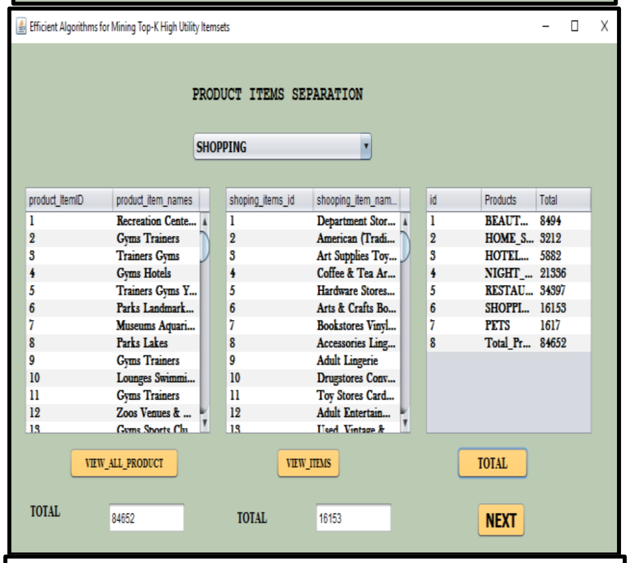
Abstract: High utility sequential pattern mining is an emergingtopic in the data mining community. Compared to the classicfrequent sequence mining, the utility framework providesmore informative and actionable knowledge since the utilityof a sequence indicates business value and impact. However,the introduction of "utility" makes the problem fundamentallydifferent from the frequency-based pattern mining frameworkand brings about dramatic challenges. Although the existinghigh utility sequential pattern mining algorithms can discoverall the patterns satisfying a given minimum utility, it is oftendifficult for users to set a proper minimum utility. A toosmall value may produce thousands of patterns, whereas atoo big one may lead to no findings. In this paper, we proposea novel framework called top-k high utility sequential patternmining to tackle this critical problem. Accordingly, an efficientalgorithm, Top-k high Utility Sequence (TUS for short) mining,is designed to identify top-k high utility sequential patternswithout minimum utility. In addition, three effective featuresare introduced to handle the efficiency problem, includingtwo strategies for raising the threshold and one pruning forfiltering unpromising items. Our experiments are conducted onboth synthetic and real datasets.

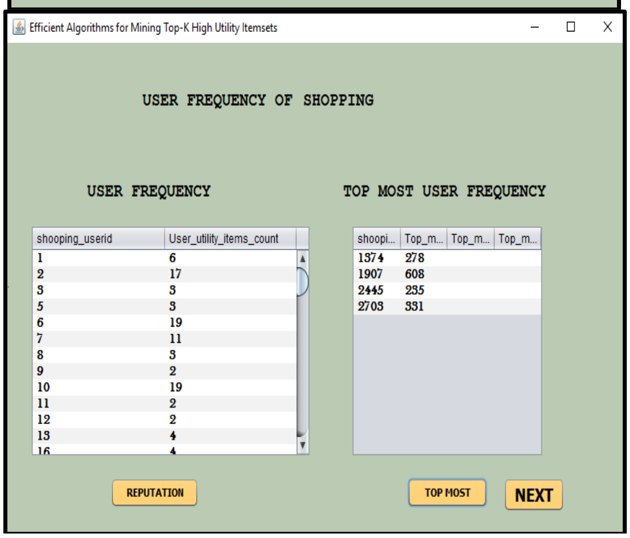
I. INTRODUCTION

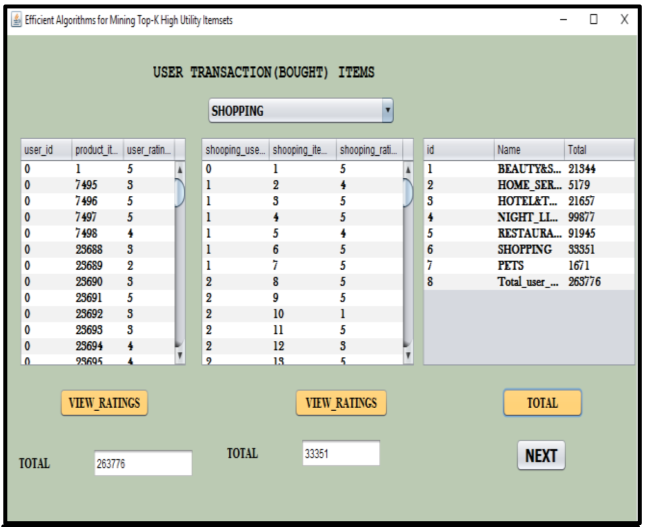
Frequent sequential pattern mining [1], as one of thefundamental research topics in data mining, discovers frequent subsequences in sequence databases. It is very usefulfor handling order-based business problems, and has beensuccessfully adopted to various domains and applicationssuch as complex behaviour analysis [1] and gene sequenceanalysis [2], [3], [4]. In the frequency-based framework fortypical sequence analysis, the *downward closure property*(also known as *Apriori property*) [1] plays a fundamentalrole in identifying frequent sequential patterns.However, taking the frequency to measure pattern in-terestingness may be insufficient for selecting actionablesequences associated with expected quality and businessimpact, because the patterns identified under the frequency(support) framework do not disclose the business valueand impact. To solve the above problems, the concept*utility* is introduced into sequential pattern mining to select sequences of high utility by considering the quality and value (such as profit) of itemsets. This leads to anemerging area, *high utility pattern mining* [1], [2], [3], [8],[9] and *high utility sequential pattern mining*, which selects interesting patterns / sequentialpatterns based on minimum utility rather than minimumsupport. The utility-based patterns are proven to be moreinformative and actionable for decision-making than the frequency-based ones [2]. For instance, in [4], [5], the authors discuss the extraction of profitable behaviours fromthe mobile commerce environments. [4] Proposes methodsto mine high utility sequences from web logs by assigningeach page an impact/significance. In [6], a US pan algorithmis built for utility-based sequential pattern mining satisfyinga predefined minimum utility.Although algorithms such as USpan can obtain high utilitysequences based on a given minimum utility, it is verydifficult for users to specify an appropriate minimum utilitythreshold and to directly obtain the most valuable patterns.This is because the complexity of utility-based sequencedatabases (which may be different from the classic itemsets),determining multiple factors including the distribution of theitems and utilities, density of the database, lengths of thesequences, and so on. Consequently, it is not surprising that,with a same minimum utility threshold, some datasets mayproduce millions of patterns while others may contributenothing. The challenge here is that it may not be doableto tune the threshold to capture the expected number ofpatterns. This is because the sensitivity of the thresholdmakes it hard to tune for a variety of databases. It maybe very costly and time consuming to achieve the properthreshold for the desired patterns.In fact, the classic frequency/support based pattern miningalso faces the same challenge. Accordingly, the concept ofextracting top-k patterns has been proposed in [2], [4], [7],[8] to select the patterns with the highest frequency. Inthe top-k frequent pattern mining, instead of letting a userspecify a threshold, the top-k pattern selection algorithmsallow a user to set the number of top-k high frequencypatterns to be discovered. This makes it much easier andmore intuitive and practical than determining a minimumsupport; also the determination of k by a user is morestraightforward than considering data characteristics, whichare often invisible to users, for choosing a proper threshold. The easiness for users to determine k does not indicatethe simplicity of developing an efficient algorithm for selecting top-k high utility sequential patterns. In the utilityframework, TKU is the only method for mining top-k high utility itemsets, to the best of our knowledge. Nowork is reported on mining top-k high utility sequences. There is significant difference between top-k utility itemsetmining and top-k utility sequence mining in which the orderbetween itemsets is considered. In fact, the problem of top-khigh utility sequence mining is much more challenging thanmining top-k high utility itemsets. First, as with high utilityItemset mining, the downward closure property does not holdin the utility-based sequence mining. This means that the existing top-k frequent sequential pattern mining algorithms [7]cannot be directly applied. Second, compared to top-k highutility itemset mining [8], utility-based sequence analysisfaces the critical combinational explosion and computationalcomplexity caused by sequencing between itemsets. Thismeans that the techniques in [9] cannot be directly transferred to top-k high utility sequential pattern mining either.Third, since the minimum utility is not given in advance, thealgorithm essentially starts the searching from 0 minimumsupports. This not only incurs very high computational costs,but also the challenge of how to raise the minimum thresholdwithout missing any top-k high utility sequences.

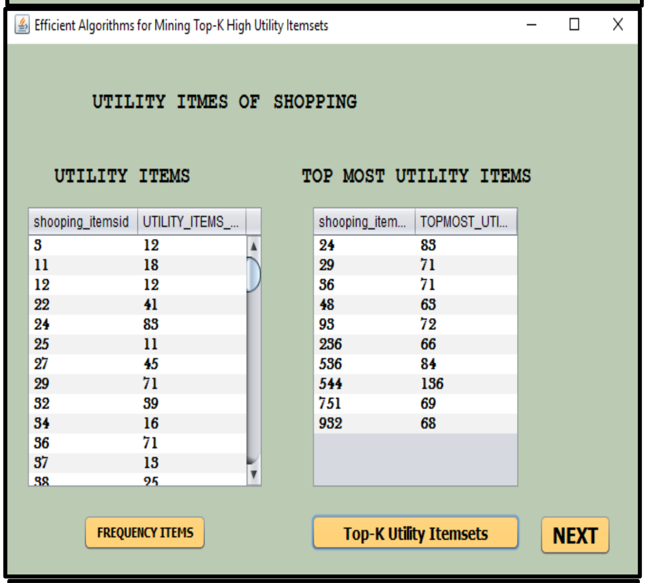
II. SIMULATION RESULTS

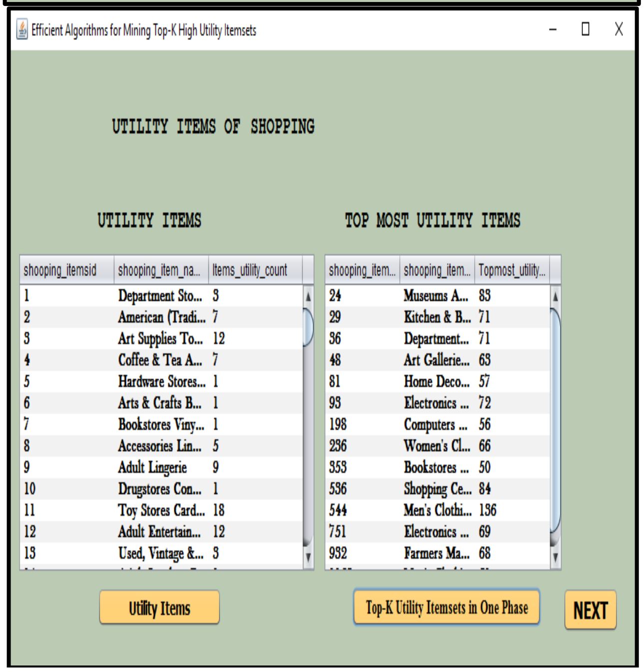


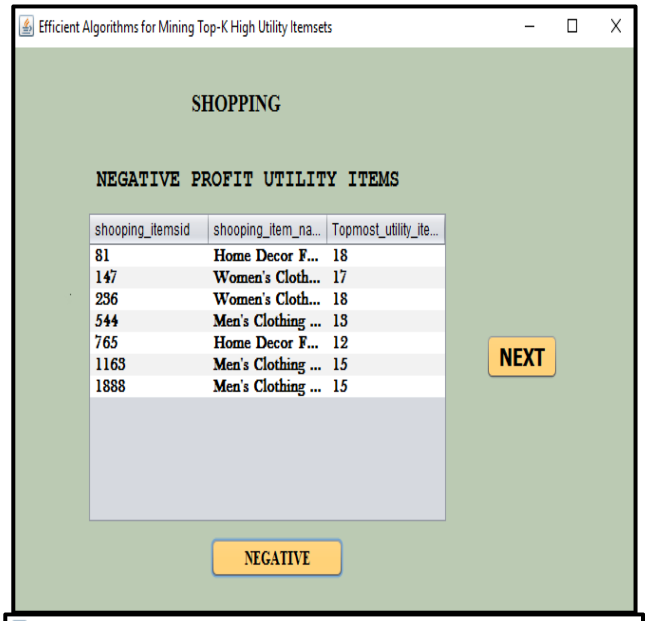


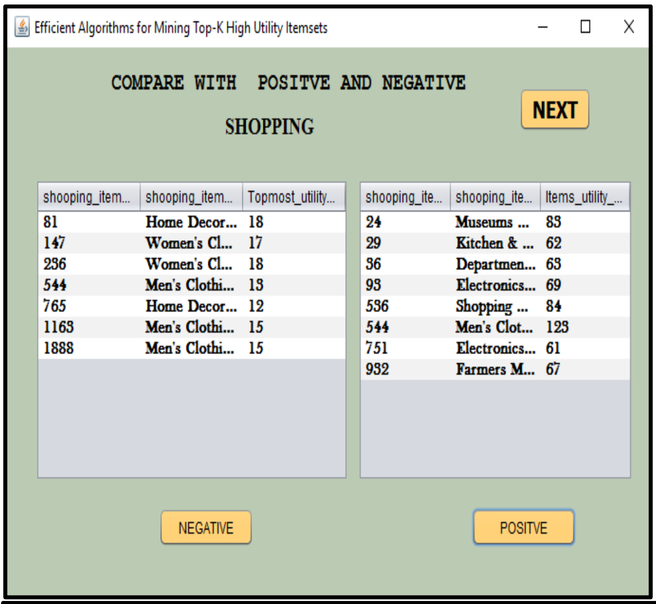


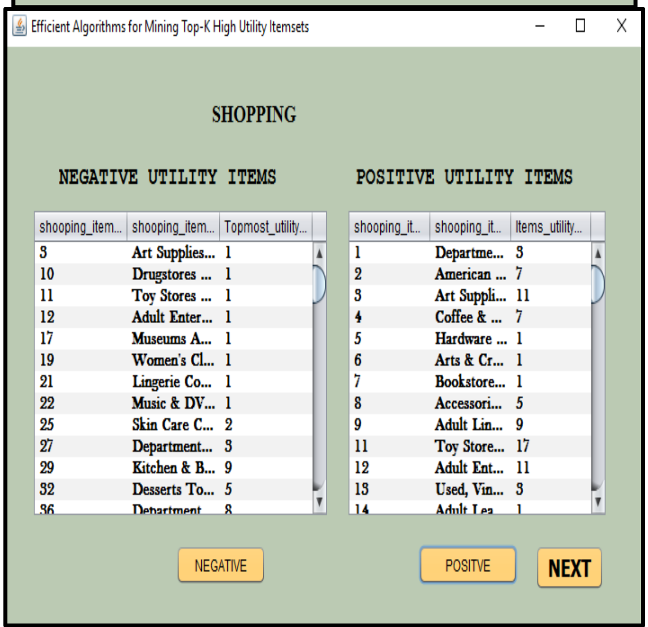


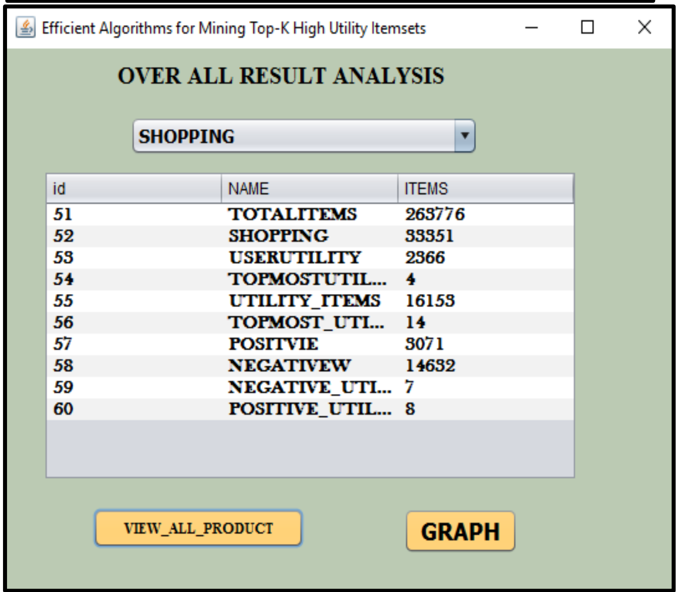


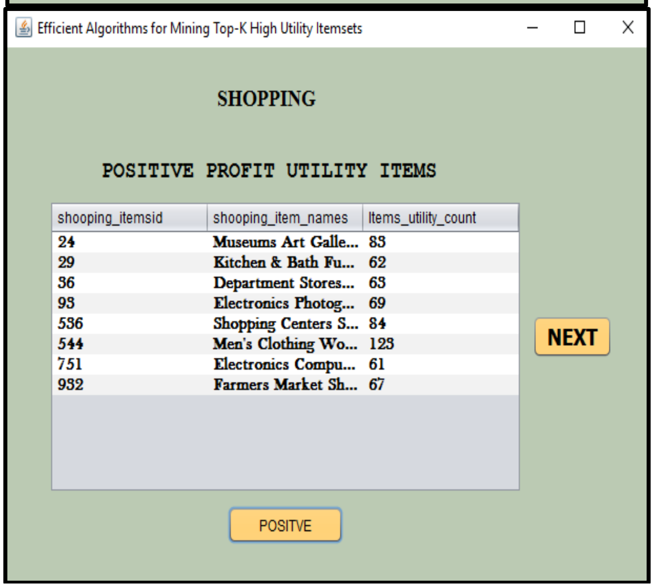


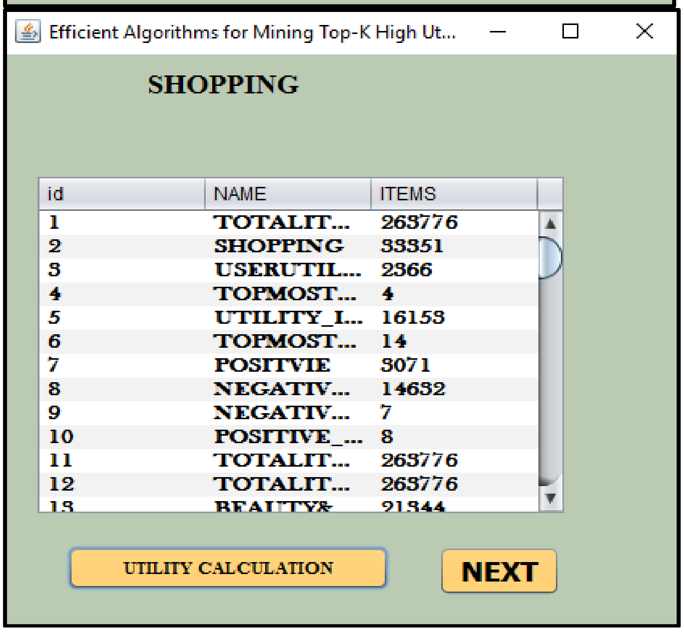

Efficient Algorithms for Mining Top-K High Utility Itemsets
SHOPPING
NEGATIVE PROFIT UTILITY ITEMS
shooping_itemsid
shooping_item_na...
Topmost_utility_ite...
81 Home Decor F... 18
147 Women's Cloth... 17
236 Women's Cloth... 18
544 Men's Clothing ... 13
765 Home Decor F... 12
1163 Men's Clothing ... 15
1888 Men's Clothing ... 15
NEXT
NEGATIVE

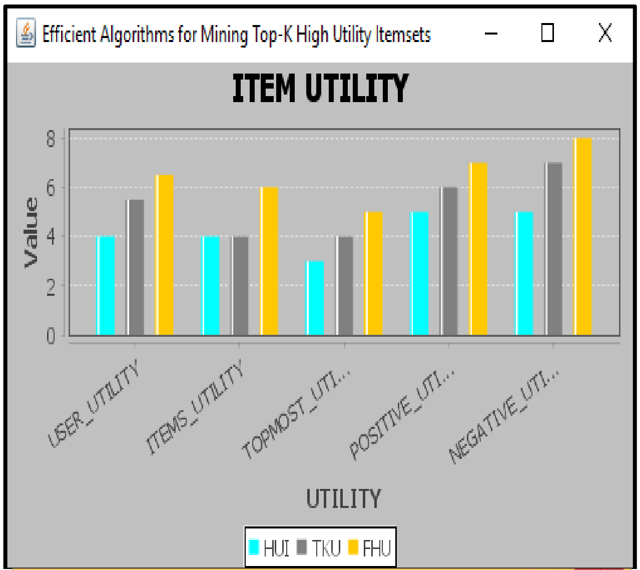

Efficient Algorithms for Mining Top-K High Utility Itemsets
COMPARE WITH POSITVE AND NEGATIVE
SHOPPING
NEXT
shooping_item...
shooping_item...
Topmost_utility...
81 Home Decor... 18
147 Women's Cl... 17
236 Women's Cl... 18
544 Men's Clothi... 13
765 Home Decor... 12
1163 Men's Clothi... 15
1888 Men's Clothi... 15
shooping_ite...
shooping_ite...
Items_utility_...
24 Museums ... 83
29 Kitchen & ... 62
36 Departmen... 63
93 Electronics... 69
536 Shopping ... 84
544 Men's Clot... 123
751 Electronics... 61
932 Farmers M... 67
NEGATIVE
POSITIVE


Efficient Algorithms for Mining Top-K High Utility Itemsets
SHOPPING
NEGATIVE UTILITY ITEMS
POSITIVE UTILITY ITEMS
shooping_item...
shooping_item...
Topmost_utility...
3 Art Supplies... 1
10 Drugstores ... 1
11 Toy Stores ... 1
12 Adult Enter... 1
17 Museums A... 1
19 Women's Cl... 1
21 Lingerie Co... 1
22 Music & DV... 1
25 Skin Care C... 2
27 Department... 3
29 Kitchen & B... 9
32 Desserts To... 5
shooping_it...
shooping_it...
Items_utility...
1 Departme... 3
2 American ... 7
3 Art Suppli... 11
4 Coffee & ... 7
5 Hardware ... 1
6 Arts & Cr... 1
7 Bookstore... 1
8 Accessori... 5
9 Adult Lin... 9
11 Toy Store... 17
12 Adult Ent... 11
13 Used, Vin... 3
NEGATIVE
POSITVE
NEXT

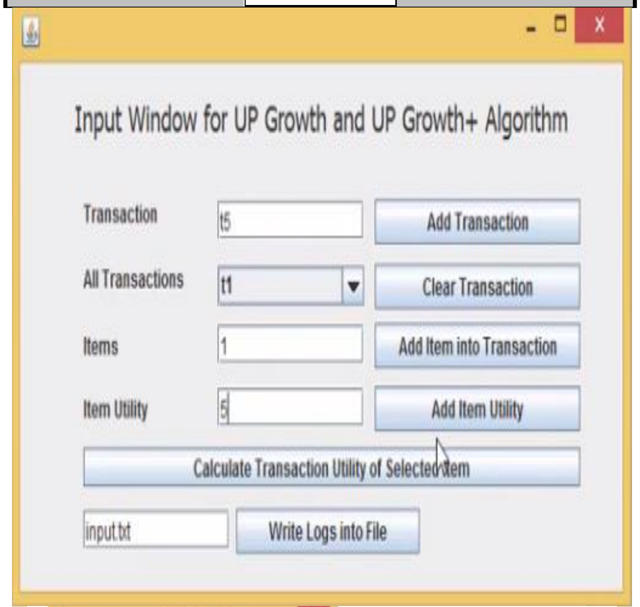

Efficient Algorithms for Mining Top-K High Utility Itemsets
OVER ALL RESULT ANALYSIS
SHOPPING
id
NAME
ITEMS
51 TOTALITEMS 263776
52 SHOPPING 33351
53 USERUTILITY 2366
54 TOPMOSTUTIL... 4
55 UTILITY_ITEMS 16153
56 TOPMOST_UTI... 14
57 POSITVIE 3071
58 NEGATIVEW 14632
59 NEGATIVE_UTI... 7
60 POSITIVE_UTIL... 8
VIEW_ALL_PRODUCT
GRAPH

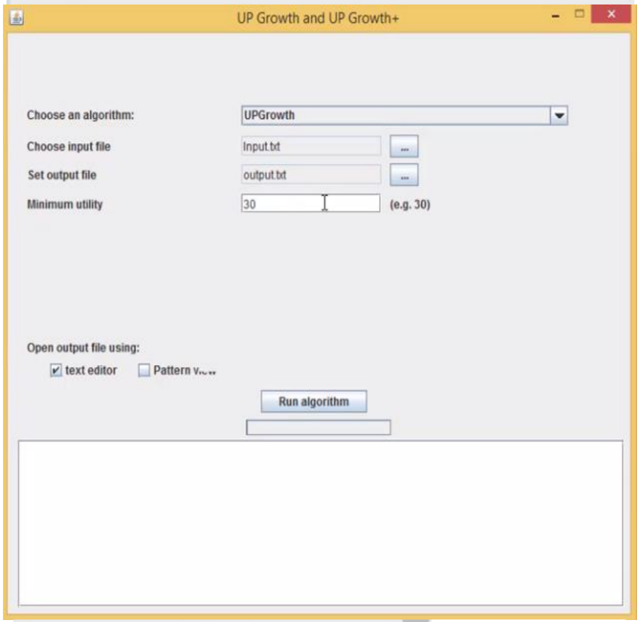

Efficient Algorithms for Mining Top-K High Utility Itemsets
SHOPPING
POSITIVE PROFIT UTILITY ITEMS
shooping_itemsid
shooping_item_names
Items_utility_count
24 Museums Art Galle... 83
29 Kitchen & Bath Fu... 62
36 Department Stores... 63
93 Electronics Photog... 69
536 Shopping Centers S... 84
544 Men's Clothing Wo... 123
751 Electronics Compu... 61
932 Farmers Market Sh... 67
NEXT
POSITVE

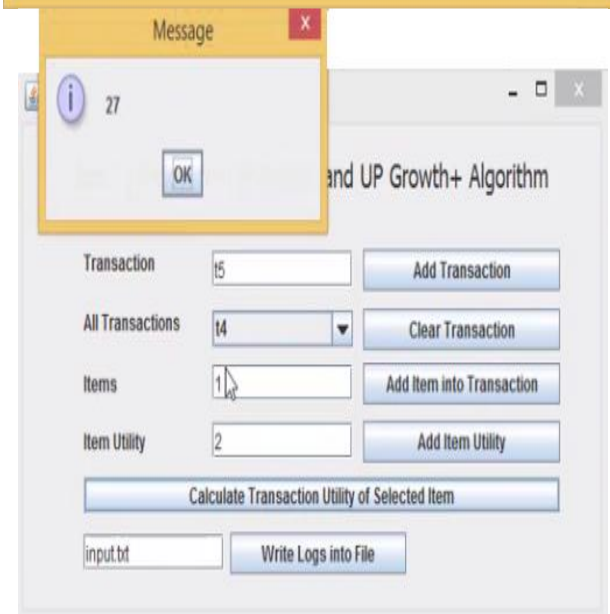

Efficient Algorithms for Mining Top-K High Ut...
SHOPPING
id
NAME
ITEMS
1 TOTALIT... 263776
2 SHOPPING 33351
3 USERUTIL... 2366
4 TOPMOST... 4
5 UTILITY_I... 16153
6 TOPMOST... 14
7 POSITVIE 3071
8 NEGATIV... 14632
9 NEGATIV... 7
10 POSITIVE_... 8
11 TOTALIT... 263776
12 TOTALIT... 263776
UTILITY CALCULATION
NEXT

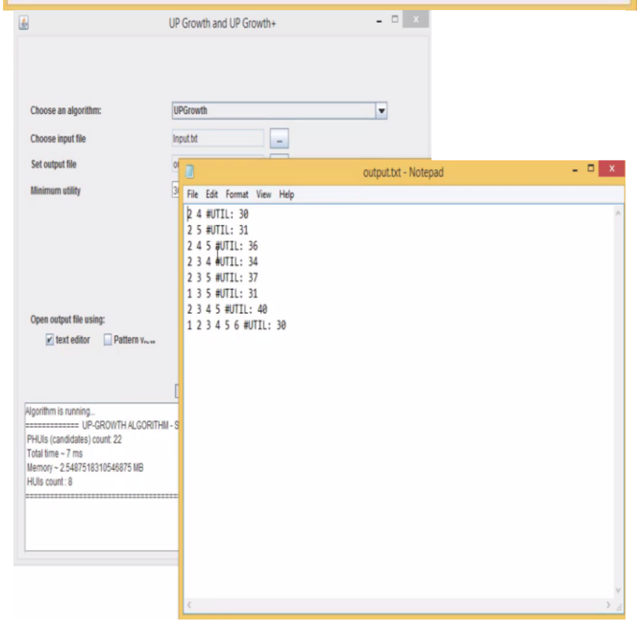

Efficient Algorithms for Mining Top-K High Utility Itemsets
ITEM UTILITY
Value
UTILITY
USER_UTILITY
ITEMS_UTILITY
TOPMOST_UTI...
POSITIVE_UTI...
NEGATIVE_UTI...
HUI TKU FHU

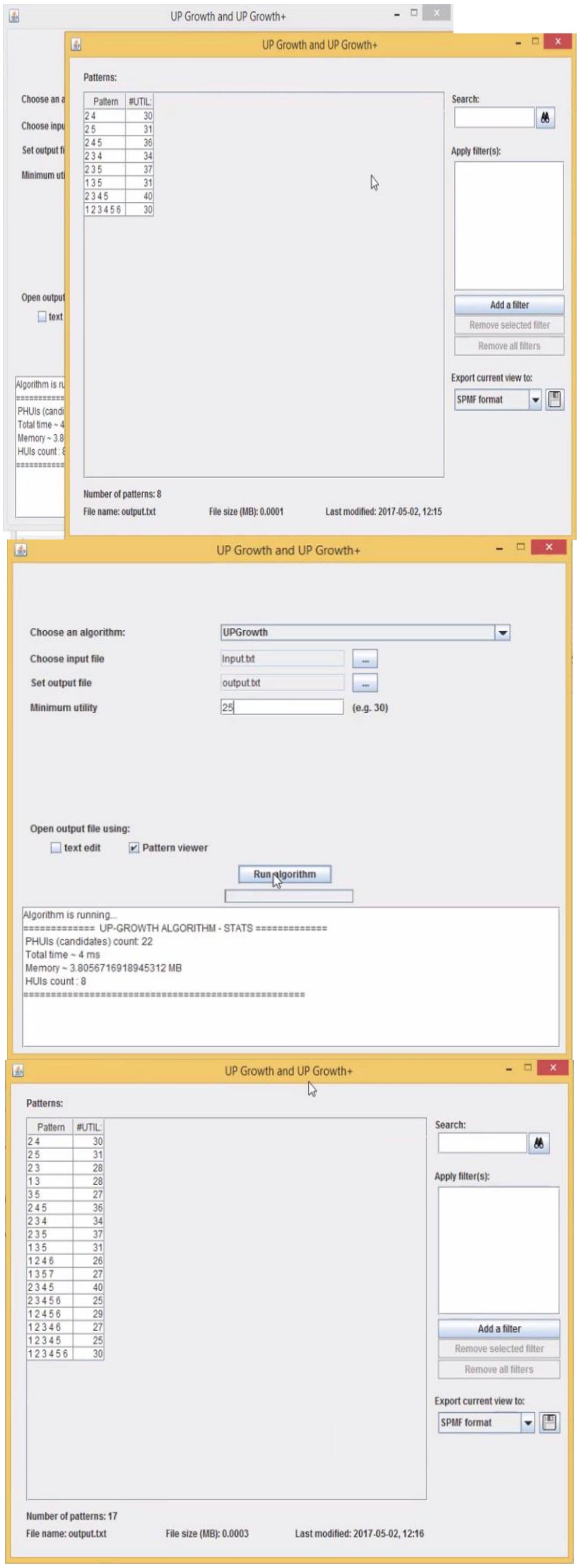

Input Window for UP Growth and UP Growth+ Algorithm
Transaction
All Transactions
Items
Item Utility
Message
Logs Successfully Written
OK
Calculate Transaction Utility of Selected Item
input.txt
Write Logs into File


Input Window for UP Growth and UP Growth+ Algorithm
Transaction
t5
Add Transaction
All Transactions
t1
Clear Transaction
Items
1
Add Item into Transaction
Item Utility
5
Add Item Utility
Calculate Transaction Utility of Selected Item
input.txt
Write Logs into File


UP Growth and UP Growth+
Choose an algorithm:
UPGrowth
Choose input file
Input.txt
Set output file
output.txt
Minimum utility
30
(e.g. 30)
Open output file using:
text editor
Run algorithm


Message
27
OK
Transaction
t5
Add Transaction
All Transactions
t4
Clear Transaction
Items
1
Add Item into Transaction
Item Utility
2
Add Item Utility
Calculate Transaction Utility of Selected Item
input.txt
Write Logs into File


UP Growth and UP Growth+
Choose an algorithm:
UPGrowth
Choose input file
Set output file
Minimum utility
Open output file using:
text editor
output.txt - Notepad
File Edit Format View Help
2 4 #UTIL: 30
2 5 #UTIL: 31
2 4 5 #UTIL: 36
2 3 4 #UTIL: 34
2 3 5 #UTIL: 37
1 3 5 #UTIL: 31
2 3 4 5 #UTIL: 40
1 2 3 4 5 6 #UTIL: 30
Algorithm is running...
PHUIs (candidates) count: 22
Total time ~ 7 ms
HUIs count : 8



Comparison with UPGrowth and UPGrowth+ Algorithm

In this project, two algorithms utility pattern growth (UP-Growth) and UP-Growth+, for mining high utility itemsets are used to compare. Performance of UP-Growth and UP-Growth+ become more efficient since database contain long transactions and generate fewer number of candidates than FP-Growth.

Two important problems are always in consideration first is how minimize number no of candidates and another is how to remove space and time complexity. Also, choosing an appropriate minimum utility threshold is a difficult task for application users: if the threshold is high, there might be no HUI; if the threshold is low, there might result too many HUIs, and the mining performance might be severely affected, even leading to memory overflow. It would also be a time-consuming task if one tries to determine the threshold value through various testing calculations.

To address this issue, Wu [10] proposes top-k algorithm, mining the top k itemsets with the highest utility values without presetting the minimum threshold.

Materials and Methods

Frequent itemset mining has been studied extensively in data mining. Recent studies have shown that it is more desirable to mine closed itemsets than the complete set of frequent itemsets. Efficient methods for mining closed itemsets, such as CLOSET, CHARM, and CLOSET+, have been developed. However, these methods all require a user-specified support threshold. Hidber presented Carma, an algorithm for online association rule mining, in which, a user can change the support threshold any time during the first scan of the data set (in other words, Carma still needs the user to specify the final support threshold), but its performance is worse than Apriori in general. In comparison with Carma, our algorithm does not need users to provide any minimum support and, in most cases, runs faster than two efficient algorithms, CHARM and CLOSET+ (running at the best tuned min_support thresholds), which, in turn, outperform Apriori substantially. Recently, there are proposals on association rule mining without support requirement, which are aimed at discovering confident rules instead of significant rules. As a result, they only use theconfidence threshold to prune rules of small confidence.Our motivation is different because our algorithm stilltargets at mining significant rules, but we do not need auser to specify any min_support threshold.

The problem of mining top-k frequent itemsets hasattracted the attention of some researchers recently. Fu et al.[12] studied mining N most interesting itemsets for everylength l, which is different from our work in several aspects:

- They mine all the itemsets instead of only the closed ones, and mining closed itemsets is not only moredesirable but also more challenging;
- They do not have minimum length constraints—since it mines itemsets at all the lengths, some heuristicsdeveloped here cannot be applied, and
- Their philosophy and methodology of FP-tree modificationare also different from ours.

To the best of our knowledge, this is the first study onmining top-k frequent closed itemsets with length constraint,therefore, we only compare our method with twowell-known efficient closed itemset mining algorithms.

From the user-interaction point of view, since ourperformance study shows that there is no real need tospecify min_l if one wants to mine frequent closed itemsetsof any length, and there is no crucial need to specify kfor top-k mining as long as k is a default number that fitsuser's expectation or application requirements, this methodgives the user the minimal burden to specify miningparameters, representing a step toward parameter-freefrequent-pattern mining.

There are extensive studies on mining frequent itemsetsfrom many different angles, such as constraint-basedmining [6], [4], [19], mining generalized and quantitativerules [2], [13], and mining correlation rules. Our study on mining top-k frequent itemsets isorthogonal to these studies. Since their mining andoptimization frameworks are based on a predefinedmin_support threshold, the techniques developed in thisstudy can be extended to the scope of these studies toimprove their corresponding algorithms for mining top-kfrequent itemsets. We also expect that the basic principlesdeveloped here can be applied to recently developed newfrequent itemset mining algorithms, such as whenthe requirement is changed to mining top-k frequentitemsets. Finally, it is expected that the philosophydeveloped here will influence the mining of top-k frequentstructured patterns, where a structured pattern may containsequences, trees, lattices, and graphs.

There are various methods for mining high utility itemsets. Mining high utility itemsets has four main methods used for from transactional databases that are given as follows:

Data Structure

Data Structure is nothing but organizing the data so that we can use that data efficiently. Mining high utility itemsets Keep in a special data structure called UP-Tree. This, compact tree structure, UP-Tree, is used for make possible the mining performance and avoid scanning original database repeatedly. It will also keep the transactions information and high utility itemsets.

UP-Growth Mining Method

In the first step we get the global UP tree that is mining UP-Tree by FP-Growth. Which can be used for generating PHUIs will generate so many candidates in order to avoid that UP-Growth method is used with two techniques mainly: First one is discarding unpromising items during constructing a local UP-Tree and second is discarding local node utilities.

An Improved Mining Method: UP-Growth+

As compared with UP-Growth FP-Growth gives the better performance. FP growth is used to find the frequent itemsets. FP-Growth uses DLU and DLN to decrease overhead utilities of itemsets. However, the overestimated utilities can be closer to their actual utilities by eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes. In this section, we propose an improved method, named UP-Growth+, for reducing overestimated utilities more effectively. In UP-Growth, minimum item utility table is used to reduce the overestimated utilities. In UP-Growth+, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database.

Efficiently Identify High Utility Itemsets

After finding all PHUIs, the third step is to identify high utility itemsets and their utilities from the set of PHUIs by scanning original database once [3], [11]. However, in previous studies, two problems in this phase occur: 1) number of HTWUIs is too large; and (2) scanning original database is very time consuming. In our framework, overestimated utilities of PHUIs are smaller than or equal to TWUs of HTWUIs since they are reduced by the proposed strategies. Thus, the number of PHUIs is much smaller than that of HTWUIs. Therefore, in phase II, our method is much efficient than the previous methods. Moreover, although our methods generate fewer candidates

III. CONCLUSION

In this paper, we have studied the problem of top-k high utility itemsets mining, where k is the desired number of high utility itemsets to be mined. 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. TKU is the first two-phase algorithm for mining top-k high utility itemsets, 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.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 itemsets, top-k high utility web access patterns and top-k mobile high utility sequential patterns. These leave wide rooms for exploration as future work.

REFERENCES

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

[2] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, "Efficient tree structures for high-utility pattern mining in incremental databases," IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, 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 itemsets mining algorithm,” in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.
[14] M. Liu and J. Qu, “Mining high utility itemsets without candidate generation,” in Proc. ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.

