International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6045
***Article Received**: 15 November 2022 **Revised**: 22 December 2022 **Accepted**: 28 December 2022*

An Approach for Mining Top-k High Utility Item Sets (HUI)

[1]K. Venkata Ramana, [2]A Muralidhar , [3]Bhanu Chander Balusa, [4]M.Bhavsingh, [5]Sravanya Majeti
Assistant Professor, Department of Computer Science and Engineering , VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad-500 090, Telangana, India.
[4]Assistant Professor, Ashoka Women's Engineering College, Kurnool, Andhra Pradesh
[2,3,5] School of Computer Science and Engineering , Vellore Institute of Technology, Chennai
Email: venkataramana_k@vnrvjiet.in

Abstract: Itemsets have been extracted by utilising high utility item (HUI) mining, which provides more benefits to the consumer. This could be one of the significant domains in data mining and be resourceful for several real-time implementations. Even though modern HUI mining algorithms may identify item sets that meet the minimum utility threshold, However, fixing the minimum threshold utility value has not been a simple task, and often it is intricate for the consumers when we keep the minimum utility value low. It might generate a massive amount of itemsets, and when the value is at its maximum, it might provide a smaller amount of itemsets. To avoid these issues, top-k HUI mining, where k represents the number of HUIs to be identified, has been proposed. Further, in this manuscript, the authors projected an algorithm called the top-k exact utility (TKEU) algorithm, which works without computing and comparing transaction weighted utilisation (TWU) values and deliberates the individual utility item values for deriving the top-k HUI. The datasets are pre-processed by the proposed algorithm to lessen the system memory space and to provide optimal outcomes for condensed datasets.

Keywords: TWU, TKEU, Top-k HUI, Itemset

I. INTRODUCTION

Data mining refers to the process of identifying useful patterns and valuable information in large amounts of data and transforming raw data into understandable and meaningful information. It plays an important role in the world of business by enhancing their business growth and providing assistance in making decisions based on market sales. HUI mining, top-k HUI mining[1], and finally mining of weighted itemset were all methods for examining data sales and retrieving the most profitable set of items from voluminous data sales.

HUI mining prioritises profit and item quantity. Creating HUI from massive datasets is an exciting area of data mining. When an item set has a utility value that is similar to or greater than the minimum utility value specified by the user, it is designated as HUI. Only when an item set meets the minimum utility value can it be designated as HUI. The customers have no idea which MU value is appropriate for their needs. The MU specification is emphasised because it has a direct impact on the number of HUIs. When the value of mu is low, a large amount of HUI may be generated; when the value of mu is high, a small number of itemises may be produced; both of these outcomes were not precise.

Top-k HUI mining has been proposed to address the issue of providing minimum utility. In this top-k HUI mining, regardless of the value of *mu*, the *k* value is required. Customers must supply the *k* parameter, which indicates the number of itemsets to be identified. Identifying *k* is thus easier than determining the value of *mu* [2]. Nonetheless, developing an efficient algorithm for HUI mining is a significant task.

The remainder of the paper is organized as follows: Section 2 contains a review of the literature, Section 3 contains a problem definition with the proposed work, Section 4 contains an experimental evaluation and performance study, and Section 5 contains the paper's conclusion and future scope.

II. LITERATURE REVIEW

The algorithms based on the two-phase method include several stages for generating HUIs. It is primarily having significant issues because it scans the dataset multiple times and produces more itemset candidates. The one-phase algorithms were created to address these issues. The HUI-miner was previously contributed to a one-phase algorithm.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6045
***Article Received**: 15 November 2022 **Revised**: 22 December 2022 **Accepted**: 28 December 2022*

Table 1. Comparison of the related work

Reference	Year of publishing	Title	Algorithm	Outcome	Drawback
[1]	2016	Effective Algorithms for Top-K Item-sets Mining	TKO, TKU algorithm	Structural comparison among algorithms and verifying the effectiveness of the algorithm	It has been taking huge search space to store itemsets utility information
[2]	2017	Top-k HUI from a stream of data under the sliding window method	Top-k SW structure of data	Isolating the massive datasets into the batches & values are stored in tree form	Using the downward closure property Utilizing DCP (downward closure property)
[3]	2017	An effective algorithm for top-k on-shelf HUI mining	KOSHU (rapid top-K On-Shelf HUI miner)	Enhancing several schemes of algorithms for augmenting the effectiveness	Several computations in algorithms require to do manifold data scans
[4]	2018	A dynamic method for mining Top K HUI	Stemming & cooperation filtering algorithm	Compute correlation in HUI mining and highly related profitable itemsets	Consuming a vast amount of memory to preserve entire itemsets, which are correlated itemsets
[5]	2019	Effective & efficient closed top-k HUI mining	FHUI miner	Producing the structures of utility list for identifying closed itemsets of utility	Consuming vast amount of search space & performing time

1.1 Literature survey table

It depicts the structure of the utility list to preserve the items' information. FHM and other 1-phase algorithms such as KOSHU, TKO, H-Miner, and EFIM were later projected. Further, the above-examined algorithms might suffer from one of the significant issues, which specifies the minimum threshold utility, which is not a simple task for consumers.

To evade the confines of providing minimum threshold utility, the top-k HUI mining algorithms have been projected. The process of mining top-k HUI algorithms is primarily fixing the minimum internal threshold utility to 0 or 1. Later, the algorithm enhances the internal threshold by utilising some schemes. The idea behind designing threshold-improvement schemes is to avoid missing any of the top-k HUI. Later, some of the schemes were also devised for search space pruning. Finally, top-k HUIs have been derived from the algorithms[6].

(i) Table 2: Here, table 2 represents the transaction table, which exhibits numerous transactions. The id of the transaction described in the column is the identification number of transactions, whereas transaction data could be items comprising a quantity.

Table 2: represents transaction table

Transaction id	T1	T2	T3	T4	T5	T6
Transaction data	(A,1), (B,2) (D,4)	(A,3), (B,6), (C,5), (D,2), (E,1)	(B,2), (C,1), (D,1)	(B,2) (C,3) (E,4)	(A,3) (B,4) (D,6) (E,2)	(C,1) (D,3) (E,2)

(ii) Table 3: It represents the profit table, which exhibits the item name and every item profit for one unit.

Table 3: represents profit table

Item	A	B	C	D	E
Unit profit (rupees)	3	5	4	1	2

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6045
Article Received: 15 November 2022 Revised: 22 December 2022 Accepted: 28 December 2022

(iii) Table 4: The below table represents the values of transaction utility over the transaction id.

Table 4: The table represents the values of transaction utility over the transaction id.

Transaction id	T1	T2	T3	T4	T5	T6
Transaction utility	17	63	15	30	39	11

(iv) Table 5: The below table has item name information and utility values of an item

Table 5: The table has item name information and utility values of an item

Item	A	B	C	D	E
Item utility	21	80	40	16	18

The challenge of top-k HUI is to avoid losing the valuable information in the item sets, thereby reducing storage space and avoiding the generation of candidate item sets. To tackle this challenge, an algorithm is proposed for mining HUI[7]. The performance is carried out on datasets, which derive prominent enhancements in accuracy and speed over contemporary top-k HUI mining algorithms.

III. PROBLEM DEFINITION

- **Utility:** Multiplication of internal utility & external utility is called utility.
  For instance Utility (A) in T1 = 1(quantity) * 3(unit profit) = 3
- **Internal utility:** The transaction table comprising of transaction id information and items along with their values of quantity. In Table 2, T1 could be id number of transactions and items through quantity (A,1) (B,2) (D,4). The transaction data internal utility has termed as quantity
- **External utility:** The profit table possess items by their profit units. In table 3, item A is having 3Rs per unit profit. Here, profit could be an external utility due to the stability of profit by the consumer. The consumer might fix any profit of his or her item.
- **Transaction utility:** Table 4 depicts the utility table of the transaction, which has been computed from the profit table and transaction data by multiplying the unit profit and quantity and aggregating the values in the form of a utility transaction. Further, in table 4, the term T1 has utility transactions of 17.
- Transaction utility (T1) = A(3*1)+ B(2*5)+D(4*1) = 3+10+4 = 17.
- **Item utility:** Table 5 is exhibiting utility item values; these computed the correct utility values of items, which calculated from data transactions.

Item Utility (A) = A [T1 (3*1) + T2(3*3) + T5(3*3)] = A [3 + 9 + 9] = 21.

- **High Utility Itemset:** When an item set is known as HUI, then item set value has to be equivalent or more than minimum threshold utility value. Or else, the item set is considered to be less utility item-set.

1.2 Proposed work

In this manuscript, the researchers propose a concept of computing the correct utility value of items regardless of the TWU. It is because of overestimating the value of the item and utilizing the DCP that when an item is not cost-effective, then the item subset is also not cost-effective, and these results in missing cost-effective items. To overcome this issue, we identify a novel method where the item utility value is compared to the minimum utility value. Moreover, if those values of utility items fulfill the minimum values of utility, then such items were deliberated to be HUI values[8].

In this algorithm, we calculate utility item and transaction utility values, and we were not computing the TWU values. Here, we are implementing a quick sort algorithm for lessening search space and organizing the overall utility item values in an incrementing sequence, which is resourceful for enhancing the value of the threshold. After computing the utility itemset values, the minimum value of utility is one and compared to the utility itemset values. Increasing the value of the minimum threshold utility by using these schemes also increases the threshold value. When compared to the increased value of the threshold, the value of the itemset should also be the same as or more than the minimum utility value. Organize all the values of item utility in an incrementing sequence. Based on item sets, a K-value has been provided. Furthermore, after providing the k-value, it may display the top-k HUIs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6045
***Article Received**: 15 November 2022 **Revised**: 22 December 2022 **Accepted**: 28 December 2022*

The procedure is described in detail, as are the results of the analysis. We computed the item utility and transaction utility values. Figure 4.1 describes a decreasing search space. The strategies for increasing the threshold value of minimum utility have been explored in figure 2.

1.3 The search spaces

Reducing the search space is a significant challenge for the top-k HUIs mining algorithm. To lessen the search space, the quick sort algorithm is applied and extensively utilised, which makes comparisons of n log n in an average case to sort the n elements of an array. Here, the divide-and-conquer approach has been followed by this algorithm for storing utility item values in an array. Organize the itemset utility values in augmenting sequence, which lessens search space.

1.4 Raising Minimum utility threshold schemes

Pivot scheme:

Input: Compute the itemset utility values

Output: Raised minimum-utility threshold-value.

- computing the values of utility itemset
- organize the overall values in the approach of array index
- The median value has picked in the form of pivot
- Values of minimum utility have set in the form of pivot value
- It has compared with divide, conquers & pivot the values of the index in the form of partitions
- arrange the values in ascending sequence.
- The minimum utility threshold-value has returned.

1.5 Proposed TKEU Algorithm

Input: The transaction dataset required several HUIs

Output: Top-k HUI

- Transaction utility values should be computed
- The table of item utility should be computed
- The minimum utility value has initialized as 1
- Implementing quick-sort-algorithm
- Arrange the values in the augmenting sequence.
- Utilizing the pivot scheme
- Increasing the threshold value of minimum utility.
- And comparing to the value of minimum utility
- K value is specified
- Exhibiting top-k HUIs

1.6 An example

In this manuscript, we have specified a simple example for explaining how the projected algorithm might identify the top-k HUIs. Five items and six transactions are assumed in the dataset, as exhibited in table 2. Table 2 shows itemets with their quantity of purchase as Quantity A (T1) = 1 and estimated Unit Profit (A2) = 3. Moreover, itemised utility has been computed by multiplying both unit profit and quantity[9]. Also, the values of transaction utility were computed and exhibited in table 4. The utility item-set value is calculated for entire item sets, as displayed in the below-shown diagram.

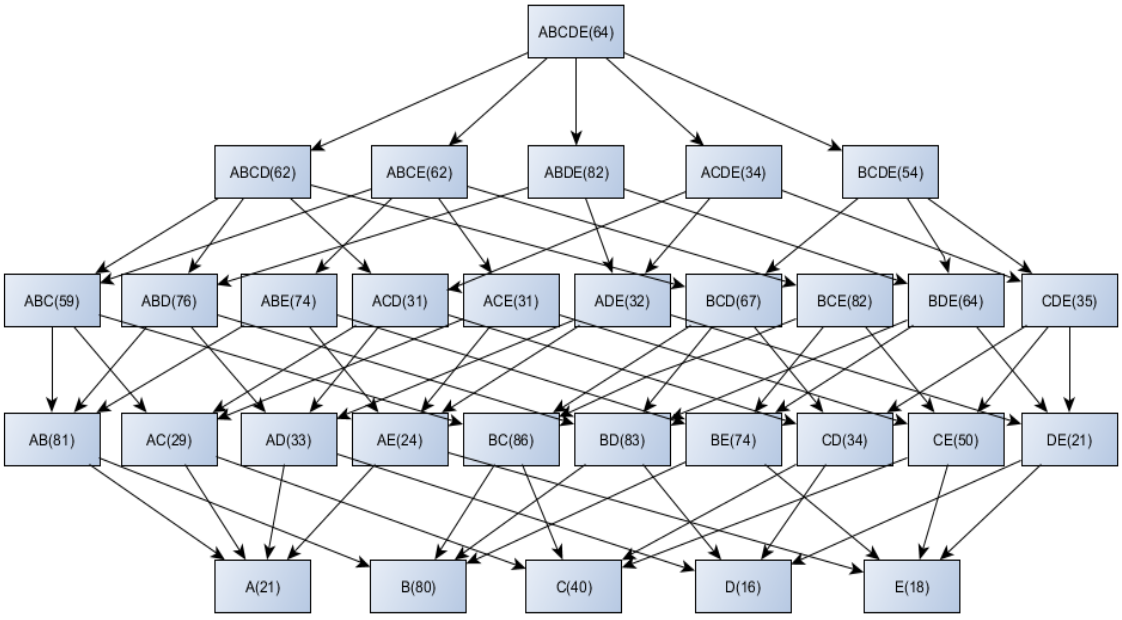

Figure 1: Itemsets with their item utility value

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6045
***Article Received**: 15 November 2022 **Revised**: 22 December 2022 **Accepted**: 28 December 2022*

Implementing a quick-sort algorithm on dataset transactions, the overall dataset is stored in an array and segmented into partitions. The range of the stored array index is 0–30 itemsets, and we consider mid-value as a pivot. In this case, we take the median of 0 and 30 to be 15. Further, in 15 indexes, we possess the value of item utility 35; as per the pivot scheme, the minimum utility value has been increased from 1 to 35. The algorithm would carefully select the itemsets whose item utility value matches the value of minimum utility. The remaining item sets were determined to be of the utmost utility [10][11].

Table 6. Shows the Itemset with utility

Itemset	Item Utility
BC	86
BD	83
ABDE	82
BCE	82
AB	81
B	80
ABD	76
ABE	74
BE	74
BCD	67
ABCDE	64
BDE	64
ABCD	62
ABCE	62
ABC	59
BCDE	54
CE	50
C	40
CDE	35

The above-stated itemsets fulfilled the threshold value of minimum utility and placed entire itemsets in ascending order. The total number of itemsets is 19, and the k value must be specified to display maximum utility itemsets[12][13]. When the value of k is 10, then it exhibits the top-10 HUIs with their utility item values. Some of the top-k HUIs {BC}=86,{BD}=83, {ABDE}=82,{BCE}=82,{AB}=81,{B}=80,{ABD}=76,{ABE}=74,{BE}=74,{BCD}=67.

IV. EXPERIMENTAL EVALUATION AND PERFORMANCE STUDY

Total experiments have carried out on CPU of 2.70 GHz with memory 8GB, Intel Core i5-6700 machine, running windows pro 10. The proposed algorithm performance has compared with TKO, TKU, on three real available datasets from repository UCI [11].

1.7 Dataset information

Table 7 . Dataset information

Dataset	Mushroom	Food mart	Retail
No of transactions	8124	4141	541909
No of items	119	1559	176740
Type	Dense	Sparse	Sparse

1.8 Memory Usage

In this section, we describe how we used proposed algorithm memory and cutting-edge algorithms on three different datasets: Food Mart, Mushroom, and Retail.

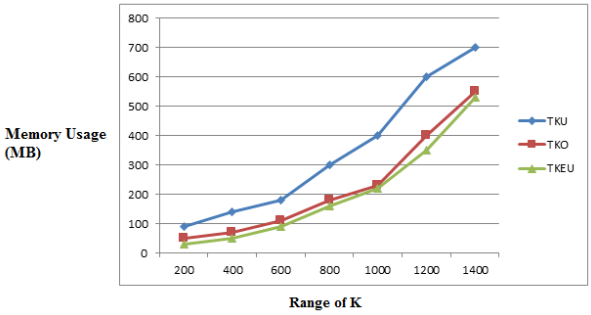

Figure 2. Memory Usage

The proposed algorithm uses less memory than TKO and TKU for the entire datasets. Moreover, the proposed algorithm, TKEU, is better for both sparse and dense datasets. Also, more memory is consumed by TKU than by total algorithms because it comprises two stages, and the succeeding structure of the UP-tree consumes more space for storing data. Here, TKO is better than TKU since it stores data in the structure of the utility list. There is minimal computation work in the TKEU algorithm, and data is stored in an array and partitioned into partitions. Further, the quicksort algorithm is implemented on datasets to prune the search space. Schemes of the proposed algorithm have also been consuming less memory.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6045
***Article Received**: 15 November 2022 **Revised**: 22 December 2022 **Accepted**: 28 December 2022*

1.9 Scalability

Scalability changes capacity size. Based on the algorithm, the overall dataset size has been segmented into small divisions from 20–100%. Here, this enhances the proposed algorithm's running time, which provides optimal outcomes compared to other algorithms.

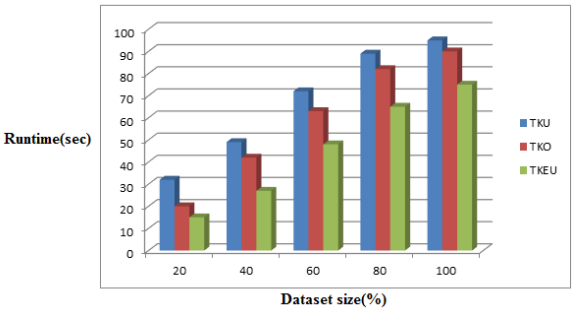

Figure 3. Scalability

We also contributed three datasets for evaluating scalability: Food Mart, retail, and mushroom. TKEU consumes less memory and time than the TKO and TKU algorithms. In this case, the running time increases linearly with the increased size of the dataset. Furthermore, we observe that the algorithm's memory utilisation increases with the number of transactions. This shows that the TKEU scales better with dataset size.

V. CONCLUSION & FUTURE DIRECTIONS

An algorithm for mining the top-k HUIs is proposed in this paper. The TKEU algorithm used techniques to increase the value of the minimum utility threshold. To prune the search space and save memory, it employs an array-based structure and a quick sort algorithm. We use item utility values rather than weighted transaction utility values. The simulation results demonstrated that the proposed algorithm outperformed the TKO and TKU algorithms. Furthermore, the proposed algorithm has the potential to be more effective and faster than existing algorithms. For both sparse and dense datasets, the analysis of running time and memory consumption is superior. This paper is an important contribution to top-k HUI mining. We also broadened our research to include data streams, shelf-based HUI mining, and sequential pattern mining.

REFERENCES

[1] Efficient Algorithms for Mining Top-K High Utility Itemsets using data stream, Vincent S. Tseng, Senior Member, Cheng WeiWu, Philippe Fournier-Viger, Philip S. Yu, Fellow, IEEE2016.

[2] Efficient Algorithms for Mining High Utility Transactional Databases, Bai-En Shie, Philip S. Yu,Fellow, Cheng-Wei Wu, and Vincent S. Tseng, IEEE transactions on knowledge and data engineering, vol. 25,no. 8, august 2017.

[3] Dawar, S., Sharma, V., & Goyal, V. (2017). Mining top-k high-utility itemsets from a data stream under sliding window model. Applied Intelligence, 47(4), 1240-1255.

[4] J. Yin, Z. Zheng, L. Cao, Y. Song, and W. Wei, "Dynamic approach for mining top-k high utility items using stemming and collaborative filtering algorithms " in Proc. IEEE Int. Conf. Data Mining, 2018, pp. 1259–1264.

[5] Nguyen, L. T., Vu, V. V., Lam, M. T., Duong, T. T., Manh, L. T., Nguyen, T. T., ... & Fujita, H. (2019). An efficient method for mining high utility closed itemsets. Information Sciences, 495, 78-99.

[6] Efficient Algorithms for Mining Top-K High Utility Itemsets using data stream, Vincent S. Tseng, Senior Member, Cheng WeiWu, Philippe Fournier-Viger, Philip S. Yu, Fellow, IEEE2015.

[7] Efficient Algorithms for Mining High Utility Transactional Databases, Bai-En Shie, Philip S. Yu,Fellow, Cheng-Wei Wu, and Vincent S. Tseng, IEEE transactions on knowledge and data engineering, vol. 25,no. 8, august 2013.

[8] Fournier-Viger, P., Zhang, Y., Lin, J. C. W., Dinh, D. T., & Le, H. B. (2018). Mining correlated high-utility itemsets using various measures. Logic J IGPL Google Scholar.

[9] Fournier-Viger P, Wu C-W, Zida S, Tseng VS (2014b) FHM: faster high-utility itemset mining using estimated utility cooccurrence pruning. Springer International Publishing, Cham, pp 83–92.

[10] Fournier-Viger P, Zida S (2015) Foshu: faster on-shelf high utility itemset mining – with or without negative unit profit. In: Proceedings of the 30th annual ACM symposium on applied computing, SAC '15. ACM, New York, pp 857–864.

[11] Y.Usha Sree,P.Ragha Vardhani (2015). Pattern Finding in Large Datasets with Big Data Analytics Mechanism. International Journal of Computer Engineering In Research Trends, 2(5), 359-364.

[12] P.M. Gavali1(2018). Investigation of Mining Association Rules on XML Document. International Journal of Computer Engineering In Research Trends, 5(1), 12-15.

[13] B.Senthilkumaran , K.Thangadurai(2017). A Comparative Study of Discovering Frequent Subgraphs – Approaches and Techniques. International Journal of Computer Engineering In Research Trends, 4(1), 41-45.