International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

# Joint Probability based Minimum Utility Threshold for Mining Top-K High Utility Itemsets

**Ms. Meghana Rajput[1], Ms.Hemali Shah[2]**

[1]*ME Computer, M.S.Bidve COE, Latur, Maharashtra, India*
[2]*Assistant Professor, M.S.Bidve COE, Latur, Maharashtra, India*

---

**Abstract:** *High Utility Itemset (HUI) mining rises as an active topic in the field of data mining that helps in identifying the itemsets that satisfy the demands of the user. The demands of the user are buried in the function name as minimum utility measure, but designing the minimum utility value is a hectic challenge of the time. The problem with the minimum utility is that the low-value of the minimum utility establishes a large number of HUIs and a greater value of the minimum utility contributes to ineffective mining. Thus, the problem of setting the minimum utility value is tackled using the proposed method of designing the minimum utility based on the Joint probability. The Transaction Utility (TU) and Total Weighed Utility (TWU) are computed using the Internal Utility (IU), and External Utility (EU) of the utility itemsets and the computation table that is employed for computing the joint probability of the itemset is formed. Experimentation performed using the datasets, such as Retail, Chess, and T10I4D100K datasets proves the effectiveness of the proposed method. The mined patterns are reported as 133 and the time is noted as 74secs, which highlights that the proposed method is better compared with the existing methods.*

***Keywords:*** Utility mining, Joint probability, Total utility, Total weighed utility, UP Tree.

## 1. Introduction

Data mining extracts highly essential and significant information buried in large databases that finds valuable application in supermarket promotions, biomedical, multimedia, mobile applications, and so on [7]. Association Rule Mining (ARM) [9–11] is a significant data mining that aims at unsheathing the interesting patterns from the transaction databases. The frequently occurring data is found based on the minimum support threshold, from which the Association Rules (ARs) are established based on the minimum confidence threshold [9] [3]. Frequent Itemset Mining (FIM) [12, 13] is one among the representative association rule mining approach as they are simple and efficient [2] and they are mainly employed on the analysis of the market basket. FIM fails in this application as they lose the valuable information of products that possess less frequency. The issue regarding the FIM is that they consider all items that possess equivalent importance/unit profit/weight and considers all the items of the transaction database in binary format either as available or unavailable. This imposes the challenge that FIM does not meet the user requirements, who aim at collecting the itemsets with high profits. Thus, utility mining [14], [15], [16] raises as an interesting research topic in data mining [1] [4] [5].

In utility mining, the individual item or product possess a unit weight and should frequently occur in the transaction and utility is a factor that gains significance with respect to the weight, cost, profit, quantity or any other data highlighting the user preference [6] [8]. Utility mining finds valuable application in almost all streams [14], [17], [18], [19], [20], and [14] [1]. An itemset is simply referred as a HUI when its utility exceeds the utility threshold set by the user, and a data is called as a low utility itemset if the threshold is greater than the utility of the item [1]. Therefore, the role of mining is to determine the significance of the high utility itemset [23]. Additionally, high utility itemset mining is used in the various analysis [21], and moreover, in the analysis of biological gene database [23] [5].

The paper proposes the method of computing the minimum threshold value using the Joint probability. Initially, the input database containing the information of the itemset is taken for the analysis and the analysis contains two parameters, such as TU and TWU. The calculation of TU and TWU is based on IU and EU of the individual items in the database, and the total utility of the transactions. Finally, the UP Tree is constructed, and the minimum threshold for mining the utility is designed based on the Joint probability of the datasets.

The contribution of the paper is:

**Joint Probability for computing the minimum threshold value:** The contribution is the joint probability for the calculation of the border minimum threshold value to perform the utility mining.

The organization of the paper is given as follows: Section 1 gives the introduction to the paper, section 2 depicts the literature works of the paper. The proposed work is discussed in section 3 and the method is analyzed based on the metrics as shown in the section 4, and finally, section 5 gives the conclusion of the paper.

## 2. Motivation

### 2.1 Literature Review

Vincent S. Tseng *et al.* [1] proposed Mining of high utility itemset by AprioriHC-D, AprioriHC, and Closed + High Utility Itemset Discovery (CHUD) algorithms. The merit of the method was that it exhibited the faster execution time, but the challenge of the methods was based on the huge number of the candidates. Unil Yun and Donggyu Kim [2] introduced a strategy named as list structure and pruning strategy that mines high average-utility itemsets. The memory usage and runtime was better, but the Sorting processes may require massive execution time. Jerry Chun-Wei Lin *et al.* [3] proposed a baseline two-phase Short-Period High-Utility Itemset (SPHUI) mining algorithm that mined SPHUIs. The scalability of the algorithm was better in case of the large-sized database, but this method did not mine more specific patterns. Quang-Huy Duong *et al.* [4] proposed Mining top- k high utility itemsets using a top- k High utility itemset Mining using Co-occurrence pruning (kHMC) algorithm and the method possess less runtime, and memory consumption. However, the method was not applicable for closed and maximal HUI mining.

### 2.2 Challenges

- ✓ HUI mining is a hectic challenge as a result of the downward closure property, and the presence of the high utility itemsets poses a challenge to the users to obtain the results. Additionally, the algorithms appear inefficient with respect to memory and time [9].
- ✓ The analysis using the mining algorithm conveys that the performance of mining reduces when the utility thresholds are less and inapplicable to deal with dense databases [1].
- ✓ Dealing with the huge number of the itemsets is a challenge as mining large itemsets is difficult and the method should aim at mining less number of highly useful itemsets [4].
- ✓

## 3. Proposed methodology: Joint probability based minimum utility threshold for mining top-k HUIs

Mining of the itemsets is applicable for analyzing the items of user preference, and the analysis is based on the transaction details. Initially, the analysis is carried out using the input database that consists of the transaction details of different users. The transaction details of the individual user consist of the items they preferred, IU of the items, and the transaction utility of the users. The transaction details of the user vary from other users, and TU is computed by adding the product of the internal and the external utility values of the items in that transaction. Once the TU is computed, the TWU is computed based on the sum of the TU of the transactions that possess the particular item. Based on the TWU, the transaction table is reordered to give the reorganized transactions that are employed to establish the UP-Tree, and the mining is based on the threshold, named as the border minimum utility value. The proposed computation of the border minimum utility value is performed using the joint probability values. The computation of the border minimum utility value depends on the TWU, IU, and EU. Figure 1 shows the proposed method of Joint probability based minimum utility threshold.

### 3.1 Developing the UP-Tree

The simple ways of developing the UP Tree [1] is depicted in this section. The original database is scanned twice to develop the UP-Tree, and the result of the scanning process is the construction of TU of individual transaction and TWU of the individual item. The transaction utility table consists of the details of the TU and items-based information of individual transactions that contains the internal utility of the individual item corresponding to the individual transaction. IU is the measure of the frequency of the particular utility item. The external utility table of the items holds the profit-based item information. Thus, the TU table comprises of the TU of the individual transaction that is based on the sum of the products of the internal and external utility of individual utility items. In the second step, the TWU is computed based on the TU of the items. The TWU of the individual utility item is the sum of the TU of the individual items in the transactions. Once the TWU is determined, the utility items of the individual transaction are ordered in the ascending order of their TWUs. Thus, the reorganized transactions are added in the UP-Tree, and the transaction after

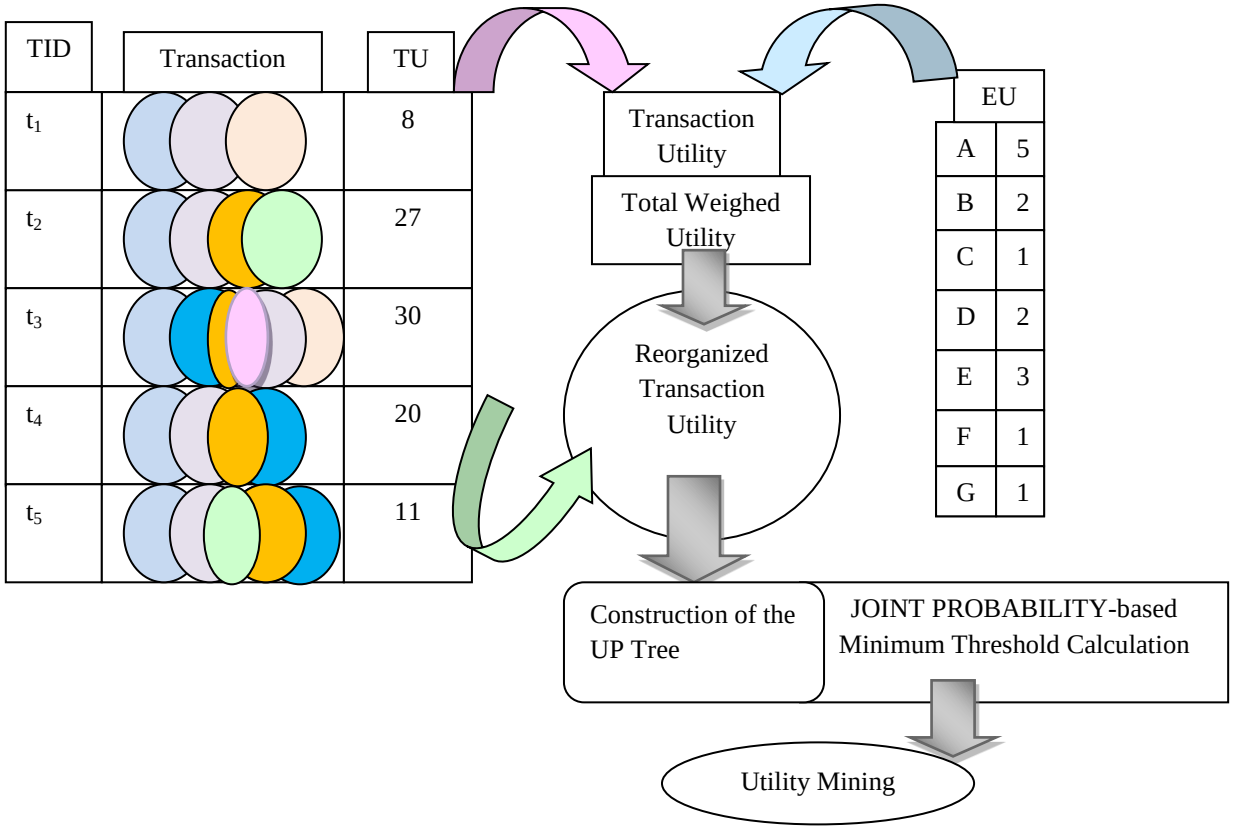


**Figure 1.** Joint probability based minimum utility threshold for mining top-k HUIs

reorganizing is named as the reorganized transaction utility. The UP-Tree is progressed with a root, and the function $\text{Insert\_Reorganized\_Transaction}$ is employed for adding a node in the UP Tree, and the inputs to the UP Tree are the Node and the item. The process of constructing the UP Tree is given as below:

i) When a node posses a child node $\text{Child}(D)$ denoting that $\text{Child}(D).\text{Item} = T_i$ then, add the count of the node by 1 or otherwise, establish a new node such that $\text{Child}(D).\text{Item} = T_i$; $\text{Child}(D).\text{count} = 1$, $\text{Child}(D).\text{Parent} = D$, and $\text{Child}(D).\text{NN} = 0$.

ii) Increment $\text{Child}(D).\text{NN} = 0$ using $\text{RTU}(t') - \sum_{k=i+1}^{G} \text{EU}(t_K, t')$, where $G$ indicates the total number of items present in the transaction, and $t'$ refers to the reorganized transaction and there is a number of transactions.

iii) Read the function, $\text{Insert\_Reorganized\_Transaction}(\text{child}(D), T_{i+1})$ if $i \le G$.

Once all the available reorganized transactions are inserted, the UP Tree is constructed.

### 3.2 Generation of the Potential top-K High Utility Itemsets (PKHUIs) using the UP-Tree

The main aim of this section is to depict the calculation of the minimum utility threshold. The algorithm, termed as, $\text{TKU}_{\text{base}}$, is the existing algorithm that is the modified UP Growth algorithm [18] aiming at mining the HUIs. The $\text{TKU}_{\text{base}}$ algorithm [8] uses a parameter, termed as the border minimum utility threshold, which is initially set to zero and then, increased dynamically when the itemsets of higher utilities are generated during the establishment of the PKHUIs. The existing method of setting the minimum utility threshold takes complex computations and takes much time. Upon the arrival of the candidate itemset, the $\text{TKU}_{\text{base}}$ algorithm follows the UP Growth instruction to verify if the estimated utility value lies below the minimum utility threshold. In case, if the estimated utility value is less then, the candidate itemset and its associates do not form the top-k HUIs. On the other hand, $\text{TKU}_{\text{base}}$ clarify whether the maximum utility of an item lies below the border minimum utility threshold, if yes, the candidate

itemset does not become the top-k HUI. If these two conditions mentioned-above fail, the candidate itemset is added in the next phase. The candidate dataset is a valid PKHUI if and only if the maximum utility value exceeds the border minimum utility threshold. The purpose of the maximum utility value is that it varies the border minimum utility threshold.

The effective update of the border minimum utility threshold is enabled using the min-heap structure, termed as TopK-MIU-List, in the existing work [1]. The TopK-MIU-List sustains the $k$ highest Minimum Utility (MIU) values of PKHUIs. The PKHUI of the candidate itemset is determined each time, and the candidate itemset is added to the top-k MIU, in case the maximum utility is found to be greater than the border minimum utility threshold. The border minimum utility threshold remains the same until the $MIU$ values are less than $k$ -MIU in the TopK-MIU list and the border minimum utility threshold is updated as $k$ -MIU value if the $k$ -MIUs in the TopK-MIU list exceeds the order minimum utility threshold. The algorithm searches until there is no candidate.

### 3.3 Proposed Idea of calculating the minimum utility based on Joint Probability

The existing method of defining the minimum utility is complex and consumed large time, and hence, a simple method of computing the minimum utility is proposed. The proposed method of computing the minimum utility is based on the Joint probability, for which initially, a table is developed using the TWU, IU, and EU values of the individual item. Then, the probability of the individual item is computed and is employed for calculating the minimum threshold value. The steps involved in the computation of the minimum utility threshold are given as,

*Formation of the Computation table:* The computation of the minimum utility threshold is progressed using the computation table that consists of the values of TWU, $\sum IU$ , and EU. TWU is computed in the initial steps before constructing the UP Tree and $\sum IU$ is obtained by summing the internal utility of individual items corresponding to the transactions present in the transaction database.

*Computation of the Probability values:* The probability values are computed based on the EU value of the utility item. The frequency of EU is inferred for the computation of the probability, and it is the ratio of the number of frequencies in the values of EU to the total number of the items present in the transaction database.

*Calculation of the minimum utility threshold using the Joint Probability:* The Joint probability of an item is computed as the sum of the probabilities computed in the above step. The joint probability value is set as the minimum threshold to perform utility mining that ensures the simple and effective analysis.

Table 1 shows the computation of the minimum threshold based on the Joint probability. The running example in [1] is inherited in this paper, and the computation of the minimum utility threshold using the Joint probability is progressed as follows:

**Table 1.** Calculation of Minimum threshold value

| Items | TWU | $\sum IU$ | EU |
|---|---|---|---|
| C | 96 | 13 | 1 |
| E | 88 | 5 | 3 |
| A | 65 | 4 | 5 |
| B | 61 | 8 | 2 |
| D | 58 | 10 | 2 |
| G | 38 | 7 | 1 |
| F | 30 | 5 | 1 |

As shown in table 1, let us consider a database that consists of the items A to G with their corresponding TWU values, $\sum IU$ , and External Utility (EU) values. The TWU and IU values are taken from [1]. The TWU calculation is based on the sum of the TU of the items in the individual transactions. IU refers to the frequency of the items in the individual transaction and $\sum IU$ is computed by summing the frequency of an item in all the transactions present in the transaction database. EU refers to the utility given based on the profit of an item depending on any other product. Thus, the proposed Joint probability-based calculation is given as,

$$P(1) = \frac{3}{7} = 0.42$$

$$P(2) = \frac{2}{7} = 0.28$$

$$P(3) = \frac{1}{7} = 0.14$$

$$P(5) = \frac{1}{7} = 0.4$$

The probabilities are based on the EU of the itemset, and the frequency of the EU is inferred to compute the probability. Once the probability is determined then, the joint probability, which is the sum of the product of the $\sum IU$ value of the individual item and probability values, is computed. Thus, the joint probability is computed as,

$$JP = [(13*0.42)+(5*0.140)+(4*0.4)+(8*0.28)+(10*0.28)+(7*0.42)+(5*0.42)]$$

$$JP = 17.84$$

The value of the border minimum utility threshold is, $JP = 17.84$ and the steps engaged in the computation of the minimum utility threshold is simple compared to the existing algorithm. Based on the minimum utility computed using the proposed joint probability, the PKHUIs can be mined effectively.

## 4. Results and Discussion

The section depicts the results and discussion with the elaborate comparative analysis to prove the effectiveness of the developed method.

### 4.1 Experimental setup

The experimentation is performed using the Windows 8 OS with 4GB RAM, and the implementation is performed using the JAVA.

### 4.2 Dataset description

The dataset employed for analysis includes the retail, chess, and T10I4D100K datasets taken from the Frequent Itemset Mining Dataset Repository [22].

### 4.3 Performance Metrics

The performance of the proposed method is analyzed using the metrics, such as the number of item sets mined and the time. The effective method offers less time for effectively performing the mining.

### 4.4 Comparative Methods

The methods taken for comparison include FP Growth [23] and mining Top-K Utility itemsets (TKU) [8] for frequent data mining, and the existing methods are compared with the proposed method.

### 4.5 Comparative Analysis

The section depicts the comparative analysis of the proposed method based on the performance metrics.

#### *4.5.1 Using Retail Dataset*

Figure 2 shows the comparative analysis using the retail dataset in terms of the performance metrics. Figures 2 a) and 2b) demonstrate the performance analysis of the methods in terms of the number of patterns mined and time. The analysis is performed based on the percentage of the data. When the percentage of the data is 40%, the patterns mined using the methods, like FP Growth, TKU, and proposed method are 60, 57, and 40, respectively, which indicates that the proposed method outperforms the existing methods. Similarly, the time taken by the proposed method for the 40% of data in frequency mining is less, taking a time of 26secs, which is low compared with the existing methods, FP Growth, and TKU, which acquire the time of 71secs and 42secs, respectively. The analysis proves that the proposed method is effective over the other existing methods.

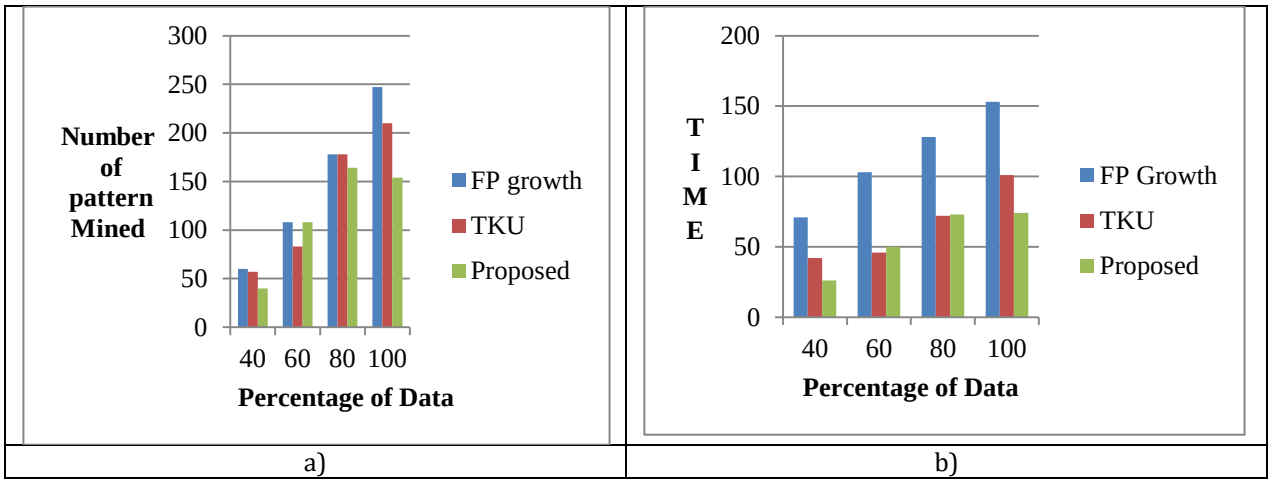


**Figure 2.** Performance Analysis using the Retail dataset based on the a) number of the patterns mined b) time

### *4.5.2 Using Chess Dataset*

Figure 3 shows the comparative analysis using the chess dataset in terms of the performance metrics. Figures 3 a) and 3 b) demonstrate the performance analysis of the methods based on the patterns mined and time. The performance analysis is carried out with respect to the varying percentages of the data. When the data is 100%, the data mined using the methods, like FP Growth, TKU, and the proposed method are 82431, 82431, and 133, respectively. The results indicate the improved performance of the proposed method. Similarly, the time taken by the proposed method for the 100% of data in frequency mining is less with a value 168secs, which is low compared with the existing, FP Growth, and TKU methods, that acquire the time of 1680secs and 2111secs, respectively. The analysis proves that the proposed method is effective over the other existing methods.

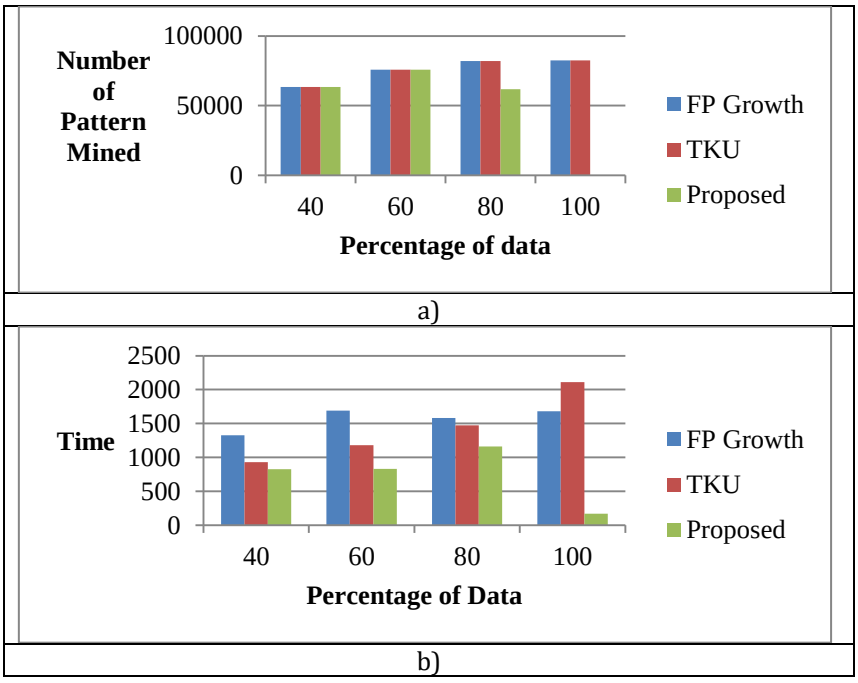


**Figure 3.** Performance Analysis using the chess dataset based on the a) number of the patterns mined b) time

### *4.5.3 Using T10I4D100K dataset*

Figure 4 shows the comparative analysis using the T10I4D100K dataset in terms of the two performance metrics. Figure 4 a) demonstrates the performance analysis of the comparative methods based on the patterns mined and time. Figure 4 b) presents the performance analysis based on the time. The analysis is performed based on the percentage of the data. When the percentage of the data is 40%, 60%, 80%, and 100%,

the data mined using the methods, FP Growth, TKU, and the proposed method are 116, 264, 2529, and 2663, respectively, indicating that the performance of the proposed method is maximum compared with the existing methods. Similarly, the time taken by the proposed method for the 80% of data in frequency mining is 221secs, which is low compared with the existing, FP Growth, and TKU, which acquire the time of 239secs and 422secs, respectively. The proposed method seems to be effective from the aforementioned analysis.

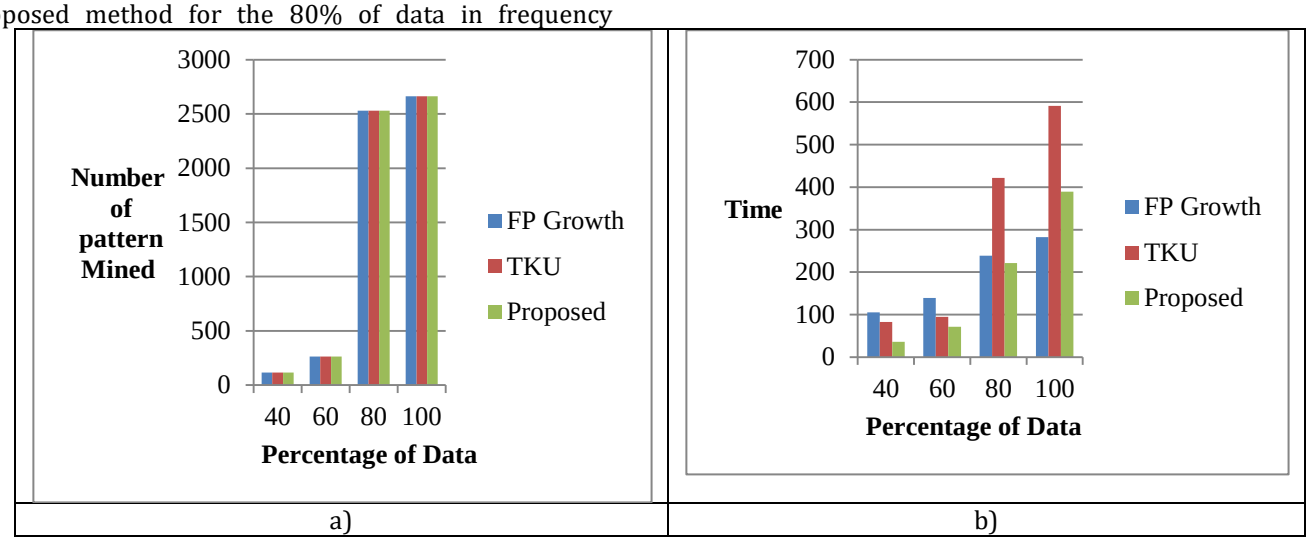


**Figure 4.** Performance Analysis using the T10I4D100K dataset a) Based on the number of the patterns mined b) Based on the time

## 5. Conclusion

The paper proposes a joint probability function for evaluating the minimum threshold for performing the utility mining. Normally, the existing method develops the UP-Tree out of the data given and computes the minimum threshold border based on the TKU algorithm that takes messy computation, and it seems to be time-consuming. Thus, the effectiveness of computing the minimum threshold is enhanced using the proposed method that concentrates on the Joint probability measure for calculating the minimum threshold. The dataset employed for the utility mining is based on two parameters namely, the TU and TWU, which in turn, depends on IU and EU of the individual itemsets belonging to the dataset. Followed by the computation of TU and TWU, the UP-Tree is developed laying a foundation for computing the Minimum threshold value that is based on the Joint Probability. The experimentation is performed using three datasets, and the comparative analysis is carried out to prove the superiority of the proposed method. The analysis in terms of the total data mined and the time taken for mining proves the effective performance and the proposed method acquired a better mining performance of 133, taking a minimum time of 74secs.

## References

[1] Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, "Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets," IEEE Transactions on knowledge and data engineering, vol. 27, no. 3, pp. 726-739, March 2015.

[2] Unil Yun and Donggyu Kim , "Mining of high average-utility itemsets using novel list structure and pruning strategy", Future Generation Computer Systems, vol. 68, pp.346–360, March 2017.

[3] Jerry Chun-Wei Lin, Jiexiong Zhang, Philippe Fournier-Viger, Tzung-Pei Hong, and Ji Zhang, "A two-phase approach to mine short-period high-utility itemsets in transactional databases," Advanced Engineering Informatics, vol. 33, pp. 29–43, August 2017.

[4] Quang-Huy Duong, Bo Liao, Philippe Fournier-Viger, and Thu-Lan Dam, "An efficient algorithm for mining the top- k high utility itemsets, using novel threshold raising and pruning strategies," Knowledge-Based Systems, vol. 104, pp. 106–122, 15 July 2016.

[5] Unil Yun, Heungmo Ryang, and Keun Ho Ryu, "High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates," vol. 41, no. 8, pp. 3861-3878, June 2014.

[6] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, pp.1772-1786, August 2013.

[7] Guo-Cheng Lan, Tzung-Pei Hong, and Vincent S. Tseng, "An efficient projection-based indexing approach for mining high utility itemsets," Knowledge and Information Systems, vol. 38, no. 1, pp. 85–107, January 2014.

[8] Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu, "Efficient Algorithms for Mining Top-K High Utility Itemsets," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 1, pp. 54-67, January 2016.

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

[10] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," Data Mining and Knowledge Discovery, vol. 8, no. 1, pp. 53–87, January 2004.

[11] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules," in Proceedings of the 7th International Conference on Database Theory, pp. 398-416, January 1999.

[12] G. Lee, U. Yun, and K. Ryu, "Sliding window based weighted maximal frequent pattern mining over data streams," Expert Systems with Applications, vol. 41, no. 2, pp. 694–708, February 2014.

[13] G. Lee, U. Yun, H. Ryang, and D. Kim, "Approximate maximal frequent pattern mining with weight conditions and error tolerance," International Journal of Pattern Recognition and Artificial Intelligence, vol. 30, no. 06, pp.1–42, July 2016.

[14] 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, December 2009.

[15] R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," In Proceedings of third IEEE International Conference on Data Mining, pp. 19–26, 2003.

[16] A. Erwin, R. P. Gopalan, and N. R. Achuthan, "Efficient mining of high utility itemsets from large datasets," in Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 554-561, 2008.

[17] H.F. Li, H.Y. Huang, Y.C. Chen, Y.J. Liu, and S.Y. Lee, "Fast and memory efficient mining of high utility itemsets in data streams," in Proceedings of the IEEE International Conference on Data Mining, pp. 881–886, 2008.

[18] V. S. Tseng, C.W. Wu, B.E. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining, " in Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 253–262, 2010.

[19] C.W.Wu, B.E. Shie, V. S. Tseng, and P. S. Yu, "Mining top-k high utility itemsets, " in Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 78–86, 2012.

[20] B.E. Shie, H.F. Hsiao, V. S. Tseng, and P. S. Yu, "Mining high utility mobile sequential patterns in mobile commerce environments, " in Proceedings of the International Conference on Database Systems for Advanced Applications, vol. 6587, pp. 224–238, 2011.

[21] A. Erwin, R. P. Gopalan, and N. R. Achuthan, "Efficient mining high utility itemsets from large datasets," in Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Advances in Knowledge Discovery and Data Mining, pp. 554–561, 2008.

[22] Frequent Itemset Mining Dataset Repository, "http://fimi.ua.ac.be/data/", Accessed on 05 December 2017.

[23] Han J, Pei J, Yin Y, Mao R, "Mining frequent patterns without candidate generation: a frequent pattern tree approach", Data Mining and Knowledge Discovery, vol.8, no.1, pp.53–87, 2004.