International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

An Improved Approach For High Utility Itemset Mining Using Length Reduction Method

Afrin Shaikh[1], Vishal Shah[2]

[1]M.E Student, Department of Computer Engineering, Sardar Vallabhbhai Patel Institute of Technology, VASAD, India

[2]Assistant Professor, Department of Computer Engineering, Sardar Vallabhbhai Patel Institute of Technology, VASAD, India

ABSTRACT

Data mining is process of analyzing data from different data repository and mine Useful and interesting patterns from them. It is also defined as the use of algorithm to discover hidden patterns and interesting relationship between large itemset. High utility itemset mining is an area research where utility based mining can be done. Mining high utility itemset from a transactional database refers to the discovery of itemset with high utility in a terms like weight, unit profit or value. High-utility item set mining (HUIM) is an important data mining task that refers to the set of items with high utility like profit in a customer transaction database. However an Important issue with traditional HUI mining algorithm is that they tend to find itemset having many items which increases memory and time overhead. To discover HUIs efficiently with length constraints, FHM+ introduced the concept of estimated utility co-occurrence structure (EUCS) and two Length Upper Bound Reduction (LUR) of itemset. EUCS has matrix structure and in that half of the matrix is not filled with data so it has memory overhead.. In this paper, we address this issue by presenting an improved algorithm based on tree data structure which can decreases the execution time and memory usage for HUI mining.

Keywords: High Utility, HUI_Miner, Transactional Database, Transaction Weight Utilization (TWU), FHM

I. INTRODUCTION

Data mining is the process of revealing nontrivial, previously unknown and potentially useful patterns form large database Data mining can be used to transform the data into meaningful and useful information for business analysis processes referred to as Business intelligence. High utility mining refers to finding an itemset with “high profit” in transaction through tree data structure.

High utility item set mining is popular data mining task it consist of enumerating all high utility item set (HUIs) groups of items (itemset) having a high utility in customer transaction databases . HUIM is generalization of the problem of frequent itemset mining (FIM) where item can appear more than once in each transaction and where each items can appear more than once in each transaction and where each item has a weight (eg. Unit profit) . High Utility itemset mining is widely viewed as more difficult than frequent item set mining because the utility measures used in HUIM is neither anti-monotonic nor monotonic high utility itemset may have supersets or subsets having lower, equal or higher utilities. HUIM has a wide range of application. However an important issue of traditional HUIM algorithm is that they tend to find itemset containing many items, as they are more likely to have a high utility. This is an issue because itemset containing many items are generally less useful than itemsets containing fewer itemsets containing fewer items [2].

Jerry Chun-Wei lin (2016) proposed a novel algorithm named FHM+ for mining HUIs, while considering length constraints to discover HUIs efficiently with length constraints FHM+ introduce the concept of Length Upper Bound Reduction (LUR) and two novel upper-bound on the utility of itemsets.an extensive experimental evaluation shows that length constraints are effective at reducing the number of patterns, and the novel upper-bound can greatly decrease the execution time, and memory usage for HUI mining. Moreover result shows that LUR concept greatly improves the algorithm efficiency. Thus prune the search space an extensive experimental evaluation shows that the proposed algorithm can be much faster than the state-of-the-art FHM algorithm. And greatly reduce the number of patterns presented to the user.[2] Previous system is used the Length Upper bound Reduction based algorithms Which results in a large time and memory consumption. Existing algorithm are based on matrix structure. Which use a matrix structure to find all utility itemset. But problem is that it will be generate large number of candidate key and are consume large memory overhead.

Example:

Example of a transaction database representing the sales data and the profit associated with the sale of each unit of the item

Table 1. Transaction Database

INPUT:

TID	TRANSACTION
T1	(a,1), (b,5),(c,1),(d,3),(e,1),(f,5)
T2	(b,4),(c,3),(d,3),(e,1)
T3	(a,1),(c,1),(d,1)
T4	(a,2),(c,6),(e,2),(g,5)
T5	(b,2),(c,2),(e,1),(g,2)

Table 2. Utility profit database[2]

Item	A	B	C	D	E	F	G
Profit	5	2	1	2	3	1	1

minutil: A minimum utility threshold set by the user (a positive integer)

TU (T1) = 5*1+2*5+1*1+2*3+3*1+1*5=30

TU (T2) =2*4+1*3+2*3+3*1=20

TU (T3) =5*1+1*1+2*1=8

TU (T4) = 5*2+1*6+3*2+1*5=27

TU (T5) = 2*2+1*2+3*1+1*2=11

Table 3. Total Utility[2]

Item	Total Utility
T1	30
T2	20
T3	8
T4	27
T5	11

OUTPUT: All high-utility itemsets (itemsets having a utility≥*minutil*)

For example, if minutil = 33$, the high-utility itemsets are:

$$u(\{b,d,e\}) = \underbrace{(5x2)+(3x2)+(3x1)}_{\text{utility in transaction t1}} + \underbrace{(4x2)+(2x3)+(1x3)}_{\text{utility in transaction t2}} = 36\$$$

Table 4

{b,d,e} 36$ 2 transactions	{b,c,d} 34$ 2 transactions
{b,c,d,e} 40$ 2 transactions	{b,c,e} 37 $ 3 transactions

II. RELATED WORK

Jerry Chun-Wei lin (Jerry chun. 2016) proposed a novel algorithm named FHM+ for mining HUIs, while considering length constraints to discover HUIs efficiently with length constraints FHM+ introduce the concept of Length Upper Bound Reduction (LUR) and two novel upper-bound on the utility of itemsets.an extensive experimental evaluation shows that length constraints are effective at reducing the number of patterns, and the novel

upper-bound can greatly decrease the execution time, and memory usage for HUI mining. Moreover result shows that LUR concept greatly improves the algorithm efficiency. Thus the search space an experimental evaluation shows that the proposed algorithm can be much faster than the FHM algorithm. And reduce the number of patterns presented to the data.[2] Previous system is used the Length Upper bound Reduction based algorithms Which results in a large time and memory consumption. Existing algorithm are based on matrix structure. Which use a matrix structure to find all utility itemset. But problem is that it will be generate large number of candidate key and are consume large memory overhead.

Tseng et al. (Viscent 2014) proposed a novel strategy based on the analysis of item co-occurrences to reduce the number of join operations that need to be performed . an extensive experimental study with four real-life datasets shows that the resulting algorithm named FHM(Fast High Utility Miner) reduce the number of join operations by up to 95% and is up to six times faster than the state-of-art algorithm HUI-Miner. Frequent Itemset Mining (FIM) is popular data mining task that is essential to a wide range of application given a transaction database FIM consist of discovering frequent itemset as group of item appering frequently in transaction[1]

Frequent Itemset Mining (FIM) is popular data mining task that is essential to a wide range of application . given a transaction database FIM consist of discovering frequent itemset as group of item appering frequently in transaction[1]. However an important limitation of FIM is that it assumes that each item cannot appear more than once in each transaction and that all item have the same importance (weight, unit profit or value).

These assumptions often do not hold in real applications the problem of HUIM is widely recognized as more difficult than the problem of FIM in FIM the downward-closure property states that the support of an itemset is anti-monotonic, that is the supersets of an infrequent itemset are infrequent and subset of a frequent itemset are frequent. The property is very powerful to prune the search space.

Souleymane Zidal (Zidal 2015) proposed a novel algorithm named EFIM (EFficient high-utility Itemset Mining), which introduces several new ideas to more efficiently discovers high-utility itemsets both in terms of execution time and memory. EFIM relies on two upper-bounds named sub-tree utility and local utility to more effectively prune the search space. It also introduces a novel array-based utility counting technique named Fast Utility Counting to calculate these upper-bounds in linear time and space. Moreover, to reduce the cost of database scans.

The proposed utility mining approach it integrated the two-phase tree structure to efficient find high utility itemset patterns . tree structure is designed and tree-construction describe below.

III. PROPOSED ALGORITHM

The High Utility Tree construction algorithm is first proposed to keep the high utility items found from a database in the tree structure based on the downward-closure property. The proposed algorithm first calculates the transaction utility of each transaction. Then finds the transaction-weighted-utilization values of all the items. If the transaction-weighted-utilization of an item is larger than or equal to the predefined minimum utility threshold, it is thus well-thought-out as a high transaction-weighted 1-itemset. The algorithm then keeps only the high transaction-weighted 1-item-sets in the transactions and sorts them according to their transaction frequencies. The updated transactions are then used to build the HUP tree tuple by tuple, from the first transaction to the last one. Each node in the tree has to store the transaction-weighted-utilization

of the item as well as the quantities of its preceding items (including itself) in the path. An array is then attached to a node to keep those values. The Flow diagram of proposed algorithm is show below.

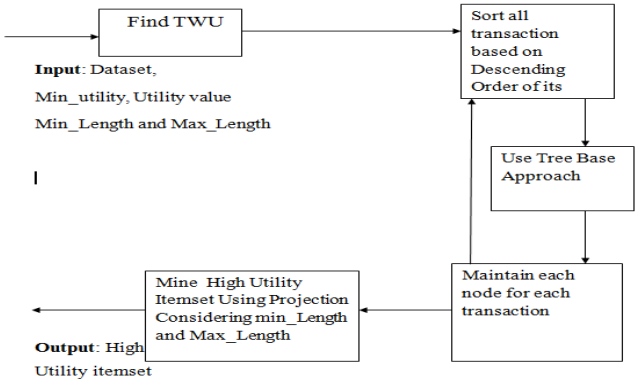


Figure 1. Proposed System

Firstly it takes dataset, utility value and utility threshold as an input and find total weighted utility (TWU) of each item, and checks it with the user threshold. If it is smaller than user threshold, then this items will be ignored. After checking it stores all the items in descending order according to their TWU. Now after that it use tree structure to mine high utility itemset.

Algorithm

Step 1: Scan Database and find TWU and TU of each item and transaction respectively.

Step 2 : Sort all transactions items based on its TWU in descending order and if

TWU(item) < min_utility ignore that item.

Step 3: For each transaction

If there is no node in tree, create node and add below values in node

i) Item

ii) Transaction number

iii) TU of item

else

Traverse into tree to insert a item into node and then add above all three values of item.

Step 4: Do projection of each item into tree and find high utility itemset from projection

tree. (To get high utility itemset, compare each item's path into tree and sum their utility count considering min_length and max_length)

If (sum_utility_count ≥ min_utility)

then

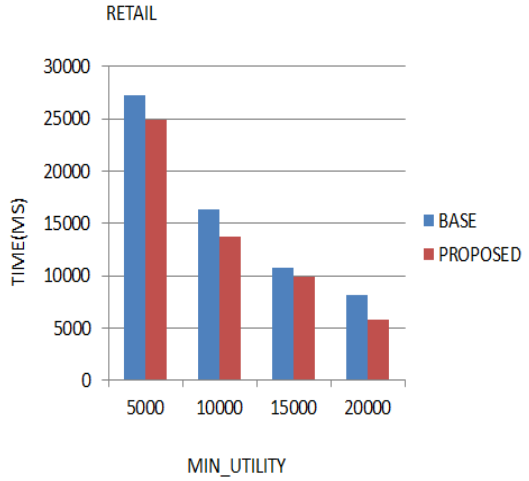
Itemset is highutility itemset. Print itemset in output file.

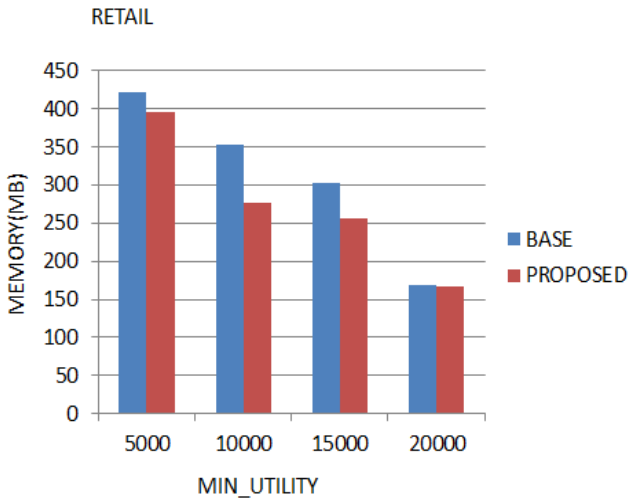
IV. EXPERIMENTAL STUDY

We performed experiments to assess the performance of the proposed algorithm. Experiments were performed on a computer with a third generation 64 bit core i5 processor running on Windows 7 Os and having 4 GB of free RAM. We compared the performance of proposed algorithm with the FHM algorithm for high utility itemset mining. All memory measurements were done using the Java API. Experiments were carried on four real-life dataset having different characteristics. The Retail dataset contains 59,601 transactions with 497 distinct items and an average transaction length of 4.85 items. The Kosarak dataset contains 9, 90,000 transactions with 41,270 distinct items and having an average length of 8.09 items. The Chainstore dataset contains 8,124 transactions with 119 distinct items having an average length of 23.0 items. The Foodmart dataset contains 4,141 transactions with 1559 distinct items having an average length of 4.4 items.

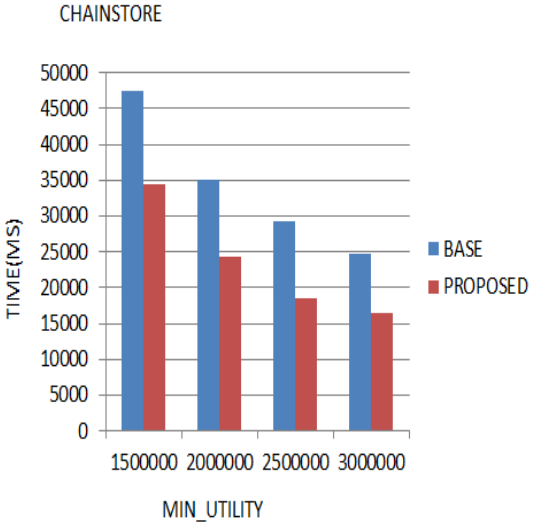
Execution Time. We first run the FHM and Proposed algorithms on each dataset while decreasing the minutil threshold until algorithms became too long to execute, run out of memory or a clear winner observed. For each dataset we recorded the execution time, the total high utility itemset count and total memory overhead. The comparison of execution times is shown in Fig 2. For Retail, Kosarak, Mushroom and Foodmart database proposed algorithm was respectively up to 5.1 times faster, 2.9 times faster, 2.1 times faster, and 3 times faster than FHM.

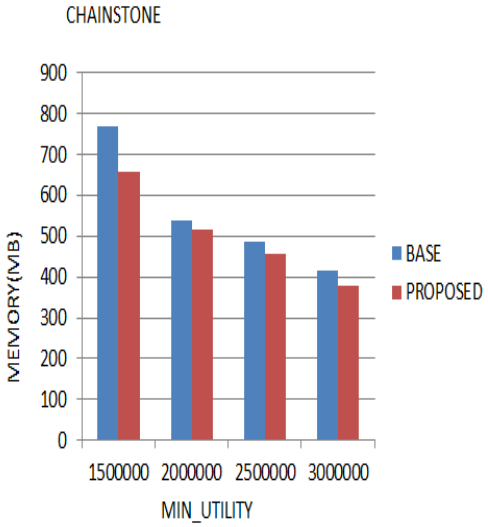
a) RETAIL



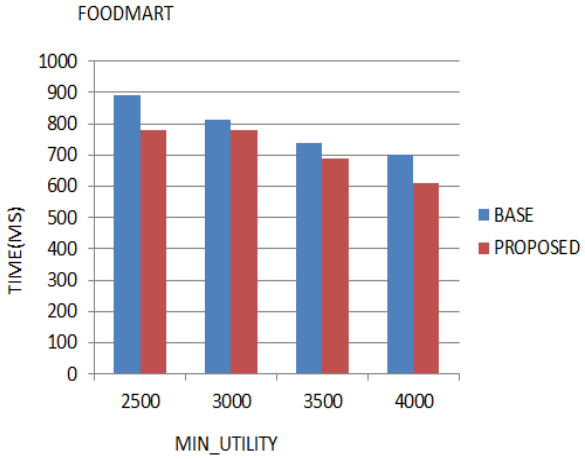


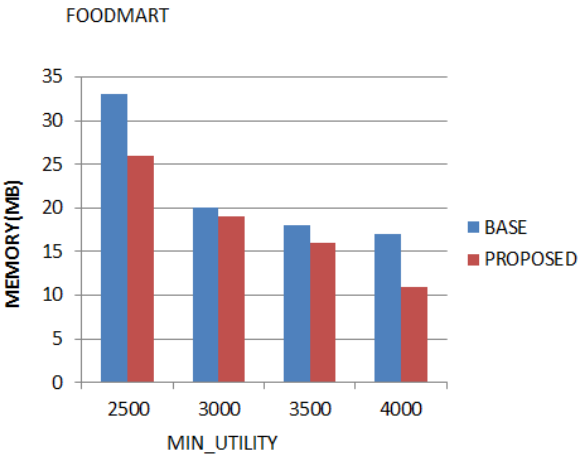
b) CHAINSTORE





C) FOODMART





d) KOSARAK

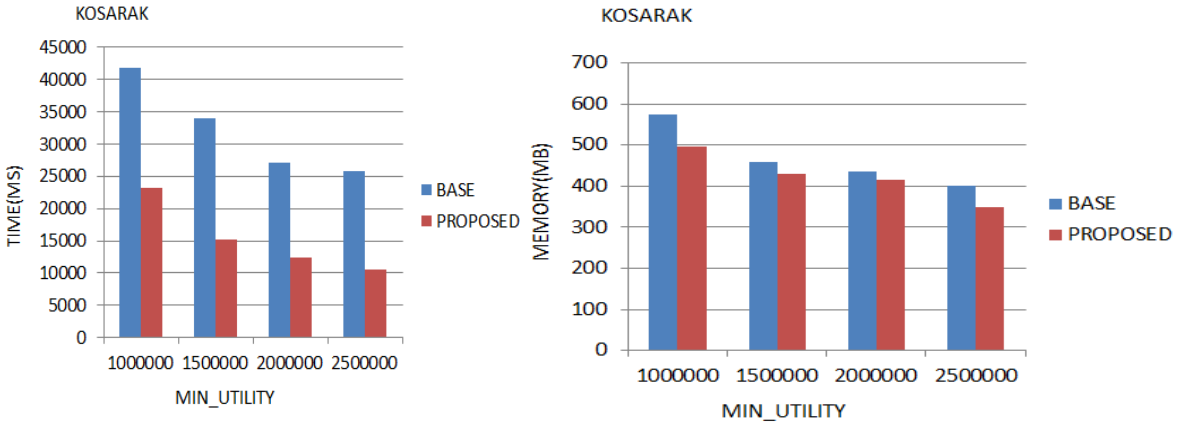


Figure 2. Execution Time and Memory Overhead For Base and Proposed Algorithm

Memory Overhead. We also studied the memory overhead of using the tree structure. We found that the Retail, Kosarak, Foodmart, and Chainstore datasets, the memory usage of our proposed algorithm was respectively 166 MB, 495MB, 26 MB, and 378 MB.

V. CONCLUSION

This Paper shows an efficient algorithm named FHM+ to discover high utility itemset and considering the length constraint also shows an tree structure algorithm proposed and through this execution time reduced Tree Structure is little like FP-Growth tree but it stored the transaction id and its utility value in a different array structure same time when tree is generate. We also proposed an algorithm to effectively mine high utility itemset from the proposed tree structure. High utility item sets can be derived effectively and efficiently from tree structure. Experimental result shows that our proposed algorithm performs faster than FHM+ algorithm and it occupies less memory and execution time than FHM+.

In this paper we assume that database is static, but in real life data will be inserted into or deleted from a database. In future we will attempt to handle incremental database where transactions are inserted, deleted or modified and worked through large dataset.

VI. REFERENCES

[1]. Philippe Viger, Cheng Wu, Souleymane Zida, Vincent S. Tseng- "FHM: Faster High Utiltiy Mining Itemset mining Using Estimated Utility Co-occurrence purning." , Springer International, Switzerland 2014,ISMIS 2014.

[2]. Philippe Fournier-Viger, Jerry Chun-Wei Lin,Quang-Huy Duong, Thu-Lan Dam, "FHM+: Faster High-Utility Itemset Mining using Length Upper-Bound Reduction.", Springer International Publishing Switzerland 2016.

[3]. Souleymane Zida, Philippe Fournier-Viger(B), Jerry Chun-Wei Lin, Cheng-Wei Wu, and Vincent S. Tseng- "EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining.", Springer International Publishing Switzerland 2015.

[4]. Sunidhi Shrivastava, Punit Kumar Johari " Analysis on High Utility Infrequent ItemSets Mining Over Transactional Database.", IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May, 2016.

[5]. Menghchi Liu, Junfeng Qu-"Mining High Utility Itemsets Without Candidate Generation", CIKM, Maui. USA, Nov 2012.

[6]. Shiming Guo, Hong Gao- "An Efficient Algorithm For Incremental and Interactive High Utility Itemset Mining" ,International Conference on Image, 2017.

[7]. Jerry Chun-Wei Lin, Wensheng Gan, Philippe Fournier-Viger, Tzung-pei Hong "Mining High-Utility Itemsets with Multiple Minimum Utility Thresholds",C3S2E 2015.

[8]. Thang Mai, Bay Vo, Loan T.T. Nguyen "A Lattice-based approach for mining high utility association rules", Information science Elsevier,2017.

[9]. Vincent S. Tseng, Bai-En Shie, Cheng Wu, philip S. Yu- "Efficient Algorithms For Mining High Utility Itemset from Transactional Databases.", IEEE transactions on knowledge and data engineering (Vol 25, no. 8), Aug 2013, ISSN No: 1041-4347, DOI: 10.1109/TKED.2012.

[10]. Jerry Chun-Lin, Member, IEEE, Shifeng Ren, Philippe Fournier-Viger Tzung-Pei Hong,"EHAUPM: Efficient High Average-Utility Pattern Mining with Tighter Upper-Bounds"IEEE 2016

[11]. Junqiang Liu, Member, IEEE, Ke Wang, Senior Member, IEEE, and Benjamin C.M. Fung, Senior Member, IEEE " Mining High Utility Patterns in one phase without generating candidates" IEEE 2015.

[12]. Philippe Fournier-Viger, Jerry Chun-Wei Lin, Ted Gueniche, Prashant Barhate, "Efficient Incremental High Utility Itemset Mining" ACM 2015.

[13]. Guo-Cheng Lan , Tzung-Pei Hong , Vincent S. Tseng "An efficient projection-based indexing approach for Mining High Utility Itemsets" Springer 2013.

[14]. Supachai Laoviboon, Komate Amphawan, "Mining High-Utility Itemsets with Irregular Occurrence" IEEE 2017.

[15]. Wei Song, jaipei Xu "Discovering High Utility Itemset Using Map Reduce" IEEE 2016.

[16]. P.Payal Swamy, Amit Pimpalkar " Improving method for graphical Analysis and representation of high utility itemsets using UP++ Growth. IEEE 2016.