2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal

AN EFFICIENT HIGH UTILITY ITEMSETS BASED ON SECTIVE DATABASE PROJECTIONS

[#1]D.BHAVYA REDDY, [#2]Dr. B.HARI BABU

[1]M.TECH STUDENT, DEPARTMENT OF EEE, KAKINADA INSTITUTE OF TECHNOLOGICAL SCIENCES (KITS), RAMACHANDRAPURAM

[2]ASSISTANT PROFESSOR, DEPARTMENT OF EEE, KAKINADA INSTITUTE OF TECHNOLOGICAL SCIENCES (KITS), RAMACHANDRAPURAM.

ABSTRACT: Data mining is used to discover interesting and useful knowledge from massive data. Finding interesting patterns play an important role in knowledge discovery process and are essential for many real life applications. Recently high utility pattern mining is important for mining high utility itemsets which overcomes the limitation of frequent pattern mining. High utility pattern mining is used to identify the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. This research paper proposes an enhanced high utility pattern approach to mine the high utility itemsets with less computation time and less memory space when larger itemsets are explored for complex datasets.

Keywords— Data mining, frequent patterns, high utility pattern mining, high utility pattern mining algorithm, high utility itemset mining performance.

I. INTRODUCTION

Frequent itemsets mining (FIM) is one of the fundamentals of data mining and has many real-life applications [1]–[3]. FIM is mainly used to find concomitantly occurring items in the transactions. FIM techniques depend on support confidence framework where the frequency of items should not be less than minimum support threshold [4]–[6]. The high-utility itemsets rarely appear but have high utility values and are often ignored in the FIM algorithms as these consider only occurrence frequencies of itemsets. Therefore, an important limitation of FIM is its assumption that gives equal importance to all items irrespective of their value to the organization. Generally, these assumptions do not hold in real world applications. For example, bread is purchased in hundreds or thousands per day while fewer diamonds are bought in a week or month. The former has higher frequency but lower profit value while the latter has lower frequency with higher profit value for retailers. FIM mining discovers many frequent itemsets generating a low profit while it fails to discover the less frequent itemsets that generate a high profit. To solve the limitation of FIM or association rule mining (ARM), high-utility itemset mining (HUIM) was designed to discover the useful and profitable itemsets from the quantitative databases [6]–[9]. These databases contain utility values of itemsets. An itemset is considered as a high utility itemset (HUI) if its utility value is no less than the user-specific, minimum utility threshold. In real-world practice, the utility of an itemset can be measured by several factors, such as weight, profit, or cost, which can be defined via user's preferences. HUI can help managers to carry out accurate financial analysis and make informed decisions. HUIM is used in a wide range of applications such as website click stream analysis [7], [10], mobile computing [11], top-k HUI mining [12]–[14] and biomedical applications [3]. HUIM also motivates and inspires several data mining tasks such as

2581-4575
International Journal For Recent Developments in Science & Technology
Crossref
A Peer Reviewed Research Journal

high-utility sequential pattern mining [16], high average utility itemset mining [17], [18] and high utility stream mining. Generating high utility itemsets is a combinatorial problem having exponential complexity both in time and space. Therefore, it falls into the NP-hard problem category. HUIM requires handling of huge search space speacially when the database contains many distinct items and has an extremely large size. FIM follows the downward-closure property where the support of an itemset is anti-monotonic, i.e. subsets of a frequent itemset are frequent and supersets of an infrequent itemset are infrequent. This property is very efficient to trim the search space. However, HUIM does not follow monotonic or anti-monotonic properties, because a high utility itemset may have a superset or subset with lower, equal or higher utility. Hence, techniques used to prune the search space in FIM cannot be directly applied in HUIM. Many studies have been carried out in the last decade to develop efficient HUIM algorithms. The drawbacks of FIM and the challenges of HUIM motivate us to design an efficient algorithm to mine high utility itemsets, which enables the user to satisfy his/her perspectives regarding the importance of the item/itemsets. Traditional HUIM algorithms such as PB [20], TWU [21], UP-Growth and UP-Growth+ [22] carry out mining in two phases. However, the two-phase model suffers from the problems of generating a huge number of candidates and repeated scanning of the database which makes the algorithms inefficient. The FHM [23], HUP-Miner [24], HUI-Miner [25], EFIM [26] and d2HUP [27]–[29] algorithms mine complete set of HUI in the single phase without generating candidates. EFIM [26] is the current state-of-the-art algorithm which outperforms all algorithms for HUIM. EFIM proposes database projection on all promising items during the depth first search and prunes the search space using sub-tree and local utility. EFIM evaluates the utility of each itemset while exploring the search space of items. It does this firstly by generating itemsets and then by performing database projection for all promising itemsets. EFIM requires less memory and time as compared to the above mentioned algorithms. It performs well on dense datasets. However, the process of high-utility itemset mining remains costly in terms of runtime and memory usage as it creates projections on all promising itemsets. There is a scope for improvement by designing more efficient algorithms for this task. In this paper, we address this challenge by providing efficient data representation and selective database projection which improves pruning technique of the search space. To reduce the memory and runtime of EFIM algorithm, we propose an efficient selective database projection based HUIM algorithm, called SPHUI-Miner. In this algorithm, we create new database projections of smaller size having less dimensions and unique data instances which result in faster HUI mining. We also provide upper bounds on the amount of memory consumed by these projections. The SPU-List structure is introduced to directly prune the search space for an itemset having utility less than loose upper bound. This avoids traversing down to the unpromising itemsets in the search space. Unlike EFIM, SPHUI-Miner uses Tail-Count list, which maintains count of each item in the database projection. The Tail-Count list is used for applying PEP, that avoids the need of exploration of each itemset. Thus, fewer branches need to be traversed to find the high utility itemsets.

II. RELATED WORK

The problem of high-utility itemset mining is quite attractive for the following reasons. (1) Practically it is more reasonable to identify itemsets revealing the profitable and useful information from customer transactions than those items that are purchased frequently. (2) The problem of high-utility itemset mining is more challenging from a research aspect. HUIM [8], [9], [12], [13] is an emerging, prominent and

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

attractive topic of the current decade. It is the extension of frequent itemset mining (FIM), but it considers more factors such as quantity (Internal) and price (External Utility Values) or profit within it. Many algorithms are designed to mine the set of complete HUIs. Chan et al. [15] first introduced high utility itemset mining problem and discovered the top-k closed utility patterns using their proposed approach. Yao et al. [8] introduced the concept of internal utility and external utility where internal utility indicates the quantity of items and external utility indicates the profit unit of items to determine the HUIs. Since prior HUIM algorithms suffer from the combinational problem to discover the HUIs, Liu et al. [21] designed the two-phase (TWU) model and established the transaction-weighted downward closure (TWDC) property which helped to prune early the unpromising HUIs. Many two-phase algorithms such as IHUP [7] and UPGrowth [22] adopted this mechanism. The two-phases of such algorithms are: Phase 1: generate candidate high-utility itemsets; Phase 2: compute exact utility of the candidates by scanning the database, and filter lowutility itemsets. Ahmed et al. [7] introduced IHUP algorithm based on the two phase model for interactive and incremental mining of HUIs. Lin et al. [30] designed a high-utility-pattern (HUP)-tree for discovering HUIs. They first find the hightransaction-weighted utilization 1-itemsets (1-Htwuis) using two phase model and then build the HUP-tree of 1-Htwuis.

HUP-tree then works like FP-tree approach [29] and performs well on dense datasets such as chess dataset. Tseng et al. [31] designed the UP-growth mining algorithm for discovering HUIs by introducing a special data structure named UP-Tree (Utility Pattern Tree) to store the information of high utility itemsets. All the above mentioned algorithms employ a twophase, candidate generation approach, that suffers scalability issue because of huge number of candidates. Liu et al. [27], [28] presented a single phase and patterngrowth based approach for HUIM (D2HUP). D2HUP constructs a tree-based structure called chain of accurate utility lists (CAUL) that maintains the utility information of the transaction set for each enumerated itemset. It is to be noted that the final HUIs generated by D2HUP (with lookahead pruning) are not complete. An additional iteration for the generated itemsets is required to enumerate the actual high utility itemsets and their utility values. FHM and D2HUP carry-out the costly join operation which results in degraded performance. Since the tree-based algorithms generate too many high utility candidate itemsets, Liu and Qu [25] designed a listbased algorithm named HUI-Miner which mines the HUIs without generating candidates. They use a vertical data structure to represent utility information which is similar to the TID lists used in ECLAT algorithm [32] for mining frequent itemsets. HUI-Miner introduced utility list structure which contains three parts: < tids, iutil,rutil > where tids are transaction ids, iutil is the utility value of an item in a transaction, and rutil is the resting utility value of an item in a transaction. This utility structure performs inefficient join operations and is not scalable. Since HUI-Miner is less efficient for mining large databases, scalability and efficiency are challenges of this algorithm. The improved version of HUIMiner is FHM and HUP-Miner. Fournier-Viger et al. [23] presented an improved algorithm, namely FHM (fast HUI mining) to quickly mine HUIs on the Estimated Utility Co-occurrence Structure (EUCS). The EUCS holds the information of 2-itemsets, which can be used to reduce the computations and database scans. HUP-Miner [24] algorithm is very efficient for pruning unpromising candidates in sparse datasets. It introduces partitioned utility list data structure as an extension of utility list data structure. Lan et al. [20] introduced projection-based algorithm (PB) which uses index mechanism for quick HUIM. Results also indicate

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

that PB algorithm is computationally more powerful than the twophase algorithms. Evolutionary based HUI mining algorithms were also designed such as [33]–[37]. Kannimuthu and Premalatha [38] first introduced the GA-based HUI mining algorithm using ranked mutation. In evolutionary approaches, large computations are required to identify the initial 1-Htwuis which act as chromosome to find HUIs. Moreover, the crossovers and mutations are required to setup the evolution process of GA/PSO for HUIM. The performances vary widely because of choice of crossover and mutation operators. We are not comparing the proposed SPHUI-Miner algorithm with the

An example database D.

T. Id	Transaction (item, quantity)	Transaction utility (tu)
T_1	a(1), c(18), e(1)	27
T_2	b(6), c(1), d(1), e(1), f(1)	67
T_3	a(2), c(1), e(1)	13
T_4	d(1), e(1)	11
T_5	c(4), e(2)	16
T_6	b(1), f(1)	10
T_7	b(10), d(1), e(1)	101
T_8	a(3), c(25), d(3), e(1)	55
T_9	a(1), b(1), f(3)	15
T_{10}	b(6), c(2), d(2), e(2), f(4)	82

External utility values.

Item	External utility
a	3
b	9
c	1
d	5
e	6
f	1

evolutionary algorithms because of their non-deterministic performance. EFIM [26] is state-of-art HUIM algorithm which introduces sub-tree utility and local-utility structure to prune the solution space. Although it outperforms all previous algorithms on standard datasets. The main drawback of EFIM algorithm is that it performs database projection on each promising itemset taking more time and memory. In this paper, we address this challenge by integrating two upper bounds using SPU-List and Tail-Count structure, thereby reducing database projection size and its scanning cost.

III. PROPOSED METHOD

In this research work an enhanced high utility pattern approach (EHUPA) has been proposed to improve the performance of high utility pattern for mining itemsets. The proposed system contains name of the item as node and after calculating transaction utility and transaction weighted utility, the item sets having less utility than predefined minimum threshold utility are identified. Local unfavorable items are removed using path utility of each item in descending order. The reorganized path is inserted into the utility pattern tree using reduce local node utility strategy. Potential high utility item sets and their utilities are identified by the proposed system. The proposed system eliminates the local unfavorable items and reduces local node utility. The proposed system improves the performance of high utility pattern for mining itemsets in large datasets with several advantages such as less memory space usage and less execution time for mining itemsets. A. System design The proposed enhanced high utility pattern approach has been designed to find effective high utility patterns for improving the performance of mining itemsets. The proposed system describes the dataset for set of transactions with profit item as input to system with calculation of transaction utility transaction weighted utility, utility pattern tree construction, high utility pattern algorithm and finally the output as the enumerated patterns for utility itemsets. The transaction utility and transaction weighted utility prune the search space of high utility itemsets. The proposed system design is shown in Figure 3.1

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

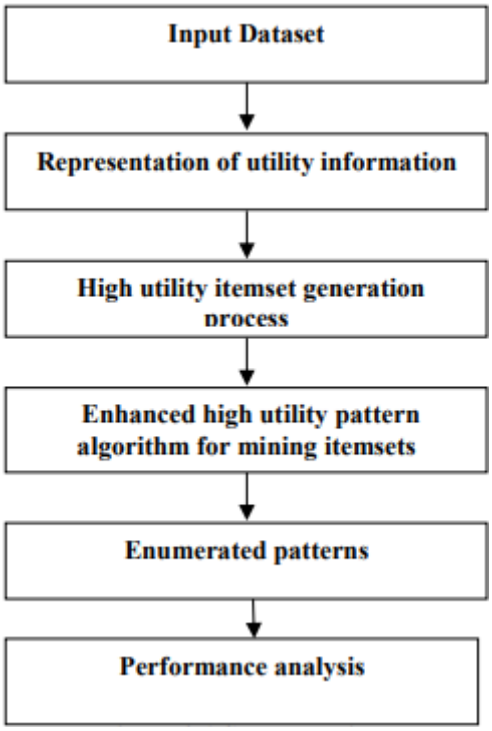


Figure 3.1 System design

The proposed system builds transaction set (TS) by scanning the database D and build the external utility table (XUT)to compute s(i), u(i),uBitem(i), and uBfpe(i) for each item i. The proposed system starts searching high utility patterns from the construction of the utility pattern tree by calling the depth first search (DFS) approach. For the each node N currently being visited, DFS prints pattern (N) as a high utility pattern if its utility is no less than the threshold which makes the set W of relevant items.

B. Enhanced high utility pattern algorithm A pattern that is of interest of one user may not be interest to another user, since users have different levels of interest in patterns. A pattern is of utility to a person if its use by that person contributes to reach a goal. People may have differing goals concerning the knowledge that can be extracted from a data set. The proposed system allows a user to conveniently express the perceptiveness concerning the usefulness of patterns as utility values higher than a threshold.The proposed enhanced high utility pattern approach (EHUPA) finds high utility pattern to enumerate each subset of item, and test if subset has a utility over the threshold. The transaction set is build scanning the database and the utility table is used to filter out the irrelevant items. The proposed system starts searching high utility patterns from the root of utility pattern tree using depth first search. The node currently is being visited computing utilities and if it's utility is no less than the threshold, makes the set of relevant item for a high utility for each relevant item belongs to the set.The proposed algorithm has been used the utilities such as transaction utility and external utility for identifying the items. The transaction utility of an item is obtained from the information stored in the transaction dataset. The external utility of an item is given by the user and is based on information not available in the transaction dataset. In this work external utility has been represented by a utility table or utility function. By combining a transaction dataset and a utility function the proposed algorithm finds the discovered pattern. The proposed algorithm performs various steps that include representation of utility information, construction of utility pattern tree and the generation of high utility pattern for itemsets. The proposed algorithm provides scalability and efficiency for mining utility itemsets along execution time and memory space on database transactions. The proposed enhanced high utility pattern algorithm for mining high utility itemsets is shown in Figure 3.1

IV. ALGORITHM

SPHUI-MINER ALGORITHM

In the proposed work, we explore two ways of optimization: i) reducing cost of database scan by representing database using

twu of each item of database in Table 1

Item	a	b	c	d	e	f
twu	110	275	260	316	372	174

efficient HUI-RTPL and creating projections selectively; and ii) enumerating less number of

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

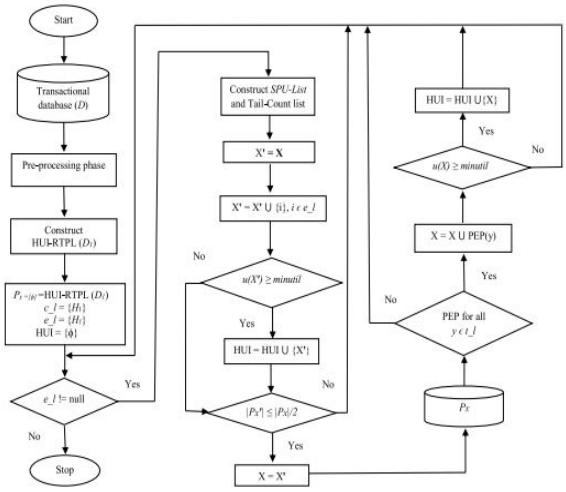
probable candidates using SPU-List and Tail-Count list (t_l). We propose selective projection based approach and several novel techniques to reduce the time and memory requirement. Selective projection approach creates new projections of smaller sizes in terms of vertical columns (number of items) and horizontal rows (number of unique transactions) which results in faster computing of utility of itemsets. In this paper, we present two upper bounds with pruning strategies to reduce the overestimation of the utilities of the itemsets. Thus the search space for discovering HUIs can be greatly reduced and many unpromising candidates can be pruned early. Our complete approach is illustrated in Figure 1 and the procedure is described in Algorithm 7

A. PRE-PROCESSING PHASE

The proposed SPHUI-Miner algorithm uses the upper bound twu model [21] to determine high transaction-weighted utilization 1-itemsets (1-Htwuis). As the twu model involves the transaction-weighted downward closure (TWDC) property, it prunes all the unpromising items. It first calculates the utility of items in each transaction as the transaction utility (tu) (Lines 2-4 in Algorithm 1). Then it calculates the transaction-weighted utility (twu) of an item by summing up the transaction utility of an item if that item is present in the transaction (Line 6-9 in Algorithm 1). This process is used to estimate the upper-bound value of an item based on the twu model. If the transaction-weighted utility of an item is not less than the minimum utility value, thus it is considered as 1-Htwui. For the example database shown in Table 2, the minimum utility value minutil is calculated as (TU × δ)(= 397 × 0.4 = 158.8). The twu(a) is calculated as: twu(a) (= 27+13+55+ 15)(= 110 < 158.8). As twu(a) < minutil, item {a} is not a 1-Htwui. The twu of the remaining items are calculated in the same way as shown in Table 4. The items except {a} are 1-Htwui and H1 is a set of all 1-Htwui. For the complete database, all 1-Htwui items in a transaction are sorted in ascending order of twu values as shown in Table 5, forming the ordered itemset for all transactions in a database.

B. HIGH UTILITY-REDUCED TRANSACTION PATTERN LIST

The horizontal representation of the database is shown in Table 5 and it can be seen that the transactions T6, T7, and T8 are same, so only one copy of these similar transactions is stored along with their total items utility. The complete database representation using this structure is called High utility-Reduced Transaction Pattern List (HUI-RTPL) (D1) as shown in Table 6



The overall process of SPHUI-Miner.

IV. RESULTS

In this work five real-world datasets are used for evaluation. The first dataset is T10I6D1M which contains the items are selected such as milk, bread, butter, jam. The data from used for 1-itemset,2-itemset,3-itemset.The second one is Chess which is a dense dataset used for transaction items. The third dataset is Chain store generated the itemsets. The fourth one is T20I6DIM in mixed dataset, it increases with the transactions. The last dataset is foodmart which contains real utility values generated from high utility values. For T10I6D1M, Chess, Chain store and T20I6DIM, food mart divide each part

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

generated itemsets in 10,000 items are selected for each transaction. The transaction database that contains items which find effective high utility patterns for improving the performance of mining itemsets. The method provides scalability and efficiency for mining utility itemsets along execution time and memory space on database transactions. The first column is the name of a dataset, the second (|t|) is the average and maximum length of transactions, the third (|I|) is the number of distinct items, the fourth (|D|) is the number of transactions, and the fifth (Type) is a rough categorization based on the number of high utility patterns to be mined, partially depending on the minimum utility threshold. The detailed of the five datasets which include transaction, distinct items, number of transaction and type are shown in Table 4.1. The proposed system contains name of the item as node and after calculating transaction utility and transaction weighted utility, the item sets having less utility than predefined minimum threshold utility are identified. Local unfavorable items are removed using path utility of each item in descending order. The reorganized path is inserted into the utility pattern tree using reduce local node utility strategy. Potential high utility item sets and their utilities are identified by the proposed system. The proposed system eliminates the local unfavorable items and reduces local node utility.The proposed enhanced high utility pattern approach has been designed to find effective high utility patterns for improving the performance of mining itemsets. The proposed system describes the dataset for set of transactions with profit item as input to system with calculation of transaction utility transaction weighted utility, utility pattern tree construction, high utility pattern algorithm and finally the output as the enumerated patterns for utility itemsets. The transaction utility and transaction weighted utility prune the search space of high utility itemsets. The proposed algorithm has been executed on same minimum utility value as per the datasets to generate itemsets. Experiments are performed to evaluate the performance of the proposed EHUPA with the existing d 2HUP algorithm based on five datasets.

data sets

Data Set	Transaction |t|	Distinct Items |I|	Number of Transaction |D|	Type
T10I6D1M	10:33	1000	933,493	Mixed
Chess	37:37	76	3,197	Dense
Chain-Store	7.2:170	46,086	1,112,949	Sparse
T2016DIM	20:49	1,000	999,287	Mixed
Foodmart	4.8:27	1,559	34,015	Dense

C. Performance analysis and results

Experiments are performed to evaluate the performance of the proposed EHUPA algorithm based on the five datasets. The performance of EHUPA algorithm has been compared with existing d 2HUP algorithm based on the metrics such as memory usage and running time for mining high utility itemsets

- **Memory usage for high utility pattern**

The proposed EHUPA is compared with existing d 2HUP method for memory usage for mining high utility itemsets and the performance graph is shown in Figure 4.1. In the graph, x-axis represents the datasets and y-axis represents the memory space. The graph shows that the proposed EHUPA method provides better high utility pattern mining with less memory space usage than the existing d 2HUP method

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

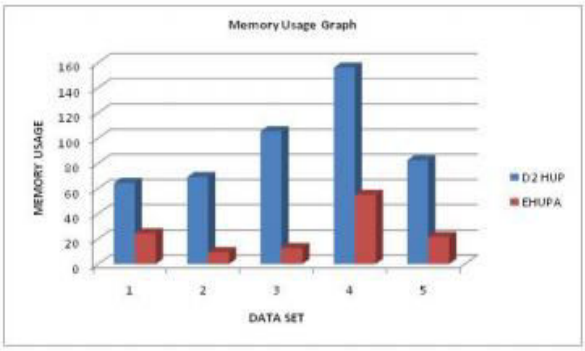


Figure 4.1 Comparison of memory usage for high utility pattern

- **Running time for high utility pattern**

The proposed EHUPA is compared with existing d 2HUP method to mine high utility itemsets for running time and the performance graph is shown in Figure 4.2. In the graph, x-axis represents the datasets and y-axis represents the running time of utility itemsets. The graph shows that the proposed EHUPA method provides better high utility pattern mining with less running time than the existing d 2HUP method.

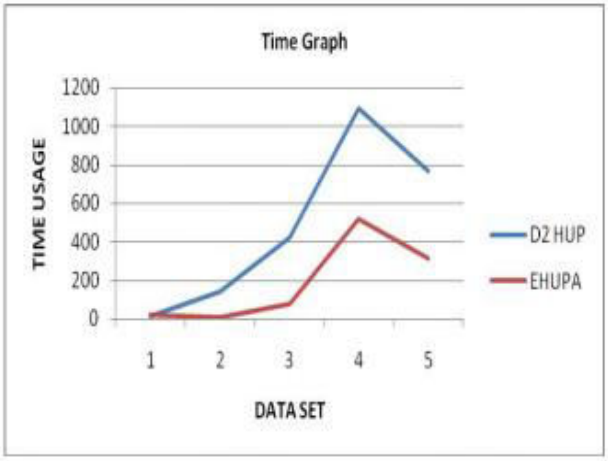


Figure 4.2 Comparison of running time for high utility pattern

V. CONCLUSION

The high utility pattern mining with the itemset share framework is more challenging than the other categories of utility mining such as weighted itemset mining, association rule mining and frequent pattern mining. During the knowledge discovery process, utility based measures are used to find the unidentified patterns to improve the mining efficiency. In this research paper an enhanced high utility pattern approach (EHUPA) has been proposed to mine high utility itemsets. The experimental results show that the proposed system provides better performance than the existing direct discovery high utility pattern algorithm in terms of memory usage and execution time for mining high utility itemsets.

REFERENCES

1) Ahmed C. F., Tanbeer S. K., Jeong B.-S., and Lee Y. -K., "Efficient tree structures for high utility pattern mining in incremental databases," IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 12, pp. 1708– 1721, 2009.
2) Arati Borker W., "Utility mining algorithm for high utility itemsets from transactional databases," International Organization of Scientific R Journals (IOSR Journal of Computer Engineering), Vol. 16, No. 2, pp. 34– 40, 2014.
3) Bonchi F., Giannotti F., Mazzanti A., and Pedreschi D., "Exante: A preprocessing method for frequent-pattern mining," IEEE Intelligent Systems, Vol. 20, No. 3, pp. 25–31, 2005.
4) Chun-Wei Lin J., Wensheng Gan., Fournier-Viger P., and Yang L., Liu Q., Frnda J., Sevcik L., Voznak M., " High utility itemset-mining and privacy-preserving utility mining," Vol. 7, No. 11, pp. 74–80, 2016.
5) Dawar S., Goya V. l., "UP - Hist tree: An efficient data structure for mining high utility patterns from transaction databases," In Proceedings of the 19 th International Database Engineering & Applications Symposium. Association for Computing Machinery, pp. 56–61, 2015.
6) Erwin A., Gopalan R. P and. Achuthan N. R., "Efficient mining of high utility itemsets from

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

large datasets," In Proceeding of the Pacific-Asia Conference on