IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

An Efficient and Scalable way using UP-Growth and UP-Growth+ Algorithms for finding Mining High Utility Item sets from Transactional Databases

Mr. S. H. Sangle[1], Prof. Mrs. R. C. Samant[2]

PG Student, Computer Engineering Department, R.H. Sapat College, Pune University, Nashik, Maharashtra, India [1]

Assistant Professor, Computer Engineering Dept, R.H. Sapat College, Pune University, Nashik, Maharashtra, India[2]

Abstract: Mining high utility itemsets from a large transactional database refers to the discovery of knowledge like high utility itemsets (profits). Since a number of relevant algorithms have been proposed in past years, they fall into the problem of producing a large number of candidate itemsets for high utility itemsets. Such a huge number of candidate itemsets decrease the mining performance in terms of time and space complexity. The situation may become worse when the database contains lots of long transactions or long high utility itemsets. An emerging concept in the field of data mining is utility mining. To identify the itemsets with highest utilities is the main objective of utility mining, by considering profit, quantity, cost or other user preferences. This topic is having many applications in website click stream analysis, cross marketing in retail stores, business promotion in chain hypermarkets, online e-commerce management, finding important patterns in biomedical applications and mobile commerce environment planning.

Keywords: Candidate pruning; frequent itemset; high utility itemset; utility mining; data mining.

I. INTRODUCTION

Now a days, data mining is one of the evolving area in the web domain. It has established its good position for studies in research area. It is useful for knowledge discovery in databases. The objective of data mining is to abstract higher-level hidden information from a very large quantity of raw data. Data mining is useful for various data domains. Data mining is the process which takes data as input and yields patterns, like classification rules, item sets, association rules, or summaries, as output. The traditional Association Rule Mining (ARM) approaches consider the utility of the items by its presence in the transaction set. The frequency of item set is not sufficient to reflect the actual utility of an item set. For example, the sales manager may not be interested in frequent item sets that do not generate significant profit. Recently, one of the most challenging data mining tasks is the mining of high utility item sets efficiently. Identification of the item sets with high utilities is called as Utility Mining.

Mining Association rules is one of the research challenge in data mining. The concept of Association rule mining wasfirst introduced in [1] later broadened in [2].An association rule is an expression of the form X=>Y, where X and Y are sets of items. Once the frequent item sets are found, association rules are generated [3]. The concept of frequent item set mining was introduced in [1]. Frequent item sets are the item sets that occur frequently in the transaction dataset. The prime drawback of association rule mining is Rare Item Problem. Many more approaches to mining association rules implicitly consider the utilities of the item sets to be equal [3].**Rare item sets** are the item sets that occur infrequently in the transaction dataset. In most business applications, frequent item sets may not generate much profit while rare item sets may generate a very high profit. For example [4], in the security field, normal behaviour is very frequent, whereas abnormal or suspicious behaviour is less frequent.

Data mining is an important process that identifies correct, formerly unknown and possibly useful patterns in data. Such patterns are used to make divination or classifications about unused data, describe an existing data, outline the contents of a huge database to support decision making and provide graphical data visualization to aid humans in discovering deeper patterns Mining useful patterns hidden in a database plays a key role in various data mining tasks, like frequent pattern mining, weighted frequent pattern mining, and high utility pattern mining. Among them, frequent pattern mining is an elemental research topic that has been applied to different kinds of databases, like transactional databases, streaming databases, and time series databases, also various application domains, such as bioinformatics, Web click-stream analysis, and mobile environments.

As a result of this, utility mining transpires as a key topic in data mining field. Mining high utility itemsets from

 DOI 10.17148/IJARCCE.2016.511111

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

databases refers to finding the itemsets with high profits. In this, the itemset utility means pleasingness, significance, or benefit of an item to users. Utility of items in a transaction database consists of two aspects, External utility: The importance of distinct items, which is called external utility and Internal utility: The importance of items in transactions, which is called internal utility. Utility of an itemset is the product of its external utility and its internal utility. An itemset is a high utility itemset onlyif its utility is no less than a user-specified minimum utility threshold; otherwise, it is called a low-utility itemset.

To facilitate the performance of utility mining, existing overestimated methods first discover the potential high utility itemsets(PHUIs) and then for identifying their utilities an additional database scan is performed. Though, existing methods often generate a huge set of PHUIs and their mining performance is degraded consequently. As the more PHUIs the algorithm generates, the higher processing time it consumes. To overcome this issue, our system propose two novel algorithms as well as a compact data structure for efficiently mining high utility itemsets from transactional databases. The proposed algorithms are utility pattern growth (UPGrowth) and UP-Growth+, and a compact tree structure is utility pattern tree (UP-Tree). To efficiently generate high-utility itemsetsfrom UP-Tree requires only two scans of original databases. Various strategies are proposed for facilitating the mining processes of UP-Growth and UP-Growth+ algorithms by upholdingthe essential information in UP-Tree. Because of these strategies, overestimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility or are not complicated in the search space. The proposed system can not only decrease the overestimated utilities of PHUIs but also greatly reduce the number of candidates.

The rest of the paper is structured as follows: Section II introduces related work for high utility itemset mining. Section III describes proposed system, mathematical model and algorithms for proposed system. Section IV reveals implementation details. Section V shows experimental results and finally, Section VI concludes the paper.

II. RELATED WORK

Apriori algorithm [2], was proposed to find frequent itemsets from the database.The problem in miming the association rules was to generate all association rules that have support and confidence greater than the user specified minimum threshold respectively. The first pass of the algorithm simply counts item occurrences to find the large 1-itemsets. First it generates the candidate sequences and then it selects the large sequences from the candidate ones. Second, the database is scanned and the support of candidates is counted. The second step is to generate association rules from frequent itemsets. The candidate itemsets are stored in a hash-tree. The hash-tree node contains either a list of itemsets or a hash table. Apriori is a usual algorithm for frequent itemset mining and association rule learning over transactional databases. Once the large itemsets are identified, only those itemsets are allowed which have the greater support than the threshold. AprioriAlgorithm creates lot of candidate item sets and scans database every time. When a new transaction is added to the database then it should rescan the entire database again.

Frequent pattern tree (FP-tree) structure was proposed in [6], it is an extended prefix tree structure for storing central information about frequent patterns, squeezed and develop. Pattern fragment growth discovers the complete set of frequent patterns using the FP-growth. It builds a highly compact FP-tree, which is normally and significantly smaller than the original database, using it costly database scans are saved in the subsequent mining processes. It applies a pattern growth method which avoids costly candidate generation. Because of that FP-growth is not able to find high utility itemsets.

Weighted association rule (WAR) was proposed in [7]. In this, author first mines frequent itemsets and the weighted association rules for each frequent itemset are generated. WAR used a twofold approach. First approach: it generates frequent itemsets, in this it ignores the weight associated with each item in the transaction. Second approach: For each frequent itemset the WAR finds that meet the support and confidence. WAR mining first proposed the concept of weighted items and weighted association rules. Furthermore, the weighted association rules does not have downward closure property, mining performance cannot be improved. Using transaction weight, weighted support can not only reflect the importance of an itemset but also maintain the downward closure property during the mining process.

A Two-phase algorithm forfinding high utility itemsets was proposed in [8]. The utility mining is the process to identify high utility itemsets that drive a large portion of the total utility. To discoverall theitemsets whose utility values are over a user specified threshold is nothing but the utility mining. Two-Phase algorithm efficiently trims down the number of candidates and acquires the complete set of high utility itemsets. The transaction weighted utilization is explained in Phase I, only the combinations of high transaction weighted utilization itemsets are attach into the candidate set at every level during the level-wise search. Phase II has only one extra database scan performed to filter the overestimated itemsets. Two-phase requires fewer database scans, less memory space and less computational cost. It performs very effectively in terms of speed and memory cost both on synthetic and real databases, even on large databases. Two-phase just only focused on traditional databases and is not suited for data streams. Two-phase was not proposed for finding temporal high

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

utility itemsets in data streams. Furthermore, it requires the whole database to be rescan while adding new transactions from data streams. But it need more times on processing I/O and CPU cost for finding high utility itemsets.

Two efficient one pass algorithms MHUI-BIT and MHUI-TID were proposed in [9].These are useful for mining high utility itemsets from data streams within a transaction sensitive sliding window. Two efficient depictions of extended lexicographical tree-based summary data structure and itemset information were developed to improve the efficiency of mining high utility itemsets.

A novel method THUI (Temporal High Utility Itemsets) was proposed in [13]. It is a Mine for mining temporal high utilityitemset mining. THUI are are effectively recognized by the novel contribution of THUI-Mine by creating fewer temporal high transaction weighted utilization 2-itemsets so that the time of the execution will be lessen substantially in mining all high utility itemsets in data streams. To produce a progressive set of itemsets THUI-Mine employs a filtering threshold in each partition. Like this, the process of mining all temporal high utility itemsets under all time windows of data streams can be achieved effectively. The temporal high utility itemsets with less candidate itemsets and higher performance can be discovered using THUI- mine. Using these candidate k-itemsets to find a set of high utility itemsets finally, it needs one more scan over the database. Huge memory requirement and lot of false candidate itemsets are the two problems of THUI- Mine algorithm.

An algorithm is defined in [12] for frequent item set mining, it identifies high utility item combinations. The aim of the algorithm is distinct from the frequent item mining techniques and traditional association rule. The algorithm is useful to find segment of data, which is defined with the combination of few items i.e. rules, a predefined objective function and satisfy certain conditions as a group. The problem considered in high utility pattern mining is different from former approaches as it conducts rule discovery with respect to the overall criterion for the mined set as well as with respect to individual attributes.

In [10] the author observed that the traditional candidate-generate-and-test approach for recognizing high utility itemsets is unsuitable for dense date sets. The high utility itemsets are extracted using the pattern growth approach is the novel algorithm called CTU-Mine.

A novel algorithm Fast Utility Mining (FUM) was proposed in [11]. FUM finds all high utility itemsets within the given utility constraint threshold. To create different types of itemsets, the authors also suggest a techniques like High Utility and High Frequency (HUHF), High Utility and Low Frequency (HULF), Low Utility and High Frequency (LUHF) and Low Utility and Low Frequency(LULF).

For discovering high utility itemsets from transactional databases a novel algorithm with a compact data structure was presented in [15]. As per the structure of a global UP-tree the high utilityitemsets are created using UP-Growth which is one of the efficient algorithms. Phase-I has three steps are followed by UP-tree framework i.e. UP-Tree construction, Generation of PHUIs from the UP-Tree and the high utility itemsets should be identified using PHUI.

Global UP-Tree construction consist of following two phases 1) To remove the low utility items and their utilities from the transaction utilities. It is done by getting rid of global unpromising items (i.e., DGU strategy), 2) During global UP-Tree creation discarding global node utilities (i.e., DGN strategy) the node utilities which are nearer to UP-Tree root node are efficiently lessen using DGN strategy. The PHUI is similar to TWU, in which the itemsets utility is calculated with the help of estimated utility and from PHUIs value the high utility itemsets (not less than min_sup) have been identified finally. The global UP-Tree contains many sub paths. From bottom node of header table the each path is considered. And the path is named as conditional pattern base (CPB).

Even the numbers of candidates in Phase 1 are effectively reduced by DGU and DGN strategies. (i.e., global UP-Tree). Yet during the creation of the local UP-Tree (Phase-2) they cannot be applied. To discard the utilities of low utility items from path utilities of the paths DLU strategy should be used instead of it and for discarding item utilities of successor nodes during the local UP-Tree construction DLN strategy should be used. Although the algorithm is facing still some performance issues are there in Phase-2.

III.PROPOSED SYSTEM

a) System Architecture:
The proposed high utility mining system will be a conceptual model built for large transactional database. Analysing high utility itemsets from transactional databases refers to mining the itemsets with high profit or other preferences like quantity, cost etc. The high utility itemset means that if itsutility is larger than a user-specified minimum utility threshold and if utility of itemset is less than the user-specified minimum utility threshold then it is called a low-utility itemset. Administrator will specify or enter the minimum Utility Threshold value. In market, every day a new product is released, and hence, the administrator would add the product, view the stock details, update the new product itemsets and the full control is only for the administrator. Customer can purchase the items and all the purchased

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

items are stored in the transaction history and then transaction details are sends to original database which shows the list of items that are purchased by the customer. When administrator wants to find the utility of items with its profit the administrator gives the minimum threshold and dataset to the system. The system will compute the operation on given input and generate the high utility itemsets. Fig.1 shows the system architecture for proposed system.

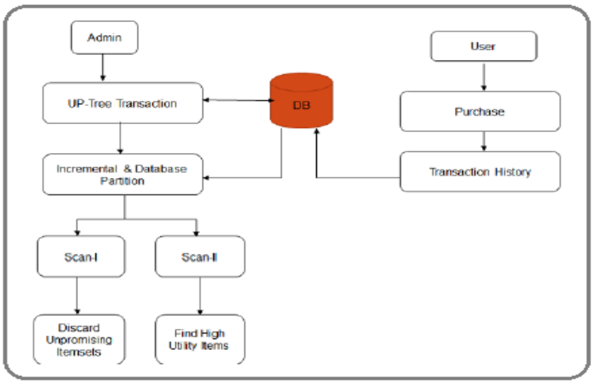


Fig. 1: System Architecture

b) Proposed System Working:

The flow of the proposed methods consists of three steps:1) Scan the database two times to build a global UP-Tree. 2)Recursively generate PHUIs(Potential High Utility Itemsets)from global UP-Tree and local UP-Trees by using UP-Growthwith two strategies DLU (Discarding Local Unpromisingitems) and DLN (Decreasing Local Node) or by UP-Growth+with the DGU (Discarding Global Utility Items) and DGN(Decreasing Global Node) strategies ; and 3) identify actual high utility itemsets from the set of PHUIs. Global UP-Tree construction is as follows as: (i). By Discarding Global Unpromising items (i.e., DGU strategy), discard or eliminate the low utility items and their utilities from the transaction utilities. (ii) During global UP-Tree construction Discarding Global Node utilities (i.e., DGN strategy) the node utilities which are nearer to UP-Tree root node are effectively reduced by DGN strategy. The PHUI is similar to TWU, in which the itemsets utility is computed with the help of estimated utility and from PHUIs value the high utility itemsets (notless than minimum utility) have been identified finally. The global UP-Tree contains many sub paths. From bottom node of header table the each path is considered. And the path is named as conditional pattern base (CPB).

Even the numbers of candidates in Phase 1 are efficiently reduced by DGU and DGN strategies. (i.e., global UP-Tree).But during the construction of the local UP-Tree (Phase-2) they cannot be applied. For discarding utilities of low utility items from path utilities of the paths DLU strategy will be used instead of it and for discarding item utilities of descendant nodes during the local UP-Tree construction DLN strategy will be used.

c) Mathematical Model

The proposed system S is defined as follows:
S: {I, D, X, M, N, K}
Where,
I = (i1, i2 … im): items,
X = (i1, i2 … ik): itemset
D = (T1, T2 … Tn) : Transaction Database .
k = number of distinct items.
M=number of finite set of items
N =number of Transaction
F =F1, F2, F3, F4, F5, F6, F7
.

Function F1:
Utility of an item $i_p$in a transactionT_d is denoted as $u(i_p, T_d)$.
$u(i_p, T_d)=pr(i_p)*q(i_p, T_d)$
Where,
$pr(i_p)$=unit profit,
q=quantity of item in transactions

Function F2:
Utility of an itemsets X in T_d is denoted as $u(X, T_d)$

$$u(X, Td) = \sum \sum_{ip \in X \wedge X \subseteq Td} u(ip, Td)$$

Function F3:
Utility of an itemset X in T_d is denoted as $u(X, T_d)$

$$u(X, Td) = \sum \sum_{X \subseteq Td \wedge Td \in D} u(X, Td)$$

Function F4:
High Utility itemsets.
Itemset is called High utility itemset if its utility is no less than a user specified minimum utility threshold which is denoted as min_util. Otherwise, it called low utility itemset.

Function F5:
Transaction utility of a transaction T_d denoted as $TU(T_d)$.
$TU(T_d)=u(T_d, T_d)$

Function F6:
Transaction-Weighted utility of an itemset X is sum of the transaction containing X, which is denoted as TWU(X)

$$TWU(X) = \sum_{X \subseteq Td \wedge Td \in D} TU(Td)$$

Function F7:
An itemset X is called a high-transaction weighted utility itemset (HTWUI) if TWU(X) is no less than min_util.

d) Algorithms:

Algorithm 1- Utility Pattern Growth (UP-Growth)

Input:
- T_x, UP-Tree.
- H_x, Header Table for T_x.
- Minimum utility threshold minimum utility
- Item set X = i_1; i_2::; i_k

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

Output:

- High utility itemsets.

Steps:
1. For each entry ik in Hx do
2. Trace each node related to item i_k : And calculate sum of node utility .
3. If sum (i_k)>= min-util do
4. Generate Potential High Utility Itemset (PHUI)
5. Set Potential Utility of as estimated utility
6. Construct Conditional Pattern Based (CPB) Y-CPB.
7. Put local promising items into Y-CBP .
8. Apply Discarding Local Unpromising (DLU) to minimize path utilities of paths.
9. Apply DLU to insert path into Ty .
10. If Ty ≠ then call to UP-Growth().
11. End if
12. End for.

Algorithm 2: UP-Growth+

Input:

- UP-tree T_X.
- A header table H_x.
- An itemset X.
- Minimum utility threshold min_util,

Output:

- All PHUIs in T_X

Steps:
1) For each entry i_k in $H_x$do
2)Trace each node related to i_k via $i_k$hlink and accumulate i_k. nu to $nu_{sum}(i_k)$; /*the sum of node utilities of i_k*/
3) If $nu_{sum}(i_k) \geq$ min_util, do
4) Generate a PHUI Y = X ∪i_k;
5) Set pu(i_k) as estimated utility Y;
6) Construct Y-CPB;
7) Put local promising item in Y-CPB into H_Y
8) Apply DPU to reduce path utilities of the paths;
9) Apply Insert_ Reorganized_ Path mnu to insert into T_Y with DPN;
10) If T_Y ≠ null then call Enhanced UP-Growth+ ()
11) End if
12) End for

UP-Growth improves the performance than FPGrowth by using DLU and DLN to reduce the unpromising utilities of itemsets. The proposed an improved method, named UP-Growth+, used for reducing minimum utilities more effectively. In UP-Growth, minimum item utility table is used to reduce the unpromising or minimum utilities. In UP-Growth+ algorithm, minimal node utilities in each path are used to create the expected pruning values closer to real utility values of the pruned items in database After finding all PHUIs(Potential High Utility Itemsets), the third step is to fond high utility itemsets and their utilities from the set of PHUIs by scanning original database once.

IV. IMPLEMENTATION DETAILS

The experiments were performed on a 2.80 GHz Intel Pentium D Processor with 3.5 GB memory. The operating system is Microsoft Windows 7. The algorithms are implemented in Php language. Both real and synthetic data sets are used in the experiments. Synthetic data sets were generated from the data generator in [1]. Real world data sets are generated from the data generator in [16].

V. RESULTS AND DISCUSSION

Figure 2 shows the Execution Time Result of utility mining for minimum utility and specified number of items using Upgrowth and Upgrowth + algorithms

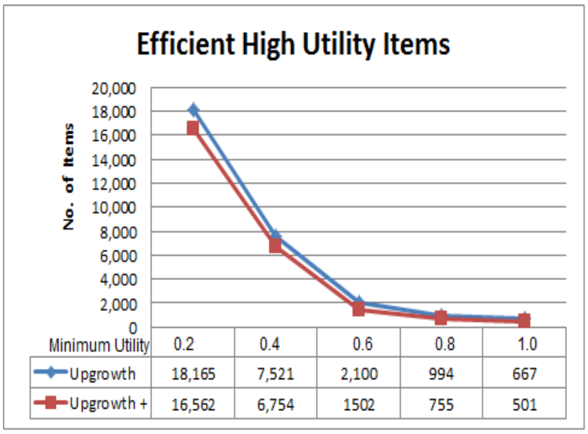
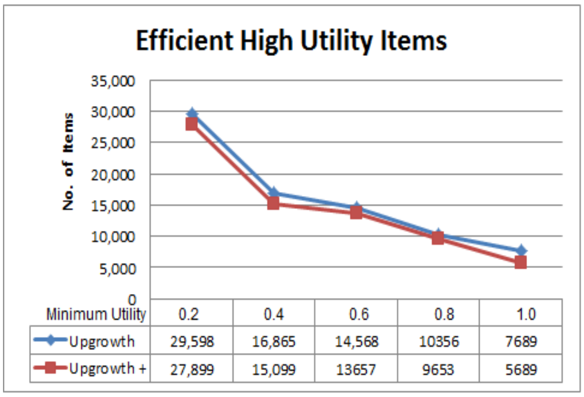


Fig. 2: Phase – I: Execution Time Result of Utility mining.



Phase – II: Execution Time Result of Utility mining.

After analysing above both charts, the Upgrowth plus Algorithm is more Efficient and scalable in terms of Execution Time while finding high utility Items from transactions.

Figure 4 shows the analysis graph of an element using an upgrowth+ algorithm.

Figure 6 shows the analysis of a **71053** item (Monthly profit) monthwise Occurrence using High Utility Transaction

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

Fig. 4: Analysis Graph of Element 85123A using Upgrowth+ Algorithm

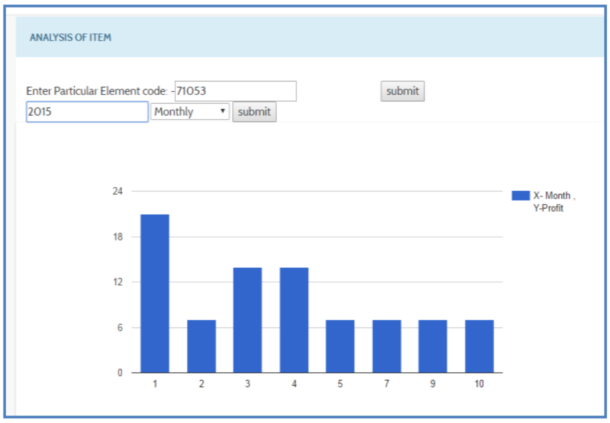


Fig. 6: Analysis of 71053 Item monthwise Occurrence in High Utility Transaction

In this way we have shown the performance of the UPGrowth and UPGrowth+ algorithm. UPGrowth+ algorithm performs better than the UPGrowth algorithm as UPGrowth+ requires less execution time to find out high utility itemsets than the UPGrowth. We have check the performance on different minimum utility thresholds. To evaluate the scalability of these two algorithms we vary the dataset size. When the size of database increases, the execution time for identifying each high utility itemset also increases.

Therefore, UPGrowth requires more processing time than UPGrowth+. The results show that our approach of UPGrowth+ algorithm outperforms on the available datasets. The information storing in the form of UPtree requires fast processing algorithm to mine the high utility itemsets. Our UPGrowth+ gives the expected and better result by pruning the unpromising candidates and efficiently generates the required candidates. Hence, UPGrowth+ is required to improve the UPGrowth algorithm. To show the better performance of our proposed system we have also analyse the yearly profit of two items using UPGrowth and UPGrowth+ algorithms. Also, we have shown the monthwise profit analysis of items for mining high utility itemsets. UPGrowth+ algorithm performs faster for finding year and monthwise profit of an item than UPGrowth.

VI. CONCLUSION AND FUTURE WORK

The system in this paper, proposed two effective algorithms named UP-Growth and UP-Growth+ for discovering high utility itemsets from transaction databases. For efficiently maintaining the information of high utility itemsets, a compact data structure i. e. UP-Tree is proposed. Only two database scans generates the PHUIs efficiently from UP-Tree. Furthermore, some strategies are developed for decreasing overestimated utility and enhancing the performance of utility mining. To evaluate the performance of system the datasets are used. The proposed strategies improve the performance by reducing both the search space and the number of candidates. Further, the proposed algorithms, especially UP-Growth+, better perform the present day algorithms considerably and mainly when databases contain lots of long transactions or a low minimum utility threshold is used.

ACKNOWLEDGMENT

I am thankful to my project guide **Prof. Mrs. R. C. Samant** for her kind support and valuable guidance. She help me time to time to improve the quality of project work in all respects. I am also thankful to the Head of Computer Engineering Department. I thank to my colleagues and friends who guide me directly and indirectly to complete the paper work. I would also thankful to all the staff members of Computer Engineering Department and Librarian, who gives their valuable guidance to me. Specially, I am thankful to my family members for their support and co-operation during this Project work.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, 1993,"Mining association rules between sets of items in large databases", in Proceedings of the ACMSIGMOD International Conference on Management of data, pp 207-216.
[2] R. Agrawal and R. Srikant, 1994, "Fast Algorithms for Mining Association Rules", in Proceedings of the 20th International Conference Very Large Databases, pp. 487-499.
[3] H.Yao, H. J. Hamilton, and C. J. Butz, "A Foundational Approach to Mining Item set Utilities from Databases", Proceedings of the Third SIAM International Conference on Data Mining, Orland, Florida, pp. 482-486, 2004.
[4] M. Add, L. Wu, Y. Fang, "Rare Item set Mining", Sixth International conference on Machine Learning and Applications, 2007, pp 73-80.
[5] R. Chan, Q. Yang, Y. D. Shen, "Mining High utility Item sets", In Proc. of the 3rd IEEE Intel.Conf. On Data Mining (ICDM), 2003.
[6] J. Han, J. Pei, Y. Yin, "Mining frequent patterns without candidate generation," in Proc. of the ACM-SIGMOD Int'l Conf. on Management of Data, pp. 1-12, 2000.
[7] W. Wang, J. Yang and P. Yu, "Efficient mining of weighted association rules (WAR)," in Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2000), pp. 270-274, 2000.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

[8] Y. Liu, W. Liao and A. Choudhary, “A fast high utility itemsets mining algorithm,” in Proc. of the Utility-Based Data Mining Workshop, 2005.
[9] 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 Proc. of the 8th IEEE Int'l Conf. on Data Mining, pp. 881-886, 2008.
[10] A. Erwin, R. P. Gopalan and N. R. Achuthan, “Efficient mining of high utility itemsets from large datasets,” in Proc. of PAKDD 2008, LNAI 5012, pp. 554-561.
[11] S.Shankar, T.P.Purusothoman, S. Jayanthi,N.Babu, A fast agorithm for mining high utility itemsets , in :Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp.1459-1464.
[12] J. Hu, A. Mojsilovic, “High-utility pattern mining: A method for discovery of high-utility item sets”, Pattern Recognition 40 (2007) 3317 – 3324.
[13] V. S. Tseng, C. J. Chu and T. Liang, “Efficient Mining of Temporal High Utility Itemsets from Data streams,” in Proc. of ACM KDD Workshop on Utility-Based Data Mining Workshop (UBDM’06), USA, Aug., 2006.
[14] V. S. Tseng, C.-W. Wu, B.-E. Shie and P. S. Yu, “UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining,” in Proc. of the 16th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD 2010), pp. 253-262, 2010.
[15] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE, “Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases”, IEEE Transactions on Knowledge and Data Engg., VOL. 25, NO.8, AUGUST 2013.
[16] Frequent Itemset Mining Implementations Repository, http://fimi.cs.helsinki.fi/, 2012.