NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 3, ISSUE 1, JAN.-2016

INFREQUENT WEIGHTED ITEMSET MINING FOR TRANSACTIONAL DATABASES USING FREQUENT PATTERN GROWTH

Mr. D. H. Dewarde
Master of Engineering-II, Computer Engineering Department SPCOE, Savitribai Phule Pune University, Pune, Savitribai Phule Pune University, Pune
Assist.Prof. S. A. Kahate
Computer Engineering Department SPCOE, Savitribai Phule Pune University, Pune, Savitribai Phule Pune University, Pune

ABSTRACT

Mining Weighted Item sets from a transactional database includes to the discovery of itemsets with high utility like profits.Although a number of relevant techniques have been planned in recent years, they obtain the problem of producing a large number of candidate itemsets for high utility itemsets. Such a large number of candidate itemsets weakens the mining performance in terms of execution time and space requirement. In this paper we have concentrate on UP-Growth and UP-Growth+ algorithmwhich will overcome this impediment. This technique includes tree based data structure finding itemsets, UP-Tree for generating candidate itemsets with two scan of database. In this paper we extend the functionality of UP-Growth and UP-Growth+ algorithms on transactional database. The situation may become poorwhen the database contains lots of long transactions or long high utility itemsets. An appearing topic in the field of data mining is utility mining. The main goal of utility mining is to identify the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. This topic includes many applications in website click stream analysis, business promotion in chain hypermarkets, cross marketing in retail stores, online e-commerce management, and mobile commerce environment planning and even finding important patterns in biomedical applications.

Keywords – Data mining, High utility mining, Utility mining

INTRODUCTION

ITEMSET mining is preparatory data mining technique widely used for discovering valuable interaction among data. The first endeavor to perform itemset mining [1] was concentrate on discovering frequent itemsets, i.e., patternswhose noticed frequency of occurrence in the source data (the support) is given threshold. Frequent itemsets discovers application in a number of real-life contexts (e.g., market basket analysis [1], medical image processing [2], biological data analysis [3]). However, many universals approaches ignore the influence/interest of each item/transaction within the analyzed data. To allow considering items/transactions differently based on their relevance in the frequent itemset mining process, the notion of weighted itemset has also been described [4], [5], [6]. A weight is correlate with each data item and characterizes its local significance within each transaction. Consider, as an example, the data set expressed in Table 1. It includes six transactions each one relaxed of four distinct items weighted by the corresponding degree of interest (e.g., item a has weight 0 in tid 1, and 100 in tid 4). In the contexts which data center resource management and application profiling, transactions may represent CPU usage readings collected at a fixed sampling rate. For instance, tid 1 means that, at a fixed point of time (1), CPU b works at a high usage rate (weight 100), CPUs c and d have an intermediate usage rate (weights 57 and 71, respectively), while CPU a is temporarily idle (weight 0). The itemsets combined from the example data set can be exploited by a domain expert to profile system usage in order to perform resource allocation and system resizing. The implication of a weighted transaction, i.e., a set of weighted items, is commonly evaluated in terms of the corresponding item weights. Furthermore, the main itemset quality amplitude (e.g., the support) have also been bespokento weighted data and recycled for driving the frequent weighted itemset mining process. For instance, when appraisingthe support of {a,b} in the example data set reported in Table 1, the occurrence of b in tid 1, which describes a highly utilized CPU, should be treated separately from the one of a, which describes an idle CPU at the same instant. In [4], [5], [6] different approaches to incorporating item weights in the itemset support computation have been proposed. Note that they are all being spoken to frequent itemsetmining, while this work focuses on infrequent itemsets.In recent years, the scrutinyof the research community has also been focused on the infrequent itemset mining problem, i.e., discovering itemsets whose frequency of occurrence in the analyzed data is less than or equal to a maximum threshold. For instance, in [7], [8] algorithms for discovering minimal infrequent itemsets, i.e., infrequent itemsets that do not incorporate any infrequent subset, have been proposed. Infrequent itemset discovery is applicable to data coming from different application contexts such as (i) statistical disclosure risk assessment from census data and (ii) fraud detection [7], [8], [9]. However, traditional infrequent itemset mining algorithms still go through from their inability to take local item interestingness into account during the mining phase. In fact, itemset quality measures used in [4], [5], [15] to produce the frequent weighted itemset mining process are not directly applicable to achieve the infrequent weighted itemset (IWI) mining task effectively, while, on the other hand, state-of-the-art infrequent itemset miners are, to the better of our knowledge, unable to cope with weighted data. This paper addresses the finding of infrequent and weighted itemsets, i.e., the infrequent weighted itemsets, from transactional weighted data sets. To address this issue, the IWI-support measure is defined as a weighted frequency of occurrence of an itemset in the scrutinized data.

Table 1.1: An Example Database

TID	Transaction	TU
T1	(A,1) (C,10) (D,1)	10
T2	(A,2) (C,6) (E,2) (G,5)	20
T3	(A,2) (B,2) (D,6) (E,2) (F,1)	25
T4	(B,4) (C,13) (D,3) (E,1)	32
T5	(B,2) (C,4) (E,1) (G,2)	34
T6	(A,1) (B,1) (C,1) (D,1) (H,2)	35

Table 1.2: Profit Table

Profit	5	2	3	6	8	9	7	6
Item	A	B	C	D	E	F	G	H

Judging the utility of items by its presence in the transaction set is the earlier methods of ARM. The occurrence of an item is not enough to emulate the actual utility. One of the most challenging data mining tasks is the mining of high utility itemsets smoothly [4]. Discovering of the itemsets with high utilities is called as Utility Mining. Cost, quantity, profit or any other user expressions of preference can be used to identify the utility. For example, a computer system may be more profitable than a telephone in terms of benefits. Utility mining model was produced to define the utility of itemset [5]. The utility is a identify of how useful or profitable an itemset X is. The utility of an itemset X, i.e., u(X), describes the sum of the all utilities of itemset X in all the transactions containing X. An itemset X is called a high utility itemset if and only if u(X) greater than or equal to min_utility, where min_utility is a user defined minimum utility threshold. In short, finding all itemsets having utility higher than user defined minimum utility is the goal of high-utility itemset mining [6].

RELATED WORK

ChowdhuryFarhan Ahmed, Syed KhairuzzamanTanbeer,Byeong-SooJeong, and Young-Koo Lee granted three novel tree structuresfor efficiently perform transactional and interactive HUP mining [2]. The first tree structure is used toorganize the items according to their lexicographic order. It admitted as Transactional HUP Lexicographic Tree (IHUPL-Tree). It captures the transactional data without any restructuring operation. The next tree structure is the IHUP Transaction Frequency Tree (IHUPTF-Tree), whichis useful arranging items according to their transaction frequency in descending order. To curtail the mining time, the last tree, IHUP-Transaction-Weighted Utilization Tree (IHUPTWU-Tree) is designed. The structure of this tree is based on the Transactional Weighted Utility (TWU) value of items in descending order.

Alva Erwin, Raj P. Gopalan, and N. R. Achuthan, Advised CTU-PROL algorithm for efficient mining of high utilityitemsets from large datasets[3].These algorithms search the large TWU items in the transaction database. If data sets is too large to be held in main memory, the algorithm generates subdivisions using parallel projections and for each subdivision, a *Compressed Utility Pattern Tree (CUP-Tree)*is used to mine the complete set of high utility itemsets.Ifthe dataset is Limited, it built a single *CUP-Tree*for mining high utility itemsets.

Shankar S., Purusothaman T., Jayanthi, S., proposeda novel algorithm for mining high utility
itemsets [4]. This fast utility mining (FUM) algorithm finds all high utility itemsets withinthe disposed utility constraint threshold. The proposed FUM algorithm scales strong as the capacity of the transaction database increaseswith regard to the number of distinct items available.

R. Chan, Q. Yang, and Y. Shen, implied mining high utility itemsets[5]. They proposeda novel concept of top-K objective-directed data mining, which spotlights the top-K highutility closed patterns. They compute the concept of utility to capture highly desirable statistical patterns and present a levelwiseitemset mining algorithm. They create a new snipingstrategy based on utilities that allow pruning of low utility itemsets to be done by means of aanemicer but antimonotonic condition.

Ramaraju C., Savarimuthu N., implied a conditional tree based novel algorithm for high utility itemset mining [6]. A novel conditional high utility tree (CHUT) reduce the transactionaldatabases in two stages to compress search space and a new algorithm known as HU-Mine isproposed to mine complete set of high utility item sets.

Y. Liu, W. Liao, and A. Choudhary, implied a fast high utility itemsets mining algorithm [7]. They are proposed a Two-Phase algorithm to efficiently snip down the number ofcandidates and can precisely obtain the complete set of high utility itemsets. In the first aspect,they propose a model that applies the “transaction-weighted downward closure property” onthe search space to facilitate the identification of candidates. Recentphase identifies the high utility itemsets

Adinarayanareddy B., O. SrinivasaRao, MHM Krishna Prasad, implied improved UP-Growthhigh utility itemset mining [8]. The compact tree structure, Utility Pattern Tree i.e.UP-Tree, maintains the history of transactions and their itemsets. It expedites the mining performanceand avert scanning original database frequently. UP-Tree scansdatabase only two times to achieve candidate items and manage them in an efficient data structuredway. UP-Growth gates more execution time for Second Phase by prating UP-Tree. Hence theyproposed modified algorithm aiming to reduce the execution time by effectively identifying highutility itemsets.

P. Asha, Dr. T.Jebarajan, G. Saranya, implied a survey on efficient transactional algorithm for mining high utility itemsets in distributed and dynamic database [9]. The proposed system applies one master node and two slave nodes. Database is subdivided for every slave node for computation. The slave node measures the existence of each item. These data's are gathered in their local table. Then each slave node forwards these tables to master node. The Master Node maintain global table for gathering these data. Based on the minimum utility threshold value it measures the promising and unpromising itemsets.

PROBLEMDEFINATION

This paper addresses the problem of mining infrequent itemsets from transactional data sets. Let be I = {i1; i2; . . . ; in} a set of data items. A transactional data set T = {t1; t2; . . . ; tn} is a set of transactions, where each transaction is a set of items in Tand is characterized by a transaction ID (tid).

An itemset I is a set of data items [13]. More specifically,we denote as k-itemset a set of k items in I. The support (or occurrence frequency) of an itemset is the number of transactions containing I in T. An itemset I is infrequent if its support is less than or equal to a predefined maximum support threshold. Otherwise, it is said to be frequent [14]. An infrequent itemset is said to be *minimal*if none of its subsets is infrequent [7]. Given a transactional data set T and a maximum support threshold, the infrequent (minimal) itemset mining problem entails discovering all infrequent (minimal) itemsets from T [9].

Unfortunately, using the traditional support measure for driving the itemset mining process entails treating items and transactions equally, even if they do not have the same relevance in the analyzed data set. To treat items differently within each transaction we introduce the concept of weighted item as a pair hik; wqki, where ik 2 I is an item contained in tq 2 T, while wqk is the weight associated with ikthat characterizes its local interest/intensity in tq[4]. Concepts of weighted transaction and weighted transactional data set are defined accordingly as sets of weighted items and weighted transactions, respectively.

If algorithm produces huge number of candidate itemsets, then higher processing time it consumes.Utility pattern growth (UP-Growth) and UP-Growth+ algorithm [1]conquer this limitation. These algorithms fount high utility itemsets by using adequate strategies. The information of high utility itemsets is managed in a tree-based data structure named *utility pattern tree (UP-Tree)* such that candidate itemsets can be produced efficiently with only two scans of database.

MATHEMATICAL MODEL

Let S be the system that illustrate dataset i.e. set of transaction with profit of item as input to system with calculates of transaction utility, transaction weighted utility, recognized transaction utility, up tree construction, UP-growth algorithm, Improved UP-growth algorithm and this all produces output as high potential utility item sets.

Variable used in Mathematical Model:
S= (Tp, TU, TWU, RTU, Up tree, UP growth, UP growth+, PHUI)
S= System
Tp= Set of transaction with profit of each item TU=Transaction Utility
TWU=Transaction Weighted Utility RTU=Recognized Transaction Utility
UUI=Utility of Unpromising Item
UP tree= Utility Pattern Tree
UP growth= Utility Pattern Growth Improved
UP growth = Advanced Utility Pattern Growth PHUI= Potentially High Utility Item set.

Inputs:
Tp= {D, P} Where, Transactional DB,
D = {T1, T2...Tn} is a set of transactions, and for each transaction
Td (1< =d< =n) has a unique id, called Tid. Td ={(i1,q1) ,(i2,q2),…..,(in,qn)}
Each item ip(1<=p<=n) is associated with a quantity qp(ip,Td) that is, the purchased quantity of ip in Td. Profit DB, P = {pr(i1),pr(i2),……,pr(in)} min-util = user defined minimum threshold.

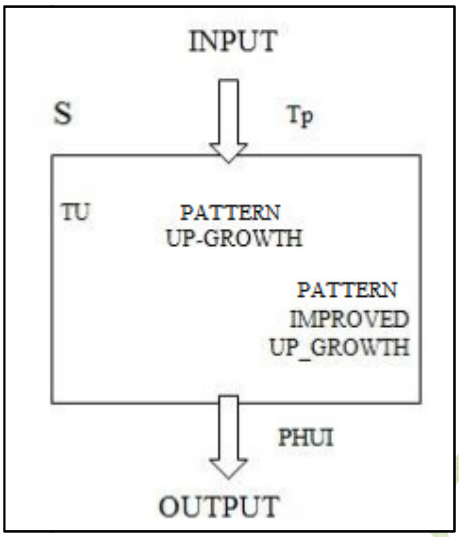


Fig. 1Input system architecture

Process:
1) TU=Σ ip Є Td [pr(ip)*qp(ip,Td)]
2) TWU (ip) = Σ TU Є ip
3) RTU (Td):= TU (Td) – UUI
4) Generate UP-tree and Mine it.with Node.name, Node. Count, Node.nu, Node.parent, Node.hlink.

Output: All Potential High Utility Itemsets in T_x

PROPOSED SYSTEM

The major three steps of proposed methods are:
1) Double scan the database with construction of the global UP-Tree.
2) Generate the potential high utility itemsets using UP-Growth and UP Growth+Algorithm.
3) From the set of PHUIs to identify actual high utility item set.

A) Construction of UP Tree

Our planned system will work on transactional database i.e. deletion or insertion of one or more records from database will recognize on database history. To achieve this it uses the existing approach [16]. Proposed system can avoidavoidable or repetition of calculations by using previous results when a database is updated, or when the threshold value is replaced.

After the original database is regenerated by removing the unpromising items and theirutilities from the database, UP-Tree is generated. A compact tree structure, UP-Tree is used for simplify the mining performance and avoid scanning original database repeatedly. It will also sustain the transaction information and high utility itemsets. UP-Tree is constructed to facilitate the mining performance and avert scanning original database repeatedly; we use a compact tree structure, named UP-Tree (Utility Pattern Tree), to sustain the information of transactions and high utility item sets are maintained in the UP-Tree. By
Implementing two strategies, the overestimated utilities stored in the nodes of global UP-Tree are minimized. Two scan of database is required from the construction of UP-Tree.

Table 2: Reorganized Database

TID	Transaction	RTU
1	(C,1)(A,1)(D,1)	12
2	(C,6)(E,2)(A,2)	26
3	(C,1)(E,1)(A,1)(B,2)(D,6)	28
4	(C,3)((E,1)(B,4)(D,3)	24
5	(C,2)(E,1)(B,2)	13

System Architecture is as follows:

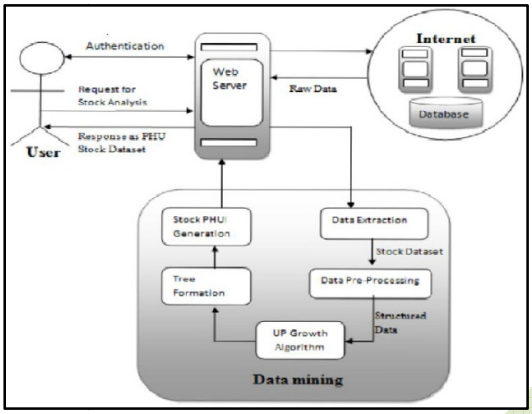


Fig. 2 System architecture

B) Discarding Global Unpromising Items

By using two scans of database it is achievable to construct the global UP-Tree. TU of eachtransaction is computed in the first scan; along with that TU of each transaction is also figure out. An item whose TWU is less than minimum utility threshold is said to be unpromising item. The following figure will show how to remove unpromising items and their utilities from transactions and TUs. Suppose the minimum utility is 40, items having TWU< minimum utility will be discarded, here unpromising items are F and G. After that Reorganized Transactions will be generated with their TU.

C) Discarding Global Node Utilities

Next step is to eliminating global node utilities. Actually it is done by contemplating utilities ofdescendant nodes during the construction of global UP-Tree.The nodes having utility that are near to the root of a global UP-Tree are further reduced by Implementing strategy DGN, It will be useful when the database contains large number of long transactions in other ways if a transactions contains more items, more utilities can be discarded by DGN.

D) Global UP-Tree with Strategies DGU and DGN

Construction of a global UP-Tree is done by using two database scans. Each transaction's TUis computed in the first scan; together with that, each item's TWU is also calculated. Thus, we can collect promising items and unpromising items. DGU is applied finally taking all unpromising item.

After that the transactions are reconstructed by pruning the unpromising items and sorting the remaining promising items in a fixed order. Any ordering can be dependent on any scheme such as alphabetic, support, or TWU order. Each transaction after the above reorganization is called a reconstructed transaction. Then a function Insert Reconstructed Transaction is called to apply DGN during constructing a global UP-Tree.

E) UP-Growth Mining Method

UP Growth algorithm uses two methods namely Discarding Local Unpromising Items and Decreasing Local Node Utilities to identify high utility itemsets from the UP Tree. By these two methods, embellished utilities of itemsets can be decreased and thus the number of PHUIs can be further reduced.

F) Discarding Local Unpromising Items (DLU)

It consists of three steps. First step is organizing conditional pattern base, second step is generate conditional trees and the final step is mine patterns from the conditional tree. Since actual utilities are not implemented in global UP-tree, DGU, DGN strategies cannot be enforced. To avert this problem items actual utilities of each transaction will be maintained in each node, but this will cause improper utilization of memory. To avoid this problem DLU, DLN strategies are used. In both cases minimum item utility table is used for keeping all promising item's minimum item utilities. In order to reduce local promising item's utilities minimum item utilities can be used. For that predictedvalue for each local unpromising item is subtracted from path utility of an extracted path which is illustrated by the following diagram.

G) Discarding Local Node utilities (DLN)

Information about the item below to the particular item will not be consist of in the UP-Tree, so that it is possible to eliminate the utilities of descendant nodes related to item. We use minimum utilities of item to estimate damaged utilities because we don't know the actual utilities of descendant nodes.

H) UP Growth+ Mining Method

In UP Growth+ Algorithm, minimal node utility in each path is determined to make the estimated pruning values closer to the real utility values of the pruned itemsets in database. Minimal Node Utility for each node in the global UP Tree is captured during the manufactured of a global UP Tree. Minimal Node Utility is estimated in each node of the global UP Tree.

EXPERIMENTAL RESULTS

For the proposed algorithms, Two methods UPT&UPG and UPT&UPG+ are developed that are composed of the proposed methods UP-Tree(DGU and DGN) and UP-Growth (DLU and DLN) and the proposed methods UP-Tree and UP-Growth+ (DNU, and DNN), respectively.

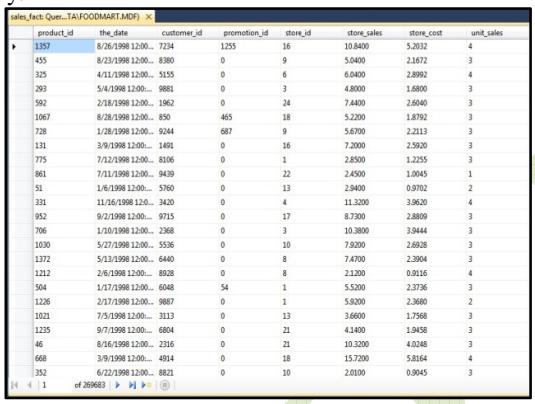


Fig. 3 Load Dataset

In phase II, these algorithms analyze high utility itemsets by scanning the database which contains no unpromising items. By applying DGN strategy, the node utilities of the nodes in a global UP-Tree are adequatelydecreased since the utilities of their descendants are discarded. By implementing DLU strategy, local unpromising items are discarded from the paths of conditional pattern base and their minimum item utilities are eliminated from the path utilities. By applying DLN, the node utilities of the nodes in a local UP-Tree are decreased since they removed the minimum item utilities of their descendants.

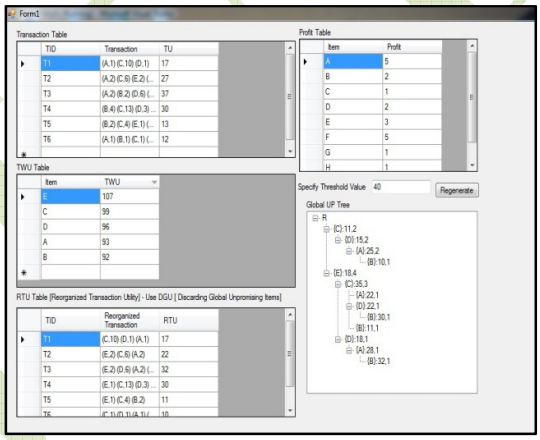
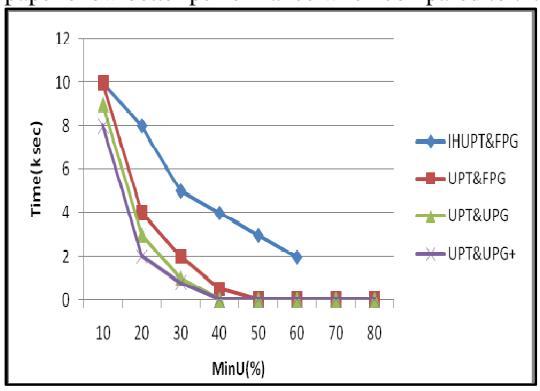


Fig.4 Generated Result

The methods resolved in this paper are compared with previous methods. The results presented in the following figures expose that the performance of the methods of this paper show better performance when compared to that of previous ones.



As can be seen in Figure 4, it is evident that the horizontal axis represents number of candidates while the vertical axis Serve the runtime. Out of all the methods the UPT & UPG+ method has higher performance. This is due to the reason that the number of generated candidates is taken into consideration.

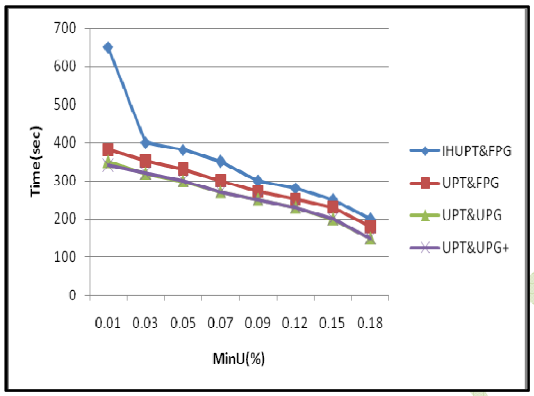


Fig 5 – Performance comparison on sparse data set (Food mart)

As can be seen in Figure 5, it is evident that the horizontal axis serves of number of candidates while the vertical axis represents the runtime. Out of all the methods the UPT & UPG+ method has higher performance. The runtime is proportional to number of candidates as expected. The reason trailing is that the overhead of scanning is very high

CONCLUSION

Especially we concentrate on the mining of high utility item sets from given data sources. We made experiments on the algorithms as conferred in [8] such as UP-Growth and UP-Growth and UP Growth+. In the experiments, a data structure is implemented by name UP-Tree for holding the high utility item sets. Such item sets can be created effectively from UP-Tree with less number of database scans. The overestimated utility has been weakening using many strategies as part of utility mining. The experiments are made using both real time and synthetic datasets. The performance of the implemented algorithms are presented and correlated with previous methods. The results reveal that the methods constructed in this paper outperform the methods of previous ones found in the literature. From this it is understood that the prototype application can be used in the real world applications.

REFERENCES

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, IEEE "EfficientAlgorithms for Mining High Utility Itemsets from Transactional Databases" IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 8, August 2013
[2] ChowdhuryFarhan Ahmed, Syed KhairuzzamanTanbeer,Byeong-SooJeong, and Young-Koo Lee, Member, IEEE "Efficient Tree Structures for High Utility Pattern Mining inTransactional Databases" IEEE Trans. Knowledge and Data Engineering, Vol. 21, Issue 12, pp. 1708-1721, December 2009.
[3] Alva Erwin, Raj P. Gopalan, and N.R. Achuthan, "Efficient Mining of High Utility Itemsets from Large Datasets",In Proc. of PAKDD 2008.
[4] Shankar, S.; Purusothaman, T.; Jayanthi, S. "Novel algorithm for mining high utilityitemsets" International Conference on Computing, Communication and Networking, Dec. 2008.
[5] Raymond Chan;Qiang Yang; Yi-Dong Shen, "Mining high utility itemsets" In Proc. of Third IEEE Int'l Conf. on Data Mining ,November 2003.
[6] Ramaraju, C.,SavarimuthuN. "A conditional tree based novel algorithm for high utilityitemset mining", International Conference on Data mining, June 2011.
[7] Ying Liu, Wei-keng Liao,AlokChoudhary "A Fast High Utility Itemsets Mining Algorithm"In Proc. of the Utility-Based Data Mining Workshop, 2005.
[8] AdinarayanareddyB ,OSrinivasaRao, MHM Krishna Prasad, "An Improved UP-GrowthHigh Utility Itemset Mining" International Journal of Computer Applications (0975-8887) Volume 58-No.2, November 2012.
[9] P. Asha, Dr. T. Jebarajan, G. Saranya, "A Survey on Efficient Transactional Algorithm for Mining High Utility Itemsets in Distributed and Dynamic Database" IJETAE Journal, Vol.4, Issue 1, January 2014.
[10] M.J. Zaki, "Scalable Algorithms for Association Mining," IEEE Trans. Knowledge and Data Eng., vol. 12, no. 3, pp. 372-390, May 2000.