International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 8, Issue 7 (September 2013), PP.19-31

Utility Fp-Tree: An Efficient Approach for Mining of Weighted Utility Itemsets

C.Saravanabhavan[1], R.M.S.Parvathi[2]
[1]*Research Scholar & Asst Professor, Department of CSE, Kongunadu College of Engineering and Technology, Tamil Nadu, India.*
[2]*Principal & Professor, Department of CSE, Sengunthar College of Engineering ,Tamil Nadu, India*

Abstract:- Conventional association rules mining cannot satisfy the demands emerging from certain real applications. By regarding the diverse values of distinct items as utilities, utility mining concentrates on discovering the itemsets with high utilities. In recent times, high utility pattern mining is one of the most significant research issues in data mining because of its ability to account for the non-binary frequency values of items in transactions and diverse profit values of each item. In this paper, we have presented an efficient tree structure for mining of high utility itemsets. At first, we have developed a novel utility frequent-pattern tree structure, an extended tree structure for storing crucial information about utility itemsets. Then, we have utilized the pattern growth methodology for mining the complete set of utility patterns. The efficiency of the high utility itemsets mining is achieved with two major concepts: 1) a large database is compressed into a smaller data structure as well as the utility FP-tree avoids repeated database scans, 2) our proposed FP-tree-based utility mining utilize the pattern growth method to avoid the costly generation of a large number of candidate sets in which it dramatically reduces the search space. Experimental analysis is carried out on our mining trees structure concept using different real life datasets. The performance evaluation of our proposed approach is efficient in mining high utility itemsets.

Keywords:- Data mining, Frequent pattern mining, FP-tree, FP-growth, Utility itemset mining, Utility FP-tree.

1. INTRODUCTION

Data mining can be regarded as an algorithmic process that takes data as input and yields patterns, such as classification rules, itemsets, association rules, or summaries, as output [4]. For example, frequent itemsets can be discovered from market basket data and used to derive association rules for predicting the conditional probability of the purchase of certain items, given the purchase of other items [1, 2, 3]. Mining frequent patterns in large transactional databases is a highly researched area in the field of data mining. The different existing frequent pattern discovering algorithms suffer from various problems regarding the computational and I/O cost, and memory requirements when mining large amount of data [14]. Frequent pattern mining discovers patterns in transaction databases based only on the relative frequency of occurrence of items without considering their utility [18]. For many real world applications, however, utility of itemsets based on cost, profit or revenue is of importance. The utility mining problem is to find itemsets that have higher utility than a user specified minimum. Unlike itemset support in frequent pattern mining, itemset utility does not have the anti-monotone property and so efficient high utility mining poses a greater challenge [19].

An emerging topic in the field of data mining is Utility Mining. The main objective of Utility Mining is to identify the itemsets with highest utilities, by considering profit, quantity, cost or other user preferences. Mining High Utility itemsets from a transaction database is to find itemsets that have utility above a user-specified threshold. Itemset Utility Mining is an extension of Frequent Itemset mining, which discovers itemsets that occur frequently. In many real-life applications, high-utility itemsets consist of rare items [13, 26, 27, 28, 29]. Rare itemsets provide useful information in different decision-making domains such as business transactions, medical, security, fraudulent transactions and retail communities. For example, in a supermarket, customers purchase microwave ovens or frying pans rarely as compared to bread, washing powder, soap. But the former transactions yield more profit for the supermarket. Similarly, the high-profit rare itemsets are found to be very useful in many application areas [12]. A retail business may be interested in identifying its most valuable customers i.e. who contribute a major fraction of overall company profit [11].

Frequent pattern mining techniques treat all items in the database equally by taking into consideration only the presence of an item within a transaction. However, the customer may purchase more than one of the same item, and the unit price may vary among items. High utility pattern mining approaches have been proposed to overcome this problem. As a result, it becomes a very important research issue in data mining and knowledge discovery. On the other hand, incremental and interactive data mining provides the ability to use previous data structures and mining results in order to reduce unnecessary calculations when the database is updated, or when the minimum threshold is changed. Most of the frequent pattern mining algorithms, including Apriori [1, 2, 3], FP-growth [6], H-mine [8], and OP (OpportuneProject) Algorithms [10], mine all frequent itemsets. These algorithms have good performance in case that the pattern space is sparse and the value of support threshold is set high. However, when the value of support threshold drops low, the number of frequent itemsets goes up dramatically, and the performance of these algorithms deteriorates quickly because of the generation of a huge number of patterns.

One of the currently fastest and most popular algorithms for frequent item set mining is the FP-growth algorithm [6]. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme: in a preprocessing step delete all items from the transactions that are not frequent individually, i.e., do not appear in a user-specified minimum number of transactions. Then select all transactions that contain the least frequent item (least frequent among those that are frequent) and delete this item from them. Recurse to process the obtained reduced (also known as projected) database, remembering that the item sets found in the recursion share the deleted item as a prefix. On return, remove the processed item also from the database of all transactions and start over, i.e., process the second frequent item etc. In these processing steps the prefix tree, which is enhanced by links between the branches, is exploited to quickly find the transactions containing a given item and also to remove this item from the transactions after it has been processed [5, 19].

In this paper, we have designed an efficient tree structure for mining the high utility itemsets efficiently. Here, we have proposed a novel utility FP- tree, an extended tree structure for storing essential information about utility frequent patterns. In addition to, we have utilized the mining technique used in the standard FP-growth algorithm for mining the complete set of utility patterns. The efficiency of the high utility pattern mining is realized by considering the two important thoughts. One is, a large database is compressed into a compact data structure as well as the FP-tree avoids repeated database scans and the other one is our proposed FP-tree-based utility mining utilizes the pattern growth method to avoid the costly generation of a large number of candidate sets in which it dramatically reduces the search space. The experimentation is carried out on different datasets in order to find the efficiency of the proposed approach in mining of high utility itemsets when compared with the standard FP-Growth algorithm.

The rest of the paper is organized as follows: a brief review of the recent related research is presented in Section 2. The proposed methodology for mining of high utility itemsets is provided in Section 3. The experimental results of the proposed approach on different datasets are given in Section 4. Finally, the conclusions are summed up in Section 5.

LITERATURE SURVEY

Numerous researches are available in the literature to perform the mining of frequent pattern based on the utilities. In recent times, developing approaches for utility based pattern mining has gained enormous importance in real life applications. A brief review of some of the recent significant research is presented here.

Jianying Hu and Aleksandra Mojsilovic [21] have presented an algorithm for frequent item set mining that identifies high-utility item combinations. In contrast to the traditional association rule and frequent item mining techniques, the goal of the algorithm is to find segments of data, defined through combinations of few items (rules), which satisfy certain conditions as a group and maximize a predefined objective function. They formulated the task as an optimization problem, presented an efficient approximation to solve it through specialized partition trees, called High-Yield Partition Trees, and investigate the performance of different splitting strategies. The algorithm has been tested on "real-world" data sets, and achieved very good results. Yu-Chiang Li *et al.* [23] have proposed the Isolated Items Discarding Strategy (IIDS), which can be applied to any existing level-wise utility mining method to reduce candidates and to improve performance. The most efficient known models for share mining are ShFSM (Fast share measure) and DCG (Direct Candidates Generation), which also work adequately for utility mining as well. By applying IIDS to ShFSM and DCG, the two methods FUM and DCG+ were implemented, respectively. For both synthetic and real datasets, experimental results revealed that the performance of FUM and DCG+ was more efficient than that of ShFSM and DCG, respectively.

Guo-Cheng Lan and Vincent S. Tseng [24] proposed a kind of pattern named Chain-Store High Utility Pattern that contains not only individual profit and quantity of items but also common selling periods and stores of items in a multi-stores environment. Moreover, they proposed a method named CS-Mine (Chain-Store High Utility Pattern Mine) for discovering the patterns efficiently. The CS-Mine algorithm needs only to scan the database twice and it can effectively filter out a large number of unnecessary itemsets with the filtration mechanism. Through a series of experiments, the method was shown to deliver excellent performance under varied system conditions. Hong Yao and Howard J. Hamilton [25] have proposed a utility based itemset mining, which permits users to quantify their preferences concerning the usefulness of itemsets using utility values. The usefulness of an itemset was characterized as a utility constraint. That is, an itemset is interesting to the user only if it satisfies a given utility constraint. They showed that the pruning strategies used in previous itemset mining approaches cannot be applied to utility constraints. Two algorithms for utility based itemset mining are developed by incorporating these pruning strategies. The algorithms were evaluated by applying them to synthetic and real world databases. Experimental results showed that the algorithms were effective on the databases tested.

Chun-Wei Lin *et al.* [17] have proposed the high utility pattern (HUP) tree for utility mining. They further handle the problem of maintaining the HUP tree in dynamic databases. A HUP maintenance algorithm has been proposed for efficiently handling new transactions. The algorithm can reduce the cost of re-constructing the HUP tree when new transactions are inserted. Experimental results also showed that it indeed executes faster than the batch maintenance algorithm and generates nearly the same tree structure as the batch one. The maintenance algorithm can thus achieve a good trade-off between execution time and tree complexity. Chowdhury Farhan Ahmed *et al.* [16] have proposed a tree-based candidate pruning technique HUC-Prune (high utility candidates prune) to efficiently mine high utility patterns without level-wise candidate generation-and-test. It exploits a pattern growth mining approach and needs maximum three database scans in contrast to several database scans of the existing algorithms. Extensive experimental results showed that the technique was very efficient for high utility pattern mining and it outperforms the existing algorithms.

There are many algorithms for mining high utility itemsets by pruning candidates based on estimated utility values, and based on transaction-weighted utilization values. These algorithms aim to reduce search space. Besides, candidate pruning based on transaction-weighted utilization value is better than other strategies. Bac Le *et al.* [9] have proposed TWU-Mining, an algorithm based-on WIT-tree for improving the cost of time and search space. Experiments showed that the algorithm was more effective on the testing databases. Chowdhury Farhan Ahmed *et al.* [22] have proposed a tree structure, called IIUT (incremental and interactive utility tree), to solve the problems together. It uses a pattern growth mining approach to avoid the level-wise candidate set generation-and-test problem, and it can efficiently capture the incremental data without any restructuring operation. Moreover, IIUT has the "build once mine many" property and therefore it is highly suitable for interactive mining. Experimental results showed that the tree structure is very efficient and scalable for incremental and interactive high utility pattern mining.

3. PROPOSED METHODOLOGY FOR MINING OF HIGH UTILITY ITEMSETS

3.1 Problem description

The problem of mining utility itemsets is discussed in this subsection and some basic definitions are also described as follows. Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items and $D = \{t_1, t_2, ..., t_n\}$ be a transaction database where the items of each transaction t_i is a subset of I. The utility of item i_p in transaction t_q, denoted by $U(i_p, t_q)$ is defined as $Iu(i_p, t_q) \times Eu(i_p)$. Let an itemset X be a subset of I. The utility of X in transaction t_q, denoted by $U(X, t_q)$ is defined as $U(X, t_q) = \sum_{i_p \in X} U(i_p t_q)$. The task of high utility mining is to find all items that have utility above a user-specified min_utility. Since utility is not anti-monotone, the concept of Frequency Weighted Utility (FWU) is used to prune the search space of high utility itemsets.

Definition: The internal utility or local transaction utility value $Iu(i_p, t_q)$ represents the quantity of item i_p in transaction t_q. The external utility $Eu(i_p)$ is the unit profit value of item i_p. ***Definition:*** Utility $U(i_p, t_q)$ is the quantitative measure of utility for item i_p in transaction t_q defined by $U(i_p, t_q) = Iu(i_p, t_q) \times Eu(i_p)$.

Definition: The utility of an itemset X in transaction t_q, $U(X, t_q)$, is defined by $U(X, t_q) = \sum_{i_p \in X} U(i_p, t_q)$; where $X = \{i_1, i_2, ... i_k\}$ is a k-itemset, $X \subseteq t_q$ and $1 \le k \le m$.

3.2. Proposed Algorithm for Mining High Utility Itemsets

The frequent pattern Mining problem does not take into account the quantity or an associated weight such as price or profit of an item but it represents only the occurrence of each item in a transaction by a binary value. But, quantity and weight are important factors for solving real world decision problems that intends to maximize the utility of an organization. Hence, all itemsets that have utility value greater than a user specified minimum utility value are identified by high utility itemset mining. Both local transaction utility and external utility contribute to the utility of an item. Identifying high utility item sets which drive a major share of the overall utility is the objective of utility mining [20]. High utility pattern mining approaches have been proposed to overcome this problem. As a result, it becomes a very important research issue in data mining and frequent pattern mining. In this paper, we have presented an efficient approach for mining the high utility itemsets from the utility FP-tree structure. The procedure used for mining high utility items is demonstrated by two important steps.

1. Construction of utility FP-tree
2. Mining of high utility itemsets from utility FP-tree

1. Construction of utility FP-tree

In general, the construction of the FP-tree and the mining patterns from the FP-tree are the major important steps in the frequent pattern tree algorithm. Similar way, the proposed approach also contains these two steps, where the utility FP-tree is constructed using the frequency weighted utility rather than the frequency value. In addition to, the mining process utilized pattern growth methodology, where the support is computed based on the frequency weighted utility rather than the frequency. In this section, we describe the construction process of our proposed utility FP- tree structure based on the frequency weighted utility. For the discussion of the proposed algorithm, we have explained with a simple example to easily understanding the entire step including tree construction and mining process. Table 1 provides an example of a transaction database and Table 2 gives the unit profit for each item belonging to the transaction database.

Table 1. Example of a transaction database

Item / TID	**A**	**B**	**C**	**D**
01	2	1	0	1
02	3	0	2	0
03	0	3	2	0

Table 2. Example of a utility table

Item	**Profit ($)**
A	2
B	1
C	5
D	1

Step 1: Ordering of transaction

Before the construction of the FP-tree, the ordering of the transaction is important, since each path of the FP-tree follows it. Here, the ordering is mainly depends on the frequency weighted utility FWU of an item. At first, the items are sorted out in descending order for each transactions based on the frequency weighted utility value and the items which are less than the minimum utility value min_util is removed from the transactions. ***Definition:*** Frequency-weighted utility FWU of an item i_p, denoted by $FWU(i_p)$, is computed using the transaction frequency (TF), transaction weightage and the external utility.

$$FWU(i_p) = \frac{TF_{(i_p)} * TW_{(i_p)} * EU_{(i_p)}}{U_F}$$

Definition: The transaction frequency of an item $TF_{(i_p)}$ denotes the actual number of occurrences of i_p in all the transactions. ***Definition:*** Transaction weightage $TW_{(i_p)}$ is defined as the overall quantity of the item i_p in all transactions. ***Definition:*** Utility factor U_F is the overall sum of the profit of each items presented in the database. ***Definition:*** If i_p is said to be a frequency weighted utility item, it should satisfy the condition, $FWU(i_p) \geq min\_util$.

Example: Let us consider the items present in table 1. An item 'A', the FWU is computed as follows: the $TF(i_p)$ of an item 'A' is 2, the $TW_{(i_p)}$ is 5 and the profit records for that item is 2. Also, the total profit value, called utility factor is found out by 9 in this case. Now, $TWU(A)$ = 2.22, $TWU(B)$ = 0.77, $TWU(C)$ = 4.44 and $TWU(D)$ = 0.11. In this case, we have taken the $\min\_util$ value as 0.3 and choose the items which are greater than the $\min\_util$ value. Based on these computed utility values, the items are re-ordered. The transactions with sorted items are taken out for illustrating the construction of the utility FP-tree. The ordered transactions are shown in table 3.

Table 3. The ordered transactions with sorted large items

TID	**Frequent Items**	
01	A	B
02	C	A
03	C	B

Step 2: Inserting of transactions into utility FP-tree

In this step, the utility FP-tree is constructed by inserting the ordered transactions so that it only necessitates two scans on the transaction database as well as works in a divide and conquer way. In the first scan, the proposed algorithm generates the 1-length frequent weighted utility items based on the frequent weighted utility measure. In the second scan, the transaction database is compressed into a utility FP- tree. The utility FP-tree is a tree structure which is defined as follows,

- ➢ Utility FP-tree consists of one root labeled as a "root" node and a set of item prefix sub-trees as the children of the root, and a utility-item header table.
- ➢ The nodes present in the item prefix tree consist of three fields: i) item-name, ii) frequency weighted utility value and iii) node-link. The item-name records the item present in the node, frequency weighted utility value records the utility measure represented by the portion of the path reaching this node, and node-link links to the next node in the utility FP-tree carrying the same item-name, or null if there is none.

Example: The insertion of each transaction is processed as follows: In the first transaction, the frequent weighted utility items (A, B) are being processed. The results after the first transaction are shown in fig. 1. The root of the tree is initially fixed as null. Then, this transaction is attached as a first branch of the root node. Each node of the branch is attached with the frequency weighted utility values.

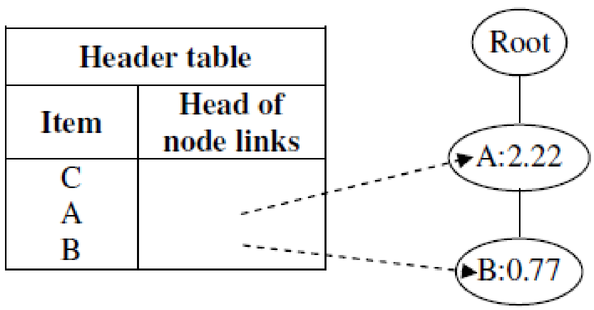

Fig 1. The Utility FP-tree after the first transaction is processed

Subsequently, the next transaction containing frequent weighted utility items (C, A) is processed. Here, the items does not contain any prefix path in the utility FP-tree after executing the first transaction so that the new nodes (C: 4.44) is attached with the root node as its child. Also, the other new node (A: 2.22) is created and linked with the child of (C: 4.44). The results after the second transaction are shown in fig. 2.

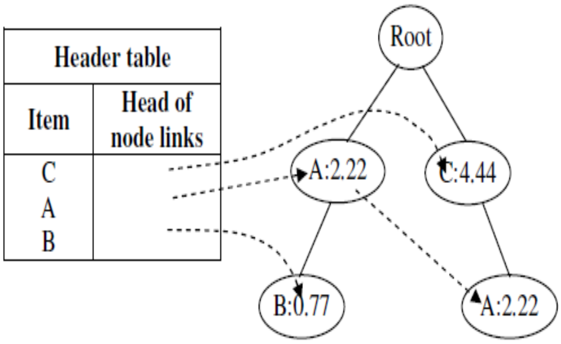

Fig 2. The Utility FP-tree after the second transcation is processed

For processing of third transaction, the path <"C" "B"> shares the same prefix "C" with the Utility FP-tree so that the count of the node (C: 4.44) is incremented by 4.44 as it shares the common prefix and created a new node (B: 0.77) is attached to the node (C: 8.88) as its child node. The results after the third transaction are shown in fig. 3.

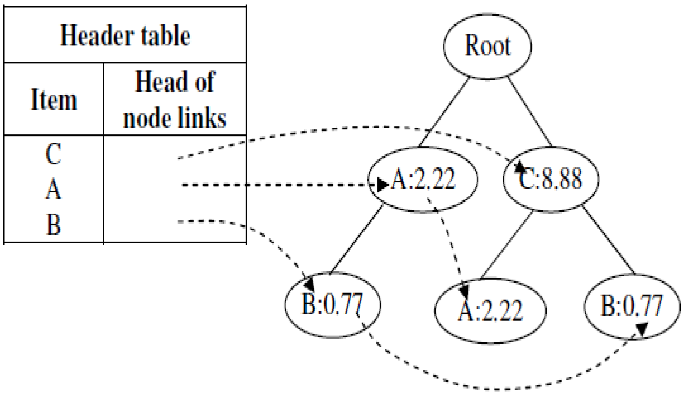

Fig. 3. The utility FP-tree after the third transaction is processed

After the Utility FP-tree is constructed from a transaction database, a mining process is executed to determine the large items. Utility FP-tree derives the utility itemsets directly from the utility FP-tree and do not necessitate generation of candidate itemse**t**s for mining. It recursively processes the utility items one by one and bottom-up with regard to the Header Table. By constructing a conditional utility FP-tree for each utility item, high utility itemsets are mined recursively from it. This process is executed until all the items in the utility FP-tree get processed.

2. Mining a high utility itemsets from utility FP-tree

The next major step is to examine the mining process based on the constructed utility FP-tree as shown in fig. 3. The mining process of utility itemsets from the utility FP-tree based on the pattern growth methodology [30] is explained as follows.

Step 1: Generating conditional utility pattern base and Conditional utility FP-tree

After the utility FP-tree is constructed from an ordered transaction database, a mining procedure starts with the generation of the conditional utility pattern base and the conditional utility FP-tree. As the utility FP-tree constructed in the fig. 3, we have generated a conditional utility pattern base and the conditional pattern tree. Here, we start with the mining process from the bottom of the nodes of the utility FP-tree and their corresponding prefix paths are extracted from it. Then, their relevant utility pattern base and conditional utility FP-tree are generated in order to mine 2-length utility patterns.

Example: At first, we process the item "B", which is the bottom item present in the header table so that two prefix paths existed for item B is extracted. For an item B, the conditional pattern base is (A: 0.77) and (C: 0.77), which are the prefix paths of the item "B". Then, the conditional utility FP tree is generated for the item "B". Again, the conditional pattern base is generated for the superset of "A" i.e., "AB" and "AC" but no prefix paths having this sequence so it generates NULL path. Subsequently, the next items A and C are processed. The conditional pattern base for item “A” is (C: 2.22) and the conditional pattern base of the patterns “C” is null. The conditional pattern-bases and the conditional FP-trees generated are summarized in Table 4.

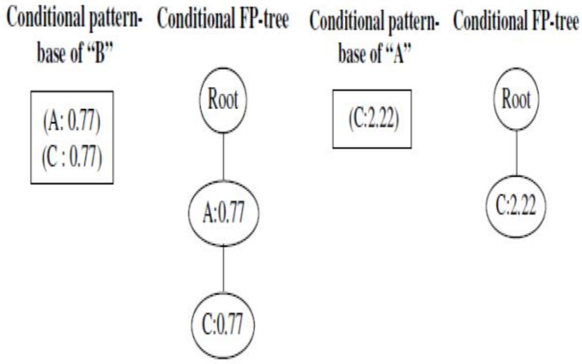

Fig 4. Mining of FP-tree by creating conditional pattern-bases

Table 4. Mining frequent patterns by creating conditional pattern-bases

Item	Conditional pattern- base	Conditional FP-tree
B	{(A:0.77), (C:0.77))	{(A,C)}/B
A	{(C:2.22)}	{(C)}/A
C	φ	φ
BA	φ	φ
BC	φ	φ
AC	φ	φ

Step 2. Mining utility patterns

After the generation of the conditional utility FP-tree, the high utility patterns are mined from it based on the minimum support threshold. Here, utility patterns are mined recursively from the conditional utility FP tree so that all length patterns having the frequency weighted utility greater than the minimum threshold is obtained. The patterns are said to be frequent weighted utility patterns if the support of those is greater than the min_util.

Example: The results obtained for the sample database given in table 1 is shown in the table 5. The frequent weighted utility patterns are {(C: 8.88) (A: 4.44), (B: 1.44), (AB: 0.77), (CB: 0.77), (CA: 2.22)}.

Table 5. Frequent weighted utility patterns for a sample database

Frequent patterns	
C: 8.88	
A: 4.44	CA: 2.22
B: 1.44	AB: 0.77 CB: 0.77

EXPERIMENTAL RESULTS

This section presents the experimental results of our proposed approach for effectual mining of high utility itemsets on transaction database. The proposed approach has been implemented in Java (jdk 1.6). The data utilized in our experimental results are real-world data obtained from various fields and widely-accepted synthetic data. We have tested our approaches in two different datasets, namely T10I4D100K and Retail [7, 15]. For real life data, we have used Retail dataset, a real market basket data and synthetic data T10I4D100K is obtained from the IBM dataset generator.

T10I4D100K: This dataset contains 100,000 transactions and 870 distinct items. T10I4D100K denotes the Average size of the transactions (T), Average size of the maximal potentially large itemsets (I) and the number of transactions (D). ***Retail Dataset:*** This dataset contains 88,162 transactions and 16,470 distinct items. This dataset was donated by Tom Brijs and contains the (anonymized) retail market basket data from an anonymous Belgian retail store. Table 6 gives the tow test dataset descriptions.

Table 6: Test dataset Description

Dataset	Size	No. of transactions	No. of Items
T10I4D100K	3.93MB	100,000	870
Retail	4.07MB	88,162	16,470

4.1 Performance Evaluation

Experimentations on both the real life and synthetic datasets are undertaken by the two pattern mining algorithms such as, proposed algorithm and FP-growth. The two algorithms are utilized to analyze the Standard FP-growth algorithm with our proposed approach for effectual mining of high utility itemsets. Here, different results are obtained by changing the support values and analyzed the results for the two datasets.

1) Retail

The experimental results are taken by varying the support values on the retail datasets. The obtained results are plotted as graphs as shown in figures 5, 6, 7, 8 and 9 that shows the performance of the two approaches on retail dataset in effectual mining of high utility itemsets. Here, the performance of our proposed approach is evaluated by different support values (normalized between 0 to 1) and the corresponding generated length of the patterns. By analyzing the plotted graphs, the performance of our proposed approach produces better results than the standard FP-growth algorithm. As the support value varies, the number of generated frequent patterns gets reduced in our proposed approach than the FP-growth algorithm by different length of patterns. In Fig 5, the number of patterns generated by varying the support thresholds of 1 length patterns gets constrained from the FP-growth algorithm. Likewise, the fig 6, 7 and 8 shows the generated number of patterns of 2 length , 3 length and 4 length patterns respectively of different supports of both the algorithms. But, as shown in fig 9, no 5 length patterns are produced by our proposed approach compared with the FP-growth algorithm which generated a limited number of patterns.

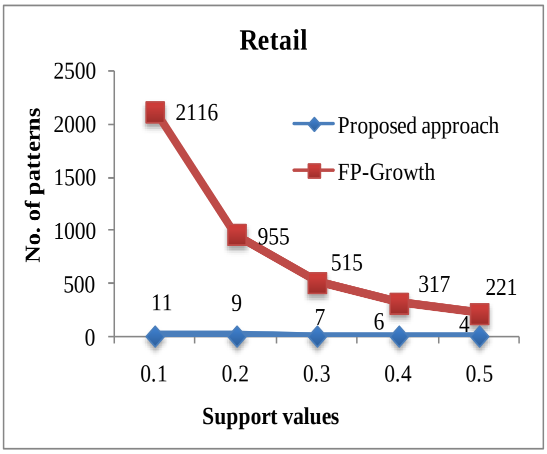

Fig 5. No. of frequent patterns (1-length) generated using various support threshold

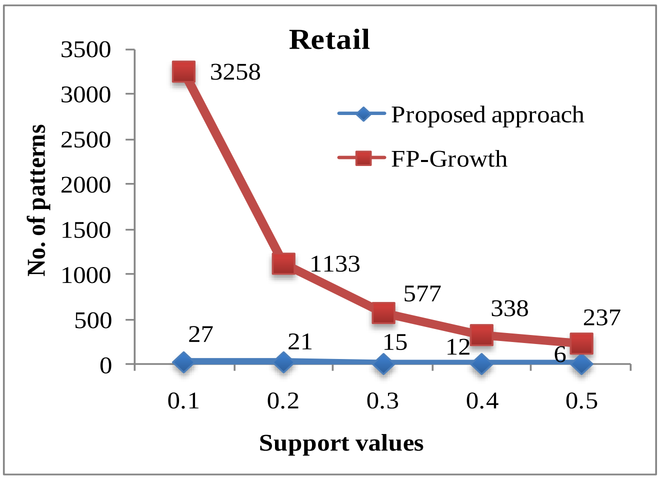

Fig 6. No. of frequent patterns (2-length) generated using various support threshold

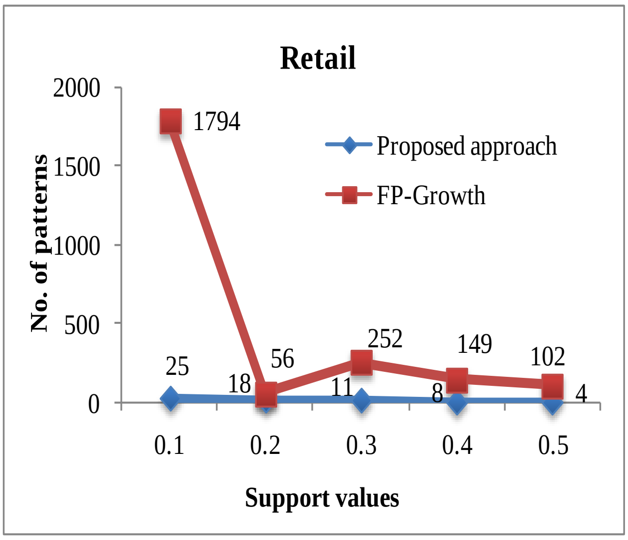

Fig 7. No. of frequent patterns (3-length) generated using various support threshold

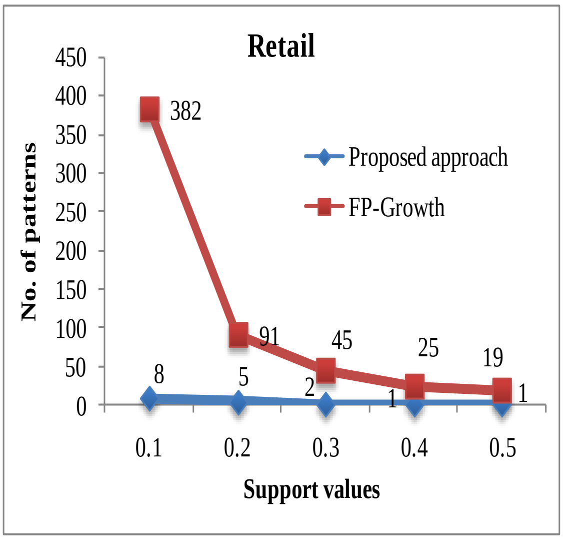

Fig 8. No. of frequent patterns (4 -length) generated using various support threshold

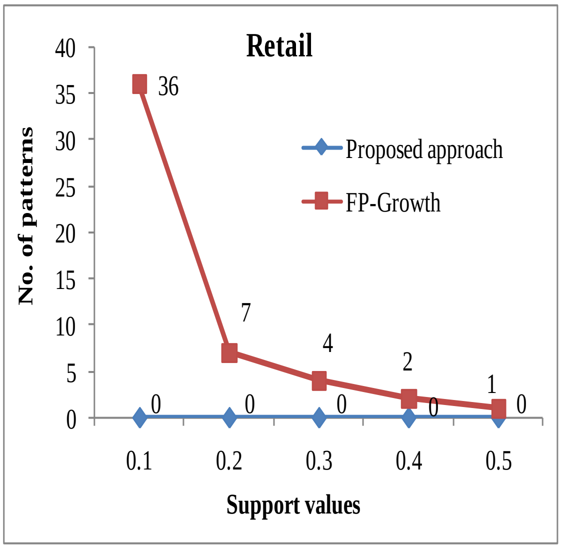

Fig 9. No. of frequent patterns (5-length) generated using various support threshold

2) T10I4D100K

The experimental results are taken by varying the support values on the T10I4D100K datasets. The attained results are plotted as graphs as shown in figures 10, 11, 12, 13 and 14 that shows the performance of the two approaches on T10I4D100K dataset in effectual mining of high utility itemsets. Here, the performance of our proposed approach is examined by different support values and the corresponding generated number of patterns with the length of the patterns. By examining the plotted graphs, the performance of our proposed approach produces better results than the standard FP-growth algorithm. The number of frequent patterns generated in different lengths gets reduced in our proposed approach than the FP-growth algorithm by diverse support thresholds. The number of patterns generated in different lengths patterns is restricted from the FP-growth algorithm with the support threshold of 0.5 is shown in fig 10. Similarly, the fig. 11 and fig. 12 shows the generated number of patterns of varying lengths with the support thresholds 0.6 and 0.7 respectively of both the algorithms. Also, our proposed approach and the FP-growth algorithm produces better results with the support values of 0.8 and 0.9 is shown in fig. 13 and fig. 14 respectively.

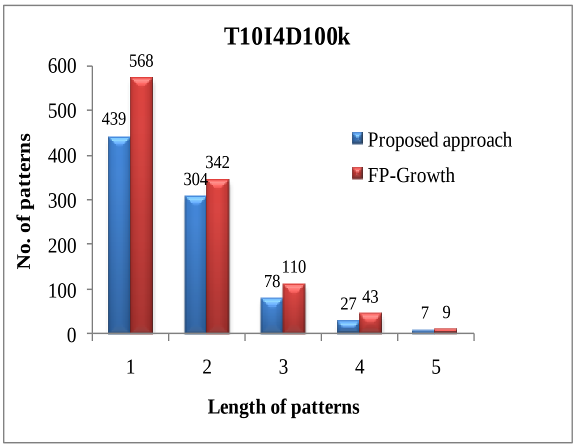

Fig 10. No. of frequent patterns generated of varying lengths with support threshold= 0.5

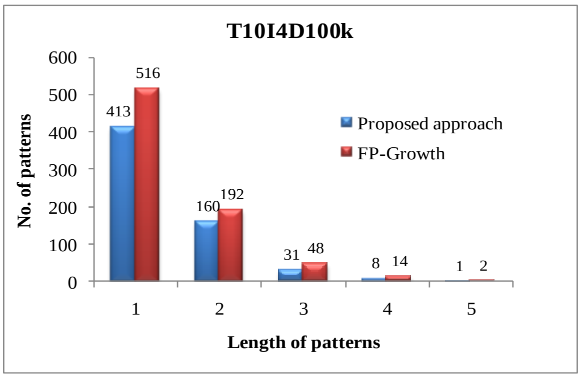

Fig 11. No. of frequent patterns generated of varying lengths with support threshold= 0.6

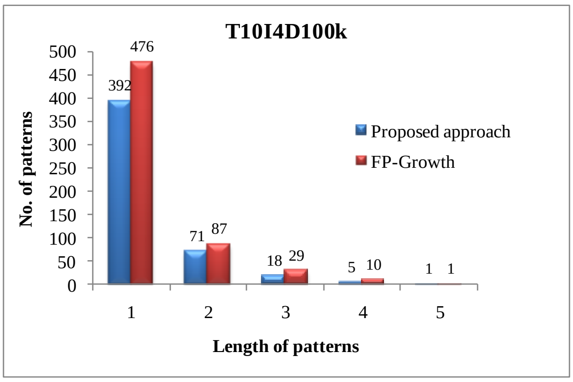

Fig 12. No. of frequent patterns generated of varying lengths with support threshold= 0.7

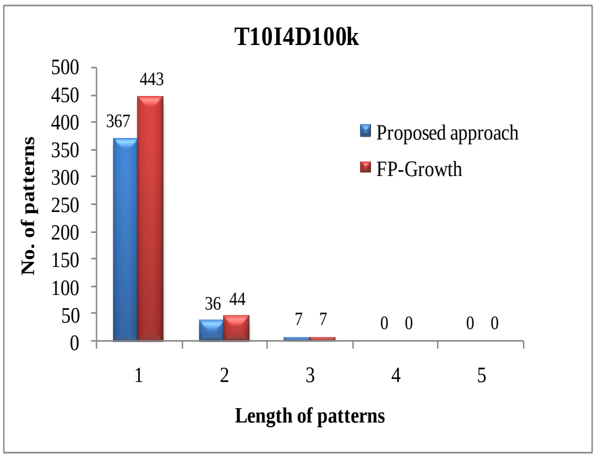

Fig 13. No. of frequent patterns generated of varying lengths with support threshold= 0.8

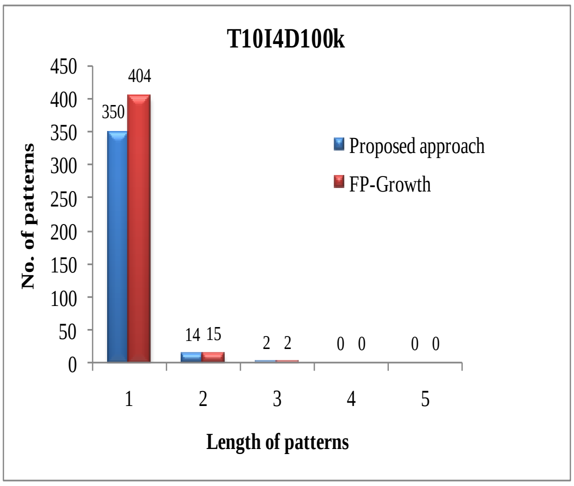

Fig 14. No. of frequent patterns generated of varying lengths with support threshold= 0.9

CONCLUSION

In this paper, we have presented a novel utility FP-tree, an extensive tree structure for storing essential information about frequent patterns for mining the high utility itemsets. We have utilized the standard FP-growth algorithm for mining the complete set of frequent patterns by pattern growth. The efficiency of the high utility pattern mining is recognized by two important thoughts. One is the construction of the utility FP-tree and the other one is the mining of utility itemsets from the utility FP-tree. Our proposed utility FP-tree-based pattern mining utilized the pattern growth method to avoid the costly generation of a large number of candidate sets in which it dramatically reduces the search space. The experimentation was carried out on our proposed approach using real life datasets and the results showed that the proposed approach is effective on the tested databases.

REFERENCES

[1]. Agrawal R., Imielinski T., and Swami, A.N., “Mining association rules between sets of items in large databases”, in Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, D.C., pp: 207-216, 1993.

[2]. Agrawal R. and Srikant, R., “Fast algorithms for mining association rules”, in Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile, pp: 487-499, 1994.

[3]. Mannila, H., Toivonen, H., and Verkamo, A.I., “Efficient algorithms for discovering association rules”, in Proceeding of the AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, pp: 181-192, 1994.

[4]. Hong Yao, Howard J. Hamilton, Liqiang Geng, "A Unified Framework for Utility Based Measures for Mining Itemsets", in Proceedings of the Int. Workshop on Utility-Based Data Mining, Philadelphia, PA, pp. 28-37, 2006.

[5]. Christian Borgelt, "An Implementation of the FP-growth Algorithm", in Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations, Chicago, Illinois, pp: 1 - 5, 2005.

[6]. J. Han, H. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation”, ACM SIGMOD Record, Vol: 29, No: 2, pp: 1 - 12, 2000.
[7]. M. Sulaiman Khan, Maybin Muyeba, Frans Coenen, "A Weighted Utility Framework for Mining Association Rules", Second UKSIM European Symposium on Computer Modeling and Simulation, Liverpool, pp: 87 - 92, 2008.
[8]. J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, “Hmine: Hyper-structure mining of frequent patterns in large databases”, in Proceedings of IEEE International Conference on Data Mining, San Jose, California, USA, pp: 441–448, 2001.
[9]. Bac Le, Huy Nguyen, Tung Anh Cao, Bay Vo, "A Novel Algorithm for Mining High Utility Itemsets", First Asian Conference on Intelligent Information and Database Systems, Dong Hoi, pp: 13 - 17, 2009.
[10]. J. Liu, Y. Pan, K. Wang, and J. Han, “Mining frequent item sets by opportunistic projection”, in Proceedings of the 8th ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining, pp: 229–238, July 2002.
[11]. Jyothi Pillai, O.P.Vyas, "Overview of Itemset Utility Mining and its Applications", International Journal of Computer Applications, Vol: 5, No.11, August 2010.
[12]. V. S. Tseng, C.J. Chu, T. Liang, “Efficient Mining of Temporal High Utility Itemsets from Data streams”, in Proceedings of Second International Workshop on Utility-Based Data Mining, August 20, 2006.
[13]. G.C.Lan, T.P.Hong and V.S. Tseng, “A Novel Algorithm for Mining Rare-Utility Itemsets in a Multi-Database Environment”, the 26th workshop on combinatorial mathematics and computation theory, pp: 293- 302, 2008.
[14]. Renata Ivancsy, Istvan Vajk, "Fast Discovery of Frequent Itemsets: a Cubic Structure-Based Approach", Informatica, Vol: 29, pp: 71–78, 2005.
[15]. Bay Vo, Huy Nguyen, Bac Le, "Mining High Utility Itemsets from Vertical Distributed Databases", in proceedings of the International Conference on Computing and Communication Technologies, Da Nang, pp: 1-4, 2009.
[16]. Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong and Young-Koo Lee, "An Efficient Candidate Pruning Technique for High Utility Pattern Mining", Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Vol: 5476, pp: 749-756, 2009.
[17]. Chun-Wei Lin, Tzung-Pei Hong, Wen-Hsiang Lu, "Maintaining High Utility Pattern Trees in Dynamic Databases", 2010 Second International Conference on Computer Engineering and Applications (ICCEA), Bali Island, pp: 304 - 308, 2010.
[18]. Mingjun Song, and Sanguthevar Rajasekaran, "A Transaction Mapping Algorithm for Frequent Itemsets Mining", IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 4, pp. 472-481, 2006.
[19]. Alva Erwin, Raj P. Gopalan, N.R. Achuthan, "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach", in proceedings of the Seventh IEEE International Conference on Computer and Information Technology, pp: 71-76, Aizu-Wakamatsu, Fukushima, 2007.
[20]. Y. Liu, W.-K. Liao, and A. Choudhary, "A Fast High Utility Itemsets Mining Algorithm", in Proceedings of the KDD Workshop on Utility-Based Data Mining, Chicago Illinois, 2005.
[21]. Jianying Hu, and Aleksandra Mojsilovic, "High-utility pattern mining: A method for discovery of high-utility item sets", Pattern Recognition, Vol: 40, No: 11, pp: 3317-3324, 2007.
[22]. Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, Young-Koo Lee, "Mining high utility patterns in incremental databases", in Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, pp: 656-663, 2009.
[23]. Yu-Chiang Li, Jieh-Shan Yeh, Chin-Chen Chang, "Isolated items discarding strategy for discovering high utility itemsets", Data & Knowledge Engineering, Vol: 64, No: 1, pp: 198-217, 2008.
[24]. Guo-Cheng Lan and Vincent S. Tseng, "A Novel Approach for Discovering Chain-Store High Utility Patterns in a Multi-Stores Environment", The Second ACM International Workshop Mining Multiple Information Sources, Las Vegas, USA, 2008.
[25]. Hong Yao and Howard J. Hamilton, "Mining itemset utilities from transaction databases", Data & Knowledge Engineering, Vol: 59, No: 3, pp: 603-626, 2006.
[26]. Jyothi Pillai, O.P. Vyas, Sunita Soni, Dr. Maybin Muyeba, "A Conceptual Approach to Temporal Weighted Itemset Utility Mining", International Journal of Computer Applications, Vol. 1, No. 28, pp. 0975 - 8887, 2010.
[27]. Younghee Kim, Wonyoung Kim, Ungmo Kim, "Mining Frequent Itemsets with Normalized Weight in Continuous Data Streams", Journal of Information Processing Systems, Vol.6, No.1, pp. 79-90, March 2010.

[28]. Jieh-Shan Yeh , Yu-Chiang Li and Chin-Chen Chang, "Two-Phase Algorithms for a Novel Utility-Frequent Mining Model", Lecture Notes in Computer Science, Springer, Berlin/ Heidelberg, vol. 4819, pp.433-444, November 2007.

[29]. Chun-Jung Chua, Vincent S. Tseng, and Tyne Liang, "An efficient algorithm for mining temporal high utility itemsets from data streams", Journal of Systems and Software, Vol. 81, No. 7, pp. 1105-1117, July 2008.

[30]. Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Vol: 8, pp: 53–87, 2004.

	**Mr.C.Saravanabhavan** received his M.C.A degree from K.S.R. College of Technology, Namakkal in 2003 and M.Tech. degree in Information Technology from Sathyabama University, Chennai in 2007. He is currently pursuing Ph.D., in Computer Science and Engineering from Anna University of Technology, Coimbatore. Presently he is working as an Assistant Professor in Kongunadu College of Engineering and Technology, Trichy.
	**Dr.R.M.S.Parvathi** has completed her Ph.D., degree in Computer Science and Engineering in 2005 in Bharathiyar University, Tamilnadu, India.Currently she is a Principal and Professor, Department of Computer Science and Engineering in Sengunthar College of Engineering , Tamilnadu, India, She has completed 22 years of teaching service. She has published more than 28 articles in International / National Journals. She has authorized 3 books with reputed publishers. She is guiding 20 Research scholars. Her research areas of interest are Software Engineering, Data Mining, Knowledge Engineering, and Object Oriented System Design.