International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 10 6013 - 6018

# Discovering High Utility Itemsets using Hybrid Approach

Amol J. Gosavi,
Department of Computer Engg,
MET's Institute of Engg., Nashik,
Savitribai Phule Pune University,
*gosaviamol@gmail.com*

Dr. K. V. Metre,
Department of Computer Engg.
MET's Institute of Engg., Nashik
Savitribai Phule Pune University,
*kvmetre@yahoo.co.in*

***Abstract*—**Mining of high utility itemsets especially from the big transactional databases is time consuming task. For mining the high utility itemsets from large transactional datasets multiple methods are available and have some consequential limitations. In case of performance these methods need to be scrutinized under low memory based systems for mining high utility itemsets from transactional datasets as well as to address further measures. The proposed algorithm combines the High Utility Pattern Mining and Incremental Frequent Pattern Mining. Two algorithms used are Apriori and existing Parallel UP Growth for mining high utility itemsets using transactional databases. The information about high utility itemsets is maintained in a data structure called UP tree. These algorithms are not only used to scans the incremental database but also collects newly generated frequent itemsets support count. It provides fast execution because it includes new itemsets in tree and removes rare itemset from a utility pattern tree structure that reduces cost and time. From various Experimental analysis and results, this hybrid approach with existing Apriori and UP-Growth is proposed with aim of improving the performance.

***Keywords*—** *Apriori, Itemsets, Utility mining, UP Growth.*

*****

## I. Introduction

Data mining refers to extracting or mining knowledge from large amounts of data. Thus, data mining should have been more appropriately named knowledge discovery in databases. Finding frequent patterns in data mining from large databases is very important and useful task in many applications over the past few years. The primary goal is to discover hidden standards, unpredicted tendency in the data. Data mining is concerned with analysis of large volumes of data to automatically locate fascinating relationships which in turn leads to better understanding of the basic processes. Fusion of techniques are used in data mining activities from database, artificial intelligence, statistics and machine learning technologies. This includes bioinformatics, genetics, medicine, clinical research, education, retail and marketing research. In frequent pattern mining the relative importance of each item is not considered, for that weighted association rule mining was proposed, thus utility mining becomes prominent in data mining field.

Utility mining is one of the most challenging tasks in the mining of high utility itemsets. Identification of the itemsets with high utilities is called as Utility Mining. The utility can be measured as per the users priority.

The limitations of frequent or rare itemset mining motivated researchers to conceive a utility based mining perspective, which allows a user to appropriately express his or her perspectives concerning the usefulness of itemsets as utility values and then find itemsets with high utility values higher than a min_util threshold. In utility based mining the term utility refers to the quantitative representation of user preference i.e. according to an itemsets utility value is the measurement of the importance of that itemset in the user's perspective.

With the increase of IT, the size of the databases created by the organizations due to the availability of low-cost storage and the evolution in the data capturing technology is also enhancing. The organization sectors include retail, petroleum, telecommunications, utilities, manufacturing, transportation, credit cards, insurance, banking and many others, extracting the valuable data, it is necessary to explore the databases completely and efficiently.

From existing methods, first found the Potential High Utility Itemsets (PHUIs) and then an additional database scan is performed for identifying utilities that generate a huge set of PHUIs and also degrade the performance as long transactional databases are used or by setting low thresholds and this forms a challenging task against performance with time. To resolve these problems two existing algorithms named UP (utility pattern) Growth and UP Growth + along with compact data structure i.e. tree structure (UP Tree) are introduced for discovering high utility itemsets from transactional databases efficiently. The UP-Tree algorithm stores all transactions in the database as a tree using two scans that occupies lots of memory and takes time for process of tree searching.

To address the issues of search space, candidate generation and to decrease overestimated utility along with enhancing the performance of the utility mining, this hybrid technique is being used which will provide the ability to develop several strategies to decrease overestimated utility and enhance the performance of utility mining and to improve data mining results to user by improving frequent count.

This approach is a combination of algorithmic results of existing data mining approaches for the purpose. The objective of this approach is to provide a generalized hybrid framework and get more accurate and actionable knowledge, no matter whether the data is large, complex, heterogeneous or distributed.

## II. Literature Survey

In this section different methods are presented to mine high utility itemsets effectively.R. Agrawal and R. Srikant proposed Fast Algorithms for Mining Association Rules. [2] Mining associations rules using algorithm like Apriori, which is the innovator for mining association rules from large databases efficiently.

Cai , Tao et al. proposed the weighted association rules [4] and weighted items [3]. The framework of weighted association rules does not have downward closure property; the Mining performance was not improved. To resolve these issues, Tao et al. proposed the weighted downward closure property concept [4]. Weighted support can reflect the impor-tance of an itemset, It also maintain the downward closure property during the mining process after using transaction weight.

J. Han, J. Pei, and Y. Yin, proposed [5] Pattern growth-based association rule mining algorithms such as FP-Growth. Pattern fragment growth mines the complete set of frequent patterns using the FP-growth. FP-Growth is widely appre-ciated to achieve desirable performance than Apriori-based algorithms as it finds frequent itemsets without generating any candidate itemset and scans database just twice.

Li et al. proposed the Two- Phase algorithm [6] which combined with two mining phases. An Apriori based level wise method used to enumerate High Transactional Weighted Utility Itemsets (HTWUTs). Candidate itemsets with length k are created from length (k-1) HTWUIs. Their TWUs (Transactional Weighted Utility) are calculated by scanning the database once in each pass. The complete set of HTWUIs is collected in phase I after above steps, and in phase II, high utility itemsets i.e. HTWUIs are identified with an extra database scan, but it generates too many candidates, for reducing the count Li et al. [7] proposed an isolated items discarding strategy (IIDS). The number of candidate itemsets for HTWUIs in phase I can be reduced by pruning isolated items during level-wise search.

Ahmed et al. [8] proposed Incremented High Utility Pattern (IHUP) which is a tree-based algorithm. IHUP-Tree is used to maintain information about itemsets and their utilities. Each node of an IHUP-Tree made up of name of item, Transactional Weighted Utility value and a support count. Prabha S et al. [9] presented survey of closed frequent pat- terns mining algorithm, features and problems, the paper also provides the details of various algorithms with comparisons.

K Vanitha , R. Santhi[10] presented survey paper on effective association rule mining in large database. The performance is examined by observing execution time of several instances and confidence related to dataset of supermarket. FP-growth method is efficient and scalable. It is faster than the Apriori algorithm. Many researchers had works in the field of correlated frequent pattern mining. Omiecinski [11] introduced three alternative interestingness measures, called any-confidence, all-confidence, and bond for mining associations.

Y.K. Lee et al. [12] presented the concept of all-confidence to discover interesting patterns, although both of them defined a pattern which satisfies the given minimum all-confidence as a correlated pattern.

Z. Zhou et al. [13] proposed mining of all independent patterns and correlated patterns synchronously in order to get more information from the results by comparing independent patterns with correlated patterns. Moreover, an effective algorithm is developed for discovering both independent patterns and correlated patterns synchronously, especially for finding long independent patterns by using the downward closure property of independence. In mining process to discover both associated and correlated patterns they have combined association with correlation in at literature. A new interesting measure corr-confidence [14] was proposed for rationally evaluating the correlation relationships. This measure has proper bounds for effectively evaluating the correlation degree of patterns and is appropriate for mining long patterns. Actually, mining long patterns is more important because a practical transactional database may contain a lot of unique items.

## III. PROPOSED SYSTEM

Mine High utility itemsets accurately, effectively and efficiently.

As per the studied literatures, a huge set of PHUIs are generated frequently and degrade mining performance consequently. This condition may become worst in case of long transactions in dataset or setting low threshold value. More PHUIs are generated by algorithms that consume higher processing time and that its a challenging task. To overcome these issues efficient algorithms are presented. These methods incorporated the newest ideas and features in algorithms almost in all cases on both synthetic and real data set. This approach still needs to be improved in case of less memory based systems.[2]

Fig. 1. Shows processing of extracting utility itemsets by hybrid approach which has been discussed in this paper.

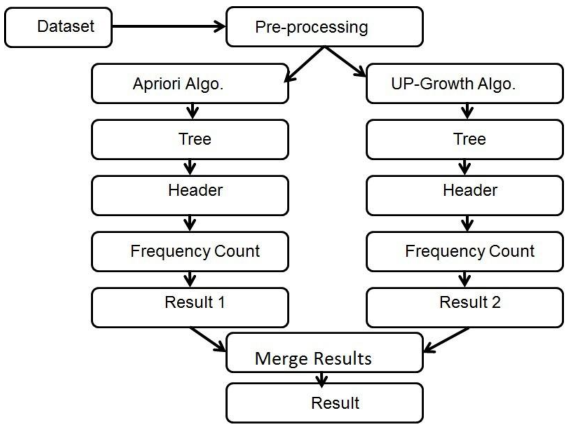


**Fig. 1 Working Flow of the system**

High utility itemsets, that may be classified or unclassified are taken into account which is an output from both Apriori and UP-Growth Algorithm, take it as an input for this Hybrid algorithmic strategy, which works on both categories i.e. classified and unclassified itemsets. Here different types of datasets are used as an input.

A series of experiments are done to compare the performance of the hybrid strategy with advanced utility mining algorithms. Fig. 1 shows system architecture of hybrid approach. Utility Pattern Growth (UP Growth), Apriori and Utility Pattern Tree (UP-Tree) which is a compact tree structure are use for discovering high utility itemsets and to preserve vital information related to utility patterns within databases.

*A. Mathematical Notations:*

*1) Problem Description:* Mine High utility itemsets accurately, effectively and efficiently.

Let S be a system of hybrid approach for discovering high utility itemsets such that

$S = \{D, P, U, A, F, T, H, X, Y, M, R, I | \Phi s\}$

TABLE I. PROBLEM DESCRIPTION

| Where | |
|---|---|
| D : Input dataset | $D = \{d_0, d_1..., d_n \| \Phi d\}$ |
| P : Pre-processing | $P = \{p_0, p_1..., p_n \| \Phi p\}$ |
| U : UP Growth | $U = \{u_0 \| \Phi u\}$ |
| A : Apriori | $A = \{a_0 \| \Phi a\}$ |
| F : Frequent Itemset | $F = \{f_0, f_1..., f_n \| \Phi f\}$ |
| I : Input Items | $I = \{i_0, i_1..., i_n \| \Phi i\}$ |
| T : Tree | $T = \{t_0, t_1 \| \Phi t\}$ |
| H : Header | $H = \{h_0, h_1..., h_n \| \Phi h\}$ |
| X : Result 1 via Apriori | $X = \{x_0 \| \Phi x\}$ |
| Y : Result 2 via UP Growth | $Y = \{y_0 \| \Phi y\}$ |
| M : Merge X and Y | $M = \{m_0 \| \Phi m\}$ |
| R : Result | $R = \{r_0 \| \Phi r\}$ |

*2) Activity:* Let $f_e$ be a rule of D into P such that for every dataset there must be preprocessing.

$f_e(D)| \rightarrow P$. e.g.$f_e(d_0)| \rightarrow \{p_0, p_1...p_n\} \in P$

Similarly for all module the relations are given with example in table II.

TABLE II. ACTIVITY-RELATIONS

| Activity | example |
|---|---|
| $f_e(P)\| \rightarrow U$ | e.g.$f_e(p_0)\| \rightarrow \{u_0\} \in U$ |
| $f_e(P)\| \rightarrow A$ | e.g.$f_e(p_0)\| \rightarrow \{a_0\} \in A$ |
| $f_e(U)\| \rightarrow T$ | e.g.$f_e(u_0)\| \rightarrow \{t_0, t_1\} \in T$ |
| $f_e(A)\| \rightarrow T$ | e.g.$f_e(a_0)\| \rightarrow \{t_0, t_1\} \in T$ |
| $f_e(T)\| \rightarrow H$ | e.g.$f_e(t_0)\| \rightarrow \{h_0, h_1...h_n\} \in H$ |
| $f_e(A)\| \rightarrow I$ | e.g.$f_e(a_0)\| \rightarrow \{i_0, i_1...i_n\} \in I$ |
| $f_e(H)\| \rightarrow F$ | e.g.$f_e(h_0)\| \rightarrow \{f_0, f_1...f_n\} \in F$ |
| $f_e(I)\| \rightarrow X$ | e.g.$f_e(i_0)\| \rightarrow \{x_0\} \in X$ |
| $f_e(I)\| \rightarrow Y$ | e.g.$f_e(i_0)\| \rightarrow \{y_0\} \in Y$ |
| $f_e(Y)\| \rightarrow M$ | e.g.$f_e(y_0)\| \rightarrow \{m_0\} \in M$ |
| $f_e(X)\| \rightarrow M$ | e.g.$f_e(x_0)\| \rightarrow \{m_0\} \in M$ |
| $f_e(M)\| \rightarrow R$ | e.g.$f_e(m_0)\| \rightarrow \{r_0\} \in R$ |

*3) Efficiency Issues:* The Sequential algorithm will execute the function n times to get the required result. The function may contain GetApriori, GetUpgrowth, GetTree commands or relevant algorithms as per efficiency issues.

## IV. ALGORITHMIC STRATEGY

*1) Apriori Algorithm :* Association rule generation is usually split up into two separate steps:

1) To find all frequent itemsets in a database apply minimum support.
2) To form rules these frequent itemsets and the mini- mum confidence constraints are used.

The second step is straight forward but first step needs more attention.

Let $I = \{i_1, i_2..., i_n\}$ be a set of n binary attributes called items. It is a difficult task to found frequent itemsets in a database as it involves searching all possible combinations of itemsets. The power set over I is the set of possible itemsets which has size (2n 1) excluding the empty set which is not a valid itemset. By using the downward-closure property of support(antimonotonicity) efficient search is possible, though the size of the powerset grows exponentially in the number of items n in I, all subsets and superset of itemsets are also frequent as well as an infrequent itemset. Capitalize on this property, Apriori and Eclat find all frequent itemsets.

1) Scans Input Dataset.
2) Build Bitwise Vertical Dataset.
3) Determine Frequent Itemset and Finding their support value.
4) Determine Closed Itemsets.
5) Data Items are sorted increasingly with respect to their Support value.
6) Decides whether it is dense or sparse dataset based on threshold value.

*2) UP Tree:* This type of compact tree structure is introduced to facilitate performance, it also avoid repeated scanning of original database and stores information about transactions and high utility itemsets.

Let N be the Node in UP Tree.

Up tree= {N.name, N.count, N.nu, N.parent, N.hlink}

N.name= Item name of nodes.

N.count= Support count of nodes.

N.nu= Node utility of nodes (Overestimated utility). N.parent= Parent of node of N.

N.hlink= Node link that points the node having item name as N.name.

UP-Tree by FP-Growth [5] is basic method for generating PHUIs after construction of UP-Tree. So many candidates are generated so UP Growth algorithm is introduced by adding two strategies named Discarding Global Unpromising (DGU) Items and Decreasing Global Node (DGN) utility [2].

patterns are generated in tree based algorithms by applying following steps

1) By tracing the paths in the original tree, generate conditional pattern bases.
2) By the information in conditional pattern bases, con- struct conditional trees or local trees.
3) Extract patterns from the conditional trees.

However, strategies DGU Items and DGN utility during constructing a Global UP-Tree cannot be applied into conditional UP-Trees since actual utilities of items in different transactions are not maintained in a global UP Tree. It cannot know actual utilities of unpromising items that need to be discarded in conditional pattern bases unless an additional database scan is performed. To overcome this problem, a native solution is to maintain items actual utilities in each transaction into each node of global UP-Tree. It needs large memory. In view of this module two strategies, named Discarding Local Unpromising (DLU) and Decreasing Local Node (DLN) during constructing a Global UP-Tree, that are applied in the first two mining steps.

For the two strategies, a minimum item utility table is maintained to keep minimum item utilities for all global promising items in the database that utilities are collected during first scan of database.

*3) UP Growth:* Minimum item utility table is used to reduce the overestimated utilities in UP Growth algorithm. Candidate count will be reduced here by using PHU model. Four different strategies are used like DGN, DGU, DLU and DLN.

Let $N_x$ be the node which records the item x in the path p in a UP-Tree and $N_x$ is composed of the items x from the set of transactions TIDSET ($T_x$).

min(x, p) be the minimal node utility of x in p.

1) Minimal node utility for each node can be acquired during the creation of a global UP Tree.
2) The conditional pattern is generated by tracing the paths in the original Catalog.
3) The Minimum item utility is evaluated by minimum utility threshold.
4) Find the local promising items in the catalog.
5) Reduce path utilities of the paths by DLU.
6) The path utility of an item is estimated.
7) The conditional Catalogs also called local Catalogs are constructed.
8) The reorganized path construction process is done too. This process is done after discarding the nodes.

Recursively call UP-Growth for generate whole set of PHUI.

Initially, UP-Growth ($T_R, H_R, null$) is called,

$T\text{-}_R$ =global UP-Tree,

$H\text{-}_R$ = global header table.

CPB =Conditional Pattern Base.

Let $T_x$ be the set of transactions containing an item x,

$i_k$ be the $k^{th}$ node in UP Tree.

Procedure: UP Growth ($T_x, H_x, X$)

1) For each entry $i_k$ in $H\text{-}_x$ do
2) Generate a PHUI Y = X $\cup$ $i_k$ ;
3) The estimate utility of Y is set as $i_k$'s utility value in $H_x$;
4) Construct Ys conditional pattern base Y-CPB;
5) Put local promising items in Y-CPB into $H_y$;
6) DLU is applied for reducing path utilities of the paths;
7) DLN applied and insert paths into Ty;
8) If $T_x$ /=null then call UP-Growth ($T_y, H_y, Y$);
*4)* End of For

*5) UP Growth+: :* By using DLU and DLN to decrease overestimated utilities of itemsets UP Growth is better per- formed then FP Growth. By eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes, the overestimated utilities can be closer to their actual utilities. UP Growth+ reduces overestimated utilities more effectively. Minimal node utilities in each path are used in UP Growth + to make the approximate pruning values closer to real utility values of the pruned items in database. Fewer candidates are generated here.

*5) IHUP ( Incremented High Utility Pattern) Algorithm :* A tree based structure called IHUP-Tree is used to maintain the information about item sets and their utilities. Each node of an IHUP-Tree consists of 1.An item name 2. TWU value 3. Support count.

IHUP algorithm has three steps:

1) Construction of IHUP.
2) Generation of HTWUI.
3) Identification of high utility item sets.

*6) Hybrid Algorithm:* The Objective of this approach is to provide a generalized hybrid framework and get more accurate and actionable knowledge, no matter whether the data is large, complex, heterogeneous or distributed. It is a combination of algorithmic results of existing data mining approaches. Here results of existing algorithms which consist of classified and unclassified itemsets are taken into account, work on both and extract more classified itemsets from unclassified one.

## I. IMPLEMENTATION DETAILS

This section presents the strategy of implementation.

Fig. 1 shows the overall steps undertaken in extracting utility itemsets by hybrid approach.

### *A. Input Dataset and preprocessing Module :*

Data Assembly also includes collecting of data, in these types of experiments for testing different types of datasets are collected from different website. The data is challenging due to the number of characteristics, the number of records, and the sparseness of the data (each records contains only small portion of items). In this experiment different dataset e.g. live, distributed, transactional, with different properties are selected to prove the efficiency of algorithm. e.g. Census data, Land registry, Retail, Zoo, Mashroom, pima.D38.N768.C2.

### *B. Apriori Module :*

The algorithm implies that frequent itemsets are mined through an iterative level-wise approach, based on candidate generation.

### *C. UP Growth Module :*

Mining high utility itemsets with a set of techniques for pruning candidate itemsets. Minimum item utility table is used here to reduce an excessively high estimated utilities. In UP-Growth+ algorithm negligible amount of node utilities in each path are used to make the approximate pruning values closer to real utility values of the pruned items in database.

### *D. Tree building Module :*

IHUP a tree based structure is used to keep the information about item sets and their utilities. Each node of an IHUP-Tree consists of 1. an item name 2. a TWU value 3. a support count.

### *E. Header generation Module :*

A table named header table is employed to facilitate the traversal of UP-Tree. Each entry records an item name, an excessively high estimate utility, and a link. The link points to the last occurrence of the node which has the same item as the entry in the UP-Tree.

### *F. Utility(Frequent)count Module :*

Count for utility patterns.

### *G. Hybrid Module :*

Fig. 2 shows classified and unclassified elements generated by both algorithms. This stage is concerned with High utility itemsets, that may be classified or unclassified are taken into account which is an output from both Apriori and UP-Growth Algorithm, take it as an input for this Hybrid algorithmic strategy, which works on both categories i.e. classified and unclassified itemsets and gives utility pattern extracted from given datasets.

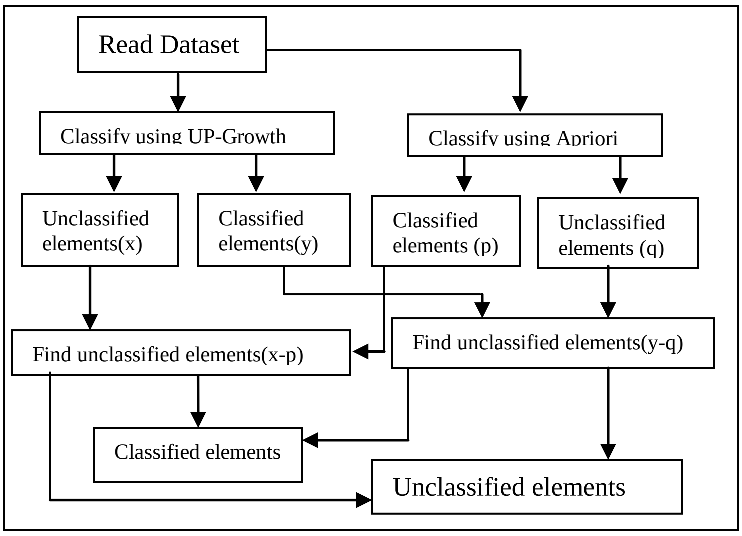


**Fig. 2 Hybrid Approach**

## V. EXPERIMENTAL EVALUATION AND RESULTS

Performance of proposed algorithms is evaluated here. The experiments were performed on Core2Duo with 3GB memory and on Microsoft Windows 10. The algorithms are implemented in java programming language. Real, Synthetic and transactional datasets are used in our experiments. Apriori, UP Growth and our Hybrid algorithms were executed on same min_util value as per the datasets to generate itemsets, HUI and PHUI. Table III and IV shows that our hybrid strategy gives better results than old ones.

TABLE III. EXECUTION-TIME IN ms.

| Datasets | Time | | |
|---|---|---|---|
| | Apriori | UP-Growth | Hybrid |
| DB_Utility (Dataset 1) | 15 | 22 | 20 |
| DB_UtilityIncremental(D2) | 141 | 15 | 6 |
| DB_UtilD2hup(Dataset3) | 422 | 100 | 75 |
| Mushroomnew(Dataset4) | 1094 | 594 | 521 |
| mushroom (Dataset5) | 766 | 375 | 318 |

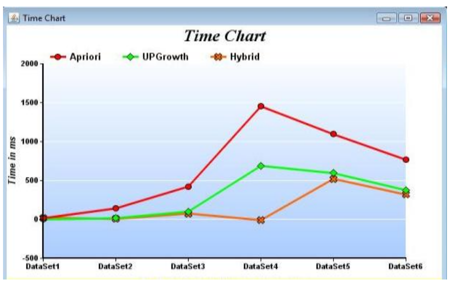


**Fig. 3 Execution Time**

TABLE IV. MEMORY-USES IN ms.

| Datasets | Space | | |
|---|---|---|---|
| | Apriori | UP-Growth | Hybrid |
| DB_Utility (Dataset 1) | 64.35 | 118.15 | 24.40 |
| DB_UtilityIncremental(D2) | 69.02 | 80.75 | 9.39 |
| DB_UtilD2hup(Dataset3) | 105.32 | 114.37 | 12.79 |
| Mushroomnew(Dataset4) | 156.03 | 194.38 | 54.91 |
| mushroom (Dataset5) | 82.39 | 99.80 | 21.53 |

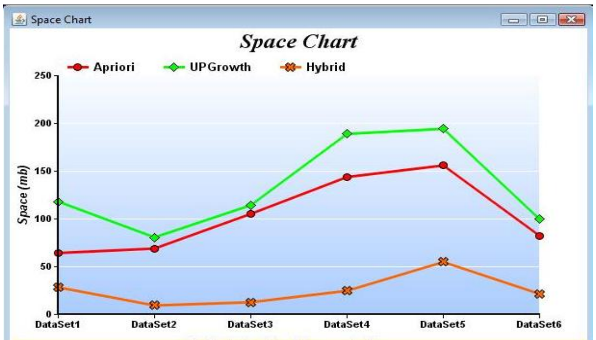


**Fig. 4 Memory usage**

## VI. CONCLUSION

Several strategies are proposed to decrease overestimated utility and enhance the performance of utility mining, Two efficient algorithms named UP Growth and Apriori are used for mining high utility itemsets from transactional datasets along with a data structure called as UP Tree for maintaining the information. The hybrid strategy is used to improve the performance by reducing both the search space and time with number of candidates. A hybrid approach will take the advantage of both algorithms. This system is aimed to reduce the size of normal implementation of any technique that has been used. Also, use of new data structure may recreate the tree by deleting all nodes of non- frequent itemsets after a scanning a specific percentage of database.

We have proposed mining method for frequent items using hybrid approach. Same method has been utilized for classification of various datasets with respective features provided by specific domain.

## REFERENCES

[1] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu,"Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases", *Knowledge and Data Engineering, IEEE Transactions onVolume: 25 , Issue: 10.1109, pp. 1772 - 1786, 2013 .*

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *In Proc. of the 20th Int'l Conf. on Very Large Data Bases (VLDB),pp. 487-499, 1994.*

[3] C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong,Mining Association Rules with Weighted Items, *Proc. Intl Database Eng. and Applications Symp. (IDEAS 98),pp. 68-77, 1998.*

[4] F. Tao, F. Murtagh, and M. Farid, Weighted Association Rule Mining Using Weighted Support and Significance Framework, *Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD 03), pp. 661-666 2003.*

[5] J. Han, J. Pei, and Y. Yin, Mining Frequent Patterns without Candidate Generation, *Proc. ACM-SIGMOD Intl Conf. Management of Data, pp. 1-12, 2000.*

[6] 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, *Proc. IEEE Eighth Intl Conf. on Data Mining, pp. 881-886, 2008.*

[7] Y.-C. Li, J.-S. Yeh, and C.-C. Chang, Isolated Items Discarding Strategy for Discovering High Utility Itemsets, *Data and Knowledge Eng., vol. 64, no. 1, pp. 198-217, Jan. 2008.*

[8] C.F. Ahmed, S.K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases, *IEEE Trans. Knowledge and Data Eng., vol. 21, no. 12, pp. 1708-1721, Dec. 2009.*

[9] Prabha S, Shanmugapriya S, and Duraiswamy K, A survey on Closed Frequent Pattern Mining,*International Journal on Computer Applications(0975-8887) Volume 63-No. 14, February 2013.*

[10] K. Vanitha And R. Santhi, 2011, Using Hash Based Apriori Algorithm To Reduce The Candidate 2- Item sets For Mining Association Rule, *Journal Of Global Research In Computer Science, Volume 2, No. 5*

[11] E. Omiecinski,Alternative interesting measures for mining association, Knowledge and Data Engineering, *IEEE Transactions on Volume:15 , Issue: 1 1041-4347 pp. 57 69, Jan.-Feb. 2003.*

[12] Y.K. Lee, W.Y. Kim, Y. D Cai, J. Han CoMine : efficient mining of correlated patterns *In proc. ICDM03 Proceedings of the Third IEEE International Conference on Data Mining Page 581, 2003.*

[13] Z. Zhou Mining Frequent Independent Patterns and Frequent Correlated Patterns Synchronously *5 th International Conference on Fuzzy Systems and Knowledge Discovery, 2008.*

[14] B. Liu, W. Hsu, and Y. Ma, Pruning and Summarizing the Discovered Association. *In Proc. 1999 ACM SIGKDD.*