ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

High Utility Itemset Mining from Transaction Database Using UP-Growth and UP-Growth+ Algorithm

Komal Surawase[1], Madhav Ingle[2]

PG Scholar, Dept. of Computer Engg., JSCOE, Hadapsar, Pune, India

Assistant Professor, Dept. of Computer Engg., JSCOE, Hadapsar, Pune, India.

ABSTRACT: Efficient discovery of itemsets with high utility like profits deals with the mining high utility itemsets from a transaction database Although a number of relevant approaches have been proposed in recent years, these algorithm incur the problem of producing a large number of candidate itemsets for high utility itemsets and probably degrades the mining performance in terms of execution time and memory space. In this paper, we propose two algorithms, viz., utility pattern growth (UP-Growth) and Improved UP-Growth i.e. Improved Utility Pattern Growth, for mining high utility itemsets with a set of effective strategies for pruning candidate itemsets. The information of high utility itemsets is maintained in a compact tree-based data structure utility pattern tree (UP-Tree), it scan the original database twice to manage data structured way. Proposed algorithms, especially Improved UP Growth, not only reduce the number of candidates effectively but also outperform other algorithms substantially in terms of runtime and memory consumption, especially when databases contain lots of long transactions**.**

KEYWORDS: High Utility Itemset; UP-Growth; Transaction Weighted Utility; Frequent Itemset Mining; Utility Mining; Downward Closure Property.

I. INTRODUCTION

Mining frequent itemsets from a transaction database refers to the discovery of the itemsets which frequently appear together in the transactions. The main objective of Utility Mining is to identify the itemsets with highest utilities above a user-specified threshold, by considering profit, quantity, cost or other user preferences. If the support of an itemset exceeds a user-specified minimum support threshold, the itemset is considered as frequent. Most frequent itemset mining algorithms employ the downward closure property of itemsets[1] ,[2],[13]. However, the unit profits and purchased quantities of items are not considered in the framework of frequent itemset mining. The basic meaning of utility is the interestedness/ importance/profitability of items to the users.

The utility of items in a transaction database consists of two aspects: (1) the importance of items of different transaction is called external utility, and (2) the importance of the items in the transaction, which is called internal utility. The utility of an itemset is defined as the external utility multiplied by the internal utility. An itemset is called a high utility itemset if its utility is greater than a user specified threshold; otherwise, the itemset is called a low utility itemset[8][17][20].

Mining high utility itemsets from databases refers to finding the itemsets with high profits and it is not an easy task since downward closure property [1],[2],[15]. In other words, pruning search space for high utility itemset mining is hard because a superset of a low utility itemset may be a high utility itemset. A simple method to address this problem is to enumerate all itemsets from databases by the principle of exhaustion. Obviously, this method couldn't tolerate the problems of a search space, especially when databases contain lots of long transactions or a low minimum utility threshold is set. Recently proposed compact tree structure, viz., UP-Tree, maintains the information of transactions and itemsets, facilitate the mining performance and avoid scanning original database repeatedly.

10.15680/ijircce.2015.0304054

II. RELATED WORK

A number of traditional ARM algorithms and optimizations have been proposed. One of the well-known algorithms is Apriori algorithm[10], which is the pioneer for efficiently mining association rules from large databases. It's widely recognized that FP-Growth[6] achieves a better performance than Apriori Algorithm since it finds frequent itemsets without generating any candidate itemset and it scans database just twice. There are also many studies that have developed different weighting functions[14] for weighted pattern mining.

Mengchi Liu [13] proposed an algorithm, called HUI-Miner (High Utility Itemset Miner), for high utility itemset mining.HUI-Miner uses a structure, called utility-list, to store the utility information of an itemset and the heuristic information for pruning the search space of HUI-Miner. By avoiding the costly generation and utility computation of numerous candidate itemsets, HUI-Miner can efficiently mine high utility itemsets from the utility lists constructed from a mined database.

Although two-phase algorithm reduces search space by using TWDC property, it still generates too many candidates to obtain HTWUIs and requires multiple database scans. To overcome this problem, Li et al. [11] proposed an isolated items discarding strategy (IIDS) to reduce the number of candidates. By pruning isolated items during level-wise search, the number of candidate itemsets for HTWUIs in phase I can be reduced. However, this algorithm still scans database for several times and uses a candidate generation-and-test scheme to find high utility itemsets. To efficiently generate HTWUIs in phase I and avoid scanning database too many times, Ahmed et al. [3] proposed a tree-based algorithm, named IHUP. A tree based structure called IHUP-Tree is used to maintain the information about itemsets and their utilities. Each node of an IHUP-Tree consists of an item name, a TWU value and a support count. IHUP algorithm has three steps: 1) construction of IHUP-Tree, 2) generation of HTWUIs, and 3) identification of high utility itemsets[21]. In step 1, items in transactions are rearranged in a fixed order such as lexicographic order, support descending order or TWU descending order. Then the rearranged transactions are inserted into an IHUP-Tree. In step 2, HTWUIs are generated from the IHUP-Tree by applying FP-Growth [14]. Thus, HTWUIs in phase I can be found without generating any candidate for HTWUIs. In step 3, high utility itemsets and their utilities are identified from the set of HTWUIs by scanning the original database once.

Although IHUP achieves a better performance than IIDS and Two-Phase, it still produces too many HTWUIs in phase I. Note that IHUP and Two-Phase produce the same number of HTWUIs in phase I since they both use TWU framework to overestimate itemsets utilities. However, this framework may produce too many HTWUIs in phase I since the overestimated utility calculated by TWU is too large. Moreover, the number of HTWUIs in phase I also affects the performance of phase II since the more HTWUIs the algorithm generates in phase I, the more execution time for identifying high utility itemsets it requires in phase II.

III. PROBLEM STATEMENT

In the literature we have studied the different methods proposed for high utility itemset mining from large datasets. But all this methods frequently generate a huge set of PHUIs and their mining performance is degraded consequently. Further in case of long transactions in dataset or low thresholds are set, then this condition may become worst. The huge number of PHUIs forms a challenging problem to the mining performance since the more PHUIs the algorithm generates, the higher processing time it consumes. Thus to overcome this challenges the efficient algorithms presented recently in [1], [3], [12]. These methods in [1] outperform the state-of-the-art algorithms almost in all cases on both real and synthetic data set. However this approach in [1] is still needs to be improved in case of less memory based systems.

IV. EXISTING SYSTEM

The framework of the existing methods consists of three steps: 1) Scan the database twice to construct a global UP Tree with the first two strategies 2) recursively generate PHUIs from global UP-Tree and local UP-Trees by UP-Growth with the third and fourth strategies or by UP-Growth+ with the last two strategies and 3) identify actual high utility item sets from the set of PHUIs[16]. To distinguish the patterns found by our methods from HTWUIs since our

10.15680/ijircce.2015.0304054

methods are not based on traditional TWU model. By our effective strategies, the set of PHUIs will become much smaller than the set of HTWUIs. After constructing a global UP-Tree, a basic method for generating PHUIs is to mine UP-Tree by FP-Growth. Thus, we propose an algorithm UP-Growth by pushing two more strategies into the framework of FP-Growth. By the strategies, overestimated utilities of item sets can be decreased and thus the number of PHUIs can be further reduced. UP-Growth achieves better performance than FP-Growth by using DLU and DLN to decrease overestimated utilities of item sets. However, the overestimated utilities can be closer to their actual utilities by eliminating the estimated utilities that are closer to actual utilities of unpromising items and descendant nodes. In UP-Growth, minimum item utility table is used to reduce the overestimated utilities. In UP-Growth+, minimal node utilities in each path are used to make the estimated pruning values closer to real utility values of the pruned items in database. The mining processes of UP-Growth and UP-Growth+ by maintaining only essential information in UP-Tree. By these strategies, overestimated utilities of candidates can be well reduced by discarding utilities of the items that cannot be high utility or are not involved in the search space.

V. PROPOSED METHOD

The goal of utility mining is to generate all the high utility itemsets whose utility values are beyond a user specified threshold in a transaction.

A. *Up Growth:*

The UP-Growth [11] is one of the efficient algorithms to generate high utility itemsets depending on construction of a global UP-Tree. In phase I, the framework of UP-Tree follows three steps: (i). Construction of UP-Tree [2]. (ii). Generate PHUIs from UP-Tree. (iii). Identify high utility itemsets using PHUI.

The construction of global UP-Tree [2] is follows,

(i). Discarding global unpromising items (i.e., DGU strategy) is to eliminate the low utility items and their utilities from the transaction utilities.

(ii). Discarding global node utilities (i.e., DGN strategy) during global UP-Tree construction. By DGN strategy, node utilities which are nearer to UP-Tree root node are effectively reduced. The PHUI is similar to TWU, which compute all itemsets utility with the help of estimated utility. Finally, identify high utility itemsets (not less than min_sup) from PHUIs values. The global UP-Tree contains many sub paths. Each path is considered from bottom node of header table. This path is named as conditional pattern base (CPB).

Disadvantages

- It requires multiple database scans.
- It Generate multiple candidate Itemset.
- Other Algorithm like Apriori treats all item with same importance or profit.
- It consumes more memory space and performs badly with long pattern data set.

These methods are further needs to be improved over their limitations presented below:

(1) Performance of this methods needs to be investigated in low memory based systems for mining high utility itemsets from large transactional datasets and hence needs to address further as well.

(2)These proposed methods cannot overcome the screenings as well as overhead of null transactions; hence, performance degrades drastically.

B. *Up-Growth+:*

Although DGU and DGN strategies are efficiently reduce the number of candidates in Phase 1(i.e., global UP-Tree). But they cannot be applied during the construction of the local UP-Tree (Phase-2). Instead use, DLU strategy (Discarding local unpromising items) to discarding utilities of low utility items from path utilities of the paths and DLN strategy (Discarding local node utilities) to discarding item utilities of descendant nodes during the local UP-Tree construction. Even though, still the algorithm facing some performance issues in phase-2. To overcome this, maximum transaction weight utilizations (MTWU) are computed from all the items and considering multiple of min_sup as a user specified threshold value as shown in algorithm. By this modification, performance will increase compare with existing UP-Tree construction also improves the performance of UP-growth algorithm. An improved utility pattern growth is abbreviated as IUPG.

Advantages

- It scan the database just twice.
- It is easy to implement.
- It reduces unnecessary calculation when database is updated, and when user specified minimum threshold is changed.
- It requires less memory space and less execution time.

C. *Up-Growth+ Algorithm:*

Input: Transaction database D, user specified threshold.
Output: high utility itemsets.

```
Begin

1. Load dataset contains number transactions Td $\in$ D
2. Determine transaction utility of Td in D and TWU of itemset (X)
3. Compute min_sup (MTWU * user specified threshold)
4. If (TWU(X) $\leq$ min_sup) then Remove Items from transaction database
5. Else insert into header table H and to keep the items in the descending order.
6. Repeat step 4 & 5 until end of the D.
7. Insert Td into global UP-Tree .
8. Apply DGU and DGN strategies on global UP- tree.
9. Re-construct the UP-Tree
10. For each item $a_i$ in H do
11. Generate a PHUI Y= X U $a_i$
12. Estimate utility of Y is set as $a_i$'s utility value in H
13. Put local promising items in Y-CPB into H
14. Apply strategy DLU to reduce path utilities of the paths
15. Apply strategy DLN and insert paths into Td
16. If Td $\neq$ null then call for loop
End for
End
```

D. *Application:*

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. For example, in medical application, the rare combination of symptoms can provide useful insights for doctors [7][18]. A retail business may be interested in identifying its most valuable customers i.e. who contribute a major fraction of overall company profit[4][19].

VI. IMPLEMENTATION DETAILS

A. System Architecture and Design

This is basic system architecture to represent the basic functionality of the system. To construct the UP-Tree to apply the two algorithms UP-Growth and UP-Growth+ to find the potential high utility item sets. Main intension of this system is reducing item sets over calculated utilities.

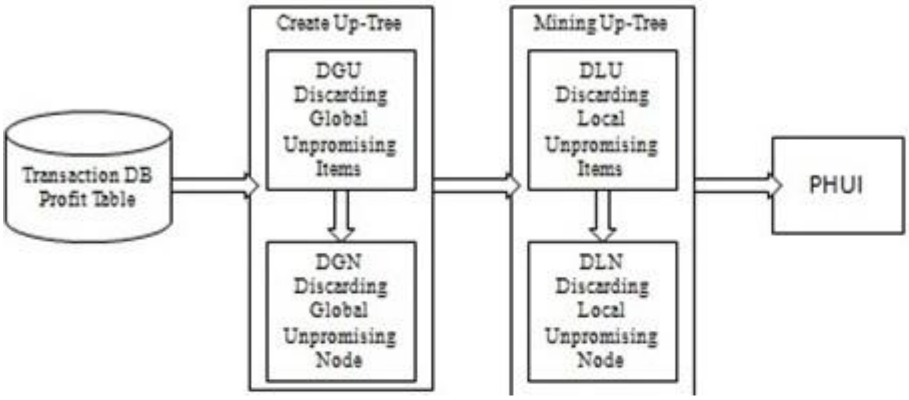

Create Up-Tree
DGU
Discarding Global Unpromising Items
DGN
Discarding Global Unpromising Node
Mining Up-Tree
DLU
Discarding Local Unpromising Items
DLN
Discarding Local Unpromising Node
Transaction DB
Profit Table
PHUI

Fig. 1 System Architecture

Fig. 1 contains the following blocks:

- Transaction DB and Profit table are input to the system to discover potential highly utilized Item sets.
- Create UP-tree: UP-tree is created using discarding unfavourable global items and reducing global node utility. UP-tree has fields as Node.name which contain name of the item, Node. Count, Node.nu, Node. parent, Node. hlink.
- Discarding global unpromising items: After calculating transaction utility and transaction weighted utility, the item sets having less utility than predefined minimum threshold utility are disposed.
- Discarding global node utility: After disposing the unfavourable items the global node utilities are reduced. And nodes are inserted into UP tree using create UP-tree algorithm.
- Mining Up-tree: In which local unpromising Item and node utility.
- Discarding local unpromising items: Construct conditional pattern base of bottom item entry in header table Retrieve the entire path related to that item CPB. Conditional UP tree created by two scans over CPB. Local unfavourable items removed using path utility of each item in CPB paths are organized in descending order.
- Discarding local node utility: Reorganized path is inserted into conditional utility pattern tree using reduce local node utility strategy.
- Potential High Utility Item sets: Identify potential high utility item sets and their utilities form UP tree mining using Dispose of local unfavourable items and Reduce local node utility.

B. Mathematical Model

Let S be the system that describes dataset i.e. set of transaction with profit of item as input to system with calculation of transaction utility, transaction weighted utility, recognized transaction utility, up tree construction, UP growth algorithm, Improved UP growth algorithm and this all gives 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(UP-Grrowth+) = Advanced Utility Pattern Growth
PHUI= Potentially High Utility Item set

Inputs:
Tp= {D,P}

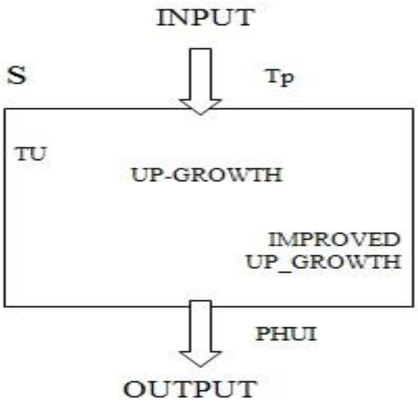


Fig. 2 Mathematical Model

Process:

1) TU=Σ i_p € T_d $[pr(i_p)*q_p(i_p,T_d)]$
2) $TWU(i_p) = \Sigma$ TU € i_p
3) RTU (T_d):= TU (T_d) - UUI
4) create UP tree and Mine it.with Node.name, Node. count, Node.nu, Node. parent, Node.hlink.

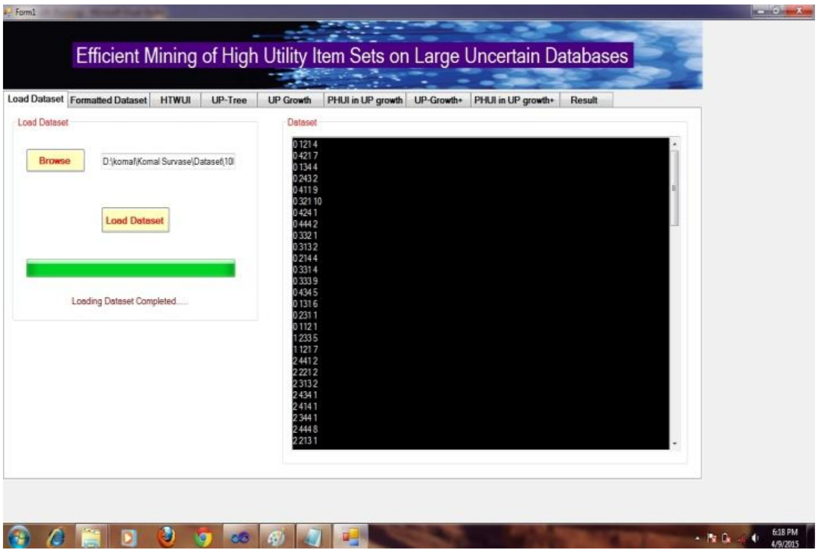
Output:

All Potential High Utility Itemsets in T_x

VII. EXPERIMENTAL RESULTS

The experiments were done on a 2.20 GHz Intel(R) Pentium(R) Processor with 3 GB RAM, and performing on Windows 7(32 bit). The proposed algorithms are implemented in .Net language.

In our project Up-Growth and Up-Growth+ outperform the state-of-the-art algorithms in all cases on both real and synthetic data sets.

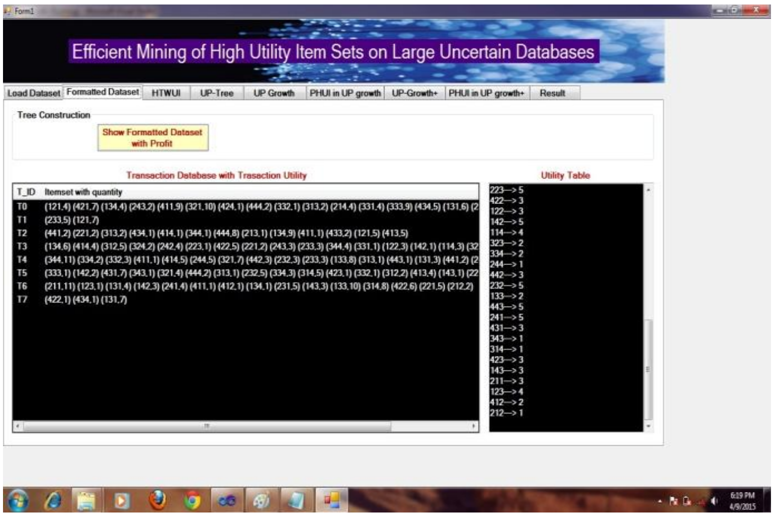


D:\komal\Komal Survase\Dataset\10

Loading Dataset Completed.....

```
0 121 4
0 421 7
0 134 4
0 243 2
0 411 9
0 321 10
0 424 1
0 444 2
0 332 1
0 313 2
0 214 4
0 331 4
0 333 9
0 434 5
0 131 6
0 231 1
0 112 1
1 233 5
1 121 7
2 441 2
2 221 2
2 313 2
2 434 1
2 414 1
2 344 1
2 444 8
2 213 1
```

Fig. 3 Load Dataset



Transaction Database with Trasaction Utility

T_ID	Itemset with quantity
T0	(121,4) (421,7) (134,4) (243,2) (411,9) (321,10) (424,1) (444,2) (332,1) (313,2) (214,4) (331,4) (333,9) (434,5) (131,6) (2
T1	(233,5) (121,7)
T2	(441,2) (221,2) (313,2) (434,1) (414,1) (344,1) (444,8) (213,1) (134,9) (411,1) (433,2) (121,5) (413,5)
T3	(134,6) (414,4) (312,5) (324,2) (242,4) (223,1) (422,5) (221,2) (243,3) (233,3) (344,4) (331,1) (122,3) (142,1) (114,3) (32
T4	(344,11) (334,2) (332,3) (411,1) (414,5) (244,5) (321,7) (442,3) (232,3) (233,3) (133,8) (313,1) (443,1) (131,3) (441,2) (2
T5	(333,1) (142,2) (431,7) (343,1) (321,4) (444,2) (313,1) (232,5) (334,3) (314,5) (423,1) (332,1) (312,2) (413,4) (143,1) (22
T6	(211,11) (123,1) (131,4) (142,3) (241,4) (411,1) (412,1) (134,1) (231,5) (143,3) (133,10) (314,8) (422,6) (221,5) (212,2)
T7	(422,1) (434,1) (131,7)

Utility Table

```
223—> 5
422—> 3
122—> 3
142—> 5
114—> 4
323—> 2
334—> 2
244—> 1
442—> 3
232—> 5
133—> 2
443—> 5
241—> 5
431—> 3
343—> 1
314—> 1
423—> 3
143—> 3
211—> 3
123—> 4
412—> 2
212—> 1
```

Fig.4 Formatted Dataset with Utility

In fig. 3 dataset are loaded as system works with respect to different transaction size and different minimum utility as shown below fig. 3.

After calculation of Transaction Utility and Transaction Weighted Utility new Transaction Utility is generated by pruning unpromising items called as Recognized transaction utility (RTU). Also we maintain a minimum item utility to keep minimum item utilities for all global promising items in the database.Up-tree is constructed by applying DGU, DGN, DLU and DLN strategy potential high utility itemset identified. And Final output is generated. Time comparison between UP-Growth and Up-Growth+ is shown in fig. 5.

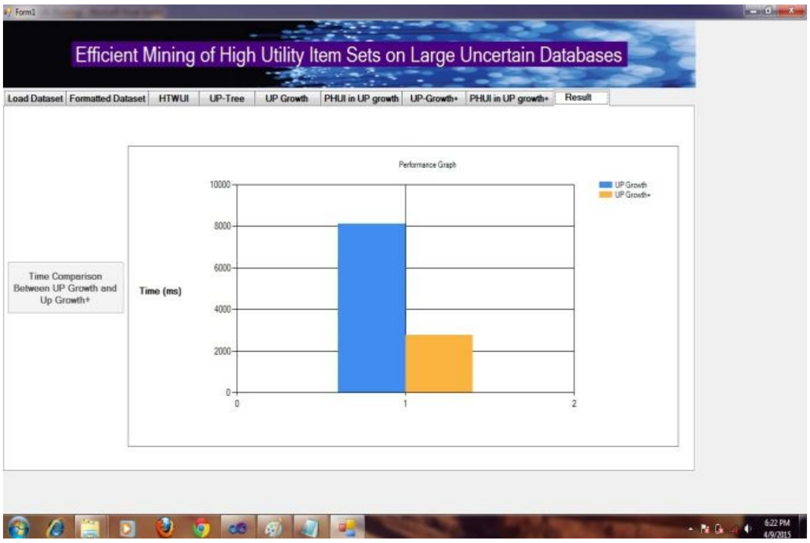


Fig.5 Generated Result

Performance evaluation of Up-Growth and Up-Growth+ for phase I and Phase II execution times are shown by graph below in fig. 6 and fig. 7.

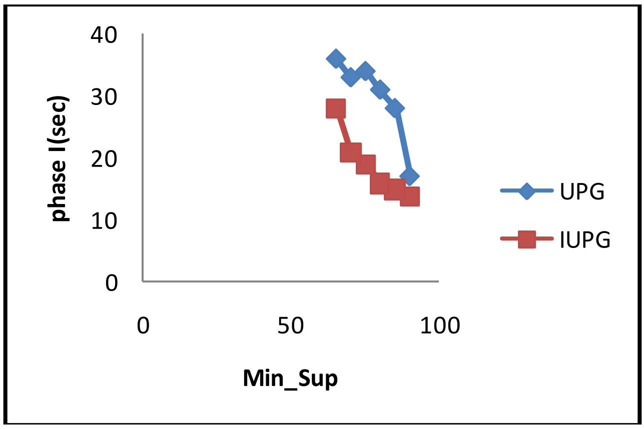


Fig. 6 Execution time for phase I on Chess

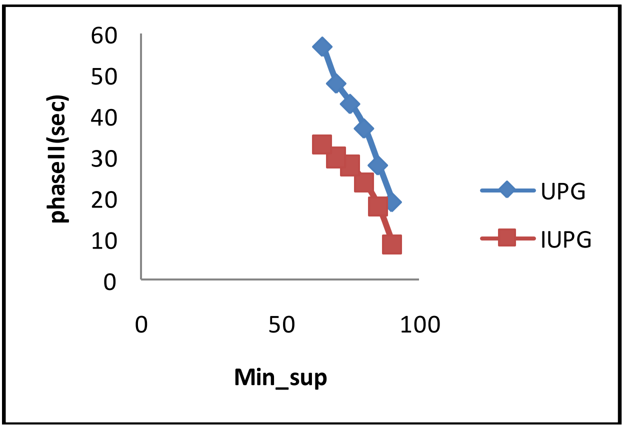


Fig. 7 Execution time for phase II on Chess

VIII. CONCLUSION AND FUTURE SCOPE

Proposed system UP-Growth and UP-Growth+ Mining for discovering High utility item sets from databases. Data Structure UP-Tree for recording the information of highly utilized item sets and four effective strategies, DGU, DGN, DLU and DLN, to minimize search space and the number of candidates for utility mining. Potential high utility item sets can be generated from Utility Pattern Tree with only two scans of the database. UP-Growth specially Up-Growth+ Algorithm is faster than previous algorithms when database have lots of long transactions.

The current study proposed two definitions to capture the effects of the noise in the data. This pointed out possible scenarios where the mining of these patterns is central as well as the challenges in developing efficient mining algorithms. Future works include the extension of the temporal utility pattern tree to mine noisy patterns[5][16], and developing more efficient techniques to handle genomic data.

REFERENCES

1. Vincent S Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu, Fellow, “Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases”, IEEE Transactions On Knowledge And Data Engineering, volume 25, Issue No. 8, pp 1772-1786, AUGUST 2013.
2. Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu, “UP-Growth: An Efficient Algorithm for High Utility Itemset Mining”, Proc. 15th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'09), Paris, France, volume 8, Issue No. 10, pp 1222-1228, 2010.
3. Liang Wang, David Wai-Lok Cheung, Reynold Cheng, Member, IEEE, Sau Dan Lee, and Xuan S. Yang, “Efficient Mining of Frequent Item Sets on Large Uncertain Databases”, IEEE Transactions On Knowledge And Data Engineering, volume 24, Issue No. 12, pp 2170-2183, DECEMBER 2012.
4. Thomas Bernecker, Hans-Peter Kriegel, Matthias Renz, Florian Verhein, Andreas Zuefle, “Probabilistic Frequent Itemset Mining in Uncertain Databases”, Proc. 15th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'09), volume 24, Issue No. 12, pp 1270-1283 , 2009.
5. Jency Varghese, K.Soundararajan, “Frequent Itemsets Mining on Large Uncertain Databases: Using Rule Mining Algorithm”, International Journal of Computer Applications, Volume 65– Issue No. 20,pp 0975-8887, March 2013.
6. Supriya P. Bhosale, “Mining High BeneficialItemsets from Transactional Database”, International Journal of Computer Science and Information Technologies, ISSN: 0975-9646, Vol. 5, Issue No. 6 , 2014.
7. Maya Joshi, Mansi Patel, “A Survey on High Utility Itemset Mining Using Transaction Databases”, (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN: 0975-9646, Vol. 5 Issue No. 6 , pp 2407-2412, 2014.
8. A. Erwin, R.P. Gopalan, and N.R. Achuthan, “Efficient Mining of High Utility Itemsets from Large Data Sets,” Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD), Volume 5, Issue No. 7 , pp. 554-561, 2008.
9. Rakesh Agrawal, Ramakrishnan Srikant, “Fast algorithms for mining association rules”, In Proc. of the 20th Int'l Conf. on Very Large Data Bases,Volume No. 4, Issue No. 5, pp. 487-499, 1994.
10. Y.-C. Li, J.-S. Yeh, and C.-C. Chang, “Isolated Items Discarding Strategy for Discovering High Utility Itemsets,” Data and Knowledge Engg., volume 64, no. 1, pp. 198-217, Jan. 2008.
11. Adinarayanareddy B, O Srinivasa Rao, MHM Krishna Prasad, “An Improved UP-Growth High Utility Itemset Mining”, International Journal of Computer Applications Volume 58, Issue No.2, pp 2275 – 2287, November 2012.
12. Mengchi Liu, Junfeng Qu, “Mining High Utility Itemsets without Candidate Generation”, Data and Knowledge Engg, Volume 58, Issue No.2, pp 1106 – 1111, Nov 2012.

13. Junfu Yin, Zhigang Zheng, Longbing Cao, "USpan: An Efficient Algorithm for Mining High Utility Sequential Patterns", Knowledge-Based Systems, vol. 24, Issue No. 9, pp 660-668, 2008.
14. More Rani N., Anbhule Reshma V., Waghmare Archana B., "Mining High Utility Item sets From Transaction Database", International Journal of Latest Trends in Engineering and Technology (IJLTET), ISSN: 2278-621X, Vol. 3 Issue No. 3, pp 180-182, January 2014.
15. S. Gomathi, M. Suganya, "An Enhanced Upgrowth Algorithm For Temporal High Utility Item Mining", International Journal Of Engineering Sciences & Research Technology, Volume 6, Issue No. 8, pp 2277-2282, January 2015.
16. A. A. Bhosale, S. V. Patil, P. M. Tare, P. S. Kadam, "Review Paper - High Utility Itemsets Mining on Incremental Transactions using UP-Growth and UP-Growth+ Algorithm", International Journal on Recent and Innovation Trends in Computing and Communication, Volume 2, Issue No. 11, pp 3366 - 3368.
17. Jyothi Pillai, O.P.Vyas, "Overview of Itemset Utility Mining and its Applications", International Journal of Computer Applications Volume 5, Issue No.11, pp 9-13, August 2010.
18. Supriya P. Bhosale, "Novel Approach to Discover High Utility Itemsets from Transactional Database", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3, Issue No.1, pp 139-142, January 2014.
19. Arshia Sultana, E.Krishnaveni Reddy, "Up Approach: Mining High Utility Itemsets", International Journal of Computer & Organization Trends, Volume 10, Issue No 1, pp 1-10, July 2014.
20. Switi Chandrakant Chaudhari, Vijay Kumar Verma, "Mining High Utility Item Set From Large Database: - A Recent Survey", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue No. 5, pp 685-688, May 2013.
21. Rajmohan C, Priya G, Niveditha C, Pragathi R, Generation of Potential High Utility Itemsets from Transactional Databases, International Journal of Innovative Research in Advanced Enginneing, Volume 1, Issue No. 1, pp 61-67, Mar 2014.

BIOGRAPHY

Ms. Komal S Surawase is Completed his BE in RSCOE, Chiplun. Later she is studying ME (Computer Engineering) in JSCOE, Hadapsar, Pune, having 3 years teaching experience. Her interested areas are data mining MANET and My SQL Database with .net technologies.

Mr. Madhav D Ingle working as an Associate Professor JSCOE, Hadapsar, Pune, He has completed ME (Computer Engineering), having 17 years teaching experience. His Area of interest are in Computer Networks, Data Mining,Data Structure.

10.15680/ijircce.2015.0304054