ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 11, November 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Efficient Algorithms for Mining HUI and Closed Itemsets

Mahendra M. Kapadnis[1]
P.G. Student, Department of Computer Engineering,
Late G. N. Sapkal COE, Nashik
Savitribai Phule Pune University,
Maharashtra – India

Prof. Nilesh R. Wankhade[2]
Associate Professor, Department of Computer Engineering,
Late G. N. Sapkal COE, Nashik
Savitribai Phule Pune University,
Maharashtra – India

Abstract: ***A task of mining a high utility itemsets is very difficult where itemsets having large utility are to be found. There are many algorithms available to do the same but having huge quantity of HUI is decrease the performance of mining procedure. To obtain a high efficiency a novel framework is introduce in this paper for mining high utility itemsets (CHUIs), propose three efficient algorithms named AprioriCH (Apriori-based algorithm for mining High utility Closed itemsets), AprioriHC-D(AprioriHC algorithm with Discarding unpromising and isolated items) and CHUD (Closed þ High Utility Itemset Discovery). Before this a traverse path technique is used for frequent itemsets mining. All High Utility Itemsets) is proposed to find all HUIs with minimum representative pattern from the set of CHUIs from the dataset without accessing original. The method proposed a massive change in number of HUI which increase the efficiency. Before the HUIs are discovered the traversal path technique is used to improve result in calculation with time and number of HUI requirements.***

Keywords: Frequent itemset, closed high utility itemset, utility mining, data mining, traverse path.

I. INTRODUCTION

A method Frequent Itemset Mining [1] has an application called as Market Basket Analysis is used to discover a frequent itemsets, which calculate the itemset (set of items) which a frequently purchase by customers. It is an old method to calculate frequent itemsets. As far it is observed that this method found the itemsets in a large amount and also the items having large selling frequency are only introduce which may be having low revenue. But the items which are of large revenue but having low frequency are discarded from the records automatically in this system. This problem persist because it consider every item in the binary form that it is present or absent. Only the existence of item is considered in this that is purchase quantity is not considered. It considers every item having same value/ weight. Hence the system is incapable to satisfy user needs.

To solve this issue a method utility mining is introduced which consider each item with its weight that is profit value and can be occur many times in transaction i. e. transaction quantity. An item which is having utility more than user specified minimum utility threshold is called as high utilityitemset. The efficiency of method decrease with low minimum utility threshold that large the threshold efficiency become more and small threshold generates more HUI then efficiency becomes less. Here propose three efficient algorithms named AprioriHC, AprioriHC-D and CHUD (Closed High Utility itemset Discovery) [2]. As in data mining the pattern mining has more importance and for business mining it is very important the A traverse path technique is use to find a frequent itemsets using via-link technique. High utility itemsets are given input to closed high utility itemsets algorithm which calculate closed itemsets and CHUD algorithm calculates closed high utility itemsets using a support count of an every item and itemsets. Which compares a support of every itemsets and duplication of same support is avoided and that itemsets are discarded. A transaction weighted utility is calculated which is weight of every itemset in each transaction. Apriori HC-D algorithm calculates frequent closed high utility itemsets. Finally DAHU algorithm is used to combine result of both algorithms and derives all high utility itemsets. A top-down method named DAHU (Derive All High Utility itemsets) for recover all HIUs from closed HUIs is used.

II. LITERATURE SURVEY

In Dec. 2009, B.-S. Jeong, C. F. Ahmed, S.K. Tanbeer, and Y.-K. Lee, defines a three structures where first is to prepare items lexicographic form called as Incremental HUP Lexicographic Tree (IHUPLTree) [4]. Without any arrangement it obtains data in incremental format. Second one is to obtain items order in descending order as the transaction frequency of that item named as IHUP transaction frequency tree (IHUPTF-Tree). The third one is based on TWU of the item to reduction the period of mining called as IHUP-transaction-weighted utilization tree (IHUPTWU-Tree). These three tree structures are very effective and accessible for incremental and shared HUP mining.

In 2008, K. Chuang, J. Huang, and M. Chen proposed a Mining top-k frequent patterns in the presence of the memory constraint [6].This paper discover a workably extraordinary mining job to retrieve top-k (closed) itemsets in the existence of the memory constraint. Exactly so, as conflicting to most surviving works that hardly focuses on refining the mining effectiveness or on reducing the memory size by best strength, It firstly try to mention the presented top memory size that can be used by mining frequent itemsets. The MTK and MTK close are found to closed itemsets and frequent itemsets to follow with the top bound of the memory intake, correspondingly, without mentioning the subtle bottom support. User only require a human clear parameter, mostly the desired quantity of closed frequent itemsets k. practically it is very hard to constrain the memory consumption while also effectively getting top-k itemsets. To effectively obtain this ,MTK and MTK Close are invented as level wise finding algorithms, where the number of candidates are to be generated-and tested in every database scan will be limited. For testing candidate itemsets with multiple itemsets length a stair search approach is used which initiate to database scan which is small and essential.

In 2003, R. Chan, Q. Yang, and Y. Shen, proposed mining high utility itemsets where mining high utility itemsets from a transactional database [7] shows the problem of huge number of candidate generation in other algorithms introduced in the year which are too many. Finding of itemsets with utility like weight or profit has a huge number of candidate generation during high utility mining which reduces the mining efficiency compare with time and space requirements. It may present database in tons of quantity of long transactions or large high utility itemsets then the problem is critical. The utility pattern growth (UP-Growth) and UP-Growth+ are introduced in this paper having a strategy called as pruning which shows set of effective rules for candidates pruning itemsets. The data of high utility itemsets is composed in a tree-based data structure named utility pattern tree (UP-Tree) like that candidate itemsets may be generated proficiently by only two scans of database. The performance of UP-Growth and UP-Growth+ is equated with the state-of-the-art algorithms on multiple types of both real and synthetic data sets.

In 1994, R. Agrawal and R. Srikant, introduced Fast algorithms for mining association rules [1] in which they consider the problem of defining association rules among items ina huge database of sales transactions. Paper provide two new algorithms for solving the problem that are basically different from the known algorithms. Experimental results shows that the algorithms perform the well-introduced algorithms by factors ranging from three for minor problems to more than an order of magnitude for huge problems. It also display how the best features of the two proposed algorithms can be joined into a hybrid algorithm, called as Apriori Hybrid. Scale-up research defines that Hybrid scales Apriori linearly using the number of transactions. An Apriori Hybrid also has outstanding scale-up assets in view of transaction size and the number of items in the database.

In 2014, Guimei Liu, Haojun Zhang, and Limsoon Wong, introduce [12] Understanding the searching structure of website visitors is a significant factor of results in the emerging business models of electronic commerce and even mobile commerce. However, Web traversal patterns used by early Web usage mining approaches are not effective for the content management of websites. They do not give the clear picture of the views of the visitors. The Web navigation patterns, termed throughout-surfing patterns (TSPs) as defined in this paper, are a superset of Web traversal patterns which finely display the trends toward the next visited Web pages in a browsing session. TSPs are more expressive for understanding the purposes of website visitors. It uses a post processing strategy.

III. System Implementation

A Frequent Itemset Mining has an application called as Market Basket Analysis is used to discover a frequent itemsets, which calculate the itemset (set of items) which a frequently purchase by customers. It is an old method to calculate frequent itemsets. As far it is observed that this method found the itemsets in a large amount and also the items having large selling frequency are only introduce which may be having low revenue. But the items which are of large revenue but having low frequency are discarded from the records automatically in this system. This problem persists because it considers every item in the binary form that it is present or absent. Only the existence of item is considered in this that is purchase quantity is both considered. It considers every item having same value/ weight. Hence the system is incapable to satisfy user needs.

HUI mining is very difficult task as because a downward closure property used in FIM is not in utility mining. The large number of high utility itemsets discovers also a problem as because it is time consuming and memory consuming and it is very difficult to user to understand the results also. It is broadly find that the e high utility itemsets the algorithms create, the more processing they consume. The performance of the mining job reduces significantly for less minimum utility thresholds or when dealing with condensed databases. To solve the issue pattern mining is introduced which only focuses on the pattern of high utility itemsets. There are many high utility itemsets having different minimum utility threshold or same threshold value. When the itemsets are under the same cluster then it will having same pattern and one can be representative for all under that cluster. So, it can be small representative for large patterns.Traverse path graph generation is done as it is representation of graph which holds one item as a node all frequent items. To find HUIs a traverse path method is introduced where it discovered the graph in which an item is found to be frequent keeping it as centre item and all itemsets combination with it.

A via-link is used to define such representation where an item node is via and frequent item is link.

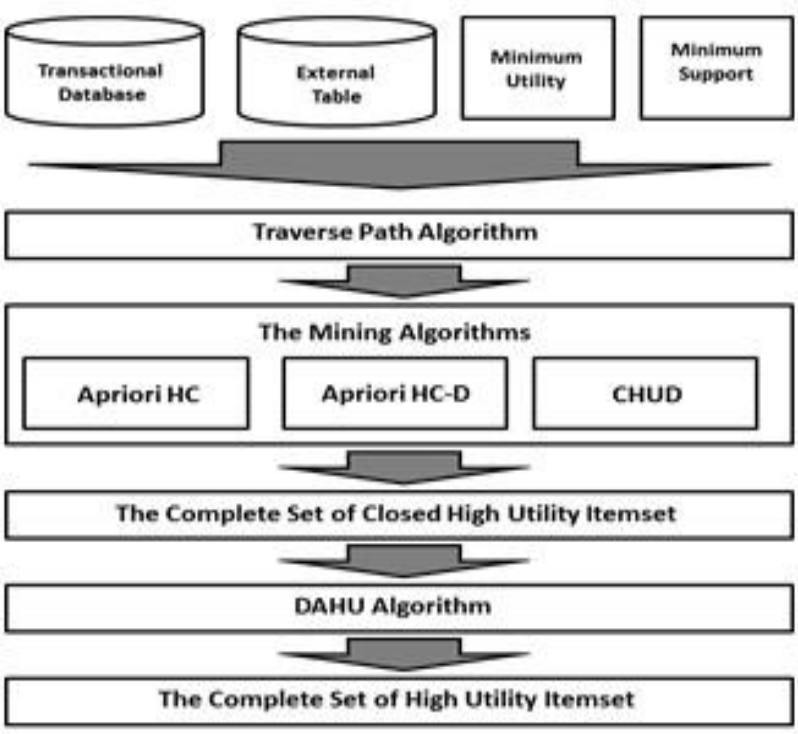


Fig. 1. Proposed System Architecture

It can abstractly view a transaction as a set of items called as itemset, and the item can be naturally represented by a directed graph with vertices and edges corresponding to the transaction. In the application of mining throughout-surfing patterns, an item is consider as a node and will go at any node, it need to know first which item has come. Accordingly, the concept of via-links is used in this paper to record the ‘‘from-to-via’’ information in the proposed graph, which is unique to the mining of throughout-surfing patterns. Therefore, a novel data structure called path traversal graph to mine a high utility candidate itemset

generation is proposed. The compact structure of the path traversal graph can help improve the efficiency of mining throughout-surfing patterns.

Further a High Utility Itemset mining is used to calculate candidate generation as a using support where a support is calculated as consider dataset D. A count of an item K containing transactions in database D is support count of K. Support of K in database D is ratio of support count of K with database D. An absolute utility of item is defined as p (ai,D)*q(ai, TR).

An Itemset is high utility itemset when it is having absolute utility greater than user specified minimum utility. An AprioriHC algorithm is used where it calculate a closed high utility itemsets and AprioriHCD algorithm is discarding unpromising and isolated items and generate closed high utility itemsets. A CHUD is closed high utility itemset discovery is for closed high utility itemsets mining algorithm. AprioriHC is horizontal database mining and CHUD is vertical database mining. CHUD mine in depth first search.

A method called DAHU (Derive All High Utility Itemsets) is proposed to recover all HUIs from the set of CHUIs which not access original database for further use. Firstly a transaction of itemsets is given as input to the algorithm of traverse path and the output of traverse path algorithm is given an input to AprioriHC algorithm to calculate closed high utility itemsets using internal utility and external utility where absolute utility is product of both internal utility and external utility. External utility is it may consider as profit of that item. AprioriHCD algorithm discard unpromising itemsets and isolated itemsets. Differently a CHUD algorithm calculates closed itemsets using transaction weighted utility which weight of transaction. A Dahu algorithm recover all closed high utility itemsets from above algorithms and final high utility itemsets are calculated.

IV. MATHEMATICAL MODEL

A. Problem Description

Let the system be described as {**S }**.

Such That {**S**} = { D, I, TSP, HC, HCD, R }

Where,

S = is a system.

D = Set of dataset.

I = Input.

TSP = Traverse Source Pattern.

HC = Apriori HC algorithm.

HCD = Apriori HC D algorithm.

R = Recovery of HUI.

B. HUI Algorithm

Algorithm:

Algorithm for High utility itemset

Input:

1) Transaction database.

2) Item profit value.

Output:

A set of High utility itemsets with Minimal number.

Begin:

1) Calculate minimum support of itemsets.

2) if support is less then min_sup discard itemsets.

3) Calculate utility of itemsets.

4) call Apriori HC-D algorithm.

5) Calculate TWU.

6) Call CHUD algorithm.

7) Call DAHU algorithm.

8) Output

End

C. Traverse Path Algorithm

Algorithm:

Graph construction

Input :

A collection of transactional database D and minimum support u

Output :

The frequent itemsets G

Begin

1. Number of transactions

2. Select one item v1 in one transaction

3. Select second item in same transaction v2.

4. calculate support of item and add itemsets in graph.

5. if support is less then min_sup discard itemsets.

6. output.

End.

V. RESULT AND ANALYSIS

As the first step is to calculate frequent itemset mining with high closed itemset given a dataset as input. After giving a dataset as input it calculate a frequent itemsets using traverse path algorithm as considering via-link and then high closed Itemset are found using apriori algorithm. Apriori HC-D algorithm is used to calculate HC itemsets. CHUD then calculate closed high utility itemset discovery. DAHU derives all high utility itemsets and complete set of HUI are output. At foodmart dataset of 3000 transactions at 200 as minimum utility results are.

Sr No.	Existing System (ms)	Proposed System (ms)	Difference (ms)

1.	198222	151491	46731
2.	177536	158052	19484
3.	217823	155984	61839

Table 1. Time difference for 1000 transactions

The difference between proposed system and existing system for 1000 Transactions time is shown above as considering minimum utility 200 which is user defined.

Sr No.	Existing System	Proposed System	Difference
1.	447	24	423

Table 2. Number of HUI Comparison

Number of high utility itemsets difference in existing system and proposed system

Sr No.	Existing System (ms)	Proposed System (ms)	Difference (ms)
1.	2024672	455739	1568933
2.	2112725	614382	1498343
3.	2098945	732988	1365957

Table 3. Time difference for 3000 transactions

The difference between proposed system and existing system for 3000 Transactions in time is shown above which is at 200 minimum utility where it is user defined.

Sr No.	Existing System	Proposed System	Difference
1.	471	63	408

Table 4. Number of HUI Comparison

The difference of candidate found i.e. number of HUI between proposed system and existing system for 3000 Transactions.

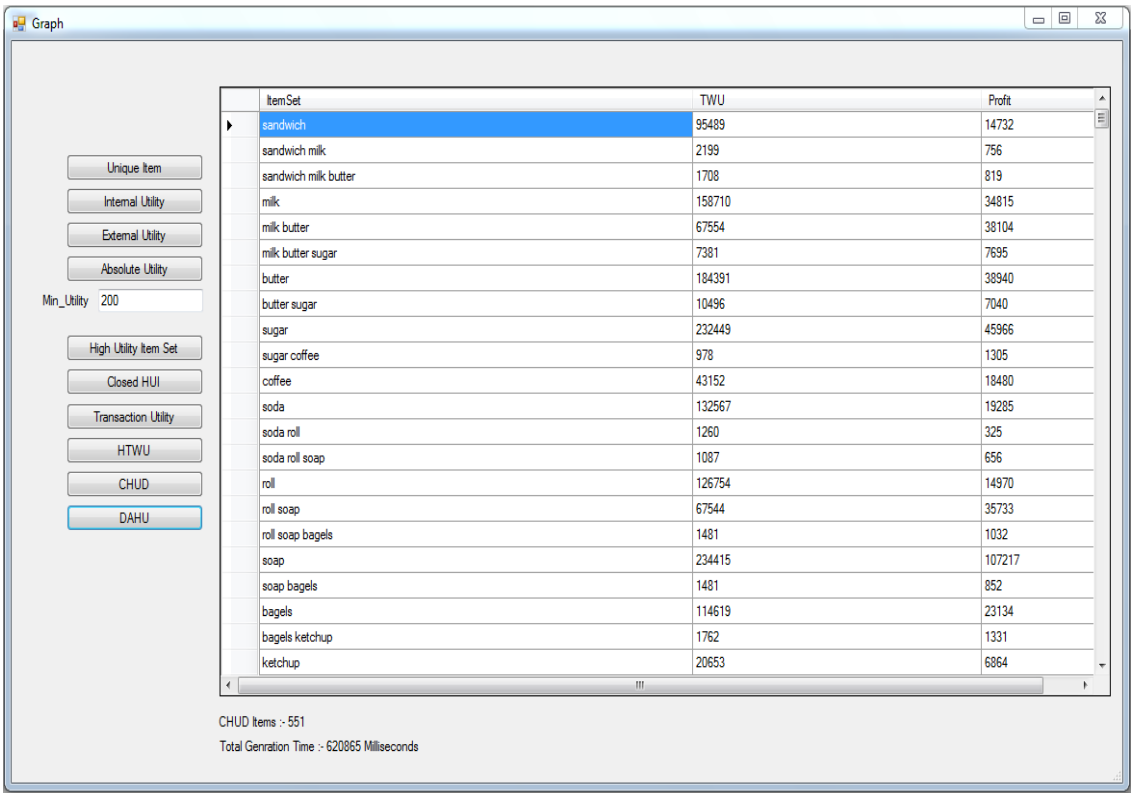
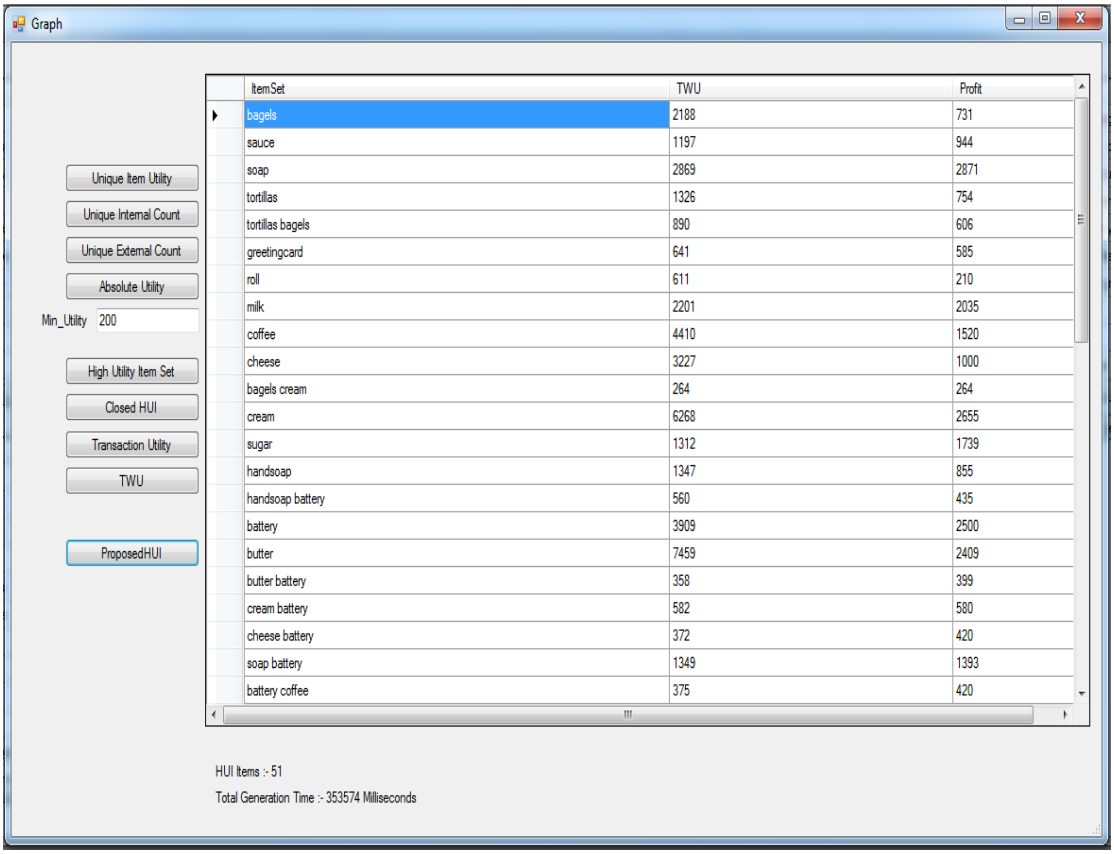


Fig. 2. Existing System Output



ItemSet	TWU	Profit
bagels	2188	731
sauce	1197	944
soap	2869	2871
tortillas	1326	754
tortillas bagels	890	606
greetingcard	641	585
roll	611	210
milk	2201	2035
coffee	4410	1520
cheese	3227	1000
bagels cream	264	264
cream	6268	2655
sugar	1312	1739
handsoap	1347	855
handsoap battery	560	435
battery	3909	2500
butter	7459	2409
butter battery	358	399
cream battery	582	580
cheese battery	372	420
soap battery	1349	1393
battery coffee	375	420

Fig. 3. Proposed System Output

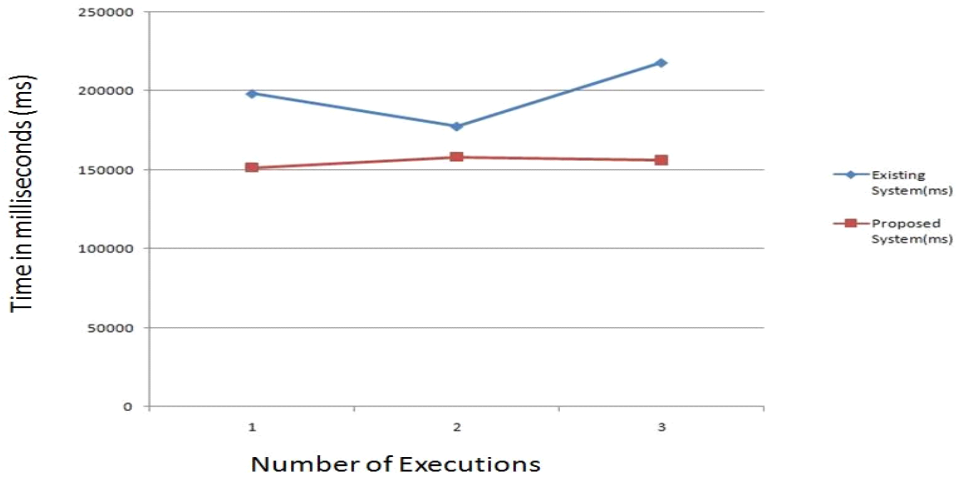


Fig. 4. HUI Comparison for 3 iteration

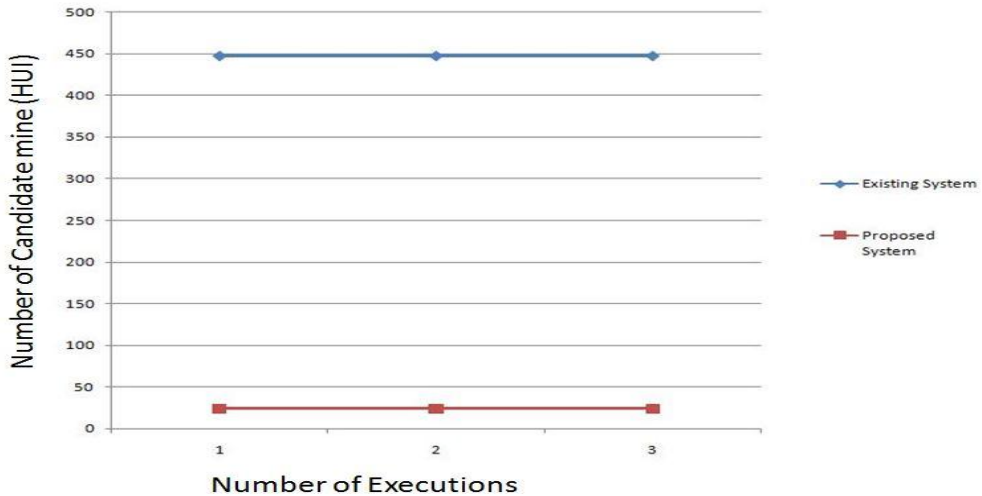


Fig. 5. Execution time comparison for 3 iteration

VI. CONCLUSION

By proposing a lossless and compact presentation termed closed high utility itemsets is solved the difficulty of redundancy in high utility itemset mining. To do the mining of this representation, three capable algorithms called AprioriHC (Apriori-based approach for mining High utility Closed itemset), AprioriHC-D (AprioriHC algorithm with Discarding unpromising and isolated items) and CHUID (Closed High Utility itemset Discovery). A traversal path algorithm is initially obtains via-links which are frequent itemset graph and then above mention three algorithms are process. DAHU derive all HUI efficiently.

ACKNOWLEDGEMENT

First and foremost, I would like to thank my guide Prof. N. R. Wankhade for his guidance and support. I would also like to thank to my friends for listening my ideas, asking questions and providing feedback and suggestions for improving ideas. I wish to express my sincere thanks to the Head of department, Prof. N. R. Wankhade also grateful thanks to the departmental staff members for their support.

References

1. R. Agrawal and R. Srikant, Fast algorithms for mining association rules, in Proc. 0th Int. Conf. Very Large Data Bases, 1994,pp. 487499.
2. Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger ,and Philip S. Yu, Fellow, Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets, in IEEE transactions on knowledge and data engineering ,vol. 27, no. 3, March 2015.
3. C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, Efficient tree structures for high utility pattern mining in incremental databases, in IEEE Trans. Knowl. Data Eng.,,vol. 21, no. 12, pp. 1708 1721, Dec. 2009.
4. J.-F. Boulicaut, A. Bykowski, and C. Rigotti, Free-sets: A condensed representation Of Boolean data for the approximation of frequency queries, in Data Mining Knowl. Discovery, vol. 7, no. 1, pp. 522, 2003.
5. T. Calders and B. Goethals, Mining all non-derivable frequent itemsets, in . Int. Conf. Eur. Conf. Principles DataMining Knowl. Discovery, 2002, pp. 7485.
6. K. Chuang, J. Huang, and M. Chen, Mining top-k frequent patterns in the presence of the memory constraint, in VLDBJ., vol. 17, pp. 13211344, 2008.
7. R. Chan, Q. Yang, and Y. Shen, Mining high utility itemsets, in in Proc. IEEE Int. Conf. Data Min., 2003, pp. 1926.
8. A. Erwin, R. P. Gopalan, and N. R. Achuthan, Efficient mining of high utility itemsets from large datasets, in Proc. Int. Conf. Pacific- Asia Conf. Knowl. Discovery Data Mining. 2008, pp. 554561.
9. K. Gouda and M. J. Zaki, Efficiently mining maximal frequent itemsets, in Proc. IEEE Int. Conf. Data Mining, 2001,pp. 163170.
10. T. Hamrouni, Key roles of closed sets and minimal generators in concise representations of frequent patterns, in Intell. Data Anal., vol. 16, no. 4, pp. 581631, 2012.
11. J. Han, J. Pei, and Y. Yin, Mining frequent patterns without candidate generation, in in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2000, pp. 112.
12. Yao-Te Wang, Anthony J.T. Lee, "Mining Web navigation patterns with a path traversal graph," Expert Systems with Applications 38 (2011) 7112–7122.

