# A Review on High Utility Mining to Improve Discovery of Utility Item set

**Vishakha R. Jaware[1], Madhuri I. Patil[2], Diksha D. Neve[3]**
**Ghrushmarani L. Gayakwad[4], Venus S. Dixit[5], Prof. R. P. Chaudhari[6]**

*[1,2,3,4,5]Student, CSE dept SSGB COET,BSL, Maharashtra, India.*
*[6] Professor, Dept. Of CSE Engineering, SSGB COET, BSL, Maharashtra, India.*

**Abstract-***Planning and designing of business strategies for today businesses became really a crucial process, because of the sale of more than hundreds of items in a single transaction. This can be realized by analyzing the sale databases of super shops, Big bazaars, mauls etc. Mining of frequent Item-sets is necessary to improve the business by maximizing the profit and minimizing the loss. Classic Apriori algorithm only considers the presence of an item in transactions. . In such cases, frequency of low profit items makes them to appear in frequent Item-sets, while items with higher profit will be listed under infrequent item-sets. Solution to this problem is CHUD (Closed High Utility Item-sets Discovery) algorithm with association matrix. CHUD finds the closed item-sets with high utility by considering an itemsets items quantity or units in the transaction as well as the profit unit of those items in item-set. Association matrix will hold the presence or absence of that item in each item. Further, a method called DAHU (Derive All High Utility Item-sets) is applied to recover all HUIs from the set of CHUIs without accessing the original database.*

***Keywords*** *— closed high utility itemsets, utility Mining, data mining.*

## 1. INTRODUCTION

The purpose of regular itemset mining is to discover items that constantly appear in a transaction database and higher than the frequency threshold given by the customer, without considering profit of the item. However, amount, weight and worth are important for addressing real world decision problems that require maximizing the utility in an organization.

Following two aspects involves as Utility of item in transaction database:

(1) The significance of items in transactions, called internal utility(i), and

(2) The significance of different items, called external utility(e)

**Utility of Itemset (U) = internal utility (i) * external utility (e)**

**Example:-**

Let Table 1 be a database containing five transactions. Each row in Table 1 represents a transaction, in which each letter represents an item and has a purchase quantity (internal utility).

**Table 1:** Transaction Database

| Trans_id | Transaction | Transaction Utility |
|---|---|---|
| T1 | A(1),B(1),E(1),W(1) | 5 |
| T2 | A(1),B(1),E(3) | 8 |
| T3 | A(1),B(1),F(2) | 8 |
| T4 | E(2),G(1) | 5 |
| T5 | A(1),B(1),F(3) | 11 |

**Table 2:** Unit Profits associated with items

| Item name | A | B | E | F | G | W |
|---|---|---|---|---|---|---|
| Unit Profit (in INR) | 1 | 1 | 2 | 3 | 1 | 1 |

## 2. IMPLEMENTATION

### 2.1 Algorithm

1) Step 1: Get the connection to the database on which CHUD to perform.
2) Step 2: Get the names of columns for required for high utility computation namely,
    - Transaction ID
    - Item sold
    - Quantity (Internal utility)
    - Item Id/Item name (whichever occurs in item sold)
    - Profit(External Utility)

    [Note: These names should be remember by the system]

3) Step 3: Compute transaction utility from each transaction to the database

    $TU(T1)= \sum^{n}_{Ti=1}$[Utility of j1+Utility of jn]n

4) Step 4: Get the minimum threshold from knowledge worker
5) Step 5: Classify the itemset item as promising and unpromising
6) Step 6: Arrange promising items in an increasing order of frequency count
7) Step 7: Creation of node repeatedly for all items in promising set- Prev-set, post-set,Trans-Id(tid)
8) Step 8: We perform SubsumeCheck

    ```
    Input: Node(N)
     For each item a  prev(X) do
         If(g(x)  g(a))
         [a preset of x]
    ```

9) Step 9: Compute closure if SubsumeCheck procedure returns false means that item is not process Compute closure of that item.
10) Step 10: Explore different possible combination for closed item sets discovered in step 9.
11) Step 11: Finally we get the high Utility item sets as a result.

### 2.2 Strategies

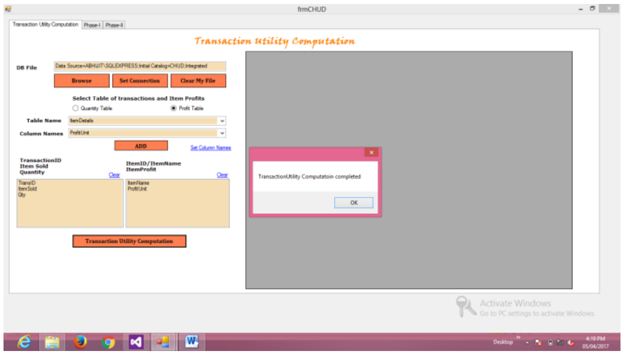
1) **Strategy 1: Ignore or discard unpromising items OR Consider only promising items :** In this strategy utilities of unpromising items discarded from the database and it's absolute utility should be subtracted from the TU(Tr).
2) **Strategy 2: IIDS (Isolated Items Discarding Strategy):** Discard isolated items and their actual utilities from transactions and transaction utilities of the database.
3) **Strategy 3: REG (Removing the Exact utilities of items from the Global TU-Table):** This strategy will discard utilities of an item $I_k$ from ordered list O each time when $I_k$ processed completely, and next item is taken as new current node.
4) **Strategy 4: RML (Removing the Mius of items from Local TU-Tables):** This strategy works in *Explore* procedure, each time when an item from the POSTSET of current node gets processed. The Minimum item utility removed from the LTU (Local Transaction Utility) table. A minimum utility of an item is defined as the miu(X,$T_r$) such that there is no transaction $T_s$ present that have u(X,$T_s$)<u(X,$T_r$).
5) **Strategy 5: DCM (Discarding Candidates with a MAU):** This strategy states that a candidate XC can be discarded from Phase-II if its estimated utility Estu(Xc) or mau(Xc) is less than abs_min_utility. Maximum item utility of item is

defined as mau(X,Tr) such that there is no transaction Ts present that have $u(X,T_s)$ > $u(X,T_r)$. XC is output with its estimated utility if it's Estu(Xc) or $mau(X_c)$ >= abs_min_utility otherwise discarded.

## 3. EXPECTED RESULT

The results from above algorithm are the collection of high utility itemsets which sold together and most frequently participating in high utility itemsets. Extraction of such itemsets from the sales database of any organization is a crucial task which includes proper use of memory and cpu usage. Algorithm processes only one node at a time, hence the memory requirement is less.

One most neglected aspect of deciding an itemset is high utility or low utility is deciding minimum threshold value.


frmCHUD
Transaction Utility Computation
Phase-I
Phase-II
Transaction utility Computation
Browse
Set Connection
Clear My File
ADD
Set Column Names
Clear
Clear
OK
Transaction Utility Computation
Activate Windows
Go to PC settings to activate Windows.
Desktop
4:19 PM
05/04/2017

DB File: Data Source=ABHIJIT\SQLEXPRESS;Initial Catalog=CHUD;Integrated

Select Table of transactions and Item Profits
○ Quantity Table ● Profit Table

Table Name: ItemDetails

Column Names: ProfitUnit

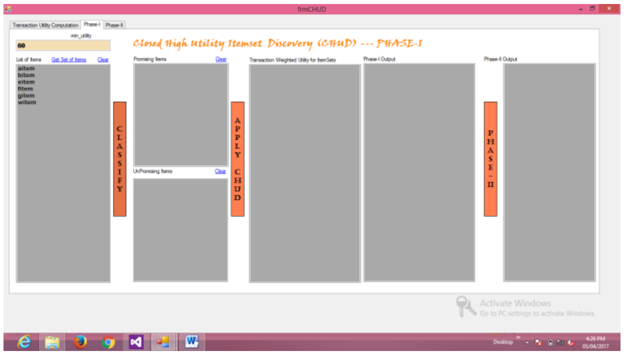
TransactionID
Item Sold
Quantity

TransID
ItemSold
Qty

ItemID/ItemName
ItemProfit

ItemName
ProfitUnit

TransactionUtility Computatoin completed



min_utility: 60

List of Items:
aitem
bitem
eitem
fitem
gitem
witem

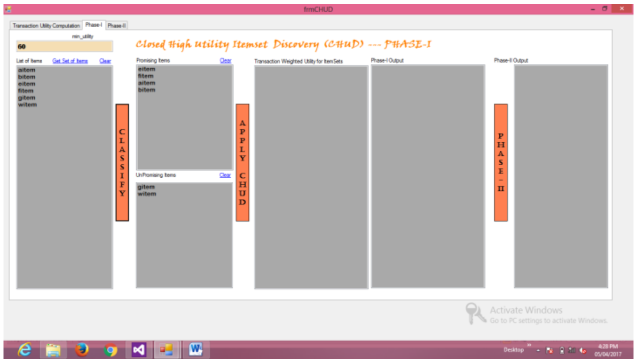
Promising Items

UnPromising Items

Transaction Weighted Utility for ItemSets

Phase-I Output

Phase-II Output



min_utility: 60

List of Items:
aitem
bitem
eitem
fitem
gitem
witem

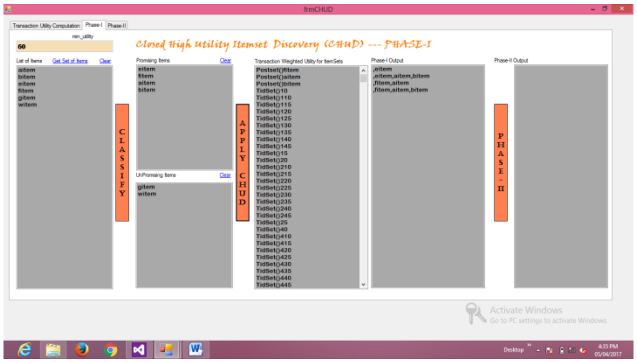
Promising Items:
eitem
fitem
aitem
bitem

UnPromising Items:
gitem
witem

Transaction Weighted Utility for ItemSets

Phase-I Output

Phase-II Output



min_utility: 60

List of Items:
aitem
bitem
eitem
fitem
gitem
witem

Promising Items:
eitem
fitem
aitem
bitem

UnPromising Items:
gitem
witem

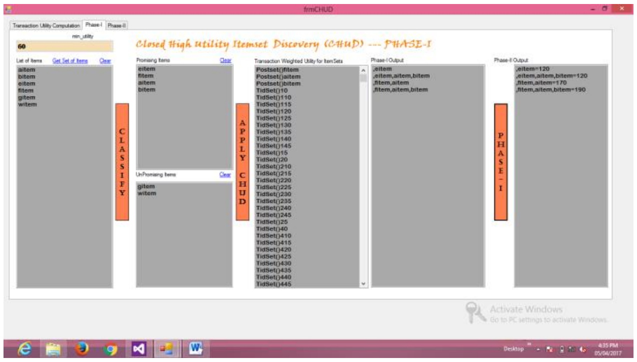
Transaction Weighted Utility for ItemSets:

```
Postset()fitem
Postset()aitem
Postset()bitem
TidSet()10
TidSet()110
TidSet()115
TidSet()120
TidSet()125
TidSet()130
TidSet()135
TidSet()140
TidSet()145
TidSet()15
TidSet()20
TidSet()210
TidSet()215
TidSet()220
TidSet()225
TidSet()230
TidSet()235
TidSet()240
TidSet()245
TidSet()25
TidSet()40
TidSet()410
TidSet()415
TidSet()420
TidSet()425
TidSet()430
TidSet()435
TidSet()440
TidSet()445
```

Phase-I Output:

```
,eitem
,eitem,aitem,bitem
,fitem,aitem
,fitem,aitem,bitem
```

Phase-II Output



min_utility: 60

List of Items:
aitem
bitem
eitem
fitem
gitem
witem

Promising Items:
eitem
fitem
aitem
bitem

UnPromising Items:
gitem
witem

Transaction Weighted Utility for ItemSets:

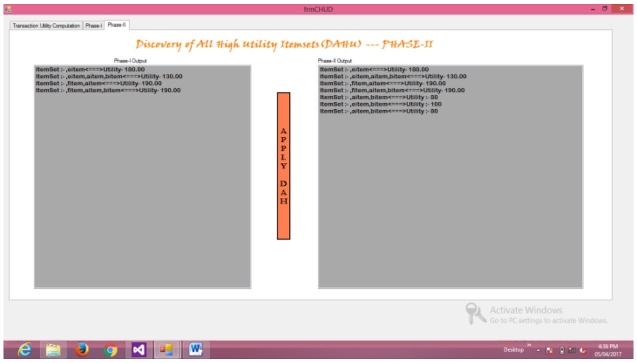
```
Postset()fitem
Postset()aitem
Postset()bitem
TidSet()10
TidSet()110
TidSet()115
TidSet()120
TidSet()125
TidSet()130
TidSet()135
TidSet()140
TidSet()145
TidSet()15
TidSet()20
TidSet()210
TidSet()215
TidSet()220
TidSet()225
TidSet()230
TidSet()235
TidSet()240
TidSet()245
TidSet()25
TidSet()40
TidSet()410
TidSet()415
TidSet()420
TidSet()425
TidSet()430
TidSet()435
TidSet()440
TidSet()445
```

Phase-I Output:

```
,eitem
,eitem,aitem,bitem
,fitem,aitem
,fitem,aitem,bitem
```

Phase-II Output:

```
,eitem=120
,eitem,aitem,bitem=120
,fitem,aitem=170
,fitem,aitem,bitem=190
```


frmCHUD
Transaction Utility Computation
Phase-I
Phase-II
Discovery of All High utility Itemsets (DAHU) --- PHASE-II
A P P L Y D A H
Activate Windows
Go to PC settings to activate Windows.
Desktop
4:36 PM
05/04/2017

Phase-I Output:

```
ItemSet :- ,eitem<===>Utility- 180.00
ItemSet :- ,eitem,aitem,bitem<===>Utility- 130.00
ItemSet :- ,fitem,aitem<===>Utility- 190.00
ItemSet :- ,fitem,aitem,bitem<===>Utility- 190.00
```

Phase-II Output:

```
ItemSet :- ,eitem<===>Utility- 180.00
ItemSet :- ,eitem,aitem,bitem<===>Utility- 130.00
ItemSet :- ,fitem,aitem<===>Utility- 190.00
ItemSet :- ,fitem,aitem,bitem<===>Utility- 190.00
ItemSet :- ,aitem,bitem<===>Utility :- 80
ItemSet :- ,eitem,bitem<===>Utility :- 100
ItemSet :- ,aitem,bitem<===>Utility :- 80
```

## 4. CONCLUSION

A person who likes to analyze, wants the accurate but compact result set at the end. Hence, only the creamy or rich result set must be expected at the output end, which makes user comfortable. CHUD algorithm discovers all closed high utility itemsets efficiently. Such efficiency of CHUD comes from the very good strategies works with it. CHUD had one more benefit that the numbers of joins it have to perform are less. POSTSET will help to find the new items which are not processed yet and join them to current node and form a new itemset. And PREVSET will help to check whether an item is already processed. Hence the redundancy of processing the same node will be drastically reduced. Overall CHUD will work fine to discover closed high utility itemsets.

## ACKNOWLEDGEMENT

We feel great pleasure in submitting this Major Project paper on "**A Review on High Utility Mining to Improve Discovery of Utility Itemset**". We wish to express true sense of gratitude towards our teacher and guide, **Prof. R. P. Chaudhari** who at every discrete step in study of this Major project contributed his valuable guidance and help us to solve every problem that arose. We would wish to thank our H.O.D., **Prof. D. D. Patil** and Principal **Dr. R. P. Singh** for opening the doors of knowledge towards the realization of the Project. Most likely we would like to express our sincere gratitude towards our family and friends for always being there when we needed them the most. With all respect and gratitude, we would like to thank all authors listed and not listed in references whose concepts are studied and used by us whenever required. We owe my all success to them.

## REFERENCES

[1]. Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu," Efficient Algorithms for Mining the Concise and Lossless Representationof High Utility Itemsets" IEEE Trans. Knowl. Data Eng ,VOL. 27, NO. 3, MARCH 2015

[2]. R. Chan, Q. Yang, and Y. Shen, "Mining high utility itemsets," in Proc. IEEE Int. Conf. Data Min., 2003, pp. 19–26.

[3]. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[4]. 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. 554–561.

[5]. U. Yun, "Efficient Mining of Weighted Interesting Patterns with a Strong Weight and/or Support Affinity," Information Sciences, vol. 177, pp. 3477-3499, 2007.

[6]. H. Yao, H.J. Hamilton, and C.J. Butz, "A Foundational Approachto Mining Itemset Utilities from Databases," Proc. Fourth SIAMInt'l Conf. Data Mining (SDM '04), pp. 482-486, 2004.

[7]. H. Yao and H.J. Hamilton, "Mining Itemset Utilities from Transaction Databases," Data and Knowledge Eng., vol. 59, pp. 603-626, 2006.

[8]. Y. Liu, W.-K. Liao, and A. Choudhary, "A Two Phase Algorithm for Fast Discovery of High Utility of Itemsets," Proc. Ninth Pacific-Asia Conf.

Knowledge Discovery and Data Mining (PAKDD '05), pp. 689-695, 2005.

[9]. Y. Liu, W.-K. Liao, and A. Choudhary, "A Fast High Utility Item sets Mining Algorithm," Proc. First Int'l Conf. Utility-Based Data Mining, pp. 90-99, 2005.