TEST Engineering & Management

January-February 2020
ISSN: 0193-4120 Page No. 12650 - 12656

High-Utility Pattern Mining by Removing Recurrent Patterns

[1]M.S.Bhuvaneswari, [2]S.Kavitha, [3]N.Balaganesh

[1], [2], [3]Mepco Schlenk Engineering College(Autonomous), Sivakasi

Article Info
Volume 82
Page Number: 12650 - 12656
Publication Issue:
January-February 2020

Article History
***Article Received**: 18 May 2019*
***Revised**: 14 July 2019*
***Accepted**: 22 December 2019*
***Publication**: 24 February 2020*

Abstract:
High Utility Itemset Mining (HUIM) algorithm is implied uniquely to identify maximum utility for an itemset at a particular time, yet it can't be utilized to discover itemsets that produce high utility. Such high utility itemsets are discovered for the products which may merchandise well for specific season but may not sell well for other time intervals. In this paper we have utilized the LHUIMiner to locate the particular time interim at which the pattern has high utility . The PHUI Miner algorithm is used for finding time interval where some products are sold with unusual profit. To remove the recurrent itemsets found within the same time window we have proposed a new algorithm Closed High Utility Itemset mining algorithm. The Closed High Utility Itemset Mining(CHUI) reduces the itemsets within specific time interval and reduces the processing time and memory usage.

Keywords: Frequent *Itemset Mining, High Utility Itemset, Peak High Utility Itemset, Closed High Utility Itemset.*

I. INTRODUCTION

Frequent Itemset Mining(FIM) identifies all group of items that occur repeatedly within user defined threshold. FIM believes that all items in the database are essential and can occur just one time in each sequence. To overcome this HUIM a method to find high profit patterns, is utilized to locate the arrangement of things that produce high benefit or it is of high significance to the user. HUIM is commonly considered as a more convoluted issue than FIM as the utility measure utilized in HUIM isn't hostile to montonic contrasted with the support utilized in FIM. The standard FIM cannot be straight forwardly utilized in HUIM to upgrade the pursuit gap. HUIM Two Phase calculation[1] ascertains an maximum limit on the utility measure to advance the hunt gap. The issue of HUIM isn't intended to discover model that clarifies how the utility of itemsetstemporaly changes.

To distinguish the itemsets which are beneficial in obscure timespans, another kind of model Local High profit patterns Mining was proposed. It gives an approach to discover itemsets that gives an utility which is more noteworthy than the client indicated limit, inside at least one time interim having a most reduced time length. LHUI-Miner is an algorithm that expands the HUI-Miner algorthim to mine local high utility itemsets. For instance to discover timeframes where client spends part of cash for the items.

Peak High Profit patterns PHUI [2] deals with identifying the timeline where an itemsets has utility higher the usual profit. For example to find time interval where some items are sold together and produce profit more than the usual such as selling schoolbags and notebooks during reopening of school.

HUIM algorithm tolerate from long response times and even problem to execution due to large storage. The issues are solved by Closed High Utility Itemsets(CHUIs) Mining. The possibility of CHUI brief the idea of shut model from FIM.A CHUI is laconic by providing lossless representation result to the user.

January-February 2020
ISSN: 0193-4120 Page No. 12650 - 12656

II. Related Work

Frequent Itemset Mining(FIM) is a famous system for discovering itemset which are over and over obtained by customer[4]. Apriori algorithm examines the hunt gap of itemsets utilizing an expansiveness breadth-first search. The set of items is received by scanning the database and only repeated item are displayed to the user. Apriori does not take into consideration the whole search gap of frequent itemsets. The obstacle for using Apriori is that it several times reviews the database.

The Elcat algorithm [5] is a FIM algorthim, to overcome the problem in Apriori.TheElcat algorithm review the database once to make a column for each item denoting the transaction where it occurs. If an itemset has more than one item the columnar structure can be obtained without scanning the database by connecting the column of two of its subgroups. Elcat explore pursuit space of itemsets by depth first search.

High Utility Itemset Mining(HUIM) finds out group of items creating large profit in a transactional information [6]. To identify the relation between the items that yields high profit and occur in multiple transaction, a Sequential rule algorithm HUSRM was proposed [7]. High utility consecutive example mining[8] proposed a tight upper bound on the utility of the itemsets to enhance the hunt space challenge. CRoM was used for pruning the itemsets before generation of appropriate patterns.

In [9] various least utility limit was utilized as opposed to utilizing single utility as everything is unique and they can't be treated likewise. To satisfy HUIM with numerous minimimum utility limit (HUIM-MMU), another sorted downward closure (SDC) property and least minimum utility threshold (LMU) was proposed.

The periodic high-utility itemsets mining [10] gain knowledge of itemsets that are now and again purchased by buyer and yield a high profit. The objective of short-period high-utility itemsets (SPHUI) is to distinguish designs that are engaging both as far as time interim and utility.

The time decaying model [11], calculation decrease the utility of transactions under to allot greater weights to latest data as against to those of oldest. The algorithm periodically refresh the utility of the data in its data structure and cuts the nodes whose utility values is smaller than the thresholds. Maintains a sensible storage by removing unwanted storage space.

A discriminative algorithm called CDSPM (Conditional Discriminative Sequential Pattern Mining) was proposed by [12] to resolve the subgroup-induced recurrent issue. [13] proposed a technique for finding high utility itemsets from the transaction database which includes items with negative profit. [14] made use of the sequence tree for mining high utility sequences.

[15] utilized the stream rather than the static databases and utilized a high normal utility pattern digging system for recognizing the examples of intrigue.

Top-k co-occurrence items were discovered from the sequential database [16] by directly examining the database in linear order to find the number of times an item has occurred. Using vertical approach mining(VAM) and vertical with index approach mining(VIAM) the searching gap in the database was reduced.

Episode mining and episode rule mining are used to identify a episodes with larger event count and confidence in a single sequence pattern [17]. [18] proposed a method to mine sequential rules from more than one sequential dataset.

III. System Design

The entire process involved in the system is shown in Fig.1. The modules involved are i) Peak High Utility Itemset Mining (PHUIM) ii) Closed High Profit Sequence Mining. The peak high utility itemset produced by PHUIM is given as input to the CHUIM which in turn removes the recurrent itemsets within the same time window and produces closed high utility itemsets.

TEST Engineering & Management

January-February 2020
ISSN: 0193-4120 Page No. 12650 - 12656

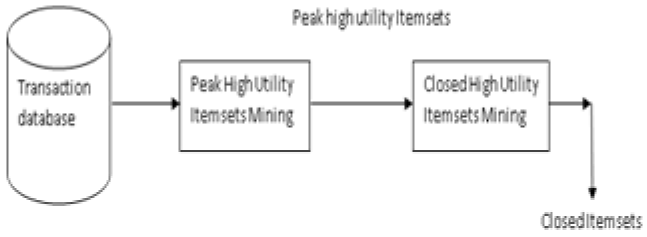

Fig 1. System Design

A. Peak High Utility Itemset Mining(PHUIM)

The PHUIMiner algorithm proposed in [2] is used for finding the peak high utility itemset. PHUI manages finding the time gap where a pattern has profit higher than the standard benefit. The idea of focal moving normal hybrid is utilized to discover the beginning and ending timespan where the benefit of itemsets is a lot higher than the standard thing.

Consider a transaction database as shown in Table 1. For instance, (b, 1) specifies that the name of the item is b and 2 represents the number of the items sold. Timestamps specifies the month during which the items were sold. Table 2 is the utility table where Utility represents the profit of selling the item(profit/unit).

Table 1 Transaction database

TID	Itemsets	Timestamps
1.	(b,1), (c,2), (d,1), (e,1)	M1
2.	(a,4), (b,1), (c,3), (d,1), (e,1)	M3
3.	(a, 4), (c,2), (d,1)	M3
4.	(c,2), (d,1), (e,1)	M5
5.	(a,5), (b,2), (d,1), (e,1)	M6
6.	(a,3), (b,4), (c,1), (d,2)	M7
7.	(c,1), (d,5)	M9

Table 2 Utility Table

Item	a	b	c	d	e
Utility	5	2	3	5	4

Let D={d_1,d_2,....d_n}be a set of items(itemset). Each transaction(T) has a unique identifier(tid). The utility of an itemset D in transaction T (u(D,T)) is the sum of the utilities of all the items contained in D that appears in T. The utility of itemsets u(D) is sum of the utilities of D in all the transactions in database. Remaining utilityof an itemset D(rem(D)) is defined as the sum of u(D) and the utilities of the remaing items that occur in the transactions were D occurs.

Example 1: The utility of itemset {c,e} can be calculated as follows:

{c,e} occurs in transaction t1,t2 and t4 , hence

u({c,e},)=(2*3)+(1*4)+(3*3)+(2*4)+(2*3)+(1*4)
= 38

Example 2: Theremaining utility of {c,e} can be computed as follows:

u({c,e})= 35

{c,e} occurs in transactions t1,t2,t4

The utilities of items(b,d) other than c and e in t1=1*2+1*5=7

The utilities of items(a,b,d) other than c and e in t2=4*5+1*2+1*5=27

The utilities of items(d) other than c and e in t4=1*5=5

rem({c,e})= 38 +(7+27+5)
= 77

Extended Utility List(EUL) Structure:

It is used for storing the information about the itemset . The database is scanned and EUL is constructed for each item in the dataset. The items are combined and EUL is constructed for the newly formed itemset and the process is repeated. The EUL can be utilized for identifying the PHUI without scanning the database. The EUL of item **a** for the util period(M3,M7)is given in Table 3.

Table 3 EUL(a)

January-February 2020
ISSN: 0193-4120 Page No. 12650 - 12656

Tid	u(a,Tid)	rem(a)
T2	20	20
T3	20	11
T5	25	13
T6	15	21

The algorithm for finding peak high utility itemset is given below:

Algorithm findPHUI
Input :
D: an itemset , $\bar{\boldsymbol{D}}$: extensions of D,
$\bar{\boldsymbol{D}}$x: expanding D with an item x,
$\bar{\boldsymbol{D}}$y: expanding D with an item y ,
MinUtil: a user-specified threshold,
windlength : a window length threshold ,
Extendsof$\bar{\boldsymbol{D}}\boldsymbol{x}$: $extensionof\bar{D}x$,
Eul:extends of utility list ,
β: threshold
Output: The set of max high utility itemsets and their maximum periods.
1. For each itemset $\bar{D}x$ ε $\bar{D}$ do
2. If $\bar{D}x$.Eul.inutilPeriods ≠ Ø
then output $\bar{D}x$ with $\bar{D}x$ .Eul.inutilPeriods;
3. If $\bar{D}x$.Eul.utilPeriods ≠ Ø then
4. Extendsof$\bar{D}x$ ← Ø ;
5. For itemset $\bar{D}y$ ε $\bar{D}$ such that $\bar{D}$y >$\bar{D}$x do
6. $\bar{D}$xy.Eul ← generate($\bar{D}$, $\bar{D}$x, $\bar{D}$y)
7. generatePeaks ($\bar{D}$xy, MinUtil, windlength, β);
8. Extendsof$\bar{D}$x ← $\bar{D}$x U $\bar{D}$xy;
9. End
10. findPHUI($\bar{D}$x, Extendsof$\bar{D}x$,MinUtil, windlength);
11. End
12. End

B. Closed High Utility Itemset Mining(CHUIM)

The peak high utility itemsets produced by PHUIM is given as input to the CHUIM. It deals with identifying closed high utility itemsets from the PHUI for each peak window. If n peak itemsets are there within the timewindow [i, j] and m itemsets are there within the timewindow [k,l] then CHUIM is applied first to itemsets in time window[i,j] then to itemsets in window[k,l]. If peak itemset has a superset within the same time window and the difference in utility(profit) between them is above a threshold then the superset is added to the list of closed high utility itemset. If the difference is below a threshold then subset is added to the list of closed high utility itemset. The CHUIM is applied to all the peak windows and the closed high utility itemsets in each peak window is obtained. The process is repeated once again for the peak window until there is no change in the itemsets identified in the peak windows.

Algorithm find CHUI
Input: PHUI- Peak High Utility Itemsets
Output: Closed High Utility Itemsets(CHUI) with peak window
1.MinUtil=30;
2.CHU I={}
3. D←constructSubset(D); *//groups itemsets within same peak window*
4.for p in D
5. D′ = itemsets in p
6. while |D′| >1 do
7. for d1 in D′:
8. for d2 in D′:
9. if d1!=d2 and
d1.start index== d2. start index and
d1.end index== d2.end index do
10. if d1 ⊆ d2 || d2 ⊆ d1 do
11. diff= |util(d1)- util(d2)|
12. if diff <Minutil;
13. if |d1|< |d2|
14. D′= D′- {d1}
15. CHUI+={d1}
16. else if |d2| < |d1|
17. D′= D′-{d2}
18. CHUI+={d2}
19. else
20. if |d1|< |d2|
21. D′= D′- {d2}
22. CHUI+={d2}
23. else if |d2| < |d1|

```
                D′= D′-{d1}
                CHUI+={d1}
            end

    end
   end
  end
 end
return CHUI
```

CHUIM is used for finding itemsets which yields more utility at peak window. This reduces the number of redundant temsets and it turn reduces storage and processing time.

IV. Results and Discussion

The proposed Closed High utility Itemset Mining Algorithm is tested on Kosarak and Retail dataset collected from:

https://philippe-fournier-viger.com/spmf/index.php?link=datasets.php

The kosarak dataset consists of 33330 transactions and the Retail dataset consists of 80915 transactions.

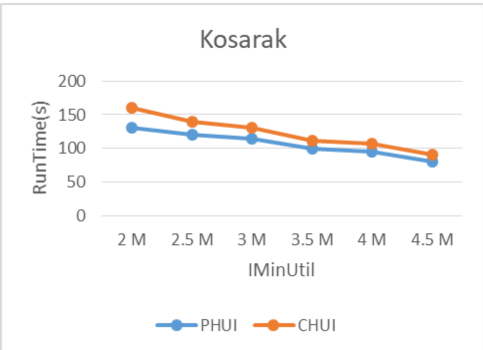

Fig 2a. IMinUtil vs RunTime for Kosarak dataset

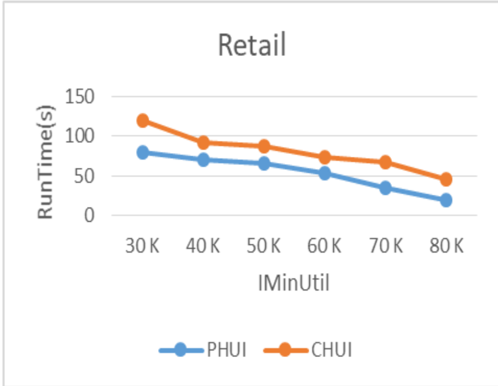

Fig 2b. IMinUtil vs RunTime for Retail dataset

Fig 2a, 2b shows the runtime for different lMinUtil values in Kosarakans Retail dataset.The runtime to find CHUI is more compared to that of the PHUI as it has to compute PHUI then CHUI.

Fig 3a and 3b represents the number of itemsets generated for different lMinUtil values in Kosarak and Retail datasets.As the lMinUtil value increases the number of itemsets decreased.

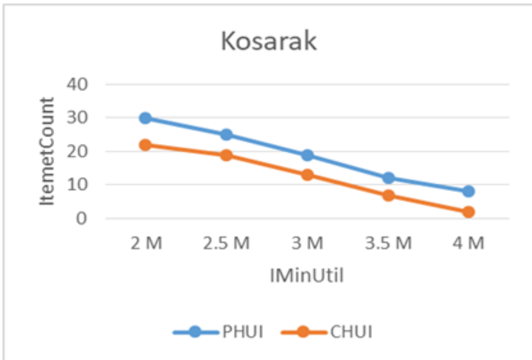

Fig 3a. IMinUtil vs ItemCount for Kosarak dataset

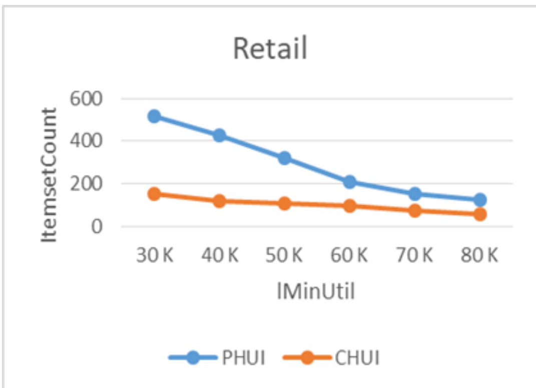

Fig 3a. IMinUtil vs ItemCount for Retail dataset

The storage space (MB) utilized by the

TEST Engineering & Management

January-February 2020
ISSN: 0193-4120 Page No. 12650 - 12656

itemsetsgenerated by PHUIM and CHUIM is given in Table 4. The memory consumed by the closed high utility itemset is less compared to peak high utility itemsets.

Table 4 Memory Usage

DataSet	Memory Usage(MB)	
	PHUI	CHUI
Retail	320	115
Kosarak	80	30

CONCLUSION

To discover the pinnacle high utility itemset with pinnacle window PHUIMiner algorithm is utilized. As there may be repetitions in the itemset generated CHUIM algorithm is used to find the closed high utility itemset within same peak window. Closed High Utility Itemsets Mining(CHUIM) utilize the Eul structure which decides the entire arrangement of CHUIs in the entire database without creating any applicants. CHUIs is laconic and produces lossless reprenstation to the user.

REFERENCES

1. Liu, Y., Liao, W., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: Proc. 9th Pacific-Asia Conf. on Knowl. Discovery and Data Mining, pp. 689–695 Springer, Hanoi (2005)
2. Fournier-Viger, P., Zhang, Y., Lin, J. C.W., Fujita, H., Koh, Y.S. (2019). Mining Local and Peak High Utility Itemsets. Information Sciences, Elsevier, 481: 344-367
3. Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: Proc. 22nd ACM Int. Conf. Info. and Know. Management, ACM, pp. 55–64 (2012)
4. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. 20th Int. Conf. Very Large Databases, pp. 487–499, Morgan Kaufmann, Santiago de Chile (1994)
5. Zaki, M. J.: Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. 12(3), 372–390 (2000)
6. Zida, S., Fournier-Viger, P., Lin, J. C.-W., Wu, C.-W., Tseng, V.S.: EFIM: A Highly E_cient Algorithm for High-Utility Itemset Mining. In: Proc. 14th Mexican Int. Conf. on Artificial Intelligence, pp. 530–546, Springer (2015)
7. Zida, Souleymane & Fournier Viger, Philippe & Wu, Cheng-Wei & Lin, Chun-Wei & S. Tseng, Vincent. (2015). Efficient Mining of High-Utility Sequential Rules. 157-171. 10.1007/978-3-319-21024-7_11
8. Alkan, O. K., Karagoz, P.: Crom and huspext: Improving eficiency of high utility sequential pattern extraction. IEEE Trans. on Knowledge and Data Engineering, 27(10), 2645–2657, (2015)
9. Jerry Chun-Wei Lin, Wensheng Gan, Philippe Fournier-Viger, and Tzung-Pei Hong. 2015. Mining High-Utility Itemsets with Multiple Minimum Utility Thresholds. In Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering (C3S2E '15). ACM, New York, NY, USA, 9-17. DOI: https://doi.org/10.1145/2790798.2790807
10. Lin, J. C. W., Zhang, J., Fournier-Viger, P., Hong, T.P., Zhang, J.: A two-phase approach to mine short-period high-utility itemsets in transactional databases. Advanced Engineering Informatics, 33, 29–43 (2017)
11. Kim, Donggyu& Yun, Unil. (2016). Mining high utility itemsets based on the time decaying model. Intelligent Data Analysis. 20. 1157-1180. 10.3233/IDA-160861
12. He, Zengyou& Zhang, Simeng& Gu, Feiyang& Wu, Jun. (2018). Mining Conditional Discriminative Sequential Patterns. Information Sciences. 478. 10.1016/j.ins.2018.11.043
13. Philippe Fournier-Viger and Souleymane Zida. 2015. FOSHU: faster on-shelf high utility itemset mining -- with or without negative unit profit. In Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15). ACM, New York, NY, USA, 857-864. DOI: https://doi.org/10.1145/2695664.2695823
14. Yin, Junfu& Zheng, Zhigang& Cao, Longbing. (2012). USpan: An efficient algorithm for mining high utility sequential patterns. KDD 2012. 10.1145/2339530.2339636
15. Yun, Unil& Kim, Donggyu& Yoon, Eunchul& Fujita, Hamido. (2017). Damped Window based

January-February 2020
ISSN: 0193-4120 Page No. 12650 - 12656

High Average Utility Pattern Mining over data streams. Knowledge-Based Systems. 144. 10.1016/j.knosys.2017.12.029

16. Tung Kieu, Bay Vo, Tuong Le, Zhi-Hong Deng, and Bac Le. 2017. Mining top-k co-occurrence items with sequential pattern. Expert Syst. Appl. 85, C (November 2017), 123-133. DOI: https://doi.org/10.1016/j.eswa.2017.05.021
17. HeikkiMannila, HannuToivonen, and A. InkeriVerkamo. 1997. Discovery of Frequent Episodes in Event Sequences. Data Min. Knowl. Discov. 1, 3 (January 1997), 259-289. DOI: https://doi.org/10.1023/A:1009748302351
18. Harms, S.K., Deogun, J.S., & Tadesse, T. (2002). Discovering Sequential Association Rules with Constraints and Time Lags in Multiple Sequences. ISMIS.