International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Towards Efficient Mining of Periodic High-Utility Itemsets in Large Databases

P. Lalitha Kumari, S. G. Sanjeevi , T.V Madhusudhana Rao

Abstract: ***High Utility Item sets mining has attracted many researchers in recent years. But HUI mining methods involves a exponential mining space and returns a very large number of high-utility itemsets. . Temporal periodicity of itemset is considered recently as an important interesting criteria for mining high-utility itemsets in many applications. Periodic High Utility item sets mining methods has a limitation that it does not consider frequency and not suitable for large databases. To address this problem, we have proposed two efficient algorithms named FPHUI(mining periodic frequent HUIs), MFPHM(efficient mining periodic frequent HUIs) for mining periodic frequent high-utility itemsets. The first algorithm FPHUI miner generates all periodic frequent itemsets. Mining periodic frequent high-utility itemsets leads to more computational cost in very large databases. We further developed another algorithm called MFPHM to overcome this limitation. The performance of the frequent FPHUI miner is evaluated by conducting experiments on various real datasets. Experimental results show that proposed algorithms is efficient and effective.***

Keywords: Periodic occurrence, frequent periodic itemsets, High Utility itemsets, Large Transactional databases

I INTRODUCTION

Frequent Itemset mining can be considered as an important task in Association Rule Mining [1],[2]. Some factors are not considered in ARM, such as profit and quantity. For example, in retail market, it is necessary to find itemsets with high profit to improve market strategies, make advertisements etc.An itemset with high frequency may not be interested, rather than users may be interested in itemsets with high profits for decision making. High-utility itemset mining is thus introduced to overcome some limitations in ARM. HUI can be thought of as an extension of frequency itemset mining by considering profits and quantities. Utility of an itemset can be found out by using profit and quantity. All the items in database are considered as equal importance in Traditional Association rule mining model. It checks whether item is present in it or not. Frequent itemset mining [1],[2], which identifies frequent itemsets in data base, has a small portion of contribution towards overall profit, whereas non-frequent itemsets may contribute a large portion towards profit.

Manuscript published on November 30, 2019.
* Correspondence Author
P. Lalitha Kumari*, Dept of CSE, National Institute of Technology, Warangal Dist, Telangana-
S. G. Sanjeevi, Dept of CSE, National Institute of Technology, Warangal Dist, Telangana-
T.V Madhusudhana Rao, Dept of CSE, Sri Sivani College of Engineering , Srikakulam, Andhra Pradesh

Frequent itemset mining may not be suitable for identifying high-utility itemsets. Weighted association rule mining was introduced [10],[11],[12],[13] to address this limitation.
For a retail business, it is necessary to identify its most valuable customers as these customers contribute more profit to the business. But these customers may not appear in more number of transactions. To address this issue, an additional parameter is added to frequent itemset mining i.e. weights or profits. High-utility itemset mining refers to mine itemsets with high profits. Traditional frequent itemset mining algorithms are not suitable to mine high-utility itemsets as mining HUIs is difficult than FIM. Recently many algorithms have been proposed to efficiently mine HUIs [6],[7]. The main difficulty in mining HUIs is that it does not maintain monotonicity or anti-monotonicity property. This property is used to reduce search space. For any frequent itemset, all its subsets must be frequent too and similarly all the supersets of infrequent itemset cannot be frequent. This property is identified as powerful which is used to provide a pruning strategy to these algorithms. In frequent itemset mining process, if algorithm identifies infrequent itemset, then no need to check its supersets further. Thus, it reduces search space for mining. However mining for frequent itemsets only considers presence or absence of an itemset. High average utility itemset mining[24],[25],[26] has been introduced to mine itemsets with high average utilities. To measure average utility of an itemset, summation of items utilities is divided by the length of the itemset. Many algorithms have been proposed to mine high average utility itemsets for incremental database[27] also.
A periodic high utility itemset is an itemset that appears at regular intervals specified by the user in the transactional database. For a given user-specified minimum utility (minutil) and maximum periodicity (maxprd) constraints, an itemset's utility and periodicity should satisfy respectively.
Minutil identifies minimum utility that the itemset must have its utility in the database. Maxprd identifies the maximum time difference between two consecutive appearances of itemset in the database. Temporal periodicity of itemset is considered recently as an important interesting criteria for high-utility itemsets in many applications. Periodic high utility itemsets mining [28] algorithm was proposed to find all high utility itemsets that appear at regular intervals. Researchers introduced three novel measures to mine periodic high utility itemsets. The limitation of this approach is it does not consider support of high utility itemsets. And it is very difficult to mine periodic high utility itemsets in very large databases using this approach, as it contain more number of different itemsets.
Generally, the time duration between to two consecutive purchases of itemsets of soaps and shampoo may be longer than the time duration between two consecutive purchases of bread and milk.

By our proposed algorithm, we are able to mine those itemsets which generates more profit and their occurrences are at regular intervals.

Some itemsets are quite interested which generate not only high profit, but also appear frequently with regular occurrences.

In this paper, we proposed an efficient approach to mine frequent periodic high-utility itemset with user specified utility, minutil and user specified maximum periodicity, maxper and minimum support. We also proposed another algorithm to minimize the computational cost of mining. The main contributions of this paper are (1) we integrate the concept of frequency into the periodic high utility itemsets to derive new patterns, (2) we present a novel algorithm FPHUI miner to derive the above patterns, and (3) we further develop another algorithm for reducing the computation cost of the FPHUI miner algorithm, (4) we develop pruning strategy by introducing the local periodicity of an itemset.

The rest of the paper is organized as follows. We discuss the background on mining periodic-high-utility itemsets in transactional databases in Section 2. In Section 3, we discuss the related work and introduce the proposed approach. Apriori based approach to mine periodic-high-utility itemset is discussed in Section 4. In Section 5, experimental results are reported. Conclusions are discussed in Section 6.

II RELATED WORK

Many studies have been discussed about high-utility itemsets mining and periodic frequent itemset mining. Many efficient algorithms to mine high-utility itemsets have been proposed. For mining high-utility itemset, two phase algorithm was proposed by Liu[5]. In this, transaction-weighted utility introduced as upper bound to utility to maintain the downward closure property, and to prune search space. Liu.M.et.al.[3] proposed an approach to mine high-utility itemset without generating candidates. They proposed novel list structure called Utility-lists structure to maintain its utility information about each itemset to mine all high-utility itemsets efficiently.

Up-growth algorithm was proposed by Tseng et.al., [6],[7] for mining high-utility itemset. In this, they proposed pattern growth based approach which maintains information about itemsets in tree structure. It needs two scans of database to mine HUIs. By using various strategies, i.e., DLU, DGU, DGN and DLN, candidate itemsets are pruned during mining process efficiently. HUI-miner was proposed by Liu et.al.,[8] with Tight overestimated utility strategy for pruning the search space. Many other studies have also been proposed to extract high-utility itemsets. Varient representations of High utility itemsets like concise representation of high-utility itemsets mining have been proposed in later studies such as closed HUI[9] , maximal HUI[14], and generators of HUIs[8]. Wu et.al.,[9] proposed three algorithms to efficiently mine concise representation of high-utility itemsets. HUIM on incremental, dynamic database and on data streams has been proposed in [14],[15],[16],[17],[18]. High-utility itemsets mining with negative unit profit has been done by using efficient algorithm FHN proposed by Philippe-Fournier et.al.. FHN discovered HUIs without generating candidates and introduced several strategies to handle items with Negative unit profits efficiently. To avoid difficulties in setting a proper utility threshold, in [17],[18],[19] attempts were done to mine a set of k-itemsets with the highest utility. Efficient algorithms have been proposed to mine high-utility itemsets in dynamic databases[4]. HUIM leads to generate huge number of itemsets. All itemsets may not be interested to user to make decisions. Some studies have been proposed according to the user interestingness like regularity, length. Efficient algorithms have been proposed to mine HUIs by integrating these constraints [25],[26],[27],[28].

Recently, Srikumar [29] proposed H-Miner to efficiently mine high utility itemsets. In this approach, novel utility list structure and novel hyperlink data structure have been proposed to maintain itemset's information. H-Miner algorithm has proved to that it has better performance than state of the art algorithms. Execution time have been reduced to thirty percent for several benchmark datasets. Memory consumption has also been improved in order of its magnitude. Zida et al., 2017[30] proposed EFIM to efficiently mine HUIs in transactional databases. Based on the transaction merging, database projection, authors developed an algorithm for fast mining of HUIs. Further, they introduced two pruning strategies to efficiently mine HUIs. They proved that their approach is faster than state-of-the-art approaches.

Komate Amphawan et. al.,[35] proposed bottom-up approach to concept of the approximate periodicity of each itemset. They presented novel tree-based structure called Interval Transaction-ids List tree to store the information about each itemsets in compact manner for mining periodic frequent itemsets. They proved that their data structure efficient for mining periodic-frequent itemsets.

Periodic frequent itemsets mining refers to observe occurrence behavior of itemsets with respect to frequency and regularity of occurrence. Mining of periodic frequent itemsets has various applications such as genetic and medical data analysis[20], to observe the behavior of moving objects analysis[21], web click stream analysis and behavior analysis of game player [22]. Tanbeer et.al.,[15] has proposed periodic frequent itemsets to address the problem of mining periodic occurrences of frequent itemset. Kiran et.al., [16] proposed efficient approach to discover all periodic-frequent itemsets by eliminating aperiodic itemsets based on suboptimal solutions. To reduce search space efficiently, they proposed two pruning techniques. These pruning techniques are used to find aperiodic itemsets at local level first, and then at global level. and Surana et.al., [23] developed an efficient model based on the multiple support and periodicity to mine rare periodic itemsets. Their proposed model satisfied downward closure property.

Komate Amphawan et al.,[36] proposed a single pass algorithm TFRC-Mine to mine the top-k frequent-regular closed patterns with minimal length. They discovered a new compact bit-vector method to prune uninterested candidate itemsets. It is computationally effective and efficient in terms of memory usage. Ashis Kumar Chanda et al.,[37] proposed an algorithm for mining weighted periodic pattern in time series databases. This proposed framework is capable to find out full,partial and symbol periodic patterns in single run.

HUIM is considerably more advantageous to traditional FIM since FIM has the limitation that it only takes support into consideration. HUIM reveals all itemsets with more profit. This makes HUIM more interesting in decision making strategies.

HUIM generates more useful itemsets than FIM. HUIM accompanies several other problems in its output. It may have many non interesting itemsets in huge number. Periodicity of high utility itemset is often useful for experts since it reveals all itemsets not only with high profit but also it will check these itemsets are appearing at regular intervals or not. FPHUI miner algorithm was proposed to mine periodic HUIs. The drawback of this approach is that it still generates more number of FPHUIs. This approach may not be suitable for mining FPHUIs in very large databases. It does not give importance to the support of itemsets. Our proposed algorithm firstly finds all FPHUIs by taking frequency into account.

The approaches mentioned previously for pattern mining considers frequency information, or utility or occurrences of itemset at regular periods. Some studies have been discussed by considering multiple constraints for mining itemsets based on user interestingness. In this model, we introduce discovery of frequent high-utility itemsets with regular occurrences of itemsets. This model provides the wider range of knowledge by taking into consideration not only utility, frequency of itemsets, but also their regular occurrences in the transactional database. This model enables to gain the valuable knowledge about high-utility itemsets with frequent and regular appearances.

III BACKGROUND

A Utility mining

Let I={i1, i2, i3, . . . , in} be a set of items and DB be a database composed of table of utilities and table consisting of transactions. Each item in I is associated with a utility value in the utility table. In the transaction table, Table 4, a unique identifier (tid) is assigned to each transaction T and is a subset of I, in which each item is associated with a count value. An itemset is a subset of I and if it contains k items, then it is called a k-itemset.

Definition 1. The utility value of i in the utility table of DB is called as the external utility of item i, i.e., eu(i).

For example, the eu of item b is 2 given in the Table 5 (Utility Table)

Definition 2: The count value associated with i in T in the transaction table of DB is called as the internal utility of item, iu(i, T), in transaction T

For example, internal utility of item b in T1 is 5, i.e., iu(b, T1) is 5 from Table 4(Transaction table)

Definition 3. The product of iu(i, T) and eu(i), where u(i, T) = iu(i, T) × eu(i) is the utility of item i, u(i, T), in transaction T.

For example, the utility of item b in transaction T1 is the product of iu(b) and eu(b) in T1. i.e., u(b) in T1 is 5*2 = 10.

Definition 4. The utility of itemset X in transaction T, u(X, T), is defined as the sum of the utilities of all the items in X in T in which X is contained, where u(X, T) =$\Sigma_{i\in X \wedge X\subseteq T}$ u(i, T).

For example, utility of itemset (ab) in transaction T1, u(ab,T1) = 1*5 + 2*5 = 15.

Definition 5. The utility of itemset X, denoted as u(X), is the sum of the utilities of X in all the transactions in which X appears in database, where u(X) = $\Sigma_{T\in DB \wedge X\subseteq T}$ u(X, T). For example, utility of itemset (ac) in transaction DB, u(ac,DB) = u(ac, T1) + u(ac, T3) +u(ac,T4). i.e., u(ac, DB) is 6+6+16 = 28.

An itemset X is called as High Utility Itemset if its utility is satisfied with given user-specified minimum utility threshold denoted as minutil, or the product of a minutil and the total utility of a mined database if the minutil is expressed as percentage. For a given database and minutil, the problem of mining high utility itemsets can be defined as to discover those itemsets from the database, whose utilities are not less than minutil. It should be noticed that the downward closure property of itemsets does not hold for high-utility itemsets (HUIs).

For example, assume a transaction database has only one transaction, {a, 1; b, 1}, where eu(a) = 1 and eu(b) = 2. And if minutil is 6, then for u({a}) = 5, u({b}) = 10 and u({a, b}) = 15, {b} and {a, b} are high-utility itemsets and {a} is not. That indicates that the downward closure property does not valid for high-utility itemsets. This shows that the mining of high-utility itemsets is much more challenging than frequent itemset mining.

An itemset X is said to be Periodic HUI if utility of X is not less than minutil and its periodicity should not be greater than maxper, For example, if we set minutil is 30, maxper is 3, and minimum support with 2 for the given transactional database, all frequent periodic high-utility itemset are (d,b),(d,b,e),(d,b,e,c),(d,b,c),(b,e),(b,e,c),(a,e,c) whose periodicity is 3 and support is 2.

Work done	Their contribution
Tanbeer, S.K et al.	Discovering complete set of periodic-frequent patterns in a database for user-given periodicity and support thresholds using PF Tee structure
Liu.M et al.	HUIM algorithm for mining high utility itemsets using novel structure called Utilitylist.
Tseng V.S et al.	Up-Growth algorithm for mining HUIs using UP(Utility Pattern)-Tree
Weng C-H	Discovering highly expected utility itemsets using FIHUM algorithm that effectively identify frequent itemsets with high utility (frequent HUIs) without generating many high-utility candidate itemsets.
Tseng V.S et al.	Proposed a novel framework for mining closed^{+} high utility itemsets(CHUIs) which serves as a compact and lossless representation of HUIs
Pauray S.M. Tsai	Proposed a single pass algorithm for high utility itemset mining based on the weighted sliding window model.
Lin C.W	Discovering high average-utility itemset mining (HAUIM) was proposed. It introduces the average utility measure, which considers both the length of itemsets and their utilities.

Amphawan k et al.	Discovering periodic-frequent patterns using ITL-tree (Interval Transaction-ids List tree)

If Transaction Weighted Utility of each item >=minutil then transactions are revised according to the ascending order of the TWU of each item. Transaction Weighted Utility of each item is shown in Table 3. If the minutil is set as 30, then all items are having their TWU greater than minutil, i.e., 30 , no item is deleted . If we set minutil is 40 then items f and g are deleted, then transactions are revised according to the ascending order of their TWUs. The arrangement of each item according to their TWU is as follows f > g >d > b > a > e > c

Table 2: Comparison of various approaches

Characteristic	Type of itemsets	Threshold
High utility itemsets HUI miner	High utility itemsets	minutil
Frequent high utility FHUI miner	Frequent high utility itemsets	minsup, minutil
Frequent itemset FIM	Frequent itemsets	minsup
Periodic frequent PFIM	Periodic frequent itemsets	minsup, maxper
FPHUI-proposed algorithm	Periodic frequent high utility itemsets	Minsup, minutil, maxper

Table 3 : Comparison of PHM and proposed algorithms

Algorithm	S	U	P	Large databases
HUIM	√	√	×	×
PHM	×	√	√	×
FPHUI	√	√	√	×
MFPHM	√	√	√	√

Table 4:Transactional database

Tid	Transaction	Count
1	{c, e, a, b, d, f}	{1, 3,5,5,3,1}
2	{c, e, b, d}	{3,3,4,3}
3	{c, a, d}	{1,5,1}
4	{c, e, a, g}	{6,6,10,5}
5	{c, e, b, g}	{1,1,2,2}

Table 5 : Utility table

a	b	c	d	e	f	g
1	2	1	2	1	1	1

Table 6: TWU (Transaction Weighted Utility)

Item	a	b	c	d	e	f	g
TWU	65	61	96	58	88	30	38

Table 7 High-utility itemsets with minutil 30, maxper 3, support 2

Initial Periodic Utility-lists

f

Tid	iutil	rutil	period
1	5	25	1

g

Tid	iutil	rutil	period
4	5	22	3
5	2	9	1

a

Tid	iutil	ruti
1	5	25
3	5	3
4	1	9

d

Tid	iutil	rutil	period
1	6	24	1
2	6	14	1
3	2	6	1

b

Tid	iutil	rutil	period
1	1	29	1
2	3	17	1
3	1	7	1
4	6	21	1
5	2	9	1

e

Tid	iutil	ruti
1	10	20
2	8	12
5	4	4

c

Tid	iutil	rutil	period
1	3	25	1
2	3	17	1
4	6	21	2
5	3	7	1

1-MFPHM(Modified High-utility Itemset) Utility-lists Construction

Each element of item periodic utility-lists contains four fields such as Field tid represents the transaction id containing itemset X, iutil field represents the internal utility of itemset X, Field rutil represents the external utility of itemset X, Field period represents the period of itemset X. From each 1-Itemset utility-lists structure, we can easily found the support of each itemset by counting number of elements in each utility-lists. Once the construction of 1-itemset utility-lists is completed, construction of 2-itemsets can be formed so that periodicity and frequency of itemset can be known.

2-Itemset MFPHM-Utility-lists construction:

Without scanning database again, intersection of the utility-lists of {a} and {b} can be used to construct utility-lists of 2-itemset. Common transactions are identified by comparing the tids in the two utility-lists. Suppose the lengths of the two utility-lists are k and l respectively, then for identifying common transactions, we need do (k + l) comparisons are enough at most, because all tids are orderly placed in a utility-list. A 2-way comparison is actually takes place in the identification process.

For the implementation of our algorithm, we have used Modified High-Utility-lists(MHUL) structure to store the entire information about each itemset in a compact manner.

IV PROBLEM STATEMENT

Itemset	Utility	period	Support
d b	30	3	2
d b c	36	3	2
d b e c	40	3	2
d b e	34	3	2
b e	31	3	3
b e c	37	3	3
a e c	31	3	3

A *Periodic high-utility itemset*

Let $\mathbf{I = \{i_1, i_2, \cdots, i_n\}}$ be a set of items. **X** is an itemset, consists of set of items where $\mathbf{X \subseteq I}$.
A transaction **t** = (**tid**,**X**) is a tuple, where **tid** represents a transaction-id (or a timestamp) and **X** is a itemset. A transactional database T over I is a set of transactions, $\mathbf{T = \{t_1, \cdots, t_m\}}$, $\mathbf{m = |T|}$, where |T| is the size of T in total number of transactions. If $\mathbf{X \subseteq Y}$, it is said that t contains X or X occurs in t and such transaction-id is denoted as t_j^X, $\mathbf{j \epsilon [1,m]}$.
Let $\mathbf{T^X = \{t_k^X, \cdots t_l^X\} \subseteq T}$, where $\mathbf{k \leq l}$ and $\mathbf{k, l \epsilon [1,m]}$ can be described as the ordered set of transactions in which pattern X has appeared. Let $\mathbf{t_j^X}$ and $\mathbf{t_{j+1}^X}$, where $\mathbf{j \epsilon [k, (l-1)]}$ be two consecutive transactions in $\mathbf{T^X}$. A **period** of X can be defined as the number of transactions or time difference between $\mathbf{t_{j+1}^X}$ and $\mathbf{t_j^X}$, say **pX** . That is, $\mathbf{pX = t_{j+1}^X - t_j^X}$. Let the set of periods for pattern X are PX = $\{pX_1, pX_2, \cdots, pX_r\}$. The **periodicity** of X, denoted as $Per(X) = \max(pX_1, pX_2, \cdots, pX_r)$. The **utility** of X, denoted as U(X) .
A periodic-frequent high utlity itemset is an itemset X which satisfies two user defined constraints minuutil, maxper. Utility of X, $U(X) \geq$ minutil and $Per(X) \leq$ maxprd, where minutil, minsup and maxprd are user-specified minimum utility, minimum support and maximum periodicity constraints.
Both periodicity and utility of a pattern can be described as a percentage of |T|.
From the transaction table, the number of transactions is 5, the periodicity of itemset f denoted as maxper ((1-1),(5-1)) i.e., 4. Utility of f is 1.The periodicities of itemset a,b,c,d,e,f,g are 2,2,1,2,2,4,3. We are initially considered the utility of an itemset to verify whether it is high-utility itemset or not. If it is high-utility itemset, then we check its periodicity. If it's periodicity is less than or equal to the maxper, then those itemsets are periodic high-utility itemsets.
Our approach mainly consists of two phases which follow apriori based technique. Initially, construction of Modified utility-lists for 1-itemsets is done, which maintains utility and its periodicity information. The detailed construction of utility list is mentioned in the [3]

In the first phase, we apply periodic HUI miner algorithm to find all periodic high utility itemsets., Support of each periodic high utility itemset is calculated based on the counting of a number of elements appeared in each utility-lists in the second phase.

Algorithm 1 : FPHUI miner

Input: Transactional database TB, User specified period-maxper, User specified support-minsup, User specified utility-minutil, Set of ULs with all 1-extension with prefix P
Output: set of frequent HUI with regular occurrences.

1. **For** each Utility-lists X in ULs do
2. Call FPHUI miner (TB, X, ULs, maxper, minsup, minutil);
3. **End for**
4. **For** each itemset Y from FPHUI miner do
5. Y.count = 0;
6. **For** each element e in Y do
7. Y.count=Y.count+1;
8. **End for**
9. **If** Y.count>=minsup then
10. Output Y;
11. **End if**
12. **End for**
13. Return Y;

Algorithm 2 :Construction of FPHUI Miner

Input: Transactional database, Set of ULs with all 1-extension with prefix P , User specified period-maxper, Utility-lists of itemset P initially empty.
Output: Periodic High-utility Itemsets with extension of P

1. For each Utility-lists X in ULs do
2. If X.sumiutil >= minutil, then
3. If X.period <= maxper, then
4. output the extension associated with X.
5. If X.sumiutil + X.sumrutil >= minutil, then
6. exULs= null
7. For each utility-lists Y after X in ULs, do
8. exULs = exULs + construct(P.UL, X,Y)
9. end
10. FPHUIMiner(exULs, X, minutil,maxper)
11. end
12.end

Algorithm 3 : Construct Algorithm

Input: P.UL : Utility-lists of itemset P
Px.UL : Utility-lists of itemset Px
Py.UL : Utility-lists of itemset Py
Output: Pxy: Utility-lists of itemset Pxy

1. Pxy.UL = NULL; $T_{initial}$=1;

2. For each element Ex ∈ Px.UL **do**

3. if ∃Ey∈Py.UL **and** Ex.tid==Ey.tid **then**

4. **if** P.UL is not empty **then**
5. search such element E∈P.UL that E.tid==Ex.tid;
6. Exy=<Ex.tid, Ex.iutil+Ey.iutil - E.iutil, Ey.rutil, Ex.tid - $T_{initial}$ >;
7. $T_{initial}$ = Ex.tid;
8. **else**
9. Exy=<Ex.tid, Ex.iutil+Ey.iutil, Ey. Ex.tid rutil, Ex.tid - $T_{initial}$ >;
10. $T_{initial}$ = Ex.tid;
11. **end**
12. append Exy to Pxy.UL;
13. **end**
14. **end**
15. **return** Pxy.UL;

B ***Discussion about proposed algorithm***

In the proposed model, we have proposed efficient algorithms for mining frequent high-utility itemsets that appears at regular occurrences. This algorithm adopts periodic high-utility itemsets with two phases, it first arranges all itemsets according to their TWUs. In the second phase, it checks itemsets that satisfied with their utility and periodicity. The construction of itemsets Each itemset periodicity is obtained from Modified utility-list of an itemset. It collects all periods of an itemset from the list to further calculates maximum periodicity.

The main limitation is of existing algorithm does not consider the frequency of periodic itemset that generates high profit. This can be overcome by FPHUI miner. This algorithm is not only considers periodicity but also frequency of high utility itemsets. The utility lists are modified such as to maintain every information of itemsets. It is easy to retrieve useful information from utilitylists for mining. FPHUI.

miner algorithm mines interested frequent periodic high utility itemsets in two phases. This algorithm may not be applicable to large databases since in very large databases, more number of itemsets are present. Each itemset may contain more number of elements. Thus, each itemset's utility-list need to maintain more number of periods. For large databases, comparing all periods of itemset to find periodicity of it, will involve more time. Comparison of all periods of itemsets consumes more time.

The following performance issues are identified: FPHUI miner algorithm suffers from the following performance issues:

1.After the initial scan of a database, uninteresting items are pruned only based on the TWU of each itemset. It does not consider periods of an itemsets at initial stage. This approach leads to increase in memory, search and update requirements of periodicity list.

For Example, in Table 4, the item 'f' satisfied with the TWU value but it does not satisfies with maximum periodicity at initial scan of data base. But this leads to consider the item 'f' till we determines its periodicity. Now, it's second period is going to be 4, this item is identified as aperiodic by the time when they were first identified in the database as its periodicity is no less than the given maxper.. Thus, there is no need to consider this item in the construction of 2-itemset utility-lists with their periodicities. However, MFPHM miner algorithm consider this item. This results in performance issues.

2. FPHUI algorithm determines the periodic interestingness of every item in Periodic itemset utility-lists by complete search on it utility-lists. When the database size is large, the number of elements of utility-lists will be more. In such cases, task of determining periodicity of an itemset need complete search on every itemset utility-lists which is computationally expensive process.

For example, if we consider item f utility-lists, all the periods of f are 1(first transaction id), and 4 (5-1). The FPHUI algorithm measures periodicity of f as maximum of (1,4). The periodicity of f is 4. Then it compares with the maxper i.e. 3. It searches for the entire list to determine its maximum periodicity. For measuring periodicity of itemset 'af', we must first construct the 'af' periodic utility-list to find its maximum periodicity. But if the item f is identified as aperiodic itemset at initial stage only, then we need not consider the construction of af utility-lists with periodicities. In FPHUI algorithm determines 'af' itemset periodicities as 1, 4 and then it searches maximum value among this list to compare with the maxper. As 'af' itemset maximum period is 4 which is greater than the maxper, it does not consider the itemset af as periodic high-utility itemset. However, to determine af as an periodic high-utility itemset or not, such complete search is not necessarily to be done. The reason is that the first period of 'af' is not to satisfied with the maxper. The item 'f' is not periodic high-utility itemset as it does not satisfy with the maxper, thus further search need not performed.

Finding periodic high-utility itemsets using the above algorithm is time consuming method as it depends upon the periodicity measure. For measuring the periodicity of an itemset, we need to keep all periodicities in a list called ip(Item-Period)–list. In this list, we maintain all periods of an itemset. The number of entries in ip-list depends on n, where n indicates the number of transactions in which itemset has appeared. For very large databases, as the n value will be very large number, it is computationally expensive to find periodic high-utility itemsets.

For example, let 'ab' be a high-utility itemset in very large database, which appears in transactions 6,9,12,15,28, and so on. The FPHUI-miner algorithm searches complete list of ip-list to find its periodicity. It determines whether the itemset is periodic or aperiodic by checking its periodicity is less than maxper. This approach leads to computationally expensive process to determine periodicity of high-utility itemsets in very large transactional databases. For very large databases, it takes large amount of time to find itemset's periodicity as itemset's periodicities list is very large.

The above mentioned issues motivated towards proposing of novel algorithm called MFPHM-miner based on concept called local periodicity to find periodic high-utility itemsets efficiently. The main idea behind this approach is to apply greedy search on ip-list to find itemset's LMax. We can define LMax as suboptimal solution and periodicity of an itemset as optimal solution for any itemset. If an itemset's LMax is not satisfied with maxper, then we can conclude that itemset is not periodic. If an itemset is found to be aperiodic by comparing its LMax, then we need not have further search in the ip-list. This approach considerably reduces the computational cost in mining periodic high-utility itemsets.

Definition 6: LMax periodicity of an high-utility itemset: Let P(X) be the list of periods of itemset in the database, denoted as P(X)= $\{p_1,p_2,p_3,\ldots p_n\}$, where n is the support of X+1. Let P'(X) can be denoted as a set of $\{p_1,p_2,p_3,\ldots p_k\}$,$1\leq k \leq n$. P'(X) be the set of periods of X such that P'(X) $\subseteq$ P(X). The LMaxof an itemset is given as the maximum period in P'(X).

For example, the set of all periods for 'bd', i.e., p(bd)= (0,1,3). Let one of the ordered set of periods from p(bd) is (1,3). Then the LMax of bd is max(1,3), i.e., 3.

For then high-utility itemset X, if LMax of X is greater than maxper, then X is aperiodic high-utility itemset.

Property 1: For an itemset X, as the P'(X) $\subseteq$ P(X), then the per(X) $\geq$ LMax of X.

If LMax of X > maxper, then per(X) > maxper as per property 1, then X is called as aperiodic high-utility itemset.

For example, the itemset 'cg' occurs at transactions 4 and 5. The periods of 'cg' are (4,1,0). The first period of 'cg' is 4, i.e., $p_1^{'cf,}$. The LMax of 'cg', at this point, max(4,1) is 4. As the LMaxof cg is greater than maxper, the further search in the periods list does not necessary. And we can conclude that the 'af' aperiodic itemset and uninteresting itemset.

Property 2: For any periodic itemset, the LMax of itemset is its periodicity.

By using property 1, if P'(X) $\subseteq$ P(X) for periodic itemset, per(X) = local periodicity(X). It in turns to be proved that property 2.

In this MFPHM miner algorithm, LMax of an itemset represents suboptimal solution where as periodicity as optimal solution. LMax determines itemsets periodic or not as if it satisfied with user given maxper. It reduces search time further to determine the periodicity. By using Property 2, it can easily determined that the Lmax prunes aperiodic itemsets at earlier stage only. This approach mainly facilitates the early pruning technique to reduce computational cost.

Algorithm 4: MFPHM **miner**

Input: Transactional database TB, User specified period threshold -maxper, User specified support threshold -minsup, User specified utility threshold -minutil, Set of ULs with all 1-extension with prefix P
Output: set of frequent HUI with regular occurrences
1. For each Utility-lists X in ULs do
2. If X.sumiutil >= minutil, then
// Calculate LMax of X.
3.Call Lmax(X);
4. If X.LMax > maxper, then
5. X is not periodic itemset.
6. Else
7. If X.period$\leq$ maxper then
8. output the extension associated with X.
9. If X.sumiutil + X.sumrutil >= minutil, then
10. exULs= null
11. For each utility-lists Y after X in ULs, do
12. exULs = exULs + construct(P.UL, X,Y)
13. end
14. FPHUIMiner(exULs, X, minutil,maxper)
15. end
16. end
17. end

Algorithm 5: Finding LMax of itemset (X)
Input : Px: Utility-lists of itemset Px
Output: LMax of X

1. Ordered set of periods of Px.UL is given as S
2. Maximum value among S as M(S)
3. Set M(S) as Px-LMax;
4. Return Px-LMax.

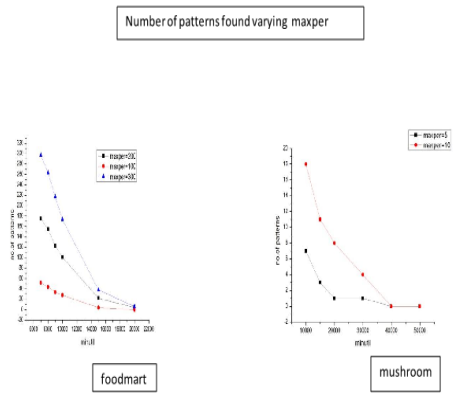
V EXPERIMENTAL RESULTS

In this section, we show the performance of our proposed algorithm FPHUI miner. Our proposed algorithm adopt basic framework from HUI miner for Modified Utility-lists(MUL). We assessed our algorithms by performing experiments on computer with a 2.10 GHz Intel Core i3 CPU with 4 gigabyte memory, and running on Windows 7. Implementation of our proposed algorithms are done in Java. The performance of our proposed algorithm can be evaluated with three data sets. We have considered absolute values for maximum periods. For Retail dataset, more number of candidates are generated when minutil is set to 8k. The count of candidates decreases with increase in minutil.

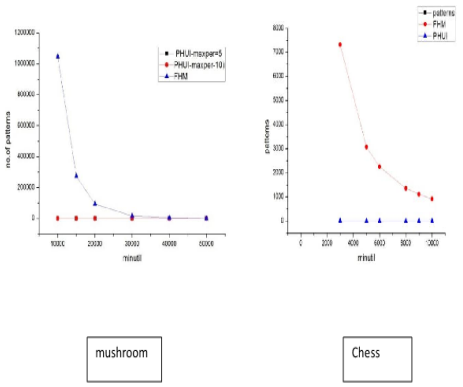
A Effect of maxper on algorithm

The effect of maxper is analyzed in this section. Periodic HUIs depends on the high maxper and low minutil. For foodmart dataset, candidate itemsets are generated more for all variations of minutil i.e., for 6k,10k,15k,20k. As a second observation, we have noticed that the number of FPHUIs are much less than compared to the number of HUIs. For retail dataset, the number of FPHUI are 8,7,6 when minimum utility is set to 3k,5k,6k. Number of high-utility itemsets for these values of minimum utilities are 7314,3071,2254. The effectiveness in the periodic high-utility itemset mining is depending on the maxper. Since the maxper controls the maximum time difference between two consecutive appearances in the transaction databases. Similarly number of patterns are also reduced in other datasets based on maxper . Now, we consider the effect FPHUI with MFPHM. The number of periodic high-utility itemsets are reduced considerably for chess dataset using MFPHM miner algorithm. As the MFPHM algorithm has generated less number of periodic high-utility itemsets, the run time for chess dataset has reduced. For mushroom dataset, periodic HUI are very high compare to frequent periodic HUIs.
MFPHM miner has generated less number of itemsets to itemsets generated with FPHUI miner. For chess dataset, the number of itemsets are reduced to 20 times to number of itemsets with PHUI miner for minutil 2k,3k,..20k. For mushroom dataset, number of itemsets are considerable reduced to 10 times compare to FPHUI miner.

Figure 1 Performance of FPHUI algorithm.



Figures 2 Comparison of algorithms



VI. CONCLUSION

In this paper, we have proposed algorithms to mine periodic high utility itemsets where we integrate frequency constraint into high-utility itemsets in transaction database. New Modified Utility-lists-MUL structure is proposed which is used to maintain utility and periodicity information about each utility-lists mine frequent high-utility itemsets with regular occurrences. To reduce computational cost, we have further developed another algorithm called MFPHM miner. In this approach we adopted greedy method to find LMaxand periodicity of itemset to lessen computation cost. Experiments were conducted on both real and synthetic datasets and results shows that proposed approach is better to mine frequent high-utility itemsets with regular occurrences in terms of efficiency. In the future, we would like extend our work to different kinds of databases like incremental databases, and data streams.

REFERENCES

1. Agrawal, R., Imielinski, T. & Swami, A., Mining Association Rules Between Sets of Items in Large Databases, in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, pp. 207-216, 1993.
2. Agrawal, R. & Srikant, R., Fast Algorithms for Mining Association Rules in Large Databases, in Proceedings of the 1994 ACM SIGMOD International conference on Management of data, Minneapolis, MN, USA, pp. 487-499, 1994.
3. Liu, M. & Qu, J., Mining High-utility Itemsets without Candidate Generation, in Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, pp. 55-64, 2012.
4. Lin, C-W., Hong, T.-P., Lan, G.-C., Wong, J.-W. & Lin, W.-Y., Efficient Updating of Discovered High-Utility Itemsets for Transaction Deletion in Dynamic Databases, Advanced Engineering Informatics, 29(1), pp. 16-27, 2015.
5. Liu, Y., Liao, W.K. Choudary,A., A Two Phase Algorithm for Fast Discovery of High utility itemsets, in Advances in Knowledge Disccovery and Data mining, pp. 689-695, 2005.
6. V. S. Tseng, B.-E. Shie, C.-W. Wu and P. S. Yu, Up growth: An efficient algorithm for high-utility itemset mining, In Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 253–262, 2010.
7. V. S. Tseng, B.-E. Shie C.-W. Wu and P. S. Yu, Efficient algorithms for mining high-utility itemsets from transactional databases, IEEE Transactions on Data and Knowledge Engineering, volume 25, pp. 1772-1786, 2013.
8. Fournier-Viger, P., Wu, C.-W., Tseng, V. S. : Novel Concise Representations of High-utility Itemsets using Generator Patterns. Proc. of 10th International Conference on Advanced Data Mining and Applications, pp. 30-43, 2014.
9. Tseng,V.,Wu,C.,Fournier-Viger,P.,Yu,P.: Efficient algorithms for mining the concise and lossless representation of closed+ high-utility itemsets. IEEE Transactions on Knowledge and Data Engineering., 726-739,2015.
10. C.H. Cai, A.W.C. Fu, C.H. Cheng and W.W. Kwong, Mining Association Rules with Weighted Items, Proc. International Database Eng. and Applications Symposium. (IDEAS '98), IEEE, pp. 68-77, 1998.
11. U. Yun, An Efficient Mining of Weighted Frequent Patterns with Length Decreasing Support Constraints, Knowledge-Based Systems, ELSEVIER, volume 21, pp. 741-752. 2008.
12. U. Yun and J.J. Leggett, WFIM: Weighted Frequent Itemset Mining with a Weight Range and a Minimum Weight, Proc. SIAM Int'l Conf. Data Mining (SDM '05), pp. 636-640, 2005.
13. U. Yun and J.J. Leggett, WIP: Mining Weighted Interesting Patterns with a Strong Weight and/or Support Affinity, Proc. SIAM Int'l Conf. Data Mining (SDM '06), SIAM, , pp 623-627, April 2006.
14. Shie,B.-E.,Yu,P.S.,Tseng,V.S.:Efficient algorithms for mining maximal high-utility itemsets from datastreams with different models. Expert Systems with Applications 39(17),pp. 12947-12960 2012.
15. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S. & Lee, Y.K., Discovering Periodic-Frequent Patterns in Transactional Databases, in Proceedings of the 13th Pacific-Asia Knowledge Discovery and Data Mining conference (PAKDD 2009), Bangkok, Thailand, pp. 242-253, 2009.
16. Kiran, R.U. & Reddy, P.K., Towards Efficient Mining of Periodic-Frequent Patterns in Transactional Databases, in Proceedings of the1st International Conference on Database and Expert Systems Applications, Bilbao, Spain, pp. 194-208, 2010.
17. Wu, C.W., Shie, B-E., Tseng, V.S. & Yu, P.S., Mining Top-k High-utility Itemsets, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 78-86, 2012.
18. Tseng, V.S., Wu, C.-W., Fournier-Viger, P. & Yu, P.S., Efficient Algorithms for Mining Top-K High-utility Itemsets, IEEE Transactions on Knowledge and Data Engineering, **28**(1), pp. 54-67, 2015.
19. Li, Z., Ding, B., Han, J., Kays, R. & Nye, P., Mining Periodic Behaviors For Moving Objects, in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington DC, DC, USA, pp. 1099-1108, 2010.
20. Surana, A., Kiran, R.U. & Reddy, P.K., An Efficient Approach to Mine Periodic Frequent Patterns in Transactional Databases, in Proceedings of International Workshops on New Frontiers in Applied Data Mining, Shenzhen, China, pp. 254-266, 2012.
21. Yun U, Kim D, Ryang H, Lee G, Lee K Mining recent high average utility patterns based on sliding window from stream data. Journal of Intelligent Fuzzy Systems.pp.3605–3617, 2016.
22. Fournier-Viger P., Lin J.CW., Duong QH., Dam TL. PHM: Mining Periodic High-Utility Itemsets. In: Perner P. (eds) Advances in Data Mining. Applications and Theoretical Aspects. ICDM 2016. Lecture Notes in Computer Science, volume 9728. Springer, Cham, 2016.

23. Srikumar K. H-Miner:Efficiently mining high utility itemsets. Expert systems with applications, 2017.
24. Song, W., Liu, Y., Li, J.: BAHUI: Fast and memory efficient mining of high utility
itemsets based on bitmap. Intern. Journal of Data Warehousing and Mining. 10-15 (2014)
25. Komate Amphawan, Athatsit Surarerks, Philippe Lenca, Mining periodic-frequent itemsets with approximate periodicity using interval transaction-ids list tree, WKDD-10 3RD International conference 2010
26. Komate Amphawan, , Philippe LencaMining top-k frequent-regular closed patterns, Expert Systems with Applications, pp7882–7894, 2015
27. Ashis Kumar Chanda,, Chowdhury Farhan Ahmed, Md. Samiullah, Carson K. Leung, A new framework for mining weighted periodic patterns in time series databases, Expert Systems with Applications, pp 207-224, 2017.

AUTHORS FROFILE

P. Lalitha Kumari, Research Scholar, Department of Computer Science and Engineering, National Institute of Technology, Warangal

Prof. S.G.Sanjeevi, Professor, Department of Computer Science and Engineering, National Institute of Technology,Warangal

Dr.T.V.Madhusudhana rao, Professor, Department of Computer Science and Engineering, Sri Sivani College of Engineering