# Utilizing Index-Based Periodic High Utility Mining to Study Frequent Itemsets

*by* Roy Setiawan

**Submission date:** 16-Jul-2021 08:19AM (UTC+0700)
**Submission ID:** 1620150170
**File name:** UtilizingIndex-BasedPeriodicHi.pdf (3.14M)
**Word count:** 5436
**Character count:** 28337

Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-021-05936-6

RESEARCH ARTICLE - SPECIAL ISSUE - FRONTIERS IN PARALLEL PROGRAMMING MODELS FOR FOG AND EDGE COMPUTING INFRASTRUCTURES

# Utilizing Index-Based Periodic High Utility Mining to Study Frequent Itemsets

**Roy Setiawan[1] · Dac-Nhuong Le[2] · Regin Rajan[3] · Thirukumaran Subramani[4] · Dilip Kumar Sharma[5] · Vidya Sagar Ponnam[6] · Kailash Kumar[7] · Syed Musthafa Akbar Batcha[8] · Pankaj Dadheech[9] · Sudhakar Sengan[10]**

Received: 8 May 2021 / Accepted: 28 June 2021


**Abstract**

The potent employability in different applications has garnered more significance for Periodic High-Utility Itemset Mining (PHUIM). It is to be noted that the conventional utility mining algorithms focus on an itemset's utility value rather than that of its periodicity in the transaction. A MEAN periodicity measure is added to the minimum (MIN) and maximum (MAX) periodicity to incorporate the periodicity feature into PHUIM in this proposed work. The MEAN-periodicity measure brings a new dimension to the periodicity factor and is arrived at by dividing itemset's period value by the total number of transactions in that dataset. Further, an algorithm to mine Index-Based Periodic High Utility Itemset Mining (IBPHUIM) from the database using an indexing approach is also proposed in this paper. The proposed IBPHUIM algorithm employs a projection-based technique and indexing procedure to increase memory and execution speed efficiency. The proposed model avoids redundant database scans by generating sub-databases using an indexing data structure. The proposed IBPHUIM model has experimented with test datasets, and the results drawn show that the proposed IBPHUIM model performs considerably better.

**Keywords** IBPHUIM · Periodic pattern · Frequent periodic pattern

✉ Sudhakar Sengan
sudhasengan@gmail.com

Roy Setiawan
roy@petra.ac.id

Dac-Nhuong Le
ledacnhuong@duytan.edu.vn

Regin Rajan
regin12006@yahoo.co.in

Thirukumaran Subramani
thirukumaran75@kluniversity.in

Dilip Kumar Sharma
dilipsharmajiet@gmail.com

Vidya Sagar Ponnam
pvsagar20@gmail.com

Kailash Kumar
k.kumar@seu.edu.sa

Syed Musthafa Akbar Batcha
syedmusthafait@gmail.com

Pankaj Dadheech
pankajdadheech777@gmail.com

[1] Department Management, Universitas Kristen Petra, Jawa Timur, Indonesia

[2] School of Computer Science/Institute of Research and Development, Duy Tan University, Danang 550000, Vietnam

[3] Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, Tamil Nadu 635109, India

[4] Department of Computer Science and Engineering, KL University, Vijayawada, Andhra Pradesh 522502, India

[5] Jaypee University of Engineering and Technology, Guna, Madhya Pradesh 473226, India

[6] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh 522502, India

[7] College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Kingdom of Saudi Arabia

[8] Department of Information Technology, M. Kumarasamy College of Engineering, Karur, Tamil Nadu 639113, India

[9] Department of Computer Science and Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan (SKIT), Jaipur, Rajasthan 302017, India

[10] Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu 627152, India

Published online: 14 July 2021

## 1 Introduction

The projection-based approaches evolve from the database management discipline, particularly from Query Language Processing (QLP) [1]. In data mining, the process of knowledge extraction utilizing queries that satisfy a particular condition has long experimented with within various works. One of the earlier works on data mining that employed QLP was Han et al. *2001* [2]. They employed Sequential Pattern Mining (SPM); the SPM was an extended form of association rule mining with particular emphasis over time. Han et al. projected mined patterns by considering every sequence, which is a potential candidate as a query condition, and then the mined patterns are nothing but the tuples in the database that include those candidate sequences. An index-based projection approach to mine IBPHUIM is proposed in this work. In addition to the new indexing structure proposed, a more efficient pruning methodology to reduce false candidates is also proposed. The proposed pruning models help achieve lesser memory consumption and faster mining of the patterns, which was proven through the experimental results.

The main drawback in mining utility itemsets is that it does not hold to downward-closure property, making the association-rule mining less complex. Many works were carried out to include anti-monotonic property in utility mining to solve this complex nature. One such work was proposed by Liu et al. 2006 [3]; in order to extract High Utility Itemsets (HUI) from the database, they proposed an II-phase utility mining algorithm that incorporated downward-closure property in the form of Transaction-Weighted Utility (TWU). The TWU-based model uses the cumulative value of the entire item's utility value in the transaction as the upper limit of an item set in that transaction maintains the downward-closure attribute. On the other hand, the actual utility value of an item set in utility mining is raised together with the rise in the number of items within it. So, employing the same utility threshold to ascertain an item set with variable length as a HUI or not is a viable solution. This complexity was addressed by Hong et al. 2011 [4]; in their approach, they proposed a new measure called the mean-utility measure. This measure helps in incorporating the length factor of an itemset in addition to its other utilities. In this way, it adds a new dimension to the itemset's utility. Then, the actual average utility of an item set can now be represented as the value arrived by dividing the summed-up utility values of all the items in the transaction by sample itemsets. Thus, the search space to mine high utility itemset is reduced significantly compared to the actual utility value. The model represents characteristics of big data to deal with various attributes of generated massive data sets. This measures all specialized areas like hardware, processing, database technology, software, and visualization through item dataset [5].

Further, Hong et al. proposed an Average-Utility Upper Bound (AUUB)-Two-Phase Average-Utility Mining (TPUM) [6] algorithm to summarize the maximal utility among utility values of items in each transaction that shows it. They employed this average upper bound to overvalue all probable high average-utility itemset's in a database. The model proposed by Hong et al. [4] was split into two phases; during the initial phase, the upper limit value is allotted to all the high AUUB itemset at each level of iteration; it is to be noted that the itemset's AUUB values must satisfy the MIN_ average-utility threshold. During the latter phase, to ascertain the itemset's ACTUAL_ average-utility values in the database, an additional database scan is carried out. Lastly, those items with their actual average-utility values larger than or equal to the MIN_Threshold are filtered as the average-HUI. Because the TPAU algorithm has to produce an excess number of candidates to measure their AUUB, it is understandable that it utilizes more time and memory [7], [8], [9].

## 2 Problem Definition

Let there be a Transactional Database ($TRAN_D$) containing *'n'* transactions $TRAN_D = \{TRAN_1, TRAN_2, ..., TRAN_N\}$, and let there be an ITEMSET (IS) $= \{IS_1, IS_2, ...., IS_n\}$. Subsequently, $U_{val}$ ($IS_n$, $TRAN_{ID}$) corresponds to the utility value of the item as $ITEM_n$ in the transaction $TRAN_{ID}$.

A set of distinct items $D_{IS} = \{D_{IS1}, D_{IS2}, D_{IS3}, ..., D_{ISn}\}$ and $IS = \{ITEM_1, ITEM_2, ...., ITEM_n\}$ provide the itemset $IS \subseteq D_{IS}$. Let there be a transactional database containing n transactions $TRAN_D = \{TRAN_1, TRAN_2, ...., TRAN_N\}$. For every transaction in the transaction database, a unique identifier ID represented as $TRAN_{id}$ is assigned to identify the transactions [10]. Each item $DITEM_i$ in a transaction $TRAN_n$ has an internal utility represented as $INU_{val}$ ($TRAN_{ID}$, $DITEM_i$); additionally, the item $DITm_i$ has an external utility value $EXU_{val}$ ($DITEM_i$) related with it. The sample items {I: Air Ionizer, II: Blu-Ray Player, III: Dehumidifier, IV: Futon Dryer, V: Cold-Pressed Juicer, VI: Garbage Disposal Unit, VII: Exhaust Hood}. Assume, for example, the transaction dataset shown in Table 1.

A. Definition *I* (Utility of an Item in a Transaction)

For the specified transaction database $TRAN_D$, the item $ITEM_n$ utility is indicated as $U_{val}(ITEM_n, TRAN_D)$, which is characterized as:

$$\mathrm{UIS_{val}}(\mathrm{IS}_n, \mathrm{TRAN}_{id}) = \mathrm{NUM}(\mathrm{IS}_n, \mathrm{TRAN}_{id}) * \mathrm{WORD}(\mathrm{IS}_n) \quad (1)$$

referring to the Equ (1), the NUM ($\mathrm{ITEM_n}$, $\mathrm{TRAN_{id}}$) denotes the count of itemset $\mathrm{ITEM_n}$ in $\mathrm{TRAN_{id}}$, and WORD ($\mathrm{ITEM_n}$) indicates the worth of an itemset $\mathrm{ITEM_n}$. Considering Tables 1 and 2, the value of the utility of ITEM (II) in the transaction with ($\mathrm{TRAN_{id}}=2$) is calculated as:

$$\begin{aligned}\mathrm{U_{Val}}(\mathrm{II},\ \mathrm{TRAN_2}) &= \mathrm{NUM}\ (\mathrm{II},\ \mathrm{TRAN_2})\ *\ \mathrm{WORD}\ (\mathrm{II})\\ &= 2 * 20\\ &= 40\end{aligned}$$

B. Definition II (Utility of an Itemset in a Transaction)

For an itemset $\mathrm{ITEM_n}$ in a transaction TRAN $D_n$, the utility is represented as $\mathrm{ITEMU_{Val}}$ (ITEM $\mathrm{T_n}$, TRAN $D_n$), which can be described as, Equ (2)

$$\mathrm{UIS_{val}}(\mathrm{IS}_n, \mathrm{TRAN}_{id}) = \sum i_n \in \mathrm{ITEM}_{T_n} \wedge \mathrm{ITEM_{id}} \subseteq \mathrm{TRAN_{id}}\mathrm{UIS_{val}}(i_n, \mathrm{TRAN_{id}}) \quad (2)$$

From the example, the utility of itemset (II, V) is measured as

$$\begin{aligned}\mathrm{ITEMU_{Val}}[(\mathrm{II},\ \mathrm{V}),\ \mathrm{TRAN_2}] &= \mathrm{U_{Val}}(\mathrm{II},\ \mathrm{TRAN_2}) + \mathrm{U_{Val}}(\mathrm{V},\ \mathrm{TRAN_2})\\ &= \mathrm{NUM}\ (\mathrm{II},\ \mathrm{TRAN_2})\ *\ \mathrm{WORD}\ (\mathrm{II})\ + \mathrm{NUM}\ (\mathrm{V},\ \mathrm{TRAN_2})\ *\ \mathrm{WORD}\ (\mathrm{V})\\ &= (2\ *\ 20)\ +\ (4\ *\ 40)\\ &= 40\ +\ 160\\ &= 200.\end{aligned}$$

C. Definition III (High Utility Itemset, HUITEM)

An itemset $\mathrm{ITEM_n}$ can be declared as HUI for the given $\mathrm{TRAN_D}$ if the value of utility in $\mathrm{TRAN_D}$ satisfies the user-specified threshold $\mathrm{MU_{Val}}$, and it is represented as Equ (3),

$$\mathrm{HUIM}(\mathrm{ItemSet}_n) \leftarrow \{IS_n | \sum IS_n \subseteq \mathrm{TRAN_{id}} \wedge \mathrm{TRAN_{id}} \in \mathrm{TRAN}_D \mathrm{UIS_{val}}(\mathrm{IS}_n, \mathrm{TRAN}_D) \geq \mathrm{Mining_{val}}\} \quad (3)$$

For example, let the MIN utility threshold MIN. $\mathrm{U_{Val}} = 1000$.

To ensure whether an IS (II, V) is a HUI, the following condition is checked

$$\begin{aligned}\mathrm{HUI}(II, V) = If\{\mathrm{Item_{val}}(II, V, \mathrm{TRAN}_D) \geq 1000\\ \mathrm{HUI_{val}}(II, V, \mathrm{TRAN}_D) = \mathrm{ISU_{val}}(II, V, \mathrm{TRAN_2}) + \mathrm{ISU_{val}}(II, V, \mathrm{TRAN_3})\\ + \mathrm{ISU_{val}}(II, V, \mathrm{TRAN_4}) + \mathrm{ISU_{val}}(II, V, TRAN_5)\\ + ISU_{val}(II, V, TRAN_6)\\ = 200\ +\ 240\ +\ 160\ +\ 200\ +\ 200\\ = 1000 \text{ which is greater than MIN.}\mathrm{HUI_{val}}\end{aligned}$$

Hence, the itemset Blu-ray player, cold-pressed juicer is a HUIS.

D. Definition IV (Transaction-Weighted Utility of an Itemset)

In the given transaction database, the value of the Transaction-Weighted Utility of an IS($\mathrm{TWU_{Val}}$). ITEMSET [11] [12] demonstrates the collective utility of all the partaking items in the transaction $\mathrm{TRAN_{id}}$ where IS is an element, which is given as, Equ (4)

$$\mathrm{TWU_{val}}(IS) = \sum_{IS} \subseteq \mathrm{TRAN_{id}} \wedge \mathrm{TRAN}_D * \mathrm{TWU_{val}}(\mathrm{TRAN_{id}}) \quad (4)$$

$$\mathrm{TWU_{Val}}(\mathrm{II},\ \mathrm{V}) = 260 + 280 + 340 + 500 = 1320$$

**Table 1** Transaction database

| $\mathrm{TRAN_{ID}}$ | Transactions | |
|---|---|---|
| 1 | {I, II, III, IV} | {2,2,2,1} |
| 2 | { II, V, VI} | {2,4,3} |
| 3 | { I, II, V, VI} | {2,4,4,2} |
| 4 | { II, V, III, VII} | {2,3,3,3} |
| 5 | { I, II, V, III, IV} | {2,2,4,4,2} |
| 6 | { I, II, V, VII} | {5,4,3,5} |

**Table 2** External utility table

| Itemset | Profit |
|---|---|
| I | 35 |
| II | 25 |
| III | 45 |
| IV | 15 |
| V | 35 |
| VI | 25 |
| VII | 25 |

E. Definition V (Periods of an Itemset)

Let $TRAN_D = \{TRAN_1, TRAN_2, \ldots, TRAN_N\}$ be a transactional database containing '*n*' transactions [13] [14] [15], and let IS = {I, II, …., N} be an itemset. The TRAN set that contains ITEMSET is denoted as Test (IS) = $\{TRAN_{Test1}, TRAN_{TST2}, \ldots, TRAN_{TSTN}\}$, where $1 \le Test_1 < Test_2 < \ldots Test_N \le k$. Two transactions $TRAN_A \supset IS$ and $TRAN_B \supset IS$ are said to be consecutive concerning IS if there does not exist a transaction $TRAN_C \in$ Test (TestSet) such that $a < c < b$. The period between the two transactions $TRAN_A$ and $TRAN_B$ in Test(IS) is represented as LSPPERIOD ($TRAN_A$, $TRAN_B$) = (a−b), which denotes the total number of transactions between $TRAN_A$ and $TRAN_B$. The list of periods for an IS can be observed as.

PERIOD(ItemSet) = $\{Test_1 - Test_0; Test_2 - Test_1; Test_3 - Test_2; \ldots .Test_n - Test_{n-1}; Test_{n+1} - Test_n\}$ in which $tst_0$ and $tst_{n+1}$ are constants initialized with $Test_0 = 0$ and $Test_{N+1} = k$. Hence, LSPPERIOD(IS) = $\cup_{1 \le z \le k+1}(Test_Z - Test_{Z-1})$.

Recommend the IS{I, III}, for example. This item set is demonstrated in $TRAN_1$ and $TRAN_5$ transactions, and therefore Test({I, III}) = {$TRAN_1$, $TRAN_5$}). The periods of this item set are LSPPERIOD ({I, III}) = {1, 4, 1}).

F. Definition VI (PFP)

The maximum periodicity of an itemset [16], [17], [18] IS is represented as MAX_PRD(IS) = MAX (LSPERIOD(IS). An itemset (IS), Periodic Frequent Pattern (PFP) If |Test(IS)| ≥ $HUI_{Val}$ and MAX_PRD(IS) < MAX_PRD, where $HUI_{Val}$ and MAX_PRD are user-specified support values.

G. Definition VII (PHUI)

Given with user-specified Minimum Utility Value "$MU_{Val}$," Minimum Mean Value "$MIN\_MEAN_{value}$," Maximum Mean Value "$MAX\_MEAN_{value}$," Minimum Periodicity Value "$MIN\_PRD_{value}$" and Maximum Periodicity Value "$MAX\_PRD_{value}$" and assume all the values as positive, an itemset Y can be declared as a candidate of PHUI only If $MIN\_MEAN_{value} \le MEAN\_PRD_{value}$ (IS) ≤ $MAX\_MEAN_{value}$, $MIN\_PRD_{value}$ (IS) ≥ $MIN\_PRD_{value}$, $MAX\_PRD_{value}$ (IS) ≤ $MAX\_PRD_{value}$, and $U_{Val}$(IS) ≥ $HUI_{Val}$.

For comparison purposes, the complete set of PHUIs is seen in Table 3, If $UHI_{Val} = 20$, $MIN\_PRD_{value} = 1$, $MAX\_PRD_{value} = 3$.

Efficient pruning strategies are essential to finding efficient algorithms for IBPHUIM. The following are theorems for using time interval measures to cut search space [19], [20], [21].

(i) Lemma 1 (Monotonicity of the $MIN\_PRD_{value}$)

Let $IS_x$ and $IS_y$ be itemsets such that $IS_x \subseteq IS_y$. It follows that $MIN\_PRD_{value}$ ($IS_y$) ≤ $MIN\_PRD_{value}$ ($IS_x$).

(ii) Lemma 2 (Monotonicity of the $MAX\_PRD_{value}$)

Let $IS_x$ and $IS_y$ be itemsets such that $IS_x \subseteq IS_y$. It follows that $MAX\_PRD_{value}$ ($IS_y$) ≤ $MAX\_PRD_{value}$ ($IS_x$).

(iii) Theorem ($MAX\_PRD_{value}$)

Let $IS_x$ be an item set appearing in a database $TRAN_D$. Then, $IS_x$ and its supersets are not PHUIs if $MAX\_PRD_{value}$ ($IS_x$) > $MAX\_PRD_{value}$. Therefore, if this criterion is valid, the search effort of $IS_x$ and its supersets can be deleted [22].

A novel framework is proposed for wavelet packet analysis employed to transmute the apprehended multi-channel stress wave signals to energy information, which was consequently flattened through principal component analysis for collecting feature vector. The recent formulae, concepts and applications of big data are analyzed and were introduced for various methods like Projection, Apriori, Tree, Data Format, List, Index-based etc. [23]. They outlined high utility derivative patterns like high average utility, high utility sequential and high utility compact pattern etc. [24]. Lemmatization must work with vocabulary properly and for morphological analysis in data by removing inflectional ending to return data lemma base. The author solved outperforming the baseline algorithm for mining the same patterns and justified that Apriori-based approaches are practical for process mining without disproportionate pattern generation [25]. A survey [26] has been made to provide a comprehensive, general, and structured overview of state-of-the-art methods of Utility Patter Mining (UPM). First, we introduced an in-depth understanding of utility mining, including various concepts, examples, and various comparisons and related concepts. Then, the author presented multiple minimum high average-utility counts as an efficient model to identify more average-utility itemset and multiple less high Average-Utility (MAU) [27] counts. Where the MAU-list structure is designed for storing, SE-tree is created to mine the high average-utility of item data sets for reducing search space [28].

**Table 3** Transaction utility table

| $TRAN_{ID}$ | Profit |
|---|---|
| 1 | 140 |
| 2 | 260 |
| 3 | 340 |
| 4 | 280 |
| 5 | 340 |
| 6 | 500 |

## 3 Proposed Method

### 3.1 Method of MIN. and MAX. Time to find IBPHUIM

Suppose X is a UPM such that $MIN\_PRD_{value}(X) \geq MIN\_PRD_{value}$, $MAX\_PRD_{value}(X) \leq MAX\_PRD_{value}$, $MIN\_MEAN_{value}(X) \geq MIN\_MEAN_{value}$, and $U_{Val}(X) \geq UPM_{Val}$ then the X is said to be IBPHUIM. $MAX\_PRD_{value}$, $MIN\_PRD_{value}$, and $MIN\_MEAN_{value}$ refer to the constraints in the periodicity, which the user gives. The mining of PHUI involves discovering the high utility patterns, which satisfy the $MAX\_PRD_{value}$, $MIN\_PRD_{value}$, $MIN\_MEAN_{value}$, and $UPM_{Val}$ constraints given by the users. The utility value and periodicity are labeled according to the number of items and TRAN in the $TRAN_D$ (Table 4).

### 3.2 The Indexing Structure

This research paper uses the indexing method to avoid copying the dataset to produce a new projected dataset [29]. An indexing table is maintained, which contains the $TRAN_{id}$ and the position of the candidate itemset. UPM is primarily used to describe particular patterns in data to identify particular transactions through feature extraction. This is feasible and simple for data sets to discover less apparent associations and unexpected associations. During the recursive call of the proposed IPHUIM algorithm, the projected dataset is not created in the memory; instead, indexing is created, which serves as the pseudo projected dataset. The link is created from each of the indexes, which serve pseudo projected dataset from the dataset for the single original dataset. The items in the transactions are arranged in the order of $TWU_{Val}$. The $TARN_{id}$ and the item position identify the transaction and the position of the item in the original dataset, respectively. Creating a new copy of the projected dataset for every call is inefficient since it consumes extreme memory. The indexing structure of the dataset reduces the duplicate copies of the projected dataset by having a single copy of the data set and an index for every projected dataset.

The index table of a single extension of k-itemset can be derived from the indexing structure of k-itemset. For example, the pseudo projection dataset, which is derived from the index table of {I, II} can be derived from the index of the table of {I}, and then the index table of {I, II, V} can be derived from {I, II}. If the projected dataset contains a null transaction, then the recursive call of the IPHUIM algorithm can be stopped.

The dataset is partitioned such that each partition contains $MAX\_PRD_{value}/2$ transactions. The partition size is selected as $MAX\_PRD_{value}/2$ since the distance between any two transactions in the neighboring partition may have a MAX distance of $MAX\_PRD_{value}$. It is done to ensure that the transactions are only $MAX\_PRD_{value}$ distance apart. It helps to address items as per the total order in each transaction in the original dataset. The projections will then be performed in a pseudo-projection, in which the projected transaction specifies the index structure position item corresponding to the new transaction. The proposed database projecting method is to generalize the results of the database projection for deals involving internal/external resources used in regular patterns mining. A transaction integration technique in the projected dataset is introduced in the IPHUIM algorithm. When the itemset in a transaction is arranged in order, and if the number of itemsets continues to reduce, then it is possible for a more significant number of transactions with the same item set. Thus, the size of the projected dataset can be reduced by using the concept of transaction integration. The equivalent transactions are identified, and a single transaction in the partition replaces multiple equivalent transactions.

I. Definition (Equivalent Transactions)

Two transactions are equivalent if the items present in both transactions are the same. However, the internal utility of the items present in the transaction can vary.

II. Definition (Transaction Integration)

Transaction integration combines two or more identical transactions into a single transaction. In transaction integration, the internal utility value of the same items present in the different identical transactions is summed up and formed as a single transaction. When the volume of transactions in the dataset is low, the model's accuracy can be improved. The transaction process is achieved to decrease the data set size. The total order is followed in the ordering of transactions in the dataset. The total ordering ensures that the identical transactions are present in consecutive positions. This makes it possible to do the transaction integration in linear time. When the size of the dataset is large, then there is enormous scope for the transaction merging.

**Table 4** The IBPHUIM set in the truth table

| ITEMSET | $U_{Val}(X)$ | $MIN\_PRD_{value}(X)$ | $MAX\_PRD_{value}(X)$ |
|---|---|---|---|
| {I} | 38 | 1 | 2 |
| {II} | 36 | 1 | 1 |
| {III} | 76 | 1 | 2 |
| { I, II} | 62 | 1 | 2 |
| { I, V} | 71 | 1 | 3 |
| {II, V} | 110 | 1 | 2 |
| {I, II, V} | 135 | 1 | 2 |

The transaction merging is more probable in the projected dataset of the *k*-candidate item set when *'k'* is considerable. When *'k'* is large, the number of candidates present in the transaction is minimal, giving more chance for the equivalent transaction.

III. Definition (Transaction integration in the projected dataset)

The concept of transaction of absorption can also be implemented in the projected dataset. When multiple transactions in the projected dataset are equivalent, they are replaced by a single transaction.

#### 3.2.1 The Proposed Algorithm

The Proposed methodology uses:

1. Single-phase scanning of the dataset; from the scanned dataset, it constructs the indexed dataset, from which the k-candidate IBPHUIM is generated.
2. Combining transactions is done to minimize the transaction size by partitioning the dataset, each of size $MAX\_PRD_{value}/2$. If the $MAX\_PRD_{value}$ threshold given by the user is significant, combining transactions achieve high efficiency.
3. It uses the pattern growth approach from the 1-candidate itemset to the k-candidate itemset by recursively finding the i + 1 candidate itemset.
4. In the lower and upper limits and the closed upper limits, the candidate set is efficiently cut out of the indexed database, and the unexceptional item set is eliminated.

The Depth-First Search exploring is used to achieve the pattern growth approach. In this, IBPHUIM first checks the single candidate itemset, which satisfies the condition $TWU_{Val}(X) \geq MU_{Val}$ that are checked for IBPHUIM. Item X, which does not satisfy $TWU_{Val} \geq HUI_{Val,}$ cannot be a part of IBPHUIM. This condition reduces the search space and the dataset's size in the projection dataset. Then, the *k*-candidate item set is generated by adding one item during each recursive call of the IBPHUIM algorithm. The TWU order of ITEM(X) is followed to add a single item; the TWU order of items reduces the search space.

### 3.3 Index-Based Periodic High Utility Mining (IBPHUIM) Algorithm

*INPUT:* A Transaction Dataset with Internal and External Utility, $HUI_{Val}$ MIN_Threshold, the periodic constraints $MAX\_PRD_{value}$, $MIN\_PRD_{value}$, and $MIN\_MEAN_{value.}$

*OUTPUT:* A set of PHUI.

*STEP 1:* For each transaction, $TRAN_I$ present in the transaction dataset $TRAN_D$, repeat the following steps.

a) Calculate the $U_{Val}(ITEM_n, TRAN_{id})$
b) $ITEM_i$ calculates the $MIN_{value}$, $MAX_{value}$, and $MEAN_{value}$ periodicity values for each item while scanning the dataset.

*STEP 2:* Find the 1-candidate itemset after the scan of the dataset, which satisfies the $HUI_{Val}$, $MAX\_PRD_{value}$, $MIN\_PRD_{value}$, and $MIN\_MEAN_{value}$ constraints.

*STEP 3:* For each item $ITEM_i$ present in the 1-candidate itemset, repeat the following steps.

(a) If the $HUI_{Val}$ $(IS_n)$ in a transaction is successful in $MIN\_PRD_{value}$ $(X) \geq MIN\_PRD_{value}$, $MAX\_PRD_{value}$ $(X) \leq MAX\_PRD_{value}$, $MIN\_MEAN_{value}$ $(X) \geq MIN\_MEAN_{value}$, and $U_{Val}(X) \geq HUI_{Val,}$ then the $ITEM_n$ is added to be IBPHUIM.
(b) The index data structure is created with $TRAN_I$ and position to the $ITEM_{i.}$
(c) Arrange the transaction present in the index structure in total order.
(d) Find out the identical transaction in the index structure, do transaction integration, and update the new $MIN_{value}$, $MAX_{value}$, and $MEAN_{value}$ periodicity values.
(e) Set n = 1*//the number of ITEM$_{i.}$ Processed in the index data structure.*

*STEP 5:* Find PHUI by calling the procedure Search-IBPHUIM (X, Index_T, n) recursively for each prefix X. Let the set of returned high mean-utility itemset be IBPHUIM.

*STEP 6:* Display all the IBPHUIM found by the recursive procedure IBPHUIM (X, $Index_T$ n).

#### 3.3.1 Search-IBPHUIM (X, $Index_T$, n) Procedure

*Input:* The prefix $ITEM_i$ X, the $ITEM_i$ Index_T index table, and the prefix item n.

*Output:* The PHUI corresponding to the prefix sub $ITEMSET_X$.

*STEP 1:* Create an itemset table with five values $HUI_{Val}$, $MAX\_PRD_{value}$, $MIN\_PRD_{value}$, and $MIN\_MEAN_{value}$.

*STEP 2:* For each itemset present in the table, repeat the following steps using the prefix X and the value *'n.'*

(a) From the IndexT, retrieve the transaction ID $TRAN_I$ and position it to the $ITEM_{i.}$ and create a link.
(b) Retrieve all the item $ITEM_{i,}$ which are present after pos in the Index_T.

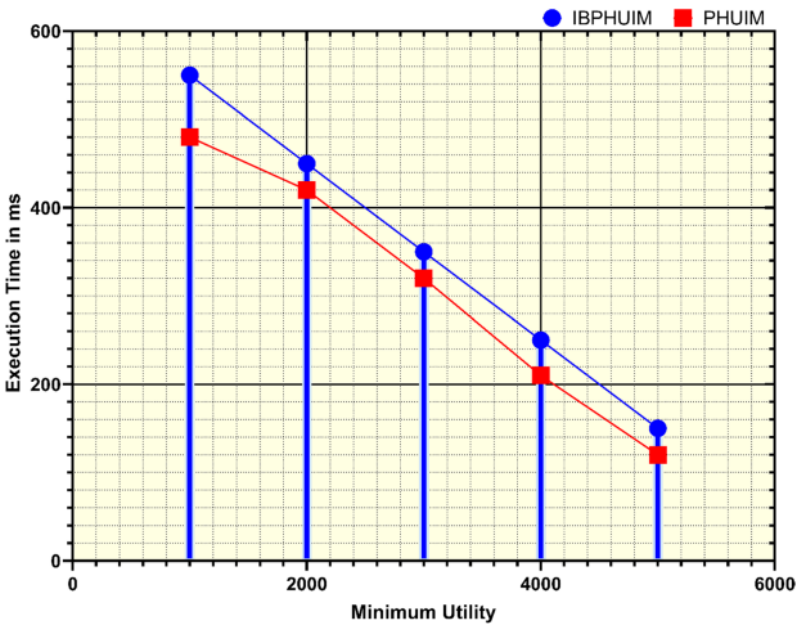


**Fig. 1** Execution time in the Supermarket Dataset

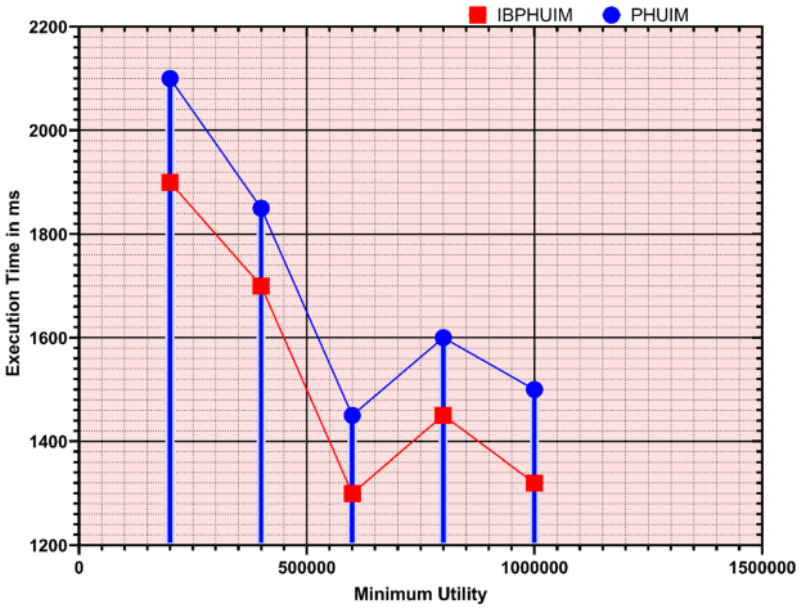


**Fig. 2** Execution time in Vegetable vs Food Market Dataset

(c) Check whether the item $ITEM_i$ and prefix X satisfy the criteria.

(d) Generate the $n+1$ candidate itemset that satisfies $MIN\_PRD_{value}(X) \geq MIN\_PRD_{value}$, $MAX\_PRD_{value}(X) \leq MAX\_PRD_{value}$, $MIN\_MEAN_{value}(X) \geq MIN\_MEAN_{value}$, and $U_{Val}(X) \geq HUI_{Val}$ then the $ITEM_n$ is added to IBPHUIM.

(e) Create an $n+1$ itemset table with itemset of prefix $n+1$ value and update the table with $U_{Val}$, $MAX\_PRD_{value}$, $MIN\_PRD_{value}$, and $MIN\_MEAN_{value}$.

*STEP 4*: For each itemset present in the $n+1$ IS table, repeat the following:

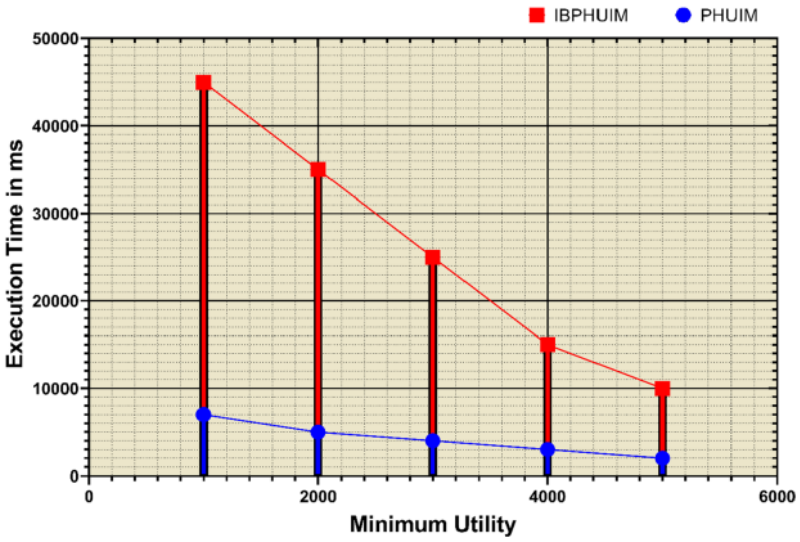


**Fig. 3** Execution Time in Food Market Dataset

(a) If the values in the itemset table satisfy MIN_PRD-$_{value}$ (X)≥MIN_PRD$_{value}$, MAX_PRD$_{value}$ (X)≤MAX_PRD$_{value}$, MIN_MEAN$_{value}$ (X)≥MIN_MEAN$_{value}$, and HUI$_{Val}$(X)≥HUI$_{Val}$, then the ITEM$_n$ is added to IBPHUIM List.

*STEP 5:* Repeat the steps for each itemset present in the itemset table.

(a) The index data structure is created with TRAN$_i$ and position to the ITEM$_i$.
(b) Arrange the transaction present in the index structure in total order.
(c) Find out the identical transaction in the index structure, do transaction integration, and update the new MIN$_{value}$, MAX$_{value}$, and MEAN$_{value}$ periodicity values.
(d) Find all the IBPHUIM by recursively calling the procedure Search PHUI(X, Index_T, n) for each prefix X.

*STEP 6*: Send all the regular high utility PHUIs identified collection.

## 4 Experimental Results

The IBPHUIM algorithms were implemented using three real datasets. The experiments were conducted by varying the MIN_Threshold and constant MIN$_{value}$, MAX$_{value}$, and MEAN_PRD$_{value}$. For example, the MIN_PRD$_{value}$ set to 10, and MAX_PRD$_{value}$ set to 75, and the MEAN_PRD$_{value}$ fixed to 15. The experiments were conducted in sample datasets: Supermarket, Vegetable Market, and Food Market datasets. The MIN_Threshold is selected according to the utility value of the itemset in the dataset. The result of the Supermarket dataset is shown in Fig. 1. The performance of the IBPHUIM shows that the execution time has improved.

The Vegetable Market data set includes 50 IS with 9500 TRAN; it's a dense dataset. Next, the performance of the IBPHUIM is compared with the PHUI. The performance of IBPHUIM is about 2 times faster than the PHUI. The results are shown below in Fig. 2.

The dataset is collected from the Open Source knowledge center. The performance of IBPHUIM is compared with PHUI. The results are given in Fig. 3.

## 5 Conclusion

In this paper, efficient indexed-based mining for finding IBPHUIM is proposed. In this approach, the pseudo dataset is projected using the indexing approach. The IBPHUIM is generated from the pseudo transaction dataset. The number of transactions in the pseudo transaction dataset is small compared to the original dataset; hence, memory efficiency and execution time are improved. Furthermore, the transaction merging can be done in the pseudo transaction dataset. The pruning technologies that can be applied are our future research interest.

## References

1. Agarwal, R.C.; Aggarwal, C.C.; Prasad, V.V.V.: A tree projection algorithm for generation of frequent item sets. J. Parallel Distrib. Comput. **61**(3), 350–371 (2001)

2. Han J.; Jian P.; Mortazavi-Asl B.; Pinto H.; Chen Q.; Dayal U.; and Hsu M. C.: "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth." In Proceedings of the 17th international conference on data engineering, pp. 215–224 (2001)
3. Berkhin, P.: A survey of clustering data mining techniques. Group. Multidimens. Data **25**, 71 (2006)
4. Bui, N.; Vo, B.; Huynh, V.N.; Lin, C.W. and Nguyen, L.T.: Mining closed high utility itemsets in uncertain databases. In Proceedings of the Seventh Symposium on Information and Communication Technology (pp. 7–14). ACM (2016) December
5. Esposito, F.; Malerba, D.; Semeraro, G.; Kay, J.: A comparative analysis of methods for pruning decision trees. IEEE Trans. Pattern Anal. Mach. Intell. **19**(5), 476–491 (1997)
6. Erwin, A.; Gopalan, R. P.; and Achuthan, N. R.: "Efficient mining of high utility itemsets from large datasets." In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 554–561. Springer, Berlin, Heidelberg, (2008)
7. Fournier-Viger, P.; Lin, J.C.W.; Duong, Q.H. and Dam, T.L.; 2016, July. PHM: mining periodic high-utility itemsets. In Industrial Conference on Data Mining (pp. 64–79). Springer International Publishing
8. Fournier-Viger, P.; Lin, J.C.W.; Gomariz, A.; Gueniche, T.; Soltani, A.; Deng, Z. and Lam, H.T.: The SPMF open-source data mining library version 2. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 36–40). Springer International Publishing (2016) September
9. Han, J.; Dong G.; and Yin, Y: "Efficient mining of partial periodic patterns in time series database." In Data Engineering, 1999. Proceedings., 15th International Conference on, pp. 106–115. IEEE, (1999)
10. Hipp, J.; Güntzer, U.; Nakhaeizadeh, G.: Algorithms for association rule mining—a general survey and comparison. ACM SIGKDD Explor. Newsl **2**(1), 58–64 (2000)
11. Hong, T.-P.; Lee, C.-H.; Wang, S.-L.: Effective utility mining with the measure of mean utility. Expert Syst. Appl. **38**(7), 8259–8265 (2011)
12. Keim, D.A.: Information visualization and visual data mining. IEEE Trans. Visual Comput. Graphics **8**(1), 1–8 (2002)
13. Lee, C.-H.; Lin, C.-R.; and Chen M.-S: "On mining general temporal association rules in a publication database." In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, pp. 337–344. IEEE, (2001)
14. Liu, K.; Kargupta, H.; Ryan, J.: Random projection-based multiplicative data perturbation for privacy-preserving distributed data mining. IEEE Trans. Knowl. Data Eng. **18**(1), 92–106 (2006)
15. Oliver, J.J.; and Hand D.J.: "On pruning and averaging decision trees." In Machine Learning: Proceedings of the Twelfth International Conference, pp. 430–437. (2016)
16. Park, J.S.; Chen, M.-S.; and Yu P.S.: An effective hash-based algorithm for mining association rules. Vol. 24, no. 2. ACM, (1995)
17. Pillai, J.; Vyas, O.P.: Overview of itemset utility mining and its applications.". Int. J. Comput. Appl. **5**(11), 9–13 (2010)
18. Sarawagi, S.; Thomas, S.; and Agrawal, R.: Integrating association rule mining with relational database systems: Alternatives and implications. **27**(2). ACM, (1998)
19. Shie, B.-E.; Tseng, V.S.; and Philip S.Y.: "Online mining of temporal maximal utility itemsets from data streams." In Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1622–1626. ACM, (2010)
20. Tseng, V.S.; Shie, B.E.; Wu, C.W.; Philip, S.Y.: Efficient algorithms for mining high utility itemsets from transactional databases. IEEE Trans. Knowl. Data Eng. **25**(8), 1772–1786 (2013)
21. Verleysen, M.; François, D.: The curse of dimensionality in data mining and time series prediction. IWANN **5**, 758–770 (2005)
22. Yao, H.; Hamilton, H.J.; and Butz C.J.: "A foundational approach to mining itemset utilities from databases." In Proceedings of the 2004 SIAM International Conference on Data Mining, pp. 482–486. Society for Industrial and Applied Mathematics, (2004)
23. Zhang, C.; and Zhang, S.: Association rule mining: models and algorithms. Springer-Verlag, (2002)
24. Zheng, Z.; Kohavi, R.; and Mason, L.: "Real-world performance of association rule algorithms." In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 401–406. ACM, (2001)
25. Yang, Yu.; Dackermann, U.; Li, J.; Niederleithinger, E.: Wavelet packet energy-based damage identification of wood utility poles using support vector machine multi-classifier and evidence theory. Struct. Health Monit. **18**(1), 123–142 (2019)
26. Zhan, C.; Han, M.; Sun, R.; Shiyu, Du.; Shen, M.: A survey of key technologies for high utility patterns mining. IEEE Access **8**, 55798–55814 (2020)
27. Javed, M.F.; Nawaz, W. and Khan, K.U.: "HOVA-FPPM: flexible periodic pattern mining in time series databases using hashed occurrence vectors and apriori approach", **2021**: 1–14, (2021)
28. Gan, W.; Lin, J.C.-W.; Fournier-Viger, P.; Chao, H.-C.; Tseng, V.S.; Philip, S.Y.: A survey of utility-oriented pattern mining. IEEE Trans. Knowl. Data Eng. **33**(4), 1306–1327 (2019)
29. Lin, J.C.-W.; Ren, S.; Fournier-Viger, P.: MEMU: more efficient algorithm to mine high average-utility patterns with multiple minimum average-utility thresholds. IEEE Access **6**, 7593–7609 (2018)

# Utilizing Index-Based Periodic High Utility Mining to Study Frequent Itemsets

ORIGINALITY REPORT

| 7% | 4% | 6% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to University of Dammam<br>Student Paper | 1% |
| 2 | link.springer.com<br>Internet Source | 1% |
| 3 | Roy Setiawan, Ramakoteswara Rao Ganga, Priya Velayutham, Kumaravel Thangavel et al. "Encrypted Network Traffic Classification and Resource Allocation with Deep Learning in Software Defined Network", Wireless Personal Communications, 2021<br>Publication | 1% |
| 4 | "High-Utility Pattern Mining", Springer Science and Business Media LLC, 2019<br>Publication | 1% |
| 5 | web-tools.uts.edu.au<br>Internet Source | 1% |
| 6 | Lan, G.C.. "Discovery of high utility itemsets from on-shelf time periods of products", Expert Systems With Applications, 201105<br>Publication | 1% |

| | | |
|---|---|---|
| 7 | GUO-CHENG LAN, TZUNG-PEI HONG, VINCENT S. TSENG. "EFFICIENTLY MINING HIGH AVERAGE-UTILITY ITEMSETS WITH AN IMPROVED UPPER-BOUND STRATEGY", International Journal of Information Technology & Decision Making, 2012<br>Publication | 1 % |
| 8 | N. Satheesh, M.V. Rathnamma, G. Rajeshkumar, P. Vidya Sagar, Pankaj Dadheech, S.R. Dogiwal, Priya Velayutham, Sudhakar Sengan. "Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network", Microprocessors and Microsystems, 2020<br>Publication | 1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | < 1% |
| Exclude bibliography | On | | |