Discovering Periodic High Utility Itemsets in a Discrete Sequence

Philippe Fournier-Viger, Youxi Wu, Duy-Tai Dinh,
Wei Song, Jerry Chun-Wei Lin

Abstract Periodic itemset mining is the task of finding all the sets of items (events or symbols) that regularly appear in a sequence. One of the most important applications is customer behavior analysis, where a periodic itemset found in a sequence of customer transactions indicates that the customer regularly buys some items together. Using this information, marketing strategies can be tailored and product recommendation can be done. However, a major limitation of traditional periodic itemset mining is that the relative importance of each item is not taken into account, and that each item cannot appear more than once at each time step of the sequence. But in real-life, not all items are equally important (e.g. selling a cake yield less profit than selling a computer) and a customer may buy multiple units of the same item at the same time (e.g. many cakes). To address these factors, the task of periodic frequent itemset mining was generalized as that of periodic high utility itemset mining, where the goal is to find the sets of items that not only periodically appear in a sequence but also have a high importance (e.g. yield a high profit). This chapter provides an overview of this task and presents an algorithm to solve this problem. Moreover, a variation of this task that con-

Philippe Fournier-Viger
Harbin Institute of Technology (Shenzhen), Shenzhen, China, e-mail: `philfv8@yahoo.com`

Youxi Wu
Hebei University of Technology, Tianjin, China, e-mail: `wuc567@163.com`

Duy-Tai Dinh
Japan Advanced Institute of Science and Technology, e-mail: `taidinh@jaist.ac.jp`

Wei Song
North China University of Technology, e-mail: `songwei@ncut.edu.cn`

Jerry Chun-Wei Lin
Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences (HVL), Bergen, Norway e-mail: `jerrylin@ieee.org`

sisting of discovering irregular high utility itemsets is also discussed. Finally, some research opportunities are listed.

1 Introduction

In recent decades, more and more data has been collected and stored in databases. Making sense of this data has become challenging as analyzing large volumes of data by hand is prone to error and time-consuming. As a solution, data mining has emerged as an important task. It consists of applying algorithms to (semi)-automatically analyze data. Various types of data can be analyzed using data mining techniques such as spatial data [65], sequences [26], trajectories [16, 80], databases [24, 27], and graphs [42]. Data mining algorithms can generally be viewed as designed to build predictive models from data (to predict the future) or to find interesting patterns that can help to understand the data (to describe the data or understand the past).

The task of finding interesting patterns in data is known as *pattern mining*. The goal is to find sets of values that appear together in the data and meet some criteria set by the user. The most fundamental problem in pattern mining is known as *frequent itemset mining* [1, 24, 52]. The input is a database of transactions (records), as well as a parameter called the minimum support threshold *minsup*. The output is the frequent itemsets, that is the sets of value that appear at least in *minsup* records of the input database. For example, frequent itemset mining can be applied on a database of transactions made by a customer to reveal information such that the customer often purchased the items *cake* and *milk* together.

Frequent itemset mining is useful and has many applications. However, it considers that records are unordered. But for several domains such as analyzing customer transactions, the records (transactions) are ordered sequentially (e.g. by time). To find patterns that regularly appear in a sequence of events (e.g. transactions), the problem of frequent itemset mining was redefined as *frequent periodic itemset mining* [3, 4, 32]. A periodic pattern is a set of values that regularly appear in a sequence. For instance, a pattern found in transaction data from a customer may indicate that the customer regularly buys milk with cake.

Although periodic frequent pattern mining is useful, a drawback is that the relative importance of each item is not taken into account. In other words, it is assumed that all items are equally important. Moreover, another limitation is that the quantities of items are assumed to be binary (an item either appear or not in a record). However, in real-life, items may appear more than once at each time step (e.g. a customer may buy five apples or just one), and some items are more important than others (e.g. selling a smartphone yield more profit than selling a pen). As a result, algorithms for periodic frequent

itemset mining may find a huge amount of periodic patterns that are not important (e.g. yield a low profit) and miss many rare periodic patterns that have a high importance (e.g. yield a high profit). To address these issues, the task of periodic pattern mining was generalized as that of *periodic high utility itemset mining* (PHUIM) [23]. The goal is to find the sets of items that not only periodically appear in a sequence but also have a high importance (e.g. yield a high profit). This chapter describes that problem of PHUIM.

The chapter is organized as follows. Section 2 briefly reviews related work on frequent periodic itemset mining. Then, Section 3 describes the problem of periodic high utility itemset mining and an efficient algorithm named PHM (Periodic High utility itemset Miner) [23] to solve this problem. Thereafter, Section 4 discusses a problem variation to find irregular high utility itemsets using an algorithm called $PHM_{irregular}$. Finally, Section 5 discusses some other variations [11] and research opportunities, and Section 6 draws a conclusion.

2 Preliminaries about Periodic Itemset Mining

This section briefly reviews concepts from periodic itemset mining that are useful for this chapter. Let there be a finite set I of items. An item is a symbol or event type. An itemset X is a set of items ($X \subseteq I$). An itemset X that has k items is called a k-itemset. The input in periodic itemset mining is a sequence of transactions, also called a transaction database.

Definition 1 (Transaction database) A transaction database is a sequence of transactions $s = \langle T_1, T_2, \ldots T_m \rangle$, where $T_j \subseteq I$. The integer w used to denote a transaction T_w is called its transaction identifier and is unique.

Example 1 Consider the database of Table 1. This database is a sequence of seven transactions made by a customer, denoted as T_1, T_2, ..., T_7. Each transaction is a set of items that the customer purchased at a given time from the sets of products $I = \{a, b, c, d, e\}$. For instance, the transaction T_5 indicates that the customer has bought items a, c and d together. In this case, transactions are ordered by purchase times. An itemset such as $\{a, b, c, d, e\}$ is a 5-itemset because it contains five items.

It is to be noted that although a sequence of transactions is often ordered by time, it can be ordered according to other criteria. For example, a sequence of nucleotides in the genome of a virus such as $\langle \{A\}, \{A\}, \{C\}, \{G\}, \{T\}, \{A\}, \ldots \rangle$ can be viewed as a sequence of transactions where the order does not depend on time [54]. Another example is to consider a text document as a sequence of transactions, where each transaction is the set of words (items) appearing in a sentence. In that case, the sequential ordering represents the

Table 1: A sequence of transactions (transaction database)

Transaction ID	Transaction
T_1	a, c
T_2	e
T_3	a, b, c, d, e
T_4	b, c, d, e
T_5	a, c, d
T_6	a, c, e
T_7	b, c, e

order between sentences rather than time. If a transaction database is ordered by time or contain timestamps, it is sometimes called a temporal database or temporal transaction database.

To find periodic patterns in a sequence of transactions, the task of frequent periodic itemset mining has been proposed, which aims at finding itemsets that regularly appear over time [3, 32]. This task is defined based on the following definitions.

Definition 2 (Sequence containment) Consider a sequence $s_v = \langle V_1, V_2, \ldots, V_k \rangle$ and another sequence $s_w = \langle W_1, W_2, ..., W_l \rangle$. It is said that the sequence s_v is a subsequence of s_w (denoted as $s_v \sqsubseteq s_w$) iff some integers $1 \leq b1 < b2 < ... < bk \leq m$ exist such that $V_1 \subseteq W_{b1}, V_2 \subseteq W_{b2}, \ldots, V_k \subseteq W_{bl}$.

Example 2 The sequence $s_v = \langle (a, e), (b, d) \rangle$ is a subsequence of the sequence of transactions depicted in Table 1 since $\{a, e\}$ is a subset of transaction T_3 and it is followed by $\{b, d\}$ in T_4. Another example is that $\langle (a, e), (b, d) \rangle \sqsubseteq \langle (a, e), (b)(b, c, d) \rangle$.

To check if an itemset is periodic in a sequence of transactions, it is necessary to define what it means to be periodic using some criteria [3, 32]. This is done by using some evaluation functions that measure how periodic a pattern is. The basic evaluation functions used in frequent periodic itemset mining to select periodic patterns are the support and the maximum periodicity, which are based on the following definitions.

Definition 3 (Support function) The support of an itemset X in a sequence of transactions s is the number of transactions from s that contain X. The support is formally defined as $sup(X, s) = |TR(X, s)|$ and simply denoted as $sup(X)$ without s when the context is clear.

Example 3 Consider the itemset $X = \{a, e\}$ and the sequence s of Table 1. The itemset X has a support of 2 since X appears in two transactions (T_3 and T_6). This is denoted as $sup(X, s) = 2$.

Definition 4 (Consecutive transactions with respect to an itemset) For a sequence of transactions s and an itemset $X \sqsubseteq s$, the ordered list of

transactions (from s) containing X is denoted and defined as $TR(X,s) = \langle T_{g_1}, T_{g_2}, \ldots, T_{g_k} \rangle \sqsubseteq s$. Two transactions T_x and T_y in $TR(X,s)$ are said to be *consecutive with respect to* X if there is no transaction $T_z \in s$ such that $x < z < y$ and $X \subseteq T_z$.

Example 4 Consider the sequence s of Table 1 and the itemset $X = \{b, c\}$. The list of transactions from s containing X is $TR(X, s_1) = \{T_3, T_4, T_7\}$. The transactions T_3 and T_4 are consecutive transactions with respect to X. The transactions T_4 and T_7 are also consecutive transactions with respect to X.

Definition 5 (Periods of an itemset) For an itemset X, the period of two consecutive transactions T_x and T_y containing X is defined as $per(T_x, T_y) = y - x$. The periods of an itemset X in a sequence s are denoted and defined as $pr(X, s) = \{per_1, per_2, ..., per_{k+1}\}$ where $per_1 = g_1 - g_0$, $per_2 = g_2 - g_1$, $\ldots per_{k+1} = g_{k+1} - g_k$, and $g_0 = 0$ and $g_{k+1} = n$, respectively.

Example 5 Continuing the previous example, the periods of X in s are $pr(X, s) = \{3, 1, 3, 0\}$. Consider another itemset $Y = \{a, c\}$. The periods of Y in s are $pr(Y, s) = \{1, 2, 2, 1, 1\}$.

Definition 6 (Maximum periodicity function) A common evaluation function to determine if an itemset X is periodic in a sequence s is called the maximum periodicity and is defined as $maxPr(X, s) = argmax(pr(X, s))$ [3].

Example 6 The maximum periodicity of $X = \{a, e\}$ and $Y = \{a, c\}$ in s are respectively $maxPr(X, s) = argmax(\{3, 1, 3, 0\}) = 3$ and $maxPr(Y, s) = argmax(\{1, 2, 2, 1, 1\}) = 2$.

Traditionally, an itemset X is called a *frequent periodic itemset* if it has a support that is no less than a $minSup$ threshold set by the user ($sup(X, s) \geq minsup$), and if the maximum periodicity is no less than a user-specified threshold $maxPer$ ($maxPr(X, s) \leq maxPer$) [3, 32]. For instance, if the user sets $minSup = 2$ and $maxPer = 2$, the itemset $\{a, c\}$ is a periodic itemset in the sequence of Table 1.

Though finding frequent periodic itemsets can be useful, a problem with this definition of periodic itemset is that it is too strict since if an itemset has a single period that is greater than $maxPer$, it is deemed non periodic. For instance, if $maxPer = 1$ week and a customer has always bought bread every day over a year for one week, then the itemset $\{bread\}$ will not be periodic. Several alternative evaluations functions have been proposed to address this problem. Some functions that have been studied are:

- the *average periodicity* [32] defined as $avgPr(X, s) = average(pr(X, s))$,
- the *minimum periodicity* [32] defined as $minPr(X, s) = argmin(pr(X, s))$,
- the *standard deviation of periods* [21, 55, 56], defined as $stanDev(X, s) = stanDev(pr(X, s))$,
- and the *variance of periods* [44, 61, 62].

Besides, some measures were proposed to evaluate whether an itemset is periodically stable over time (consecutive periods remains more or less the same) [28, 33], and statistical testing as also been employed to find statistically significant periodic patterns [55].

In the remaining of this chapter, the following definition of periodic itemsets is used, as it is more general than the traditional definition [23, 32].

Definition 7 (Periodic itemset) An itemset X is considered to be a *periodic itemset* if it satisfies three conditions: (1) $minAvg \leq avgper(X) \leq maxAvg$, (2) $minper(X) \geq minPer$, and (3) $maxper(X) \leq maxPer$, where $minAvg$, $maxAvg$, $minPer$ and $maxPer$ are positive numbers.

This definition is quite flexible as it let the user specifies on average how large the periods of an itemset should be (by condition 1), and two constraints on the minimum and maximum size of periods (by conditions 2 and 3). It is to be noted that some conditions can also be ommitted if needed. For example, if only condition 3 is used, the traditional definition of periodic pattern is obtained.

Example 7 Consider that $minAvg = 1$, $maxAvg = 2$, $minPer = 1$ and $maxPer = 3$. The itemset $X = \{b, c, e\}$ is a periodic itemset since $avgper(X) = 1.75 \leq maxAvg$, $avgper(X) = 1.75 \geq minAvg$, $minper(X) = 1 \geq minPer$, $maxper(X) = 3 \leq maxPer$.

It can be observed that the definition of periodic itemset does not consider timestamps. However, it can be easily generalized to be used for transactions that have timestamps. In that case, the transaction identifiers can be replaced by timestamps for the calculation of the periods of an itemset. In the case where two transactions have the same timestamp, an arbitrary order between them can be established.

To develop efficient algorithms for finding periodic itemsets using the above definition, a few important properties of the evaluation functions are the following [3, 23, 32].

Property 1 (The average, minimum and maximum periodicity are monotonic) Consider two itemsets X and Y satisfying the relationship $X \subset Y$. It follows that $avgper(Y) \geq avgper(X)$. Moreover, it can be proved that $minper(Y) \geq minper(X)$ and $maxper(Y) \geq maxper(X)$.

Property 2 (Eliminating non periodic itemsets using the maximum periodicity) Consider an itemset X and a sequence of transaction s. The itemset X and its supersets are not periodic itemsets if $maxper(X) > maxPer$ [3].

Property 3 (Eliminating non periodic itemsets using the average periodicity) An itemset X is not a periodic itemset in a sequence of transactions s if $avgper(X) > maxAvg$. This condition can be rewritten as $|sup(X, s)| < (|s|/maxAvg) - 1$ [23, 32].

As it will be shown, the Property 2 and 3 can be used to avoid exploring the whole search space of itemsets, and thus to improve efficiency of algorithms for mining periodic itemsets.

3 Periodic High Utility Itemset Mining

Though discovering periodic frequent itemsets is useful and can reveal interesting patterns in data [3], it has two important limitations inherited from frequent itemset mining [1, 24, 40, 79]. They are that (1) an item cannot appear more than once in each transaction and (2) all items are considered to have the same importance. However, these assumptions are not true for several applications. For instance, in the context of customer transaction data analysis, a client may buy more than one bottle of milk in a transaction and some items may be viewed as more important than others because their sale yield a higher profit. Ignoring this information may lead to finding many uninteresting patterns (e.g. patterns that yield a low profit for a store).

To find patterns that have a high importance (e.g. a high profit) rather than only frequent patterns, the problem of *frequent itemset mining* was generalized as *high utility itemset mining* [27]. Then, inspired by this, frequent periodic itemset mining was generalized as periodic high utility itemset mining [23]. The next subsection first briefly reviews concepts from high utility itemset mining and presents the problem of periodic high utility itemset mining that is a generalization of periodic frequent itemset mining. Then, the next subection presents an efficient algorithm, named PHM (Periodic High utility itemset Miner) [23].

3.1 The Problem

The input data in high utility itemset mining is a quantiative transaction database, and the goal is to find *high utility itemsets* (itemsets that have a high importance such as that yield a high profit) [27]. The concept of quantitative transaction database is defined next.

Definition 8 (Quantitative transaction database) A quantitative transaction database s (also called sequence of quantitative transactions) is a transaction database where a positive number $p(i)$ named external utility is given for each item i, which represents its relative importance in the database. Moreover, a positive number $q(i, T_c)$ called internal utility is provided to indicate the importance of each item i in each transaction T_c.

Example 8 Table 2 lists seven transactions of a quantitative transaction database that are the transactions made by a customer over time. There are

five items $I = \{a, b, c, d, e\}$. In each transaction, the relative importantce of each item represents its purchase quantity. For example, in transaction T_6, 2 units of item a is purchased $(q(a, T_5) = 2)$, 6 units of item c is purchased $(q(c, T_5) = 6)$, and 2 units of item e is purchased $(q(e, T_5) = 2)$. The relative importance of items (the external utility) is given in the last line of Table 2. For instance, it is indicated that selling one unit of item d yields a profit of $p(d) = 2\$$ while the sale of a unit of a gives a $p(a) = 5\$$ profit.

Table 2: A quantitative transaction database

Transaction ID	Quantitative transaction
T_1	$(a, 1), (c, 1)$
T_2	$(e, 1)$
T_3	$(a, 1), (b, 5), (c, 1), (d, 3), (e, 1)$
T_4	$(b, 4), (c, 3), (d, 3), (e, 1)$
T_5	$(a, 1), (c, 1), (d, 1)$
T_6	$(a, 2), (c, 6), (e, 2)$
T_7	$(b, 2), (c, 2), (e, 1)$
External utility values of items	
$p(a) = 5, p(b) = 2, p(c) = 1, p(d) = 2, p(e) = 3$	

Transactions in a quantitative transaction database can be ordered by time or according to other criteria, and may represent shopping data or other types of symbolic data. In the case where all external utility values and internal utility values are either 0 or 1, a quantitative transaction database is a transaction database, as defined in the previous section.

In high utility itemset mining, patterns are selected based on their utility (e.g. importance or profit), an evaluation function defined as follows.

Definition 9 (Utility of items and itemset) Let there be a transaction T, an itemset X, and an item i. The utility of i in T is the product of its internal and external utility, that is $u(i, T) = p(i) \times q(i, T)$. For an itemset $X \subseteq T$, the utility of X in T is the sum of the product of the internal and external utility of each item in X, that is : $u(X, T) = \sum_{i \in X} u(i, T)$. For an itemset $X \nsubseteq T$, the utility of X in T is 0, that is $u(X, T) = 0$. The utility of X in the input database s is the sum of its utility for all transactions that is: $u(X) = \sum_{X \subseteq T \in D} u(X, T)$.

Example 9 For the database of Table 2, the utility of b in T_7 is $u(b, T_7) = q(b, T_7) \times p(b) = 2 \times 2 = 4$. The utility of c in T_7 is $u(c, T_7) = q(c, T_7) \times p(c) = 2 \times 1 = 2$. The utility of $\{b, c\}$ in T_7 is $u(\{b, c\}, T_7) = u(b, T_7) + u(c, T_7) = 4 + 2 = 6$. The utility of $\{b, c\}$ in the database is $u(\{b, c\}) = u(\{b, c\}, T_3) + u(\{b, c\}, T_4) + u(\{b, c\}, T_7) = (10 + 1) + (8 + 3) + (4 + 2) = 28$.

In the problem of high utility itemset mining, the goal is to find all high utility itemsets. The problem is defined as follows.

Definition 10 (High utility itemset mining) Let there be an itemset X, a database s and a minimum utility threshold $minUtil$ set by the user (a positive number. If $u(X) \geq minUtil$ then X is a high utility itemset. Otherwise, it is a low utility itemset. The goal of *high utility itemset mining* is to find all high utility itemsets.

Example 10 Continuing the running example, assume that $minUtil = 35$. Then, there are three high utility itemsets, which are $\{b, c, d, e\}$ with a utility of 40, $\{b, c, e\}$ with a utility of 37, and $\{b, d, e\}$ with a utility of 36.

To find high utility itemsets, several efficient algorithms have been designed such as Two-phase [50], HUP-Growth [48], FHM [30], ULB-Miner [14], mHUIMiner [58], EFIM [81], and HUI-Miner [49]. A good overview of techniques and algorithms for high utility itemset mining can be found in a recent survey [27].

The concept of high utility itemset has combined with that of periodic patterns to find patterns that are not only periodic but also have a high importance (yield a large profit). This problem, called periodic high utility itemset mining [23] is defined as follows.

Definition 11 (Periodic high utility itemset mining) An itemset X is considered to be a *periodic high utility itemset* (PHUI) if it satisfies four conditions: (1) $minAvg \leq avgper(X) \leq maxAvg$, (2) $minper(X) \geq minPer$, (3) $maxper(X) \leq maxPer$, and (4) $u(X) \geq minUtil$, where $minAvg$, $maxAvg$, $minPer$, $maxPer$ and $minUtil$ are positive numbers.

Example 11 Consider again the sequence of Table 2 as example. Let $minPer = 1$, $maxPer = 3$, $minAvg = 1$, $maxAvg = 2$, and $minUtil = 28$. The periodic high utility itemsets discovered in this data are listed in Table 3.

Table 3: The set of PHUIs in the running example

Itemset	u(X)	$sup(X)$	minper(X)	maxper(X)	avgper(X)
$\{b, e\}$	31	3	1	3	1.75
$\{b, c, e\}$	37	3	1	3	1.75
$\{b, c\}$	28	3	1	3	1.75
$\{a, c\}$	34	4	1	2	1.4

It can be shown that the traditional problem of periodic frequent itemset mining is a special case of that problem where all internal and external utility values are set to 0 or 1 and only condition (3) is used.

3.2 The PHM algorithm

This subsection describes the PHM algorithm to discover all periodic high utility itemsets in a quantitative transaction database, where transactions are ordered by time or other criteria. The PHM algorithm is an extension of the FHM [30] algorithm, a popular, simple and efficient algorithm for mining high utility itemsets, which is an improved version of the HUI-Miner algorithm [49].

The search space of periodic high utility itemset mining contains $2^{|I|} - 1$ possible itemsets. For instance, in the running example, $I = \{a, b, c, d, e\}$. Hence, there are $2^{|5|} - 1 = 31$ possible itemsets such as $\{a\}$, $\{b\}$, $\dots \{a, b\}$, $\{a, c\} \dots \{b, c, d\}, \{b, c, e\} \dots \{a, b, c, d, e\}$. A naive algorithm to find all periodic high utility itemsets would scan the database and calculate the utility and periods of all possible itemsets to find the solution. But this is inefficient as the search space can be very large. The PHM algorithm adopts a different approach to avoid looking at all possibilities. The next paragraphs introduce the key ideas of this algorithm, and then the pseudo-code is presented.

For the purpose of processing, the PHM algorithm supposes that there exists a *total order* $\succ$ on the items from I. This order $\succ$ can be any order such as the alphabetical order. This order is used by PHM to search for itemsets in a systematic way, that is to not look at the same itemset more than once.

To eliminate many itemsets that are not high utility itemsets from the search space, PHM uses the *Transaction-Weighted Utilization (TWU)* measure, which was proposed by Liu et al. [50].

Definition 12 (Transaction utility, TWU) Let there be an itemset X and a transaction T. The notation $TU(T)$ denotes the *transaction utility* of transaction T, which is defined as $TU(T) = \sum_{x \in T} u(x, T)$. The notation $TWU(X)$ denotes the TWU of X, which is defined as $TWU(X) = \sum_{X \subseteq T \in D)} TU(T)$.

Example 12 Following the previous example, the transactions $T_1, T_2, \dots T_7$, respectively have transaction utility values of 6, 3, 25, 20, 8, 22 and 9. The TWU of the itemset $\{b, c\}$ is $TWU(\{b, c\}) = TU(T_3) + TU(T_4) + TU(T_7) = 25 + 20 + 9 = 54$.

A powerful property for reducing the search space using the TWU is the following [50].

Theorem 1 (Pruning search space using the TWU)

For an itemset X, if $TWU(X) < minUtil$, then $u(X) < minUtil$, and for any superset $Y \supset X$, we have $u(Y) < minUtil$. In other words, both X and Y are not high utility itemsets and can be ignored.

To be able to calculate the utility and periods of itemsets, the PHM algorithms utilize a structure called *utility list* [49], which is defined as follows.

Initially, PHM reads the database to create the utility list of each itemset containing one item. Then, PHM generates utility lists of larger itemsets by joining utility lists of smaller itemsets. The advantage of this approach is that all itemsets can be explored without having to repeatedly read the database. The utility list structure is defined as follows.

Definition 13 (Utility list) Recall that $\succ$ is a total order defined on items in I. Let there be an itemset X an a database s. The utility list of X is denoted as $ul(X)$ and is defined as a list of tuples where there is a tuple $(tid, iutil, rutil)$ for each transaction T where $X \subseteq T$. The $iutil$ element of a tuple stores the value $u(X, T)$. The $rutil$ element of a tuple stores the value $\sum_{i \in T \wedge i \succ x \forall x \in X} u(i, T)$ called *remaining utility*.

Example 13 For instance, assume that $\succ$ is the alphabetical order. The utility list of $\{a\}$ contains four tuples: $\{(T_1, 5, 1),\ T_3, 5, 20),\ (T_5, 5, 3),\ (T_6, 10, 12)\}$. The utility list of $\{d\}$ has three tuples: $\{(T_3, 6, 3),\ (T_4, 6, 3),\ (T_5, 2, 0)\}$. And the utility list of $\{a, d\}$ contain two tuples: $\{(T_3, 11, 3),\ (T_5, 7, 0)\}$.

The utility lists of itemsets having single items are constructed by reading the database. For itemsets having more than one item, the following join operation is used to build the utility list [49]. Let there be two items $x \succ y$ such that $x, y \in I$. The utility list of itemset $\{x, y\}$, denoted as $ul(\{x, y\})$ can be created by adding a tuple $(ex.tid, ex.iutil + ey.iutil, ey.rutil)$ to $ul(\{x, y\})$ for each pair of tuples $ex \in ul(\{x\})$ and $ey \in ul(\{y\})$ such that $ex.tid = ey.tid$. For itemsets containing more than two items, the join operation is done as follows. Let there be two itemsets $P \cup \{x\}$ and $P \cup \{y\}$ such that $x \succ y$ and $P \subset I$. The utility list of itemset $P \cup \{x, y\}$, denoted as $ul(P \cup \{x, y\})$ is created by adding a tuple $(ex.tid, ex.iutil + ey.iutil - ep.iutil, ey.rutil)$ to $ul(P \cup \{x, y\})$ for each set of tuples $ex \in ul(\{x\})$, $ey \in ul(\{y\})$, $ep \in ul(P)$ such that $ex.tid = ey.tid = ep.tid$.

The utility-list of an itemset is useful as it allows to directly obtain its utility and to reduce the search space. The utility of an itemset X is simply the sum of the *iutil* values in $ul(X)$. To reduce the search space using the utility, the following property is used [49]:

Theorem 2 (Search space reduction with utility list) *For an itemset X, an extension of X is an itemset that is obtained by appending an item y to X such that $y \succ i$, $\forall i \in X$. If the sum of iutil and rutil values in $ul(X)$ is less than minUtil, X and its extensions are low utility itemsets and can be ignored.*

The utility-list structure can also be used to directly calculate the periods of an itemset X, and thus determine if it is a periodic itemset. This can be done by looking at the *tid* elements of the utility list of X.

Pseudocode. The pseudocode of the main procedure of the PHM algorithm is shown in Algorithm 1. The input is a quantitative transaction database as well as the five thresholds $minUtil$, $minAvg$, $maxAvg$, $minPer$

and $maxPer$. The algorithm outputs the set of all periodic high utility item-sets. PHM initially reads the input database to calculate the following values for each item $i \in I$: $sup(\{i\})$, $TWU(\{i\})$, $minper(\{i\})$ and $maxper(\{i\})$. Moreover, PHM calculates $\gamma = (|s|/maxAvg) - 1$, which is needed by Property 3 to reduce the search space. Afterward, a set of items I^* is created by PHM, which contains each item i that has a TWU value that is greater or equal to $minUtil$, a maximum periodicity that is no less than $maxPer$, and where i appears in at least γ transactions (as per the Property 3). Thereafter, the total order $\succ$ on items is established as in the HUI-Miner algorithm [49] as the order of ascending TWU values and the alphabetical order when two items have the same TWU. The next action done by PHM is to read the database again and sort in memory each transaction according to $\succ$. At the same time, a utility-list is built for each item $i \in I^*$. Then, PHM starts to recursively search for periodic high utility itemsets by calling the *Search* procedure with the following parameters: the empty itemset $\emptyset$, the set I^*, γ, $minUtil$, $minAvg$, $minPer$, $maxPer$ and $|s|$.

Algorithm 1: The PHM algorithm

input : s: a transaction database,
$minUtil$, $minAvg$, $maxAvg$, $minPer$ and $maxPer$: the thresholds
output: the set of periodic high-utility itemsets

1 Scan s once to calculate $TWU(\{i\})$, $minper(\{i\})$, $maxper(\{i\})$, and $|g(\{i\})|$
   for each item $i \in I$;
2 $\gamma \leftarrow (|s|/maxAvg) - 1$;
3 $I^* \leftarrow$ each item i such that $\text{TWU}(i) \geq minUtil$, $sup(\{i\}) \geq \gamma$ and
   $maxper(\{i\}) \leq maxPer$;
4 Establish the total order $\succ$ on I^* as that of TWU ascending values;
5 Search ($\emptyset$, I^*, γ, $minUtil$, $minAvg$, $minPer$, $maxPer$, $|s|$);

Algorithm 2 presents the *Search* procedure. It receives an itemset P, some extensions of P that have the form Pz (which was obtained by adding an item z to P), as well as the original input parameters and the input sequence length $|s|$. The first step done by this procedure is to iterate over the extensions in P. For each such extension Px, the procedure first calculates the average periodicity of Px as the ratio of $|s|$ to the number of elements in the utility list of Px plus one. Thereafter, the procedure checks if the following conditions are met: (1) the average periodicity of Px is no less than $minAvg$ and no greater than $maxAvg$, (2) the sum of the *iutil* values of the utility list of Px is no less than $minUtil$ (by Theorem 2), and (3) according to the utility list of Px, the minimum (maximum) periodicity of Px is no less (not greater) than $minPer$ ($maxPer$). If these conditions are met, then Px is output as a periodic high utility itemset. Afterward, if the number of elements in the utility list of Px is at least γ, the sum of *iutil* and *rutil* values in the utility list of Px is no less than the minimum utility treshold, and $maxper(Px)$ is

no greater than $maxPer$, extensions of Px will be considered as potential periodic high utility itemsets (based on Theorem 2 and Properties 2 and 3). To explore these extensions, a loop is done where each extension Py of P is merged with Px to produce a new extension Pxy having $|Px|+1$ items. The *Construct* procedure of FHM [30] is applied to combine the utility-lists of P, Px and Py to generate the utility list of Pxy, with some minor modifications. The difference is that periods are calculated for Pxy while building its utility list so that $maxPer(Pxy)$ and $minPer(Pxy)$ can be obtained. Then, the *Search* procedure is recursively invoked with Px and all extensions of the form Pxy to explore the search space and find all periodic high utility itemsets that are transitive extensions of Px.

When the PHM algorithm terminates all periodic high utility itemsets have been output. This can be demonstrated by observing that the algorithm can recursively explore the whole search space and only applies Theorem 2, Properties 2 and 3, to eliminate non periodic high utility itemsets.

Algorithm 2: The *Search* procedure

---

input : P: an itemset, *ExtensionsOfP*: a set of extensions of P, γ, $minUtil$,
    $minAvg$, $minPer$, $maxPer$, $|s|$
output: the set of periodic high-utility itemsets

1 foreach *itemset $Px \in ExtensionsOfP$* **do**
2 $avgperPx \leftarrow |s|/(|Px.utilitylist| + 1)$;
3   **if** $SUM(Pxy.utilitylist.iutils) \geq minUtil \wedge$
    $minAvg \leq avgperPx \leq maxAvg \wedge Px.utilitylist.minp \geq$
    $minPer \wedge Px.utilitylist.maxp \leq maxPer$ **then** output Px;
4   **if** $SUM(Px.utilitylist.iutils)+SUM(Px.utilitylist.rutils) \geq minUtil \wedge$
    $avgperPx \geq \gamma$ and $Px.utilitylist.maxp \leq maxPer$ **then**
5 $ExtensionsOfPx \leftarrow \emptyset$;
6   **foreach** *itemset $Py \in ExtensionsOfP$ such that $y \succ x$* **do**
7 $Pxy \leftarrow Px \cup Py$;
8 $Pxy.utilitylist \leftarrow$ Construct (P, Px, Py);
9 $ExtensionsOfPx \leftarrow ExtensionsOfPx \cup \{Pxy\}$;
10   **end**
11   Search $(Px, ExtensionsOfPx, \gamma, minUtil, minAvg, minPer,$
    $maxPer, |s|)$;
12   **end**
13 end

Optimizations. Note that for a fast implementation of PHM, there are several possible optimizations. Three optimizations proposed in PHM are called (1) Estimated-Utility Co-occurrence Pruning, (2) Estimated Average Periodicity Pruning, and (3) Abandoning List Construction early. Details about these optimizations can be found in the PHM paper [23]. Besides, several other optimizations used in other HUI-Miner or FHM-based algorithms could be integrated into PHM to further enhance its performance such as the memory-buffering technique of ULB-Miner [14].

Implementation and datasets. The original Java implementation of PHM with all optimizations, and datasets are offered in the open-source SPMF data mining library [25] at `http://www.philippe-fournier-viger.com/spmf/`.

4 Irregular High Utility Itemset Mining

The problem of mining periodic high utility itemsets reviewed in the previous section is interesting as it can reveal itemsets that periodically appear in a sequence of quantitative transactions and also have a high importance (e.g. yield a high profit). This section presents a related problem, which is that of discovering *irregular high utility itemsets*. Intuitively, an irregular itemset is an itemset that typically has a long time delay between each of its consecutive occurrences. The problem of irregular itemset mining [45] is defined as follows.

Definition 14 (Regularity) The regularity of an itemset X in a sequence of quantitative transactions s, denoted as $reg(X)$, is the smallest period among the periods in $pr(X, s)$, when the first and last periods are excluded.

Example 14 Consider the sequence s of Table 1 and the itemsets $X = \{b, c\}$ and $Y = \{a, c\}$. The periods of X in s are $pr(X, s) = \{3, 1, 3, 0\}$ while that of Y in s are $pr(Y, s) = \{1, 2, 2, 1, 1\}$. Hence, the regularity of X and Y in s are respectively $reg(X) = 1$ and $reg(Y) = 1$.

Definition 15 (Irregular high utility itemset mining) An itemset X is considered to be an *irregular high utility itemset* (IHUI) if it satisfies two conditions: (1) $reg(X) \geq minReg$, and (2) $u(X) \geq minUtil$, where $minReg$ and $minUtil$ are positive numbers.

The PHM_irregular Algorithm. The problem of discovering irregular high utility itemsets can be solved using the PHM algorithm presented in the previous section by simply setting $minPer = minReg$, $maxPer = \infty$, $minAvg = 0$ and $maxAvg = \infty$. The original implementation of that variation of PHM is called $PHM\_irregular$ and is available in the open-source SPMF pattern mining library [25] at `http://www.philippe-fournier-viger.com/spmf/`.

5 Other Variations and Research Opportunities

The previous section has presented the basic problem of periodic high utility itemset mining and the variation of irregular high utility itemset mining. Some other variations have been proposed such as (1) mining periodic high

utility sequential patterns [9, 11–13, 46, 47, 59, 60] where each transaction is a quantitative sequence, (2) productive-associated periodic high utility itemsets mining [41], where a statistical test is used to filter spurious patterns, and (3) partial periodic high utility itemsets [63], which utilizes a different measure of periodicity.

There are several possibilities for future work. A few of them are:

- Designing faster and more memory-efficient algorithms for periodic high utility itemset mining by taking advantage of the numerous work on high utility itemset mining [27], or other optimizations in periodic itemset mining such as approximate periodicity calculations [5].
- Proposing new problems by drawing inspiration from variations of the high utility itemset mining problem such as considering multiple minimum utility thresholds [64], using the average utility function [68], discovering the top-k high itemsets having the highest utility [15, 66], mining high utility itemsets in a stream or incrementally updated data [77, 78], discovering on-shelf high utility itemsets [37], finding a summary of all high utility itemsets [29], and considering negative utility values [17].
- Using different measures of periodicity or other measures related to time such as the stability [28, 33].
- Applying periodic high utility itemset mining in new applications such as for smart homes [53], intelligent systems [43], location prediction [71] and sequence prediction [38].
- Designing measures to find periodic high utility itemsets common to multiple sequences similarly to studies on mining periodic patterns common to multiple sequences [21, 31],
- Integrating various correlation measures in the mining process to filter spurious patterns besides the one that was used in productive-associated periodic high utility itemset mining [41], such as the bond [22, 36, 57, 76], affinity [2], all-confidence [57, 70], coherence and mean [7, 67].
- Combining the concept of high utility periodic patterns with other concepts such as episode patterns [6, 34, 35], subgraphs [18, 19, 39, 42], sequential patterns [20, 26, 69, 74, 75], trajectory patterns [80], and periodic patterns with gap constraints [72, 73], and clustering [8, 10, 51].

6 Conclusion

This chapter has presented an overview of how to discover periodic and irregular high utility itemsets in a sequence of quantitative transactions (also called a transaction database). The problems have been described and two algorithms have been explained, namely PHM and PHM_irregular. There are several possible research opportunities on this topic. Some of them have been listed in this chapter.

References

[1] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of 20th International Conference on Very Large Data Bases, pp. 487–499 (1994)

[2] Ahmed, C.F., Tanbeer, S.K., Jeong, B., Choi, H.: A framework for mining interesting high utility patterns with a strong frequency affinity. Inf. Sci. **181**(21), 4878–4894 (2011). DOI 10.1016/j.ins.2011.05.012

[3] Amphawan, K., Lenca, P., Surarerks, A.: Mining top-K periodic-frequent pattern from transactional databases without support threshold. In: Proc. Third International Conference on Advanced in Information Technology, pp. 18–29 (2009)

[4] Amphawan, K., Lenca, P., Surarerks, A.: Mining top-K periodic-frequent pattern from transactional databases without support threshold. In: Proc. of Third International Conference on Advances in Information Technology, pp. 18–29 (2009). DOI 10.1007/978-3-642-10392-6_3

[5] Amphawan, K., Surarerks, A., Lenca, P.: Mining periodic-frequent itemsets with approximate periodicity using interval transaction-ids list tree. In: 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 245–248. IEEE (2010)

[6] Ao, X., Shi, H., Wang, J., Zuo, L., Li, H., He, Q.: Large-scale frequent episode mining from complex event sequences with hierarchies. ACM Transactions on Intelligent Systems and Technology (TIST) **10**(4), 1–26 (2019)

[7] Barsky, M., Kim, S., Weninger, T., Han, J.: Mining flipping correlations from large datasets with taxonomies. Proc. VLDB Endow. **5**, 370–381 (2011)

[8] Dinh, D.T., Huynh, V.N.: k-pbc: an improved cluster center initialization for categorical data clustering. Applied Intelligence pp. 1–23 (2020)

[9] Dinh, D.T., Huynh, V.N., Le, B., Fournier-Viger, P., Huynh, U., Nguyen, Q.M.: A survey of privacy preserving utility mining. In: High-Utility Pattern Mining, pp. 207–232 (2019)

[10] Dinh, D.T., Huynh, V.N., Songsak, S.: Clustering mixed numerical and categorical data with missing values. Information Sciences (2021)

[11] Dinh, D.T., Le, B., Fournier-Viger, P., Huynh, V.N.: An efficient algorithm for mining periodic high-utility sequential patterns. Applied Intelligence **48**(12), 4694–4714 (2018)

[12] Dinh, T., Huynh, V.N., Le, B.: Mining periodic high utility sequential patterns. In: Proc. 2017 Internal Conference on Intelligent Information and Database Systems, pp. 545–555. Springer (2017)

[13] Dinh, T., Quang, M.N., Le, B.: A novel approach for hiding high utility sequential patterns. In: Proceedings of the 6th International Symposium on Information and Communication Technology, pp. 121–128 (2015)

[14] Duong, Q.H., Fournier-Viger, P., Ramampiaro, H., Nørvåg, K., Dam, T.L.: Efficient high utility itemset mining using buffered utility-lists. Applied Intelligence **48**(7), 1859–1877 (2018)

[15] Duong, Q.H., Liao, B., Fournier-Viger, P., Dam, T.L.: An efficient algorithm for mining the top-k high utility itemsets, using novel threshold raising and pruning strategies. Knowledge-Based Systems **104**, 106–122 (2016)

[16] Feng, Z., Zhu, Y.: A survey on trajectory data mining: Techniques and applications. IEEE Access **4**, 2056–2067 (2016)

[17] Fournier-Viger, P.: Fhn: efficient mining of high-utility itemsets with negative unit profits. In: International Conference on Advanced Data Mining and Applications, pp. 16–29. Springer (2014)

[18] Fournier-Viger, P., Cheng, C., Lin, J.C.W., Yun, U., Kiran, R.U.: Tkg: Efficient mining of top-k frequent subgraphs. In: Proc. 7th International Conference on Big Data Analytics, pp. 209–226. Springer (2019)

[19] Fournier-Viger, P., He, G., Cheng, C., Li, J., Zhou, M., Lin, J.C.W., Yun, U.: A survey of pattern mining in dynamic graphs. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery p. e1372 (2020)

[20] Fournier-Viger, P., Li, J., Lin, J.C.W., Truong, T.: Discovering low-cost high utility patterns. Data Science and Pattern Recognition **4**(2), 50–64 (2020)

[21] Fournier-Viger, P., Li, Z., Lin, J.C., Kiran, R.U., Fujita, H.: Discovering periodic patterns common to multiple sequences. In: Proc. 20th International Conference on Big Data Analytics and Knowledge Discovery, pp. 231–246 (2018). DOI 10.1007/978-3-319-98539-8_18

[22] Fournier-Viger, P., Lin, J.C., Dinh, T., Le, H.B.: Mining correlated high-utility itemsets using the bond measure. In: Proc. 11th Intern. Conf. on Hybrid Artificial Intelligent Systems, pp. 53–65 (2016). DOI 10.1007/978-3-319-32034-2_5

[23] Fournier-Viger, P., Lin, J.C., Duong, Q., Dam, T.: PHM: mining periodic high-utility itemsets. In: P. Perner (ed.) Proc. 16th Industrial Conference, ICDM 2016, New York, NY, USA, July 13-17, pp. 64–79. Springer (2016). DOI 10.1007/978-3-319-41561-1_6

[24] Fournier-Viger, P., Lin, J.C., Vo, B., Truong, T.C., Zhang, J., Le, H.B.: A survey of itemset mining. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **7**(4) (2017). DOI 10.1002/widm.1207

[25] Fournier-Viger, P., Lin, J.C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., 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 (2016)

[26] Fournier-Viger, P., Lin, J.C.W., Kiran, U.R., Koh, Y.S.: A survey of sequential pattern mining. Data Science and Pattern Recognition **1**(1), 54–77 (2017)

[27] Fournier-Viger, P., Lin, J.C.W., Truong-Chi, T., Nkambou, R.: A survey of high utility itemset mining. In: High-Utility Pattern Mining, pp. 1–45. Springer (2019)

[28] Fournier-Viger, P., Wang, Y., Yang, P., Lin, J.C.W., Unil, Y.: A survey of sequential pattern mining. Applied Intelligence (2021)

[29] Fournier-Viger, P., Wu, C.W., Tseng, V.S.: Novel concise representations of high utility itemsets using generator patterns. In: International Conference on Advanced Data Mining and Applications, pp. 30–43. Springer (2014)

[30] Fournier-Viger, P., Wu, C.W., Zida, S., Tseng, V.S.: Fhm: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In: International symposium on methodologies for intelligent systems, pp. 83–92. Springer (2014)

[31] Fournier-Viger, P., Yang, P., Li, Z., Lin, J.C.W., Kiran, R.U.: Discovering rare correlated periodic patterns in multiple sequences. Data Knowl. Eng. (126), 101,733 (2020). DOI 10.1016/j.datak.2019.101733

[32] Fournier-Viger, P., Yang, P., Lin, J.C.W., Duong, Q.H., Dam, T., Sevcik, L., Uhrin, D., Voznak, M.: Discovering periodic itemsets using novel periodicity measures. Advances in Electrical and Electronic Engineering **17**(1), 33–44 (2019). DOI 10.15598/aeee.v17i1.3185

[33] Fournier-Viger, P., Yang, P., Lin, J.C.W., Kiran, R.U.: Discovering stable periodic-frequent patterns in transactional data. In: Proc. 32nd Intern. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 230–244. Springer (2019)

[34] Fournier-Viger, P., Yang, P., Lin, J.C.W., Yun, U.: Hue-span: Fast high utility episode mining. In: Proc. 14th Intern. Conf. on Advanced Data Mining and Applications, pp. 169–184. Springer (2019)

[35] Fournier-Viger, P., Yang, Y., Yang, P., Lin, J.C.W., Yun, U.: Tke: Mining top-k frequent episodes. In: Proc. 33rd Intern. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer (2020)

[36] Fournier-Viger, P., Zhang, Y., Lin, J.C.W., Dinh, D.T., Bac Le, H.: Mining correlated high-utility itemsets using various measures. Logic Journal of the IGPL **28**(1), 19–32 (2020)

[37] Fournier-Viger, P., Zida, S.: 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, pp. 857–864 (2015)

[38] Gueniche, T., Fournier-Viger, P., Raman, R., Tseng, V.S.: Cpt+: Decreasing the time/space complexity of the compact prediction tree. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 625–636. Springer (2015)

[39] Halder, S., Samiullah, M., Lee, Y.K.: Supergraph based periodic pattern mining in dynamic social networks. Expert Systems with Applications **72**, 430–442 (2017)

[40] Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. **8**(1), 53–87 (2004). DOI 10.1023/B:DAMI.0000005258.31418.83

[41] Ismail, W., Hassan, M.M., Fortino, G.: Productive-associated periodic high-utility itemsets mining. In: 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), pp. 637–642. IEEE (2017)

[42] Jiang, C., Coenen, F., Zito, M.: A survey of frequent subgraph mining algorithms. Knowledge Engineering Review **28**, 75–105 (2013)

[43] Kim, H., Yun, U., Vo, B., Lin, J.C.W., Pedrycz, W.: Periodicity-oriented data analytics on time-series data for intelligence system. IEEE Systems Journal (2020)

[44] Kumar, V., Kumari, V.: Incremental mining for regular frequent patterns in vertical format. International Journal of Engineering and Technology **5**(2), 1506–1511 (2013)

[45] Laoviboon, S., Amphawan, K.: Mining high-utility irregular itemsets. In: High-Utility Pattern Mining, pp. 175–205. Springer (2019)

[46] Le, B., Dinh, D.T., Huynh, V.N., Nguyen, Q.M., Fournier-Viger, P.: An efficient algorithm for hiding high utility sequential patterns. International Journal of Approximate Reasoning **95**, 77–92 (2018)

[47] Le, B., Huynh, U., Dinh, D.T.: A pure array structure and parallel strategy for high-utility sequential pattern mining. Expert Systems with Applications **104**, 107–120 (2018)

[48] Lin, C.W., Hong, T.P., Lu, W.H.: An effective tree structure for mining high utility itemsets. Expert Systems with Applications **38**(6), 7419–7424 (2011)

[49] Liu, M., Qu, J.: Mining high utility itemsets without candidate generation. In: Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 55–64 (2012)

[50] Liu, Y., Liao, W.k., Choudhary, A.: A two-phase algorithm for fast discovery of high utility itemsets. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 689–695. Springer (2005)

[51] Lu, N.V., Vuong, T.N., Dinh, D.T.: Combining correlation-based feature and machine learning for sensory evaluation of saigon beer. International Journal of Knowledge and Systems Science (IJKSS) **11**(2), 71–85 (2020)

[52] Luna, J.M., Fournier-Viger, P., Ventura, S.: Frequent itemset mining: A 25 years review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **9**(6), e1329 (2019)

[53] Mukhlash, I., Yuanda, D., Iqbal, M.: Mining fuzzy time interval periodic patterns in smart home data. International Journal of Electrical and Computer Engineering **8**(5), 3374 (2018)

[54] Nawaz, S., Fournier-Viger, P., Shojaee, A., Fujita, H.: Using artificial intelligence techniques for covid-19 genome analysis. Applied Intelligence (2021)

[55] Nofong, V.M.: Discovering productive periodic frequent patterns in transactional databases. Annals of Data Science **3**(3), 235–249 (2016)

[56] Nofong, V.M.: Fast and memory efficient mining of periodic frequent patterns. In: Proceedings of the 10th Asian Conference on Intelligent Information and Database Systems, pp. 223–232. Springer (2018)

[57] Omiecinski, E.: Alternative interest measures for mining associations in databases. IEEE Trans. Knowl. Data Eng. **15**(1), 57–69 (2003). DOI 10.1109/TKDE.2003.1161582

[58] Peng, A.Y., Koh, Y.S., Riddle, P.: mhuiminer: A fast high utility itemset mining algorithm for sparse datasets. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 196–207. Springer (2017)

[59] Quang, M.N., Dinh, T., Huynh, U., Le, B.: MHHUSP: An integrated algorithm for mining and Hiding High Utility Sequential Patterns. In: Proceedings of the 8th International Conference on Knowledge and Systems Engineering, pp. 13–18 (2016)

[60] Quang, M.N., Huynh, U., Dinh, T., Le, N.H., Le, B.: An Approach to Decrease Execution Time and Difference for Hiding High Utility Sequential Patterns. In: Proceedings of the 5th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making, pp. 435–446 (2016)

[61] Rashid, M.M., Gondal, I., Kamruzzaman, J.: Regularly frequent patterns mining from sensor data stream. In: Proc. 20th International Conference on Neural Information Processing, pp. 417–424. Springer (2013)

[62] Rashid, M.M., Karim, M.R., Jeong, B.S., Choi, H.J.: Efficient mining regularly frequent patterns in transactional databases. In: Proc. 17th International Conference on Database Systems for Advanced Applications, pp. 258–271. Springer (2012)

[63] Reddy, T.Y., Kiran, R.U., Toyoda, M., Reddy, P.K., Kitsuregawa, M.: Discovering partial periodic high utility itemsets in temporal databases. In: International Conference on Database and Expert Systems Applications, pp. 351–361. Springer (2019)

[64] Ryang, H., Yun, U., Ryu, K.H.: Discovering high utility itemsets with multiple minimum supports. Intelligent data analysis **18**(6), 1027–1047 (2014)

[65] Shekhar, S., Evans, M.R., Kang, J.M., Mohan, P.: Identifying patterns in spatial information: A survey of methods. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1**(3), 193–214 (2011)

[66] Song, W., Liu, L., Huang, C.: Tku-ce: Cross-entropy method for mining top-k high utility itemsets. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 846–857. Springer (2020)

[67] Soulet, A., Raïssi, C., Plantevit, M., Crémilleux, B.: Mining dominant patterns in the sky. In: Proc. 11th IEEE International Conference on Data Mining, pp. 655–664 (2011). DOI 10.1109/ICDM.2011.100

[68] Truong, T., Duong, H., Le, B., Fournier-Viger, P.: Efficient vertical mining of high average-utility itemsets based on novel upper-bounds. IEEE Transactions on Knowledge and Data Engineering **31**(2), 301–314 (2018)

[69] Truong, T., Tran, A., Duong, H., Le, B., Fournier-Viger, P.: Ehusm: Mining high utility sequences with a pessimistic utility model. Data Science and Pattern Recognition **4**(2), 65–83 (2020)

[70] Venkatesh, J.N., Kiran, R.U., Reddy, P.K., Kitsuregawa, M.: Discovering periodic-frequent patterns in transactional databases using all-confidence and periodic-all-confidence. In: Proc. 27th International Conference on Database and Expert Systems Applications Part I, pp. 55–70 (2016). DOI 10.1007/978-3-319-44403-1_4

[71] Wong, M.H., Tseng, V.S., Tseng, J.C., Liu, S.W., Tsai, C.H.: Long-term user location prediction using deep learning and periodic pattern mining. In: Proc. 12th Conference on Advanced Data Mining and Applications, pp. 582–594. Springer (2017)

[72] Wu, Y., Shen, C., Jiang, H., Wu, X.: Strict pattern matching under non-overlapping condition. Science China Information Sciences **60**(1), 1–16 (2017)

[73] Wu, Y., Tong, Y., Zhu, X., Wu, X.: Nosep: Nonoverlapping sequence pattern mining with gap constraints. IEEE transactions on cybernetics **48**(10), 2809–2822 (2017)

[74] Wu, Y., Wang, L., Ren, J., Ding, W., Wu, X.: Mining sequential patterns with periodic wildcard gaps. Applied intelligence **41**(1), 99–116 (2014)

[75] Wu, Y., Zhu, C., Li, Y., Guo, L., Wu, X.: Netncsp: Nonoverlapping closed sequential pattern mining. Knowledge-based systems **196**, 105,812 (2020)

[76] Younes, N.B., Hamrouni, T., Yahia, S.B.: Bridging conjunctive and disjunctive search spaces for mining a new concise and exact representation of correlated patterns. In: Proc. 13th International Conference on Discovery Science, pp. 189–204 (2010). DOI 10.1007/978-3-642-16184-1_14

[77] Yun, U., Kim, D., Yoon, E., Fujita, H.: Damped window based high average utility pattern mining over data streams. Knowledge-Based Systems **144**, 188–205 (2018)

[78] Yun, U., Ryang, H., Lee, G., Fujita, H.: An efficient algorithm for mining high utility patterns from incremental databases with one database scan. Knowledge-Based Systems **124**, 188–206 (2017)

[79] Zaki, M.J.: Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. **12**(3), 372–390 (2000). DOI 10.1109/69.846291

[80] Zhang, D., Lee, K., Lee, I.: Hierarchical trajectory clustering for spatio-temporal periodic pattern mining. Expert Systems with Applications **92**, 1–11 (2018)

[81] Zida, S., Fournier-Viger, P., Lin, J.C.W., Wu, C.W., Tseng, V.S.: Efim: a fast and memory efficient algorithm for high-utility itemset mining. Knowledge and Information Systems **51**(2), 595–625 (2017)