Ann. Data. Sci. (2016) 3(3):235–249
DOI 10.1007/s40745-016-0078-8

Discovering Productive Periodic Frequent Patterns in Transactional Databases

Vincent Mwintieru Nofong[1]

Received: 9 December 2015 / Revised: 10 April 2016 / Accepted: 12 April 2016 /
Published online: 23 April 2016

Abstract Periodic frequent pattern mining is an important data mining task for various decision making. However, it often presents a large number of periodic frequent patterns, most of which are not useful as their periodicities are due to random occurrence of uncorrelated items. Such periodic frequent patterns would most often be detrimental in decision making where correlations between the items of periodic frequent patterns are vital. To enable mine the periodic frequent patterns with correlated items, we employ a correlation test on periodic frequent patterns and introduce the productive periodic frequent patterns as the set of periodic frequent patterns with correlated items. We finally develop the productive periodic frequent pattern (PPFP) framework for mining our introduced productive periodic frequent patterns. PPFP is efficient and the productiveness measure removes the periodic frequent patterns with uncorrelated items.

Keywords Frequent patterns · Periodic frequent patterns · Productiveness measure

1 Introduction

Periodicity detection in databases has been studied in two distinct areas based on the data types: (i) time series datasets [2,9], and (ii) transactional datasets [4,10,12,13]. In time-series datasets, periodicity is detected under the names *segment* and/or *symbol periodicity* [2,9] while in transactional datasets it is detected under the names *periodic frequent patterns* (PFPs) [4,12,13] or *regular frequent patterns* (RFPs)

✉ Vincent Mwintieru Nofong
vincent.nofong@mymail.unisa.edu.au

[1] School of Information Technology and Mathematical Science, University of South Australia, Mawson Lakes Campus, Adelaide, Australia

[10,11]. Though transactional datasets can always be accumulated as time-series datasets, in this work, we focus on periodicity detection in purely transactional datasets.

Mining periodic frequent patterns in transactional datasets was proposed by Tanbeer et al. [13] with the aim of identifying frequent patterns that occur at regular intervals in databases. Tanbeer et al. [13] proposed the concept of periodic frequent patterns since the interestingness measures (such as support and closure) used in frequent pattern mining do not provide information on the occurrence shapes of patterns in databases. Additionally, in data analysis such as the analysis of customer behaviour or trends in crimes, though frequent pattern mining will reveal frequently occurring patterns, retail management or crime analysts may not just be interested in frequently occurring purchases or crimes, but the periodic nature (shapes) of purchases or crime for decision making.

To address this issue, Tanbeer et al. in [13] introduced the concept of periodicity for patterns in transaction-like databases where they refer to a patterns' periodicity as its maximal occurring period. This concept has since been used in mining periodic frequent pattern in transaction-like databases in works such as [4–6,12].

Recently, Rashid et al. in [10] also proposed a different periodicity measure for patterns in transaction-like databases under the name "regularity" where they refer to a patterns' regularity (periodicity) as the variance among its periods. Their concept has also been used in mining periodic frequent patterns in transactional datasets under the name *regular frequent patterns* in works such as [8,11].

Mining periodic frequent patterns in transactional databases is however faced with several challenges. For instance, the periodicity measure in [13], which is susceptible to noise, might often report the noised maximal period of a pattern as its regular period. Additionally, both methods in [13] and [10] often report periodic (regular) frequent patterns with totally distinct periods for decision making. This makes it difficult for users who are interested in patterns with similar regular periods identifying the desired periodic (regular) frequent patterns.

It is also worth noting that some of these reported periodic (regular) frequent patterns often are periodic due to random occurrence of items without inherent item relationships. Using such periodic frequent patterns without inherent item relationships (that is, false periodic frequent patterns) could be very detrimental in decision making.

Our work in this paper aims at solving these mentioned challenges in the discovery of periodic frequent patterns in transactional databases as follows. We firstly introduce our periodicity measure and subsequently restrict our periodicity to a range to ensure only periodic frequent patterns with similar periods are reported. We further employ a productiveness measure to ensure periodic frequent patterns due to random chance without inherent item relationships are eliminated.

The contributions of this work to the discovery of periodic frequent patterns are as follows. We firstly present a periodicity measure for mining the set of periodic frequent patterns with similar periods and introduce the productive periodic frequent pattern set. We also propose and develop PPFP, an efficient productive periodic frequent pattern mining framework.

2 Preliminaries

The associated notations for periodic frequent pattern mining in transactional databases can be given as follows.

Let I = $\langle i_1, i_2, \ldots, i_n\rangle$ be a set of literals, called items. Then, a transaction is a nonempty set of items. A pattern S is a set of transactions satisfying some conditions of measures like frequency. A pattern is of length-k if it has k items, for instance, $S = \{a, b, c\}$ is a length-3 pattern.

Given a transactional database of n transactions, $\mathbf{D} =< T_1, T_2, T_3, \ldots, T_n >$, where each T_m in $\mathbf{D}$ is identified by m called *TID*, the *cover* of a pattern S in $\mathbf{D}$, $cov_{\mathbf{D}}(S)$, is the set of *TIDs* of transactions that contain S. That is,

$$cov_{\mathbf{D}}(S) = \{m : T_m \in \mathbf{D} \wedge S \subseteq T_m\} \tag{1}$$

The *support* of a pattern S in $\mathbf{D}$, $sup_{\mathbf{D}}(S)$, is defined as,

$$sup_{\mathbf{D}}(S) = \frac{|cov_{\mathbf{D}}(S)|}{|\mathbf{D}|} \tag{2}$$

where $|cov_{\mathbf{D}}(S)|$ is called the *support count* of S in $\mathbf{D}$.

A pattern S in $\mathbf{D}$ is said to be frequent if its support in $\mathbf{D}$ is larger than or equal to a user specified minimum support (ε). A pattern S in $\mathbf{D}$ is said to be productive in $\mathbf{D}$ if [14]: for all S_1, S_2 (such that, $S_1 \subset S \wedge S_2 \subset S \wedge S_1 \cup S_2 = S \wedge S_1 \cap S_2 = \emptyset$), $sup_{\mathbf{D}}(S) > sup_{\mathbf{D}}(S_1) sup_{\mathbf{D}}(S_2)$.

Let S be a pattern in a transactional database $\mathbf{D}$ and $cov_{\mathbf{D}}(S)$ be its coverset in $\mathbf{D}$. We use the notation $e.cov_{\mathbf{D}}(S)$ to indicate the extension of $cov_{\mathbf{D}}(S)$ by inserting a starting time 0 and the last time n to $cov_{\mathbf{D}}(S)$. That is,

$$e.cov_{\mathbf{D}}(S) = \{0 \cup cov_{\mathbf{D}}(S) \cup n\} \tag{3}$$

where $n = |\mathbf{D}|$. However, n will be duplicated if it is already in $cov_{\mathbf{D}}(S)$. For example, given $|\mathbf{D}| = 6$ and $cov_{\mathbf{D}}(S) = \{1, 4, 6\}$, then, $e.cov_{\mathbf{D}}(S) = \{0\} \cup \{1, 4, 6\} \cup \{6\} = \{0, 1, 4, 6, 6\}$.

Let $(m_j, m_{j+1}) \in e.cov_{\mathbf{D}}(S)$ be two consecutive occurrence times (TIDs) of S in $\mathbf{D}$, then $p_j^S = m_{j+1} - m_j$ is the jth period of S in $\mathbf{D}$. The set of all periods of S obtained from its extended cover, $e.cov_{\mathbf{D}}(S)$, is denoted as:

$$P^S = \left\{p_1^S, \cdots, p_r^S\right\} \tag{4}$$

where $r = |e.cov_{\mathbf{D}}(S)| - 1$.

For example, given $e.cov_{\mathbf{D}}(S) = \{0, 1, 4, 6, 6\}$, then $p_1^S = (1 - 0) = 1$, $p_2^S = (4 - 1) = 3$, $p_3^S = (6 - 4) = 2$, $p_4^S = (6 - 6) = 0$, and, $P^S = \{1, 3, 2, 0\}$.

To mine the set of patterns with periodic occurrence shapes in transactional datasets for decision making, Tanbeer et al. in [13] proposed a periodicity measure on patterns as follows.

Definition 1 [13] Given a database **D**, a pattern S and its set of periods P^S in **D**, the periodicity of S, $Per(S)$, is defined as, $Per(S) = \max\{p|p \in P^S\}$.

For a pattern S and its set of periods P^S, Definition 1 reports the maximal occurring period (maximal time interval between any consecutive occurrence times) of S as its periodic interval.

With the periodicity measure in Definition 1, Tanbeer et al. [13] defined a periodic frequent pattern as follows.

Definition 2 [13] Given a dataset D, minimum support threshold, ε and maximum periodicity threshold, $maxPer$, a frequent pattern S is periodic if $Per(S) \leq maxPer$.

This periodicity measure and periodic frequent pattern proposed by Tanbeer et al. [13] have been used mining periodic frequent patterns in transaction-like datasets in works such as [4,5,7,12].

Recently, Rashid et al. [10] argued that the periodicity measure in [13] is inappropriate as it returns the maximum period for which a pattern does not appear in a dataset as its periodicity. To address this issue, Rashid et al. [10] define the periodicity of a pattern under the name patterns' *regularity* as follows.

Definition 3 [10] Given a database **D**, a pattern S and its set of periods P^S in **D**, the regularity of S, $Reg(S)$, is defined as $Reg(S) = var(P^S)$, where $var(P^S)$ is the variance of P^S.

For a pattern S and its set of periods P^S, Definition 3 reports the variance among all periods of S as its periodic interval.

With the regularity (periodicity) measure in Definition 3, Rashid et al. [10] define a regular (periodic) frequent pattern as follows.

Definition 4 [10] Given a dataset D, minimum support threshold, ε and maximum regularity threshold, $maxReg$, a frequent pattern S is regular if $Reg(S) \leq maxReg$.

The concept of mining regular (periodic) frequent patterns proposed by Rashid et al. [10] has also been useful in mining regular frequent patterns in works such as [8,11].

Though Definitions 2 and 4 have been useful in mining periodic (regular) frequent patterns in transaction-like datasets for decision making, they are faced with the following challenges:

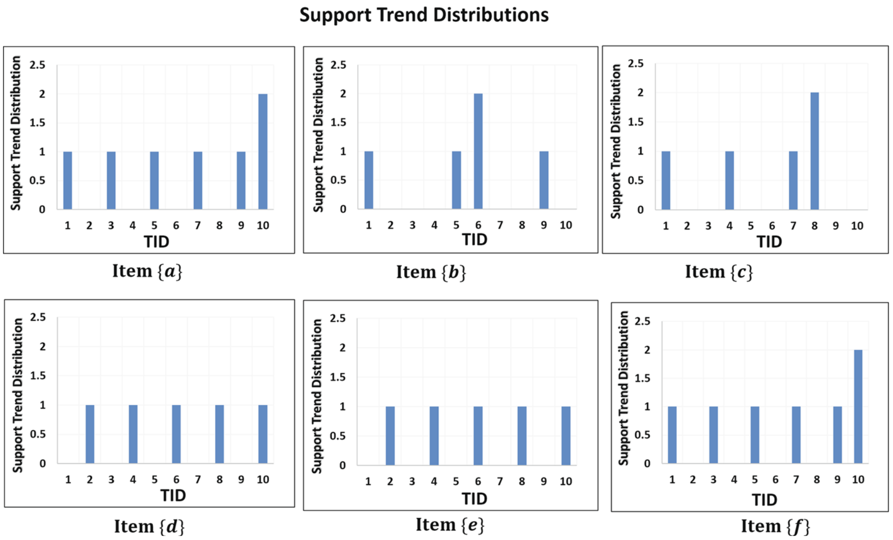
1. They often report periodic (regular) frequent patterns with total distinct periods. For instance, in Table 2 (which shows the occurrence properties of the length-1 transactions from Table 1), given $maxPer = 3$, works such as [4,5,7,13] will report items $\{a\}$, $\{c\}$, $\{d\}$, $\{e\}$ and $\{f\}$ as periodic. Similarly, given $maxReg = 0.8$, [8,10,11] will also report items $\{a\}$, $\{c\}$, $\{d\}$, $\{e\}$ and $\{f\}$ as periodic (regular). However, from Fig. 1, considering the support trends with time, $\{a\}$ and $\{f\}$ have similar occurrence periods which are totally distinct from those of $\{d\}$ and $\{e\}$, and that of $\{c\}$. In decision making such as analysis of associated purchases where periodic frequent patterns with similar periods are required, users of [4,5,7,8,10–13] will have to manually select from the reported periodic (regular) frequent patterns those with similar periods for decision making.

Table 1 Sample database

TID	Transaction
1	$\{a, b, c, f\}$
2	$\{d, e\}$
3	$\{a, f\}$
4	$\{c, d, e\}$
5	$\{a, b, f\}$
6	$\{b, d, e\}$
7	$\{a, c, f\}$
8	$\{c, d, e\}$
9	$\{a, b, f\}$
10	$\{a, d, e, f\}$

Table 2 Periodic intervals of length-1 items in Table 1

Item	TID set	Period, P	$\bar{x}(P)$	$std(P)$	$var(P)$	$\max(P)$
a	{1, 3, 5, 7, 9, 10}	{1, 2, 2, 2, 2, 1, 0}	1.429	0.728	0.530	2
b	{1, 5, 6, 9}	{1, 4, 1, 3, 1}	2.0	0.943	0.889	4
c	{1, 4, 7, 8}	{1, 3, 3, 1, 2}	2.0	0.894	0.799	3
d	{2, 4, 6, 8, 10}	{2, 2, 2, 2, 2, 0}	1.667	0.745	0.555	2
e	{2, 4, 6, 8, 10}	{2, 2, 2, 2, 2, 0}	1.667	0.745	0.555	2
f	{1, 3, 5, 7, 9, 10}	{1, 2, 2, 2, 2, 1, 0}	1.429	0.728	0.530	2



Item $\{a\}$ Item $\{b\}$ Item $\{c\}$

Item $\{d\}$ Item $\{e\}$ Item $\{f\}$

Fig. 1 Support trend distributions of length-1 items in Table 1

2. In noisy datasets, where noise delays the occurrence of a pattern in datasets, works such as [4,5,7,12,13] which mine periodic frequent patterns based on the maximal period, will often report the maximal noisy period as a pattern's regular period.
3. Some reported periodic (regular) frequent patterns may be detrimental in decision making as they might be periodic (regular) by random chance and not due to inherent item relationships. Reporting periodic (regular) frequent patterns which do not encode inherent item relations for decision making such as crime or disease control could be detrimental as such decisions will be made with false periodic (regular) frequent patterns.

3 Definitions and Problem Statement

As mentioned earlier, the periodic frequent patterns often reported in Definitions 2 and 4, might always have totally distinct periods, or, might be periodic due to random chance. To avoid these situations, we begin by defining the periodicity of a pattern as follows.

Definition 5 Given a database **D**, a pattern S and its set of periods P^S in **D**, the periodicity of S, $Prd(S)$, is defined as $Prd(S) = \bar{x}(P^S)$, where $\bar{x}(P^S)$ is the mean of P^S.

For a pattern S and its set of periods P^S, Definition 5 will report the mean among all periods of S as its periodic interval.

Though Definition 5 will report the mean period as the periodic occurrence interval of a pattern, we cannot directly employ Definition 5 in mining periodic frequent patterns with similar periods. This is because, with Definition 5, periodic frequent patterns with total distinct occurrence periods may still be reported. For instance, given $P^X = \{1, 4, 15, 30, 50\}$ and $P^Y = \{20, 20, 20, 20, 20\}$, though patterns X and Y have totally distinct occurrence periods, $Prd(X) = Prd(Y) = 20$. Hence using only the periodicity of patterns proposed in Definition 5 will thus not solve the problem of reporting periodic frequent patterns with totally distinct periods.

To enable mine the set of periodic frequent patterns with similar periods in databases, we restrict our periodicity to a range and formally define a periodic frequent pattern as follows.

Definition 6 Given a database **D**, minimum support threshold ε, periodicity threshold p, difference factor p_1, a pattern S and P^S, S is a periodic frequent pattern if $sup_{\mathbf{D}}(S) \geq \varepsilon$, $(p - p_1) \leq Prd(S) - std(P^S)$ and $Prd(S) + std(P^S) \leq (p + p_1)$.

where $std(P^S)$ in Definition 6 is the standard deviation in P^S, while p and p_1 are the user desired periodicity threshold, and difference factor respectively. We use the range $p \pm p_1$ in Definition 6 to ensure only periodic frequent patterns with similar range of regular periods are reported. For example, in Table 2 of our running example, if $p = 1.4$ and $p_1 = 0.8$, unlike Definitions 2 and 4, only $\{a\}$ and $\{f\}$ which have similar regular periods as shown in Fig. 1 will be reported as being periodic.

With Definition 6 only periodic frequent patterns with similar regular periods will be reported. However, some might be periodic due to random occurrence of items without inherent item relationships. Such periodic frequent patterns not encoding inherent item relationships will often be detrimental in decision making where inherent item relationships are vital.

To enable detect and report only periodic frequent patterns with inherent item relationships, we test for positive correlations among items of a periodic frequent pattern and refer to periodic frequent patterns with inherent item relationships as productive. Formally we define a productive periodic frequent pattern as follows.

Definition 7 A periodic frequent pattern, S in $\mathbf{D}$, is a productive periodic frequent pattern if, for all S_1, S_2 such that, $(S_1 \subset S)$, $(S_2 \subset S)$, $(S_1 \cup S_2 = S)$, and $(S_1 \cap S_2 = \emptyset)$, then, $\left(\frac{|D|-Prd(S)}{Prd(S)\cdot|D|}\right) > \left(\frac{|D|-Prd(S_1)}{Prd(S_1)\cdot|D|} \times \frac{|D|-Prd(S_2)}{Prd(S_2)\cdot|D|}\right)$.

Our productiveness test $\left(\frac{|D|-Prd(S)}{Prd(S)\cdot|D|}\right) > \left(\frac{|D|-Prd(S_1)}{Prd(S_1)\cdot|D|} \times \frac{|D|-Prd(S_2)}{Prd(S_2)\cdot|D|}\right)$, in Definition 7, is same as the proposed productivity test in [14] as follows:

For any pattern S_n, $\frac{|D|-Prd(S_n)}{Prd(S_n)\cdot|D|}$ can be re-written as $\frac{|D|-Prd(S_n)}{Prd(S_n)} \times \frac{1}{|D|}$ where $\frac{|D|-Prd(S_n)}{Prd(S_n)} = |cov_D(S_n)|$. As such, $\frac{|D|-Prd(S_n)}{Prd(S_n)\cdot|D|}$ can thus be expressed as $\frac{|cov_D(S_n)|}{|D|} = sup_D(S_n)$. Hence, our productiveness test can be expressed as that proposed in [14] as: $\left(\frac{|D|-Prd(S)}{Prd(S)\cdot|D|}\right) > \left(\frac{|D|-Prd(S_1)}{Prd(S_1)\cdot|D|} \times \frac{|D|-Prd(S_2)}{Prd(S_2)\cdot|D|}\right) = sup_D(S) > sup_D(S_1) \times sup_D(S_2)$.

Definition 7 requires a periodic frequent pattern, S in $\mathbf{D}$ is productive if and only if every subset that can be formed from it is productive (that is, formed by items with inherent relations) in $\mathbf{D}$. This productiveness measure for every subset is to ensure all items of a periodic frequent pattern are correlated and not due to random occurrences. Since the supersets of a non-productive pattern will always contain the non-productive pattern, we use the productiveness of patterns as one of our pruning criteria to avoid reporting periodic frequent patterns with non-productive subsets.

In the rest of this work, we represent the set of productive periodic frequent patterns discovered by Definition 7 in a database $\mathbf{D}$ as $\mathbf{Per_D}$.

4 Mining Productive Periodic Frequent Patterns

To mine the productive periodic frequent patterns, we propose *PPFP*, an efficient productive periodic frequent pattern mining algorithm shown in Algorithm 1. *PPFP* employs the Apriori-like candidate generation technique in [1]. However, it stores the transaction *IDs* for each item to avoid repeated dataset scans and for quick execution. For a given database and minimum support, *PPFP* employs two major steps in mining the productive periodic frequent patterns:

1. Finding the set of frequent length-1 items from the input database, and
2. Mining the set of productive periodic frequent patterns from the frequent length-1 items.

We discuss the functions of these steps in *PPFP* as follows.

4.1 Finding Frequent Length-1 Items

This step (Lines 1 to 16 of Algorithm 1) finds the set of frequent length-1 items and their coversets in $\mathbf{D}$ with regards to the minimum support (ε) as follows.

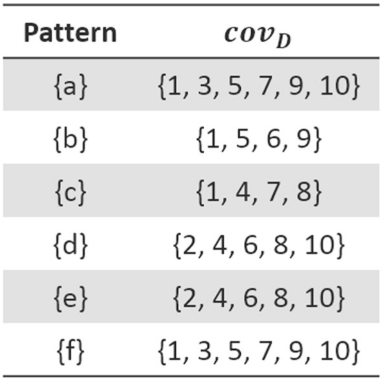
```
Algorithm 1: PPFP($D$, $\varepsilon$, $p$, $p_1$)
   Input: Dataset $D$, min. support $\varepsilon$, periodicity, $p$ and difference factor, $p_1$
   Output: Productive PFP set $Per_D$
 1 Create HashMap $h_n$                    /* to store all length-1 items in $D$ */
 2 Create set $L$
 3 for each transaction $T \in D$ do
 4     for each length-1 item $a_y \in T$ do
 5         if $a_y \notin h_n$ then
 6             Create $cov_D(a_y) = \{$ TID of $a_y\}$        /* TID = Transaction ID */
 7             Add $(a_y, cov_D(a_y))$ to $h_n$
 8         else
 9             Let $(a_y, cov_D(a_y)) = h_n(a_y)$
10             Udate $cov_D(a_y)$ as $cov_D(a_y) = cov_D(a_y) \cup$ TID of $a_y$
11             Update $h_n$ with $(a_y, cov_D(a_y))$
12 for each item $a_y \in h_n$ do
13     Let $(a_y, cov_D(a_y)) = h_n(a_y)$
14     if $sup_D(a_y) \geq \varepsilon$ then
15         Add $(a_y, cov_D(a_y))$ to $L$
16 Sort $L$ in descending order of items
17 MinePFPs($L$, $\varepsilon$, $p$, $p_1$)
18 return $Per_D$
```

For any dataset **D**, as shown in Lines 1 and 2 of Algorithm 1, a hashmap h_n and the set L respectively are created. From Lines 3 to 11, for each item a_y in each transaction T of **D**, if a_y is not contained in h_n, its coverset $cov_{\mathbf{D}}(a_y)$ is created and the transaction ID of T (occurrence time) added to $cov_{\mathbf{D}}(a_y)$ in Line 6. The tuple $(a_y, cov_{\mathbf{D}}(a_y))$ is then added to h_n in Line 7. Else, if a_y is already contained in h_n, $(a_y, cov_D(a_y))$ is obtained from h_n in Line 9 as $h_n(a_y)$ and the transaction ID of T (occurrence time) added to $cov_{\mathbf{D}}(a_y)$. Hashmap h_n is then updated with $(a_y, cov_{\mathbf{D}}(a_y))$ in Line 11.

After all items and their coversets in **D** are added to h_n, the set of frequent length-1 items in **D** are obtained from Lines 12 to 15 of Algorithm 1 as follows. For each item a_y in h_n, $(a_y, cov_{\mathbf{D}}(a_y))$ is obtained from h_n in Line 13 as $h_n(a_y)$. If a_y is frequent, that is, $sup_{\mathbf{D}}(a_y) \geq \varepsilon$, the tuple $(a_y, cov_D(a_y))$ is added to L in Line 15. After all frequent length-1 items in **D** are obtained as the set L, Line 16 sorts L, which contains all frequent length-1 items in **D** and their coversets, in item-descending order. In Fig. 2, we illustrate the outcome of this stage on Table 1 at $\varepsilon = 0.3$. The next step in PPFPs mines the set of productive periodic frequent patterns from L by calling MinePFPs() in Line 17 of Algorithm 1.

4.2 Mining Productive Periodic Frequent Patterns

This step mines the productive periodic frequent patterns from L by calling MinePFPs(L, ε, p, p_1) (Algorithm 2) in Line 17 of Algorithm 1. Algorithm 2 mines the set of productive periodic frequent patterns from L as follows. Per_D, to store the set of productive periodic frequent patterns is created in Line 1. A temporary set, TempL to store the frequent patterns is also created in Line 2. If there are no items in L,



Pattern	cov_D
{a}	{1, 3, 5, 7, 9, 10}
{b}	{1, 5, 6, 9}
{c}	{1, 4, 7, 8}
{d}	{2, 4, 6, 8, 10}
{e}	{2, 4, 6, 8, 10}
{f}	{1, 3, 5, 7, 9, 10}

Fig. 2 L: sorted frequent length-1 items from Table 1 at $\varepsilon = 0.3$

that is, $|L| = 0$, the productive periodic frequent pattern mining terminates and Per_D returned in Line 5. Else, while $|L| > 0$, the productive periodic frequent patterns are mined from L in the nested for-loop (from Lines 7 to 27 of Algorithm 2) as follows.

In the first for-loop within L (from index $k = 0$ to $|L|$-1), the tuple $(a_k, cov_{\mathbf{D}}(a_k))$ at the k^{th}-index is obtained in Line 9 as $L[k]$. If a_k is a length-1 item, P^{a_k} is obtained from $e.cov_{\mathbf{D}}(a_k)$ in Line 11. $Prd(a_k)$ and $std(P^{a_k})$ are then evaluated from P^{a_k} in Line 12. If a_k is periodic, it is added to Per_D in Line 14. While still at the kth-index, the second for-loop within L (from index $l = (k+1)$ to $|L|-1$) starts in Line 15 as follows. Each tuple $(a_l, cov_{\mathbf{D}}(a_l))$ in the lth-index is obtained in Line 16 as $L[l]$. If a_k and a_l have common length-$(|a_k|-1)$ prefixes, that is, $P_{a_k}[0, |a_k|-1] = P_{a_l}[0, |a_l|-1]$, a candidate frequent pattern, S, is created in Line 18 as $S = (a_k \cup a_l, cov_{\mathbf{D}}(a_k) \cap cov_{\mathbf{D}}(a_l))$.

If S is frequent and productive in $\mathbf{D}$, it is added to TempL in Line 20. This ensures only frequent and productive patterns are kept as we do not want to report periodic frequent patterns with non-productive subsets. In Line 21, P^S is obtained from $e.cov_{\mathbf{D}}(S)$ and, $Prd(S)$ and $std(P^S)$ evaluated in Line 22. If S is periodic, that is, $Prd(S) \pm std(P^S)$ falls within the periodicity range $(p \pm p_1)$, S is added to Per_D in Line 24.

For each kth-index in the first for-loop, the second for-loop repeats till all indexes in L are iterated in the second for-loop. When both nested loops are complete, L is re-created in Line 25 from TempL and the content of TempL is cleared in Line 26. The size of L is checked and the nested looping repeats on L until $|L| = 0$ at which point the periodic frequent pattern mining process terminates and Line 27 returns the set of productive periodic frequent patterns.

We illustrate our productive periodic frequent pattern mining process on Table 1 given $\varepsilon = 0.3$, $p = 1.4$ and $p_1 = 0.8$. As seen in Fig. 3, four stages (I, II, III, and IV) are involved in mining the productive periodic frequent patterns from L. We discuss the processes at each stage as follows.

1. *Stage I* This stage shows the L during the first nested looping. During the first nested looping within L, length-1 frequent patterns $\{a\}$ and $\{f\}$ are added to Per_D in Line 14 of Algorithm 2 since they are both periodic with regards to p and p_1.

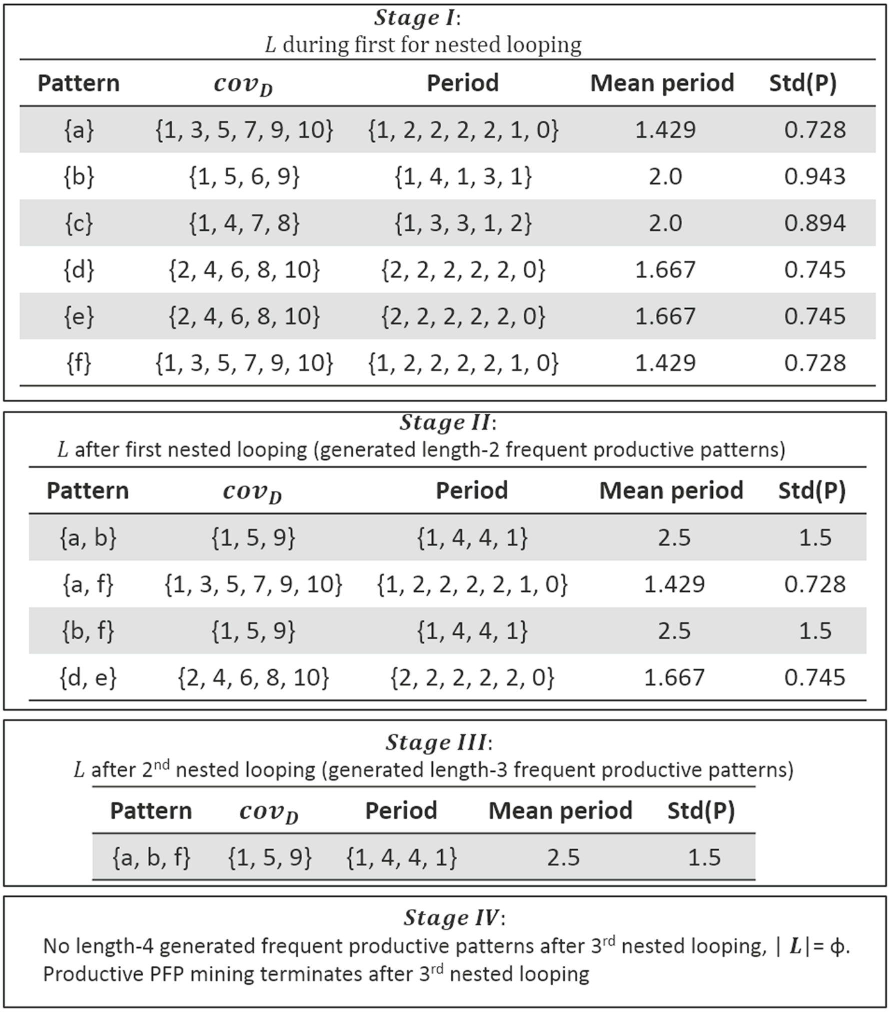
Algorithm 2: MinePFPs(L, ε, p, p_1)

```
   Input: Set $L$, periodicity, $p$, difference factor, $p_1$, and minimum support $\varepsilon$
   Output: Productive PFP set $Per_D$
1  Create $Per_D$
2  Create set TempL = Ø
3  Let $P_{a_n}[0, b]$ be the the length-$b$ prefix of $a_n$
4  if $|L| = 0$ then
5      return $Per_D$
6  else
7      while $|L| > 0$ do
8          for $k = 0$ to $|L|$-1 do
9              Let $(a_k, cov_D(a_k)) = L[k]$
10             if $|a_k| = 1$ then
11                 Obtain $P^{a_k}$ from $e.cov_D(a_k)$
12                 Evaluate $Prd(a_k)$ and $std(P^{a_k})$ from $P^{a_k}$
13                 if $a_k$ is periodic then
14                     Add $a_k$ to $Per_D$          /* length-1 items are productive */
15             for $l = (k + 1)$ to $|L|$-1 do
16                 Let $(a_l, cov_D(a_l)) = L[l]$
17                 if $P_{a_k}[0, |a_k|$-1] = $P_{a_l}[0, |a_l|$-1] then
18                     Create $S = (a_k \cup a_l, cov_D(a_k) \cap cov_D(a_l))$
19                 if $sup_D(S) \geq \varepsilon$ and $S$ is productive then
20                     Add $S$ to TempL
21                     Get $P^S$ from $e.cov_D(S)$
22                     Evaluate $Prd(S)$ and $std(P^S)$ from $P^S$
23                     if $S$ is periodic then
24                         Add $S$ to $Per_D$
25         $L$ = TempL
26         TempL.clear()
27 return $Per_D$
```

Frequent pattern $\{a, f\}$ generated in Line 18 is also added to Per_D in Line 24 it is also periodic.

2. *Stage II* This stage shows L recreated from TempL after the complete first nested for-looping. The nested for-looping repeats on L in Stage II. No productive periodic frequent patterns are detected in this stage during the nested looping as the only frequent generated length-3 pattern, $\{a, b, f\}$ is not periodic.
3. *Stage III* This stage shows L recreated from TempL after the complete second nested for-looping. The nested for-looping repeats on L in this stage and with no generated length-4 patterns.
4. *Stage IV* This stage shows L recreated from TempL after the complete third nested looping. L in this stage has no items since no length-4 patterns were generated in Stage III. The productive periodic frequent pattern mining process thus terminate in this stage as $|L| = 0$.

Line 18 of Algorithm 1 thus reports $Per_D = \{\{a\}, \{f\}, \{a, f\}\}$ as the set of productive periodic frequent patterns in Table 1 at $\varepsilon = 0.3$, $p = 1.4$ and $p_1 = 0.8$.



Stage I:
L during first for nested looping

Pattern	cov_D	Period	Mean period	Std(P)
{a}	{1, 3, 5, 7, 9, 10}	{1, 2, 2, 2, 2, 1, 0}	1.429	0.728
{b}	{1, 5, 6, 9}	{1, 4, 1, 3, 1}	2.0	0.943
{c}	{1, 4, 7, 8}	{1, 3, 3, 1, 2}	2.0	0.894
{d}	{2, 4, 6, 8, 10}	{2, 2, 2, 2, 2, 0}	1.667	0.745
{e}	{2, 4, 6, 8, 10}	{2, 2, 2, 2, 2, 0}	1.667	0.745
{f}	{1, 3, 5, 7, 9, 10}	{1, 2, 2, 2, 2, 1, 0}	1.429	0.728

Stage II:
L after first nested looping (generated length-2 frequent productive patterns)

Pattern	cov_D	Period	Mean period	Std(P)
{a, b}	{1, 5, 9}	{1, 4, 4, 1}	2.5	1.5
{a, f}	{1, 3, 5, 7, 9, 10}	{1, 2, 2, 2, 2, 1, 0}	1.429	0.728
{b, f}	{1, 5, 9}	{1, 4, 4, 1}	2.5	1.5
{d, e}	{2, 4, 6, 8, 10}	{2, 2, 2, 2, 2, 0}	1.667	0.745

Stage III:
L after 2^{nd} nested looping (generated length-3 frequent productive patterns)

Pattern	cov_D	Period	Mean period	Std(P)
{a, b, f}	{1, 5, 9}	{1, 4, 4, 1}	2.5	1.5

Stage IV:
No length-4 generated frequent productive patterns after 3^{rd} nested looping, $|\,\boldsymbol{L}| = \phi$.
Productive PFP mining terminates after 3^{rd} nested looping

Fig. 3 Mining productive periodic frequent patterns at $\varepsilon = 0.3,\ p = 1.4$ and $p_1 = 0.8$ from Table 1

5 Experimental Analysis

The following implementations were used in our experimental analysis:

- *PPFP* This is our implementation based on Definitions 6 and 7. *PPFP* detects and reports the set of productive periodic frequent patterns with similar regular periods. The productiveness measure is used as a pruning strategy to ensure periodic frequent patterns due to random occurrence of uncorrelated items are removed, and for fast periodic frequent pattern discovery.
- *PPFP+* This is our implementation based on only Definition 6 without the productiveness measure. *PPFP+* detects and reports both productive and non-productive periodic frequent patterns with similar regular periods.

Table 3 Datasets

Dataset	Origin	Characteristics
Kosarak25K	SPMF[3]	25,000 Transactions
Accident	FIMI Repository	7593 Transactions
T10I4D100K	FIMI Repository	100,000 Transactions

- *Existing* This is our implementation of the approach proposed in [13]. *Existing* detects and reports periodic frequent patterns whose maximum periods fall below the given periodicity threshold p.

We conduct experimental analysis with regards to: (i) time performance and scalability, and (ii) effects of productiveness measure on reported periodic frequent patterns. The outcome of our analysis are as discussed below.

All compared approaches are implemented in Java and experiments carried on a 64-bit Windows 7 PC (Intel Core i5, CPU 2.50GHz, 4GB). We show our experimental results on the datasets shown in Table 3.

5.1 Time Performance and Scalability

Figures 4 and 5 show the runtime of our proposed *PPFP* and *Existing* on the Kosarak25K and Accident datasets respectively. As can be seen in both Figs. 4 and 5, *PPFP* is significantly more time efficient in periodic frequent pattern discovery compared to the *Existing*. This efficiency improvement is due to the pruning based on productivity. We also observed that the runtime for periodic frequent pattern detection were not significantly affected by increasing or decreasing periodicity threshold p.

5.2 Reported Periodic Frequent Patterns

Tables 4, 5 and 6 show the number of reported periodic frequent patterns for *PPFP*, *PPFP+* and *Existing*. We observed that the productiveness measure removes quite

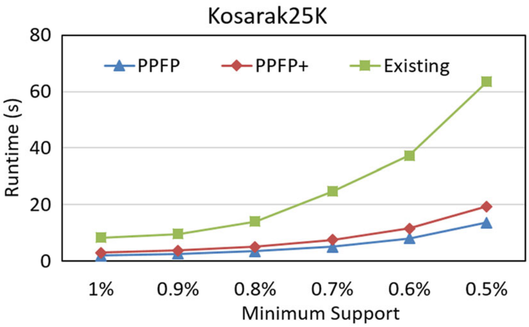


Fig. 4 Periodic frequent pattern discovery runtime at $p = 30$ in Kosarak25K dataset

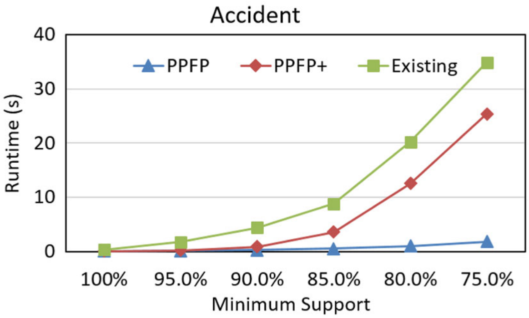


Fig. 5 Periodic frequent pattern discovery runtime at $p = 3.0$ in accident dataset

Table 4 Reported periodic frequent patterns in Kosarak25K dataset

ε	*PPFP*			*PPFP+*			*Existing*		
	$p = 15$	$p = 150$	$p = 200$	$p = 15$	$p = 150$	$p = 200$	$p = 15$	$p = 150$	$p = 200$
	$p_1 = 14.5$	$p_1 = 135$	$p_1 = 190$	$p_1 = 14.5$	$p_1 = 135$	$p_1 = 190$			
0.8 %	13	2	13	16	4	17	1	36	62
0.7 %	13	6	45	16	9	54	1	36	62

Table 5 Reported periodic frequent patterns in accident Dataset

ε	*PPFP*			*PPFP+*			*Existing*		
	$p = 3$	$p = 10$	$p = 15$	$p = 3$	$p = 10$	$p = 15$	$p = 3$	$p = 10$	$p = 15$
	$p_1 = 2.4$	$p_1 = 9.3$	$p_1 = 14.6$	$p_1 = 2.4$	$p_1 = 9.3$	$p_1 = 14.6$			
80 %	36	27	48	199	83	223	7	111	127
75 %	51	27	70	324	83	389	7	187	219

Table 6 Reported periodic frequent patterns in T10I4D100K dataset

ε	*PPFP*			*PPFP+*			*Existing*		
	$p = 15$	$p = 20$	$p = 30$	$p = 15$	$p = 20$	$p = 30$	$p = 15$	$p = 20$	$p = 30$
	$p_1 = 14.5$	$p_1 = 19.5$	$p_1 = 29.5$	$p_1 = 14.5$	$p_1 = 19.5$	$p_1 = 29.5$			
4 %	1	6	10	1	6	10	0	0	0
3 %	1	6	18	1	6	18	0	0	0

a number of non-productive periodic frequent patterns. We also noticed that for a given minimum support, the number of detected periodic frequent patterns in *Existing* increases proportionally with increasing periodicity threshold p. In *PPFP* and *PPFP+* however, the number of detected periodic frequent patterns do not increase exponentially as p increases unless the difference factor p_1 is also incremented by the same proportion.

The disadvantage of employing the maximum period in periodic frequent pattern mining was also observed. For instance, in the Kosarak25K dataset, though patterns such as {6, 11}, {1, 11} and {6, 218} have regular periods between 15 ± 14.5, they were missed by *Existing* as their noisy maximum periods, 22, 78 and 106 respectively are greater than 15.

6 Conclusions and Future Works

Productive PFPs are frequent patterns whose regular periodic occurrences in databases are not due to random occurrence of uncorrelated items. We have presented a measure to identify PFPs with similar regular periods, and a measure to identify the set of productive PFPs in databases. We subsequently develop *PPFP*, an efficient framework for mining the set of productive PFPs in transactional databases. Our future works include an extension of *PPFP* to enable predict future occurrence times of periodic frequent patterns.

References

1. Agrawal R, Srikant R (1995) Mining sequential patterns. In: 11th IEEE international conference on data engineering, IEEE pp 3–14
2. Elfeky MG, Aref WG, Elmagarmid AK (2005) Periodicity detection in time series databases. IIEEE Trans Knowl Data Eng 17(7):875–887
3. Fournier-Viger P, Gomariz A, Gueniche T, Soltani A, Wu C, Tseng VS (2014) SPMF: a java open-source pattern mining library. J Mach Learn Res 15:3389–3393
4. Kiran RU, Reddy PK (2010) Towards efficient mining of periodic-frequent patterns in transactional databases. In: Bringas PG, Hameurlain A, Quirchmayr G (eds) DASFAA 2010. LNCS, Springer, Berlin, pp 194–208
5. Kiran RU, Kitsuregawa M (2014) Novel techniques to reduce search space in periodic-frequent pattern mining. In: Bhowmick SS, Dyreson CE, Jensen CS, Lee ML, Muliantara A, Thalheim B (eds) DASFAA 2014. LNCS, Springer International Publishing, Berlin, pp 377–391
6. Kiran RU, Kitsuregawa M (2013) Discovering quasi-periodic-frequent patterns in transactional databases. In: Bhatnagar V, Srinivasa S (eds) BDA 2013. LNCS, Springer International Publishing, Heidelberg, pp 97–115
7. Kiran RU, Reddy PK (2011) An alternative interestingness measure for mining periodic-frequent patterns. In: Yu JX, Kim MH, Unland R (eds) DASFAA 2011. LNCS, Springer, Heidelberg, pp 183–192
8. Kumar V, Valli Kumari V (2013) Incremental mining for regular frequent patterns in vertical format. Int J Eng Technol 5(2):1506–1511
9. Ma S, Hellerstein JL (2001) Mining partially periodic event patterns with unknown periods. In: 17th IEEE international conference on data engineering, Heidelberg, pp 205–214
10. Rashid MM, Karim MR, Jeong BS, Choi HJ (2012) Efficient mining regularly frequent patterns in transactional databases. In: Lee S, Peng Z, Zhou X, Moon Y, Unland R, Yoo J (eds) DASFAA 2012. LNCS, Springer, Heidelberg, pp 258–271
11. Rashid MM, Gondal I, Kamruzzaman J (2013) Regularly frequent patterns mining from sensor data stream. In: Lee M, Hirose A, Hou ZG, Kil R (eds) NIP 2013. LNCS, Springer, Berlin, pp 417–424
12. Surana A, Kiran RU, Reddy PK (2012) An efficient approach to mine periodic-frequent patterns in transactional databases. In: Cao L, Huang JZ, Bailey J, Koh YS, Luo J (eds) PAKDD 2011 Workshops. LNAI, Springer, Heidelberg, pp 254–266

13. Tanbeer SK, Ahmed CF, Jeong BS, Lee YK (2009) Discovering periodic-frequent patterns in transactional databases. In: Theeramunkong T, Kijsirikul B, Cercone N, Ho T (eds) PAKDD 2009. LNAI, Springer, Heidelberg, pp 242–253
14. Webb GI (2010) Self-sufficient itemsets: an approach to screening potentially interesting associations between items. ACM Trans Knowl Discov Data 4(1):3:1–3:20