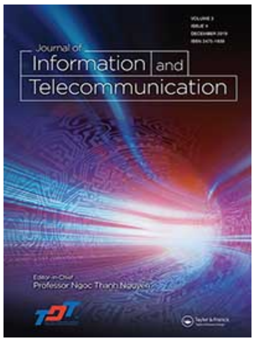


Taylor & Francis
Taylor & Francis Group

Journal of Information and Telecommunication

ISSN: 2475-1839 (Print) 2475-1847 (Online) Journal homepage: https://www.tandfonline.com/loi/tjit20

Towards fast and memory efficient discovery of periodic frequent patterns

Vincent Mwintieru Nofong & John Wondoh

To cite this article: Vincent Mwintieru Nofong & John Wondoh (2019) Towards fast and memory efficient discovery of periodic frequent patterns, Journal of Information and Telecommunication, 3:4, 480-493, DOI: 10.1080/24751839.2019.1634868

To link to this article: https://doi.org/10.1080/24751839.2019.1634868

Published online: 10 Jul 2019.

JOURNAL OF INFORMATION AND TELECOMMUNICATION
2019, VOL. 3, NO. 4, 480–493
https://doi.org/10.1080/24751839.2019.1634868

OPEN ACCESS

Towards fast and memory efficient discovery of periodic frequent patterns

Vincent Mwintieru Nofong [a] and John Wondoh [b]

[a]Faculty of Engineering, University of Mines and Technology, Tarkwa, Ghana; [b]School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, Australia

ABSTRACT

Periodic frequent pattern (PFP) mining, the process of discovering frequent patterns that occur at regular periods in databases, is an important data mining task for various decision-making. Although several algorithms have been proposed for discovering PFPs, most of these algorithms often employ a two-stage approach to mining these periodic frequent patterns. That is, by firstly deriving the set of periods of a pattern from its coverset and subsequently evaluating the patterns' periodicity from the derived set of periods. This two-stage approach in discovering periodic frequent patterns as a result make existing algorithms inefficient in both runtime and memory usage. This paper presents solutions towards reducing the runtime, as well as, memory usage in discovering periodic frequent patterns. This is achieved by evaluating the periodicity of patterns without deriving the set of periods from their coversets. Experimental analysis on benchmark datasets show that the proposed solutions are efficient in reducing both the runtime and memory usage in mining periodic frequent patterns.

ARTICLE HISTORY
Received 22 June 2018
Accepted 19 June 2019

KEYWORDS
Frequent patterns; periodic frequent patterns; efficient measures

1. Introduction

Frequent pattern mining (the process of discovering patterns which occur frequently together) over the past years has been widely studied for knowledge discovery in databases for various decision making. Several algorithms based on various approaches have been developed for mining frequent patterns from database. Typical of such include algorithms that use the: apriori candidate generation approach (Agrawal, Imieliński, & Swami, 1993; Zaki, Parthasarathy, Ogihara, & Li, 1997); frequent pattern growth approach (Han, Pei, & Yin, 2000; Pei et al., 2001); vertical representation approach (Shenoy et al. 2000; Zaki, 2000; Zaki & Gouda, 2003) and hierarchical approach (Tseng, 2013). Though the frequent pattern mining approaches can reveal the frequently occurring patterns in databases, the frequency measure alone in these algorithms often fail in revealing the occurrence shapes of patterns. For example, in crime data or customer transaction analysis, though the frequent pattern mining algorithms will reveal the frequent crimes or customer purchases, they will fail to report the periodic occurrence

CONTACT Vincent Mwintieru Nofong ✉ vnofong@umat.edu.gh; vincent.nofong@mymail.unisa.edu.au

shapes of crimes or customer transactions. However, the ability to detect and understand the periodic occurrence shapes of patterns in databases could be vital in decision-making, for instance, in curbing crime or preventing customer attrition. This limitation in frequent pattern mining algorithms and the relevance of patterns' occurrence shapes in decision-making resulted in the start of research on periodic frequent pattern mining.

Periodic frequent pattern (PFP) mining from transactional datasets has been widely researched on in works such as: Fournier-Viger et al. (2017), Kiran and Reddy (2010), Kiran and Kitsuregawa (2013), Nofong (2016), Surana, Kiran, and Reddy (2012) and Tanbeer, Ahmed, Jeong, and Lee (2009). Several algorithms have been proposed for discovering periodic frequent patterns in transactional databases. Notwithstanding the usefulness of these algorithms in discovering periodic frequent patterns from transactional databases, they are faced with the following challenges:

- Algorithms for mining periodic frequent patterns proposed in works such as: Kiran and Kitsuregawa (2013, 2014), Kiran and Reddy (2010) and Surana et al. (2012) that discover periodic frequent patterns using the maximum periodicity threshold (proposed in Tanbeer et al., 2009) will often miss some important periodic frequent patterns if such patterns have just one periodic (occurrence) interval being greater than the user desired maximum periodicity threshold.
- Algorithms for mining periodic frequent patterns proposed in works such as Kumar and Valli-Kumari (2013) and Rashid, Gondal, and Kamruzzaman (2013) that discover periodic frequent patterns using the maximum variance threshold (proposed in Rashid, Karim, Jeong, & Choi, 2012) will often report a set of periodic frequent patterns having distinct periods for decision-making.
- Most existing algorithms for mining periodic frequent patterns often use a two-stage process to evaluate the periodicity of patterns. That is, they firstly derive the set of periods of a pattern from its coverset and subsequently evaluate the periodicity from the derived set of periods. This thus make existing algorithms employing this two-stage process in mining periodic frequent patterns inefficient in both runtime and memory usage.

Although some of these challenges have been addressed in some recent works, the case of time and memory inefficiency in the discovery of periodic frequent patterns due to the two-stage process, to the best of our knowledge, is yet to be addressed. This paper presents effective and efficient solutions towards mining periodic frequent patterns without employing the two-stage approach in evaluating the periodicity of patterns. Eliminating this two-stage process will in turn reduce the runtime and memory used mining periodic frequent patterns from transactional databases.

The main contributions of this paper towards PFP discovery in transactional databases include:

- It proposes effective and efficient techniques for evaluating the periodicity of patterns without the traditional two-stage approach used in existing works.
- The proposed techniques are incorporated on existing periodic frequent pattern mining algorithms which showed a reduction in both runtime and memory usage in mining periodic frequent patterns.

The rest of this paper is organized as follows. Section 2 discusses related work while the proposed periodicity evaluation measures are introduced in Section 3. Section 4 presents the experimental analysis and Section 5 draws the conclusion and outlines some future works.

2. Related work

The associated notations for periodic frequent pattern discovery 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 items satisfying some conditions of measures like frequency. A pattern is of length-k if it has k items, for example, $S = \{b, c, d, e\}$ is a length-4 pattern.

Given a transactional database of k transactions, $\mathbf{D} = \langle n_1, n_2, n_3, \ldots, n_k \rangle$, where each n_m in $\mathbf{D}$ is identified by m called transaction identifier (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,

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

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

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

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

Given a user desired minimum support (ε), a pattern $S \in \mathbf{D}$ is said to be frequent if $sup_{\mathbf{D}}(S) \geq \varepsilon$.

For any given pattern S in a transactional database $\mathbf{D}$ with $cov_{\mathbf{D}}(S)$ as its coverset, the notation $e.cov_{\mathbf{D}}(S)$ is used to indicate the extension of $cov_{\mathbf{D}}(S)$ by inserting a starting time 0 and the last time m to $cov_{\mathbf{D}}(S)$. That is,

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

where $m = |\mathbf{D}|$. The last time, m will be duplicated if it is already in $cov_{\mathbf{D}}(S)$. For instance, given $|\mathbf{D}| = 7$ and $cov_{\mathbf{D}}(S) = \{1, 2, 4, 7\}$, then, $e.cov_{\mathbf{D}}(S) = \{0\} \cup \{1, 2, 4, 7\} \cup \{7\} = \{0, 1, 2, 4, 7, 7\}$.

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

$$P^S = \{p^S_1, p^S_2, \ldots, p^S_{r-1}, p^S_r\}, \tag{4}$$

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

For example, given $e.cov_{\mathbf{D}}(S) = \{0, 1, 2, 4, 7, 7\}$, then $p^S_1 = (1-0) = 1$, $p^S_2 = (2-1) = 1$, $p^S_3 = (4-2) = 2$, $p^S_4 = (7-4) = 3$, $p^S_5 = (7-7) = 0$, giving $P^S = \{1, 1, 2, 3, 0\}$. Thus, for any pattern S, it can be derived that:

$$|P^S| = |cov_D(S)| + 1. \tag{5}$$

To discover the set of patterns in transactional databases with periodic occurrence shapes, Tanbeer et al. (2009) introduced a periodicity measure on patterns as follows.

Definition 2.1 (Tanbeer et al., 2009): 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, $Per(S) = \max\{p|p \in P^S\}$.

With the periodicity measure proposed in Definition 2.1, Tanbeer et al. (2009) subsequently defined a periodic frequent pattern as a frequent pattern whose periodicity is not greater than a user defined maximum periodicity threshold, *maxPer*.

Given a pattern S and its set of periods P^S, the approach in Tanbeer et al. (2009) returns S as periodic if the maximal occurring period (that is, the maximal time interval between any two consecutive occurrence times) of S is not greater than the maximum periodicity threshold, *maxPer*. This idea of discovering periodic frequent patterns using the maximal occurring period as proposed in Tanbeer et al. (2009) have been used in periodic frequent pattern mining in transactional databases in works such as: Kiran and Kitsuregawa (2014), Kiran and Reddy (2010, 2011), Lin, Zhang, Fournier-Viger, Hong, and Zhang (2017) and Surana et al. (2012).

Rashid et al. (2012) however argued that discovering periodic frequent patterns using the periodicity measure proposed in Tanbeer et al. (2009) is inappropriate as it returns the maximum time-interval (period) for which a pattern does not appear in a database as its periodicity. Rashid et al. (2012) thus defined the periodicity of a pattern under the name *regularity* as follows.

Definition 2.2 (Rashid et al., 2012): 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.

Based on the regularity (periodicity) measure proposed in Definition 2.2, Rashid et al. (2012) defined a regular (periodic) frequent pattern as a frequent pattern whose variance among its set of periods is not greater than a user desired maximum regularity threshold, *maxReg*. This concept of discovering regular frequent patterns based on the proposition in Rashid et al. (2012) has been used in discovering regular frequent patterns in works such as Kumar and Valli-Kumari (2013) and Rashid et al. (2013).

Nofong (2016) however argued that, though the proposition in Rashid et al. (2012) will not miss interesting periodic frequent patterns as in Tanbeer et al. (2009), algorithms that mine periodic frequent patterns using the propositions in both Tanbeer et al. (2009) and Rashid et al. (2012) will always report periodic frequent patterns having totally distinct periods. To report only periodic frequent patterns with similar periods for decision-making, Nofong (2016) defined a periodic frequent pattern as follows.

Definition 2.3 (Nofong, 2016): 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, $Prd(S)$ is the periodicity of S (defined as the mean of P^S, that is, $Prd(S) = \bar{x}(P^S)$) and $std(P^S)$ the standard deviation in P^S.

With Definition 2.3, though PFPs having similar periodic shapes will be mined and reported, Nofong (2016) observed that some of the reported periodic frequent patterns may be periodic due to random chance without inherent item relationship. To ensure only periodic frequent patterns having inherent item relationships are mined and returned, the productiveness measure (proposed in Webb, 2010) was incorporated by

Nofong (2016) in defining the productive periodic frequent patterns as the set of periodic frequent patterns with inherent item relationship.

Fournier-Viger et al. (2017) introduced PFPM, an efficient algorithm having novel pruning techniques for discovring periodic frequent patterns in transactional databases. PFPM unlike the proposed techniques in Nofong (2016), Rashid et al. (2012) and Tanbeer et al. (2009), Fournier-Viger et al. (2017) proposed three periodicity measures (that is, the *minimum*, *maximum* and *average* periodicity measures) for mining user desired periodic frequent patterns. The three measures proposed in Fournier-Viger et al. (2017) for periodic frequent pattern mining in transactional datasets thus give users the advantage of more flexibility in discovering periodic frequent patterns.

As mentioned previously, the propositions in Fournier-Viger et al. (2017), Nofong (2016), Rashid et al. (2012) and Tanbeer et al. (2009) and works based on these propositions (Kiran & Kitsuregawa, 2014; Kiran & Reddy, 2010, 2011; Kumar & Valli-Kumari, 2013; Rashid et al., 2013; Surana et al., 2012) are faced with the challenges of time and memory inefficiency and difficulty in finding early termination mechanisms in periodic frequent pattern discovery in transactional databases.

3. Proposed periodicity evaluation measures

We adopt the periodic frequent pattern definition (Definition 2.3) proposed in Nofong (2016). To enable the mining of PFPs based on Definition 2.3 while addressing the time and memory inefficiencies in discovering periodic frequent patterns, we show that the periodicity of a pattern can be evaluated directly from its coverset and the size of the database as follows.

Lemma 3.1: *Given a database* $\mathbf{D} = \langle n_1, n_2, n_3, \ldots, n_k \rangle$ *from which a pattern* S *is mined, the periodicity of* S *in* $\mathbf{D}$ *can be expressed as* $Prd_D(S) = |D|/(|cov_D(S)| + 1)$.

Proof. Let $cov_D(S) = (n_1, n_2, n_3, \ldots, n_{m-1}, n_m)$, then based on Equation (3), $e.cov_D(S)$ becomes $e.cov_D(S) = (0, n_1, n_2, n_3, \ldots, n_{m-1}, n_m, |D|)$. Hence, P^S becomes, $P^S = ((n_1 - 0), (n_2 - n_1), (n_3 - n_2), \ldots, (n_m - n_{m-1}), (|\mathbf{D}| - n_m))$. As such, $Prd_D(S)$ (that is, $Prd_D(S) = \sum_{i=1}^{|P^S|} P_i^S / |P^S|$) can thus be expressed as,

$$Prd_D(S) = \frac{(n_1 - 0) + (n_2 - n_1) + (n_3 - n_2) + \cdots + (n_m - n_{m-1}) + (|\mathbf{D}| - n_m)}{|P^S|}.$$

This simplifies to $Prd_D(S) = |\mathbf{D}|/|P^S|$. However, from Equation (5), since $|P^S| = |cov_D(S)| + 1$, then the periodicity of S, $Prd_D(S)$ can be expressed as $Prd_D(S) = |\mathbf{D}|/(|cov_D(S)| + 1)$.

For instance, given $|\mathbf{D}| = 7$ and $cov_{\mathbf{D}}(S) = \{2, 3, 7\}$ (that is, $|cov_D(S)| = 3$), then, based on Lemma 3.1, $Prd_D(S) = 7/(3 + 1) = 1.75$. Though the periodicity of S, $Prd_D(S)$ can be evaluated using the traditional two-stage process[1] without Lemma 3.1, more time and memory will be required in evaluating $Prd_D(S)$ without Lemma 3.1 as explained below.

Let $\mathbf{D}$ be a dataset and n the set of frequent patterns in $\mathbf{D}$ whose periodicities are to be evaluated. The functions for evaluating the periodicities based on Lemma 3.1 (Function 1) and without Lemma 3.1 (Function 2) are as shown below.

```
Function 1
for $i = 1 \rightarrow n$ do
    Get $|cov_D(i)|$
    $Prd_D(i) = \frac{|D|}{|cov_D(S)|+1}$
return $Prd_D(i)$
```

```
Function 2
for $i = 1 \rightarrow n$ do
    Get $|cov_D(i)|$
    Let $m = |cov_D(i)|$
    Create $P^i$
    for $k = 1 \rightarrow m$ do
        Obtain elements of $P^i$
    for $j = 1 \rightarrow |P^i|$ do
        $sum+ = P^i[j]$
    $Prd_D(i) = \frac{sum}{|P^i|}$
return $Prd_D(i)$
```

Analysing both Functions 1 and 2 based on the Big-O notation, Function 1 employs only one for-loop in evaluating the periodicity of all potential periodic frequent patterns while Function 2 uses a nested for-loop for the same purpose. As such, the runtime complexity of Function 1 based on Lemma 3.1 turns out as $O(n)$ while that of Function 2 turns out as $O(n^2)$. Hence, there will be a significant reduction in runtime if Lemma 3.1 is employed in evaluating the periodicity of patterns.

It is, however, worth nothing that, though Lemma 3.1 will evaluate the periodicity of a pattern, it will not be able to evaluate the standard deviation[2] among the set of periods of patterns. In existing works on discovering periodic frequent patterns, the set of periods for each pattern are often derived from their coversets before evaluating the standard deviation among the derived set of periods. As mentioned previously, deriving the set of periods from a pattern's coverset and subsequently evaluating its periodicity from the derived set of periods make existing algorithms on discovering periodic frequent patterns inefficient in both runtime and memory usage.

To eliminate the two-stage process in periodic frequent pattern discovery, we show how the standard deviation among the set of periods can be directly derived from the coverset without necessarily evaluating the set of periods as follows.

Lemma 3.2: *Given $cov_D(S) = \{n_1, n_2, n_3, \ldots, n_{m-1}, n_m\}$, the standard deviation among the set of periods of* S *can be evaluated as $std(P^S) = \sqrt{(X_S + Y_S + Z_S)/(|cov_D(S)| + 1)}$ where:*

$$X_S = n_1^2 + Prd(S)^2, \tag{6}$$

$$Y_S = n_m^2 + |D|^2 + Prd(S)^2 - 2|D|(n_m + Prd(S)), \tag{7}$$

$$Z_S = \sum_{j=2}^{m-1} (n_j^2 + n_{j-1}^2 + Prd(S)^2 - 2n_j n_{j-1}). \tag{8}$$

Proof. Let $\bar{x} = Prd(S)$. Given $cov_D(S) = \{n_1, n_2, n_3, \ldots, n_{m-1}, n_m\}$, then the set of periods of S becomes $P^S = \{n_1 - 0, n_2 - n_1, n_3 - n_2, \ldots, n_m - n_{m-1}, |D| - n_m\}$. As such, the variance among the set of periods of S, that is, $var(P^S) = \sum_{i=1}^{|P^S|} ((P_i^S - \bar{x})^2/|P^S|)$ expands to

$$var(P^S) = \frac{((n_1 - 0) - \bar{x})^2 + ((n_2 - n_1) - \bar{x})^2 + +((n_m - n_{m-1}) - \bar{x})^2 + ((|D| - n_m) - \bar{x})^2}{|cov_D(S)| + 1}.$$

Expanding the expressions in the numerator gives:

$((n_1 - 0) - \bar{x})^2 = n_1^2 + \bar{x}^2 - 2n_1\bar{x}$.

$((n_2 - n_1) - \bar{x})^2 = n_1^2 + n_2^2 + \bar{x}^2 - 2n_1n_2 - 2n_2\bar{x} + 2n_1\bar{x}$.

$((n_3 - n_2) - \bar{x})^2 = n_2^2 + n_3^2 + \bar{x}^2 - 2n_2n_3 - 2n_3\bar{x} + 2n_2\bar{x}$.

.

.

$((n_m - n_{m-1}) - \bar{x})^2 = n_{m-1}^2 + n_m^2 + \bar{x}^2 - 2n_{m-1}n_m - 2n_m\bar{x} + 2n_{m-1}\bar{x}$.

$((|D| - n_m) - \bar{x})^2 = n_m^2 + |D|^2 + \bar{x}^2 - 2n_m|D| - 2|D|\bar{x} + 2n_m\bar{x}$.

Summing the above expansion results in the following:

$$X_S = n_1^2 + Prd(S)^2 \text{ (for the first expansion) } \ldots \textcircled{1}$$

where X_S is the variance value for the first period in P^S.

$$Y_S = n_m^2 + |D|^2 + Prd(S)^2 - 2n_m|D| - 2|D|Prd(S) \text{(for the last expansion)} \ldots \textcircled{2}$$

where Y_S is the variance value for the last period in P^S.

$$Z_S = \sum_{j=2}^{m-1} (n_j^2 + n_{j-1}^2 + Prd(S)^2 - 2n_jn_{j-1}) \text{ (for any other period in } P^S\text{) } \ldots \textcircled{3}$$

where Z_S is the variance value for any other period in P^S which is not the first or last period.

Hence for any given pattern S and its coverset, the variance and standard deviation among its periods can be obtained respectively as:

$$var(P^S) = \frac{X_S + Y_S + Z_S}{|cov_D(S)| + 1}$$

and;

$$std(P^S) = \sqrt{\frac{X_S + Y_S + Z_S}{|cov_D(S)| + 1}}.$$

We compare the proposed techniques for evaluating the periodicity of patterns, that is, Lemmas 3.1 and 3.2 (as Function 3) vis-a-vis the existing two-stage approach of evaluating the periodicity of patterns (as Function 4) as follows.

Function 3

```
for $i = 1 \rightarrow n$ do
    Get $|cov_D(i)|$
    Let $m = |cov_D(i)|$
    $Prd_D(i) = \frac{|D|}{|cov_D(i)|+1}$
    for $k = 1 \rightarrow m$ do
        case 1:
        Evaluate $X_S$
        case $m$:
        Evaluate $Y_S$
        default:
        Evaluate $Z_S$
return $Prd_D(i)$, $var(P^i)$, $std(P^i)$
```

Function 4

```
for $i = 1 \rightarrow n$ do
    Get $|cov_D(i)|$
    Let $m = |cov_D(i)|$
    Create $P^i$
    for $k = 1 \rightarrow m$ do
        Obtain elements of $P^i$
    for $j = 1 \rightarrow |P^i|$ do
        $sum += P^i[j]$
    $Prd_D(i) = \frac{sum}{|P^i|}$
    for $k = 1 \rightarrow |P^i|$ do
        $vsum += (Prd_D(i) - P^i[k])^2$
    $var(P^i) = \frac{vsum}{|P^i|}$
    $std(P^i) = \sqrt{var(P^i)}$
return $Prd_D(i)$, $var(P^i)$, $std(P^i)$
```

Let **D** be a dataset and n the set of frequent patterns in **D** whose periodicities are to be evaluated. With the Big-O notation analysis on Functions 3 and 4, the runtime complexity of Function 3 (based on Lemmas 3.1 and 3.2) turns out to be $O(n^2)$ while that of Function 4 is $O(3n^2)$. However, in the worse case scenario, both Functions 3 and 4 will have same runtime complexities of $O(n^2)$.

4. Experimental analysis

To show the effectiveness of our proposed periodicity evaluation measures, we incorporate them on existing algorithms for mining periodic frequent patterns and test their effectiveness on benchmark datasets. The effectiveness of our proposed measures were analysed with regards to runtime (execution time) and memory usage in discovering periodic frequent patterns.

For our experimental analysis,[3] the following implementations were used:

- PFP*: PFP* is our implementation of the technique for mining all periodic frequent patterns. For any given user thresholds and a given dataset, PFP* discovers and returns the set of all periodic frequent patterns having similar periodicities.
- PFP+: PFP+ is our improved implementation of PFP* which incorporates our proposed periodicity evaluation measures. For any given user thresholds and a given dataset, PFP + discovers and returns the set of all periodic frequent patterns having similar periodicities.
- PPFP: PPFP is an implementation of the periodic frequent pattern mining technique proposed in Nofong (2016). For any given user thresholds and a given dataset, PPFP discovers and returns all productive periodic frequent patterns having similar periodicities.
- PPFP+: PPFP+ is our improved implementation of PPFP which incorporates our proposed periodicity evaluation measures. For any given user thresholds and a given dataset, PPFP+ discovers and returns the set of all productive periodic frequent patterns having similar periodicities.

For the above four algorithms, experimental analysis were conducted with regards to (i) execution time and (ii) memory usage. The following datasets described below were used for our experimental analysis.

- **Accident Dataset**: This was obtained from the FIMI repository. The Accident dataset which consists of 7593 transactions is by nature very dense.
- **Kosarak10K Dataset**: This was obtained from SPMF (Fournier-Viger et al., 2016). The Kosarak10K which is partly dense consists of 10,000 transactions.
- **Kosarak45K Dataset**: This was obtained from SPMF (Fournier-Viger et al., 2016). The Kosarak45K which is partly dense consists of 45,000 transactions.
- **Tafeng Nov. 2000 Dataset**: This was obtained from the AIIA Lab. Consisting of 31,807 transactions in the month of November 2000, this dataset is very sparse.

It is worth noting that the compared algorithms were implemented in Java and the experiments carried on a 64-bit Windows 10 PC (Intel Core i7, CPU 2.10GHz, 12GB).

The results of the experimental analysis with regards to execution time and memory usage are discussed below.

4.1. Execution time

For scalability and time performance, we compare the four implementations mentioned above on the datasets described above. The values recorded and plotted for each

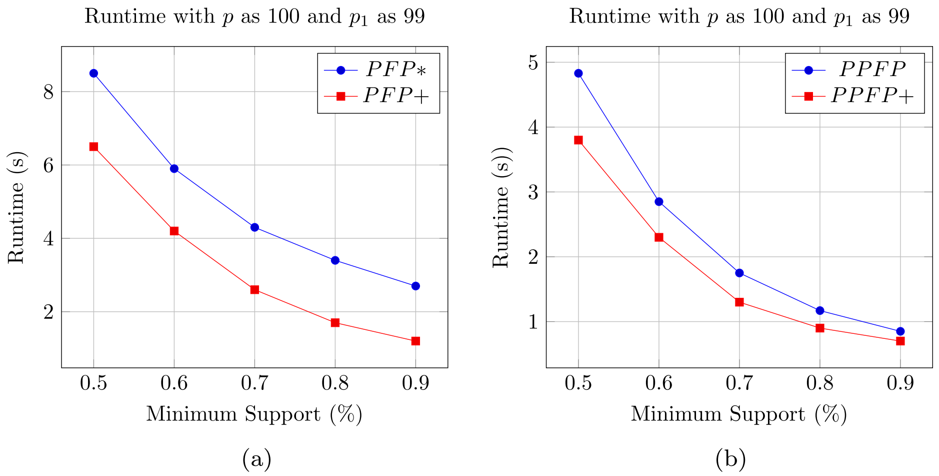


Figure 1. PFP discovery: runtime Kosarak10K dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.

dataset are average values of the experiments which were run ten (10) times. Figures 1–4 show the execution comparison of the above mentioned implementations in mining periodic frequent patterns from the Kosarak10K, Kosarak45k, Accident and Tafeng datasets respectively.

As can be seen in Figures 1–4, incorporating our proposed periodicity evaluation techniques on existing periodic frequent pattern mining algorithms significantly reduces the runtime required in periodic frequent pattern discovery. For instance, in Figures 1(a) and 2(a), PFP+ which is an implementation based on our proposed techniques is almost twice as efficient as PFP* with regards to the time required in discovering periodic frequent patterns. Also, as can be seen in Figures 1(b), 2(b), 3 and 4, PFP+ and PPFP+ (which are all implementations incorporating our proposed techniques) are also slightly more efficient compared to PFP* and PPFP in periodic frequent pattern discovery.

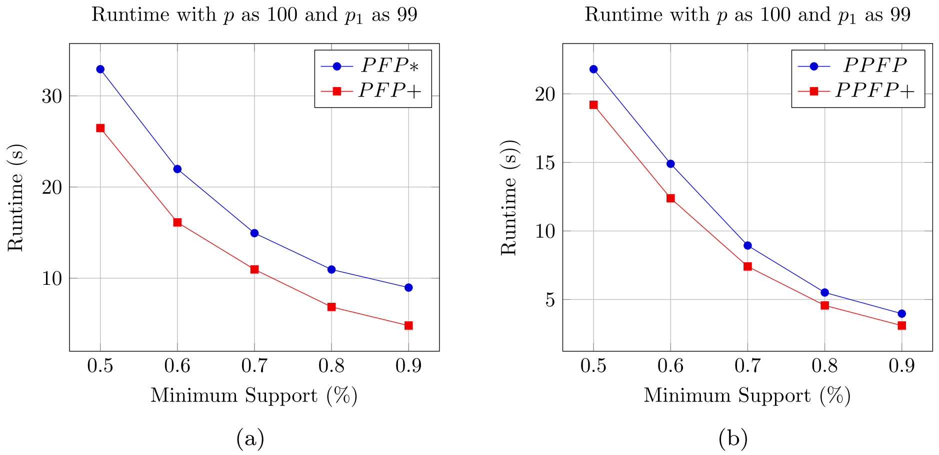
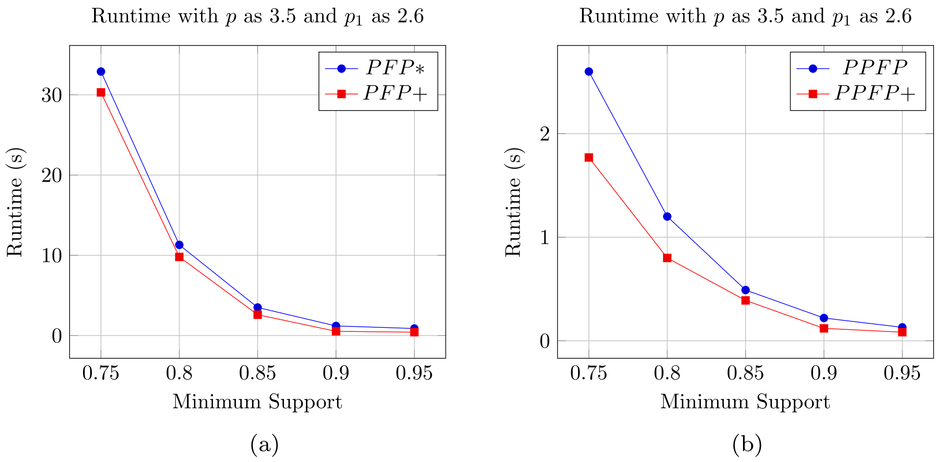
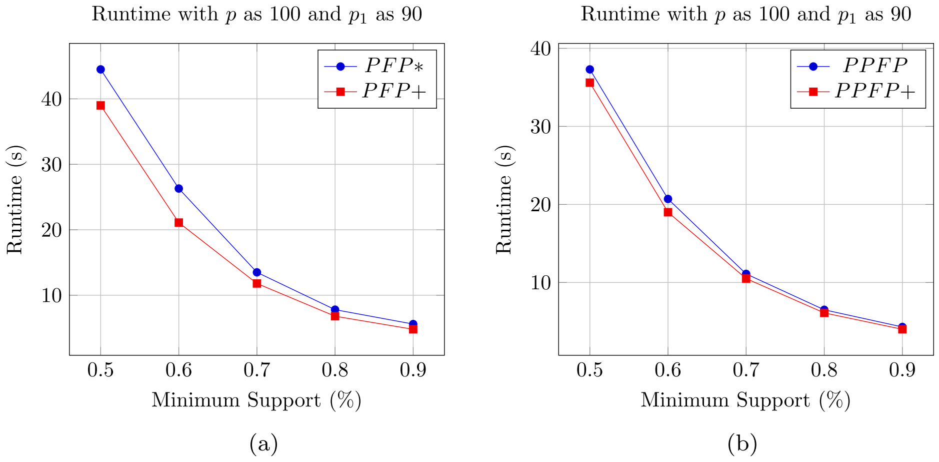


Figure 2. PFP discovery: runtime Kosarak45K dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.



(a) (b)

Figure 3. PFP discovery: runtime accident dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.



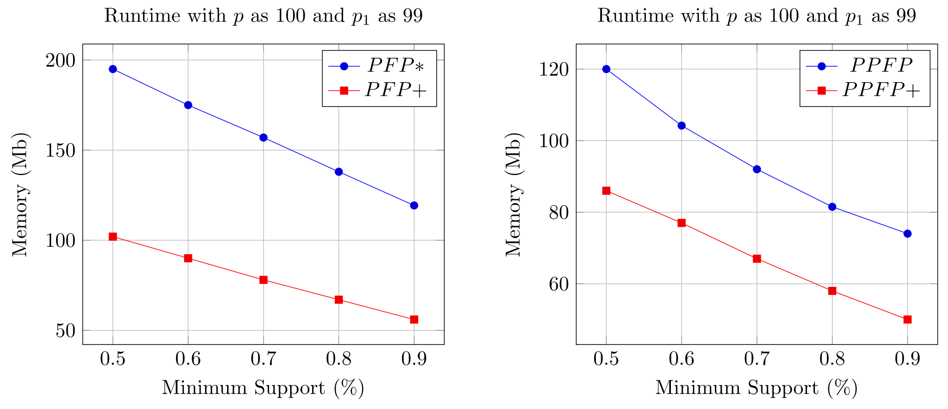
(a) (b)

Figure 4. PFP discovery: runtime Tafeng dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.

4.2. Memory usage

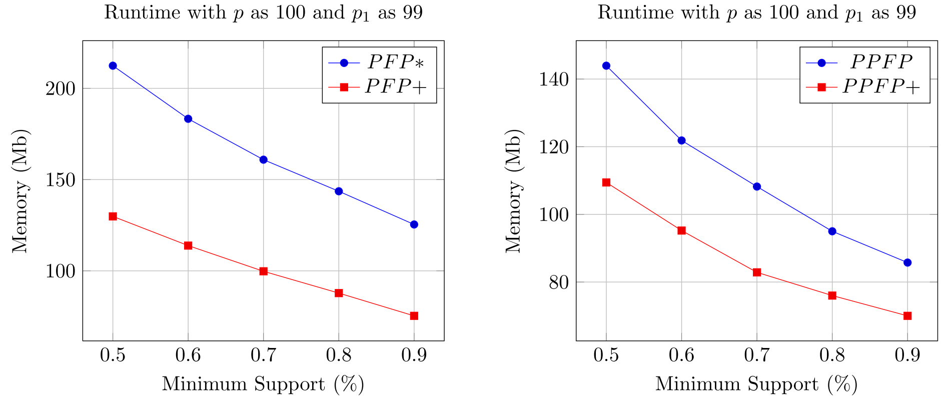
We also compare the memory used in discovering periodic frequent patterns by the four mentioned implementations on the datasets described above. The values recorded and plotted for each dataset are average values of the experiments which were run ten (10) times. Figures 5–8 show the memory usage comparison of the four implementations on the Kosarak10K, Kosarak45K, Accident and Tafeng datasets respectively.

As can be seen in Figures 5–8, incorporating our proposed periodicity evaluation techniques on existing periodic frequent pattern mining algorithms significantly reduces the memory usage in periodic frequent pattern discovery. In Figures 5–7 for instance, both PFP+ and PPFP+ (which are implementations incorporating our proposed techniques) are almost twice as efficient in memory usage compared to PFP* and PPFP in periodic frequent pattern discovery.



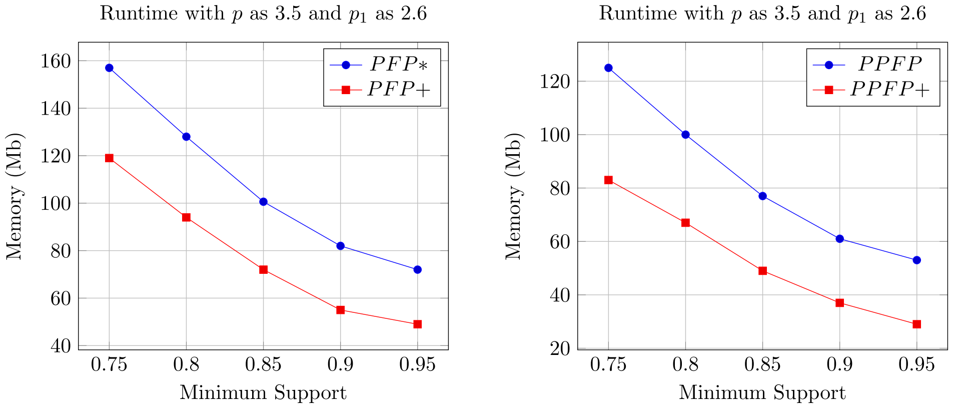
(a) (b)

Figure 5. PFP discovery: memory usage Kosarak10K dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.



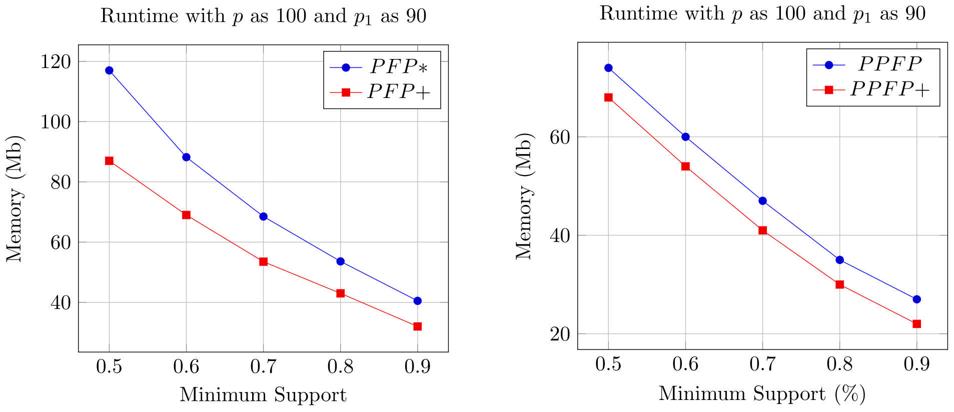
(a) (b)

Figure 6. PFP discovery: memory usage Kosarak45K dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.



(a) (b)

Figure 7. PFP discovery: memory usage accident dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.



(a) (b)

Figure 8. PFP discovery: memory usage Tafeng dataset. (a) PFP* vs PFP+, (b) PPFP vs PPFP+.

5. Conclusion

Despite the usefulness of periodic frequent patterns in revealing useful occurrence shapes in databases, existing algorithms for their discovery often employ a two-stage process, thus making them inefficient in runtime and memory usage. This paper proposes effective and efficient techniques towards reducing the runtime and memory used in discovering periodic frequent patterns from databases. Incorporating these techniques on existing periodic frequent pattern mining algorithms, we show experimentally on benchmark datasets that our proposed techniques are efficient in reducing both the runtime and memory used in periodic frequent pattern discovery. Our future works will be towards further improvement of the algorithm through pseudo-projection in order to reduce the memory used in periodic frequent pattern mining.

Notes

1. That is, deriving the set of periods and subsequently evaluating the periodicity from the set of periods.
2. Which will be required to identify periodic frequent patterns with similar periodicities – see Definition 2.3.
3. We do not compare our implementations with that proposed in Tanbeer et al. (2009) since PPFP in Nofong (2016) is shown to outperform the proposition in Tanbeer et al. (2009).

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Vincent Mwintieru Nofong is a lecturerat the University of Mines and Technology, Tarkwa, Ghana. He received a Ph.D. degree in Computer and Information Science in 2016 at University of South Australia, Adelaide and a B.Sc. degree in 2010 from the University of Mines and Technology, Tarkwa, Ghana.

His current research interests include data mining, pattern mining, classification and trend prediction.

John Wondoh is a researcher and lecturerat the University of South Australia, Adelaide. He received a Ph.D. degree in Computer and Information Science in 2018 at University of South Australia, Adelaide. He currently works as part of the team for several courses at the university. His research areas of interest include event recognition and processing, process optimizations and machine learning.

ORCID

Vincent Mwintieru Nofong http://orcid.org/0000-0001-9123-2840
John Wondoh http://orcid.org/0000-0002-7810-6396

References

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.* (Vol. 22(2), pp. 207–216). ACM.

Fournier-Viger, P., Lin, C. W., Duong, Q. H., Dam, T. L., Ševčík, L., Uhrin, D., & Voznak, M. (2017). PFPM: Discovering periodic frequent patterns with novel periodicity measures. *Proceedings of the 2nd Czech-China scientific conference*. InTech.

Fournier-Viger, P., Lin, J. C. W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016). The SPMF open-source data mining library version 2. *Proceedings of European conference on machine learning and knowledge discovery in databases* (pp. 36–40). Cham: Springer.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Rec.* (Vol. 29(2), pp. 1–12). ACM.

Kiran, R. U., & Kitsuregawa, M. (2013). Discovering quasi-periodic-frequent patterns in transactional databases. In V. Bhatnagar, & S. Srinivasa (Eds.), *BDA 2013*. LNCS, Vol. 8302 (pp. 97–115). Springer International.

Kiran, R. U., & Kitsuregawa, M. (2014). Novel techniques to reduce search space in periodic-frequent pattern mining. In S. S. Bhowmick, C. E. Dyreson, C. S. Jensen, M. L. Lee, A. Muliantara, & B. Thalheim (Eds.), *DASFAA 2014*. LNCS, Vol. 8422 (pp. 377–391). Springer International.

Kiran, R. U., & Reddy, P. K. (2010). Towards efficient mining of periodic-frequent patterns in transactional databases. In P. G. Bringas, A. Hameurlain, & G. Quirchmayr (Eds.), *DASFAA 2010*. LNCS, Vol. 6262 (pp. 194–208). Springer, Heidelberg.

Kiran, R. U., & Reddy, P. K. (2011). An alternative interestingness measure for mining periodic-frequent patterns. In J. X. Yu, M. H. Kim, & R. Unland (Eds.), *DASFAA 2011*. LNCS, Vol. 6587 (pp. 183–192). Springer, Heidelberg.

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

Lin, J. C. W., Zhang, J., Fournier-Viger, P., Hong, T. P., & Zhang, J. (2017). A two-phase approach to mine short-period high-utility itemsets in transactional databases. *Advanced Engineering Informatics*, *33*, 29–43.

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

Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., & Yang, D. (2001). H-mine Hyper-structure mining of frequent patterns in large databases. *Proceedings IEEE international conference on data mining* (pp. 441–448). IEEE.

Rashid, M. M., Gondal, I., & Kamruzzaman, J. (2013). Regularly frequent patterns mining from sensor data stream. In M. Lee, A. Hirose, Z. G. Hou, & R. Kil (Eds.), *NIP 2013*. LNCS, Vol. 8227 (pp. 417–424). Springer, Berlin, Heidelberg.

Rashid, M. M., Karim, M. R., Jeong, B. S., & Choi, H. J. (2012). Efficient mining regularly frequent patterns in transactional databases. In S. Lee, Z. Peng, X. Zhou, Y. Moon, R. Unland, & J. Yoo (Eds.), *DASFAA 2012*. LNCS, Vol. 7238 (pp. 258–271). Heidelberg: Springer.

Shenoy, P., Haritsa, J. R., Sudarshan, S., Bhalotia, G., Bawa, M., & Shah, D. (2000). Turbo-charging vertical mining of large databases. In *ACM SIGMOD Record* (Vol. 29(2), pp. 22–33). ACM.

Surana, A., Kiran, R. U., & Reddy, P. K. (2012). An efficient approach to mine periodic-frequent patterns in transactional databases. In L. Cao, J. Z. Huang, J. Bailey, Y. S. Koh, & J. Luo (Eds.), *PAKDD 2011 Workshops*. LNAI, Vol. 7104 (pp. 254–266). Springer Heidelberg.

Tanbeer, S. K., Ahmed, C. F., Jeong, B. S., & Lee, Y. K. (2009). Discovering periodic-frequent patterns in transactional databases. In: T. Theeramunkong, B. Kijsirikul, N. Cercone, & T. Ho (Eds.), *PAKDD 2009*. LNAI, Vol. 5476 (pp. 242–253). Springer Heidelberg.

Tseng, F. C. (2013). Mining frequent itemsets in large databases: The hierarchical partitioning approach. *Expert Systems with Applications*, *40*(5), 1654–1661.

Webb, G. I. (2010). Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Transactions on Knowledge Discovery from Data*, *4*(1), 3:1–3:20.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, *12*(3), 372–390.

Zaki, M. J., & Gouda, K. (2003). *Fast vertical mining using diffsets. Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 326–335) Washington, D.C.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). Parallel algorithms for discovery of association rules. *Data Mining and Knowledge Discovery*, *1*(4), 343–373.