An Efficient Algorithm to Discover Intra-Periodic Frequent Sequences

Kenmogne Edith Belise* — Tayou Djamegni Clementin*,**

* Department of Mathematics and Computer Science, URIFIA
** Department of Computer Engineering, IUT-FV
University of Dschang, Cameroon
ebkenmogne@gmail.com
dtayou@gmail.com

ABSTRACT. Sequential pattern mining techniques permit to discover recurring structures or patterns from very large datasets, with a very large field of applications. It aims at extracting a set of attributes, shared across time among a large number of objects in a given database. It is a challenging problem since mining algorithms are well known to be both time and memory consuming for large databases. In this paper, we extend the traditional problem of mining frequent sequences with intra-periodicity constraints. Then, we study issues related to intra-periodicity constraints such as search space pruning and partitioning. This study leads to a new efficient algorithm called Intra-Periodic Frequent Sequence Miner (IPFSM). Experimental results confirm the efficiency of IPFSM.

RÉSUMÉ. Les techniques de recherche des motifs séquentiels permettent de découvrir des structures ou modèles récurrents à partir de très grandes bases de données, avec un très large champ d'applications. Elles visent à extraire un ensemble d'attributs, partagés dans le temps entre un grand nombre d'objets dans une base de données. C'est un problème difficile, car les algorithmes de recherche des motifs séquentiels sont gourmandes en temps CPU et en mémoire sur des grandes bases de données. Dans ce papier, nous étendons le problème traditionnel de l'extraction des séquences fréquentes avec des contraintes d'intra-périodicité. Ensuite, nous étudions les problèmes liés aux contraintes de périodicité, notament l'élagage et le partitionnement de l'espace de recherche. Cette étude conduit à un nouvel algorithme efficace appelé *Intra-Periodic Frequent Sequence Miner* (IPFSM). Les résultats expérimentaux confirment l'éfficacité de l'IPFSM.

KEYWORDS : Frequent sequence, intra-periodicity, pruning, partitioning

MOTS-CLÉS : Séquence fréquente, intra-périodicité, élargage, partitionnement

1. Introduction

Nowadays, the generalized use of new technologies of information and communication allows us to gather more data automatically. Because of the fast computerization of administrations, enterprises, trade and telecommunications, the rate of stored data increases quickly. However, the analysis and exploitation of these data is sometimes very difficult. In this context, sequential pattern mining [13, 5, 4, 2, 8, 17, 1, 9, 11, 12, 3] is an important data mining problem widely addressed by the data mining community. It aims at extracting a set of attributes, shared across time among a large number of objects in a given data base. It is a challenging problem since mining algorithms are well known to be both time and memory consuming for large databases, and improvements are motivated by the need to process more data at a faster speed with lower cost. This trend and the integration of intra-periodicity constraints in the mining process are the main motivations for this paper. Previous work in frequent sequence mining and periodicty only consider extra periodicity [18, 19, 20].

In this paper, we extend the traditional problem of mining frequent sequences with intra-periodicity constraints. Then, we study issues related to intra-periodicity constraints such as search space pruning and partitioning. This study leads to a new efficient algorithm called Intra-Periodic Frequent Sequence Miner (IPFSM). Experimental results confirm the efficiency of IPFSM.

The sequel of this paper is organized as follows. Section 2 states the problem. Section 3 studies search space pruning and partitioning under intra-periodicity constraints. Section 4 presents algorithm IPFSM. Section 5 presents experimental results. Concluding remarks are stated in section 6.

2. Statement of the problem

2.1. The traditional problem of mining frequent sequences

The traditional problem of mining sequential patterns [13, 5, 4, 2, 8, 17, 1, 9, 11, 12, 3] and its associated notation, can be given as follows:

Let $I = \{i_1,\ i_2,\ ...,\ i_n\}$ be a set of literals, termed **items**, which comprise the alphabet. An **itemset** is a subset of items. For sake of simplicity [13, 5, 12, 3], we assume that all the items of an itemset are alphabetically sorted.

A **sequence** is an ordered list of itemsets. A sequence s is denoted by $\prec s_1,\ s_2,\ ...s_n \succ$, where s_j is an itemset. s_j is also called an **element** of the sequence, and denoted as $(x_1, x_2,\ ..., x_m)$, where x_k is an item. For brevity, the brackets are omitted if an element has only one item, i.e. element (x) is written as x. An item can occur at most once in an element of a sequence, but can occur multiple times in different elements of a sequence. The number of instances of items (resp. elements) in a sequence α is denoted $|\alpha|$ (resp. $||\alpha||$). The value of $|\alpha|$ is called the length of the sequence. The number of A sequence with length l is called an **l-sequence**. A sequence $\alpha = \prec a_1 a_2 ... a_n \succ$ is called **subsequence** of another sequence $\beta = \prec b_1 b_2 ... b_m \succ$ and β a **supersequence** of α, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \ ... \ < j_n \leq m$ such that $a_1 \subseteq b_{j1},\ a_2 \subseteq b_{j2},\ ...\ ,\ a_n \subseteq b_{jn}$. Symbol ϵ denotes the **empty sequence**.

We are given a database S of input-sequences. A **sequence database** is a set of tuples of the form $\prec sid, s \succ$ where *sid* is a **sequence_id** and s a sequence. A tuple $\prec sid, s \succ$

is said to contain a sequence α, if α is a subsequence of s. The support of a sequence α in a sequence database S is the number of tuples in the database containing α, i.e.

$$support(S, \alpha) = |\{\prec sid, s \succ \mid \prec sid, s \succ \in S \ \wedge \ \alpha \subseteq s\}|.$$

It can be denoted as $support(\alpha)$ if the sequence database is clear from the context. Given a user-specified positive integer denoted *min_support*, termed the **minimum support** or the **support threshold**, a sequence α is called a **sequential pattern** in sequence database S if $support(S, \alpha) \geq min\_support$. A sequential pattern with length l is called an **l-pattern**. Given a sequence database and the *min_support* threshold, **sequential pattern mining** is to find the complete set of sequential patterns in the database.

2.2. Extending the traditional problem with intra-periodicity

Definition 1 ("." and "_" operators) *Let e and e' be two itemsets that do not contain the underscore symbol (_). Assume that all the items in e' are alphabetically sorted after those in e. Let l (resp. l') denotes itemset e (resp. e') without brackets. Let $\gamma = \prec e_1 \ \ldots \ e_{n-1}a \succ$ and $\mu = \prec be'_2 \ \ldots \ e'_m \succ$ be two sequences, where e_i and e'_i are itemsets that do not contain the underscore symbol, $a \in \{(l), \ (\_l), \ (l\_), \ (\_l\_)\}$ and $b \in \{(l'), \ (\_l'), \ (l'\_), \ (\_l'\_)\}$. The dot operator is defined as follows : (1) $(l).(l') = (l)(l')$, (2) $(l).(\_l') = (ll')$, (3) $(l).(l'\_) = (l)(l'\_)$, (4) $(l).(\_l'\_) = (ll'\_)$, (5) $(l_{)}.(l') = (ll')$, (6) $(i_{)}.(\_l') = (ll')$, (7) $(l\_).(\_l'\_) = (ll'\_)$, (8) $(l\_).(l'\_) = (ll'\_)$, (9) $(\_l).(l') = (\_l)(l')$, (10) $(\_l).(l'\_) = (\_l)(l'\_)$, (11) $(\_l).(\_l'\_) = (\_ll'\_)$, (12) $(\_l).(\_l') = (\_ll')$, (13) $(\_l\_).(l') = (\_ll')$, (14) $(\_l\_).(\_l'\_) = (\_ll'\_)$, (15) $(\_l\_).(l'\_) = (\_ll'\_)$, (16) $(\_l\_).(\_l') = (\_ll')$, (17) $\gamma.\mu = \prec e_1 \ \ldots \ e_{n-1}a.be'_2 \ \ldots \ e'_m \succ$.*

For example, denote $s = \prec$a(abc)(ac)(efgh)$\succ$, we have $s = \prec (a).(a\_).(\_b\_).(\_c).(a\_) .(\_c).(e\_).(\_f\_). \ (\_g\_). \ (\_h) \succ$ and $s = \prec (a) \succ . \prec (a\_) \succ . \prec (\_b\_) \succ . \prec (\_c) \succ . \prec (a\_) \succ . \prec (\_c) \succ . \prec (e\_) \succ . \prec (\_f\_) \succ . \prec (\_g\_) \succ . \prec (\_h) \succ$.

Definition 2 (Prefix and suffix of a sequence) *Consider three sequences α, β and γ such that $\alpha = \beta.\gamma$. Sequence β (resp. γ) is a prefix (resp. suffix) of α*

Definition 3 (Sequence partition) *Given a sequence $s = \prec s_1, s_2, \ldots, s_n \succ$, a partition of s is any subsequence of s which is made of consecutive itemsets of s, i.e. which is on the form $p = \prec s_{j_1}, \ s_{j_1+1}, \ s_{j_1+2} \ \ldots, \ , s_{j_k-1} \ , s_{j_k} \succ$, where $1 \leq j_1 < j_k \leq n$. Partition p is said to be strict if $p \neq s$, i.e. $j_1 > 1$ or $j_k < n$.*

Definition 4 (inclusion-carried mapping) *An inclusion-carried mapping $im_{\alpha,s}$: index(α) $\rightarrow$ index(s) from the set of indexes of the elements sequence $\alpha = \prec \alpha_1, \ \alpha_2, \ \ldots, \alpha_m \succ$ to the set of indexes of the elements of a supersequence $s = \prec s_1, \ s_2, \ \ldots, s_n \succ$ is an injective index-mapping which is (1) is monotonous, i.e. $im_{\alpha,s}(i) < im_{\alpha,s}(i+1), \ 1 \leq i < m$, (2) and such that element α_i of α is mapped to a distinct element $s_{im_{\alpha,s}(i)}$ of s which contains α_i, i.e. $\alpha_i \subseteq s_{im_{\alpha,s}(i)}$.*

For example, let $\alpha = \prec$ (ab)(gh)$\succ$ and $s = \prec$a(abc)(ac)(efgh)$\succ$. Denote $\alpha_1 = (ab)$, $\alpha_2 = (gh)$, $s_1 = (a)$, $s_2 = (abc)$, $s_3 = (ac)$ and $s_4 = (efgh)$. We have $\alpha = \prec \alpha_1\alpha_2 \succ$ and $s = \prec s_1s_2s_3s_4 \succ$. Thus index$(\alpha) = \{1, \ 2\}$ and index$(s) = \{1, \ 2, \ 3, \ 4\}$. Denote $im_{\alpha,s}$: index$(\alpha) \rightarrow$ index(s), $im_{\alpha,s}(1) = 2$ and $im_{\alpha,s}(2) = 4$. Function $im_{\alpha,s}$ is monotonous and $\alpha_i \subseteq s_{im_{\alpha,s}(i)}$ for all $i \in$ index(α). Thus function $im_{\alpha,s}$ is an inclusion-carried mapping from sequence α to s.

For sake of simplicity, when the sets of indexes are known, $im_{\alpha,s}$ will be referred to as an inclusion-carried mapping from sequence α to sequence s.

Lemma 1 (The set inclusion-carried mappings is closed under " $\circ$ " operator) *Consider three sequences $\alpha = \prec \alpha_1,\ \alpha_2,\ \dots, \alpha_m \succ$, β and γ, an inclusion-carried mapping $im_{\alpha,\beta}$ from index(α) to index(β), and another one from index(β) to index(γ), denoted $im_{\beta,\gamma}$. The composition of $im_{\alpha,\beta}$ and $im_{\beta,\gamma}$ defined as $(im_{\alpha,\beta} \circ im_{\beta,\gamma})(i) = im_{\beta,\gamma}(im_{\alpha,\beta}(i)), 1 \leq i \leq m$, is an inclusion-carried mapping from index(α) to index(γ).*

PROOF Function $im_{\alpha,\beta} \circ im_{\beta,\gamma}$ is injective and monotonous as $im_{\alpha,\beta}$ and $im_{\beta,\gamma}$ are injective and monotonous. From definition 4, we have $\alpha_i \subseteq \beta_{im_{\alpha,\beta}(i)} \subseteq \gamma_{im_{\beta,\gamma}(im_{\alpha,\beta}(i))} = \gamma_{(im_{\alpha,\beta} \circ im_{\beta,\gamma})(i)}, 1 \leq i \leq m$.

Definition 5 (The restriction of an inclusion-carried mapping) *Consider three sequences α, β and s such that $\alpha \subseteq \beta$ and $\beta \subseteq s$, and an inclusion-carried mapping $im_{\beta,s}$ from index(β) to index(s). A restriction of $im_{\beta,s}$ to index(α), denoted $im_{\alpha,\beta,s}$, is the composition of an inclusion-carried mapping from index(α) to index(β), denoted $im_{\beta,s}$, and $im_{\beta,s}$, i.e. $im_{\alpha,\beta,s} = im_{\alpha,\beta} \circ im_{\beta,s}$: index(α) $\rightarrow$ index(β) $\rightarrow$ index(s).*

A restriction of $im_{\beta,s}$ to index(α) is unique if β contains only one occurrence of α.

Definition 6 (Sequence intra-periodicity) *Let s and $\alpha = \prec \alpha_1,\ \alpha_2,\ \dots, \alpha_m \succ$, $m > 1$, be two sequences such that $\alpha \subseteq s$. Consider an inclusion-carried mapping $im_{\alpha,s}$ from index(α) to index(s). The set of intra-periods of α in s following mapping $im_{\alpha,s}$, also called periods of appearance of the elements of α in s with respect to $im_{\alpha,s}$, is defined as $ips(\alpha, s, im_{\alpha,s}) = \{im_{\alpha,s}(i+1) - im_{\alpha,s}(i) \mid i \in \{1,\ 2,\ \dots, m-1\}\}$.*

Definition 7 (Minimal and maximal intra-periodicities) *Consider two sequences α and s such that $\alpha \subseteq s$. The minimal (resp. maximal) intra-periodicity of α in s following an iclusion-carried mapping $im_{\alpha,s}$ is the minimal (resp. maximal) value of $ips(\alpha, s, im_{\alpha,s})$.*

Definition 8 (Sequence inclusion) *Denote $i1$ (resp. $i2$) the minimal (resp. maximal) intra-periodicity threshold. A sequence α is contained in another sequence s following $(i1, i2)$, denoted as $\alpha \subseteq^{(i1,i2)} s$, if there exists an inclusion-carried mapping $im_{\alpha,s}$ from index(α) to index(s) such that $i1 \leq min(ips(\alpha, s, im_{\alpha,s}))$ and $max(ips(\alpha, s, im_{\alpha,s})) \leq i2$. Sequence α is called (i1, i2)-subsequence of s, and s is called (i1, i2)-supersequence of α.*

Lemma 2 (Distributivity of $\subseteq^{(i1,i2)}$ operator) *Consider a couple of intra-periodicity thresholds $(i1, i2)$ and three sequences α, β and s that do not contain the underscore operator (_). If $\alpha.\beta \subseteq^{(i1,i2)} s$ then there exist three sequences $\alpha\prime$, μ and $\beta\prime$ such that $s = \alpha\prime.\mu.\beta\prime$, $\alpha \subseteq^{(i1,i2)} \alpha\prime$ and $\beta \subseteq^{(i1,i2)} \beta\prime$*

PROOF Assume that $\alpha.\beta \subseteq^{(i1,i2)} \gamma$. From definition 8, this means that there exists an inclusion-carried mapping $im_{\alpha.\beta,s}$ from $\alpha.\beta$ to s such that $i1 \leq min(ips(\alpha.\beta, s, im_{\alpha.\beta,s}))$ and $max(ips(\alpha.\beta, s, im_{\alpha.\beta,s})) \leq i2$. Denote $\alpha\prime = \prec s_1,\ \dots s_{im_{\alpha.\beta,s}(1)},\ \dots, s_{m_{\alpha.\beta,s}(||\alpha||)} \succ$, $\mu = \epsilon$ if $im_{\alpha.\beta,s}(||\alpha|| + 1) = (m_{\alpha.\beta,s}(||\alpha||) + 1)$ and $\mu = \prec s_{im_{\alpha.\beta,s}(||\alpha||)+1)},\ \dots,\ s_{im_{\alpha.\beta,s}(||\alpha||+1)-1)} \succ$ otherwise, and $\beta\prime = \prec s_{im_{\alpha.\beta,s}(||\alpha||+1)},\ \dots, s_{im_{\alpha.\beta,s}(||s||)} \succ$. We have $s = \alpha\prime.\mu.\beta\prime$, $\alpha \subseteq^{(i1,i2)} \alpha\prime$ and $\beta \subseteq^{(i1,i2)} \beta\prime$. Hence the result.

Definition 9 (irreducible supersequence) *A supersequence s of another sequence α is said to be irreducible following α and a couple of intra-periodicity thresholds (i1, i2) if no strict partition of s is a (i1, i2)-supersequence of α.*

Definition 10 (Sequence support) *The support of a sequence α in a dataset S following a couple (i1, i2) of intra-periodicity thresholds, denoted $support_{i1,i2}(S,\alpha)$, is defined as the number of sequences of S which contain α following (i1, i2), i.e. $support_{i1,i2}(S,\alpha) = |\{\prec sid, s \succ \in S \mid \alpha \subseteq^{(i1,i2)} s\}|$.*

Definition 11 (Intra-periodic frequent sequence) *Given a minimum support threshold $minS$, and a couple $(i1, i2)$ of intra-periodicity thresholds, a sequence α is an Intra-Periodic Frequent Sequence (IPFS) frequent if $support_{i1,i2}(s,\alpha) \geq minS$.*

Definition 12 (Problem definition) *Let there be a user-specified database D and three thresholds $i1 \geq 0$, $i2 \geq 0$, $minS \geq 0$. The problem of mining intra-periodic frequent sequences is to find all IPFS in D.*

3. Search space pruning and partitioning

Due to space restriction, the proofs of lemmas are removed.

Lemma 3 (Intra-periodicity-set stability/growth based itemset growth/addition) *Let α, β and s be three sequences such that $\beta \subseteq s$. Assume that $\beta = its1\_.\alpha.\_its2$ or $\beta = its1.\alpha.\_its2$ or $\beta = its1\_.\alpha.its2$ or $\beta = its1.\alpha.its2$, where $its1$ and $its2$ are two itemsets, and α, $its1$ and $its2$ do not contain the underscore operator(_). Given an inclusion-carried mapping $im_{\beta,s}$ from $index(\beta)$ to $index(s)$, there exists a restriction of $im_{\beta,s}$ to $index(\alpha)$, denoted $im_{\alpha,\beta,s}$, such that $ips(\alpha, s, im_{\alpha,\beta,s}) = ips(\beta, s, im_{\beta,s})$ if $\beta = its1\_.\alpha.\_its2$ and $ips(\alpha, s, im_{\alpha,\beta,s}) \subseteq ips(\beta, s, im_{\beta,s})$ otherwise.*

Lemma 4 (Intra-periodicity-set growth based prefix-suffix growth) *Consider three sequences α, β and s such that $\beta \subseteq s$ and $\beta = \mu.\alpha.\gamma$ where μ and γ denote sequences that may contain the underscore operator (_). Given an inclusion-carried mapping $im_{\beta,s}$ from $index(\beta)$ to $index(s)$, there exists a restriction of $im_{\beta,s}$ to $index(\alpha)$, denoted $im_{\alpha,\beta,s}$, such that $ips(\alpha, s, im_{\alpha,\beta,s}) \subseteq ips(\beta, s, im_{\beta,s})$.*

Lemma 5 (Search space pruning using intra-periodicity thresholds) *Denote $minip$ (resp. $maxip$) the minimal (resp. maximal) intra-periodicity threshold. Consider three sequences α, β and s such that $\beta \subseteq s$ and $\beta = \mu.\alpha.\gamma$ where μ and γ denote sequences that may contain the underscore operator (_). We have :*

1) If $min(ips(\alpha, s, im_{\alpha,s})) < minip$ for any inclusion-carried mapping $im_{\alpha,s}$ from $index(\alpha)$ to $index(s)$ then $min(ips(\beta, s, im_{\beta,s})) < minip$ for any inclusion-carried mapping $im_{\beta,s}$ from $index(\beta)$ to $index(s)$.

2) If $max(ips(\alpha, s, im_{\alpha,s})) > maxip$ for any inclusion-carried mapping $im_{\alpha,s}$ from $index(\alpha)$ to $index(s)$ then $max(ips(\beta, s, im_{\beta,s})) > maxip$ for any inclusion-carried mapping $im_{\beta,s}$ from $index(\beta)$ to $index(s)$.

Lemma 6 (Search space pruning using $\subseteq^{(i1,i2)}$ operator) *Consider three sequences α, β and s such that $\beta = \mu.\alpha.\gamma$, where μ and γ are two sequences, and two inta-periodicity thresholds i1 and i2, we have: (1) if $\beta \subseteq^{(i1,i2)} s$ then $\alpha \subseteq^{(i1,i2)} s$, (2) if $\alpha \not\subseteq^{(i1,i2)} s$ then $\beta \not\subseteq^{(i1,i2)} s$.*

Lemma 7 (Anti-monotonicity of the support following prefix-suffix growth) *Given a couple (i1, i2) of intra-periodicity thresholds and two sequences α and s such that $s = \mu.\alpha.\gamma$, we have: $support_{i1,i2}(S,\alpha) \geq support_{i1,i2}(S,s)$.*

Given two sequences s and α, and a couple $(i1, i2)$ of intra-periodicity thresholds, $lmip(s, \alpha, i1, i2)$ denotes the leftmost partition of s which is irreducible following α and $(i1, i2)$. Such a partition may not exist, and in this case, we assume that $lmip(s, \alpha, i1, i2) = \epsilon$. If such a partition exists, it induces a decomposition of sequence s into three parts, (1) the left part, denoted $lp(s, \alpha, i1, i2)$, (2) the middle part, denoted $lmip(s, \alpha, i1, i2)$, (3) and the right part, denoted $rp(s, \alpha, i1, i2)$. We have $s = lp(s, \alpha, i1, i2)$. $lmip(s, \alpha, i1, i2)$. $rp(s, \alpha, i1, i2)$. Denote $q(s, \alpha, i1; i2)$ = $lmip(s, \alpha, i1, i2)$ $.rp(s, \alpha, i1, i2)$ the concatenation of the middle and right parts. If $lmip(s, \alpha, i1, i2) = \epsilon$, we set $lp(s, \alpha, i1, i2) = \epsilon$, $rp(s, \alpha, i1, i2) = \epsilon$ and $q(s, \alpha, i1, i2) = \epsilon$.

The projection of dataset S following sequence α and the couple of intra-periodicity thresholds $(i1, i2)$, denoted $S(\alpha, i1, i2)$, is the set obtained by removing the left part of any sequence for which the middle part exists : $S(\alpha, i1, i2) = \{\prec sid, lmip(s, \alpha, i1, i2) .rp(s, \alpha, i1, i2) \succ \ \mid \ \prec sid, s \succ \in S \ and \ lmip(s, \alpha, i1, i2) \neq \epsilon\}$. If the couple (i1, i2) is known, they could be removed from the notation of projected databases, i.e. $S(\alpha, i1, i2) = S(\alpha)$, Note that, this definition is slightly different from the one introduced in [2].

Lemma 8 (Search-space partitioning based on prefix) *We have the following:*

1) Let $\{x_1,\ x_2,\ \ldots,\ x_n\}$ be the complete set of length-1 intra-periodic frequent sequences in a sequence database S. The complete set of sequential patterns in S can be divided into n disjoint subsets based on prefix-items. The i-th ($1 \leq i \leq n$) subset of the search-space partitioning is the set of intra-periodic frequent sequences with prefix x_i.

2) Let α be a length-l intra-periodic frequent sequence and $\{\beta_1, \beta_2,\ \ldots,\ \beta_p\}$ be the complete of length-(l+1) intra-periodic frequent sequences with prefix α. The complete set of intra-periodic frequent sequences with prefix α, except for α itself, can be divided into p disjoint subsets. The i-th subset ($1 \leq i \leq p$) is the set of intra-periodic frequent sequences prefixed with β_i.

4. The IPFSM algorithm

Algorithm 1 Intra-Periodic Frequent Sequence Miner. The initial call is IPFSM(S, ϵ, minS, minip, maxip) with S as the initial dataset

```
function IPFSM(Dataset $S$, Prefix $\alpha$, float minS, int minip, int maxip)
    $X \leftarrow \{$Item x $\mid minS \leq |\{s \in S \mid lmip(s, \alpha.x, minip, maxip) \neq \epsilon\}|\}$
    Comment: Item x may contains the underscore operator (_)
    for all $x_i \in X$ do
        SaveIntraPeriodicFrequentSequence($\alpha.x_i$)
    end for
    for all $x_i \in X$ do
        IPFSM($S(\alpha.x_i, minip, maxip), \alpha.x_i$, minS, minip, maxip)
    end for
end function
```

In this section, we translate the study made in section 2 into a function called Intra Periodic Frequent Sequence Miner (IPFSM). It is presented in algorithm 1. A IPFSM call

(1) takes as arguments a database S, the current prefix value, the minimal support threshold, the minimal and maximal intra-periodicity thresholds, (2) searches for the complete list $X = \{x_1,\ x_2,\ \dots,\ x_p\}$ of all the length-1 sequential patterns of S which are such that $\alpha.x_i,\ i \in \{1,\ 2,\ \dots,\ p\}$, are frequent intra-periodic sequences of S, (4) saves $\alpha.x_i$ as a new sequential pattern for each pattern x_i found, assuming that the current prefix is α, (5) constructs, following lemma 8, a new database $S(\alpha.x_i, minip, maxip)$ for each length-1 pattern $x_i \in X$ found, and (6) makes a recursive call per new constructed database with $\alpha.x_i$ as the new current prefix value.

Function IPFSM recursively generates sub-databases from a partitioning of the current database following lemma 8. We consider that initial database, denoted S, is of depth 0. The initial database is used to generate databases of depth-1 dadabases of the form $s(y_1)$, where y_1 is an item. The depth-1 database $S(y_1)$ is used to generate depth-2 dadabases of the form $S(y_1)(y_2)$, where y_2 is an item. A generated database is of depth d if it has been constructed using d length-1 patterns. Such a database is denoted $S(y_1.y_2\ ...\ y_d)$, where y_1, y_2, ... , y_d are the length-1 patterns used to construct that database step by step in this order. Database $S(y_1.y_2\ ...\ x_d)$, $d > 1$, is generated from $S(y_1.y_2\ ...\ x_{d-1})$ In terms of IPFSM calls, the initial call, i.e. IPFSM(S, ϵ, minS, minip, maxip), if of depth 0. The depth of a IPFSM call is the depth of its database argument. This depth is equal to the length of its prefix argument.

5. Experimental evaluation

We consider four real live data sets collected from the webpage (http://www.philippe-fournier-viger. com/spmf/index.php) of SPMF software [12]. This webpage provides large data sets in SPMF format that are often used in the data mining litterature for evaluating and comparing algorithm performance. All experiments are done on a 4-cores of 2.16GHz Intel(R) Pentium(R) CPU N3530 with 4 gigabytes main memory, running Ubuntu 18.04 LTS. All the algorithms are implemented in Java and grounded on SPMF software [12].

For each data set, we consider a number of support thresholds. For each support threshold, we fix the minimal intra-periodicity threshold at zero (0), initialize the maximal intra-periodicity threshold at zero (0) and run algorithm IPFSM while increasing the maximal intra-periodicity threshold until all the frequent sequences are found. The experiments presented in the annex section show that the number of intra-periodic frequent sequences, the runtime and the memory usage increase with the maximal intra-periodicity threshold for a given support threshold.

6. conclusion

Previous work in frequent sequence mining and periodicty only consider extra periodicity. In this paper, we have formalised the problem of mining intra-periodic frequent sequences and studied its related issues, namely search space pruning and partitioning. This study has enabled us to design a new efficient algorithm called Intra-Periodic Frequent Sequence Miner (IPFSM). Experimental results confirm its efficiency. In future work, we will consider adapting the proposed model for various pattern structures.

7. References

[1] CHIA-YING HSIEH , DON-LIN YANG , JUNGPIN WU, “An Efficient Sequential Pattern Mining Algorithm Based on the 2-Sequence Matrix”, *Workshops Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, 583–591, 2008.

[2] JIAN PEI , JIAWEI HAN , BEHZAD MORTAZAVI-ASL , JIANYONG WANG , HELEN PINTO , QIMING CHEN , UMESHWAR DAYAL , MEICHUN HSU, “Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach”, *IEEE Trans. Knowl. Data Eng.*, vol. 16, num. 11, 1424–1440, 2004.

[3] JIAWEI HAN , MICHELINE KAMBER, “Data Mining: Concepts and Techniques”, *Morgan Kaufmann*, 2000.

[4] JIAWEI HAN , JIAN PEI , YIWEN YIN, “Mining Frequent Patterns without Candidate Generation”, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, 1–12, 2000.

[5] KARAM GOUDA , MOSAB HASSAAN , MOHAMMED J. ZAKI, “Prism: An effective approach for frequent sequence mining via prime-block encoding”, *J. Comput. Syst. Sci.*, vol. 276, num. 1, 88–102 2010.

[6] KENMOGNE EDITH BELISE , TADMON CALVIN , ROGER NKAMBOU, “A pattern growth-based sequential pattern mining algorithm called prefixSuffixSpan”, *EAI Endorsed Trans. Scalable Information Systemsurnal*, vol. 4, num. 12, e4, 2017.

[7] KENMOGNE EDITH BELISE, “Contribution to the sequential and parallel discovery of sequential patterns with an application to the design of e-learning recommenders”, *PhD Thesis. The University of Dschang, Faculty of Sciences, Department of Mathematics and Computer Science*, October 2018.

[8] LIONEL SAVARY , KARINE ZEITOUNI, “Indexed Bit Map (IBM) for Mining Frequent Sequences”, *Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*, 659–666, 2005.

[9] MOHAMMED JAVEED ZAKI, “TSPADE: An Efficient Algorithm for Mining Frequent Sequences”, *Machine Learning*, vol. 42, num. 1/2, 31–60 2001.

[10] MOHAMMED JAVEED ZAKI, “Parallel Sequence Mining on Shared-Memory Machines, *J. Parallel Distrib. Comput.*, vol. 61, num. 3, 401–426 2001.

[11] NIZAR R. MABROUKEH , CHRISTIE I. EZEIFE, “A taxonomy of sequential pattern mining algorithms”, *ACM Comput. Surv.*, vol. 43, num. 1, 3 2010.

[12] PHILIPPE FOURNIER-VIGER , ANTONIO GOMARIZ , TED GUENICHE , AZADEH SOLTANI , CHENG-WEI WU , VINCENT S. TSENG, “SPMF: a Java open-source pattern mining library”, *Journal of Machine Learning Research*, vol. 15, num. 1, 3389–3393 2014.

[13] RAKESH AGRAWAL , RAMAKRISHNAN SRIKANT, “Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan”, *Mining Sequential Patterns*, 3–14, 1995.

[14] SABEUR ARIDHI , LAURENT D’ORAZIO , MONDHER MADDOURI , ENGELBERT MEPHU NGUIFO, “Density-based data partitioning strategy to approximate large-scale subgraph mining”, *Inf. Syst.*, vol. 48, 213–223 2015.

[15] SHENGNAN CONG , JIAWEI HAN , JAY HOEFLINGER , DAVID A. PADUA, “A sampling-based framework for parallel data mining”, *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP 2005, June 15-17, 2005, Chicago, IL, USA*, 255–265, 2005.

[16] VALERIE GURALNIK , GEORGE KARYPIS, "Parallel tree-projection-based sequence mining algorithms", *Parallel Computing*, vol. 30, num. 4, 443–472 2004.

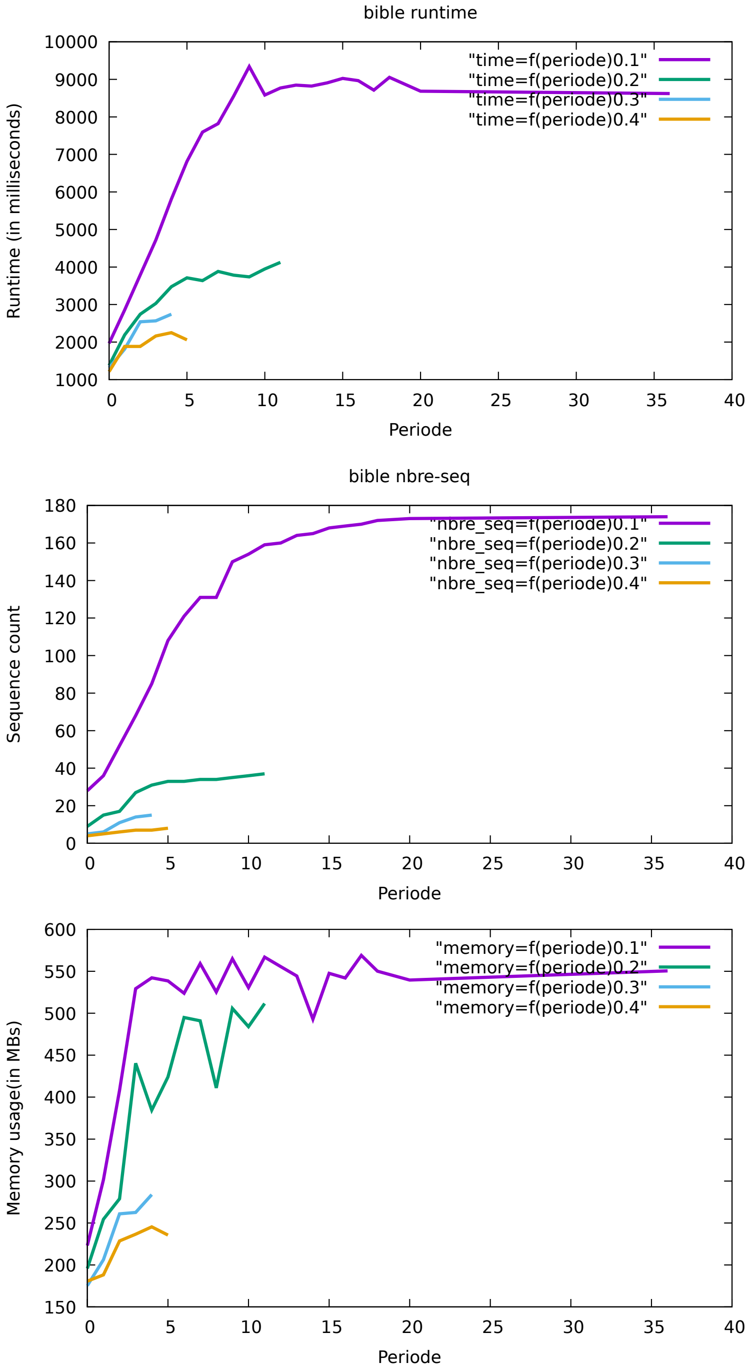
[17] ZHENGLU YANG , YITONG WANG , MASARU KITSUREGAWA, "LAPIN: Effective Sequential Pattern Mining Algorithms by Last Position Induction for Dense Databases", *Advances in Databases: Concepts, Systems and Applications, 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007, Proceedings*, 1020–1023 2007.

[18] PHILIPPE FOURNIER-VIGER , ZHITIAN LI , , JERRY CHUN-WEI LI, , RAGE UDAY KIRAN , , HAMIDO FUJITA, " EFFICIENT ALGORITHMS TO IDENTIFY PERIODIC PATTERNS IN MULTIPLE SEQUENCES, *Inf. Sci.*, 489, 205–226, 2019.

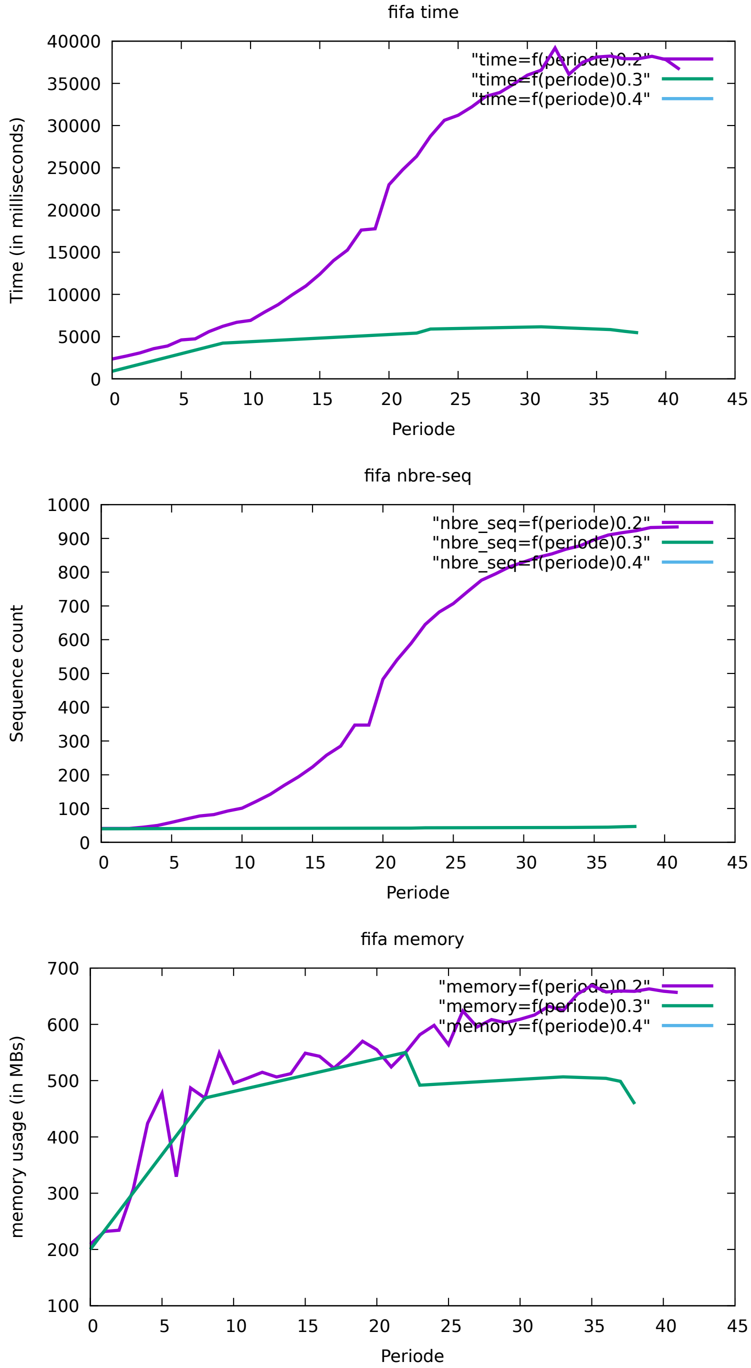
[19] DUY-TAI DINH , BAC LE, , PHILIPPE FOURNIER-VIGER , , VAN-NAM HUYNH, " An efficient algorithm for mining periodic high-utility sequential patterns, *Appl. Intell.*, 48, 12, 4694–4714,2018

[20] , J. N. Venkatesh , R. UDAY KIRAN , , P. KRISHNA REDDY , , MASARU KITSUREGAWA, " Discovering Periodic-Correlated Patterns in Temporal Databases, *T. Large-Scale Data- and Knowledge-Centered Systems*, 38, 146–172, 2018.

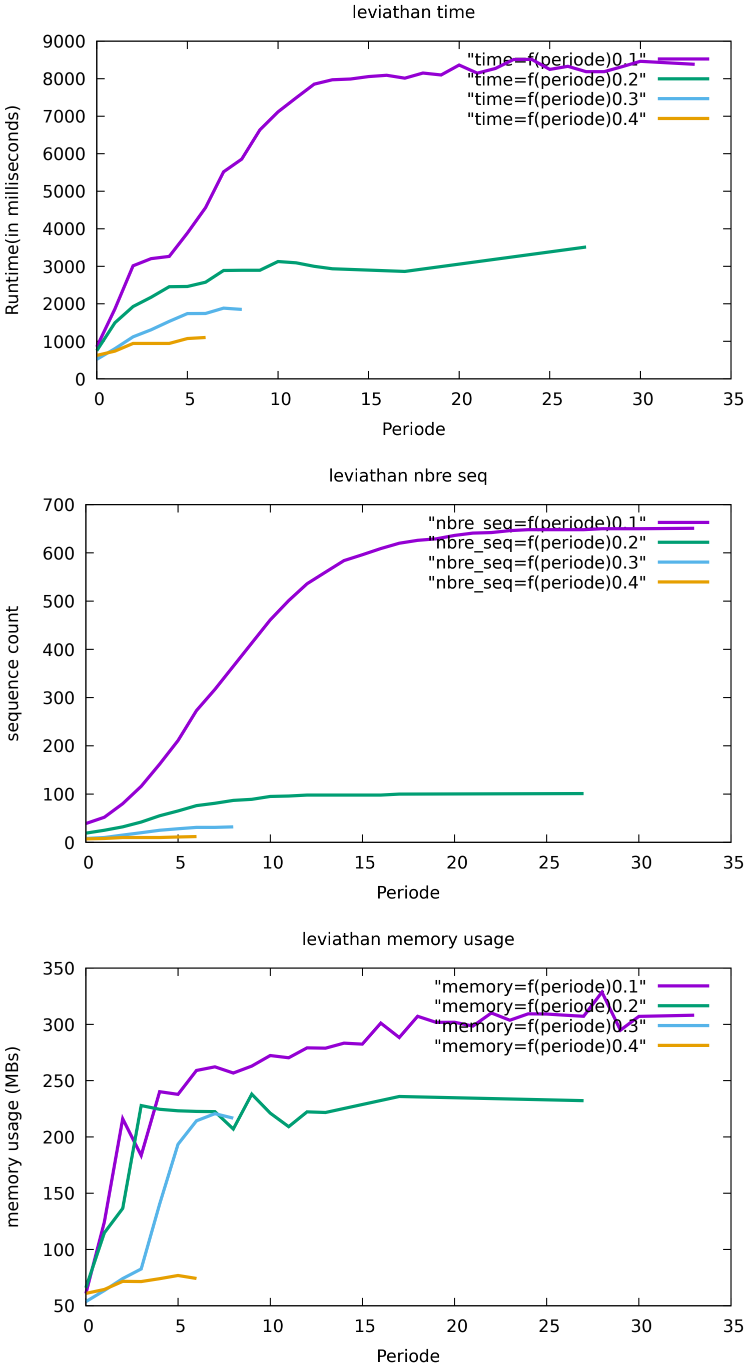
8. Annex



Performance analysis of IPFSM on the real-life data set BIBLE



Performance analysis of IPFSM on the real-life data set FIFA



Performance analysis of IPFSM on the real-life data set LEVIATHAN