Mining Periodic Frequent Patterns Common to Multiple Sequences

Zhitian Li, Philippe Fournier-Viger et al.

1. Introduction

<{*a*, *b*, *c*}, {*b*, *d*}, {*a*, *b*, *e*}, {*c*}, {*a*, *b*, *d*, *e*}, {*a*, *c*}, {*b*, *c*}, {*b*, *e*}>

An example sequence.

Frequent pattern mining: Discovering frequent itemsets that appear in at least ***minSup*** transactions in a sequence.

If ***minSup*** = **2**, the itemset **{*a*, *b*}** is a frequent pattern since it appears **3** times in the sequence.

1. Introduction

Periodic pattern mining:

Discover frequent patterns that appear at a time interval.

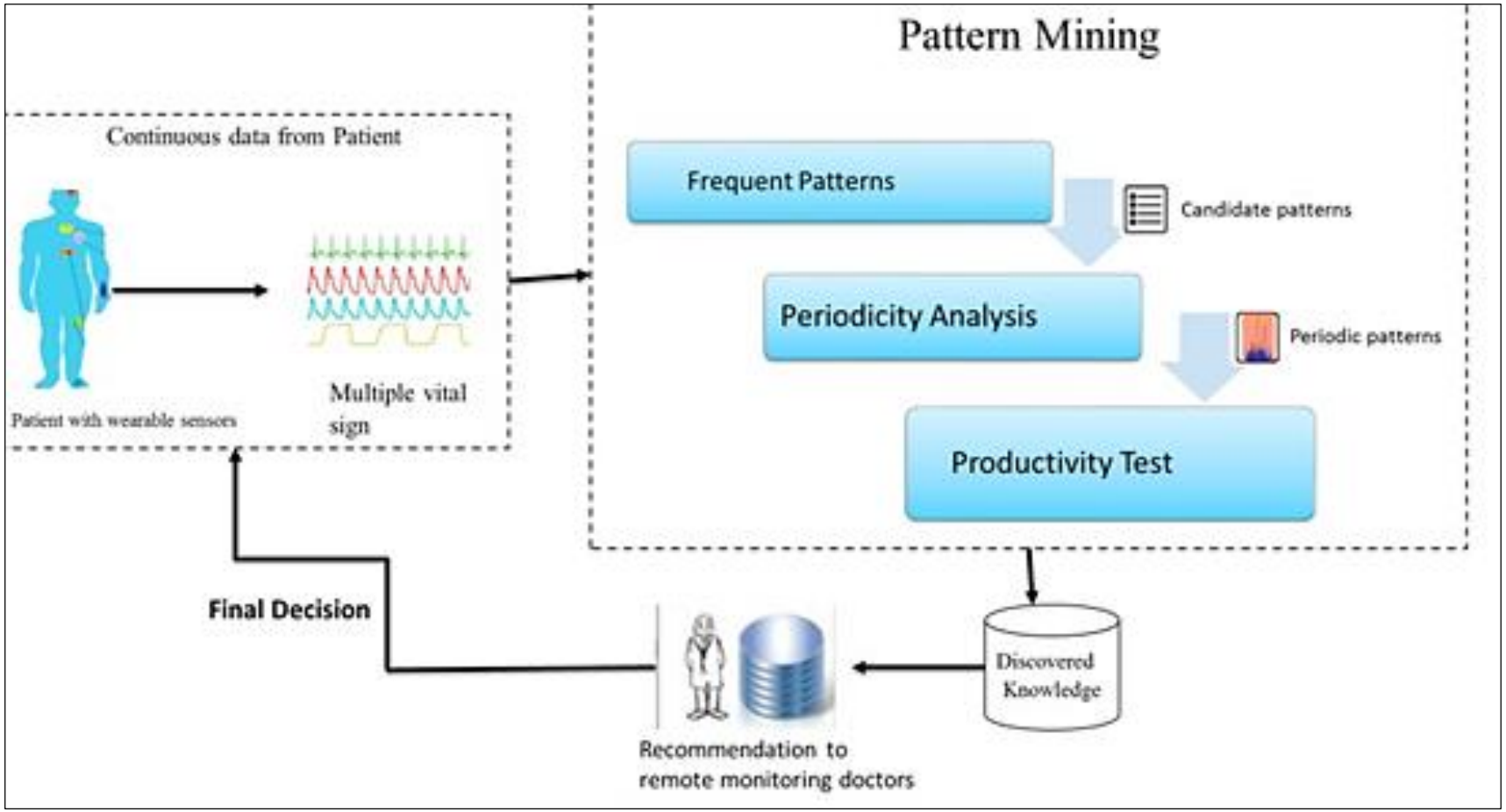
Periodicity 

<{*a*, *b*, *c*}, {*b*, *d*}, {*a*, *b*, *e*}, {*c*}, {*a*, *b*, *d*, *e*}, {*a*, *c*}, {*b*, *c*}, {*b*, *e*}>

An example sequence.

Periods of itemset {*a*, *b*}: it appears in the 1st, 3rd, and 5th transactions, the periods are (3 – 1 = 2) and (5 – 3 = 2), respectively.

1. Introduction



Reference:

Nishi, M A, Ahmed, C F, Samiullah, M, et al. Effective periodic pattern mining in time series databases [J]. Expert Systems with Applications. 2013, 40: 3015–3027.

1. Introduction

Previous measure: ***maxPr*** (maximum periodicity). A pattern is periodic if each of its periods is not greater than ***maxPr***.

A problem: this condition is **too strict**.

- If ***maxPr*** is set to a large number, then many unimportant patterns may be found;
- If it is set to a small number, then many interesting information will be eliminated.

1. Introduction

Another problem:

Most algorithms find patterns in **a single sequence**.

But periodic patterns also commonly appear in **sequence databases**.

An example sequence database.

Sequence_id	Sequence
0	<{*a*, *b*, *e*}, {*a*, *b*, *e*}, {*a*, *d*}, {*a*, *e*}, {*a*, *b*, *c*}>
1	<{*c*}, {*a*, *b*, *c*, *e*}, {*c*, *d*}, {*a*, *b*, *c*, *e*}, {*a*, *b*, *d*}>
2	<{*b*, *c*}, {*a*, *b*}, {*a*, *c*, *d*}, {*a*, *c*}, {*a*, *b*}>
3	<{*a*, *b*, *d*, *e*}, {*a*, *b*, *e*}, {*a*, *b*, *c*}, {*a*, *b*, *d*, *e*}, {*a*, *b*}>

1. Introduction

Dinh et al (2017):

- Proposed the **PHUPSM** algorithm to mine periodic patterns with high utility in a sequence database.
- But the algorithm does not consider whether a pattern is periodic in each sequence.

Reference:

Dinh, T, Huynh V N, Lee, B. Mining Periodic High Utility Sequential Patterns [C]// Proceedings of Ninth Asian Conference on Intelligent Information and Database Systems. Springer, 2017: 545–555.

2. Problem Statement

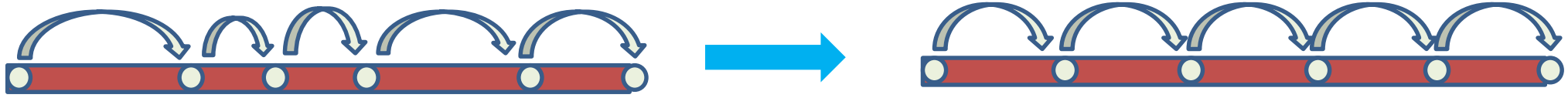
New measures

Standard deviation of periods

stanDev(X, s): the standard deviation of the periods of a frequent itemset ***X*** in a sequence ***s***.

maxStanDev: If ***stanDev(X, s)*** <= ***maxStanDev*** , ***X*** is a periodic pattern in sequence ***s***.

Why? The *maxPr* condition is too strict. If the period standard deviation of a frequent itemset is small, it means that this pattern always appears at a certain time interval.



2. Problem Statement

New measures

Sequence Periodic Ratio (SPR)

$$ra(X) = numSeq(X) \,/\, |D|$$

$$numSeq(X) = |\{s | maxPr(X, s) \leq \mathbf{maxPr} \wedge sup(X, s) \geq \mathbf{minSup} \wedge stanDev(X, s) \leq \mathbf{maxStd} \wedge s \in D\}|$$

$|D|$: The number of sequences in database D.

An itemset X is a **P**eriodic **F**requent **P**attern common to multiple **S**equences (**PFPS**) in D if $ra(X) \geq minRa$.

3. Pruning Properties

Search space
is too big!

New measure
(Upper bound)!

3. Pruning Properties

New upper-bound

$$\textbf{\textit{boundRa}}(X) = numCand(X)/|D|$$

$$numCand(X) = |\{s|maxPr(X,s) \leq maxPr \wedge sup(X,s) \geq minSup \wedge s \in D\}|$$

$|D|$: The number of sequences in D.

boundRa is an upper bound on ***ra***. It does not consider *maxStd*.

3. Pruning Properties

Pruning properties

Pruning property 1

If $boundRa(X') < minRa$,

Then X' and any superset $X \supset X'$ are not PFPS.

X, X' and $X^{"}$: itemsets.

Pruning property 2

If $\exists X^{"} \subset X'$ such that $boundRa(X^{"}) < minRa$,

Then X' is not a PFPS and its supersets are not PFPS.

4. The Algorithms

The PFPS-list data structure

PFPS-list
- *i-set*
- *sid-list*
- *tidlist-list*

The PFPS-list of itemset {*a*}

i-set	***{a}***
sid-set	{0, 1, 2, 3}
tidlist-list	{{0, 1, 2, 3, 4}, {1, 3, 4}, {1, 2, 3, 4}, {0 ,1, 2, 3, 4}}

The PFPS-list of itemset {*e*}

i-set	***{e}***
sid-set	{0, 1, 3}
tidlist-list	{{0, 1, 3}, {1, 3, 4}, {0 ,1, 3}}

4.2 The MPFPS Algorithm

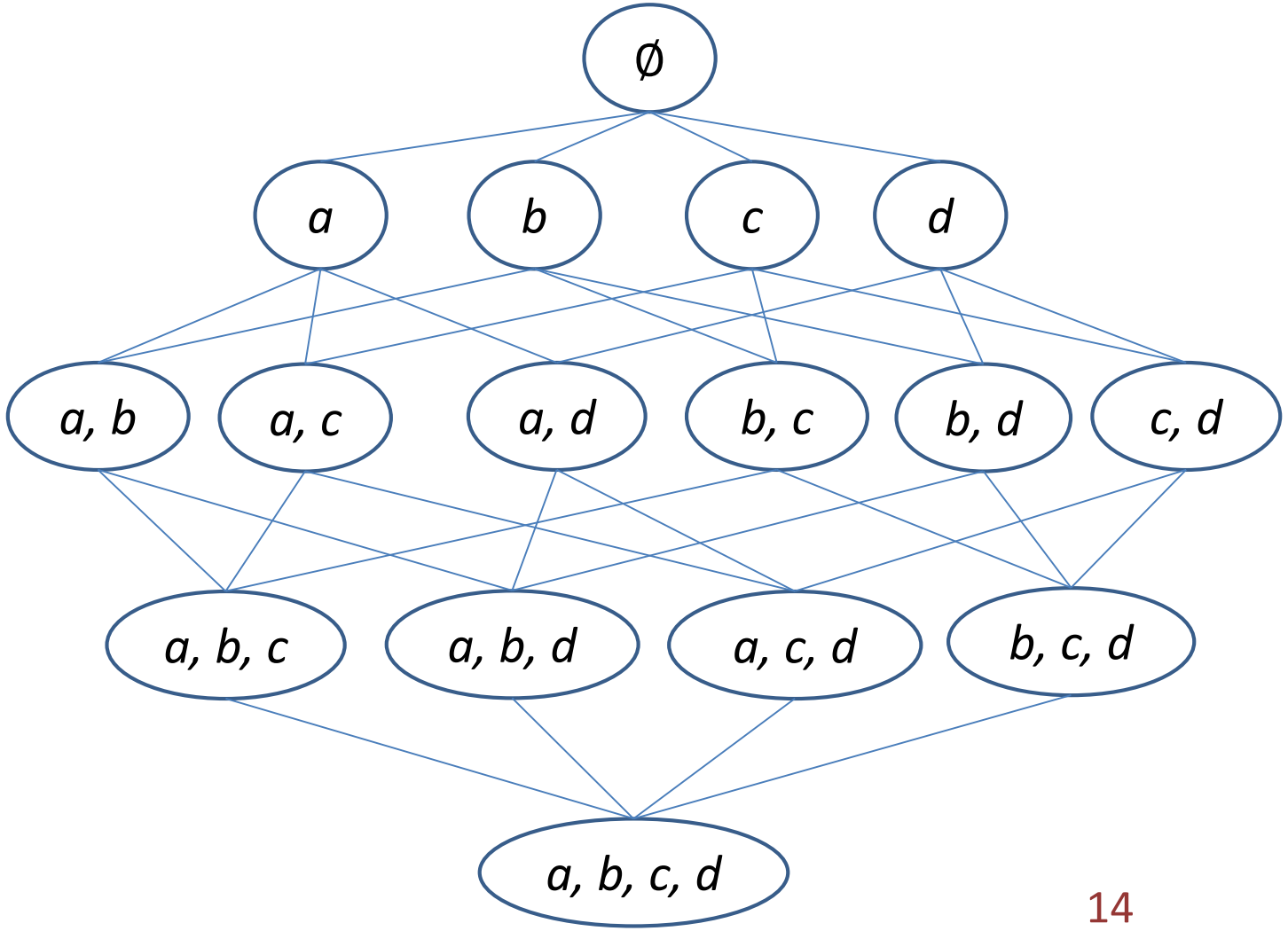
Sequence
<{**a**, *b*, *e*}, {**a**, *b*, *e*}, {**a**, *d*}, {**a**, *e*}, {**a**, *b*, *c*}>

The sequence is represented in a vertical format.

Item	Transactions
a	**T0, T1, T2, T3, T4**
b	T0, T1, T4
c	T4
d	T2
e	T0, T1, T3

{*a*}	∩	{*b*}	→	{*a*, *b*}
T0		T0		T0
T1		T1		T1
T2		T4		T4
T3				
T4				

4.1 The $MPFPS_{BFS}$ Algorithm

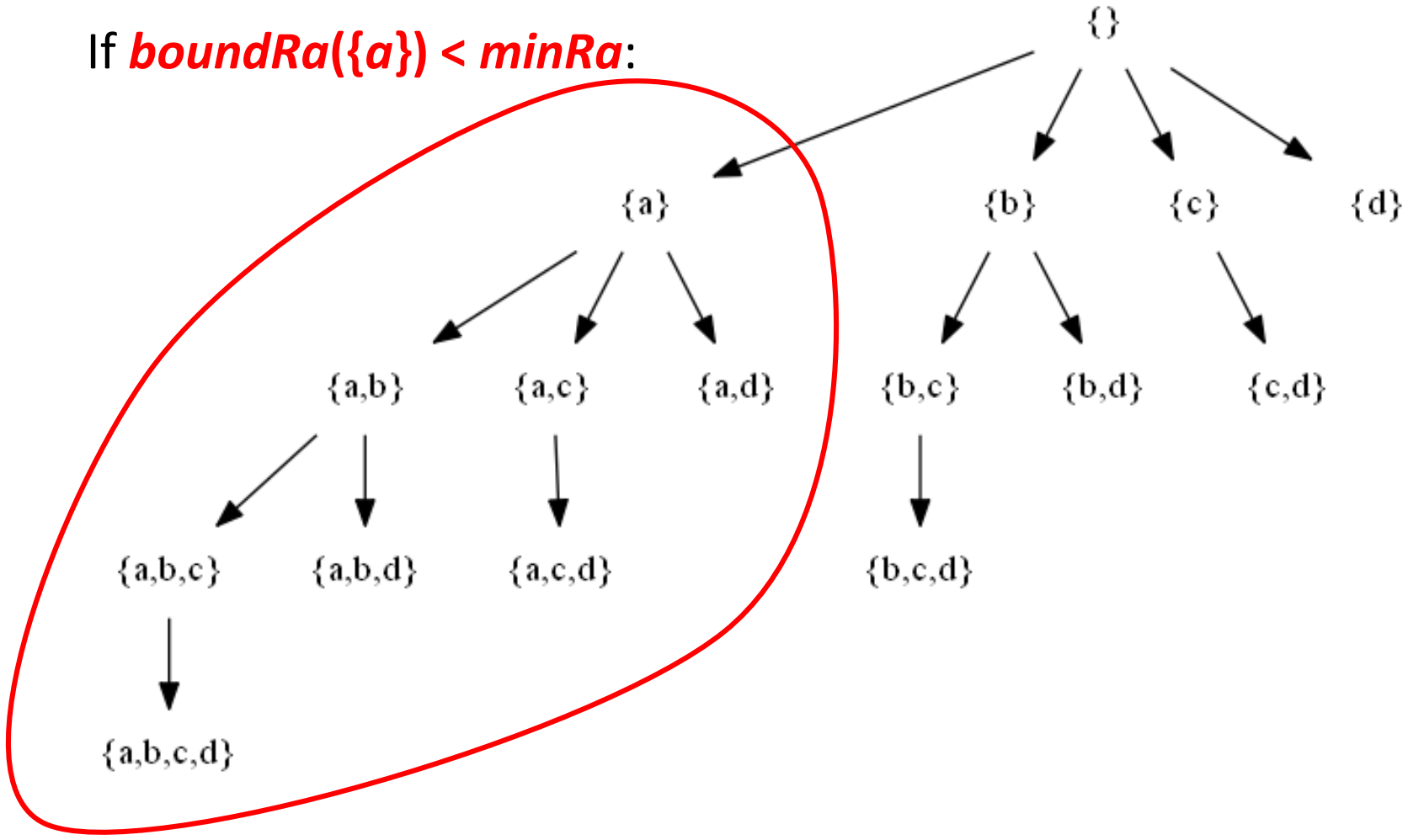


The algorithm explores the search space in a **breadth-first way**.

It applies the two pruning properties to reduce the search space.

4.2 The MPFPS$_{DFS}$ Algorithm

If ***boundRa({a}) < minRa***:



The algorithm explores the search space in a **depth-first way**.

It applies the two pruning properties to reduce the search space.

4. The MPFPS Algorithm

MPFPS has four parameters: *minSup*, *maxStd*, *maxPr*, and *minRa*.

Patterns found for different threshold values

No.	*minSup*	*maxPr*	*maxStd*	*minRa*	Patterns Found
1	2	3	1.0	0.6	{*a*}, {*e*}, {*a*, *e*}
2	**3**	3	1.0	0.6	{*a*}
3	2	**1**	1.0	0.6	{*a*}
4	2	3	**1.5**	0.6	{*a*}, {*b*}, {*e*}, {*a*, *b*}, {*a*, *e*}
5	2	3	1.0	**0.4**	{*a*}, {*b*}, {*c*}, {*e*}, {*a*, *b*}, {*a*, *e*}, {*b*, *e*}, {*a*, *b*, *e*}

5. Experiments

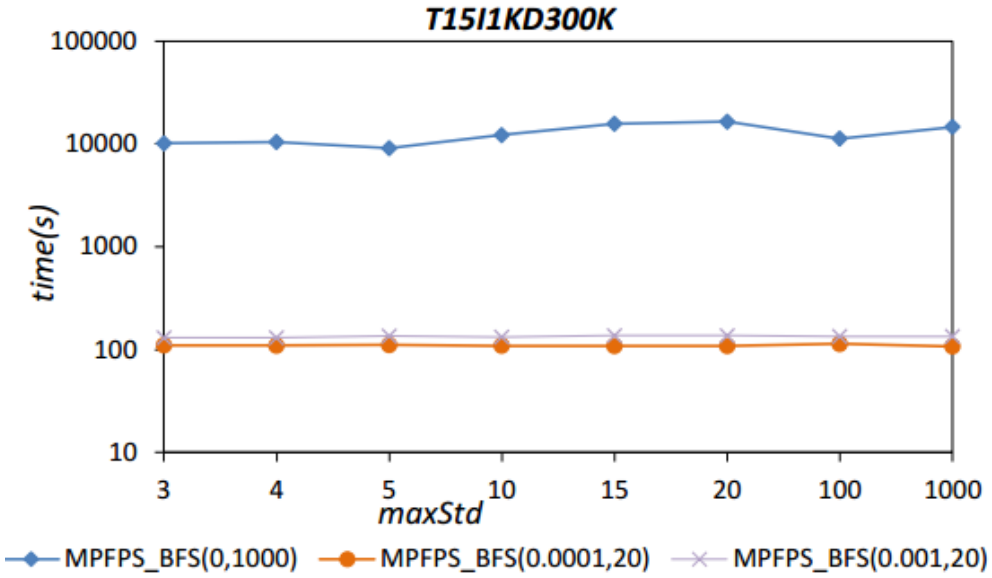
- We compared:
  - **MPFPS(x,y)** denotes MPFPS with *minRa* = x, *maxPr* = y and *minSup* = 2.
  - **Baseline**: MPFPS algorithm with *maxStd* = 1000, *minRa* = 0 and *maxPr* = 1000.

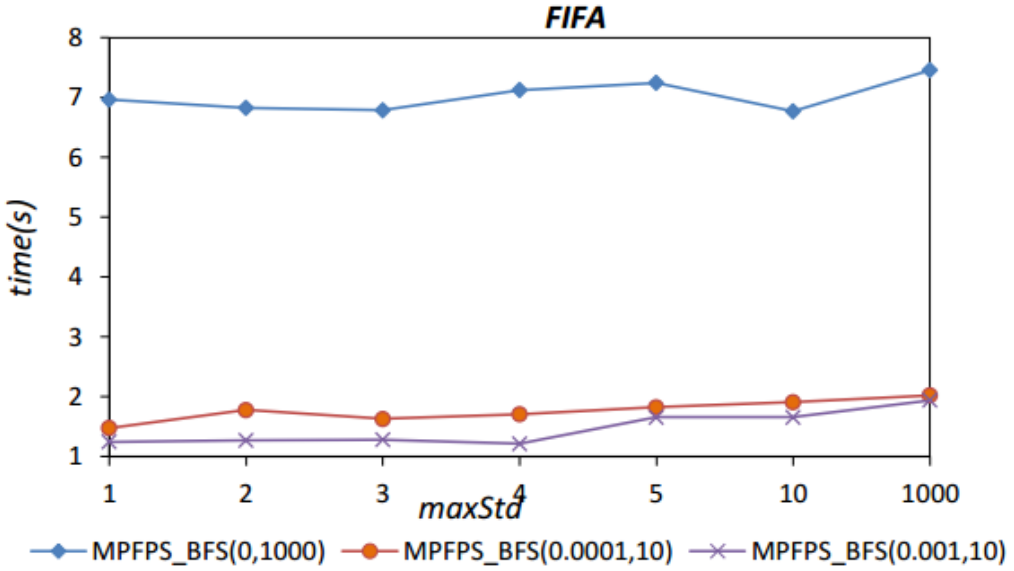
5. Experiments

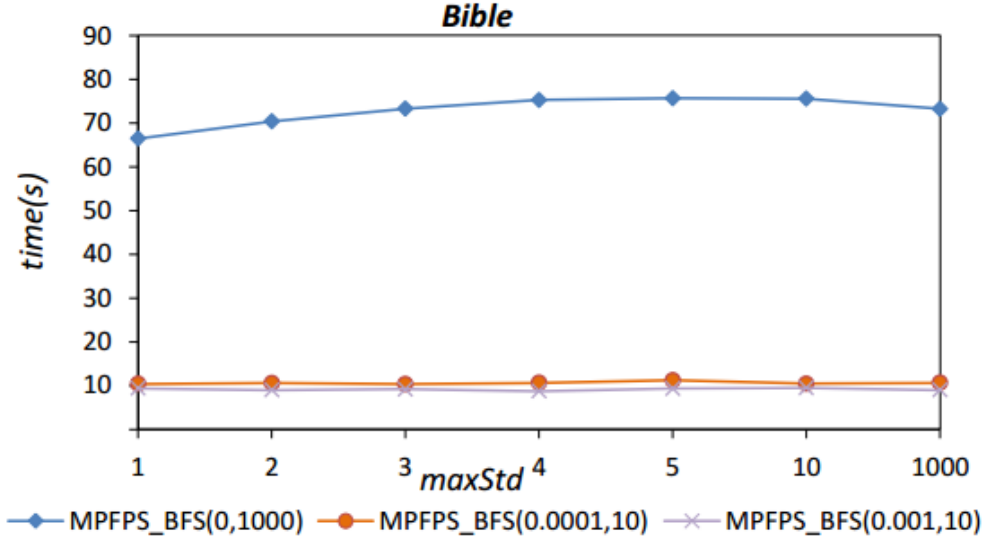
Databse	# items per transaction	# distinct items	Avg. sequence length	# sequences
FIFA	1	2,990	34.74	20,450
Bible	1	13,905	21.6	36,369
T15I1KD300K	10	1,000	15	30,000
Leviathan	1	9,025	33.8	5,834

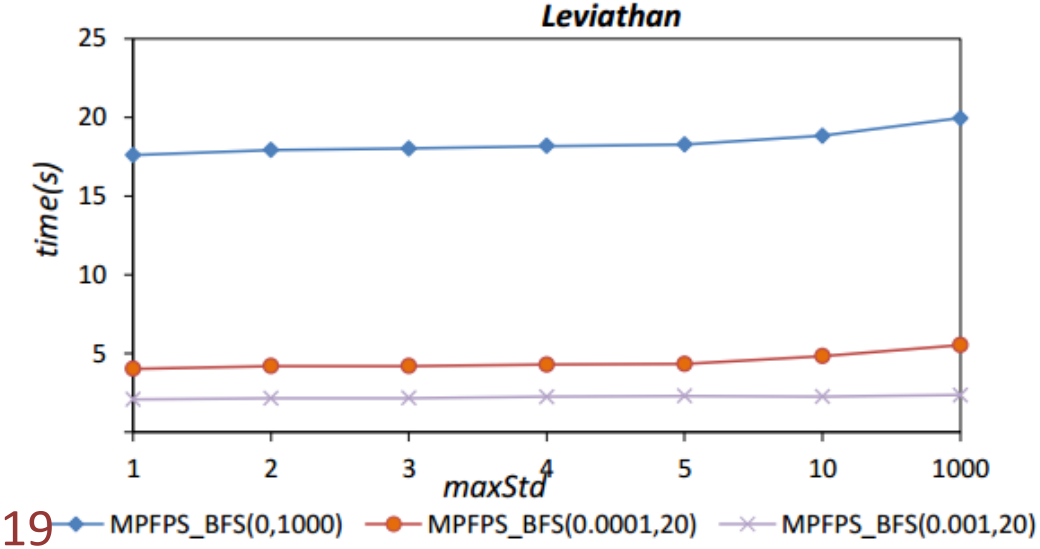
Source: http://www.philippe-fournier-viger.com/spmf/

5.1 Runtime of BFS algorithm

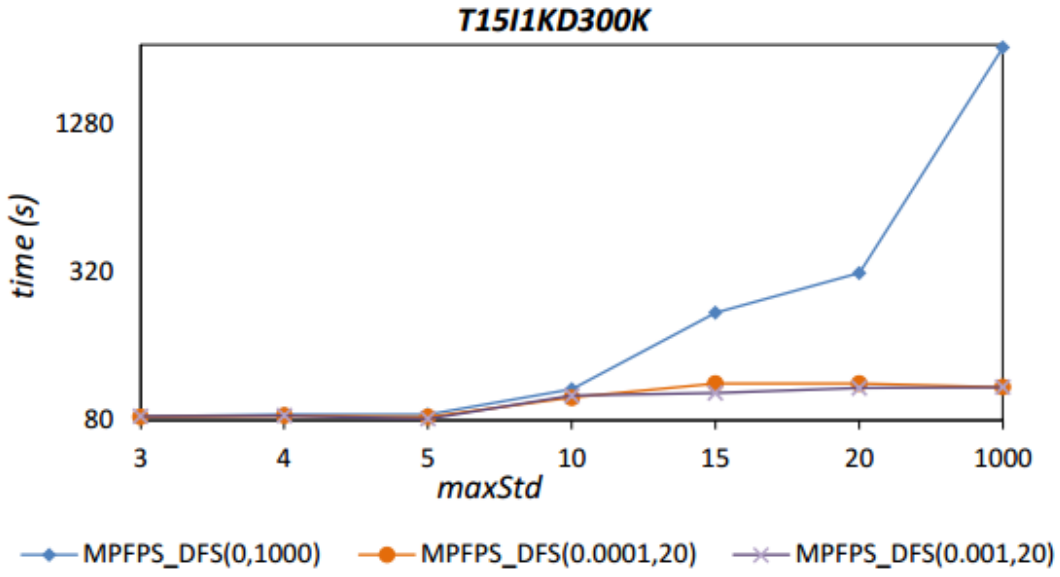


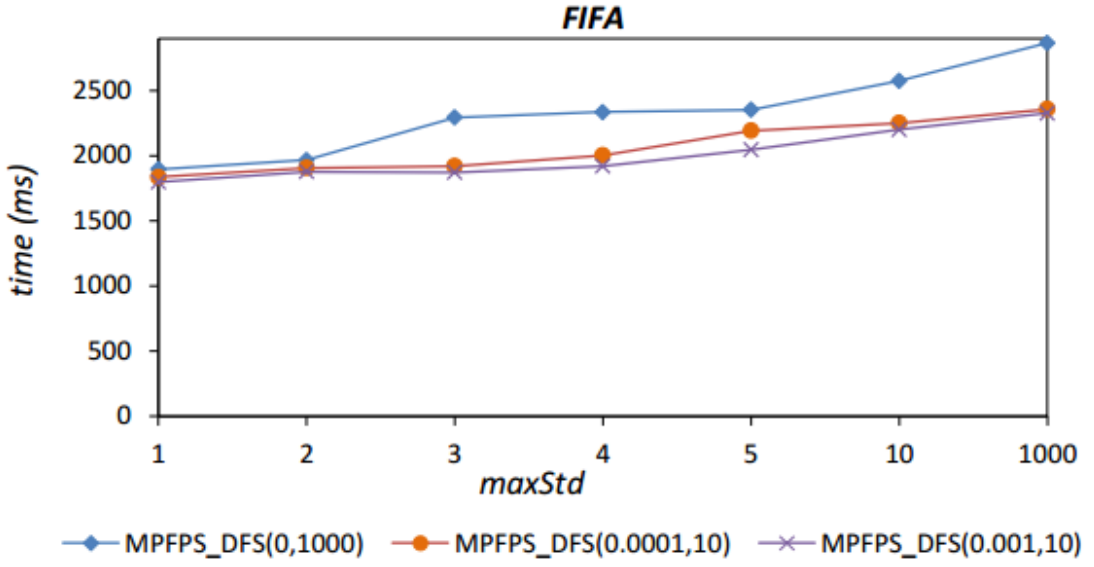


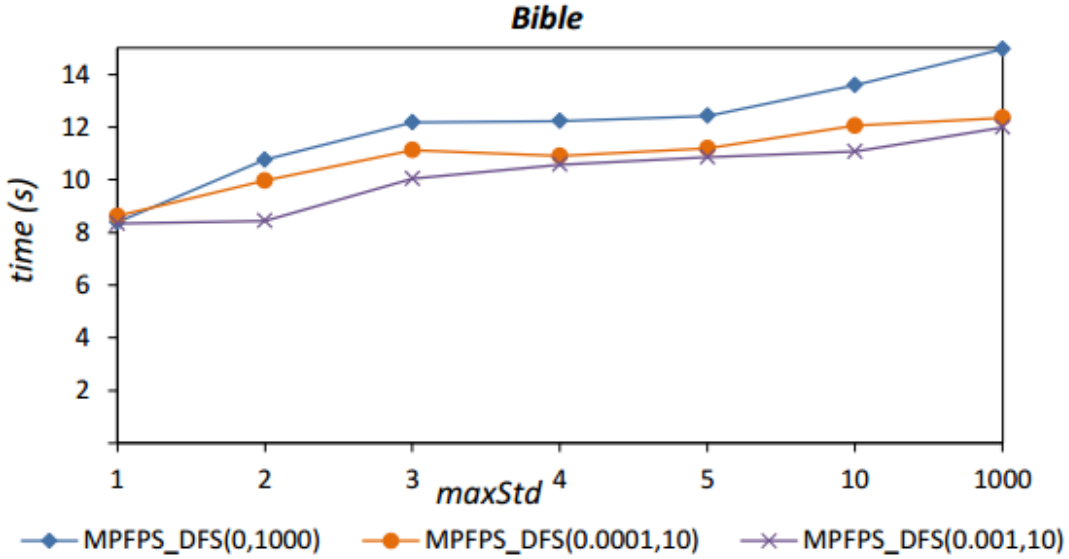


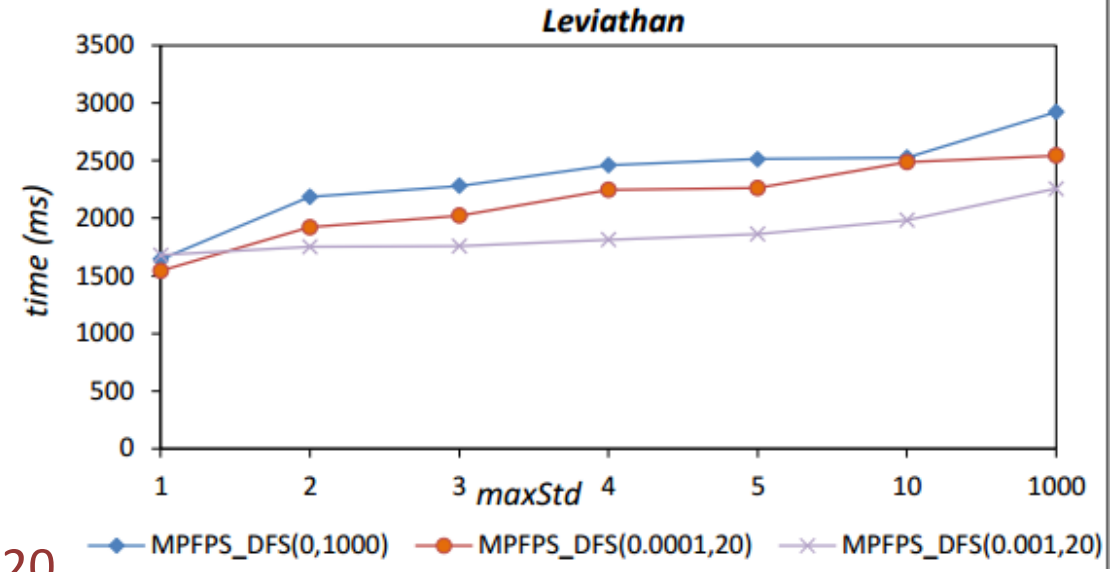


5.2 Runtime of DFS algorithm

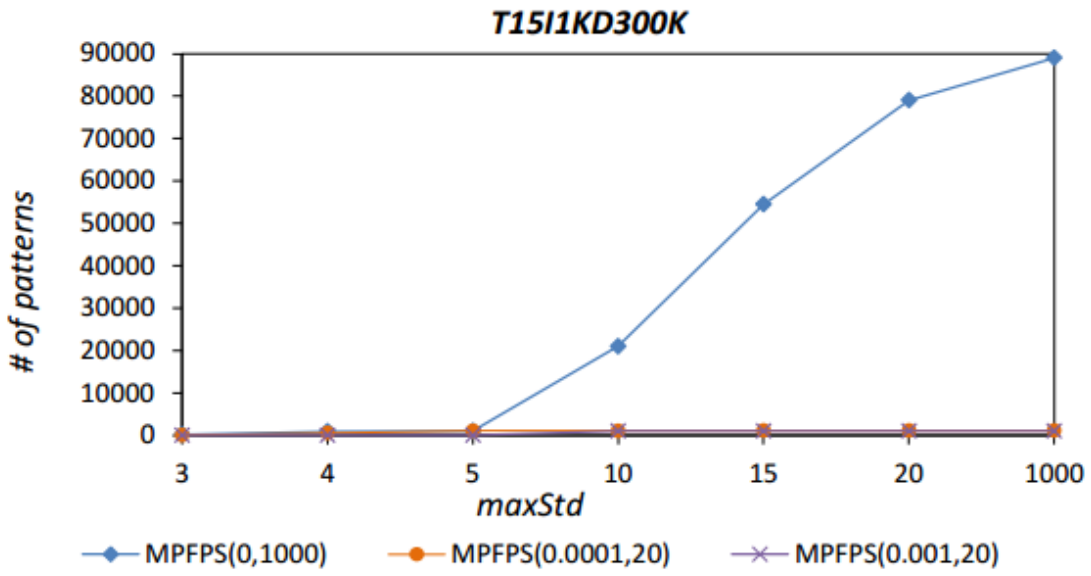


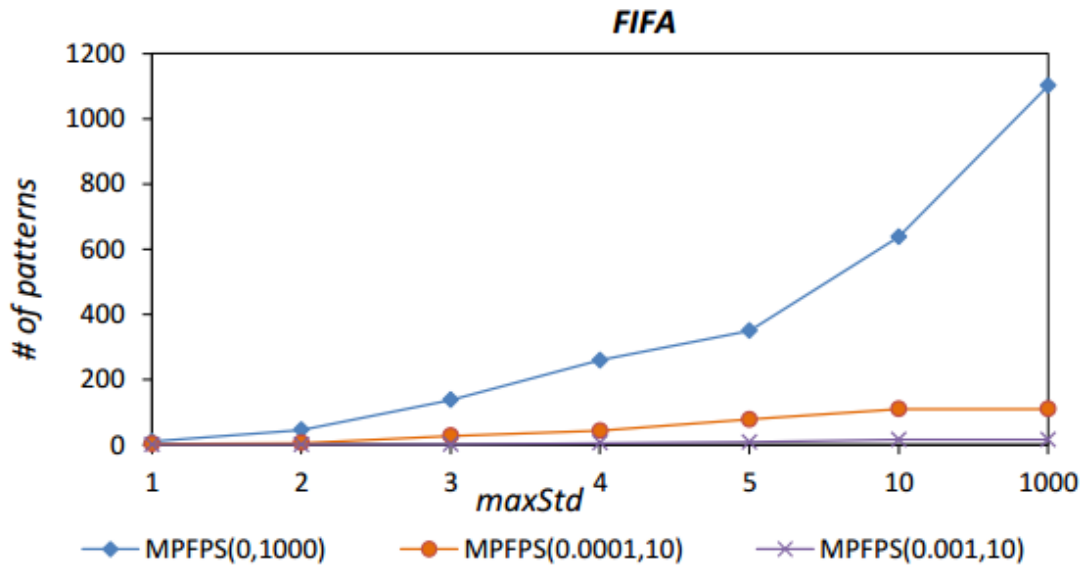


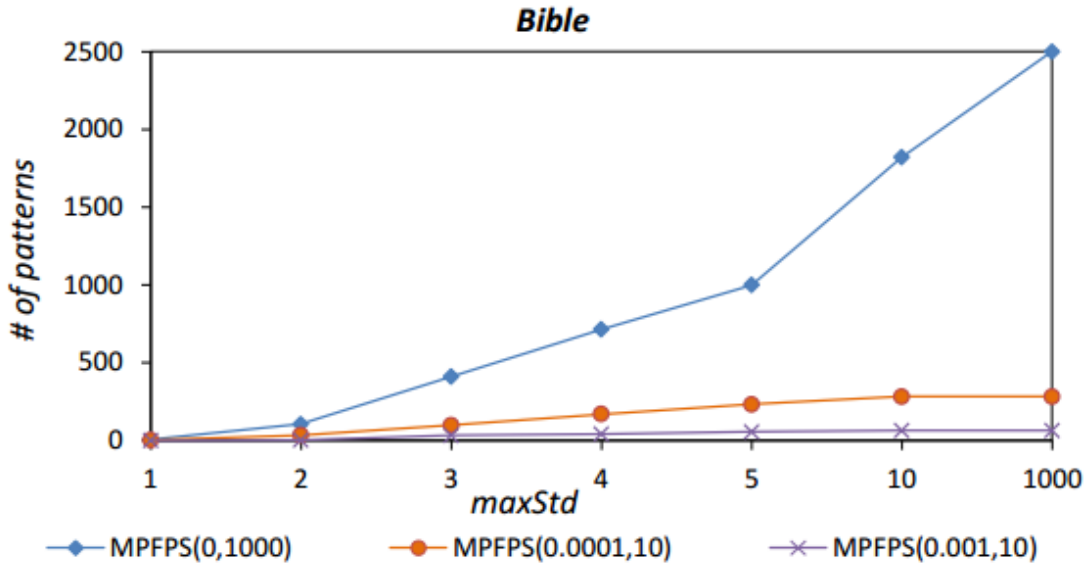


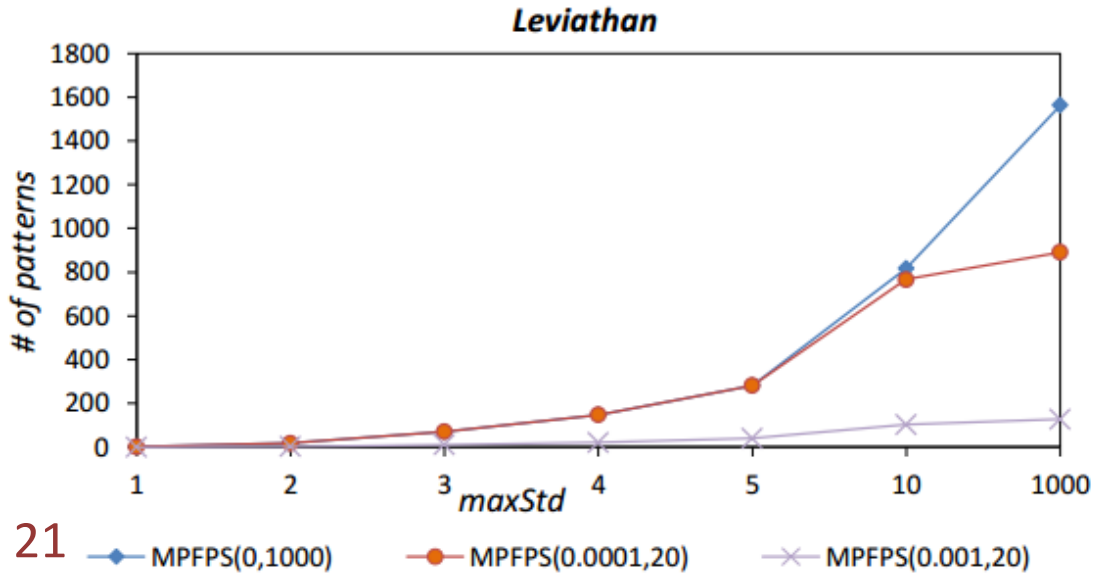


5.3 Number of patterns









5.4 Analysis of patterns found

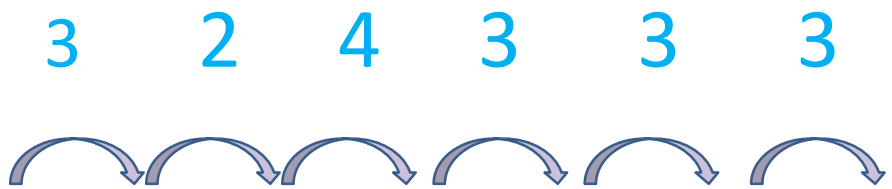
Some insightful patterns were found.
For example, in the *FIFA* database a periodic pattern is:

{webpage_255}.

It is periodic in the following sequences:
[586, 651, 780, 1344, 1908, 1974, 2361, 4165, 4926, 5544, 6070, 8420, 10295, 13231, 13634, 14595, 16231, 16609, 17641, 19925, 19996, 20266]

5.4 Analysis of patterns found

In the **780th** sequence, item **255** appears in the following transactions:



# of transactions: [0, 3, 5, 9, 12, 15, 18]

6. Conclusion

- **Novel problem**: mining periodic frequent patterns in a sequence database.
  - Two new measures: the standard deviation of periods and the *ra* measure
  - An upper-bound *boundRa* and 2 pruning properties to reduce the search space
  - The MPFPS algorithm was proposed and experimental results show it is effective.

Thanks!

Q&A...

Publications:

- Fournier-Viger, P., Li, Z., Lin, J. C.-W., Fujita, H., Kiran, U. (2018). **Discovering Periodic Patterns Common to Multiple Sequences**. Proc. 20th Intern. Conf. on Data Warehousing and Knowledge Discovery (DAWAK 2018), Springer, pp. 231-246 [EI]
- Fournier-Viger, P., Li, Z., Lin, J. C.-W., Fujita, H., Kiran, U. (2019). **MPFPS: Mining Periodic Patterns in Multiple Sequences**. Information Sciences. [SCI, Q1 journal]