Mining Periodic Patterns with a MDL Criterion

E. Galbrun  P. Cellier  N. Tatti
A. Termier  B. Crémilleux

ECML PKDD 2018 - Dublin
Introductory example

Simple periodic patterns: Cycles

Expressive periodic patterns: Tree patterns

Algorithms

Experiments

Conclusion and future work
Given an event sequence recording every day activities

\[ S = \langle (16-04-2018 \ 7:30, \ \text{wake up}) ,
(16-04-2018 \ 7:40, \ \text{prepare coffee}) ,
\ldots
(16-04-2018 \ 8:10, \ \text{take metro}) ,
\ldots
(16-04-2018 \ 11:00, \ \text{attend meeting}) ,
\ldots
(16-04-2018 \ 11:00, \ \text{eat dinner}) ,
\ldots
(17-04-2018 \ 7:32, \ \text{wake up}) ,
(17-04-2018 \ 7:38, \ \text{prepare coffee}) ,
\ldots
(20-04-2018 \ 7:28, \ \text{wake up}) ,
(20-04-2018 \ 7:41, \ \text{prepare coffee}) ,
\ldots
(15-06-2018 \ 7:28, \ \text{wake up}) ,
\ldots\rangle \]}
Introductory example

Extract activity patterns

$$S = \langle (16-04-2018 \ 7:30, \ \text{wake up}), \leftarrow \#1 \\ \ (16-04-2018 \ 7:40, \ \text{prepare coffee}), \ldots \\ \ (16-04-2018 \ 8:10, \ \text{take metro}), \ldots \\ \ (16-04-2018 \ 11:00, \ \text{attend meeting}), \ldots \\ \ (16-04-2018 \ 11:00, \ \text{eat dinner}), \ldots \\ \ (17-04-2018 \ 7:32, \ \text{wake up}), \leftarrow \#2 \\ \ (17-04-2018 \ 7:38, \ \text{prepare coffee}), \ldots \\ \ (20-04-2018 \ 7:28, \ \text{wake up}), \leftarrow \#5 \\ \ (20-04-2018 \ 7:41, \ \text{prepare coffee}), \ldots \\ \ (15-06-2018 \ 7:28, \ \text{wake up}), \ldots \rangle$$

16-04-2018 7:30, wake up
repeat every 24 hours for 5 days
Introductory example

Simple periodic patterns: Cycles

Expressive periodic patterns: Tree patterns

Algorithms

Experiments

Conclusion and future work
Event cycles

On April 16, at 7:30 AM, wake up, repeat every 24 hours for 5 days
Event cycles

On April 16, at 7:30 AM, wake up, repeat every 24 hours for 5 days

starting point $\tau$: the timestamp of the first occurrence,

event $\alpha$: the repeating event,

period $p$: the inter-occurrence distance, and

length $r$: the number of repetitions of the event.
Event cycles

**starting point** \( \tau \): the timestamp of the first occurrence,
**event** \( \alpha \): the repeating event,
**period** \( p \): the inter-occurrence distance, and
**length** \( r \): the number of repetitions of the event.

Reconstruct occurrences timestamps of repetitions recursively:

\[
\begin{align*}
  t_1 &= \tau, \\
  t_2 &= t_1 + p, \\
  \vdots \\
  t_r &= t_{r-1} + p.
\end{align*}
\]
Event cycles

Tolerate variation in inter-occurrence distances, shift corrections $E = \langle e_1, \ldots, e_{r-1} \rangle$.

Reconstruct occurrences timestamps of repetitions recursively:

\[
\begin{align*}
t_1 &= \tau, \\
t_2 &= t_1 + p + e_1, \\
&\quad \vdots \\
t_r &= t_{r-1} + p + e_{r-1}.
\end{align*}
\]
Event cycles

A cycle is specified by:

- **event** \( \alpha \): the repeating event,
- **length** \( r \): the number of repetitions of the event,
- **period** \( p \): the inter-occurrence distance,
- **starting point** \( \tau \): the timestamp of the first occurrence, and
- **shift corrections** \( E \): a list of time offsets.

Hence, a cycle is a 5-tuple \( C = (\alpha, r, p, \tau, E) \).
Problem statement (informal)

Given an event sequence, our goal is to extract a representative collection of periodic patterns called cycles.
Cycle cover

Denote as $\text{cover}(C)$ the corresponding set of reconstructed timestamp–event pairs:

$$\text{cover}(C) = \{(t_1, \alpha), (t_2, \alpha), \ldots, (t_r, \alpha)\},$$

and for a collection $\mathcal{C}$ of cycles

$$\text{cover}(\mathcal{C}) = \bigcup_{C \in \mathcal{C}} \text{cover}(C).$$

For a sequence $S$ and cycle collection $\mathcal{C}$ we call residual the timestamp–event pairs of $S$ not covered by any cycle in $\mathcal{C}$:

$$\text{residual}(\mathcal{C}, S) = S \setminus \text{cover}(\mathcal{C}).$$
Problem statement

We associate
- a cost $L(o)$ to each individual occurrence
- a cost $L(C)$ to each cycle

Then, we can reformulate our problem as follows:

**Problem**

*Given an event sequence $S$, find the collection of cycles $C$ minimizing the cost*

$$L(C, S) = \sum_{C \in C} L(C) + \sum_{o \in \text{residual}(C, S)} L(o).$$
A MDL criterion

This problem definition can be instantiated with different choices of costs.

- We propose costs motivated by the MDL principle
A MDL criterion

We propose costs motivated by the MDL principle

The MDL principle is a concept from information theory based on the insight that any structure in the data can be exploited to compress the data, and aiming to strike a balance between the complexity of the model and its ability to describe the data.
A MDL criterion

- We propose costs motivated by the **MDL principle**
- We design a scheme for encoding the input event sequence using cycles and individual occurrences
- Cost of an element = length of the code word assigned under this scheme
Introductory example

Simple periodic patterns: Cycles

Expressive periodic patterns: Tree patterns

Algorithms

Experiments

Conclusion and future work
Expressive periodic patterns: Tree patterns

More complex periodic patterns

Pattern language so far: cycles over single events
However, several events might recur regularly together and repetitions might be nested with several levels of periodicity.

More expressive pattern language: hierarchy of cyclic blocks, organised as a tree
More complex periodic patterns

Extract activity patterns

\[ S = \langle \begin{array}{c}
    (16-04-2018 \ 7:30, \ \text{wake up}), \leftarrow \#1 \\
    (16-04-2018 \ 7:40, \text{prepare coffee}), \\
    \ldots \\
    (16-04-2018 \ 8:10, \text{take metro}), \\
    \ldots \\
    (16-04-2018 \ 11:00, \text{attend meeting}), \\
    \ldots \\
    (16-04-2018 \ 11:00, \text{eat dinner}), \\
    \ldots \\
    (17-04-2018 \ 7:32, \ \text{wake up}), \leftarrow \#2 \\
    (17-04-2018 \ 7:38, \text{prepare coffee}), \\
    \ldots \\
    (20-04-2018 \ 7:28, \ \text{wake up}), \leftarrow \#5 \\
    (20-04-2018 \ 7:41, \text{prepare coffee}), \\
    \ldots \\
    (15-06-2018 \ 7:28, \ \text{wake up}), \\
    \ldots
\rangle \]

16-04-2018 7:30, wake up repeat every 24 hours for 5 days
Expressive periodic patterns: Tree patterns

More complex periodic patterns

Extract more complex activity patterns

\[ S = \langle (16-04-2018 \ 7\:30 \ , \ \text{wake up} ), \leftarrow \#1 \\
16-04-2018 \ 7\:40 \ , \ \text{prepare coffee} \rangle, \]

\[ \ldots \\
16-04-2018 \ 8\:10 \ , \ \text{take metro} \rangle, \]

\[ \ldots \\
16-04-2018 \ 11\:00 \ , \ \text{attend meeting} \rangle, \]

\[ \ldots \\
16-04-2018 \ 11\:00 \ , \ \text{eat dinner} \rangle, \]

\[ \ldots \\
17-04-2018 \ 7\:32 \ , \ \text{wake up} ), \leftarrow \#2 \\
17-04-2018 \ 7\:38 \ , \ \text{prepare coffee} \rangle, \]

\[ \ldots \\
20-04-2018 \ 7\:28 \ , \ \text{wake up} ), \leftarrow \#5 \\
20-04-2018 \ 7\:41 \ , \ \text{prepare coffee} \rangle, \]

\[ \ldots \\
(15-06-2018 \ 7\:28 \ , \ \text{wake up} ), \]

\[ \ldots \rangle \]

16-04-2018 7:30, wake up
10 min later, prepare coffee
repeat every 24 hours for 5 days
More complex periodic patterns

Extract more complex activity patterns

\[ S = \langle \begin{array}{c}
(16-04-2018\ 7:30,\ \text{wake up}) , \leftarrow \#1 - 1\text{st week} \\
(16-04-2018\ 7:40,\ \text{prepare coffee}) , \\
\vdots \\
(16-04-2018\ 8:10,\ \text{take metro}) , \\
\vdots \\
(16-04-2018\ 11:00,\ \text{attend meeting}) , \\
\vdots \\
(16-04-2018\ 11:00,\ \text{eat dinner}) , \\
\vdots \\
(17-04-2018\ 7:32,\ \text{wake up}) , \leftarrow \#2 \\
(17-04-2018\ 7:38,\ \text{prepare coffee}) , \\
\vdots \\
(20-04-2018\ 7:28,\ \text{wake up}) , \leftarrow \#5 \\
(20-04-2018\ 7:41,\ \text{prepare coffee}) , \\
\vdots \\
(15-06-2018\ 7:28,\ \text{wake up}) , \leftarrow \#5 - 9\text{th week} \\
\mapsto \end{array} \rangle \]

16-04-2018 7:30, wake up
10 min later, prepare coffee
repeat every 24 hours for 5 days
repeat every 7 days for 3 months
More complex periodic patterns

On April 16, at 7:30 AM, wake up, repeat every 24 hours for 5 days

\( \tau = 16-04-2018 \ 7:30 \)

\( r = 5 \)

\( p = 24 \text{ hours} \)
More complex periodic patterns

On April 16, at 7:30 AM, wake up, 10 minutes later, prepare coffee, repeat every 24 hours for 5 days, repeat this every 7 days for 3 months

\[ \tau = 16-04-2018 \ 7:30 \]

\[ r = 10 \quad r = 5 \]
\[ p = 7 \text{ days} \quad p = 24 \text{ hours} \]
More complex periodic patterns

We extend the encoding for this more expressive pattern language, i.e. define the cost of a pattern $P$, $L(P)$.

Then, we can extend our problem statement:

**Problem**

*Given an event sequence $S$, find the collection of patterns $\mathcal{P}$ minimising the cost*

$$L(\mathcal{P}, S) = \sum_{P \in \mathcal{P}} L(P) + \sum_{o \in \text{residual}(\mathcal{P}, S)} L(o).$$
Introductory example

Simple periodic patterns: Cycles

Expressive periodic patterns: Tree patterns

Algorithms

Experiments

Conclusion and future work
Finding periodic patterns that compress

We have a pattern language and associated encoding. How do we actually find such patterns?
Finding periodic patterns that compress

A natural way to build patterns:
start with cycles and combine them together

GrowVertically:

GrowHorizontally:

Mining Periodic Patterns, with a MDL Criterion
Algorithm outline

We propose an algorithm with three stages:

Extracting cycles: extract cycles for each event in turn, using a dynamic programming routine and a heuristic extracting triples and chaining them

Building tree patterns from cycles: perform combination rounds to generate increasingly complex patterns

Selecting the final pattern collection: solve weighted set cover problem with greedy algorithm
**Algorithm outline**

**Input:** A multi-event sequence $S$, a number $k$ of top candidates to keep  
**Output:** A collection of patterns $P$

1: $I \leftarrow \text{ExtractCycles}(S, k)$  
2: $C \leftarrow \emptyset$; $V \leftarrow I$; $H \leftarrow I$  
3: while $H \neq \emptyset$ OR $V \neq \emptyset$ do  
4: $V' \leftarrow \text{CombineVertically}(H, P, S, k)$  
5: $H' \leftarrow \text{CombineHorizontally}(V, P, S, k)$  
6: $C \leftarrow C \cup H \cup V$; $V \leftarrow V'$; $H \leftarrow H'$  
7: $P \leftarrow \text{GreedyCover}(C, S)$  
8: return $P$
Introductory example

Simple periodic patterns: Cycles

Expressive periodic patterns: Tree patterns

Algorithms

Experiments

Conclusion and future work
Datasets

We run experiments on real-world event log datasets.
Life-tracking (a.k.a. quantified self): sacha
Execution traces: 3zap and bugzilla
Example patterns

a) $\tau = 2016-03-16\ 11:45$
   $r = 291$
   $p = 2\ h\ 30$
   [childcare 1 h30 min childcare]

b) $\tau = 2014-12-18\ 00:15$
   $r = 76$
   $p = 1\ d$
   [sleep 8 h30 min sleep]

c) $\tau = 2015-12-16\ 00:00$
   $r = 48$
   $p = 1\ d\ 15\ min$
   [sleep]

d) $\tau = 2017-01-09\ 18:15$
   $r = 7$
   $p = 10\ d$
   [dinner 0 min]
   [clean ktch. 30 min clean ktch.]

e) $\tau = 2015-01-08\ 08:45$
   $r = 14$
   $p = 7\ d$
   [subway 45 min subway]

f) $\tau = 151772$
   $r = 4$
   $p = 221$
   6:C 2
   2395:X $r_1 = 4, p_1 = 2$

Figure: Example patterns from sacha (a–e) and 3zap (f).
Compression ratios

\[ \%L = 100 \cdot \frac{L(C, S)}{L(\emptyset, S)} \]

Figure: Compression ratios for three sequences.
Introductory example

Simple periodic patterns: Cycles

Expressive periodic patterns: Tree patterns

Algorithms

Experiments

Conclusion and future work
Conclusion

We propose
- an approach to mine periodic patterns with a MDL criterion
- an algorithm to put it into practise

In our experiments, we show that we are able to
- extract sets of patterns that compress the input sequences
- identify meaningful patterns
Directions for future work

- Taking into account background knowledge
- Making the algorithm more robust to noise and more scalable
- Designing tailored visualizations
Simple periodic patterns: Cycles

Experiments
Cycle cover

Denote as $cover(C)$ the corresponding set of reconstructed timestamp–event pairs:

$$cover(C) = \{(t_1, \alpha), (t_2, \alpha), \ldots, (t_r, \alpha)\}.$$

A cycle $C$ covers an occurrence if the corresponding timestamp–event pair belongs to $cover(C)$. 
Cycle cover

- Time is represented in an absolute manner
- An event can occur only once at any given timestamp

Hence we do not need to worry about overlapping cycles nor about an order between cycles
Cycle cover

Denote as $\text{cover}(C)$ the set of timestamp–event pairs for a collection $C$ of cycles $C = \{C_1, \ldots, C_m\}$:

$$\text{cover}(C) = \bigcup_{C \in C} \text{cover}(C).$$

For a sequence $S$ and cycle collection $C$ we call residual the timestamp–event pairs of $S$ not covered by any cycle in $C$:

$$\text{residual}(C, S) = S \setminus \text{cover}(C).$$
Simple periodic patterns: Cycles

Code lengths as costs

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

Look more closely at

- the range in which each of these pieces of information takes value,
- what values—if any—should be favoured, and
- how the values of the different pieces depend on one another.
Code lengths as costs

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

**Cycle event**

Events that occur more frequently receive shorter code words:

\[ L(\alpha) = - \log(fr(\alpha)) = - \log\left(\frac{|S^{(\alpha)}|}{|S|}\right) \]
Code lengths as costs

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

**Cycle length**

The length of a cycle cannot be greater than the number of occurrences of the event:

\[ L(r) = \log(|S^{(\alpha)}|) \]
Code lengths as costs

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

Cycle shift corrections

- **value digits**: Each correction \( e \) is represented by \( |e| \) ones,
- **sign digits**: prefixed by a single bit indicating the shift direction,
- **separating digits**: separated from the next correction by a zero

Example: \( \langle 3, -2, 0, 4 \rangle \rightarrow 01110111000011110 \)

\[ L(E) = 2|E| + \sum_{e \in E} |e| \]
Code lengths as costs

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

Cycle period

The cycle can span at most the time of the whole sequence:

\[ L(p) = \log \left( \left\lceil \frac{\Delta(S) - \sigma(E)}{r - 1} \right\rceil \right) \]
Code lengths as costs

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

Cycle starting point

The cycle can start anytime between \( t_{start}(S) \) and \( t_{end}(S) - \Delta(C) \):

\[ L(\tau) = \log(\Delta(S) - \sigma(E) - (r - 1)p + 1) \]
Simple periodic patterns: Cycles

**Code lengths as costs**

Putting everything together, the cost of a cycle is

\[
L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \\
= \log(|S|) + \log \left( \left\lceil \frac{\Delta(S) - \sigma(E)}{r - 1} \right\rceil \right) \\
+ \log(\Delta(S) - \sigma(E) - (r - 1)p + 1) \\
+ 2|E| + \sum_{e \in E} |e|
\]
Putting everything together, the cost of a cycle is

\[ L(C) = L(\alpha) + L(r) + L(p) + L(\tau) + L(E) \]

\[ = \log(|S|) + \log \left( \left\lfloor \frac{\Delta(S) - \sigma(E)}{r - 1} \right\rfloor \right) \]

\[ + \log(\Delta(S) - \sigma(E) - (r - 1)p + 1) \]

\[ + 2|E| + \sum_{e \in E} |e| \]

On the other hand, the cost of an individual occurrence is

\[ L(o) = L(t) + L(\alpha) = \log(\Delta(S) + 1) - \log \left( \frac{|S(\alpha)|}{|S|} \right) \]
Problem statement

Problem

*Given an event sequence* $S$, *find the collection of cycles* $C$ *minimising the cost*

$$L(C, S) = \sum_{C \in C} L(C) + \sum_{o \in \text{residual}(C, S)} L(o) .$$
Choosing the best period

Given an ordered list of occurrences \( \langle t_1, t_2, \ldots, t_l \rangle \) of event \( \alpha \)

Goal determine the best cycle to cover these occurrences
Choosing the best period

Given an ordered list of occurrences \( \langle t_1, t_2, \ldots t_l \rangle \) of event \( \alpha \)

Goal determine the best cycle to cover these occurrences

- clearly, \( \alpha, r, \) and \( \tau \) are determined
- need to find \( p \) such that \( L(C) \) is minimised
Choosing the best period

Given an ordered list of occurrences \( \langle t_1, t_2, \ldots, t_l \rangle \) of event \( \alpha \)

Goal determine the best cycle to cover these occurrences
- clearly, \( \alpha, r, \) and \( \tau \) are determined
- need to find \( p \) such that \( L(C) \) is minimised

Find \( p \) that minimises \( L(E) \)
\( \rightarrow \) let \( p \) equal the median of the inter-occurrence distances
Simple periodic patterns: Cycles

Experiments
Compression ratios

Figure: Compression ratios for *sacha* sequences with various time granularities.
Running times

Figure: Running times for sequences from the different datasets, in hours (left) and zoomed-in in minutes (middle) and seconds (right).