Episode Mining

Episode is a set of events occurring in a sequence
- that occur often enough
- that occur in their vicinity
- that may have some restricted order
Episode is specified by a DAG. A sequence covers an episode if
- a node is mapped to a unique episode
- parents occur before children

Support \( \rightarrow \) number of fixed-size windows covering the episode

\[ a, b, c, d, e \]

5 windows of length 5 cover the example

Simultaneous Events

Extend episodes to handle simultaneous events. 4 different type of relationships between two events.
- order between \( a \) and \( b \) doesn’t matter
- \( a \) and \( b \) must occur at the same time
- \( b \) must follow \( a \) or occur at the same time
- \( b \) must follow \( a \) properly
Introduce two types of edges:
- weak \( \leftrightarrow \) event will follow or occur at the same time
- proper \( \leftrightarrow \) event will follow properly
- \( a \) and \( b \) must occur together
- \( b \) follows or occurs at the same time as \( a \)
- \( c \) follows or occurs at the same time as \( c \)
- \( c \) must follow \( a \) (and \( b \)) properly

Goal & Approach

Find all episodes that have enough support

Key step \( \leftrightarrow \) Support is higher for subepisodes
Discover episodes in depth-first style. Three different levels for traversal
1. add new nodes
2. add weak edges
3. turn weak edges into proper edges
Prune branches with infrequent episodes.
Apply closure and prune branch if it has been already processed.

Problems

- Pattern Explosion:
  If \( a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_N \) is frequent, then at least

\[
\begin{array}{cccccc}
N & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 4 & 16 & 64 & 256 & 1024 \\
7 & 8 & 9 & & & \\
139,387 & 3,730,216 & 145,605,024 & & & \\
\end{array}
\]

episodes are frequent

- Redundancy issues:
  Episodes are same
  \[
  \begin{array}{ccc}
  a & b & c \\
  a & b & c \\
  b & b & b \\
  \end{array}
  \]

Coverage test \( \leftrightarrow \) NP-complete

Closed Episodes

Technique for reducing number of patterns
Closed pattern \( \leftrightarrow \) no superpattern with the same support

\[
\begin{array}{cccc}
1 & b & c & a \\
1 & a & d & c \\
1 & c & b & d \\
2 & a & d & c \\
2 & a & b & c \\
\end{array}
\]

G\(_1\), \ldots, G\(_4\) have support 2. G\(_4\) and G\(_6\) are closed
Problem \( \leftrightarrow \) closure of an episode is not unique: G\(_4\) and G\(_6\) are both closures for G\(_1\)

Use number of instances instead of support

\[
\begin{array}{cccc}
1 & a & b & c \\
1 & a & b & c \\
1 & a & b & c \\
1 & a & b & c \\
1 & a & b & c \\
\end{array}
\]

G\(_1\) and G\(_2\) have 4 mappings, G\(3, \ldots, G_6\) have 2 mappings \( \rightarrow \) G\(_2\), G\(_4\), G\(_6\) are closed and are closures for G\(_1\), G\(_3\), G\(_5\)

Theorem: mapping-closed episode is also support-closed

Mine mapping-closed episodes, keep only support-closed in postprocessing

Subset relationship

Need a proper subset relationship

- for pruning non-closed episodes
- for removing similar episodes

Definition:
episode G is a subepisode of H if \( s \) covers H \( \rightarrow \) s covers G.

Theorem: testing subset relation \( \leftrightarrow \) NP-hard.

Not a problem in practice
- episodes are typically small
- most of them are simple cases
Simple case:
if all events have unique labels, relationship can be checked by checking the edges.

General case:
check by
- generating all sequences that cover H
- if they cover G, then G \( \subset \) H
- generate in a clever way
- remove sinks from H and try
- to find corresponding sinks from G
- continue recursively

Experiments

- alarms dataset
- alarms generated in a factory
- 514,502 events of 9,595 different types
- 18 months of data

<table>
<thead>
<tr>
<th>w (s)</th>
<th>( \sigma \times 10^3 )</th>
<th>s-cl.</th>
<th>m-cl.</th>
<th>freq. (est)</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>500</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>180</td>
<td>400</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>180</td>
<td>300</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>180</td>
<td>210</td>
<td>26</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>600</td>
<td>2000</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>600</td>
<td>1000</td>
<td>24</td>
<td>27</td>
<td>39</td>
</tr>
<tr>
<td>600</td>
<td>500</td>
<td>90</td>
<td>137</td>
<td>493</td>
</tr>
<tr>
<td>600</td>
<td>280</td>
<td>422</td>
<td>698</td>
<td>2,821</td>
</tr>
<tr>
<td>900</td>
<td>2000</td>
<td>24</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>900</td>
<td>1000</td>
<td>52</td>
<td>58</td>
<td>94</td>
</tr>
<tr>
<td>900</td>
<td>500</td>
<td>280</td>
<td>426</td>
<td>1,997</td>
</tr>
<tr>
<td>900</td>
<td>350</td>
<td>1,845</td>
<td>9,484</td>
<td>190,990</td>
</tr>
</tbody>
</table>

- trains dataset
- delays at a railway station in Belgium
- 10,115 events, 1,280 different train IDs
- one month of data
- window size \( \leftrightarrow \) 30 minutes

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>s-cl.</th>
<th>m-cl.</th>
<th>freq. (est)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30,000</td>
<td>141</td>
<td>141</td>
<td>141</td>
</tr>
<tr>
<td>20,000</td>
<td>1,994</td>
<td>1,995</td>
<td>2,903</td>
</tr>
<tr>
<td>17,000</td>
<td>8,352</td>
<td>8,416</td>
<td>22,542</td>
</tr>
<tr>
<td>15,000</td>
<td>26,170</td>
<td>26,838</td>
<td>172,967</td>
</tr>
<tr>
<td>13,000</td>
<td>94,789</td>
<td>101,882</td>
<td>3,552,104</td>
</tr>
<tr>
<td>12,000</td>
<td>189,280</td>
<td>211,636</td>
<td>33,660,094</td>
</tr>
</tbody>
</table>