Mining Episodes and Episode Rules

Juho Rinta-Paavola

10 October 2023

Who am I?

Juho Rinta-Paavola

- Studied CS at Aalto in 2015–2022
- Did my Master's thesis on episode mining
- Now working as a software engineer for the company I did my thesis for

Overview

- Definitions: what's an episode?
- Overview of approaches to episode mining
- The WINEPI algorithm
- Redundant episode rules
- Practical application of episode mining

Definitions

- **Episode** A is a partially ordered collection of **event types** $L(e) \in \Sigma$ that occur close together in an **event sequence** s
- **Episode rule** $B \rightarrow A$ describes the relationship between an episode A and its **subepisode** $B \preceq A$

Mannila et al., "Discovery of frequent episodes in event sequences".

Definitions

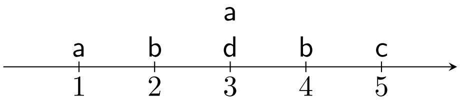
- **Episode** A is a partially ordered collection of **event types** $L(e) \in \Sigma$ that occur close together in an **event sequence** $\mathbf{s}$
- **Episode rule** $B \rightarrow A$ describes the relationship between an episode A and its **subepisode** $B \preceq A$

Note: much of what you have learned about itemsets and association rules applies with little modification to episodes and episode rules!

Mannila et al., "Discovery of frequent episodes in event sequences".

Definitions

- Data: exactly one long-ish event sequence



- Event e consists of an event type $L(e)$ and a timestamp $t(e)^*$
- Multiple event sequences can be concatenated, but if you have a lot of short sequences, other kinds of pattern might be more appropriate

---

*However, the sequence is not required to be a time sequence

Definitions

- What does it mean for events in an episode to occur close together?
- **Window**-constrained: limit maximum duration between first and last event of the occurrence.
- Does the episode occur in the window of size `win` starting at $t_{\text{win\_start}}$, i.e. in the interval between $t_{\text{win\_start}}$ and $t_{\text{win\_start}} + \texttt{win}$?[*]
- Problem: a fixed window size can be simultaneously too long for simple episodes and too short for complex episodes.
- **Gap**-constrained: limit maximum duration between consecutive events in an occurrence.[†]

[*] Mannila et al., "Discovering Frequent Episodes in Sequences".

[†] Casas-Garriga, "Discovering Unbounded Episodes in Sequential Data"; Méger and Rigotti, "Constraint-Based Mining of Episode Rules and Optimal Window Sizes".

Definitions

- In general, an episode may have multiple copies of an event type, and may require them to be in any partial order
  Example: first a occurs, then b and c occur in either order, and finally a occurs again
- If the episode is totally ordered, then it is a **serial** episode
  Example: the event types (a, c, b, a) occur in that order
- If the episode is unordered, then it is a **parallel** episode
  Example: a occurs twice, b and c occur once each, order doesn't matter
- If the episode has at most one of each event type, then it is an **injective** episode.

Remark: serial and parallel episodes can be thought of as sequences and multisets of event types, respectively

Approaches: frequent episode mining

- Various definitions of episode frequency, e.g.
  - Number of fixed-size windows in $\mathbf{s}$ in which episode occurs*
  - Number of non-overlapping **minimal occurrences**†
    A window of $\mathbf{s}$ is a minimal occurrence of A iff A occurs in the window, but not in any smaller window
- Beware! Some definitions of frequency in the literature have later turned out not to be monotone

*Mannila et al., "Discovering Frequent Episodes in Sequences".
†Tatti, "Significance of Episodes Based on Minimal Windows".

Approaches: statistically significant episodes

- Various null hypotheses are possible
  - If events are independent and identically distributed, then episode A isn't statistically significant $(p > 0.5)$ if its non-overlapping occurrence based frequency is $\leq |\mathbf{s}| \Big/ |A||\Sigma|$[*]
  - If events are independent and distributed according to event type frequencies, or according to a Markov model, then distribution of window-based episode frequencies is asymptotically normal[†]
  - If events are independent and distributed according to event type frequencies, then distribution of the sum of lengths of minimal non-overlapping occurrences is asymptotically normal[‡]

---

[*] Laxman et al., "Discovering frequent episodes and learning hidden Markov models: a formal connection".

[†] Atallah et al., "Detection of Significant Sets of Episodes in Event Sequences"; Gwadera et al., "Reliable detection of episodes in event sequences", "Markov models for identification of significant episodes".

[‡] Tatti, "Significance of Episodes Based on Minimal Windows".

Approaches: kind of episodes

- General episodes have severe pattern explosion: a 9-event serial episode has over 140 million general subepisodes* but only 510 serial subepisodes
- Deciding if a general episode occurs in a sequence is NP-complete, although cases seen in practice are easy to solve†
- Many algorithms mine only parallel or serial episodes, which are easier to handle than general episodes
- However, there are intermediates between the extremes‡

---

*Tatti and Cule, "Mining closed strict episodes".

†Tatti and Cule, "Mining closed episodes with simultaneous events".

‡E.g. the above and Fournier-Viger et al., "Mining Partially-Ordered Episode Rules in an Event Sequence"

The WINEPI algorithm

- WINEPI* is the 'original' episode mining algorithm
- Mines frequent episodes and their rules according to the window-based definition of frequency
- Variations for parallel and serial, injective and non-injective episodes

* Mannila et al., "Discovering Frequent Episodes in Sequences", "Discovery of frequent episodes in event sequences".

The WINEPI algorithm

Goal: find the set of frequent episodes

$$\mathcal{F} = \{A \mid \mathbb{P}(A) \geq \mathbb{P}_{\mathrm{thresh}}\},$$

and the set of episode rules

$$\mathcal{R} = \{B \to A \mid A \in \mathcal{F}, B \preceq A, \varphi(B \to A) \geq \varphi_{\mathrm{thresh}}\},$$

given the kind of episodes to mine, the event sequence `s`, the window size `win`, the frequency threshold $\mathbb{P}_{\mathrm{thresh}}$, and the confidence threshold $\varphi_{\mathrm{thresh}}$.

The WINEPI algorithm

$\mathcal{C}_1 \leftarrow \{\{L(e)\} \mid e \in \mathbf{s}\}$ ▷ Base case of candidate generation
$l \leftarrow 1$
while $\mathcal{C}_l \neq \emptyset$ **do**
  $\mathcal{F}_l \leftarrow \text{Recognise}(\mathcal{C}_l, \mathbf{s}, \mathtt{win}, \mathbb{P}_{\text{thresh}})$
  $\mathcal{C}_{l+1} \leftarrow \text{GenerateCandidates}(\mathcal{F}_l)$
  $l \mathrel{+}= 1$
Output $\mathcal{F} = \bigcup_{l=1}^{\infty} \mathcal{F}_l$
Output $\mathcal{R} = \text{GenerateRules}(\mathcal{F}, \varphi_{\text{thresh}})$

Does this algorithm look familiar?

The WINEPI algorithm

$\mathcal{C}_1 \leftarrow \{\{L(e)\} \mid e \in \mathbf{s}\}$ ▷ Base case of candidate generation
$l \leftarrow 1$
while $\mathcal{C}_l \neq \emptyset$ **do**
  $\mathcal{F}_l \leftarrow \text{Recognise}(\mathcal{C}_l, \mathbf{s}, \mathtt{win}, \mathbb{P}_{\text{thresh}})$
  $\mathcal{C}_{l+1} \leftarrow \text{GenerateCandidates}(\mathcal{F}_l)$
  $l \mathrel{+}= 1$
Output $\mathcal{F} = \bigcup_{l=1}^{\infty} \mathcal{F}_l$
Output $\mathcal{R} = \text{GenerateRules}(\mathcal{F}, \varphi_{\text{thresh}})$

Does this algorithm look familiar? It's APRIORI!

WINEPI: recognition pass

Goal: given $\mathcal{C}_l$, the set of complexity l candidate episodes, output $\mathcal{F}_l$, the set of complexity l frequent episodes and their frequencies:

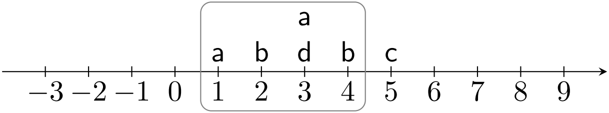
$$\{(A, \mathbb{P}(A)) \in \mathcal{C}_l \times [\mathbb{P}_{\text{thresh}}, 1]\}$$

Sliding window approach: as the window slides by one unit, we

- process events that just entered the window: keep track of which episodes are now inside the window
- process events that just exited the window: keep track of which episodes are no longer inside the window and update their frequencies

WINEPI: recognition pass

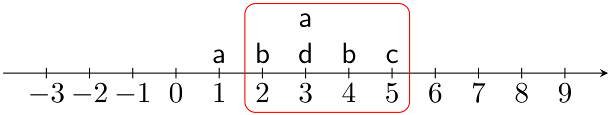
Consider the parallel episode $A = \{a, b, c\}$
A does not occur: event type c is missing



WINEPI: recognition pass

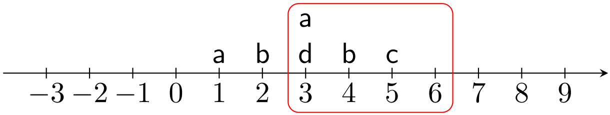
Consider the parallel episode $A = \{a, b, c\}$

A occurs: event $(c, 5)$ entered



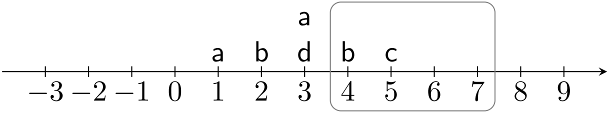
WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$
A occurs: event $(b, 2)$ exited but $(b, 4)$ remains



WINEPI: recognition pass

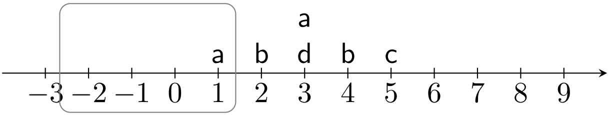
Consider the parallel episode $A = \{a, b, c\}$
A no longer occurs: event $(a, 3)$ exited



WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$

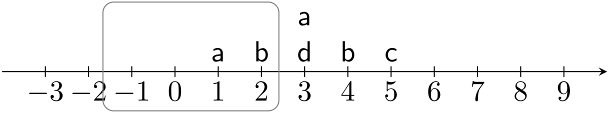
Two distinct windows contain A



There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$
Two distinct windows contain A

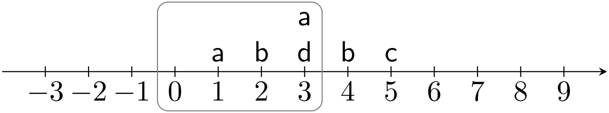


There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$

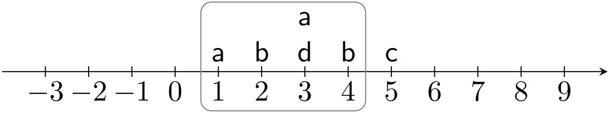
Two distinct windows contain A



There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$
Two distinct windows contain A

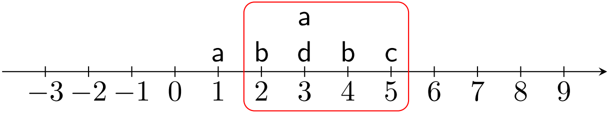


There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$

Two distinct windows contain A

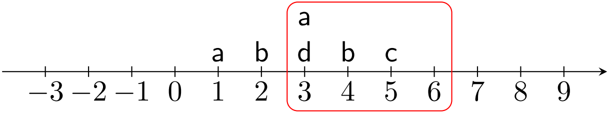


There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$

Two distinct windows contain A

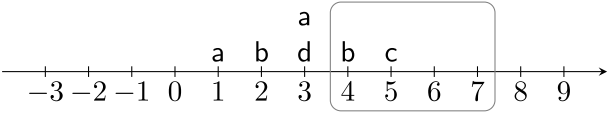


There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$

Two distinct windows contain A

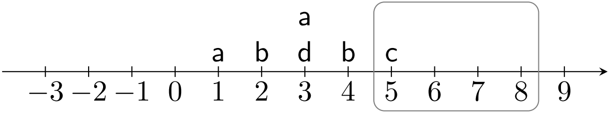


There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

Consider the parallel episode $A = \{a, b, c\}$

Two distinct windows contain A



There are eight windows in total, so $\mathbb{P}(A) = \frac{2}{8}$

WINEPI: recognition pass

How we keep track of episodes depends on if we are mining parallel or serial ones

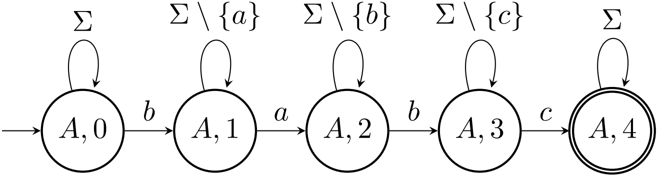
Parallel: for each parallel episode, count the number of events that are inside the window and have event types contained in the episode

- Once all event types' counts reach the required multiplicity, the episode occurs
- Once any event type's count falls below the required multiplicity, the episode no longer occurs

WINEPI: recognition pass

How we keep track of episodes depends on if we are mining parallel or serial ones

Serial: we identify each serial episode with a non-deterministic finite state machine



FSM for episode $A = (b, a, b, c)$

- When an instance of the FSM is in the final state, the episode occurs
- An FSM instance is deinitialised `win` time steps after moving out of the initial state

WINEPI: candidate generation

Goal: given $\mathcal{F}_l$, the set of complexity l frequent episodes, output $\mathcal{C}_{l+1}$, the set of candidate episodes of complexity $l+1$

for all $\mathcal{B} \in$ Blocks($\mathcal{F}_l$) **do** ▷ Episodes sharing first $l-1$ events
  for all $(I, J) \in$ Pairs($\mathcal{B}$) **do**
    $A \leftarrow I \oplus$ Last(J) ▷ Combine two episodes
    if DirectSubepisodes(A) $\subseteq \mathcal{F}_l$ **then** ▷ Monotonicity
      $\mathcal{C}_{l+1} \leftarrow \mathcal{C}_{l+1} \cup \{A\}$
return $\mathcal{C}_{l+1}$

- If mining parallel episodes, give event types a fixed order

Pairs($\mathcal{B}$)	Parallel	Serial
Non-injective	2-Combinations $\mathcal{B}$ w/ replacement	$\mathcal{B} \times \mathcal{B}$
Injective	2-Combinations $\mathcal{B}$ w/out replacement	2-Permutations

WINEPI: candidate generation

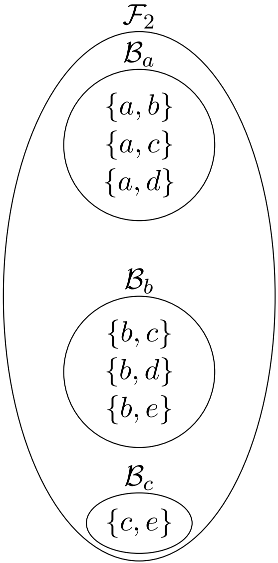
$\mathcal{F}_2$

$\{a, b\}$
$\{a, c\}$
$\{a, d\}$

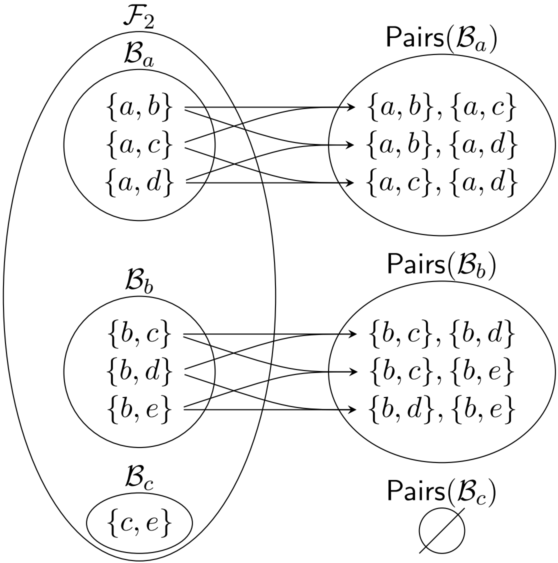
$\{b, c\}$
$\{b, d\}$
$\{b, e\}$

$\{c, e\}$

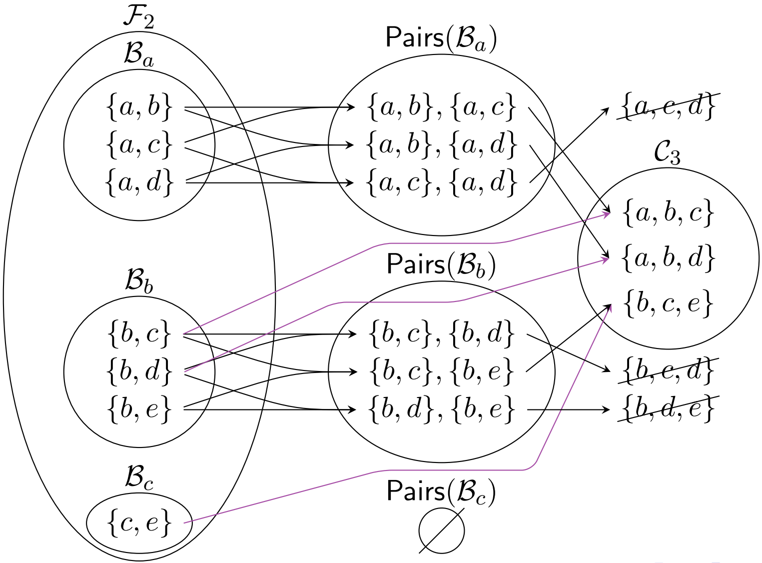
WINEPI: candidate generation



WINEPI: candidate generation



WINEPI: candidate generation



WINEPI: rule generation

Goal: given $\mathcal{F}$, the set of frequent episodes, and a confidence treshold φ_{thresh}, output $\mathcal{R}$, the rules that hold in $\mathcal{F}$ w.r.t. φ_{thresh}

for all $A \in \mathcal{F}$ such that $|A| > 1$ **do**
  for all $B \in$ DirectSubepisodes(A) **do**
    if $\varphi = \frac{\mathbb{P}(A)}{\mathbb{P}(B)} \geq \varphi_{\text{thresh}}$ **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{$Rule$(B \to A,$ confidence $\varphi)\}$

return $\mathcal{R}$

WINEPI: rule generation

Episode A	$\mathbb{P}(A)$
$\{a\}$	0.624
$\{b\}$	0.031
$\{c\}$	0.167
$\{a,b\}$	0.029
$\{a,c\}$	0.100
$\{b,c\}$	0.007
$\{a,b,c\}$	0.004

Rule $B \to A$	$\varphi(B \to A)$
$\{a\} \to \{a,b\}$	$0.029/0.624 = 0.046$
$\{b\} \to \{a,b\}$	$0.029/0.031 = 0.935$
$\{a\} \to \{a,c\}$	$0.100/0.624 = 0.160$
$\{c\} \to \{a,c\}$	$0.100/0.167 = 0.599$
$\{b\} \to \{b,c\}$	$0.007/0.031 = 0.226$
$\{c\} \to \{b,c\}$	$0.007/0.167 = 0.042$
$\{a,b\} \to \{a,b,c\}$	$0.004/0.029 = 0.138$
$\{a,c\} \to \{a,b,c\}$	$0.004/0.100 = 0.040$
$\{b,c\} \to \{a,b,c\}$	$0.004/0.007 = 0.571$

The problem with confidence

Episode A	$\mathbb{P}(A)$
$\{a\}$	0.624
$\{b\}$	0.031
$\{c\}$	0.167
$\{a, b\}$	0.029
$\{a, c\}$	0.100
$\{b, c\}$	0.007
$\{a, b, c\}$	0.004

Rule $B \to A$	$\varphi(B \to A)$
$\{a\} \to \{a, b\}$	$0.029/0.624 = 0.046$
$\{b\} \to \{a, b\}$	$0.029/0.031 = 0.935$
$\{a\} \to \{a, c\}$	$0.100/0.624 = 0.160$
$\{c\} \to \{a, c\}$	$0.100/0.167 = 0.599$
$\{b\} \to \{b, c\}$	$0.007/0.031 = 0.226$
$\{c\} \to \{b, c\}$	$0.007/0.167 = 0.042$
$\{a, b\} \to \{a, b, c\}$	$0.004/0.029 = 0.138$
$\{a, c\} \to \{a, b, c\}$	$0.004/0.100 = 0.040$
$\{b, c\} \to \{a, b, c\}$	$0.004/0.007 = 0.571$

The problem with confidence

- Rule $\{c\} \rightarrow \{a, c\}$ has the second-highest confidence, 0.599
- So is it a good rule?

The problem with confidence

- Rule $\{c\} \to \{a, c\}$ has the second-highest confidence, 0.599
- So is it a good rule? No!
- Confidence has the same problem with episode rules as with association rules
- $\mathbb{P}(\{a, c\}|\{c\}) = 0.599 < 0.624 = \mathbb{P}(\{a\})$

Other goodness measures

- The definitions of goodness measures, such as lift γ, leverage δ, mutual information MI, and Fisher's exact test p_F can be straightforwardly extended to parallel injective episodes*
- Other kinds of episodes require more care: how to compute expected probability of a occurring after b assuming statistical independence?

*Rinta-Paavola, "Identifying Interesting Episode Patterns in User Interaction Log Data".

Redundant episode rules

- An episode rule is redundant if it conveys no more information than a simpler, more general rule
- Let A, B be disjoint parallel episodes and $D \preceq A$. The specialised rule $D \cup B \to A \cup B$ is **superfluous** given the more general rule $D \to A$ if the improvement on goodness measures is not statistically significant
- Test using conditional versions of Fisher's exact test or G-test $p_F(B \to B \cup C \mid D)$, $G = 2 \cdot \mathtt{fr}(D) \cdot MI(B \to B \cup C \mid D)$ where $C = A \setminus D$
- Note: much of the theory for dependency rules* applies also to parallel injective episode rules

*see e.g. Hämäläinen and Webb, "A tutorial on statistically sound pattern discovery"

Redundant episode rules

- Many episodes can be involved only in superfluous rules, and can be pruned already in the candidate generation pass
- Does not affect the results—we still have to prune the remaining superfluous rules at the end—but can speed up mining by about an order of magnitude*

*Rinta-Paavola, "Identifying Interesting Episode Patterns in User Interaction Log Data".

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
If there is $a \in A$ such that $\mathbb{P}(a \mid A \setminus \{a\}) = 1$, then

- all rules with A in the antecedent are superfluous given the corresponding rule with $A \setminus \{a\}$ in the antecedent,
- but it (or a superepisode) can still be a good consequent!
- That is, there may exist a rule $A \setminus \{b\} \to A$ that is **not** superfluous given $A \setminus \{a, b\} \to A \setminus \{a\}$

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\{x\} \qquad \{y\} \qquad \{z\}$$

$$\{x, y\} \qquad \{x, z\} \qquad \{y, z\}$$

$$\{x, y, z\}$$

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are **all rules** $B \to A$ be superfluous?

$$
\begin{tikzcd}
\{x\} & \{y\} & \{z\} \\
\{x, y\} \arrow[dr] & \{x, z\} \arrow[d] & \{y, z\} \arrow[dl] \\
& \{x, y, z\} &
\end{tikzcd}
$$

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \rightarrow A$ be superfluous?

$$\{x\} \qquad \{y\} \qquad \{z\}$$

$$\{x, y\} \qquad \{x, z\} \qquad \{y, z\}$$

$$\{x, y, z\}$$

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\begin{tikzcd}
\{x\} \arrow[dr] & \{y\} & \{z\} \\
\{x, y\} \arrow[dr] & \{x, z\} & \{y, z\} \\
 & \{x, y, z\} &
\end{tikzcd}$$

If $\mathbb{P}(z \mid x) = 1$, then $\mathbb{P}(z \mid \{x, y\}) = 1$
Rule $\{x, y\} \to \{x, y, z\}$ is superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$
\begin{tikzcd}
\{x\} & \{y\} \arrow[dr] & \{z\} \\
\{x, y\} \arrow[dr] & \{x, z\} & \{y, z\} \\
 & \{x, y, z\} &
\end{tikzcd}
$$

If $\mathbb{P}(z \mid y) = 1$, then $\mathbb{P}(z \mid \{x, y\}) = 1$
Rule $\{x, y\} \to \{x, y, z\}$ is superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\begin{tikzcd}
\{x\} \arrow[d] & \{y\} & \{z\} \\
\{x, y\} & \{x, z\} \arrow[d] & \{y, z\} \\
& \{x, y, z\} &
\end{tikzcd}$$

If $\mathbb{P}(y \mid x) = 1$, then $\mathbb{P}(y \mid \{x, z\}) = 1$
Rule $\{x, z\} \to \{x, y, z\}$ is superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\begin{tikzcd}
\{x\} & \{y\} & \{z\} \arrow[d] \\
\{x, y\} & \{x, z\} \arrow[d] & \{y, z\} \\
 & \{x, y, z\} &
\end{tikzcd}$$

If $\mathbb{P}(y \mid z) = 1$, then $\mathbb{P}(y \mid \{x, z\}) = 1$
Rule $\{x, z\} \to \{x, y, z\}$ is superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\begin{tikzcd}
\{x\} & \{y\} \arrow[dl] & \{z\} \\
\{x, y\} & \{x, z\} & \{y, z\} \arrow[dl] \\
 & \{x, y, z\} &
\end{tikzcd}$$

If $\mathbb{P}(x \mid y) = 1$, then $\mathbb{P}(x \mid \{y, z\}) = 1$
Rule $\{y, z\} \to \{x, y, z\}$ is superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$
\begin{tikzcd}
\{x\} & \{y\} & \{z\} \arrow[dl] \\
\{x, y\} & \{x, z\} & \{y, z\} \arrow[dl] \\
 & \{x, y, z\} &
\end{tikzcd}
$$

If $\mathbb{P}(x \mid z) = 1$, then $\mathbb{P}(x \mid \{y, z\}) = 1$
Rule $\{y, z\} \to \{x, y, z\}$ is superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\begin{tikzcd}
\{x\} \arrow[d] \arrow[dr] & \{y\} & \{z\} \arrow[dl] \\
\{x, y\} \arrow[dr] & \{x, z\} \arrow[d] & \{y, z\} \arrow[dl] \\
& \{x, y, z\} &
\end{tikzcd}$$

Informally: when, for all direct subepisodes $B \preceq A$, there is a sub-subepisode that 'makes' the rule $B \to A$ superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$\begin{tikzcd}
\{x\} \arrow[d] & \{y\} \arrow[dr] & \{z\} \arrow[dl] \\
\{x, y\} \arrow[dr] & \{x, z\} \arrow[d] & \{y, z\} \arrow[dl] \\
& \{x, y, z\} &
\end{tikzcd}$$

Informally: when, for all direct subepisodes $B \preceq A$, there is a sub-subepisode that 'makes' the rule $B \to A$ superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules?
Assume for sake of example: $A = \{x, y, z\}$
When are all rules $B \to A$ be superfluous?

$$
\begin{tikzcd}
\{x\} & \{y\} \arrow[dl] \arrow[dr] & \{z\} \arrow[d] \\
\{x, y\} \arrow[dr] & \{x, z\} \arrow[d] & \{y, z\} \arrow[dl] \\
 & \{x, y, z\} &
\end{tikzcd}
$$

Informally: when, for all direct subepisodes $B \preceq A$, there is a sub-subepisode that 'makes' the rule $B \to A$ superfluous

Redundant episode rules

How do we know an episode A is involved only in superfluous rules? Given a parallel injective episode A, if

$$\forall a \in A\, \exists b \in A \setminus \{a\} : \mathbb{P}(A \setminus \{a, b\}) = \mathbb{P}(A \setminus \{a\}) \vee \mathbb{P}(A \setminus \{a, b\}) = \mathbb{P}(A \setminus \{b\}),$$

then all rules involving A are superfluous given the corresponding rule involving $A \setminus \{e\}$ for some $e \in A$.*

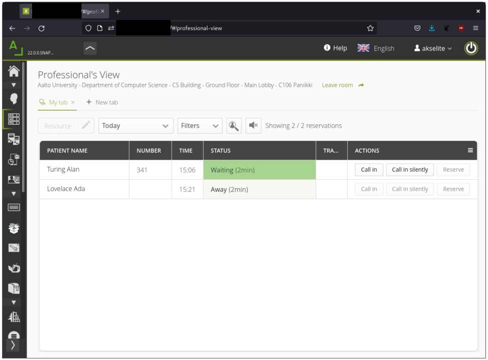
*Rinta-Paavola, "Identifying Interesting Episode Patterns in User Interaction Log Data".

A practical application: my thesis

- For my Master's thesis project*, I used WINEPI enhanced with goodness measures and superfluousness pruning to find interesting patterns of user interaction
- Background: the company, Axel Health, develops a patient flow management system whose purpose is to help patients and healthcare professionals get to the right place at the right time.
- Problem: we have little insight into how users use the system in reality
- Solution: collect log data from the *Professional's View* and analyse it using episode mining methods

*Rinta-Paavola, "Identifying Interesting Episode Patterns in User Interaction Log Data".

The event sequence: simplified example

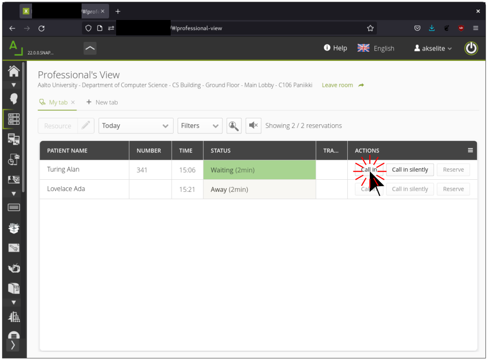


Logged event:

`RESERVATION_LIST_SHOW`

2023-10-10T17:01:51

The event sequence: simplified example

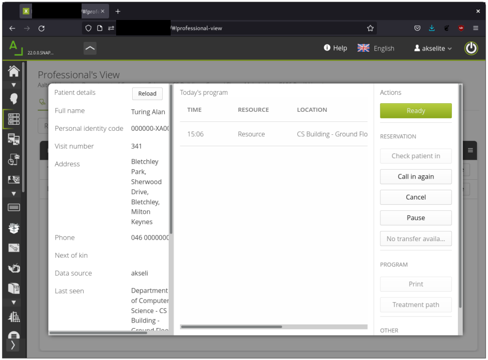


Logged event:

`CALL_IN(IS_SILENT=false)`

2023-10-10T17:15:07

The event sequence: simplified example

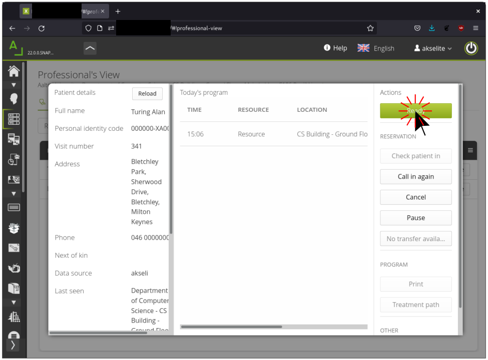


Logged event:

`RESERVATION_WINDOW`

2023-10-10T17:15:08

The event sequence: simplified example



Logged event:
`CALL_IN_CLOSE`
2023-10-10T17:59:27

Experiments

- Collected six weeks of data from six healthcare organisers: in total, more than two million events from more than three thousand users
- Preprocessing: parametric events were turned into two, one with parameters and one without; 'uninteresting' events discarded; concatenated event sequences from different users
- Mined episodes with frequency threshold $\mathbb{P}_{\text{thresh}} = 0.001$; tried window sizes from one minute to one hour
- Mined rules with confidence threshold $\varphi_{\text{thresh}} = 0$, so that we can analyse other goodness measures
- Post-processing: discarded rules that were technical artefacts of how the system works, that did not show a statistical dependence ($\delta \leq 0$), or that were superfluous at the significance threshold $\alpha = 0.01$

Results

- Top rules were obvious, corresponding to basic functionality
- Digging a bit deeper, we found some practically relevant rules that reveal parts of the UI that users have trouble with
- All goodness measures have their place; Fisher's exact test and G-test gave nearly identical results
- Superfluousness pruning was vital for both performance and ease of interpretation
- Choice of window size can be difficult; more rules found at larger sizes, but the rules in common at two sizes were ranked similarly

What I learned

- Pattern explosion is a difficult problem: despite pruning reducing the number of rules by up to over an order of magnitude, still impossible to go through all results
- Frequency-based mining is not ideal: most frequent patterns can easily be far from the most interesting ones, seldom-used features got drowned out in the noise
- Statistical significance has its problems: with enough data, you will find statistically significant differences that are so small they have no real-life significance*

*Example: one dataset had a rule that improved confidence and lift of its more general rule by 0.03 %, and was worse by other measures, but was deemed non-superfluous with $p \approx 4 \cdot 10^{-971}$

References I

Atallah, M., Gwadera, R., & Szpankowski, W. (2004). Detection of significant sets of episodes in event sequences. In R. Rastogi, K. Morik, M. Bramer & X. Wu (Eds.), *Proceedings of the fourth IEEE international conference on data mining* (pp. 3–10). IEEE Computer Society. https://doi.org/10.1109/icdm.2004.10090

Casas-Garriga, G. (2003). Discovering unbounded episodes in sequential data. In N. Lavrač, D. Gamberger, L. Todorovski & H. Blockee (Eds.), *Proceedings of the 7th European conference on principles and practice of knowledge discovery in databases. Lecture notes in artificial intelligence: Vol 2838* (pp. 83–94). Springer. https://doi.org/10.1007/978-3-540-39804-2_10

References II

Fournier-Viger, P., Chen, Y., Nouioua, F., & Lin, J. C.-W. (2021). Mining partially-ordered episode rules in an event sequence. In N. T. Nguyen, S. Chittayasothorn, D. Niyato & B. Trawiński (Eds.), *Proceedings of the 13th Asian conference on intelligent information and database systems. Lecture notes in artificial intelligence, Vol 12672* (pp. 3–15). Springer. https://doi.org/10.1007/978-3-030-73280-6_1

Gwadera, R., Atallah, M., & Szpankowski, W. (2003). Reliable detection of episodes in event sequences. In X. Wu, A. Tuzhilin & J. Shavlik (Eds.), *Proceedings of the third IEEE international conference on data mining* (pp. 67–74). IEEE Computer Society. https://doi.org/10.1109/icdm.2003.1250904

References III

Gwadera, R., Atallah, M., & Szpankowski, W. (2005). Markov models for identification of significant episodes. In H. Kargupta, J. Srivastava, C. Kamath & A. Goodman (Eds.), *Proceedings of the fifth SIAM international conference on data mining* (pp. 404–414). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611972757.36

Hämäläinen, W., & Webb, G. I. (2018). A tutorial on statistically sound pattern discovery. *Data Mining and Knowledge Discovery*, *33*(2), 325–377. https://doi.org/10.1007/s10618-018-0590-x

References IV

Laxman, S., Sastry, P. S., & Unnikrishnan, K. P. (2005). Discovering frequent episodes and learning hidden Markov models: A formal connection. *IEEE Transactions on Knowledge and Data Engineering*, *17*(11), 1505–1517. https://doi.org/10.1109/tkde.2005.181

Mannila, H., Toivonen, H., & Verkamo, A. I. (1995). Discovering frequent episodes in sequences. In U. Fayyad & R. Uthurusamy (Eds.), *Proceedings of the first international conference on knowledge discovery and data mining* (pp. 210–215). The AAAI Press. https://cdn.aaai.org/KDD/1995/KDD95-024.pdf

Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, *1*(3), 259–289. https://doi.org/10.1023/A:1009748302351

References V

Méger, N., & Rigotti, C. (2004). Constraint-based mining of episode rules and optimal window sizes. In J.-F. Boulicaut, F. Esposito, F. Giannotti & D. Pedreschi (Eds.), *Proceedings of the 8th European conference on principles and practice of knowledge discovery in databases. Lecture notes in artificial intelligence: Vol 3202* (pp. 313–324). Springer. https://doi.org/10.1007/978-3-540-30116-5_30

Rinta-Paavola, J. (2022). *Identifying interesting episode patterns in user interaction log data* [Master's thesis]. Aalto University. School of Science. http://urn.fi/URN:NBN:fi:aalto-202301291820

References VI

Tatti, N. (2009). Significance of episodes based on minimal windows. In J.-F. Boulicaut, F. Esposito, F. Giannotti & D. Pedreschi (Eds.), *Proceedings of the ninth IEEE international conference on data mining* (pp. 513–522). IEEE Computer Society. https://doi.org/10.1109/icdm.2009.23

Tatti, N., & Cule, B. (2011). Mining closed episodes with simultaneous events. In C. Apte, J. Ghosh & P. Smyth (Eds.), *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1172–1180). Association for Computing Machinery. https://doi.org/10.1145/2020408.2020589

Tatti, N., & Cule, B. (2012). Mining closed strict episodes. *Data Mining and Knowledge Discovery*, *25*(1), 34–66. https://doi.org/10.1007/s10618-011-0232-z