Mining Closed Episodes with Simultaneous Events

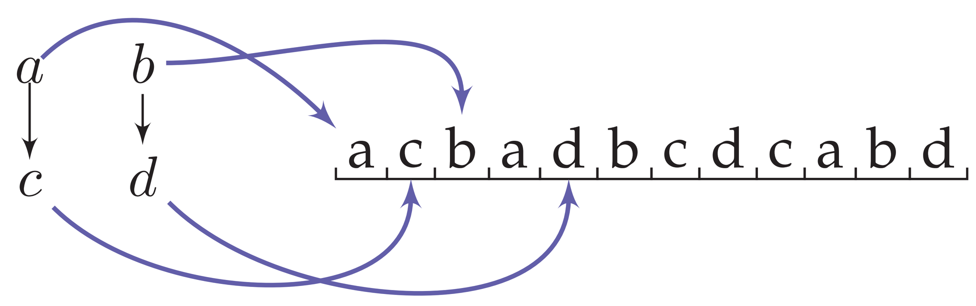
NIKOLAJ.TATTI@UA.AC.BE · BORIS.CULE@UA.AC.BE

Universiteit Antwerpen

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 event
- parents occur before children



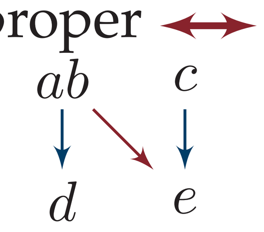
Support ⟷ number of fixed-size windows covering the episode

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 ⟷ event will follow or occur at the same time
- proper ⟷ event will follow properly



- a and b must occur together
- d follows or occurs at the same time as a
- e follows or occurs at the same time as c
- e must follow a (and b) properly

Goal & Approach

Find all episodes that have enough support

Key step ⟷ Support is higher for subepisodes

Discover episodes in depth-first style. Three different levels for travelsal
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

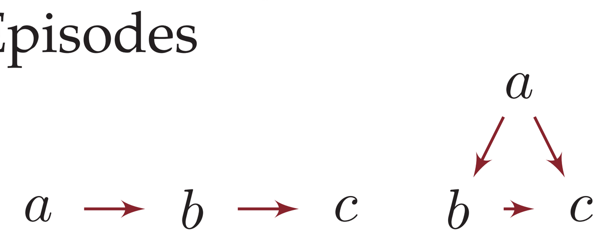
- Pattern Explosion:
  If
  $$a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_N$$
  is frequent, then at least

N	1	2	3	4	5	6
	1	4	16	84	652	7 742

N	7	8	9
	139 387	3 730 216	145 605 024

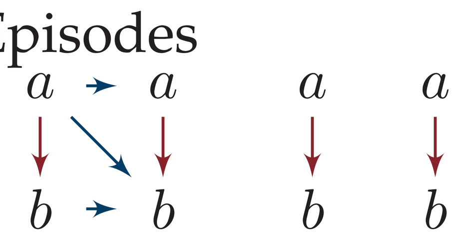
episodes are frequent

- Redundancy issues:
  Episodes



are same

Episodes



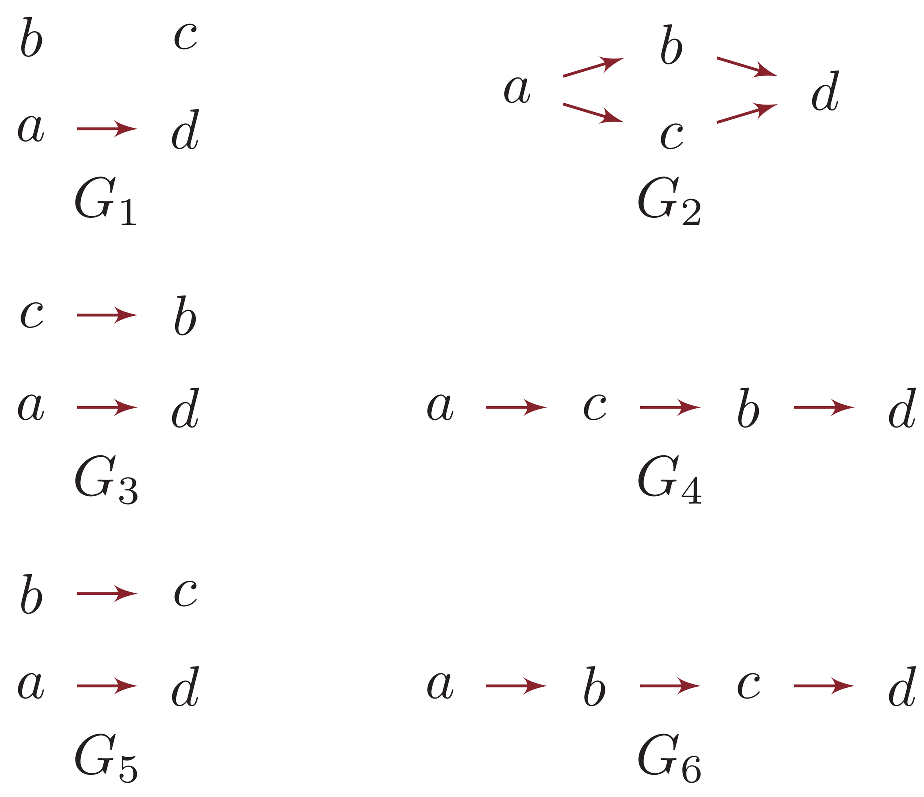
are same

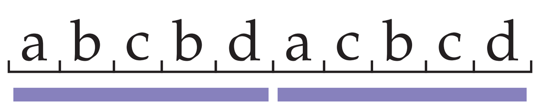
- Coverage test ⟷ **NP**-complete

Closed Episodes

Technique for reducing number of patterns

Closed pattern ⟷ no superpattern with the same support

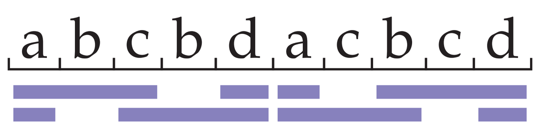




$G_1, \ldots, G_4$ have support 2. G_4, G_6 are closed

Problem ⟷ 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



G_1 and G_2 have 4 mappings, $G_3, \ldots, G_6$ have 2 mappings ⟶ 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
⟷ s covers H ⟶ s covers G.

Theorem: testing subset relation ⟷ **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 $G \subseteq H$
- 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

w (s)	$\sigma/10^3$	s-cl.	m-cl.	freq.(est)
180	500	6	6	6
180	400	8	8	8
180	300	12	12	12
180	240	23	26	26
600	2 000	4	4	4
600	1 000	24	27	39
600	500	90	137	493
600	280	422	698	2 321
900	2 000	24	26	40
900	1 000	52	58	94
900	500	280	426	1 997
900	350	1 845	9 484	190 990

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

σ	s-cl.	m-cl.	freq.(est)
30 000	141	141	141
20 000	1 994	1 995	2 593
17 000	8 352	8 416	22 542
15 000	26 170	26 838	172 067
13 000	94 789	101 882	3 552 104
12 000	189 280	211 636	33 660 094