Sampling for finding association rules

- two causes for complexity
- lots of attributes
- lots of rows
- potentially exponential in the number of attributes
- linear in the number of rows
- too many rows: take a sample from them
- in detail later

Chapter 3: Alarm correlation

Part II. Episodes in sequences

- Chapter 3: Alarm correlation
- Chapter 4: Frequent episodes
- Chapter 5: Minimal occurrences of episodes
- Chapter 6: Episode discovery process

3. Alarm correlation: networks and alarms

- network elements: switches, base stations, transmission equipment, etc.
- 10–1000 elements in a network
- an alarm: a message generated by a network element

```
1234  EL1  BTS  940926  082623  A1  Channel missing
```

- hundreds of different alarm types
- 200 – 10000 alarms a day
- each contains only local information

Characteristics of the alarm flow

- a variety of situations
- bursts of alarms
- hardware and software change fast

Alarm correlation

"correlating" alarms: combining the fragmented information contained in the alarm sequence and interpreting the whole flow of alarms

- removing redundant alarms
- filtering out low-priority alarms
- replacing alarms by something else
- systems exist
  - knowledge base (correlation rules) constructed manually
  - look at the alarms occurring in a given time window
  - apply actions given in the matching correlation rules

Problem

- how to obtain the information needed for the preparation of an alarm correlation system
- more generally: how to obtain insight into the behavior of the network (alarms)

Solutions

- how to analyze a flow of alarms?
- lots of possibilities: hazard models, neural networks, rule-based representations
- comprehensibility of the discovered knowledge
- simple rule-based representations
- "if certain alarms occur within a time window, then a certain alarm will also occur"

Episodes

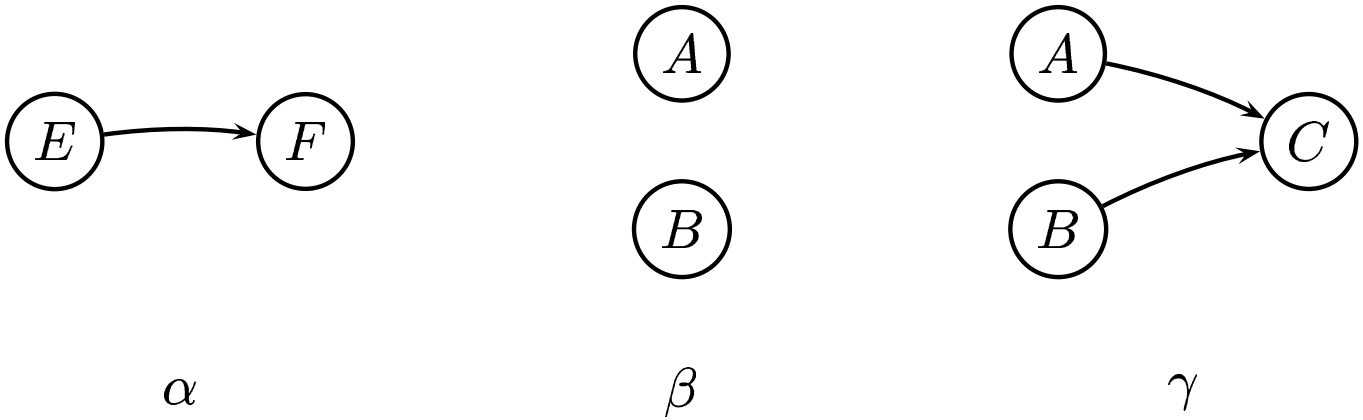


Figure 3.2: Episodes

Basic solution

- look for repeated occurrences of episodes in the alarm flow sequences
- occurrence: alarms of the specified type occur in the specified order
- why this form?
  - comprehensible
  - "standard" for correlation systems
  - represent simple causal relationships
  - insensitive to inaccurate clocks
  - allows analysis of merged, unrelated sequences

Chapter 4: Episodes

4. Frequent episodes

- The framework
- Algorithms
- Experiments

Example sequence

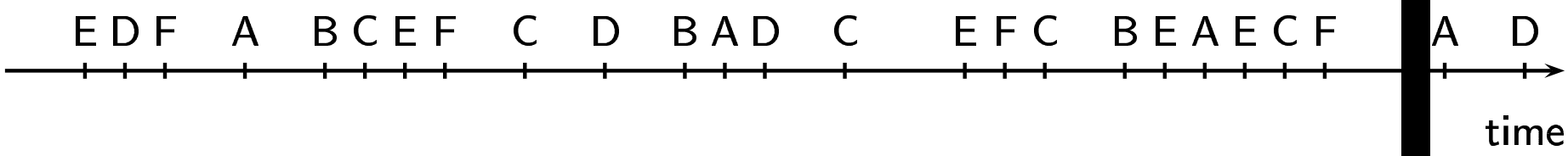


Figure 3.1: A sequence of alarms

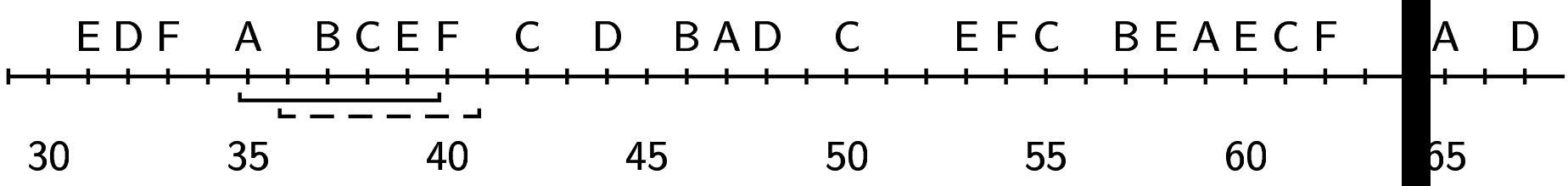
Observations:

- whenever E occurs, F occurs soon
- whenever A and B occur (in either order), C occurs soon

Data

- a set R of *event types*
- an *event* is a pair (A, t)
- $A \in R$ is an event type
- t is an integer, the *(occurrence) time* of the event
- *event sequence* s on R: a triple (s, T_s, T_e)
- $T_s < T_e$ are integer (starting and ending time)
- $s = \langle (A_1, t_1), (A_2, t_2), \ldots, (A_n, t_n) \rangle$
- $A_i \in R$ and $T_s \leq t_i < T_e$ for all $i = 1, \ldots, n$
- $t_i \leq t_{i+1}$ for all $i = 1, \ldots, n-1$

Example



Figu

4.1: The example event sequence s and two windows of width 5.

Windows

- event sequence $\mathbf{s} = (s, T_s, T_e)$
- a window on it: $\mathbf{w} = (w, t_s, t_e)$
- $t_s < T_e, t_e > T_s$
- w consists of those pairs (A, t) from s where $t_s \leq t < t_e$
- $width(\mathbf{w}) = t_e - t_s$: the *width* of the window $\mathbf{w}$
- $\mathcal{W}(\mathbf{s}, win)$: all windows $\mathbf{w}$ on $\mathbf{s}$ such that $width(\mathbf{w}) = win$
- first and last windows!

Episodes

- an *episode* α is a triple $(V, \leq, g)$
- V is a set of nodes
- $\leq$ is a partial order on V
- $g : V \to R$ is a mapping associating each node with an event type
- intuition: the events in $g(V)$ have to occur in the order described by $\leq$
- *size* of α, denoted $|\alpha|$, is $|V|$
- *parallel episode*: the partial order $\leq$ is trivial
- *serial episode*: $\leq$ is a total order
- *injective*: no event type occurs twice in the episode

Example

α

Figure 4.2: An episode

the set V, the mapping g

Example, subepisode

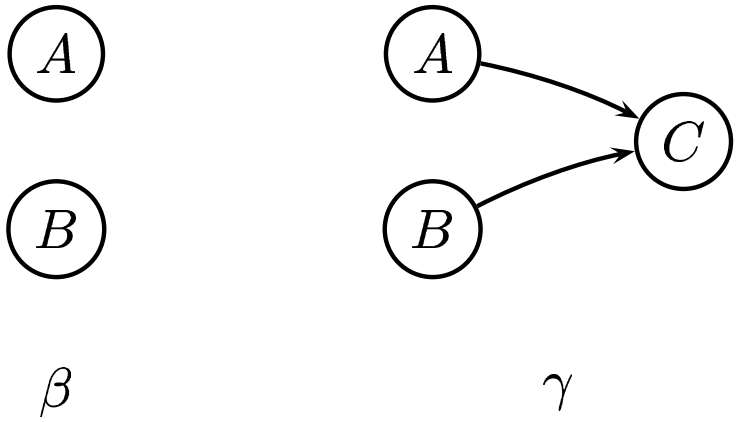


Figure 4.3: A subepisode and episode

Subepisodes

$\beta = (V', \leq', g')$ is a *subepisode* of $\alpha = (V, \leq, g)$, $\beta \preceq \alpha$, if:

there exists an injective mapping $f : V' \to V$ such that

- $g'(v) = g(f(v))$ for all $v \in V'$
- for all $v, w \in V'$ with $v \leq' w$ also $f(v) \leq f(w)$

An episode α is a *superepisode* of β if and only if $\beta \preceq \alpha$

$\beta \prec \alpha$ if $\beta \preceq \alpha$ and $\alpha \not\preceq \beta$

In the example: $\beta \preceq \gamma$

Occurrences of episodes

$\alpha = (V, \leq, g)$ *occurs* in an event sequence
$\mathbf{s} = (\langle (A_1, t_1), (A_2, t_2), \ldots, (A_n, t_n) \rangle, T_s, T_e)$, if there exists an injective mapping $h : V \to \{1, \ldots, n\}$ from nodes to events, such that

- $g(x) = A_{h(x)}$ for all $x \in V$
- for all $x, y \in V$ with $x \neq y$ and $x \leq y$ we have $t_{h(x)} < t_{h(y)}$ (or $h(x) < h(y)$)

$(w, 35, 40)$ on the example sequence: events of types A, B, C, and E

both β and γ occur

Frequency of occurrence

- the *frequency* of an episode α in $\mathbf{s}$ is

$$fr(\alpha, \mathbf{s}, win) = \frac{|\{\mathbf{w} \in \mathcal{W}(\mathbf{s}, win) \mid \alpha \text{ occurs in } \mathbf{w}\}|}{|\mathcal{W}(\mathbf{s}, win)|},$$

- i.e., the fraction of windows on $\mathbf{s}$ in which α occurs.
- a *frequency threshold min_fr*
- α is *frequent* if $fr(\alpha, \mathbf{s}, win) \geq min\_fr$
- $\mathcal{F}(\mathbf{s}, win, min\_fr)$: collection of frequent episodes in $\mathbf{s}$ with respect to *win* and *min_fr*
- size $= l$: $\mathcal{F}_l(\mathbf{s}, win, min\_fr)$.

Pattern discovery task

given an event sequence $\mathbf{s}$, a set $\mathcal{E}$ of episodes, a window width *win*, and a frequency threshold *min_fr*, find $\mathcal{F}(\mathbf{s}, \mathit{win}, \mathit{min\_fr})$

Algorithms

Algorithm 4.13

Input: A set R of event types, an event sequence $\mathbf{s}$ over R, a set $\mathcal{E}$ of episodes, a window width *win*, and a frequency threshold *min_fr*.

Output: The collection $\mathcal{F}(\mathbf{s}, \mathit{win}, \mathit{min\_fr})$ of frequent episodes.

Method:

1. compute $\mathcal{C}_1 := \{\alpha \in \mathcal{E} \mid |\alpha| = 1\}$;
2. $l := 1$;
3. **while** $\mathcal{C}_l \neq \emptyset$ **do**
4. // Database pass (Algorithms 4.19 and 4.21):
5. compute $\mathcal{F}_l(\mathbf{s}, \mathit{win}, \mathit{min\_fr}) := \{\alpha \in \mathcal{C}_l \mid \mathit{fr}(\alpha, \mathbf{s}, \mathit{win}) \geq \mathit{min\_fr}\}$;
6. $l := l + 1$;
7. // Candidate generation (Algorithm 4.14):
8. compute $\mathcal{C}_l := \{\alpha \in \mathcal{E} \mid |\alpha| = l, \text{ and } \beta \in \mathcal{F}_{|\beta|}(\mathbf{s}, \mathit{win}, \mathit{min\_fr}) \text{ for all } \beta \in \mathcal{E} \text{ such that } \beta \prec \alpha \text{ and } |\beta| < l\}$;
9. **for** all l **do** output $\mathcal{F}_l(\mathbf{s}, \mathit{win}, \mathit{min\_fr})$;

Basic lemma, once again

Lemma 4.12 If an episode α is frequent in an event sequence $\mathbf{s}$, then all subepisodes $\beta \preceq \alpha$ are frequent. □

Parallel, serial, injective episodes

- *parallel episode*: the partial order $\leq$ is trivial
  (= frequent sets)
- *serial episode*: $\leq$ is a total order
  (= frequent subsequence)
- *injective*: no event type occurs twice in the episode (= proper sets, not multi sets)
- useful cases: (serial or parallel) [injective] episodes
  - reduce redundancy in generated episodes
  - keep episodes comprehensible
  - simpler to implement

Generation of candidate episodes

- parallel episodes, serial episodes (injective or non-injective)
- same idea as for association rules
- a candidate episode has to be a combination of two episodes of smaller size
- very small variations to the candidate generation procedure

Recognizing episodes in sequences

- first problem: given a sequence and an episode, find out whether the episode occurs in the sequence
- finding the number of windows containing an occurrence of the episode can be reduced to this
- successive windows have a lot in common
- how to use this?
- an incremental algorithm