DATA MINING IN TIME RELATED DATA

Time Series Data Mining

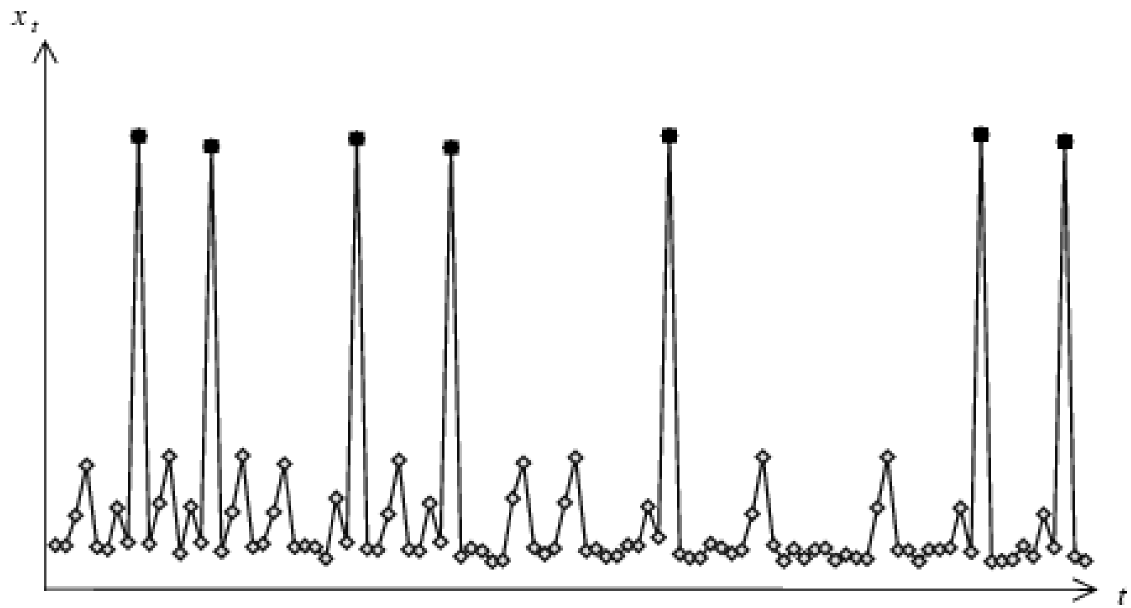
- Data mining concepts to analyzing time series data
- Revels hidden patterns that are characteristic and predictive time series events
- Traditional analysis is unable to identify complex characteristics (complex, non-periodic, irregular, chaotic)

Time series

- „a sequence of observed data, usually ordered in time”
- $X=(x_t,\ t=1..N)$

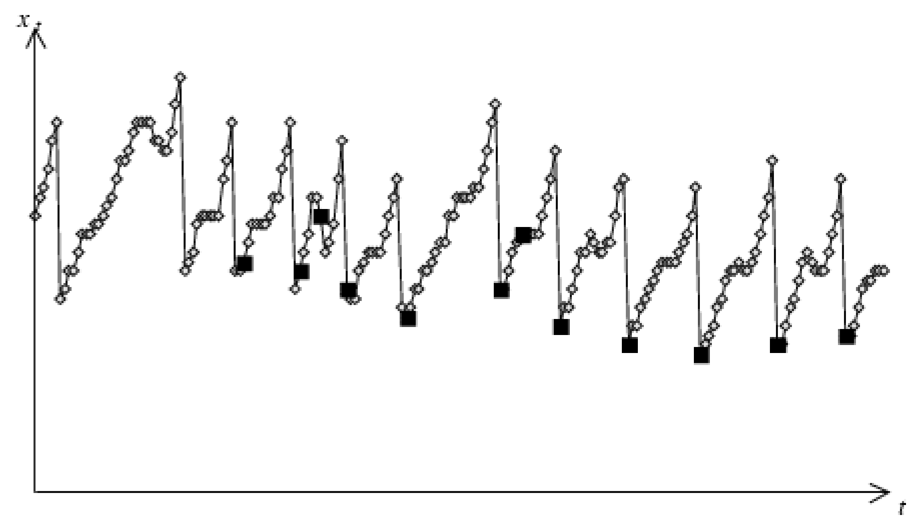
Example 1: seismic time series

- Diamonds = observations
  - E.g. Seismic activity
- Squares = important observations = events
  - E.g. Earthquakes
- Goal: to characterize, when peeks occur



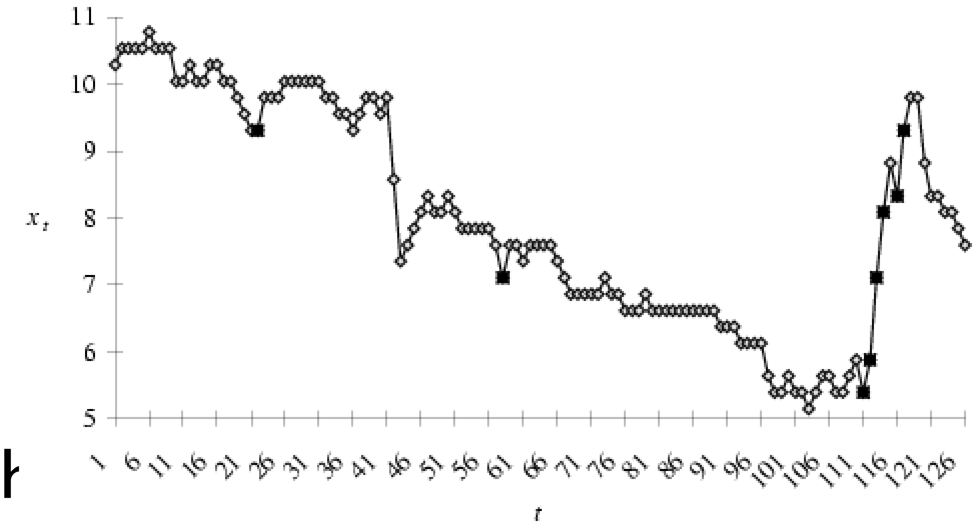
Example 2: welding time series

- Diamonds: measured stickout length of droplet (in pixels)
- Squares: droplet release (chaotic, noisy, irregular nature – impossible using traditional methods)
- Goal: prediction of release of metal droplet



Example 3: stock prices

- Diamonds: daily open price
- Squares: days when price increases more than 5%
- Goal: to find hidden patterns that provide th desired trading edge

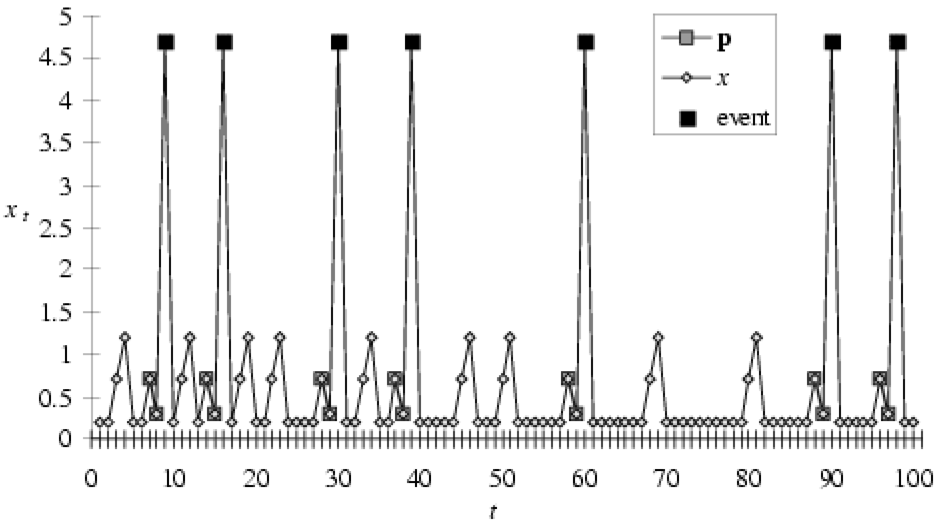


Event = important occurrence

- Ex1: earthquake
- Ex2: release of the droplet
- Ex3: sharp rise (fall) of stock price

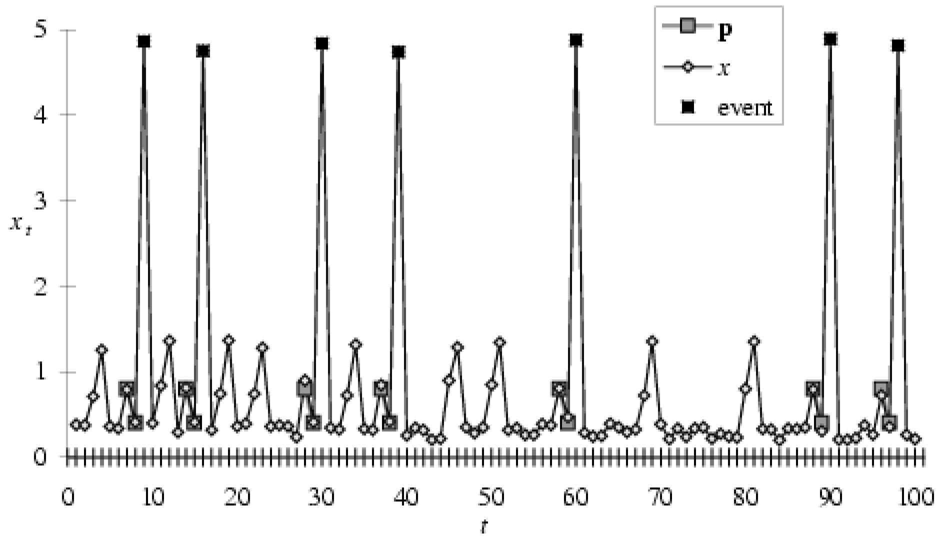
Temporal pattern

- Hidden structure in time series that is characteristic and predictive of events
- Temporal pattern **p** = real vector of length Q



Temporal pattern cluster

- Temporal patterns usually do not match time series
- TPC is a set of all points within delta from temporal pattern: $P=\{a \in \mathbf{R}^Q: d(\mathbf{p}, a) \leq \delta\}$

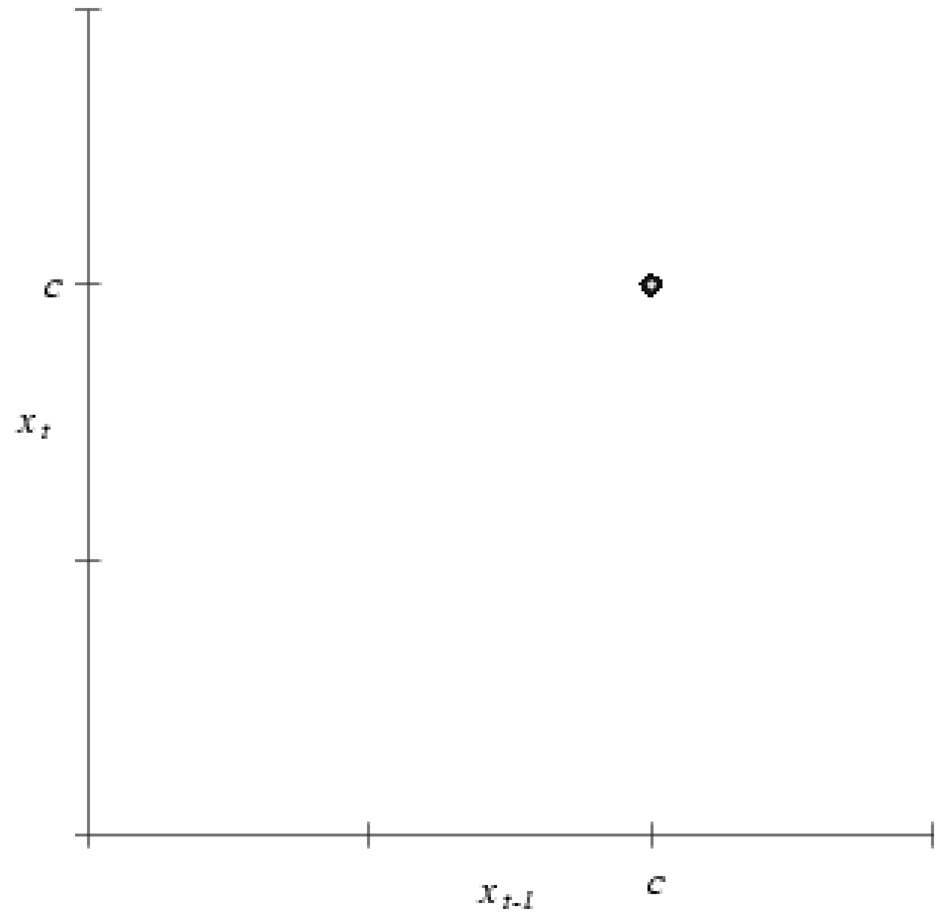


Phase space

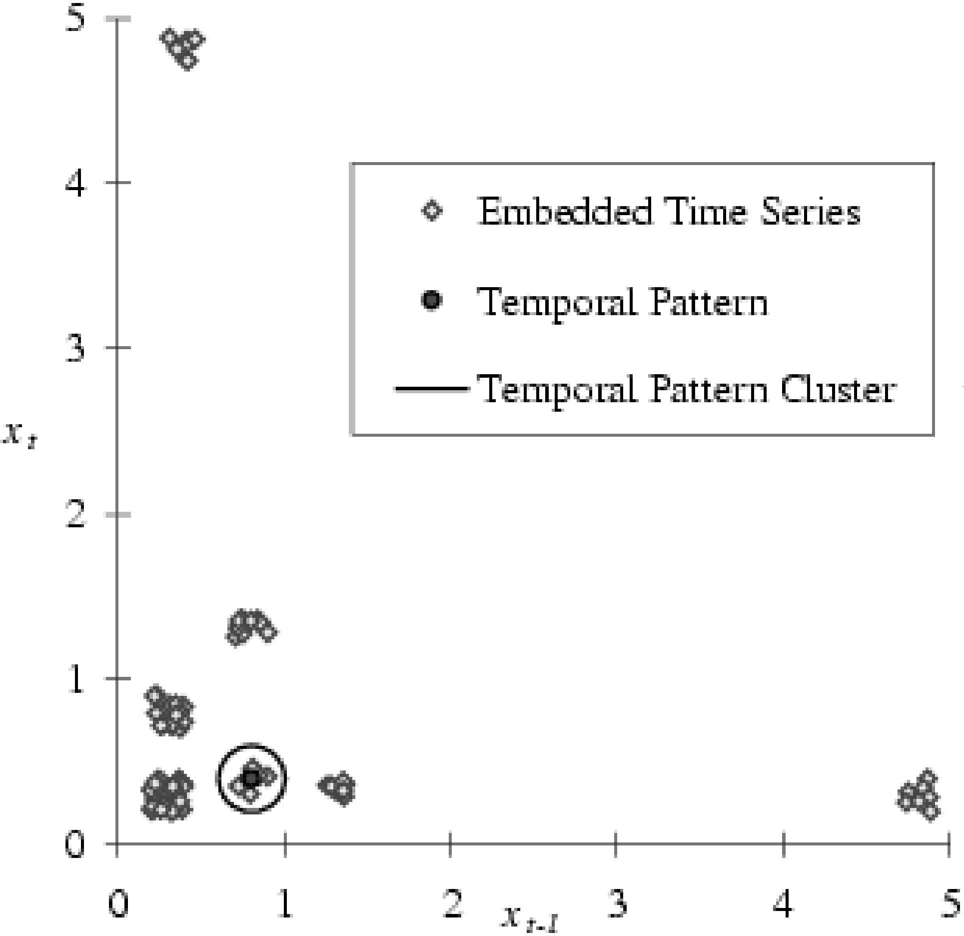
- Q dimensional metric space embedding time series
- Mapping of set of Q observations of time series into $x_t=(x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t)$

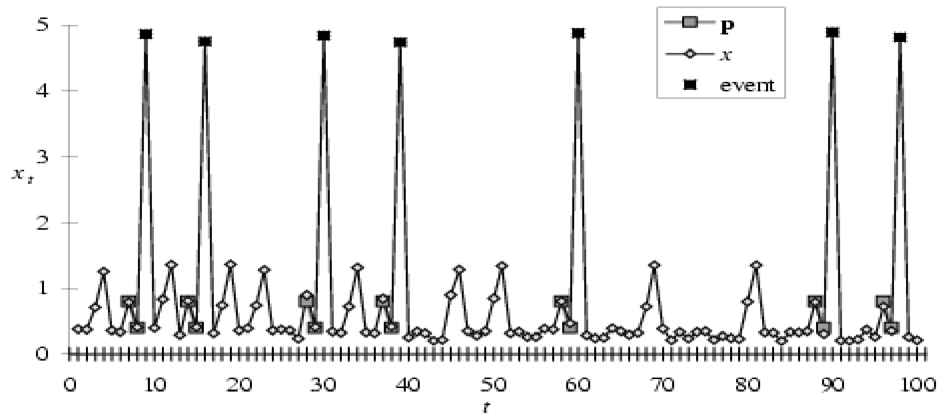
Phase space example - constant

- $X=\{x_t=c: t=1..N\}$
- $\tau=1$, Q=2

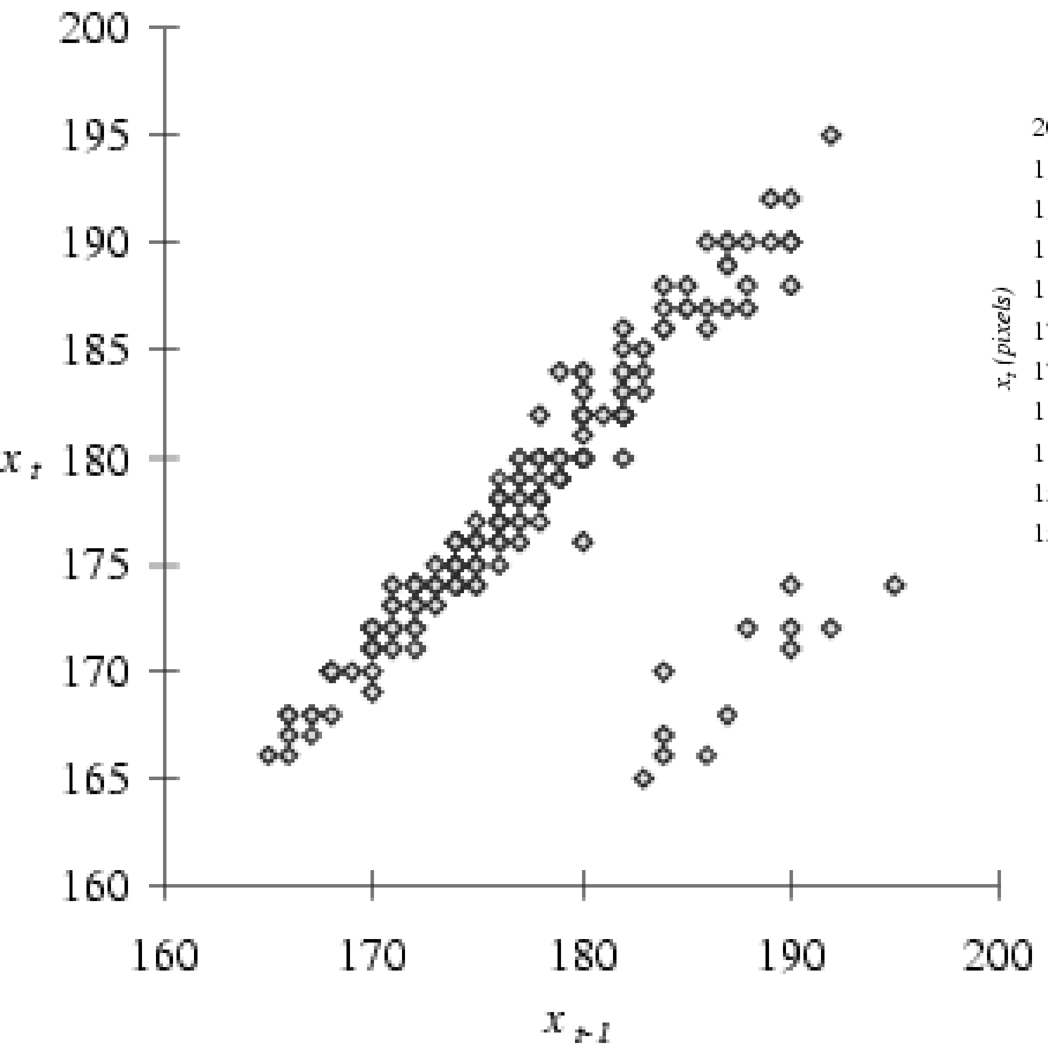


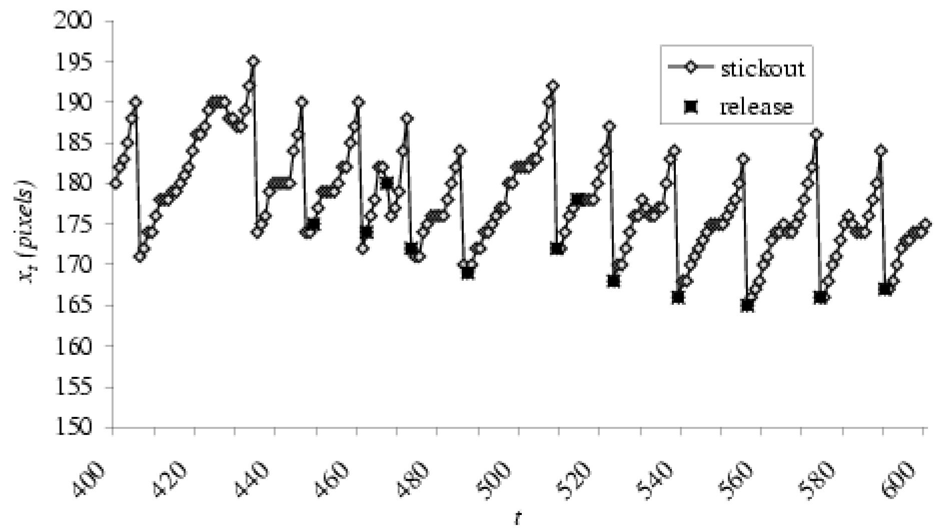
Phase space example - seismic



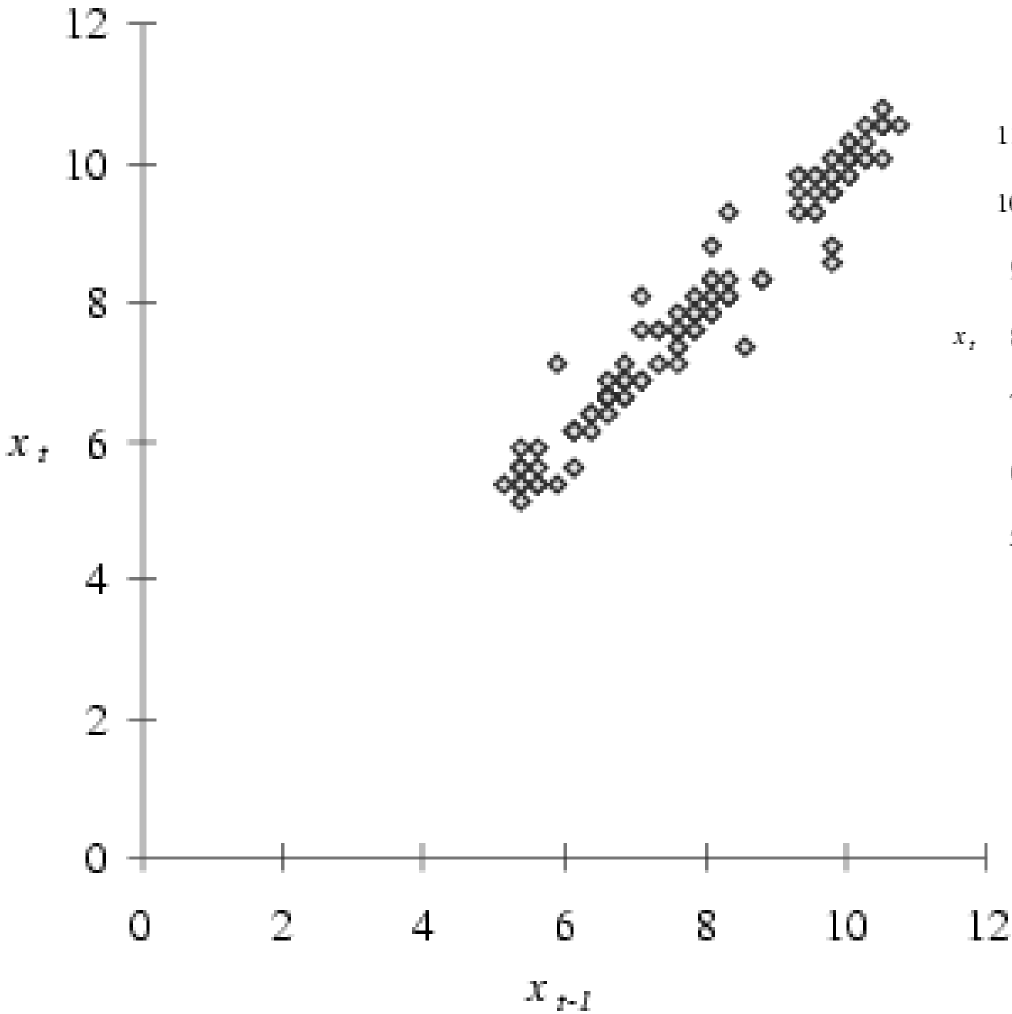


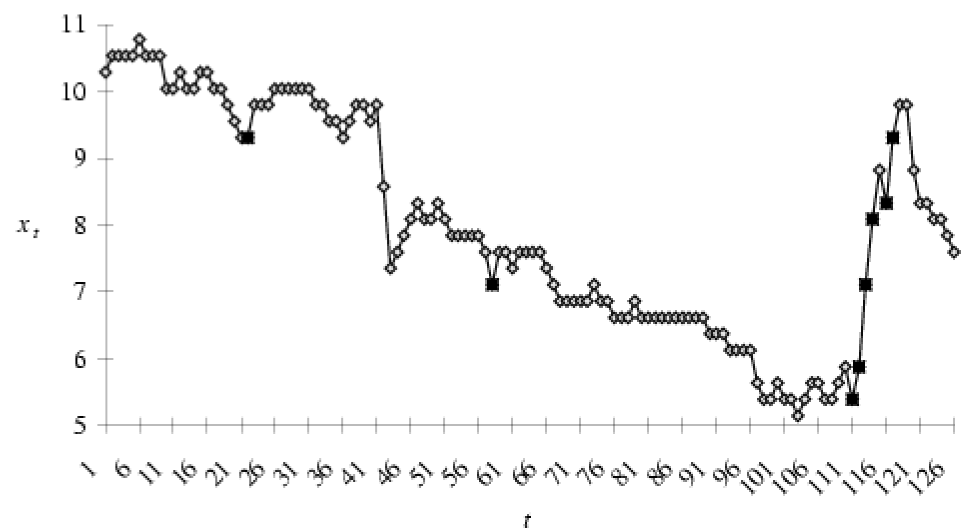
Phase space example - welding





Phase space example – stock open price





Event characterization function

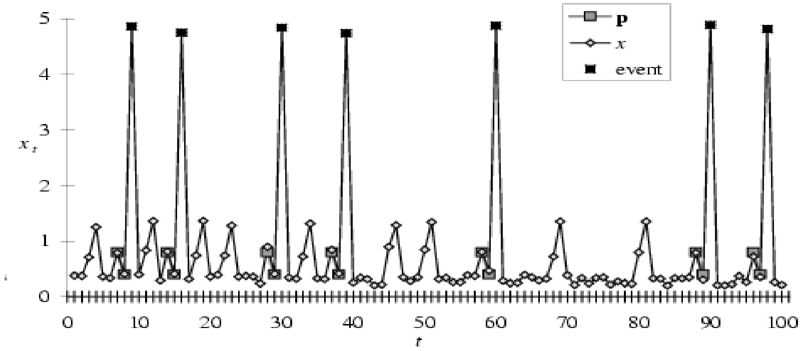
- Represents the value of future „eventness" for current time index
- Addresses the specific goal
- Examples:
  $g(t)=x_{t+1}$;
  $g(t)=x_{t+3}$;
  $g(t)=\max\{x_{t+1}, x_{t+2}, x_{t+3}\}$
- Welding: $g(t)=y_{t+1}$;
- Stock prices change: $g(t)=(x_{t+1}-x_t)/x_t$

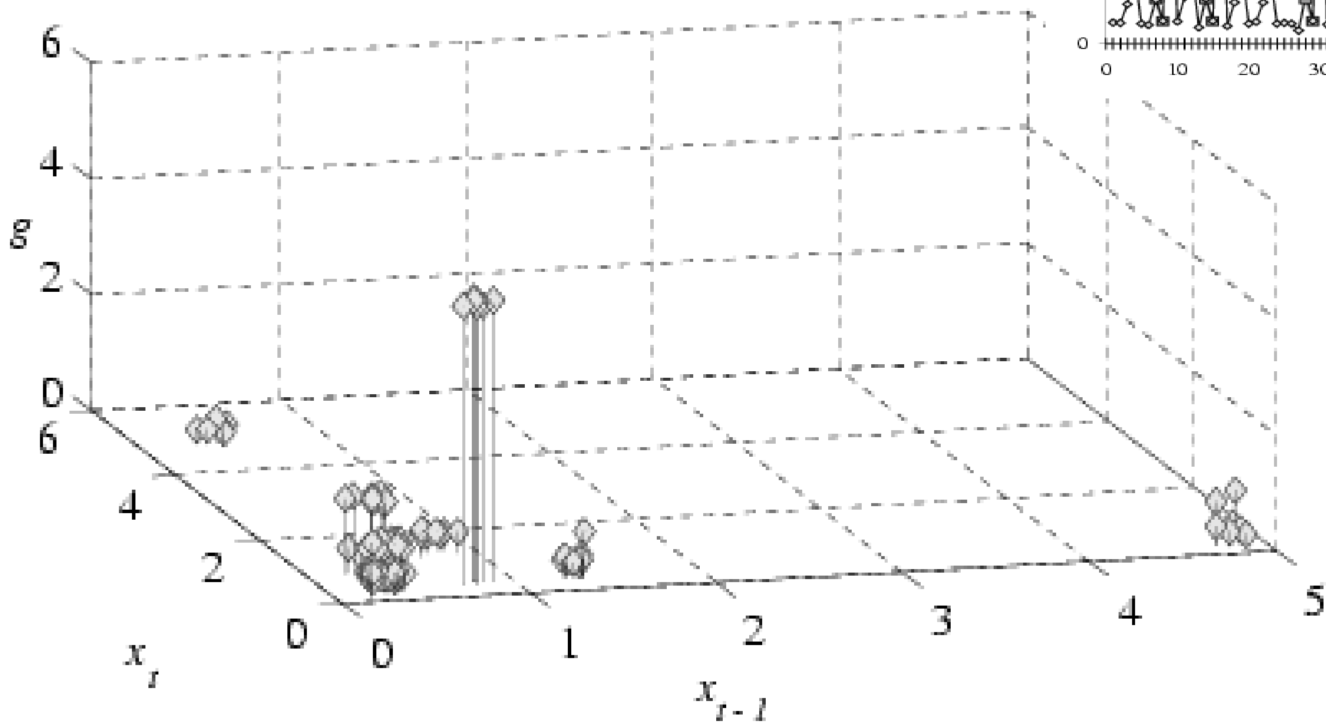
Augmented Phase space

- Q+1 dimensional space formed by extending phase space with $g(\cdot)$ = space of vectors $<\mathbf{x}_t, g(t)> \in \mathbf{R}^{Q+1}$

Augmented Phase space example

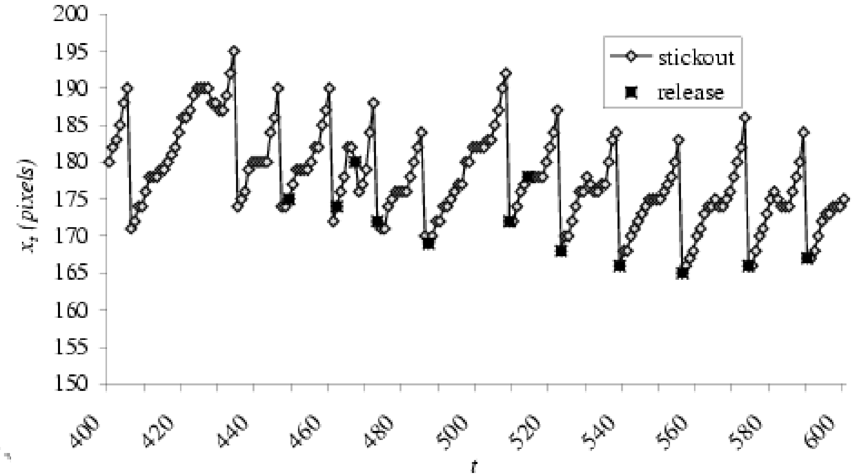
- seismic

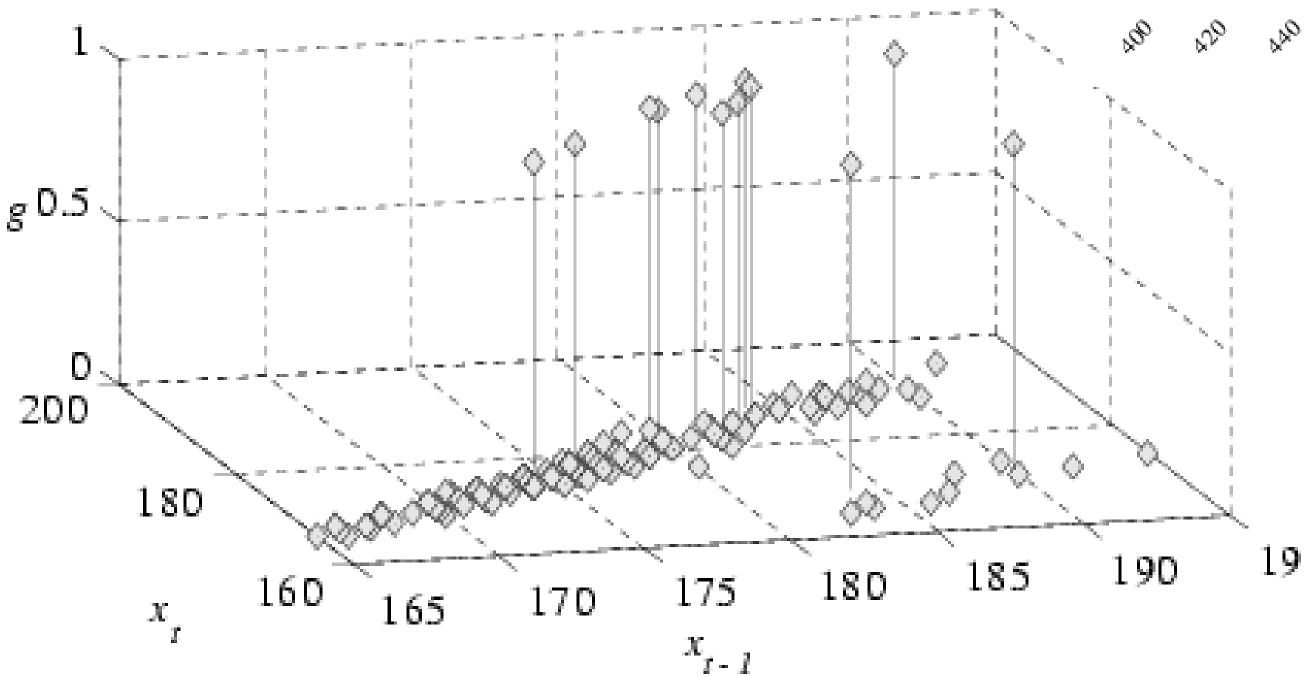




Augmented Phase space example

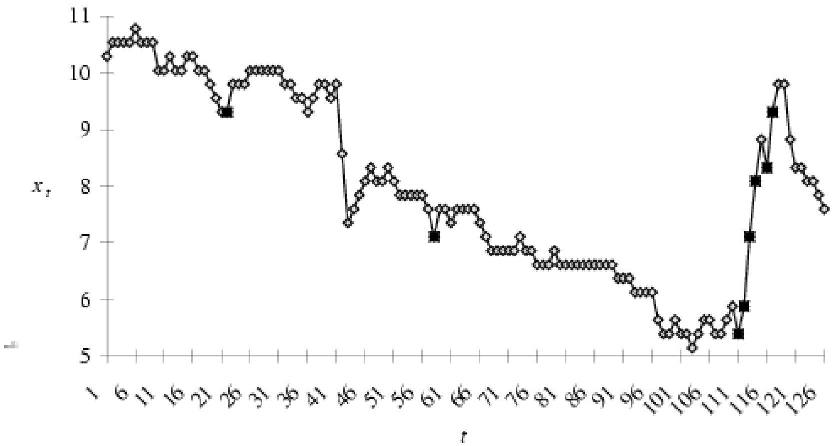
- welding

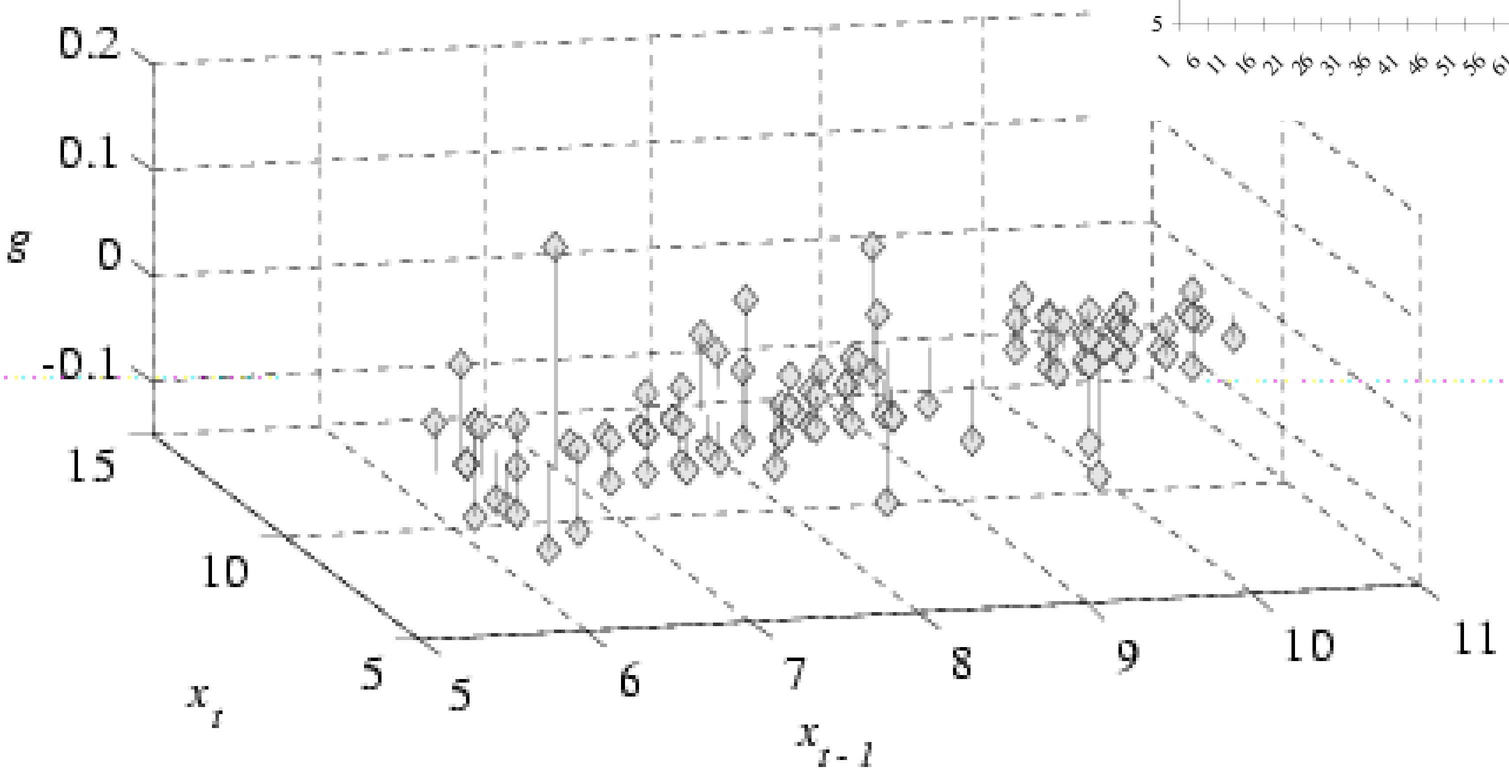




Augmented Phase space example

- stock open price





Objective function

- Measures how a temporal pattern cluster characterizes events
- M ($\tilde{M}$) – set of all time indices t when $\mathbf{x}_t$ is within (outside) temporal pattern cluster P

$$M = \{t: \mathbf{x}_t \in P,\ t \in \Lambda\}$$

$$\mu_M = \frac{1}{card(M)} \sum_{t \in M} g(t)$$

$$\sigma_M^2 = \frac{1}{card(M)} \sum_{t \in M} (g(t) - \mu_M)^2$$

Objective function

- *t* test for the difference between two independent means (for statistically significant and high average eventness clusters)

$$f(P) = \frac{\mu_M - \mu_{\tilde{M}}}{\sqrt{\frac{\sigma_M^2}{card(M)} - \frac{\sigma_{\tilde{M}}^2}{card(\tilde{M})}}}$$

Objective function

- When every event is required to be predicted by temporal pattern
- *g*() is binary
- *C* - collection of temporal pattern clusters
- Ratio of correct predictions to all predictions

$$f(C) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

- $t_p = \mathrm{card}(\{\mathbf{x}_t : \exists P_i \in C \; \mathbf{x}_t \in P_i \wedge g(t)=1\})$
- $f_p = \mathrm{card}(\{\mathbf{x}_t : \exists P_i \in C \; \mathbf{x}_t \in P_i \wedge g(t)=0\})$
- $t_n = \mathrm{card}(\{\mathbf{x}_t : \forall P_i \in C \; \mathbf{x}_t \notin P_i \wedge g(t)=1\})$
- $f_n = \mathrm{card}(\{\mathbf{x}_t : \forall P_i \in C \; \mathbf{x}_t \notin P_i \wedge g(t)=0\})$

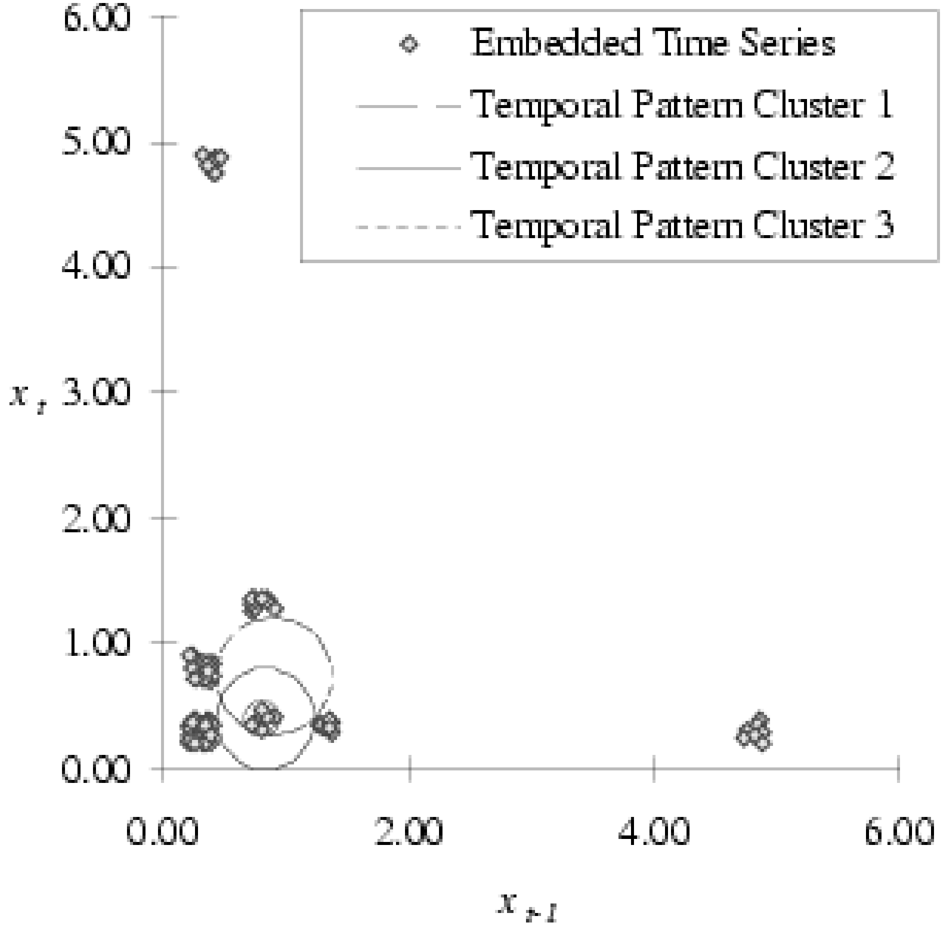
Optimization problem

$$\max_{\mathbf{x},\delta} f(p)$$

Genetic Algorithm

- Chromosome consists of Q+1 genes
- E.g. Q=2
- (x_{t-1}, x_t, δ)

Seismic example



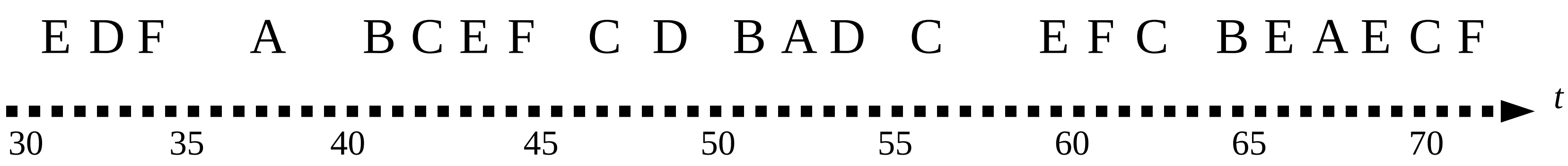
DISCOVERY OF FREQUENT EPISODES IN EVENT SEQUENCES

Events, event sequences

- event: (A,t) $A \in E$
- event sequence **s** on E: (s, T_s, T_e)

$$s = \langle (A_1,t_1),(A_2,t_2),\ldots,(A_n,t_n) \rangle$$

- window on **s**: $\mathbf{w}=(w,t_s,t_e)$, $t_s < T_e$, $t_e > T_s$
- $width(\mathbf{w}) = t_e - t_s$



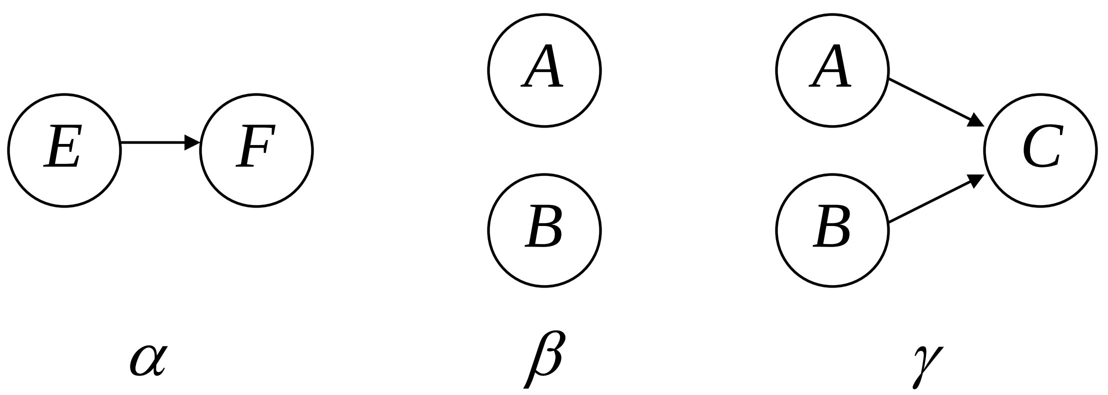
Episodes

- Collection of events occurring together
- serial, parallel, non-serial & non-parallel
- $(V, \leq, g)$

  V – set of nodes

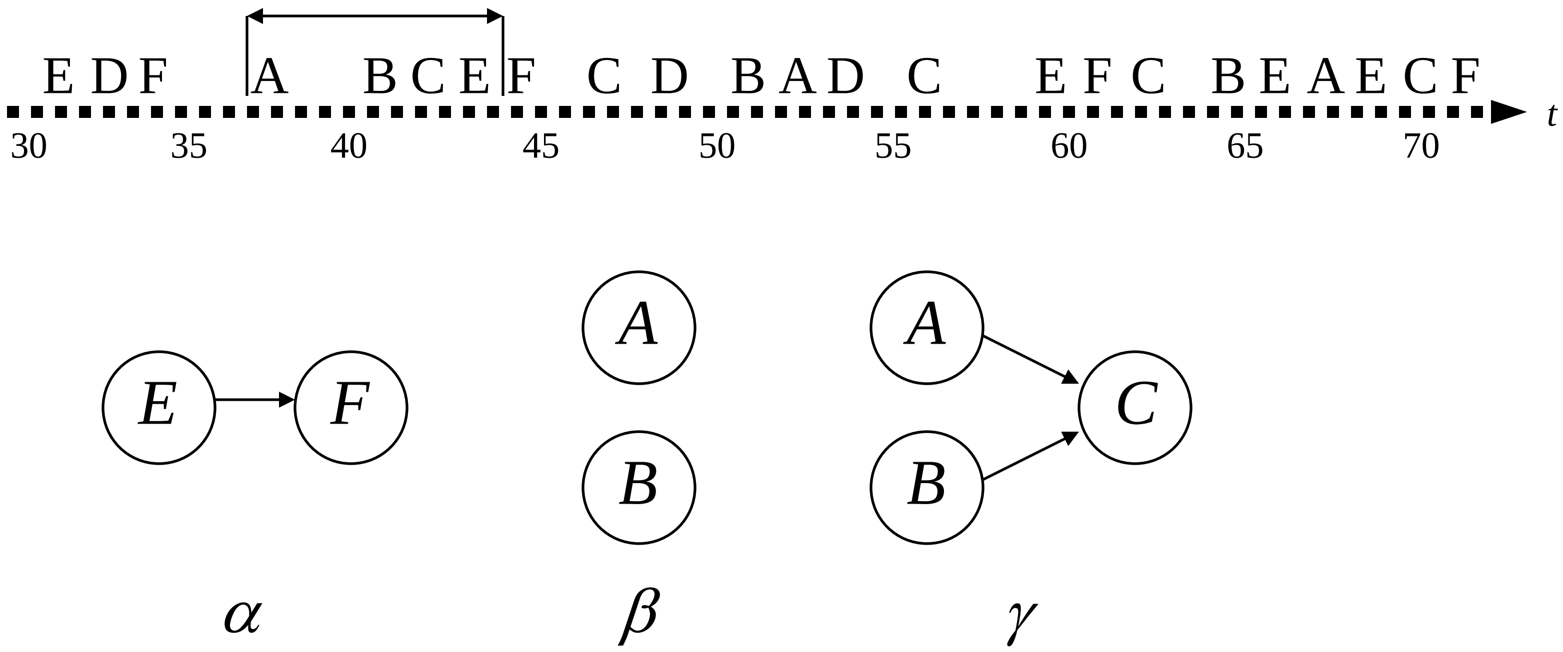
  $\leq$ – partial order on V

  $g{:}V \rightarrow E$ mapping associating each node with event type



Occurrence of episodes

- **w**=(w,37,44)



Frequency of an episode

- W(**s**,*win*) – all windows in **s** of length *win*

$$fr(\alpha, \mathbf{s}, win) = \frac{card(\{\mathbf{w} \in W(\mathbf{s}, win) : \alpha \text{ occurs in } \mathbf{w}\})}{card(W(\mathbf{s}, win))}$$

Goal

- Given (1) a frequency threshold *min_fr*, (2) window width *win*, discover all episodes α (from a given class of episodes) such that

$$fr(\alpha,\mathbf{s},win) \geq min\_fr$$

Episode rule generation algorithm

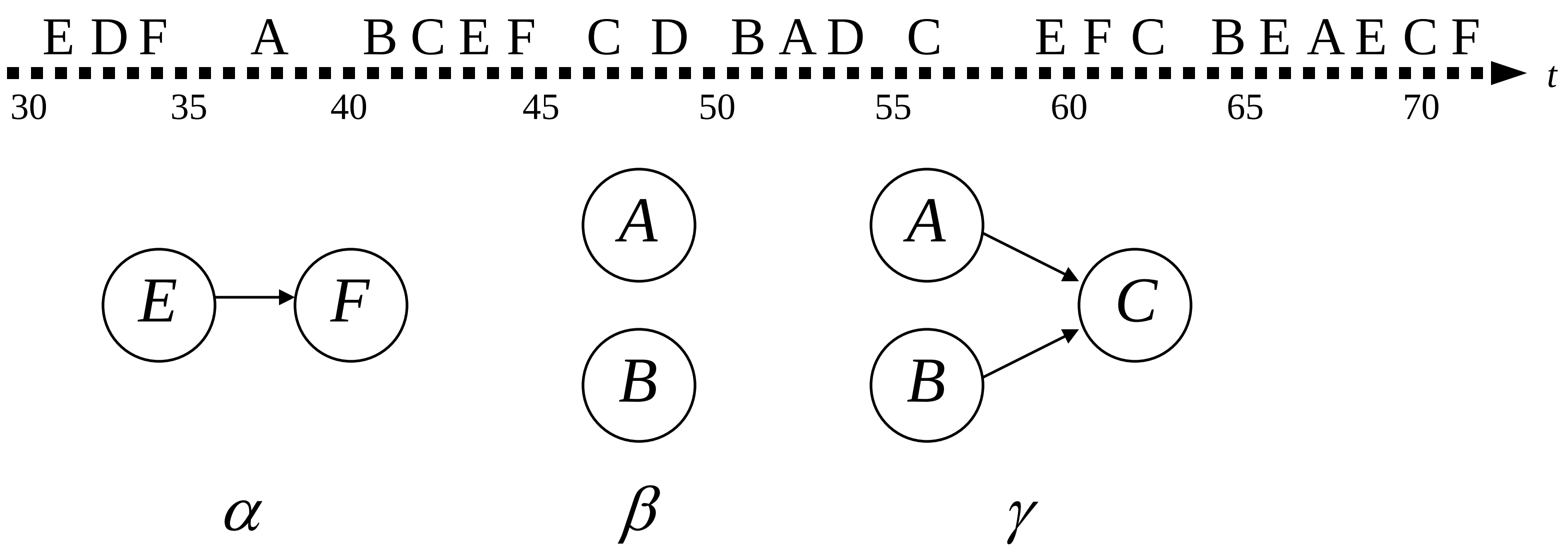
INPUT: event sequence **s**, *win*, *min_fr*, confidence threshold *min_conf*

OUTPUT: Episode rules that hold in **s** with respect to *win*, *min_fr*, *min_conf*

1. /* find all frequent episodes */
2. compute $\mathcal{F}$(**s**,*win*,*min_fr*)
3. /* generate rules */
4. **for** all $\alpha \in \mathcal{F}$(**s**,*win*,*min_fr*) **do**
5. **for** all $\beta \prec \alpha$ **do**
6. **if** $fr(\alpha)/fr(\beta) \geq min\_conf$ **then**
7. output the rule $\beta \rightarrow \alpha$ and the conf. $fr(\alpha)/fr(\beta)$

Example

- $\beta \prec \gamma$
- if we know that β occurs in 4.2% of windows and γ in 4.0% we can estimate that after seeing a window with A and B there is a chance 0.95 that C follows in the same window.



Frequent episode generation algorithm

INPUT: event sequence **s**, *win*, *min_fr*

OUTPUT: Collection $\mathcal{F}(\mathbf{s}, win, min\_fr)$ of frequent episodes

1. compute $C_1 = \{\alpha : |\alpha| = 1\}$
2. $l = 1$
3. **while** $C_l \neq \emptyset$ **do**
4. compute $\mathcal{F}_l = \{\alpha \in C_l : fr(\alpha, \mathbf{s}, win) \geq min\_fr\}$
5. $l = l + 1$
6. compute $C_l = \{\alpha : |\alpha| = l$ and for all $\beta \prec \alpha$ such that $|\beta| < l$ we have $\beta \in \mathcal{F}_{|\beta|}\}$
7. **for** all l **do** output $\mathcal{F}_l$