KAOS project

University of Innsbruck

June 14, 2016

Andrea Burattin

Agenda

Topics covered (in isolation) relevant for the KAOS project

- Prediction of completion time of process instances
- Online mining of event streams

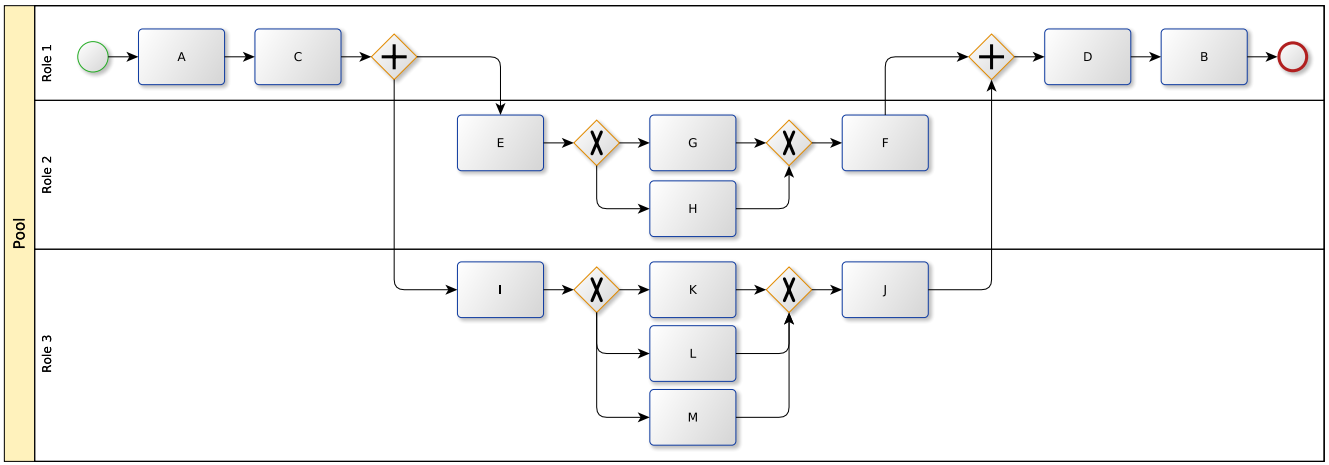
In KAOS, our goal is to "merge" these two aspects

Chapter 1

Prediction of completion time of process instances

Business Processes

- Set of tasks or activities
- Set of dependencies among activities
- The execution of an activity corresponding to a process task triggers an event logging on one or more information systems



Event log

	Basic data		Additional data		
Ev.#	**Activity**	**Time**	**Resource**	**Cost**	...
		Case Id: C_1			
1	*A*	2016-01-01	John	75	...
2	*B*	2016-01-02	John	5	...
3	*C*	2016-01-03	Joe	7	...
4	*E*	2016-01-04	John	1	...
		Case Id: C_2			
1	*A*	2016-01-02	Jack	65	...
2	*B*	2016-01-03	Joe	8	...
3	*D*	2016-01-04	John	7	...
4	*E*	2016-01-05	Jack	10	...

Remaining Time Prediction

Goal

Given a partial trace (i.e. running process instance), we want to predict the remaining time (i.e. time to complete).

Strong Business Motivations

- Avoid SLAs violations
- Take precautions if cases seem to take too much time
- Good waiting time estimation for customers

State of the art

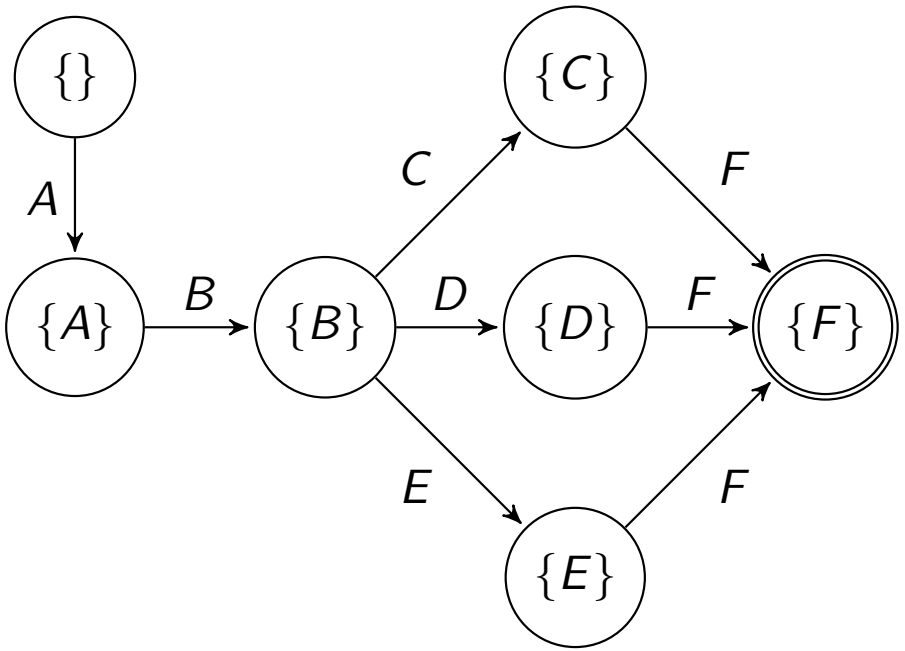
Time Prediction Based on Process Mining,
van der Aalst et al. (2011)

Steps:

1. Creation of a transition system
2. Transition system annotation with statistical information

$\langle A^0, B^4, C^{10}, F^{15} \rangle$
$\langle A^0, B^6, D^{13}, F^{15} \rangle$
$\langle A^0, B^7, E^9, F^{16} \rangle$
$\langle A^0, B^2, C^{15}, F^{19} \rangle$
$\langle A^0, B^{11}, D^{13}, F^{14} \rangle$
$\langle A^0, B^6, E^9, F^{12} \rangle$
...

$\longrightarrow$



State of the art (cont.)

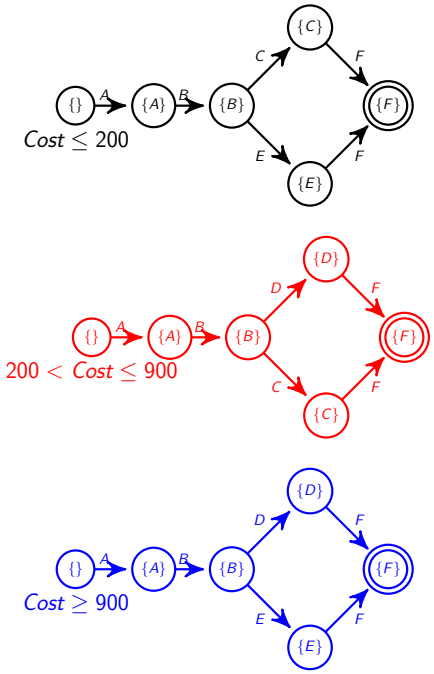
Discovering High-Level Performance Models for Ticket Resolution Processes, Folino et al. (2013)

Steps:

1. Traces clustering *(Clustering Tree)*
2. van der Aalst et al. approach on each cluster

$\langle A^0, B^4, C^{10}, F^{15} \rangle, Cost = 100$
$\langle A^0, B^6, D^{13}, F^{15} \rangle, Cost = 500$
$\langle A^0, B^7, E^9, F^{16} \rangle, Cost = 600$
$\langle A^0, B^2, C^{15}, F^{19} \rangle, Cost = 1000$
$\langle A^0, B^{11}, D^{13}, F^{14} \rangle, Cost = 200$
$\langle A^0, B^6, E^9, F^{12} \rangle, Cost = 900$
...

$\longrightarrow$



Prediction framework overview

Idea

Using additional data attributes in the event log in order to improve the prediction quality:

- Estimate the likelihood of the next activity
- Estimate the remaining time

- Exploit the idea in "*Time prediction based on Process Mining*" (2011)
- Consider additional data attributes
- Application of Machine Learning techniques
  - Naïve Bayes (NB) classifier
  - Support Vector Regressor (SVR)

Transition System (TS)

Id	Case Id	Timestamp	Activity	Resource	Cost	...
e_5	2	19-02-2016:09.10	A	Jack	200	
e_6	2	19-02-2016:13.22	B	John	200	
e_7	2	10-02-2016:17.17	D	John	200	
e_8	2	21-02-2016:10.38	F	Joe	200	

Transition System (TS)

Id	Case Id	Timestamp	Activity	Resource	Cost	...
e_5	2	19-02-2016:09.10	A	Jack	200	
e_6	2	19-02-2016:13.22	B	John	200	
e_7	2	10-02-2016:17.17	D	John	200	
e_8	2	21-02-2016:10.38	F	Joe	200	

- Event representation function f^{event}. E.g., $f^{event}(e_5) = A$

Transition System (TS)

Id	Case Id	Timestamp	Activity	Resource	Cost	...
e_5	2	19-02-2016:09.10	A	Jack	200	
e_6	2	19-02-2016:13.22	B	John	200	
e_7	2	10-02-2016:17.17	D	John	200	
e_8	2	21-02-2016:10.38	F	Joe	200	

- Event representation function f^{event}. E.g., $f^{event}(e_5) = A$
- State representation function f^{state}. E.g :
  - $f^{state}_{set^1}(\langle A, B, D, F\rangle) = \{F\}$
  - $f^{state}_{set^\infty}(\langle A, A, B, D, F\rangle) = \{A, B, D, F\}$
  - $f^{state}_{list^\infty}(\langle A, B, D, F\rangle) = \langle A, B, D, F\rangle$

Transition System (TS)

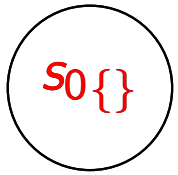
Id	Case Id	Timestamp	Activity	Resource	Cost	...
e_5	2	19-02-2016:09.10	A	Jack	200	
e_6	2	19-02-2016:13.22	B	John	200	
e_7	2	10-02-2016:17.17	D	John	200	
e_8	2	21-02-2016:10.38	F	Joe	200	

- Event representation function f^{event}. E.g., $f^{event}(e_5) = A$
- State representation function f^{state}. E.g :
  - $f^{state}_{set^1}(\langle A, B, D, F\rangle) = \{F\}$
  - $f^{state}_{set^\infty}(\langle A, A, B, D, F\rangle) = \{A, B, D, F\}$
  - $f^{state}_{list^\infty}(\langle A, B, D, F\rangle) = \langle A, B, D, F\rangle$

- f^{event} and f^{state} are *abstraction*

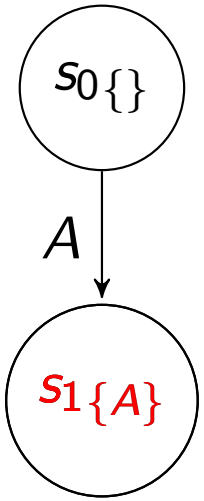
TS Construction

$$\sigma_1 = \langle A, B, C, F \rangle$$



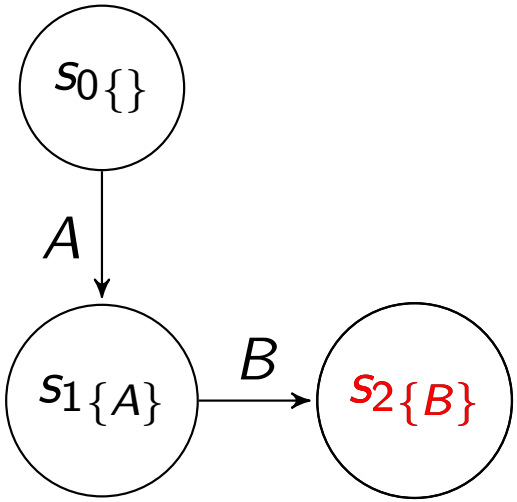
TS Construction

$$\sigma_1 = \langle \boldsymbol{A}, B, C, F \rangle$$



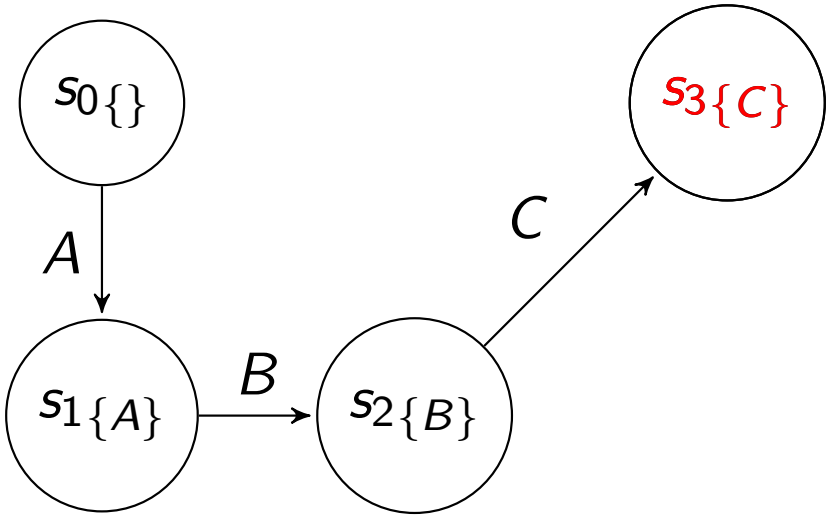
TS Construction

$$\sigma_1 = \langle \boldsymbol{A}, \boldsymbol{B}, C, F \rangle$$



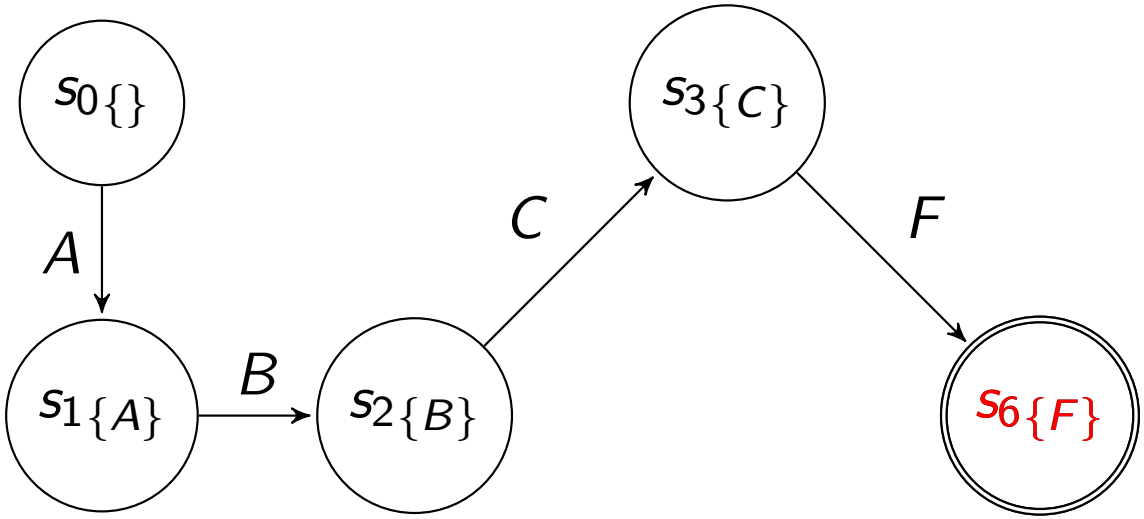
TS Construction

$$\sigma_1 = \langle \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, F \rangle$$



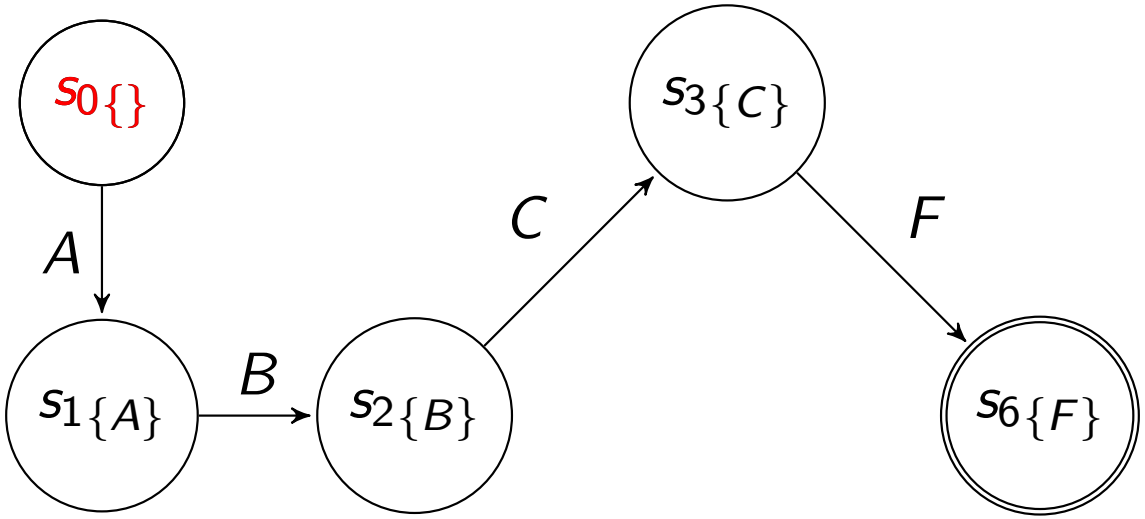
TS Construction

$$\sigma_1 = \langle \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{F} \rangle$$



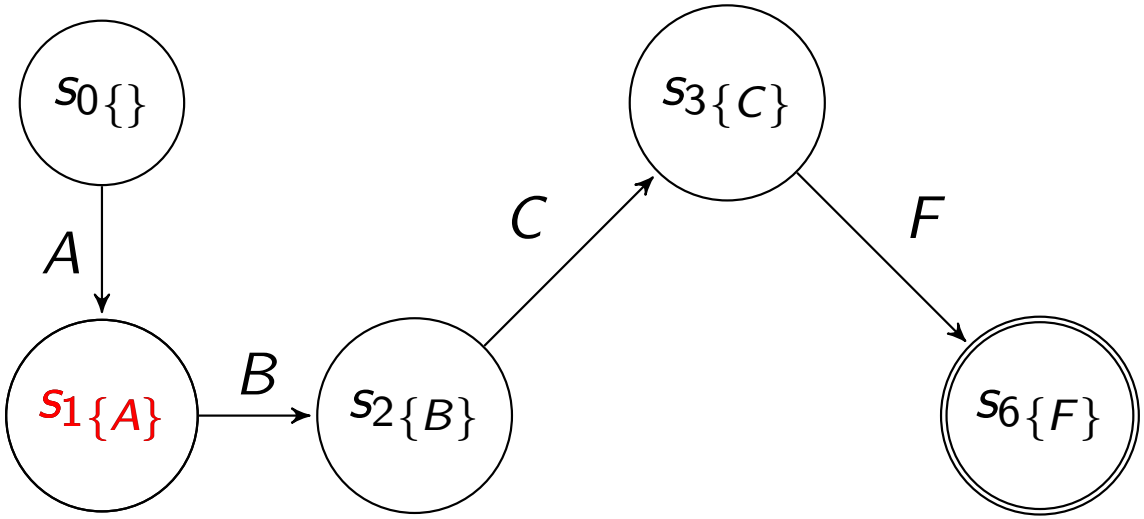
TS Construction

$$\sigma_2 = \langle A, B, D, F \rangle$$



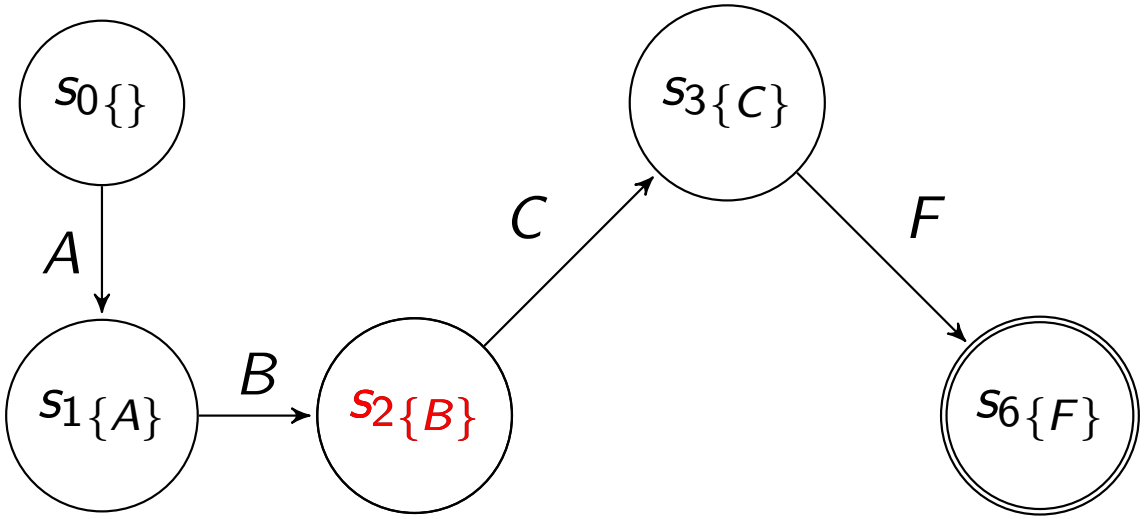
TS Construction

$$\sigma_2 = \langle \boldsymbol{A}, B, D, F \rangle$$



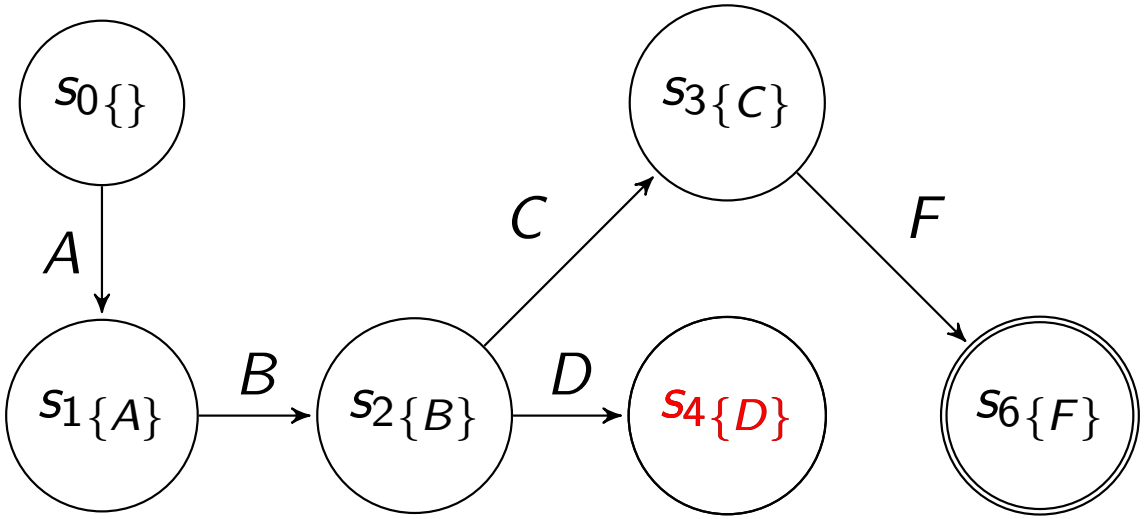
TS Construction

$$\sigma_2 = \langle \boldsymbol{A}, \boldsymbol{B}, D, F \rangle$$



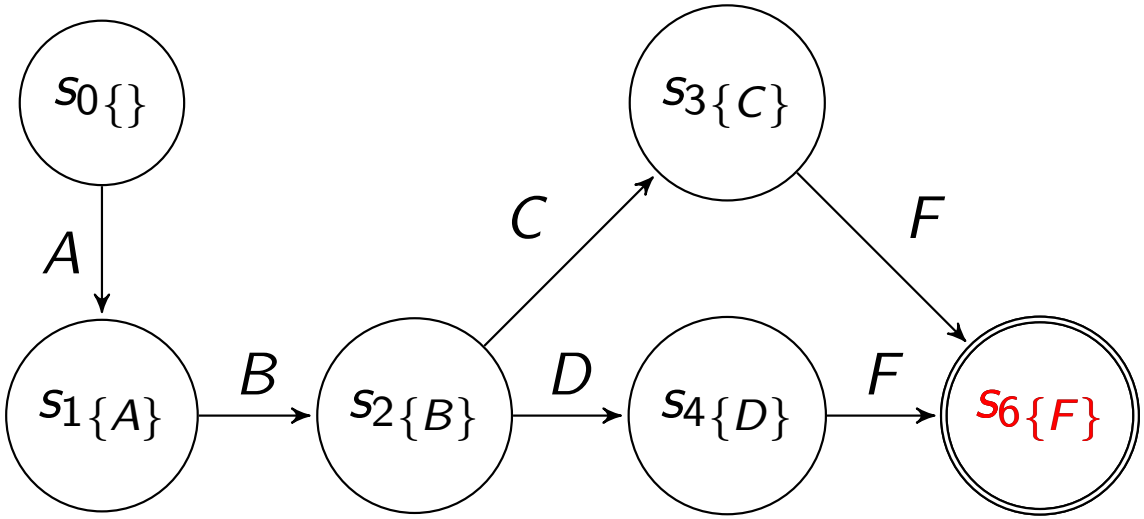
TS Construction

$$\sigma_2 = \langle \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{D}, F \rangle$$



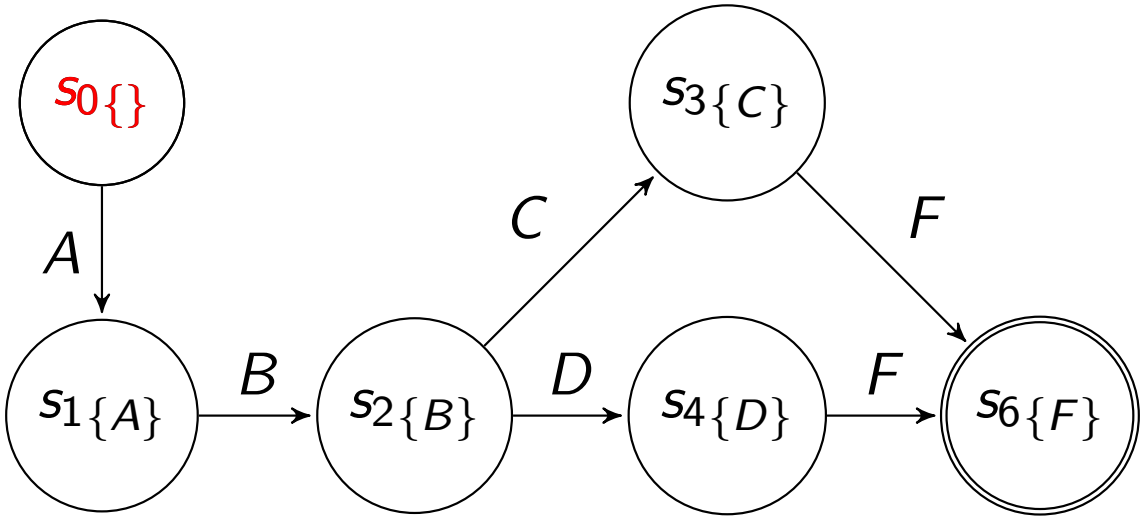
TS Construction

$$\sigma_2 = \langle \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{F} \rangle$$



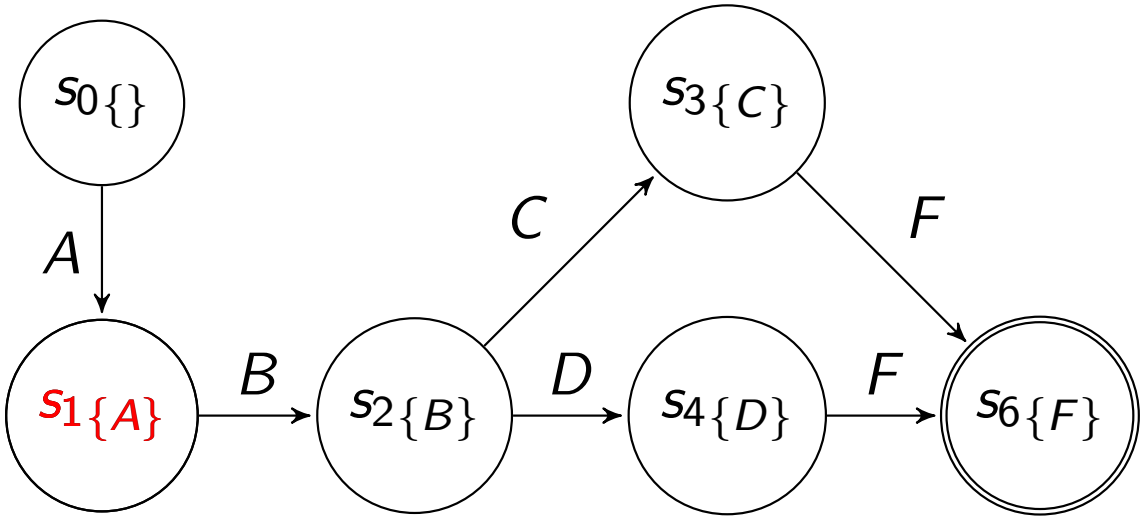
TS Construction

$$\sigma_3 = \langle A, B, E, F \rangle$$



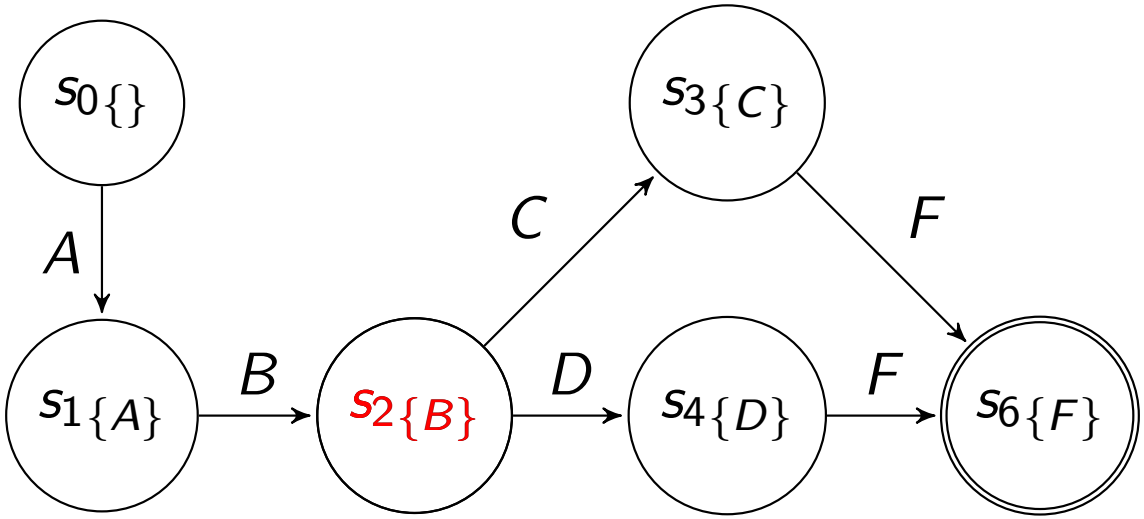
TS Construction

$$\sigma_3 = \langle \boldsymbol{A}, B, E, F \rangle$$



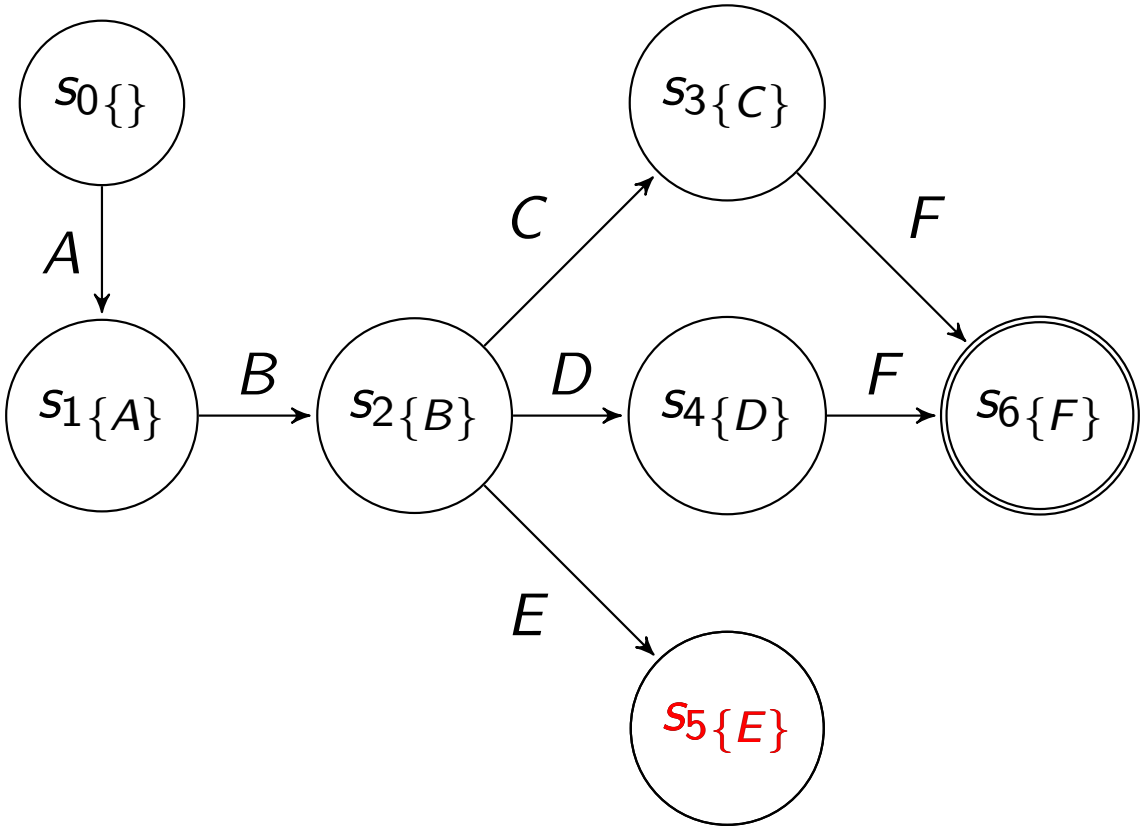
TS Construction

$$\sigma_3 = \langle \boldsymbol{A}, \boldsymbol{B}, E, F \rangle$$



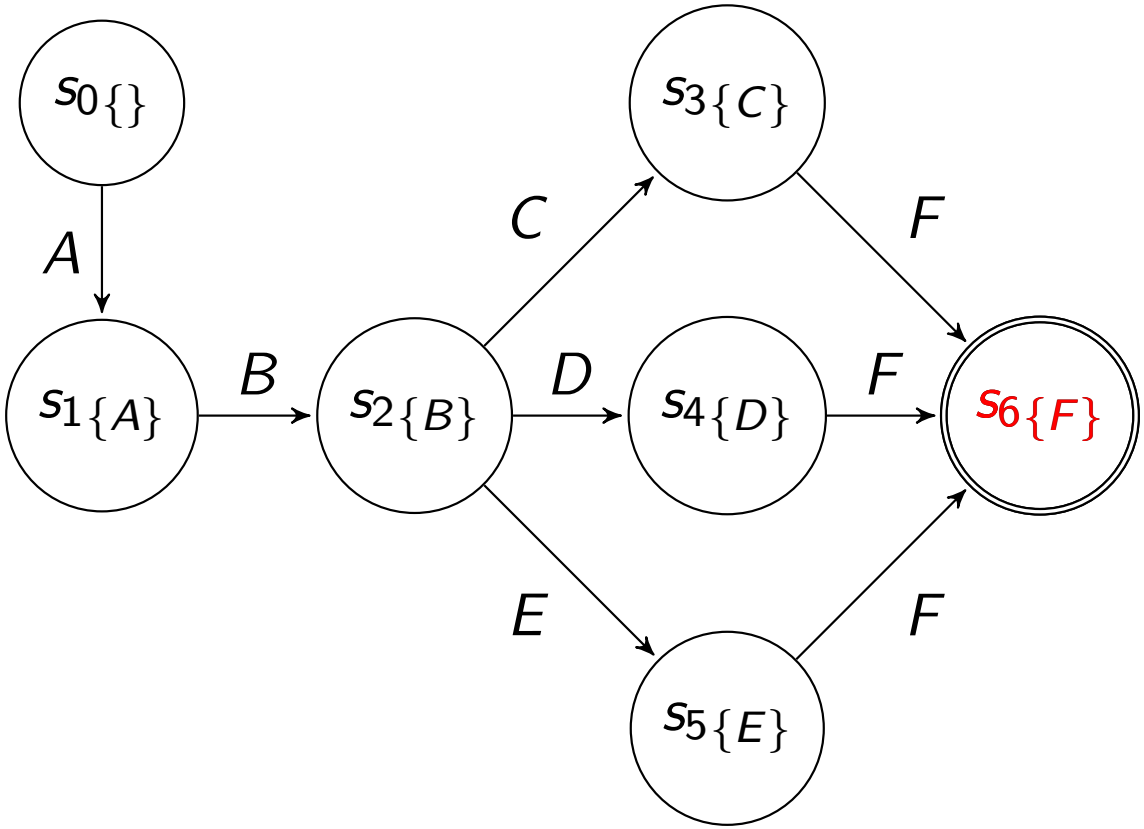
TS Construction

$$\sigma_3 = \langle \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{E}, F \rangle$$



TS Construction

$$\sigma_3 = \langle \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{E}, \boldsymbol{F} \rangle$$



Training examples

Id	Case Id	Timestamp	Activity	Resource	Cost	...
e_5	2	19-02-2016:09.10	A	Jack	200	...
e_6	2	19-02-2016:13.22	B	John	200	...
e_7	2	10-02-2016:17.17	D	John	200	...
e_8	2	21-02-2016:10.38	F	Joe	200	...

- One-Hot encoding for literal attributes, e.g.,
  $\mathrm{Res} \in \{Jack, John, Joe\}$; $Jack = [1, 0, 0]$; $John = [0, 1, 0]$, ...

- Naïve Bayes training example for state s_2

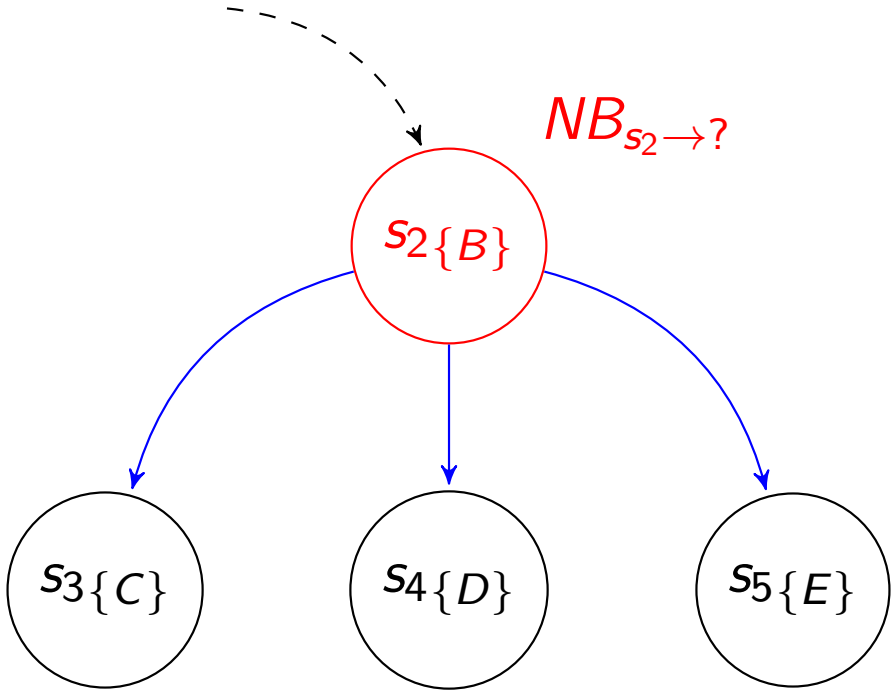
$e_7^{Activity}$	Resource	Cost	...
D	0 1 0	200	...

- SVR training example for transition $s_2 \rightarrow s_4$

$e_8^{time} - e_6^{time}$	Resource	Cost	...
162960	0 1 0	200	...

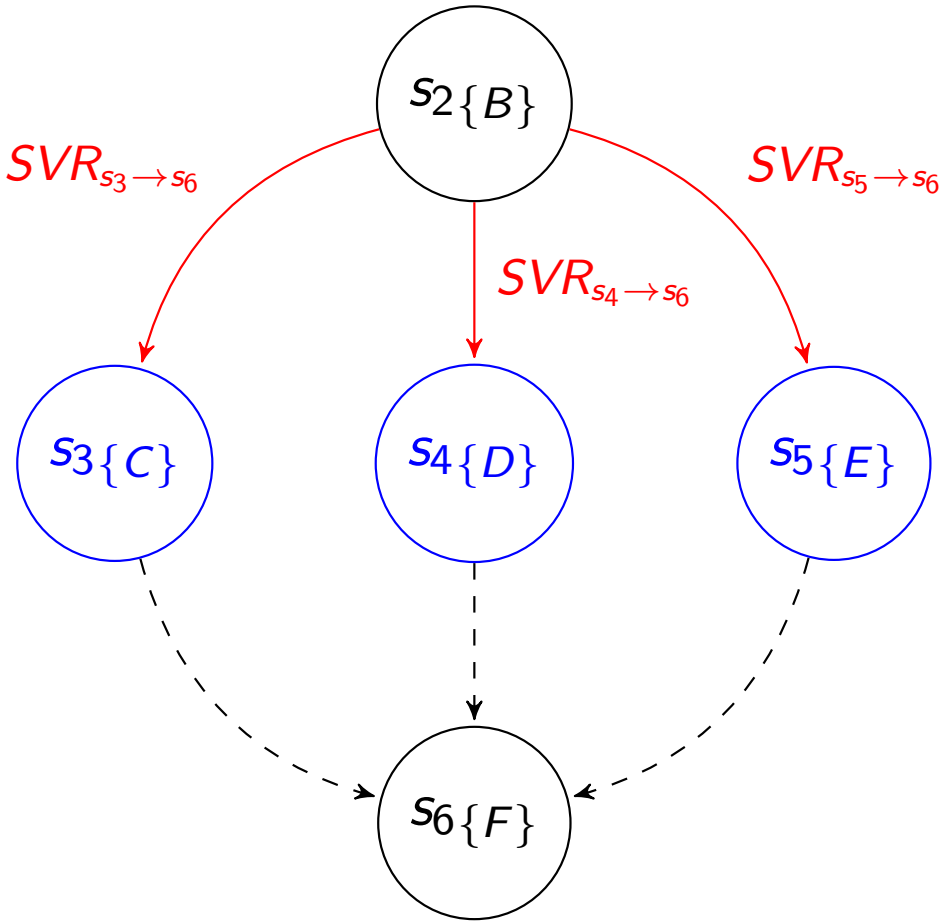
State Annotations

Naïve Bayes Classifier estimates the likelihood of the next activity



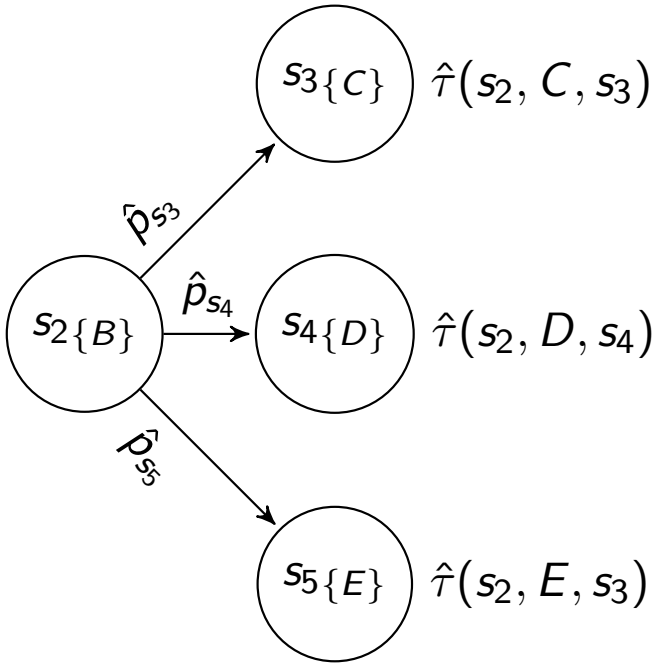
Transition Annotations

SVR estimates the remaining time starting from the target state of the transition



Prediction

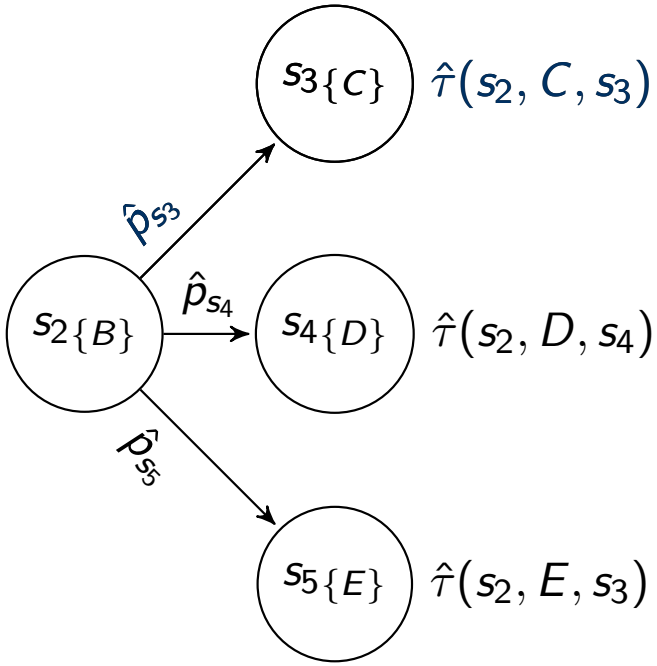
Given a partial trace σ s.t. $f^{state}(\sigma) = s_2$



$$prediction(\sigma) = \sum_{(s_2, e, s') \in T_{s_2}} \hat{p}_{s'} \cdot \hat{\tau}(s_2, e, s')$$

Prediction

Given a partial trace σ s.t. $f^{state}(\sigma) = s_2$

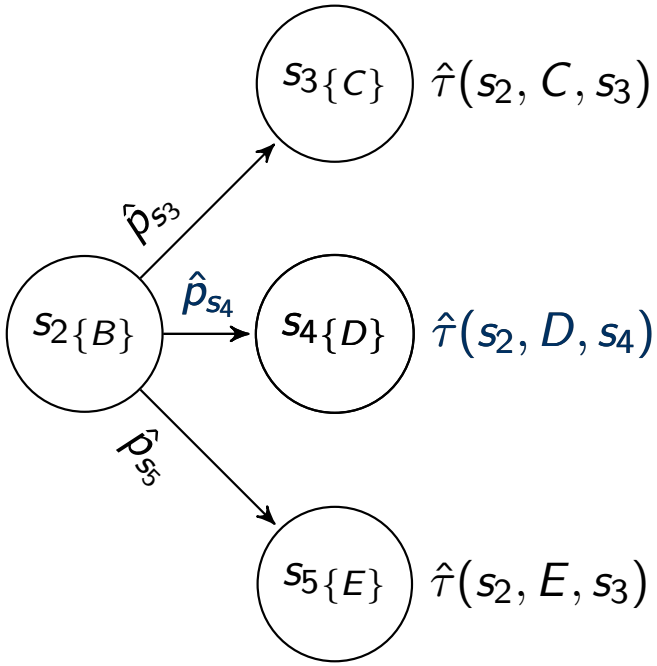


$$prediction(\sigma) = \sum_{(s_2,e,s') \in T_{s_2}} \hat{p}_{s'} \cdot \hat{\tau}(s_2, e, s')$$

$$= \hat{p}_{s_3} \cdot \hat{\tau}(s_2, C, s_3)$$

Prediction

Given a partial trace σ s.t. $f^{state}(\sigma) = s_2$

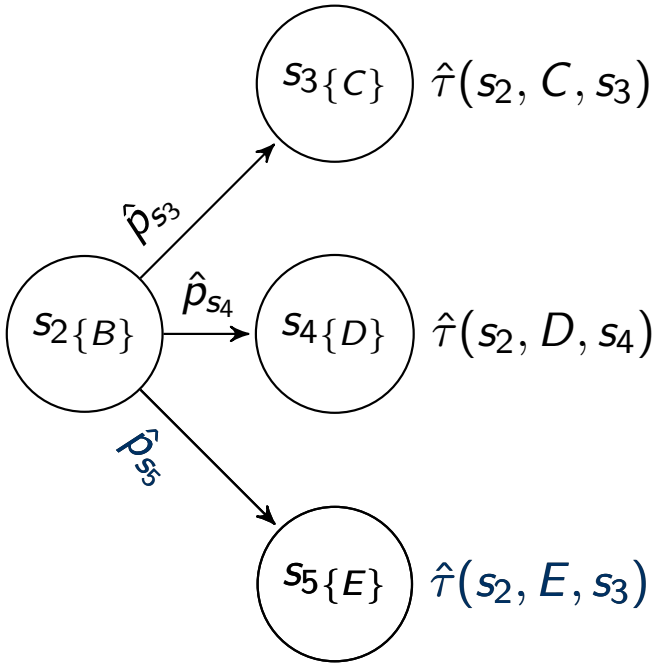


$$prediction(\sigma) = \sum_{(s_2,e,s') \in T_{s_2}} \hat{p}_{s'} \cdot \hat{\tau}(s_2, e, s')$$

$$= \hat{p}_{s_3} \cdot \hat{\tau}(s_2, C, s_3) + \hat{p}_{s_4} \cdot \hat{\tau}(s_2, D, s_4)$$

Prediction

Given a partial trace σ s.t. $f^{state}(\sigma) = s_2$



$$prediction(\sigma) = \sum_{(s_2,e,s') \in T_{s_2}} \hat{p}_{s'} \cdot \hat{\tau}(s_2, e, s')$$

$$= \hat{p}_{s_3} \cdot \hat{\tau}(s_2, C, s_3) + \hat{p}_{s_4} \cdot \hat{\tau}(s_2, D, s_4) + \hat{p}_{s_5} \cdot \hat{\tau}(s_2, E, s_D)$$

Experimental settings

- Implementation in ProM 6 (Java)
- Weka as machine learning module
- 5-cross fold validation
- Road-traffic fines log
- SVR ($C = 1$) kernels
  - RBF with $\gamma = 10$
  - Polynomial with degree $= 3$

How we measure error (A: actual; F: predicted):

- $\frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right|$ ..... (Mean Absolute Percentage Error)
- $100\% \sqrt{\frac{\sum_{i=1}^{n} (A_i - F_i)^2}{n}}$ ..... (Root Mean Square Prediction Error)

Results

	Our Approach				van der Aalst et al.	
	Polynomial Kernel		RBF Kernel			
Abs.	MAPE	RMSPE	MAPE	RMSPE	MAPE	RMSPE
Event Log with 1500 Traces						
Set 1	**8.93%**	**2.69%**	9.13%	2.73%	23.04%	3.53%
Set ∞	7.61%	1.96%	**7.60%**	**1.80%**	22.95%	3.48%
List ∞	7.68%	1.97%	**7.61%**	**1.80%**	22.90%	3.41%
Event Log with 5000 Traces						
Set 1	**5.78%**	**1.87%**	7.42%	2.14%	9.98%	2.33%
Set ∞	**5.07%**	**1.13%**	6.68%	1.37%	9.82%	2.37%
List ∞	**5.08%**	**1.13%**	6.69%	1.38%	9.13%	2.27%

Conclusions and future work

Conclusions

We present a new prediction approach which:

- Extracts a transition system from the event log
- Enriches the transition system with machine learning models (Naïve Bayes + SVR)
- Exploits additional data attributes
- Better predictions with a real-life case study

Future work

- Hyperparameters autotuning
- Simplify the usage for non-expert users
- Prediction with non-fitting traces

Chapter 2

Online mining of event streams

Typical Event Log

Event #	Activity	Orig.	Time	...
		Case Id: C_1		
1	A	U_1	2014-01-01	...
2	B	U_1	2014-01-02	...
3	C	U_2	2014-01-03	...
4	E	U_2	2014-01-04	...
		Case Id: C_2		
1	A	U_1	2014-01-02	...
2	B	U_1	2014-01-03	...
3	D	U_3	2014-01-04	...
4	E	U_3	2014-01-05	...

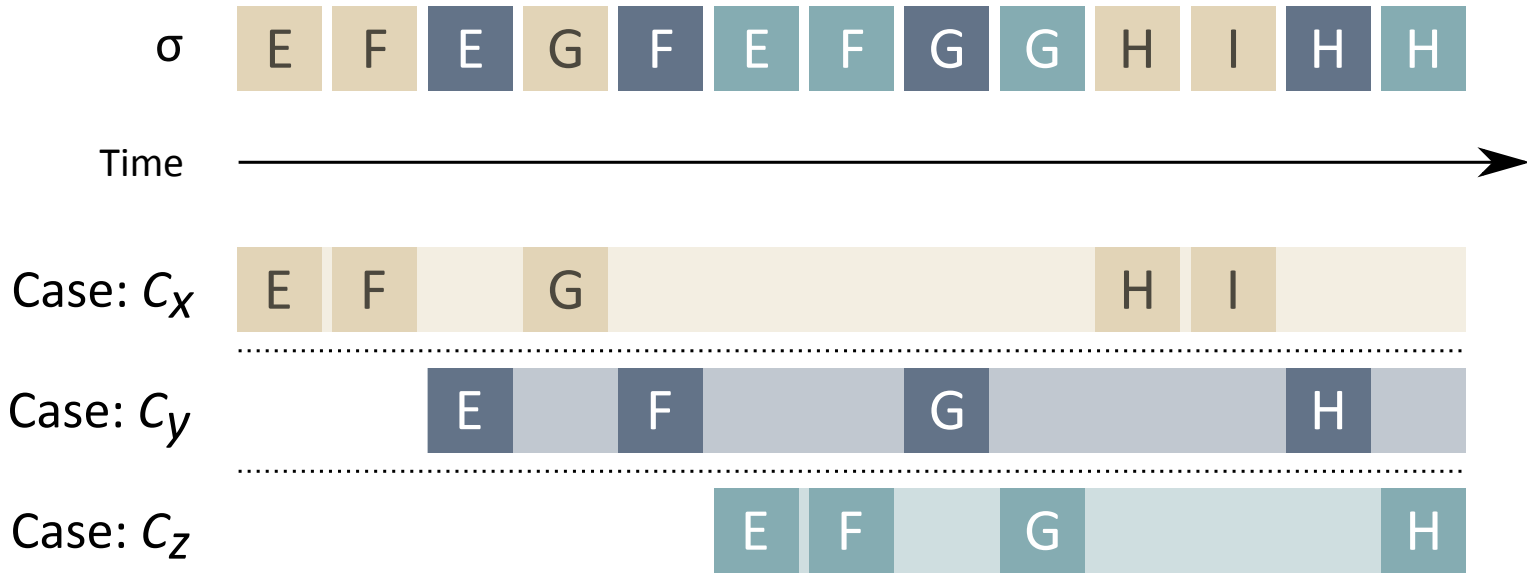
Typical Event Log

Event #	Activity	Orig.	Time	...
		Case Id: C_1		
1	A	U_1	2014-01-01	...
2	B	U_1	2014-01-02	...
3	C	U_2	2014-01-03	...
4	E	U_2	2014-01-04	...
		Case Id: C_2		
1	A	U_1	2014-01-02	...
2	B	U_1	2014-01-03	...
3	D	U_3	2014-01-04	...
4	E	U_3	2014-01-05	...

Event #	Case Id	Activity	Orig.	Time	...
1	C_1	A	U_1	2014-01-01	...
2	C_2	A	U_1	2014-01-02	...
3	C_1	B	U_1	2014-01-02	...
4	C_2	B	U_1	2014-01-03	...
5	C_1	C	U_2	2014-01-03	...
6	C_1	E	U_2	2014-01-04	...
7	C_2	D	U_3	2014-01-04	...
8	C_2	E	U_3	2014-01-05	...

Event Stream

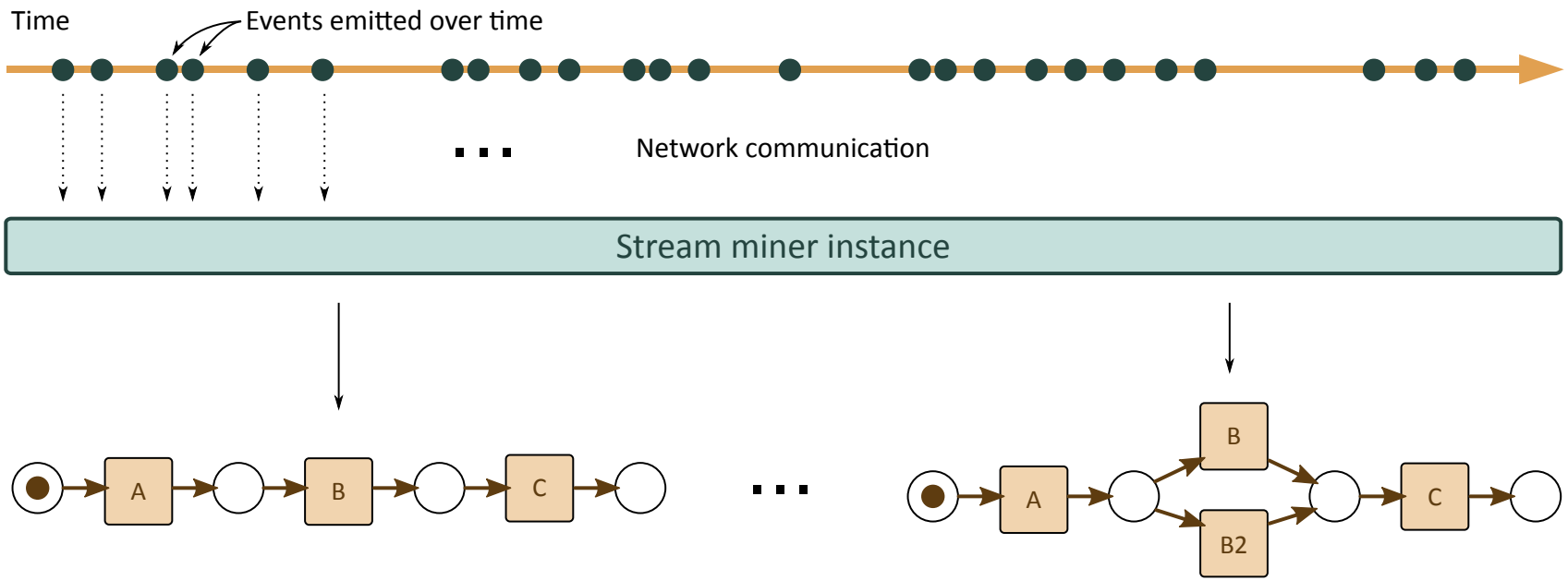
Representation of an event stream σ:



Boxes represent events

- Background colors represent the case id
- Letters inside are the activity names

Streaming Process Discovery



The stream miner continuously receives events and, using the latest observations, updates the process model (e.g., a Petri Net)

Stream Mining Peculiarities

Peculiarities of the stream mining problem

Stream Mining Peculiarities

Peculiarities of the stream mining problem

1. Cannot store the entire stream (approximation)

Stream Mining Peculiarities

Peculiarities of the stream mining problem

1. Cannot store the entire stream (approximation)
2. Backtracking not feasible over streams (algorithms required to make one pass over data $\rightarrow$ scale linearly w.r.t. the number of processed items)

Stream Mining Peculiarities

Peculiarities of the stream mining problem

1. Cannot store the entire stream (approximation)
2. Backtracking not feasible over streams (algorithms required to make one pass over data $\rightarrow$ scale linearly w.r.t. the number of processed items)
3. The approach must deal with variable system conditions, such as fluctuating stream rates

Stream Mining Peculiarities

Peculiarities of the stream mining problem

1. Cannot store the entire stream (approximation)
2. Backtracking not feasible over streams (algorithms required to make one pass over data $\rightarrow$ scale linearly w.r.t. the number of processed items)
3. The approach must deal with variable system conditions, such as fluctuating stream rates
4. It is important to quickly adapt the model to cope with unusual data values (concept drifts)

Heuristics Miner

Historical Background

Our approaches are based on Heuristics Miner, quite old ($\sim$ 2003) but still one of the most used algorithm

Heuristics Miner

Historical Background

Our approaches are based on Heuristics Miner, quite old ($\sim$ 2003) but still one of the most used algorithm

Fundamental metric is "*dependency measure*" between two activities (e.g. a, b):

$$a \Rightarrow b = \frac{|a > b| - |b > a|}{|a > b| + |b > a| + 1} \in [-1, 1]$$

Where:

- $|a > b|$ is the number of times that $a > b$ holds in the log
- $a > b$ holds if a executed at time t and b at $t + 1$

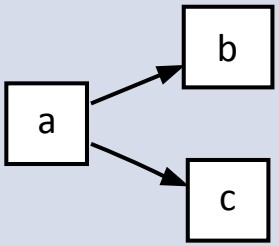
Heuristics Miner (cont.)

Given the dependency measure for all activity pairs and a threshold τ_{dep}, the algorithm builds a directed dependency graph

Heuristics Miner (cont.)

Given the dependency measure for all activity pairs and a threshold τ_{dep}, the algorithm builds a directed dependency graph

If both $a \Rightarrow b > \tau_{\text{dep}}$ and $a \Rightarrow c > \tau_{\text{dep}}$ then:



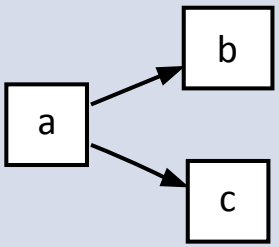
Relation ambiguity between b and c:

- XOR: either b or c is executed
- AND: both b and c are executed

Heuristics Miner (cont.)

Given the dependency measure for all activity pairs and a threshold τ_{dep}, the algorithm builds a directed dependency graph

If both $a \Rightarrow b > \tau_{\text{dep}}$ and $a \Rightarrow c > \tau_{\text{dep}}$ then:



Relation ambiguity between b and c:

- XOR: either b or c is executed
- AND: both b and c are executed

Heuristics Miner proposes the "AND-measure"

$$a \Rightarrow (b \wedge c) = \frac{|b > c| + |c > b|}{|a > b| + |a > c| + 1} \in [0, 1]$$

If $a \Rightarrow (b \wedge c) > \tau_{\text{and}}$ then AND relation, XOR otherwise

Direct Following Matrix

Basic data structure for HM is **Direct Following Matrix**

Direct Following Matrix

Given activities A, B, C, D and a log L, $|a > b|$ is the number of times that a is directly followed by b (within the same process instance) in the log L

	A	B	C	D
A	0	52	64	91
B	52	0	24	87
C	64	24	0	13
D	91	87	13	0

Proposed Approaches

Our Proposal

We present three approaches, based on Heuristics Miner, for process discovery from event streams:

- (SW) Heuristics Miner with Sliding Window *(as baseline)*
- (LC) Heuristics Miner with Lossy Counting
- (LCB) Heuristics Miner with Lossy Counting with Budget

Proposed Approaches

Our Proposal

We present three approaches, based on Heuristics Miner, for process discovery from event streams:

(SW) Heuristics Miner with Sliding Window *(as baseline)*

(LC) Heuristics Miner with Lossy Counting

(LCB) Heuristics Miner with Lossy Counting with Budget

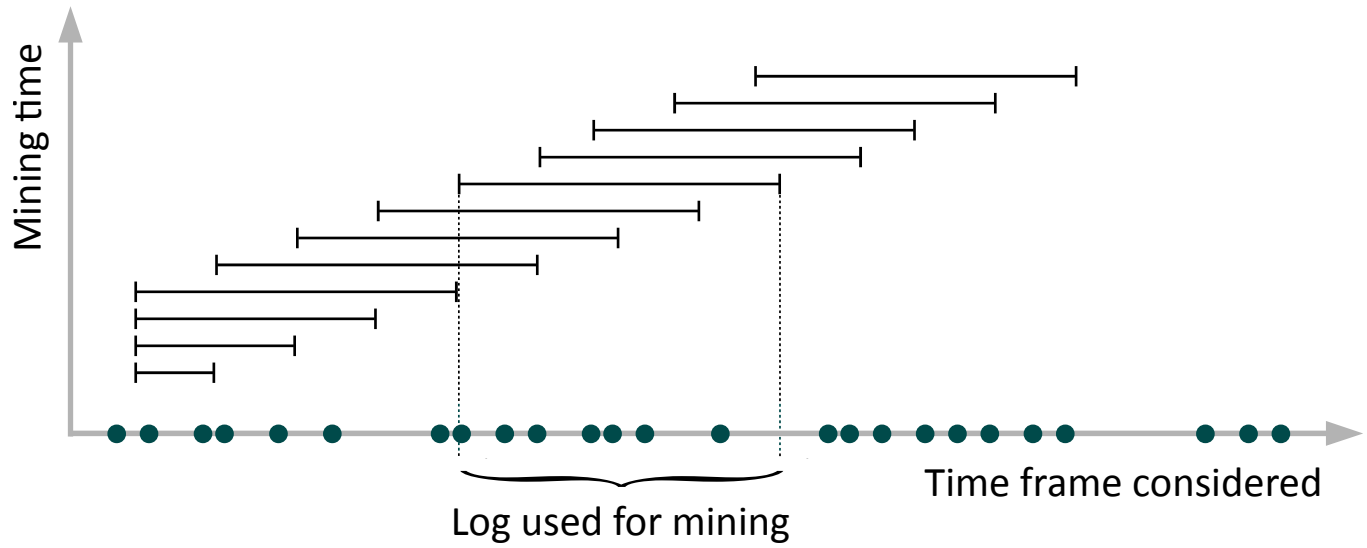
Fundamental Principle

Recent observations are more important than older ones

Heuristics Miner with SW

Basic idea is to iterate these steps

1. Collect events for a given time span
2. Generate a finite event log
3. Apply the "offline version" of the algorithm



Frequency Counting with Lossy Counting

Given

- Max approximation error ϵ
- Variables: A, B, C
- *Bucket* size is $w = \lceil \frac{1}{\epsilon} \rceil$
- $b_{current} = \lceil \frac{\text{no. of observed items}}{w} \rceil$

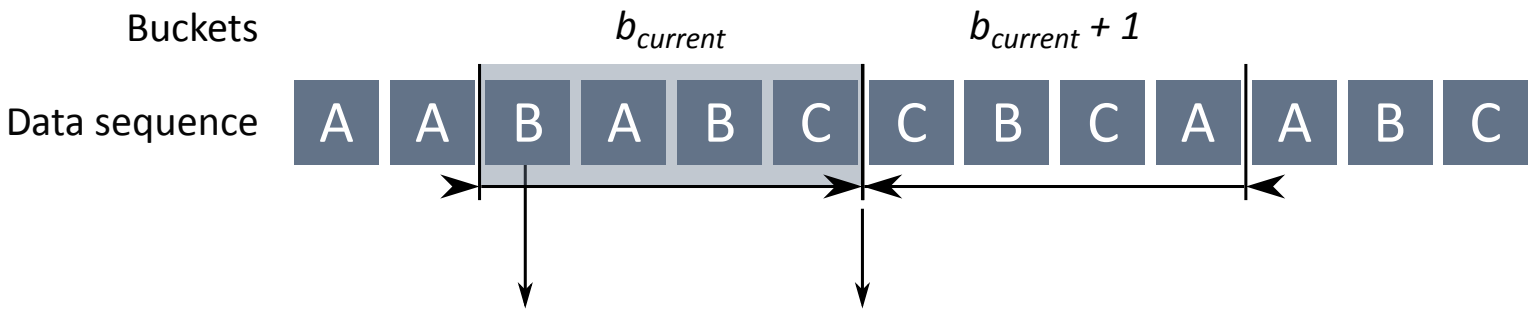
Lossy Counting uses a data structure $\mathcal{D} = \{(\text{var}, \text{freq}, \text{max error})\}$

Frequency Counting with Lossy Counting

Given

- Max approximation error ϵ
- Variables: A, B, C
- *Bucket* size is $w = \lceil \frac{1}{\epsilon} \rceil$
- $b_{current} = \lceil \frac{\text{no. of observed items}}{w} \rceil$

Lossy Counting uses a data structure $\mathcal{D} = \{(\text{var}, \text{freq}, \text{max error})\}$

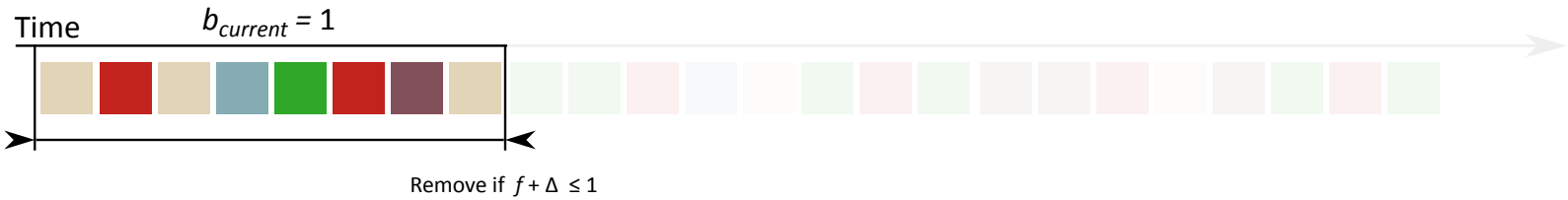


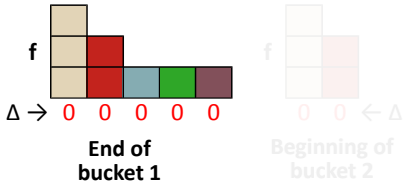
If B *not present* **then**
(B, f = 1, Δ = $b_{current}$ - 1)
else
Update frequency f of B

Remove all elements s.t. $f + \Delta \leq b_{current}$

LC/LCB Demo

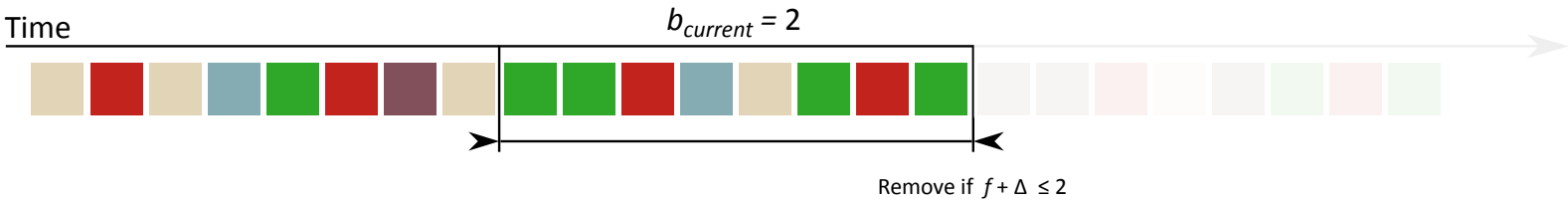
Lossy Counting Demo

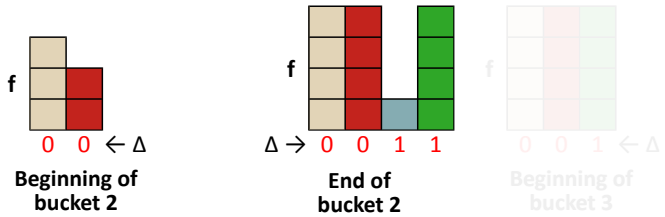




LC/LCB Demo

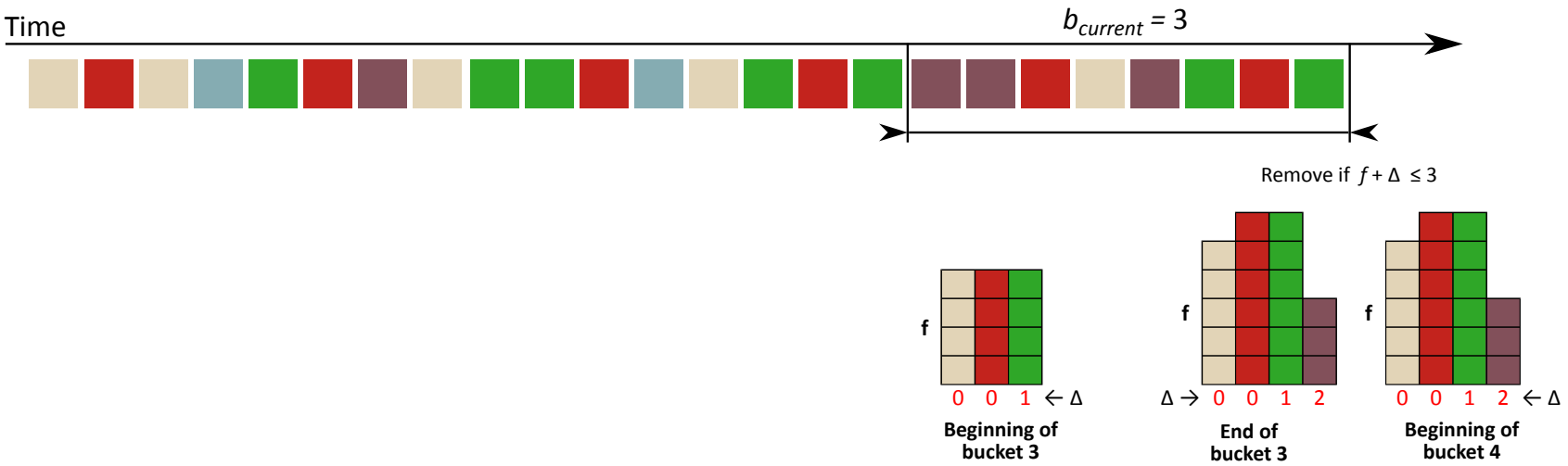
Lossy Counting Demo





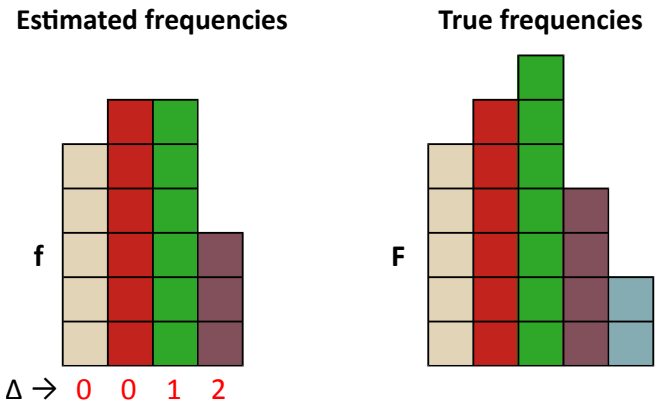
LC/LCB Demo

Lossy Counting Demo



LC/LCB Demo

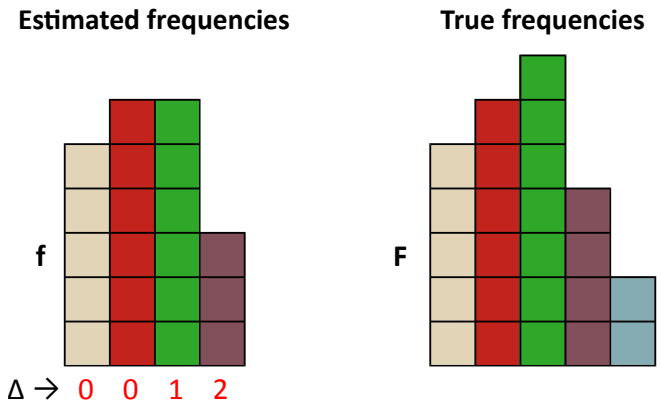
Comparison between Lossy Counting frequencies and true frequencies



These inequalities hold: $f \leq F \leq f + \Delta \leq f + \epsilon N$

LC/LCB Demo

Comparison between Lossy Counting frequencies and true frequencies



These inequalities hold: $f \leq F \leq f + \Delta \leq f + \epsilon N$

Lossy Counting with Budget Idea

New bucket when there is no more space, then $\epsilon = \frac{1}{\text{bucket size}}$

Adaptation of LC/LCB to HM

To count direct following relations we need

- $\mathcal{D}_{\text{rel}}$ Actual relations frequencies, tuples: (a_s, a_t, f, Δ)
- $\mathcal{D}_{\text{act}}$ Latest activity names, tuples: (a, f, Δ)
- $\mathcal{D}_{\text{cases}}$ Latest activity of a case, tuples: (c, a, f, Δ)

Adaptation of LC/LCB to HM

To count direct following relations we need

$\mathcal{D}_{\text{rel}}$ Actual relations frequencies, tuples: (a_s, a_t, f, Δ)

$\mathcal{D}_{\text{act}}$ Latest activity names, tuples: (a, f, Δ)

$\mathcal{D}_{\text{cases}}$ Latest activity of a case, tuples: (c, a, f, Δ)

With a certain periodicity, model update

- Activities from $\mathcal{D}_{\text{act}}$
- Dependencies and AND/XOR rules from $\mathcal{D}_{\text{rel}}$

Adaptation of LC/LCB to HM

To count direct following relations we need

$\mathcal{D}_{\text{rel}}$ Actual relations frequencies, tuples: (a_s, a_t, f, Δ)

$\mathcal{D}_{\text{act}}$ Latest activity names, tuples: (a, f, Δ)

$\mathcal{D}_{\text{cases}}$ Latest activity of a case, tuples: (c, a, f, Δ)

With a certain periodicity, model update

- Activities from $\mathcal{D}_{\text{act}}$
- Dependencies and AND/XOR rules from $\mathcal{D}_{\text{rel}}$

Actually, we show that updates on $\mathcal{D}$ data structures affect only local parts of the model (incremental update of the process model)

Evaluation Datasets

Artificial dataset characteristics:

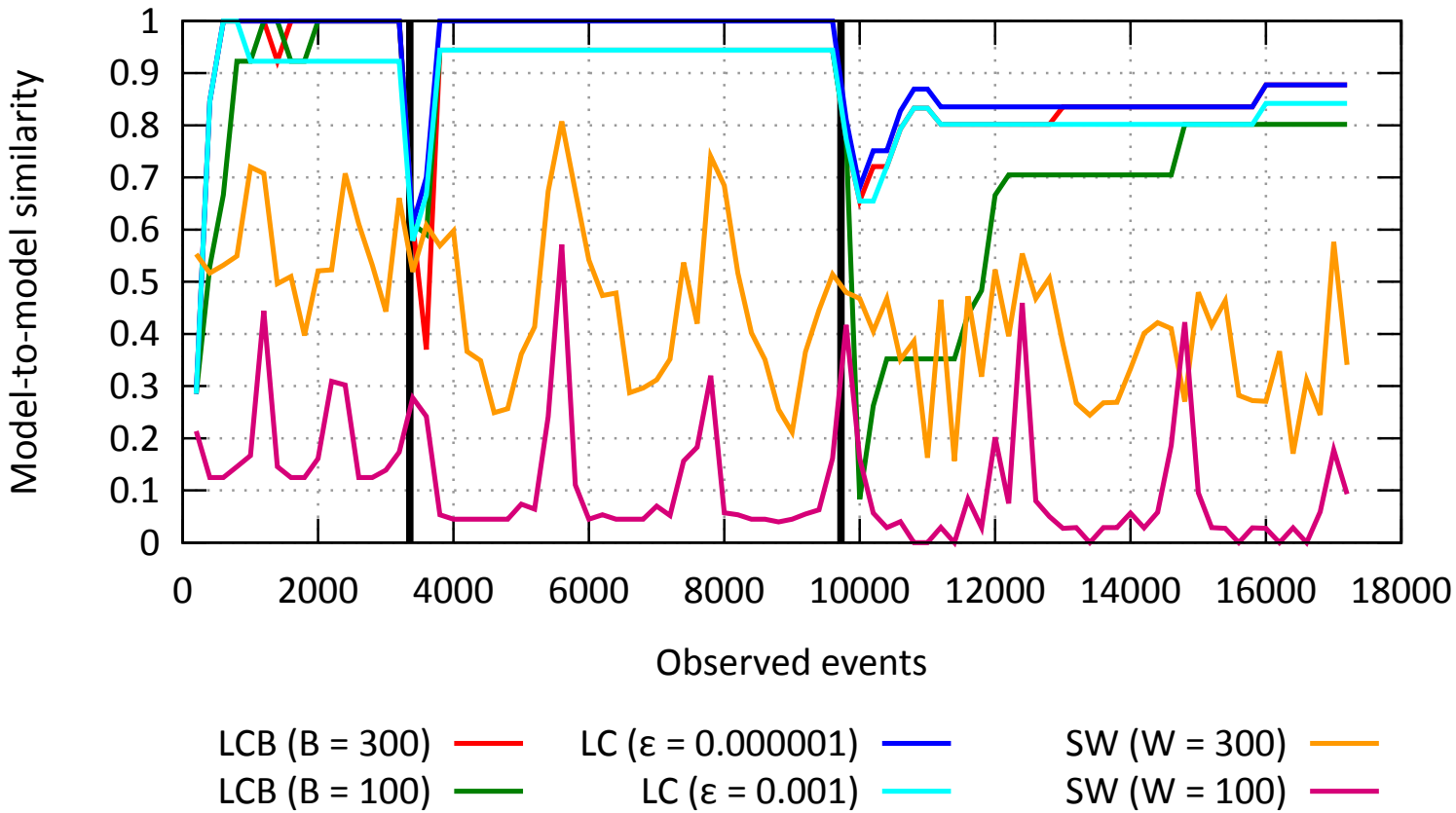
- Three randomly generated processes (to simulate concept drifts)
- Most complex model has 3 splits (1 AND and 2 XOR)
- Longest process has 16 activities
- Stream with 17 265 events

Real-world dataset (BPI Challenge 2012 log) characteristics

- Dutch Financial Institute
- 36 activities
- 262 198 events, among 13 087 process instances

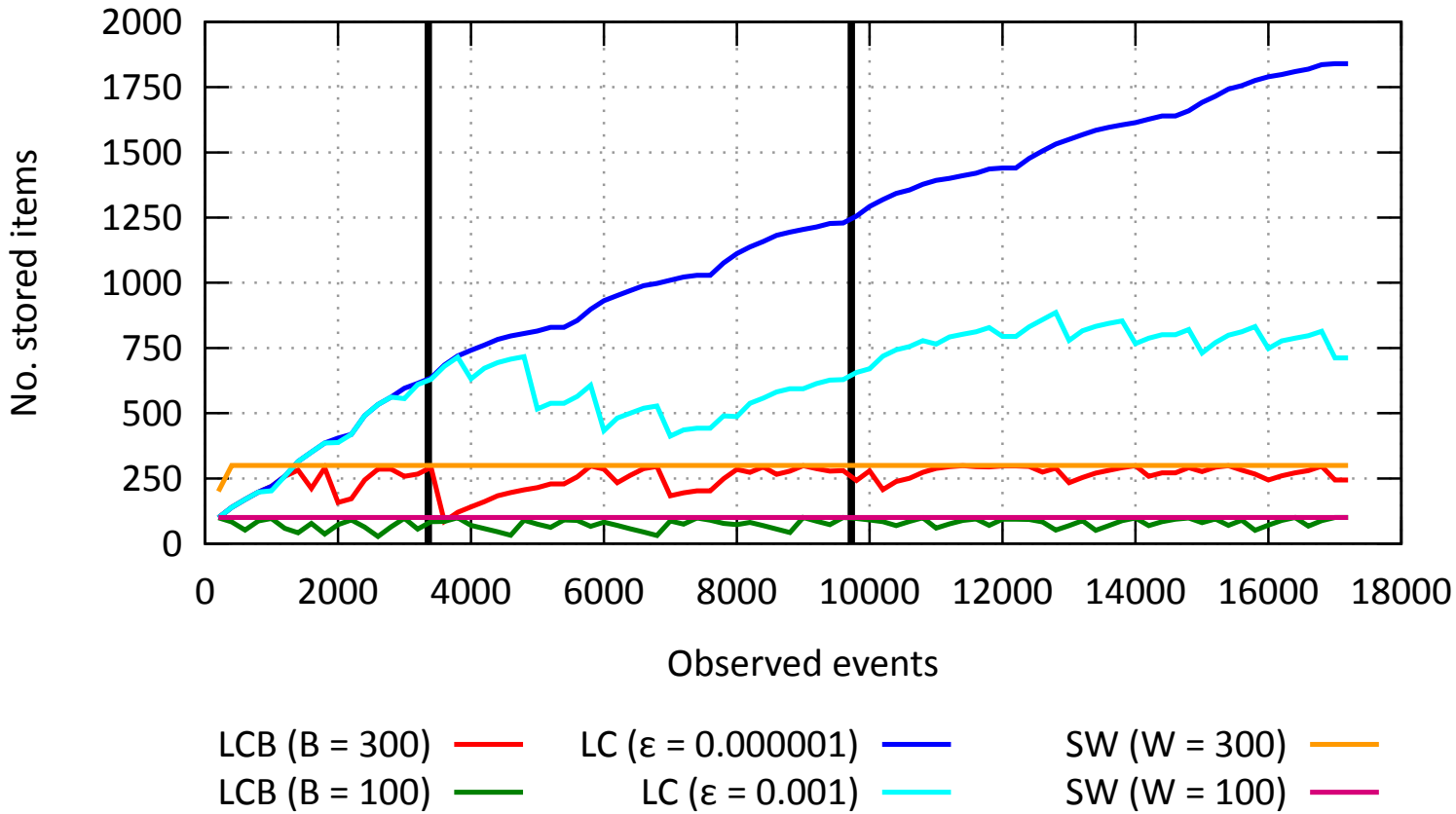
Evaluation on Artificial Dataset

Assess the correspondence between original and discovered model



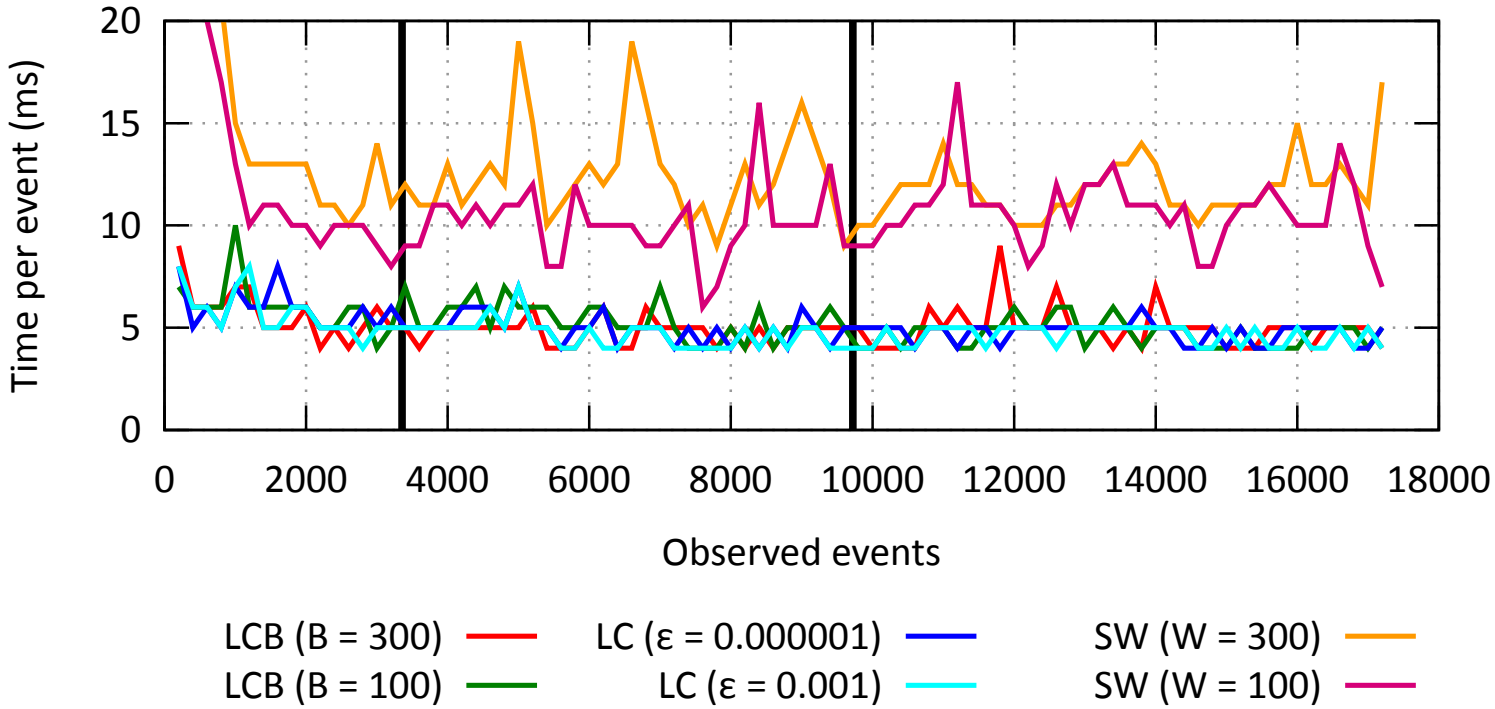
Evaluation on Artificial Dataset

Space expressed as number of stored items



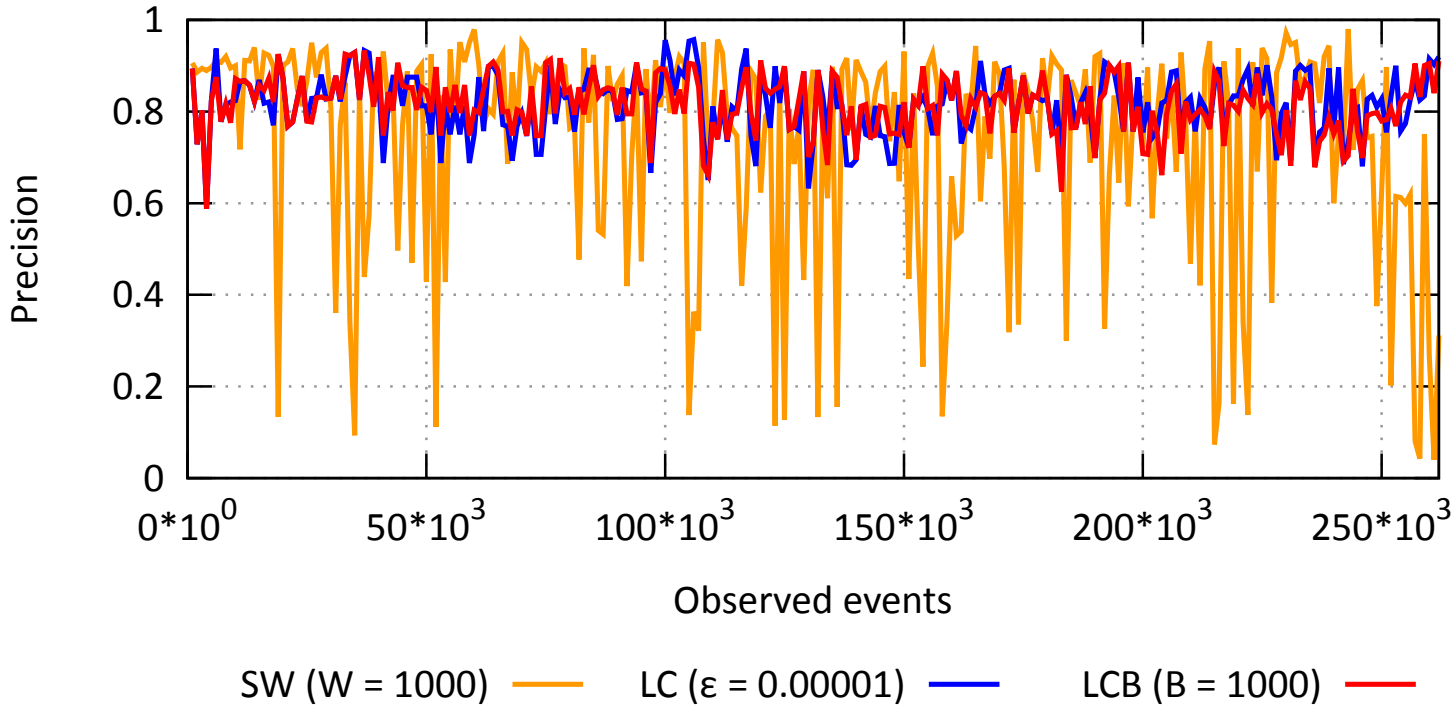
Evaluation on Artificial Dataset

Time required to process each event



Evaluation on BPI Challenge 2012

Precision of the discovered models



Time required to process an event:
SW: 24.59ms **LC**: 5.68ms **LCB**: 2.56ms

Conclusions and Future Work

Conclusions

- We discover process models from event streams
- Three approaches proposed, based on Heuristics Miner (with Sliding Window, with Lossy Counting, and with Lossy Counting with Budget)
- Experimental results on both artificial and real dataset, with improvements in terms of quality of the mined models, execution time, and space requirements as well

Future Work

- Improve the process analyst to mine different perspectives
- Animations to point out process drifts locations

With respect to KAOS...

We can model the prediction problem as a stream-based approach:

- Prediction model based on a-priori knowledge
- Prediction model could be online adapted to deal with concept drifts
- Predictions asked in real time