Ludwig-Maximilians-Universität München
Lehrstuhl für Datenbanksysteme und Data Mining
Prof. Dr. Thomas Seidl

Knowledge Discovery and Data Mining I

Winter Semester 2018/19

Agenda

1. Introduction

2. Basics

3. Unsupervised Methods

4. Supervised Methods

5. Advanced Topics
   5.1 Process Mining
   5.2 Outlook

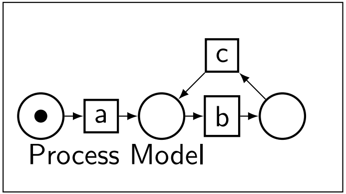
Motivation

supports/controls

configures

log events

model analyses



discovery

conformance

enhancement

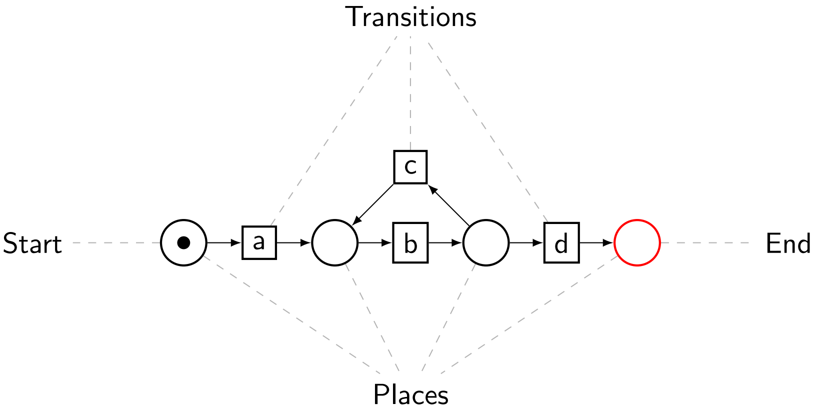
ID	Time	Location	Piece
12	14.01.2018 10:32	MUNICH	1
42	14.01.2018 11:40	MUNICH	2
5	14.01.2018 15:17	MUNICH	3
12	14.01.2018 22:12	LEIPZIG	1
...			

Notions

- Process: System of actions, movements (e.g. sign document, customer call, financial transaction, delivery of goods)
- Different instances/cases should follow a common process description
- Each case contains actions as events (their sequence is called *trace*)
- An event is represented by at least
  - A case identifier
  - An activity label
  - A timestamp

  but may also comprise additional (meta-)information (e.g. involved (work) resources)

Petri Nets as Process Model



Tasks

Main Tasks

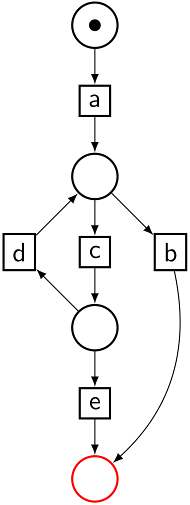
1. *Process Discovery*:
   Mine multiple sequences of actions to derive a workflow pattern
2. *Conformance Checking*:
   Use previously mined model to judge the validity of a new case
3. *Process Enhancement*:
   Evolve models with new data, find deviations

Process Discovery

Input	
#	trace
2048	ace
1234	acdce
404	acdcdce
120	acdcdcdce
42	ab
5	acdb

Quality Measures	
Fitness	ability to replay the log
Simplicity	simplified as much as possible
Generalization	no underfitting of log
Precision	no overfitting of log

Output



Example Discovery Algorithm: α-Miner[22]

1. Scan the log for all activities
2. For each pair of activities and , we define the relations
   - $a > b$ if for some case a is immediately followed by b (direct succession)
   - $a \parallel b$ if $a > b$ and $b > a$ (parallelism)
   - $a \rightarrow b$ if $a > b$ and not $b > a$ (causality)
   - $a \# b$ if not $a > b$ and not $b > a$
3. All activities, having only $\#$ or $\rightarrow$ in their row are starting activities. They are collected in T_{in}.
4. Analogously, $\#$ or $\leftarrow$ determine T_{out}.

Example: $\{abcd, acbd, acd\}$

	a	b	c	d
a		$\rightarrow$	$\rightarrow$	$\#$
b	$\leftarrow$		$\parallel$	$\rightarrow$
c	$\leftarrow$	$\parallel$		$\rightarrow$
d	$\#$	$\leftarrow$	$\leftarrow$	

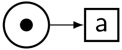
$T_{in} = \{a\}$, $T_{out} = \{d\}$

[22] van der Aalst, Weijters, Maruster (2003). "Workflow Mining: Discovering process models from event logs", IEEE Transactions on Knowledge and Data Engineering, vol 16

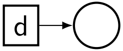
Example Discovery Algorithm: α-Miner

1. Prepare a Petri net. The set of transitions is equal to activities
2. A starting place is created and connected to each node in T_{in}
3. Also, a final place is created and each node in T_{out} is connected to it
4. Determine all pairs of sets A and B, such that
    - $\forall a_1, a_2 \in A : a_1 \# a_2$
    - $\forall b_1, b_2 \in B : b_1 \# b_2$
    - $\forall a \in A, b \in B : a \rightarrow b$
5. A place is added in between A and B and connected accordingly

2.

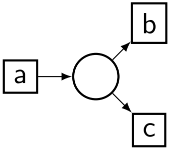


3.



4. $A = \{a\}, B = \{b, c\}$

5.



Conformance Checking

Use previously mined model to judge the validity of a new case (similar to binary classification: valid vs. invalid)

Input

- Model
- Trace

Aims

- Model reasoning
- auditing
- security (fraud detection)

Example Conformance Checking Algorithm: Token-Replay

Replay the event in the model. Count:

- the number of produced tokens (p)
- the number of consumed tokens (c)
- the number of missing tokens (m)
- the number of remaining tokens (r)

Output a *fitness* value

$$f = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right)$$

The fitness value ranges between 0 and 1, where 1 is a perfect match.