Process Mining I
Process Discovery

prof.dr.ir. Wil van der Aalst
www.processmining.org
Positioning Process Mining

Process mining
(data mining, machine learning, business intelligence)

Process model analysis
(simulation, verification, etc.)

Performance-oriented questions,
problems and solutions

data-oriented analysis
(data mining, machine learning, business intelligence)

Compliance-oriented questions,
problems and solutions
Big Data
“Enterprises globally stored more than 7 exabytes of new data on disk drives in 2010, while consumers stored more than 6 exabytes of new data on devices such as PCs and notebooks.”

“All of the world's music can be stored on a $600 disk drive.”

“Indeed, we are generating so much data today that it is physically impossible to store it all. Health care providers, for instance, discard 90 percent of the data that they generate.”


THE WORLD’S CAPACITY TO STORE INFORMATION
This chart shows the world’s growth in storage capacity for both analog data (books, newspapers, videotapes, etc.) and digital (CDs, DVDs, computer hard drives, smartphone drives, etc.)

In gigabytes or estimated equivalent

1986
ANALOG
2.62 billion

ANALOG STORAGE

DIGITAL
0.02 billion

2000

2007

ANALOG

18.86 billion gigabytes

Paper, film, audiotape and vinyl: 6.2%

Analogue videotapes: 93.8%

Other digital media: 0.8%*

Portable media players, flash drives: 2%

Portable hard disks: 2.4%

CDs and minidisks: 6.8%

Computer servers and mainframe hard disks: 8.9%

Digital tape: 11.8%

DVD/Blu-ray: 22.8%

PC hard disks: 44.5%

276.12 billion gigabytes

*Other includes chip cards, memory cards, floppy disks, mobile phones/PDAs, cameras/camcorders, video games

COMPUTING POWER
In 1986, pocket calculators accounted for much of the world’s data-processing power.

Percentage of available processing power by device:

<table>
<thead>
<tr>
<th>Year</th>
<th>Pocket calculators</th>
<th>Personal computers</th>
<th>Video game consoles</th>
<th>Servers, mainframes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1986</td>
<td>41%</td>
<td>33%</td>
<td>9%</td>
<td>17%</td>
</tr>
<tr>
<td>2007</td>
<td>66%</td>
<td>25%</td>
<td>3.6%</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

Mobile phones, PDAs

Supercomputers

2007
Process Discovery
Overview

"world" including people, machines, components, organizations, business processes

models analyzes

software system records events, e.g., messages, transactions, etc.

specifies configures implements analyzes

(discovery conformance enhancement)

(event logs)

(process) model

supports/controls

(event logs)

specifies configures implements analyzes

(discovery conformance enhancement)
Process Discovery (small selection)

- automata-based learning
- heuristic mining
- genetic mining
- stochastic task graphs
- fuzzy mining
- mining block structures
  - \(\alpha\) algorithm
  - \(\alpha\#\) algorithm
  - \(\alpha++\) algorithm
- distributed genetic mining
- language-based regions
- state-based regions
- LTL mining
- neural networks
- hidden Markov models
- conformal process graph
- partial-order based mining
- ILP mining
Petri net view: Just discover the places ...

Adding a place limits behavior:
- overfitting ≈ adding too many places
- underfitting ≈ adding too few places

\[ A = \{a_1, a_2, \ldots, a_m\} \]
\[ B = \{b_1, b_2, \ldots, b_n\} \]
The Petri net below can replay any trace over \{a,b,c,d,e\}

\[
L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]
\]
Place limits behavior

1. abcd
2. aed
3. acbd
4. abed
5. abcd
6. acbd
Example: Process Discovery Using State-Based Regions

Event log

Diagram showing state-based regions and transitions between events.
Example of State-Based Region

- enter: b,e
- leave: d
- do-not-cross: a,c
A place is **feasible** if it can be added without disabling any of the traces in the event log.

For any $\sigma \in L$, $k \in \{1, \ldots, |\sigma|\}$, $\sigma_1 = h d^{k-1}(\sigma)$, $a = \sigma(k)$, $\sigma_2 = h d^k(\sigma) = \sigma_1 \oplus a$:

$$c + \sum_{t \in X} \partial_{\text{multiset}}(\sigma_1)(t) - \sum_{t \in Y} \partial_{\text{multiset}}(\sigma_2)(t) \geq 0.$$
Example of Language-Based Regions

1. accd
2. bd
3. bce
4. ace
5. acd
6. bcce
7. ade

\[
\begin{align*}
\text{accd} & : 0 + 0 - 0 \geq 0 \\
ac & \downarrow \text{ccd} : 0 + 1 - 1 \geq 0 \\
\text{ace} & \downarrow \text{cd} : 0 + 2 - 2 \geq 0 \\
\text{acd} & \downarrow \text{d} : 0 + 3 - 3 \geq 0 \\
\text{ade} & : 0 + 0 - 0 \geq 0 \\
a & \downarrow \text{de} : 0 + 1 - 1 \geq 0 \\
ad & \downarrow \text{e} : 0 + 1 - 2 < 0
\end{align*}
\]
Creating a Transition System
Learning a Transition System

- past, future, past+future
- sequence, multiset, set abstraction
- limited horizon to abstract further
- filtering e.g. based on transaction type, names, etc.
- labels based on activity name or other features

trace: a b c d c d c d e f a g h h h i
Sometimes called the "prefix automaton"

\[ L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle] \]
Future Without Abstraction

\[ L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle] \]
Past with Multiset Abstraction

\[ L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle] \]
Only Last Event Matters For State

\[ L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle] \]
Only Next Event Matters For State

$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$
Only Last Event Matters

\[ L_2 = \left[ \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle \right] \]
Only Set of Last Two Events Matters

\[ L_2 = \left[ \langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^4, \langle a, b, c, e, f, b, c, d \rangle^2, \langle a, b, c, e, f, c, b, d \rangle, \langle a, c, b, e, f, b, c, d \rangle^2, \langle a, c, b, e, f, b, c, e, f, c, b, d \rangle \right] \]
Using ProM
$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$
Inspect Traces
Run Plugin
Select (scroll or by name)
Start Plugin "Mine Transition System"
Start Window

past

future

attributes
Abstraction

list, multiset, or set

all, or only last k events
Which events to filter?
Which labels need to be visible?
Any repair actions?

- Remove self loops
- Improve diamond structure
- Merge states with identical inflow
Resulting transition system
Convert transition system to Petri net
Resulting Petri net
\[ L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle] \]
State-Based Regions
What is a (state-based) region?

a = enter
b = enter
c = exit
d = exit
e = do not cross
f = do not cross
Starting point: A Transition System

- We assume that there is only one initial state (otherwise preprocessing needed).
- It is convenient to also have just one final state that can always be reached (not strictly necessary).
- All states need to be reachable!
Definition

• **A region** \( r \) is a set of states, such that for all transitions \((s_0, e, s_0')\), \((s_1, e, s_1')\) *in the transition system* holds that:
  1) \( s_0 \in r \) and \( s_0' \notin r \) implies that \( s_1 \in r \) and \( s_1' \notin r \)
  2) \( s_0 \notin r \) and \( s_0' \in r \) implies that \( s_1 \notin r \) and \( s_1' \in r \)

• In words: A region is a set of states, such that, if a transition *exits* the region, then all equally labeled transitions *exit* the region, and if a transition *enters* the region, then all equally labeled transitions *enter* the region. All events not entering or exiting the region *do not cross* the region.
Example of a region

- a enters
- b exits
- c does not cross
- d does not cross
- e does not cross
Example of a region

- a does not cross
- b enters
- c does not cross
- d does not cross
- e exits
Example of a region

- a exits
- b does not cross
- c does not cross
- d does not cross
- e does not cross

Places corresponding to regions containing the initial state are initially marked.
Example of a region

- a enters
- b does not cross
- c does not cross
- d exits
- e does not cross
- a enters
- b does not cross and exits
- c does not cross and exits
- d does not cross and exits
- e does not cross
Not a region

- a does not cross
- b does not cross and exits
- c does not cross and exits
- d does not cross and exits
- e does not cross
Multiple regions

Etc.
Selectively chosen regions ...
Another example

\{i,s0,s1\}  \{s2,s0,s1\}  \{s5,s0,s1\}  \{s6,s0,s1\}  \{i,s0,s3\}  \{s2,s0,s3\}  \{s5,s0,s3\}  \{i,s1,s4\}  \{s2,s1,s4\}  \{s5,s1,s4\}  \{s6,s1,s4\}  \{i,s3,s4\}  \{s2,s3,s4\}  \{s5,s3,s4\}  \{i,s5\}  \{s6\}  \\
\{s0,s1\}  \{s0,s3\}  \{s1,s4\}  \{s3,s4\}  \{i,s2\}  \{i,s5\}  \{i,s6\}  \{s2,s5\}  \{s2,s6\}  \{s5,s6\}

\{s2,i,s0,s1\}  \{s6,i,s0,s1\}  \{s6,s2,s0,s1\}  \{s2,i,s0,s3\}  \{s6,i,s0,s3\}  \{s6,s2,s0,s3\}  \{s2,i,s1,s4\}  \{s6,i,s1,s4\}  \{s6,s2,s1,s4\}  \{s2,i,s3,s4\}  \{s6,i,s3,s4\}  \{s6,s2,s3,s4\}  \\
\{s6,s5,i,s2\}  \{s6,i,s3,s4\}  \{s6,s2,s3,s4\}  \{s6,s5,i,s2\}  \{s6,s5,i,s3\}  \{s6,s5,i,s4\}  \\
\{s0,s1,s2,s3,s4\}  \{s0,s1,s3,s4\}  \{s0,s1,s3,s4,s5\}  \{s0,s1,s2,s3,s5\}  \{s0,s1,s2,s3,s6\}  \{i,s0,s1,s2,s3,s6\}  \\
\{i,s0,s1,s2,s3,s6\}  \{i,s0,s1,s3,s4,s6\}  \{i,s0,s1,s2,s3,s4,s6\}  \{i,s0,s1,s2,s3,s4,s5\}  \\
\{i,s0,s1,s2,s3,s4,s5\}  \{i,s0,s1,s2,s3,s4,s6\}  \\
\{i,s0,s1,s2,s3,s4,s5,s6\}
Regions – Region Properties

- Let S be the set of all states of a transition system.
- **Trivial Regions**: Both S and are called the trivial regions,
- **Complements**: If r is a region, then S\ r is a region,
- **Pre-/Post-regions**: If event e exits (enters) a region r, then r is a pre- (post-)region of e,
- **Minimal regions**: If r₀ and r₁ are regions, and r₀ is a subset of r₁, then r₁ \ r₀ is a region.
- The latter implies the existence of (non-trivial) minimal regions.
Trivial regions

a, b, c, d, e do not cross
Complement:
If $r$ is a region, then $S \setminus r$ is a region

"exits" and "enters" are swapped
If $r_0$ and $r_1$ are regions, and $r_0$ is a subset of $r_1$, then $r_1 \setminus r_0$ is a region.
Not minimal yet …
Pre and post regions

- If event $e$ enters a region $r$, then $r$ is a post-region of $e$.
  - $r$ is post-region of $a$
  - $r$ is post-region of $b$

- If event $e$ exits a region $r$, then $r$ is a pre-region of $e$.
  - $r$ is pre-region of $c$
  - $r$ is pre-region of $d$

• pre($e$) is the set of all (minimal) pre-regions or $e$.
• Both are sets of sets!
Pre/Post-regions (of event $b$)

$\{s_0, s_1\}$  $\{s_0, s_3\}$  $\{s_1, s_4\}$
$\{s_2, s_3, s_4\}$  $\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$

$\{s_0, s_1\}$  $\{s_0, s_3\}$
$\{s_1, s_4\}$  $\{s_2, s_3, s_4\}$
$\{s_5, s_6, s_7\}$
Minimal Regions

\[
\begin{align*}
&\{i, s0, s1\} \\
&\{s2, s0, s1\} \\
&\{s5, s0, s1\} \\
&\{s6, s0, s1\} \\
&\{i, s0, s3\} \\
&\{s2, s0, s3\} \\
&\{s5, s0, s3\} \\
&\{s6, s0, s3\} \\
&\{i, s1, s4\} \\
&\{s2, s1, s4\} \\
&\{s5, s1, s4\} \\
&\{s6, s1, s4\} \\
&\{i, s3, s4\} \\
&\{s2, s3, s4\} \\
&\{s5, s3, s4\} \\
&\{s6, s3, s4\} \\
&\{i, s2\} \\
&\{s2, s5\} \\
&\{s6\} \\
&\{s0, s1\} \\
&\{s0, s3\} \\
&\{s1, s4\} \\
&\{s3, s4\} \\
&\{i, s2\} \\
&\{i, s5\} \\
&\{i, s6\} \\
&\{s2, s5\} \\
&\{s2, s6\} \\
&\{s5, s6\}
\end{align*}
\]

\[
\begin{align*}
&\{s2, i, s0, s1\} \\
&\{s6, i, s0, s1\} \\
&\{s6, s2, s0, s1\} \\
&\{s2, i, s0, s3\} \\
&\{s6, i, s0, s3\} \\
&\{s6, s2, s0, s3\} \\
&\{s2, i, s1, s4\} \\
&\{s6, i, s1, s4\} \\
&\{s6, s2, s1, s4\} \\
&\{s2, i, s3, s4\} \\
&\{s6, i, s3, s4\} \\
&\{s6, s2, s3, s4\} \\
&\{s6, s5, i, s2\} \\
&\{s6, s5, i, s3\} \\
&\{s6, s5, s2, s3\} \\
&\{s0, s1, s2, s3, s4\} \\
&\{s0, s1, s3, s4, s6\} \\
&\{s0, s2, s3, s5, s6\} \\
&\{i, s0, s1, s2, s5\} \\
&\{i, s0, s1, s3, s4, s5, s6\} \\
&\{i, s0, s1, s3, s4, s6\} \\
&\{i, s0, s1, s3, s5, s6\} \\
&\{i, s0, s1, s2, s5\} \\
&\{i, s0, s1, s2, s6\} \\
&\{i, s0, s1, s5, s6\} \\
&\{i, s0, s3, s4, s5, s6\} \\
&\{i, s0, s3, s5, s6\} \\
&\{i, s0, s6, s3, s4\} \\
&\{i, s3, s4, s5, s6\} \\
&\{i, s3, s5, s6\} \\
&\{i, s6, s3, s4\}
\end{align*}
\]
Basic algorithm to construct a Petri net

• For each event in the transition system, a transition is generated in the Petri net.
• Compute the minimal non-trivial regions.
• For each minimal non-trivial in the transition system, a place is generated in the Petri net.
• Add corresponding arcs (post-regions are output places and pre-regions are input places).
• A token is added to each place that corresponds to a region containing the initial state.

The resulting Petri net is called the minimal saturated net.
Construct Petri net

\[\{i\} \rightarrow a \rightarrow \{s0,s1\} \rightarrow e \rightarrow \{s2\} \rightarrow f \rightarrow \{s5\}\]

\[\{s0,s3\} \rightarrow c \rightarrow \{s1,s4\}\]

\[\{s1,s4\} \rightarrow d \rightarrow \{s6\}\]
Load Petri net with 10 parallel activities
Construct reachability graph
Reachability graph \((1+2^{10}+1 = 1026\ \text{states})\)
Apply state-based regions to fold state space
Discovered Petri net

- Petri net is rediscovered!
- Odd example, normally the transition system is constructed from an event log.
40.825 states, 221.618 transitions

26 transitions, 28 places, 1 token
But .....
Consider an event log containing just \(<a,a>\) traces

Prefix automaton

\[
\begin{align*}
\text{s1} & \xrightarrow{a} \text{s2} \\
\text{s2} & \xrightarrow{a} \text{s3}
\end{align*}
\]

Only trivial regions: \(\emptyset\) and \(\{s1,s2,s3\}\)

Petri net

Also allows for:
- a
- aaaa
- aaaaaaaaaa
Consider an event log containing traces \(<a,c>, <a,b,c>, <a,b,b,c>, <a,b,b,b,c>, \ldots\) 

transition system able to generate log

Regions:
- \(\{s1\}\) (a exits, b and c do not cross)
- \(\{s2\}\) (a enters, b does not cross, c exits)
- \(\{s3\}\) (a and b do not cross, c enters)

Petri net

Also allows for:
- bbac
- acbbbb
- babcb
Consider an event log containing traces \(<a,b>, <b>\).

Transition system able to generate log

Regions:
- \{s1,s2\} (a does not cross, b exits)
- \{s3,s4\} (a does not cross, b enters)
- \{s1,s3\} (a exits and b does not cross)
- \{s2,s4\} (a enters and b does not cross)
Petri net

Also allows for trace \(<b,a>!\)
All underfitting, but feasible
Using ProM
(uses label splitting to solve problem)

two "a" transitions
Using ProM
(addresses self-loop problem)
Using ProM
(uses label splitting to solve problem)
Some literature

A wannabe food critic will definitely not be able to make these!

Exercises: BPMS-instruction-10-process-mining-l.ppt/pdf