Knowledge Discovery and Data Mining 1
(Data Mining Algorithms 1)

Wintersemester 2019/20
Agenda

1. Introduction
2. Basics
3. Supervised Methods
4. Unsupervised Methods
5. Process Mining
   5.1 Introduction
   5.2 Process Models – An Overview
   5.3 Process Discovery
   5.4 Conformance Checking
   5.5 Additional Mining Tasks
   5.6 Streams
5. Process Mining
5.1 Introduction
    Motivation
    Getting the Data
5.2 Process Models – An Overview
5.3 Process Discovery
5.4 Conformance Checking
5.5 Additional Mining Tasks
5.6 Streams
5. Process Mining

5.1 Introduction
Example: The Sushi Process

A process transforms an initial state into a final state via multiple actions.
5. Process Mining

5.1 Introduction

- Many actions are performed in consecutive order
Some actions are performed in parallel.
All branches have to be performed.
The exact temporal order between branches is not strict.
Process Properties: Choice

- One branch is selected.
- Either by active decision (manager) or passive selection (environment).
Process Properties: Loop

- Repeated execution of actions.
- Often used as a "continuous improvement cycle".

![Diagram of the process loop](image)
Benefits of Process Models

• Insights by changing perspectives and highlights.

• Specification / Documentation for certifications or legal contract purposes.

• Verification of executions to reveal problems.

• Performance analysis to identify issues like bottlenecks.

• Simulation (digital twin) to experiment virtually with changed settings.
Information Flow of Event Data

- **People** supports/controls **Software Systems** by recording events e.g., messages, transactions, etc.
- **Components** (e.g., businesses, organizations) relate to **World** with people, machines, and supports/controls.
- **Event Logs** specify, configure, implement, analyze, discover, conformance, and enhance **Process Models**.
## Event Logs as Starting Point

<table>
<thead>
<tr>
<th>case id</th>
<th>activity</th>
<th>timestamp</th>
<th>resource 1</th>
<th>resource 2</th>
<th>execution quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sushi 113</td>
<td>get ingredients</td>
<td>09:31</td>
<td>Andreas</td>
<td>bag</td>
<td>good</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>slice salmon</td>
<td>09:35</td>
<td>Bianca</td>
<td>knife 1</td>
<td>medium</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>spread on nori sheet</td>
<td>09:42</td>
<td>Bianca</td>
<td></td>
<td>very good</td>
</tr>
<tr>
<td>Sushi 248</td>
<td>eat</td>
<td>09:43</td>
<td>Charlie</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Sushi 249</td>
<td>get ingredients</td>
<td>09:47</td>
<td>Andreas</td>
<td>bag</td>
<td>good</td>
</tr>
<tr>
<td>Sushi 113</td>
<td>cook rice</td>
<td>09:51</td>
<td>Bianca</td>
<td>rice cooker 3</td>
<td>poor</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>roll and slice</td>
<td>09:51</td>
<td>Charlie</td>
<td>knife 1</td>
<td>good</td>
</tr>
<tr>
<td>Sushi 113</td>
<td>peel avocado</td>
<td>09:53</td>
<td>Andreas</td>
<td>knife 2</td>
<td>poor</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>add soy sauce</td>
<td>09:54</td>
<td>Bianca</td>
<td></td>
<td>good</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>add soy sauce</td>
<td>09:55</td>
<td>Bianca</td>
<td></td>
<td>poor</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>eat</td>
<td>09:57</td>
<td>Andreas</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

...
Event Logs Technically

- Data collection mostly fully automated.
- Process-Aware Information Systems (PAIS)
  - ERP (Enterprise-Resource Planning) [SAP, Oracle]
  - BPM (Business Process Management) [IBM BPM]
  - CRM (Customer Relationship Management)
- Popular data format: XES
  - XML-based
  - easy to understand
Event Logs Formally

An **event** $e$ is a tuple $e = (c, a, t, ...)$ containing a case identifier $c$, an activity label $a$ and a timestamp $t$.

An event can contain additional attributes.

For an event $e = (c, a, t)$, we define the projections $\#_{case}(e) = c$, $\#_{activity}(e) = a$, and $\#_{time}(e) = t$.

An **event log** $L$ is a multiset of events.
Event Logs Formally

A case $C$, identified by $c$ in the log, is the set of events

$$C = \{ e \in L \mid \#_{case}(e) = c \}$$

A trace $\sigma_c$ is the sequence of activities for a case $C = \{e_1, \ldots, e_n\}$ with

$$\sigma_c = \#_{activity}(e_{\pi(1)}), \ldots, \#_{activity}(e_{\pi(n)})$$

such that $\#_{timestamp}(e_{\pi(i)}) < \#_{timestamp}(e_{\pi(j)})$ for $\pi(i) < \pi(j)$. 

<table>
<thead>
<tr>
<th>case id</th>
<th>activity</th>
<th>timestamp</th>
<th>resource 1</th>
<th>resource 2</th>
<th>execution quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sushi 113</td>
<td>get ingredients</td>
<td>09:31</td>
<td>Andreas</td>
<td>bag</td>
<td>good</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>slice salmon</td>
<td>09:35</td>
<td>Bianca</td>
<td>knife 1</td>
<td>medium</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>spread on nori sheet</td>
<td>09:42</td>
<td>Bianca</td>
<td>very good</td>
<td></td>
</tr>
<tr>
<td>Sushi 248</td>
<td>eat</td>
<td>09:43</td>
<td>Charlie</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sushi 249</td>
<td>get ingredients</td>
<td>09:47</td>
<td>Andreas</td>
<td>bag</td>
<td>good</td>
</tr>
<tr>
<td>Sushi 113</td>
<td>cook rice</td>
<td>09:51</td>
<td>Bianca</td>
<td>rice cooker 3</td>
<td>poor</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>roll and slice</td>
<td>09:51</td>
<td>Charlie</td>
<td>knife 1</td>
<td>good</td>
</tr>
<tr>
<td>Sushi 113</td>
<td>peel avocado</td>
<td>09:53</td>
<td>Andreas</td>
<td>knife 2</td>
<td>poor</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>add soy sauce</td>
<td>09:54</td>
<td>Bianca</td>
<td></td>
<td>good</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>add soy sauce</td>
<td>09:55</td>
<td>Bianca</td>
<td></td>
<td>poor</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>eat</td>
<td>09:57</td>
<td>Andreas</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Integration into the Data Mining World

Itemsets (e.g. frequent itemset mining)

\{rice, avocado, salmon\}

Processes

- get ingredients
- peel avocado
- slice salmon
- cook rice
- slice avocado
- add wasabi
- add soy sauce
- season rice
- season
- spread on nori sheet
- roll and slice
- eat

Sequences (e.g. sequential pattern mining)

get ingredients → prepare ingredients → spread on nori sheet → roll and slice → season with wasabi → season with soy sauce → eat

- unordered
- set-based

- partially ordered
- sequences can occur
- models are directed graphs
- branches break order (concurrency)

- strictly totally ordered
- sequence-based
Process Mining Task: Discovery

- Given an event log, find a process model which
  - must be able to replay the log ⇒ *Fitness*
  - simplifies as far as possible ⇒ *Simplicity*
  - does not overfit the log ⇒ *Generalization*
  - does not underfit the log ⇒ *Precision*

<table>
<thead>
<tr>
<th>case id</th>
<th>activity</th>
<th>timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sushi 113</td>
<td>get ingredients</td>
<td>09:31</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>slice salmon</td>
<td>09:35</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>spread on nori sheet</td>
<td>09:42</td>
</tr>
<tr>
<td>Sushi 248</td>
<td>eat</td>
<td>09:43</td>
</tr>
<tr>
<td>Sushi 249</td>
<td>get ingredients</td>
<td>09:47</td>
</tr>
<tr>
<td>Sushi 113</td>
<td>cook rice</td>
<td>09:51</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>roll and slice</td>
<td>09:51</td>
</tr>
<tr>
<td>Sushi 113</td>
<td>peel avocado</td>
<td>09:53</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>add soy sauce</td>
<td>09:54</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>add soy sauce</td>
<td>09:55</td>
</tr>
<tr>
<td>Sushi 239</td>
<td>eat</td>
<td>09:57</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Process Mining Task: Conformance Checking

- Given an event log and a process model, decide for each case whether it conforms to the model or not. If not, give the issues.

- A case instance can perform better than others. Then reveal the beneficial deviations to improve the general workflow.

- If the case performs worse, identify the root cause to avoid misbehavior.

cook rice, add wasabi, roll and slice, eat

[Diagram showing process and outcomes: conform, non-conform]

Housebreaking

Trails

Tool choice
Process Mining Task: Enhancement

• Given a process model, augment with additional information.
  • Temporal information
  • Social networks
  • Organisational roles
  • Decision rules
Mostly: Cases related to people. But what is in the data?

- Students: Who asks the most questions?
- Employees: Who is associated with long execution terms?
- Tenants: Who needs maintenance often?
- Clients: Who calls most for service?

5. Process Mining

5.1 Introduction

neutral, objective, data-oriented
Process Mining Risks and Green Data Science

• Same results, but with intentional mindset:
  • Students  Who is the least intelligent student?
  • Employees  Who is the slowest worker?
  • Tenants  Who caused the most repairs?
  • Clients  Who complains the most?

5. Process Mining

5.1 Introduction

Made with StyleGAN
arXiv:1912.04958

bad intention, negative-subjective, pessimistic
• And the other extreme, changed mindset:
  • Students *Who is the most interested student?*
  • Employees *Who handles the most difficult tasks?*
  • Tenants *Who takes care of the rental property?*
  • Clients *Who gives a lot of constructive feedback?*

Made with StyleGAN arXiv:1912.04958

**Process Mining Risks and Green Data Science**
• Be careful with interpretations.
• Even if you are objective, can your results be interpreted otherwise?
• Can you obscure the results so they stay meaningful, but protect individuals?
  e.g. Cluster individuals, top-k-rankings, k-anonymity, hashing, noise addition,…
Scientific Process Mining Tools

• **PROM:**
  • First version in 2010.
  • Java-based.
  • Provides many algorithms in a GUI.

• **pm4py:**
  • First version in 2019
  • Python-based
  • Documentation: [https://pm4py.fit.fraunhofer.de/](https://pm4py.fit.fraunhofer.de/)
  • Several algorithms available
1. Introduction
2. Basics
3. Supervised Methods
4. Unsupervised Methods

5. Process Mining
   5.1 Introduction
   5.2 Process Models – An Overview
   5.3 Process Discovery
   5.4 Conformance Checking
   5.5 Additional Mining Tasks
   5.6 Streams
Why do we need Process Models?

• Predetermine operational processes in the form of guidelines
  • Descriptive vs. Normative model

• Visualization of processes

• Process reasoning

• Analysis of given processes
  • Starting point for initial implementation and re-design
  • Distribution of responsibilities
  • Planning and controlling
  • Compliance checking
  • Performance prediction via simulation
  • …
Process Model – BPMN (Business Process Modeling Notation)

Remember?

5. Process Mining

5.2 Process Models – An Overview
Process Model – BPMN (Business Process Modeling Notation)

Exemplary subset of elements contained in BPMN

- **AND-split gateway**
  - **AND-join gateway**
- **XOR-split gateway**
  - **XOR-join gateway**
- **OR-split gateway**
  - **OR-join gateway**
- **Start event**
  - **End event**

![Process Model – BPMN](image)

Remember?

- **Get ingredients**
- **Spread on non-slip**
- **Eat**

5. Process Mining
5.2 Process Models – An Overview
5. Process Mining

5.2 Process Models – An Overview

Definition (Transition system)

Triplet $T = (S, A, T)$, where
- $S$ is the set of states
- $A \subseteq A$ is the set of activities
- $T \subseteq S \times A \times S$ is the set of transitions
- $S^{\text{start}} \subseteq S$ is the set of initial states
- $S^{\text{end}} \subseteq S$ is the set of final states
5. Process Mining

5.2 Process Models – An Overview

Process Model – Petri Nets

- AND-split
- p1 -> cook rice -> p4 -> season rice -> p6
- peel avocado -> p5 -> slice avocado
- slice salmon -> p7
- spread on nori sheet -> p9
- roll and slice
- XOR-join
- end -> eat
- add wasabi
- add soy sauce
- XOR-split
- p11

- cook rice
- transition
- place
- token
As already seen the Petri net is a bipartite graph.

**Definition (Petri Net)**

Triplet $N = (P, T, F)$, where

- $P$ is a finite set of places
- $T$ is a finite set of transitions, $P \cap T = \emptyset$
- $F \subseteq (T \times P) \cup (P \times T)$ is a set of directed arcs (called flow relation)

**Exemplary formalization of given Petri Net:**

$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, \text{end}\}$

$T = \{\text{cook rice, season rice, peel avocado, slice avocado, slice salmon, spread on nori sheet, roll and slice, add wasabi, add soy sauce, eat}\}$

$F = \{(p_1, \text{cook rice}), (p_2, \text{peel avocado}), (p_3, \text{slice salmon}), (\text{cook rice}, p_4), (\text{peel avocado}, p_5), \ldots\}$
**Process Models – Workflow-Nets (WF-Nets)**

Subclass of Petri Nets

**Definition (Workflow Net)**

\[ N = (P, T, F), \text{ where} \]

\((P, T, F)\) is a Petri net as already defined

\(N\) is a workflow net iff.

a) \(P\) contains a source place \(i\) s.t. \(\bullet i = \emptyset\)

b) \(P\) contains a sink place \(o\) s.t. \(o \bullet = \emptyset\)

c) If we add a transition \(t^*\) to \(N\) which connects \(o\) with \(i\)

i.e. \(\bullet t^* = \{o\}\) and \(t^* \bullet = \{i\}\), then

the resulting Petri net is strongly connected.

**Definition (Strongly connected)**

A Petri net is strongly connected iff for every pair of nodes (i.e. places and transitions) \(x\) and \(y\), there is a path leading from \(x\) to \(y\)

---

**Can the Petri Net shown be considered a Workflow Net?**
A WF-net does not necessarily represent a correct process
→ Deadlocks, livelocks, not activatable activities etc. are possible

**Definition (Soundness)**

Let $N = (P, T, F)$ be a workflow net with $i$ and $o$ as input and output places. $N$ is sound iff.

- **(safeness)** Places do not hold multiple tokens at the same time
- **(proper completion)** The moment the procedure terminates there is a token in place $o$ and all the other places are empty
- **(option to complete)** For any case the procedure will terminate eventually
- **(absence of dead parts)** For any $t \in T$ there is a firing sequence enabling $t$
Process Models – Methods (Verification)

**Verification** is a method to analyze process models against specific properties (*Model checking*).

- Those properties can be expressed in temporal logic.
- Specifically in LTL (Linear Temporal Logic) which is an significant example in relation to process models.

Two further exemplary verification tasks in the following:

1. Two process models can be checked against each other using **Verification**.
   E.g. Trying to match a descriptive and a normative model to see where reality differs from guidelines
2. Soundness as a correctness criterion can be checked using **Verification**.

<... roll and slice, add wasabi> leads to a dead marking \([p_{11}, p_{14}]\) (Deadlock)
**Known process model types so far:**

- Transitions systems
- BPMN
- Petri Nets
- Workflow Nets

There are still others like
- Reachability graphs
- Causal nets
- ...

**Benefit:**

- Process analysis gets simplified
- Predict performance via simulation
- Predetermine guidelines
- Purpose determines outcome
- ...
Creating a model is not an easy task

- Capturing human behavior
  - Human covers multiple processes with different priorities → dependencies evolve → Difficult to model one process in isolation
  - Productivity of a human is varying over time. It also depends on other factors e.g. Yerkes-Dodson law
Process Models – Discussion (cont.)

- **Idealization of reality**
  - Hand-made models tend to be
    - subjective
    - oversimplified

- The choice of a representative sample of cases is crucial
  - Biased focus on *normal / desirable* behavior

Actually it only covers 80% of cases but is seen as a representative

Remaining 20% could possibly cover high amount of problems
• **Granularity**
  
  E.g. there are many types of sushi: Nigiri, Sashimi, Maki, Uramaki…

  ![Image](image.png)

  I just want to eat sushi...

  E.g. *discrete* vs. *continuous*

  ```
  cook rice  VS.  get pot → get rice → fill rice into pot → add water → ...
  ```

  ⇒ A suitable granularity for the process model depends on
  
  - the input data
  - the model’s purpose
References

