Process Mining
(ProMi)

Winter 2015/16

Matthias Weidlich
Recall: presentation is requirement for taking the final exam

By December 4, 2015:
- Team up in pairs of two
- Submit by mail (matthias.weidlich@hu) the name of your ranked selection of three topics

Assignment of topics/slots will happen soon after that

Preparation time:
- No lecture/recitation on 05.01.16, but, option for consultation

Links to all papers can be found on the course website
III. Conformance Checking
Alignment-based Techniques
Idea

Assessing the conformance of an event log with a model based on an alignment of activities and events

• Consider the set of activities of the model as a set of symbols
• Then, each execution sequence of the model is a sequence of symbols
• Each trace of the event log is also a sequence of symbols

An alignment between two sequences is established by

• Linking pairs of symbols in each sequence
• Such that the order between aligned symbols is preserved

The notion of an alignment allows for quantification of conformance and insights on non-conformance
Example

Model

Event log

Slides partly due to Boudewijn van Dongen
Example

Model

Event log

Alignment
Example

Model

Event log

Alignment
Example

Model

Event log

Alignment
Example

Model

Event log

Alignment

abfdg
...
...

ab d
d
ab f d

g
pay compensation
h
reject request
...
Example

Model

Event log

Alignment

a b d e

a b f d
Example

Model

Event log

Alignment

abfdg

...  ...

abfdg  abfdg  abfdg

Event log

Alignment
Example

Model

Event log

Alignment

Non-Conformance!
Idea Cont.

<table>
<thead>
<tr>
<th>Execution Sequence</th>
<th>a</th>
<th>d</th>
<th>b</th>
<th>e</th>
<th>f</th>
<th>d</th>
<th>c</th>
<th>e</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>a</td>
<td>d</td>
<td>b</td>
<td>e</td>
<td>f</td>
<td>d</td>
<td>c</td>
<td>e</td>
<td>g</td>
</tr>
</tbody>
</table>
Approach

Steps to realise the idea of alignment-based conformance analysis

- Definition of alignments
- Construction of (optimal) alignments
- Conformance measures

Here, focus is on basic approach, various variations have been presented in the literature:

- For different types of process models
- Using different strategies to construct optimal alignments
- Incorporating weighting schemes to fine-tune the importance of activities for conformance checking

Details:
The Notion of Moves

Alignment is based on “moves” in trace or execution sequence

- Specific symbol $\bot$ to denote “no move”
- Set $T_\sigma$ as the transitions of trace $\sigma$ and $T_\sigma^\perp = T_\sigma \cup \{\bot\}$
- Set $T_\pi$ as the transitions of execution sequence $\pi$ and $T_\pi^\perp = T_\pi \cup \{\bot\}$

One step is a pair $(x, y) \in T_\sigma^\perp \times T_\pi^\perp$ and

- $(x, y)$ is a move in log if $x \in T_\sigma$ and $y = \bot$
- $(x, y)$ is a move in model if $x = \bot$ and $y \in T_\pi$
- $(x, y)$ is a move in both if $x \in T_\sigma$ and $y \in T_\pi$
- $(x, y)$ is an illegal move in log if $x = \bot$ and $y = \bot$

$T_{\sigma\pi} = \{(x, y) \in T_\sigma^\perp \times T_\pi^\perp \mid x \in T_\sigma \lor y \in T_\pi\}$ is the set of all legal moves
Alignment Definition

An alignment of trace $\sigma$ and execution sequence $\pi$ is a sequence of steps $\gamma \in T_{\sigma\pi}^*$, such that

- The projection of $\gamma$ on its first component, ignoring $\bot$, is $\sigma$
- The projection of $\gamma$ on its second component, ignoring $\bot$, is $\pi$

Example:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>$\bot$</th>
<th>d</th>
<th>e</th>
<th>g</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>$\bot$</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>$\bot$</td>
<td>h</td>
<td></td>
</tr>
</tbody>
</table>

Note: Given a trace and an execution sequence

- There is more than one alignment
- The set of possible alignments is finite
Alignment Examples

\[ \gamma_1 = \begin{array}{cccccc}
  a & c & d & e & h \\
  \hline
  a & c & d & e & h \\
\end{array} \]

“Ideal” alignment!

\[ \gamma_2 = \begin{array}{cccccc}
  a & b & \bot & d & e & g & \bot \\
  \hline
  a & \bot & c & d & e & \bot & h \\
\end{array} \]

“Imperfect” alignment!

\[ \gamma_3 = \begin{array}{cccccc}
  a & b & \bot & d & e & \bot & \bot & g \\
  \hline
  \bot & a & c & d & \bot & e & \bot & h \\
\end{array} \]

“Non-optimal” alignment!

\[ \gamma_4 = \begin{array}{cccccc}
  a & \bot & b & d & e & g & \bot \\
  \hline
  a & \bot & c & d & e & \bot & h \\
\end{array} \]

Not an alignment!
Cost of Moves

Introduce cost for legal moves

- Function $\delta: T_{\sigma\pi} \rightarrow \mathbb{N}$ assigns cost to each move
- Typically, $\delta(x, x) = 0$

Standard cost function, for $x \in T_{\sigma}$ and $y \in T_{\pi}$:

- $\delta(x, \bot) = 1$ (move in log)
- $\delta(\bot, y) = 1$ (move in model)
- $\delta(x, y) = 0$ if $x = y$ (equal move in both)
- $\delta(x, y) = \infty$ if $x \neq y$ (different move in both)

Various further cost functions possible

- Take into account importance of activities (skip payment vs. skip logging)
- Consider similarity of activities (preliminary check vs. partial check)
Examples Again

\[\gamma_1 = \begin{array}{cccccc}
a & c & d & e & h \\
a & c & d & e & h \\
\end{array}\]
Cost: 0

\[\gamma_2 = \begin{array}{ccccccc}
a & b & \bot & d & e & g & \bot \\
a & \bot & c & d & e & \bot & h \\
\end{array}\]
Cost: 4

\[\gamma_3 = \begin{array}{cccccccc}
a & b & \bot & d & e & \bot & \bot & g \\
\bot & a & c & d & \bot & e & h & \bot \\
\end{array}\]
Cost: \(\infty\)
Optimal Alignment

So far, cost of *some* alignment of trace with respect to *some* execution sequence

- Optimal alignment of trace $\sigma$ and execution sequence $\pi$: alignment with minimal cost
- Optimal alignment of trace $\sigma$ and model, i.e., set of complete execution sequences $\Pi$: alignment with minimal cost of all optimal alignments of $\sigma$ and complete execution sequence $\pi \in \Pi$
- Notation: optimal alignment of $\sigma$ and execution sequence $\pi$ is $\gamma^*(\sigma, \pi)$, optimal alignment of $\sigma$ and execution sequences $\Pi$ is $\gamma^*(\sigma, \Pi)$

Optimal alignment of trace $\sigma$ and model always exists

- Maybe the trivial alignment that first only moves in trace and then in shortest complete execution sequence, or vice versa
- Not unique, multiple alignments may show minimal cost

Finding the optimal alignment is expensive in general
Example

Trace: abefbh
Cost: 3
The Problem of Finding Optimal Alignments

The search space is a “product” of the statespace of the model and the trace

Each node is a combination of a state in the model and the executed events in the trace

Each arc is a move in model, move in log or move in both

Example:

Trace: < A, A, B, A >

9 states, 13 transitions
Alignment search space

Find the shortest path from the top-left to the bottom right

45 states,
121 transitions

Find the shortest path from the top-left to the bottom right

Slides partly due to Boudewijn van Dongen
The alignment for trace \langle A, A, B, A \rangle:
\langle (0, -), (1, -), (2, A), (3, -), (2, A), (4, B), (5, A) \rangle
Implicit Execution

Implicit execution:
- Of silent transitions in a net system
- Of operator nodes in a Process Tree

6 states, 9 transitions
Implicit execution – Reduced Statespace

- 45 – 30 states,
- 121 – 86 transitions
Computing Alignments

In general, finding the shortest path in a weighted, directed graph

Baseline: Dijkstra's algorithm

• Visit each node at most once
• Proceed by iteratively exploring nodes with minimal distance to source
• In each iteration, compute distances to neighbours of current node

Yet, A* as a more efficient algorithm, where the search is guided by heuristics
Recall: A* Search

OPEN set, initialised with source node
CLOSED set, initialised as empty set

function g, cost on best known path
g(source) := 0; g(n) := ∞ for all other n

function f, cost estimate to goal
f(source) := g(source) + \texttt{heuristic}(source, goal); f(n) := ∞ for all other n

while OPEN is not empty
    find node q with least f in OPEN, remove q from OPEN, add q to CLOSED
    if q is the goal, stop the search

    for each successor of q
        if successor in CLOSED, continue
        score := g(q) + weight(q, successor)
        if successor not in OPEN, add successor to OPEN
        else if score >= g(successor), continue
        g(successor) := score
        f(successor) := g(successor) + \texttt{heuristic}(successor, goal)

return none
Search Heuristics

In general: heuristic must be admissible, i.e., never overestimate the cost to reach goal and be monotonic.

Simple estimator for our setting: length of remaining trace

• Never overestimates: best solution needs to align all the remaining transitions in this trace
• Is monotonic: move in model yields the same estimate, move in log and move in both lower the estimate

Not here: more advanced estimators based on ILP
Example

Estimator: remaining trace length
- 45 30  21 states,
- 121 86  32 transitions
Conformance Analysis

Alignments enable quantification of conformance of event logs w.r.t. a process model.

Compared to approaches discussed earlier, two common issues are not problematic when using alignments:

- Silent transitions do not need to be treated.
- Duplicated transitions in the model can be integrated directly.

Alignments form the basis not only for conformance analysis, but a full range of analysis techniques.
Fitness Measures

Absolute fitness of a trace $\sigma$ regarding a given model

- Defined by the cost $\delta$ of the optimal alignment of $\sigma$ with the model
- Using the standard cost function, this corresponds to the number of
  moves in log and moves in model

Absolute fitness of a multiset $S = [\sigma_1^{n_1}, \ldots, \sigma_m^{n_m}]$ of traces regarding model $M$ with complete execution sequences $\Pi$

- Sum of costs of optimal alignments of traces

$$f\text{cost}(S, \Pi) = \sum_{1 \leq i \leq m} n_i \delta(\gamma^*(\sigma_i, \Pi))$$
Normalised Fitness

Various solutions to normalise the absolute fitness measure

Here: division by maximal possible value

- Assume that a move in both \((x, y)\) happens only if \(x = y\)
- Worst case: alignment is built only from moves in model and moves in log
- With \(S\) as the multiset of traces, the cost of moving through the log without moving in the model is:
  \[
  c(S) = \sum_{1 \leq i \leq m} n_i \sum_{x \in \sigma_i} \delta(x, \perp)
  \]
  - The cost of moving only in the model (via the least expensive path) is:
  \[
  c(\Pi) = \min_{\pi \in \Pi} \sum_{y \in \pi} \delta(\perp, y)
  \]

Normalised fitness measure:

\[
\text{fitness}(S, \Pi) = 1 - \frac{f\text{cost}(S, \Pi)}{c(S) + \sum_{1 \leq i \leq m} n_i c(\Pi)}
\]
Local Feedback in Trace/Model

Characterise “hotspots” of non-conformance in trace or model

• Approach similar to the one introduced for relational conformance checking

• Instead of considering elements of behavioural relations as entities for feedback, the violation pair (transition, move type) is used (move type is either move in log or move in model)

• Frequent occurrence of such a pair hints at hotspot

Illustration:
Global Feedback

To obtain further insights on root causes of non-conformance
  • Lifting the notions of violation support and confidence of introduced for relational approaches to the (transition, move type) violation pairs

Violation support: number of traces that show particular violation in optimal alignment

Confidence of violation rules: given two violations $v$ and $v'$, determine ratio of number of traces showing both violations and number of traces showing only violation $v$
Beyond Conformance Checking

Alignments can not only be used to quantify fitness

Specifically, the precision-generalisation trade-off can be addressed

One approach to measure precision and generalisation

- Use alignment prefixes to characterise states (assuming a fully fitting and deterministic model)
- Determine enabled transitions per state based on alignment
- Observe changes in the set of enabled transitions for each state change
Event Log Analysis

So far, alignments between a trace of an event log and an execution sequence of a model

• Pairwise alignment
• Each trace is considered in isolation

But: Alignments are also useful in the absence of a model

• Support explorative analysis of an event log
• Insights into regularities and most common behaviour among the traces
• Identification of rare deviations
• Powerful visual feedback
Multiple Trace Alignment

Construct alignment between traces of event log
  • No longer a question of optimising the alignment cost for a single trace
  • Global view: overall alignment cost should be minimal

Problem well-known in genomics
  • Alignments of nucleic acid sequences
  • Yet, also known to be an NP-complete problem

Various heuristic techniques to find multiple trace alignment that may be non-optimal
  • Typically based on iterative approach
  • Often based on hierarchical clustering
Intuition

$T_1: jgcflebd$
$T_2: jgclebdfi$
$T_3: jgclebd$
$T_4: jgclfebd$
$T_5: jgclefbd$
Specific Techniques

Standard techniques may lead to results that are not intuitive from a business process point of view

Example: tendency to penalise gaps at the beginning and end of traces

- Not in line with “looping” behaviour in processes
- Consider locality in the alignment

\[ \text{j g c - a h b --- f d} \]
\[ \text{j g c f a h b d k a h b d} \]
\[ \text{j g c - a h b f d ---} \]
\[ \text{j g c f a h b - d k a h b d} \]
Insights

Alignment of a cluster of real world traces (house/apartment rentals):
b0 is check activity (inspection done?), sequence b4a8b0 is done only in negative case (tenant not at home -> send letter, reschedule)
Insights

h1g9 is well preserved (final inspection, recording of defects)
d1d0 is offering a flat, before final inspection in first trace!
Insights

b5e5 are related to second inspection – rarely needed.
c9 is determination of candidate tenant – happened multiple times in one trace (featuring f0 termination of provisional lease)
Take Away

Alignments are a basic technique to compare event logs and models.

Main idea: establish the cost-optimal relation between each observed trace and an execution sequence of a model.

Enables quantification of conformance.

Additional techniques for behavioural comparison based on alignments.