MDL Principle in Process Models Evaluation

Toon Calders,
Mykola Pechenizkiy
Information Systems Group
Dept. of Computer Science
Eindhoven University of Technology
the Netherlands

Anne Rozinat,
Christian Günther
Information Systems Group
Dept. of Technology Management

Spring Workshop on Mining and Learning 2008
Outline

- Process mining
  - Tasks, techniques, challenges
- Evaluation of process models
  - Commonly used measures and their limitations
- MDL-based measure of process model quality
  - Compression-based DM and MDL principle
  - Applicability to process mining
  - Model complexity/log compression ratio trade-off
- Ongoing and Future work
  - Evaluation
  - MDL principle for guiding process mining
Process mining

- Extracts process models from event logs
  - Discovery;
  - Conformance; Extension

- Different model classes exist
  - Petri-nets
  - EPC, YAWL, BPEL;
  - Markov models, …

- Several process mining techniques exist
  - Alpha miner;
  - Heuristic miner;
  - Genetic miner, …
Examples:
The objective is to mine …

- structured, easy to understand process models
  - just like this one

- but …
... in reality they often look like spaghetti
A few measures are popular
- accuracy/fitness related
- structural (number of places, transitions etc)

These measure have certain limitations
- are model-dependent,
- assume that the model that generated the log is known
- need negative examples of event sequences

Our focus here:
- MDL-base process model(s) quality measure
Our Approach for Defining the Process Quality

- **MDL principle**
  - Minimizing the total encoding costs equal to
  $$\text{EncodingCost}(\text{EventLog} | \text{Model}) + \text{EncodingCost}(\text{Model})$$

- **Rationale:**
  - the more accurate a model fits the reality, the better,
  - i.e., more succinct, it will be able to describe the event log and vice versa.

- **Need to define how**
  - to encode a Petri-net
  - to encode traces from an event log given a Petri-net
  - both is possible ;-) (the manuscript is available upon your request)
Encoding costs with PetriNets

- M: Number of places, transitions, incoming/outgoing links
- L: Explicit encoding with violating transitions EEVT
  - enabled transitions in a replay have a much shorter encoding than faulty transitions,
    - no need to trigger an error recovery mechanism in the encoding.

Running Petri-net example from:
Rozinat et al. 2007 Towards an Evaluation Framework for Process Mining Algorithms. BPM Center Report, BPMcenter.org
MDL Principle in Process Models Evaluation

Flower Reference Model (FRM)

<table>
<thead>
<tr>
<th>No. of Instances</th>
<th>Log Traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>1207</td>
<td>ABDEI</td>
</tr>
<tr>
<td>145</td>
<td>ACDGHFI</td>
</tr>
<tr>
<td>56</td>
<td>ACGDHFI</td>
</tr>
<tr>
<td>23</td>
<td>ACHDFI</td>
</tr>
<tr>
<td>28</td>
<td>ACDHFI</td>
</tr>
</tbody>
</table>
Explicit Reference Model (ERM)
Encoding costs

\[ 1 - \frac{\text{ERM}_e}{\text{FRM}_e} \]

Heaven – high compression and low complexity

\[ 1 - \frac{\text{model}_e}{\text{ERM}_e}; \quad 1 - \frac{\text{model}_e}{\text{FRM}_e}; \]

FRM – definitely bad (low) compression

ERM – definitely bad (high) complexity

worse than definitely bad

model simplicity

23.04.08
Traben-Trarbach, Germany

MDL Principle in Process Models Evaluation
by M. Pechenizkiy
Optimal: Kolmogorov (undecidable)
Close to optimal: LZip or similar
Baseline: min (FRM, ERM)
Ongoing and Future Work

- **Short run**
  - Extensive experimental studies
  - Evaluation of trace clustering
  - MDL principle for guiding process mining
    \[ I = \alpha \cdot \text{Model}_e + (1 - \alpha) \cdot \text{Model}_c \]

- **Long run**
  - Some success stories exist, but
    \[ \text{to a large extend the state-of-the art techniques still have problems with scalability and robustness} \]
  - Adaptation of sequence mining, graph mining and other data mining approaches for
    - development of the new robust and scalable process mining techniques
Questions  Suggestions  Collaboration
all warmly welcome

Please consider submitting your work and attending
ECML/PKDD Workshop on
Discovery of Process Models
Trace Clustering

- Given the whole log, process mining techniques find spaghetti-type of process models
- The hope is that if traces are clustered into homogeneous partitions, process mining techniques can do better
  - i.e. instead of one global spaghetti model there will be several local more intuitive to the user models

- How many clusters?
  - Current approaches
    - minimization of MAE etc, plus
    - maximization of the (weighted) fitness of the local models
    - => if \#cluster = \#traces then we can get same number of perfectly precise models
      - some penalization is needed …
    - but MDL-based measure does this without any extra effort
MDL Principle in Process Models Evaluation
by M. Pechenizkiy
Process models from the clusters

Diagnosis process

Treatment process

23.04.08
Traben-Trarbach, Germany