Keynote

Cracking the Nut: Unraveling Challenges in Predictive Process Monitoring

Jochen De Weerdt
jochen.deweerdt@kuleuven.be
About me

• Professor at KU Leuven (Research Centre for Information Systems Engineering)
• PhD in Business Economics (2012)
• Postdoc at the IS School, Queensland University of Technology (QUT), Brisbane (2012-2013)

• Research expertise
  • Process Mining
    • Trace clustering
    • Discovery and conformance checking
    • Predictive Process Monitoring
  • Business Analytics
    • Fraud analytics, learning analytics, real-estate, NLP, marketing, etc.
Successfully supervised PhDs

Pieter De Koninck (2019)
Process mining – trace clustering
AI Lead at Silverfin

Sandra Mitrovic (2019)
Network analytics for churn prediction
Postdoc at IDSIA

Daria Bogdanova (2021)
Feedback in smart learning environments
Customer Manager at Sitecore

Galina Deeva (2021)
Learning analytics – Process mining
Data Scientist at KBC Bank

Rafael Van Belle (2023)
Network analytics for fraud detection
Machine Learning Engineer at Dataroots

Björn Rafn Gunnarsson (2023)
Predictive Process Monitoring with LSTMs
Data Scientist at NATO

Hans Weytjens (2023)
Uncertainty for Predictive Process Monitoring
Senior Researcher at TUM & KU Leuven

Jari Peeperkorn (2023)
Predictive Process Monitoring generalization & conformance
Postdoc at LIRIS
Current team

Philipp Borchert
NLP for business analytics

Margot Geerts
Real estate valuation modelling

Carlos Eduardo Ortega Vázquez
Imbalanced learning for fraud detection

Xin Pang
Process execution visualization

Philipp Borchert
NLP for business analytics

Yongbo Yu
Process Model Forecasting

Brecht Wuyts
Predictive Process Monitoring

Yannis Bertrand
IoT process mining

Kevin Biermans
Inter-case Featurization for Predictive Process Monitoring

Jakob De Moor
Prescriptive Process Monitoring

Zahra Ahmadi
IoT process mining

Keynote, ML4PM Workshop at ICPM 2023, Rome (Italy)
Agenda

• Five key challenges in Predictive Process Monitoring (PPM)
  1. Strategies for PPM evaluation
  2. Generalization in deep learning models
  3. The inter-case perspective
  4. From case-level to model-level predictions
  5. Increasing adoption
Challenge 1: Strategies for PPM evaluation
Widespread bad practices in PPM

• Train 80%, validation 16%, test 4%
• Evaluate the model for different prefix lengths, then average those results
• Compare with previously published results using totally different setups
• Random train-test split for outcome or remaining time prediction
• Random k-fold cross-validation for outcome or remaining time prediction
• Test set overuse: too many models tested on the same test set
• Example leakage* and other data leakage

Failing to factor in the dynamics of event logs

Data leakage
Removing test set bias

(Different) prefixes but obtained from the same traces should not be part of both training and testing set – “strict temporal splitting”

Cases for which we don’t observe the outcome (unknown), should not be in the test set (and are often not part of the dataset in general)

This causes two types bias: the number of running cases and their average length no longer reflect the underlying reality (e.g. inter-case variables)

→ Remove the black prefixes from test set
→ Grey prefixes of the red-gray cases should be included in the test set

Out-of-time cross-validation might become difficult
Learn from other domains

CORE: A Few-Shot Company Relation Classification Dataset for Robust Domain Adaptation.

Philipp Borchert¹², Jochen De Weerdt², Kristof Coussement¹,
Arno De Caigyn¹, Marie-Francine Moens³
¹IESEG School of Management, Univ. Lille, CNRS, UMR 9221 - LEM -
Lille Economie Management, F-59000 Lille, France
²Research Centre for Information Systems Engineering, KU Leuven, Belgium
³Department of Computer Science, KU Leuven, Belgium

Abstract
We introduce CORE, a dataset for few-shot relation classification (RC) focused on company relations and business entities. CORE includes 4,708 instances of 12 relation types corresponding to textual evidence extracted from company Wikipedia pages. Company names and business entities pose a challenge for few-shot RC models due to the rich and diverse information associated with them. For example, a company name may represent the legal entity, products, people, or business divisions depending on the context. Therefore, deriving the relation type between entities is highly dependent on textual context. To evaluate the performance of state-of-the-art RC models on the CORE dataset, we conduct experiments in the few-shot domain adaptation setting. Our results reveal substantial performance gaps, confirming advertising is pervasive, as one of Woodruff’s stated goals was to ensure that everyone on Earth drank Coca-Cola as their preferred beverage.

Figure 1: Example for information richness embedded in named business entities.

et al., 2019; Sabo et al., 2021), and out-of-domain adaptation (Gao et al., 2019). However, most existing RC datasets comprise a mixture of domains, and the sets of entities across different domains are often easily distinguishable from one another. The
Welcome to FewRel

a Few-shot Relation classification dataset

We have now moved to English competition.

What is FewRel?

FewRel is a Few-shot Relation classification dataset, which features 70,000 natural language sentences expressing 100 relations annotated by crowdworkers.

Please refer to our EMNLP 2018 paper to learn more about FewRel.

We add two more challenging settings: few-shot domain adaptation (DA) and few-shot none-of-the-above detection (NOTA) in FewRel 2.0 dataset. Please refer to our EMNLP 2019 paper for more details.

Leaderboard

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>Domain Adaptation 5-way-1-shot</th>
<th>Domain Adaptation 5-way-5-shot</th>
<th>Domain Adaptation 10-way-1-shot</th>
<th>Domain Adaptation 10-way-5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GPT Anonymous</td>
<td>80.04</td>
<td>92.58</td>
<td>69.25</td>
<td>86.88</td>
</tr>
<tr>
<td>2</td>
<td>DualGraph Anonymous</td>
<td>80.11</td>
<td>91.01</td>
<td>73.89</td>
<td>82.34</td>
</tr>
<tr>
<td>3</td>
<td>PAMN Anonymous</td>
<td>77.54</td>
<td>90.40</td>
<td>65.98</td>
<td>82.03</td>
</tr>
<tr>
<td>4</td>
<td>Anonymous Pony</td>
<td>76.71</td>
<td>86.69</td>
<td>66.72</td>
<td>78.46</td>
</tr>
<tr>
<td>5</td>
<td>Anonymous Python</td>
<td>66.41</td>
<td>83.32</td>
<td>51.85</td>
<td>73.60</td>
</tr>
<tr>
<td>6</td>
<td>Anonymous Groundhog</td>
<td>67.23</td>
<td>82.09</td>
<td>54.32</td>
<td>71.01</td>
</tr>
<tr>
<td>7</td>
<td>BERT-PAIR(paper</td>
<td>code) THUNLP, Tsinghua University</td>
<td>67.41</td>
<td>78.57</td>
<td>54.89</td>
</tr>
</tbody>
</table>

Get Started

It is fairly easy for you to get started with FewRel. First, you can download a copy of the dataset by the following link. The dataset is distributed under the CC BY 4.0 license.

Keynote, ML4PM Workshop at ICPM 2023, Rome (Italy)
Challenge 1: Key takeaways

• With training data, you can (intrinsically) do whatever you like
• Test data should be created rigorously
  • Constructing benchmark datasets with fixed test sets, and/or apply best practices
  • Public data sets should become available with masked test data, as done in other ML domains
  • We need better scientific recognition of making PPM datasets available
• Assuming steady-state is naïve at best
  • Consider the deployment setting: what will your models do in a real-life environment?
  • Creating “hard” test data is what we should aim for
• Of course, you cannot assume that models can “learn” concept drift, however, dealing with concept drift is also a main task for model monitoring during deployment (MLOps)
Challenge 2: Generalization in deep learning models
Key question

• How capable are deep learning models to generalize process behavior?

• Imagine, process discovery:
  • You provide an algorithm with a simulated event log based on the model below
  • You remove one of the 120 possible variants
  • Would you expect a process discovery technique to fail to detect the parallel construct?

Leave One Variant Out Cross-Validation (LOVOCV)

Results

• We built LSTMs to predict the next activity
• Optimized hyperparameters: nr. layers, hidden unit size, dropout rate, L1/L2 regularization

<table>
<thead>
<tr>
<th>Mod.</th>
<th>Pattern</th>
<th>#Var.</th>
<th>$F_{ \text{PMSL} }$</th>
<th>$P_{ \text{PMSL} }$</th>
<th>$G_{ \text{PMSL} }$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PAR</td>
<td>120</td>
<td>0.96 ± 0.01</td>
<td>0.95 ± 0.01</td>
<td>0.00 ± 0.01</td>
</tr>
<tr>
<td>2</td>
<td>XOR</td>
<td>128</td>
<td>0.94 ± 0.01</td>
<td>0.94 ± 0.01</td>
<td>0.80 ± 0.14</td>
</tr>
<tr>
<td>3</td>
<td>XOR+LTD</td>
<td>128</td>
<td>0.95 ± 0.01</td>
<td>0.94 ± 0.00</td>
<td>0.04 ± 0.17</td>
</tr>
<tr>
<td>4</td>
<td>IOR</td>
<td>64</td>
<td>0.93 ± 0.02</td>
<td>0.92 ± 0.02</td>
<td>0.66 ± 0.17</td>
</tr>
<tr>
<td>5</td>
<td>PAR</td>
<td>126</td>
<td>0.95 ± 0.02</td>
<td>0.94 ± 0.02</td>
<td>0.07 ± 0.20</td>
</tr>
<tr>
<td>6</td>
<td>LOOP</td>
<td>27</td>
<td>0.87 ± 0.02</td>
<td>0.86 ± 0.02</td>
<td>0.75 ± 0.28</td>
</tr>
</tbody>
</table>
Is generalization important?
Can we solve the problem?

• Better validation set sampling strategies can help:

  Val. log selection:
  1) Random traces (RAND)
  2) Half: random + variant-based (RVBR)
  3) Variant-based (VBR)

  Trained to predict next event of a prefix

  Training Log

  Validation Log

  Simulated Log

  RNN

  Train

  Early stopping

  Simulate

Peeperkorn J., vanden Broucke, S., & De Weerdt, J. Validation Set Sampling Strategies for Predictive Process Monitoring, Under Review
Insights

• On 6 simple process models, we show that we can increase generalization at a little cost in fitness/precision
  • See table

• If models become too complex, the variant-based resampling becomes less effective

• Also important effect of event log incompleteness, but not yet fully understood

<table>
<thead>
<tr>
<th>Model</th>
<th>Val.</th>
<th>Test Acc.</th>
<th>$F_{PMSL}$</th>
<th>$P_{PMSL}$</th>
<th>$G_{PMSL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RAND</td>
<td>0.726</td>
<td>0.982</td>
<td>0.889</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>RVBR</td>
<td>0.791</td>
<td>0.885</td>
<td>0.873</td>
<td>0.764</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>0.759</td>
<td>0.872</td>
<td>0.845</td>
<td>0.591</td>
</tr>
<tr>
<td>2</td>
<td>RAND</td>
<td>0.749</td>
<td>0.932</td>
<td>0.917</td>
<td>0.772</td>
</tr>
<tr>
<td></td>
<td>RVBR</td>
<td>0.817</td>
<td>0.880</td>
<td>0.878</td>
<td>0.860</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>0.813</td>
<td>0.815</td>
<td>0.815</td>
<td>0.814</td>
</tr>
<tr>
<td>3</td>
<td>RAND</td>
<td>0.735</td>
<td>0.975</td>
<td>0.880</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>RVBR</td>
<td>0.810</td>
<td>0.884</td>
<td>0.877</td>
<td>0.813</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>0.812</td>
<td>0.876</td>
<td>0.870</td>
<td>0.817</td>
</tr>
<tr>
<td>4</td>
<td>RAND</td>
<td>0.794</td>
<td>0.931</td>
<td>0.895</td>
<td>0.532</td>
</tr>
<tr>
<td></td>
<td>RVBR</td>
<td>0.851</td>
<td>0.865</td>
<td>0.841</td>
<td>0.593</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>0.799</td>
<td>0.877</td>
<td>0.853</td>
<td>0.607</td>
</tr>
<tr>
<td>5</td>
<td>RAND</td>
<td>0.798</td>
<td>0.988</td>
<td>0.881</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>RVBR</td>
<td>0.843</td>
<td>0.813</td>
<td>0.795</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>0.850</td>
<td>0.838</td>
<td>0.805</td>
<td>0.538</td>
</tr>
<tr>
<td>6</td>
<td>RAND</td>
<td>0.844</td>
<td>0.840</td>
<td>0.841</td>
<td>0.868</td>
</tr>
<tr>
<td></td>
<td>RVBR</td>
<td>0.862</td>
<td>0.863</td>
<td>0.862</td>
<td>0.838</td>
</tr>
<tr>
<td></td>
<td>VBR</td>
<td>0.862</td>
<td>0.844</td>
<td>0.840</td>
<td>0.779</td>
</tr>
</tbody>
</table>
Challenge 2: Key takeaways

• We need to further investigate validation set sampling techniques
  • Factoring in the data perspective (case + event features)

• Alternative model architectures might work better
  • Moving away from classical RNNs
    • Transformers/attention
    • Graph Neural Networks (GNNs)
    • Transfer learning/finetuning
Challenge 3: The inter-case perspective
Motivation

- Research in predictive process monitoring has generally relied on **intra-case features** in order to make predictions.
- Therefore, assume that the processing of case is solely dependent on the attributes of the case itself.
- However, cases are not processed in isolation:
  - Can be influenced by the processing of other cases.
  - Can be influenced by the general state of a business process.
  - These dynamics can be captured by **inter-case features**.
Motivation

• Research in predictive process monitoring has generally relied on **intra-case features** in order to make predictions.
• Therefore, assume that the processing of case is solely dependent on the attributes of the case itself.
• However, cases are not processed in isolation:
  • Can be influenced by the processing of other cases.
  • Can be influenced by the general state of a business process.
  • These dynamics can be captured by **inter-case features**.
Examples of PPM research including an intercase perspective


MLS-ICE: A Load Point Inter-Case Encoding Framework for PPM

• The MLS-ICE framework enriches events with the load state of relevant “load points” in a business process
  • Load points can be physical locations, activities, etc.

• Can be configured in several ways, MLS-ICE framework includes:
  • Two approaches for deriving the load state of a single location in a business process
    • Number of cases currently processed at a load point
    • Number of cases in an optimal time window at each load point
  • Two approaches for identifying relevant locations in a business process
    • System-based load point state (all important locations in the system)
    • Case-based load point state (encodes the state at load points in close proximity)
Remaining trace prediction performance

Additional value of the MLS-ICE framework for remaining trace prediction

• Consistent performance gain for models that use features encoded using the proposed MLS-ICE framework

• Up to 5.1% compared to solely relying on intra-case features

• Up to 5.7% compared to using the “Senderovich” feature vector

Table 4.6: Prediction results for remaining trace prediction as evaluated using the normalized Levenshtein similarity. Models that use inter-case features during prediction and outperform a model that solely relies on using intra-case features in order to make predictions are given in bold. An underscore is used to indicate the overall best performing model per event log. The performance differences between models that use inter-case features and the baseline model that solely relies on intra-case features in order to make prediction are given within brackets.

<table>
<thead>
<tr>
<th>Feature vector</th>
<th>BPIC20</th>
<th>BPIC17</th>
<th>BPIC12</th>
<th>BPIC12-Sub</th>
<th>BAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-case</td>
<td>0.9326 (+)</td>
<td>0.6125 (+)</td>
<td>0.5930 (+)</td>
<td>0.6011 (+)</td>
<td>0.7204 (+)</td>
</tr>
<tr>
<td>Senderovich vector</td>
<td>0.9313 (+0.1%)</td>
<td>0.6100 (+0.3%)</td>
<td>0.5953 (+0.2%)</td>
<td>0.5953 (+0.6%)</td>
<td>0.7204 (+0.4%)</td>
</tr>
<tr>
<td>Last activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case based MLS-ICE vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act. Cases</td>
<td>0.9326 (+0.6%)</td>
<td>0.6137 (+0.1%)</td>
<td>0.6405 (+4.4%)</td>
<td>0.6405 (+4.6%)</td>
<td>0.7276 (+1.1%)</td>
</tr>
<tr>
<td>Opt. window</td>
<td>0.9281 (+0.5%)</td>
<td>0.6086 (+0.4%)</td>
<td>0.6178 (+1.7%)</td>
<td>0.6259 (+0.9%)</td>
<td></td>
</tr>
<tr>
<td>System based MLS-ICE vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act. cases</td>
<td>0.9334 (+0.1%)</td>
<td>0.6146 (+0.2%)</td>
<td>0.5385 (+3.6%)</td>
<td>0.6524 (+5.1%)</td>
<td>0.7217 (+0.5%)</td>
</tr>
<tr>
<td>Opt. window</td>
<td>0.9313 (+0.1%)</td>
<td>0.6144 (+0.2%)</td>
<td>0.4945 (+0.9%)</td>
<td>0.6356 (+3.5%)</td>
<td>0.7250 (+0.8%)</td>
</tr>
<tr>
<td>System and case based MLS-ICE vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act. cases</td>
<td>0.9327 (0.0%)</td>
<td>0.6101 (+0.4%)</td>
<td>0.5002 (+0.3%)</td>
<td>0.6217 (+2.1%)</td>
<td>0.7202 (+1.3%)</td>
</tr>
<tr>
<td>Opt. window</td>
<td>0.9327 (0.0%)</td>
<td>0.6103 (+0.2%)</td>
<td>0.5020 (+0.1%)</td>
<td>0.6454 (+4.4%)</td>
<td>0.7260 (+0.9%)</td>
</tr>
</tbody>
</table>
Remaining time prediction performance

Additional value of the MLS-ICE framework for remaining time prediction

• Consistent performance gain for models that use features encoded using the proposed MLS-ICE framework
  
  • Up to 9.5% compared to solely relying on intra-case features
  
  • Up to 6.9% compared to using the “Senderovich” feature vector

<table>
<thead>
<tr>
<th>Feature vector</th>
<th>DPIC20</th>
<th>DPIC17</th>
<th>DPIC12</th>
<th>DPIC12-Sub</th>
<th>DAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-case</td>
<td>6206 (+)</td>
<td>11882 (+)</td>
<td>5219 (+)</td>
<td>5036 (+)</td>
<td>526 (+)</td>
</tr>
<tr>
<td>Last activity</td>
<td>6241 (-0.5%)</td>
<td>11672 (+1.8%)</td>
<td>5404 (-3.5%)</td>
<td>5541 (+6.7%)</td>
<td>525 (+0.2%)</td>
</tr>
<tr>
<td>Case based MLS-ICE vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act. cases</td>
<td>6279 (-1.2%)</td>
<td>11667 (+1.8%)</td>
<td>5236 (-0.3%)</td>
<td>5549 (+6.5%)</td>
<td>523 (+0.5%)</td>
</tr>
<tr>
<td>Opt. window</td>
<td>6228 (-0.4%)</td>
<td>11756 (+1.1%)</td>
<td>5179 (+0.8%)</td>
<td>5774 (+2.7%)</td>
<td>529 (+1.1%)</td>
</tr>
<tr>
<td>System based MLS-ICE vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act. cases</td>
<td>6273 (-1.1%)</td>
<td>11670 (+1.8%)</td>
<td>5029 (+3.6%)</td>
<td>5403 (+0.9%)</td>
<td>527 (-0.2%)</td>
</tr>
<tr>
<td>Opt. window</td>
<td>6234 (-0.5%)</td>
<td>11676 (+1.7%)</td>
<td>5156 (+1.2%)</td>
<td>5472 (+7.8%)</td>
<td>522 (+0.8%)</td>
</tr>
<tr>
<td>System and case based MLS-ICE vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act. cases</td>
<td>6240 (-0.5%)</td>
<td>11827 (+0.4%)</td>
<td>5169 (+1.0%)</td>
<td>5507 (+7.2%)</td>
<td>522 (+0.8%)</td>
</tr>
<tr>
<td>Opt. window</td>
<td>6164 (+0.7%)</td>
<td>11752 (+1.1%)</td>
<td>5406 (-3.6%)</td>
<td>5375 (+9.5%)</td>
<td>521 (+1.0%)</td>
</tr>
</tbody>
</table>

Table 4.7: Prediction results for remaining runtime prediction evaluated using MAE. Models that use inter-case features during prediction and outperform a model that solely relies on using intra-case features in order to make predictions are given in bold. An underscore is used to indicate the overall best performing model per event log. The performance differences between models that use inter-case features and the baseline model that solely relies on intra-case features in order to make prediction are given within brackets.
Challenge 3: Key takeways

• The inter-case featurisation / inter-case prediction model learning problem is far from solved

• Inter-case featurisation and prediction requires:
  1. Even more robust evaluation setups, cfr. challenge 1
     • Debiased test set
  2. Capable model learning architectures required (high-dimensional, dynamic event attributes)
     • Transformers?

• How to characterize the system?
  • What about context? E.g. IoT
  • What about the object-centric perspective?
  • What about interprocess dependencies?
Challenge 4: From case-level to model-level predictions
Process Model Forecasting (PMF)

- Shift from operational to tactical decision support


Turning event logs into DF time series

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Activity</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a_1)</td>
<td>11:30</td>
</tr>
<tr>
<td>1</td>
<td>(a_2)</td>
<td>11:45</td>
</tr>
<tr>
<td>1</td>
<td>(a_1)</td>
<td>12:10</td>
</tr>
<tr>
<td>1</td>
<td>(a_2)</td>
<td>12:15</td>
</tr>
<tr>
<td>2</td>
<td>(a_1)</td>
<td>11:40</td>
</tr>
<tr>
<td>2</td>
<td>(a_1)</td>
<td>11:55</td>
</tr>
<tr>
<td>3</td>
<td>(a_1)</td>
<td>12:20</td>
</tr>
<tr>
<td>3</td>
<td>(a_2)</td>
<td>12:40</td>
</tr>
<tr>
<td>3</td>
<td>(a_2)</td>
<td>12:45</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Directly-follows</th>
<th>Equitemporal</th>
<th>Equisized</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;_L_s (a_1, a_1))</td>
<td>[0,1,0]</td>
<td>[1,0,0]</td>
</tr>
<tr>
<td>(&lt;_L_s (a_1, a_2))</td>
<td>[1,1,1]</td>
<td>[1,1,1]</td>
</tr>
<tr>
<td>(&lt;_L_s (a_2, a_1))</td>
<td>[0,1,0]</td>
<td>[0,1,0]</td>
</tr>
<tr>
<td>(&lt;_L_s (a_2, a_2))</td>
<td>[0,0,1]</td>
<td>[0,0,1]</td>
</tr>
</tbody>
</table>

**Equitemporal**: 12:45-11:30 = 75 minutes
3 intervals of 25 minutes:

**Equisized**: 9 events: 3 intervals of 3 events
“Predict” the future DFG

<table>
<thead>
<tr>
<th>Directly-follows</th>
<th>Equitemporal (encoding)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt;_{ls} (a_1, a_1)$</td>
<td>$[0,1,3,2,2,4,5,5,1,8]$</td>
</tr>
<tr>
<td>$&lt;_{ls} (a_1, a_2)$</td>
<td>$[1,1,1,1,1,3,1,1,6,0]$</td>
</tr>
<tr>
<td>$&lt;_{ls} (a_2, a_1)$</td>
<td>$[0,1,0,3,6,2,4,1,0,2]$</td>
</tr>
<tr>
<td>$&lt;_{ls} (a_2, a_2)$</td>
<td>$[0,0,1,0,0,1,0,0,0,1]$</td>
</tr>
</tbody>
</table>

DFG of the past

DFG of the future
Process Model Forecasting

Italian help desk dataset, https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb
Process Change Exploration tool
Challenge 4: Key takeaways

• Process Model Forecasting (PMF) can predict the to-be process model (in the future)
• Could yield opportunities to apply predictive modelling at a different level of granularity
  • Answering different questions
• Current technique relies on simple univariate time series modelling
Challenge 5: Increasing adoption
Pathways to increased adoption

1. Uncertainty
2. Interpretability
3. Robustness
4. Prescriptive process monitoring
5. Data scarcity
1. Uncertainty

- Bayesian Neural Networks
  - Allow to estimate epistemic and aleatoric uncertainty
  - We developed a PPM technique for this purpose

- Models allow to predict point value together with the uncertainty (confidence interval)
  - Allows for enriched symbiosis of automated and manual decision making
  - One can also apply PPM to smaller datasets

- Conformal prediction?

---

2. Interpretability

• XAI tailored to PPM seems a must to have any chance at improved adoption

• Only few works in PPM already addressed the problem directly:
  • Stevens, A., & De Smedt, J. (2023). Explainability in process outcome prediction: Guidelines to obtain interpretable and faithful models. European Journal of Operational Research
3. Robustness

• We need to build trust
• Show that PPM models are robust
• E.g. robustness against adversarial attacks

• Adversarial training as a proactive defense mechanism
  • Warmly invited to attend our presentation on Tuesday at 2pm
4. Prescriptive process monitoring

• How far can we get with PPM?
• Shouldn’t we build models that can tell us “what to do” instead of “what will happen”?

• Difficult to demonstrate effectiveness in offline setting
  • How to define “correct” counterfactuals?
  • Difficult to manage complexity (isolate decision, intervention timing, intervention types, resource constraints, etc.)

• Online setting: a variety of challenges

• Industry Keynote Marlon Dumas
  • “Walking the Way from Process Mining to AI-Driven Process Optimization”
  • Wednesday at 1.30pm
5. Data scarcity

• Current datasets often used in PPM research are not necessarily very significant to the problem
  • E.g. labels used for outcome prediction

• If we want to grow as a community, we should take inspiration from related fields
  • In ML domains ranging from NLP to graph learning, there exist a wide variety of benchmark datasets and agreed upon evaluation approaches

• This should bring not only better techniques/models, but also a better recognition in practice

• Data privacy and other AI regulations might have an important negative impact, with a large number of companies already totally opposed to any form of data sharing
Conclusion
Conclusion

• Five key challenges in Predictive Process Monitoring (PPM)
  1. Strategies for PPM evaluation
  2. Generalization in deep learning models
  3. The inter-case perspective
  4. From case-level to model-level predictions
  5. Increasing adoption
Questions